Letter from the Editors

In this issue, we are highlighting IBM’s strategy on cloud computing. Our feature article is written by Frank De Gilio, IBM’s Chief Architect for Cloud. Frank’s visage is seen peeking from around a server on the cover of z/OS Hot Topics.

As we know, “every cloud has a silver lining,” and this issue is no exception. Our issue is rich with content, including several articles on z/OSMF.

We continue our story with Part 2 of “Life in the fast lane.” Since our last issue, there have been several enhancements to the SMC Applicability Tool (SMCAT).

We also include two articles on z/OS migration, by authors Marna Walle and Shigeki Kimura.

We are pleased to welcome several new volunteers to the Hot Topics editorial staff, as well as a number of other volunteers on our website, in our creative brainstorming, and on graphics and production work. A sincere thanks to all who made this issue memorable.

The Editors
mhvrdfs@us.ibm.com

Let chaos storm! Let cloud shapes swarm! I wait for form.
– Robert Frost
Contents: Featured articles

4 Becoming a Hybrid Cloud hero in three easy steps
BY FRANK DE GILIO

7 Connecting System z with the Cloud
BY QAIS NOORSHAMS, ACHIM STÄBLER, AND ROBERT VAUPEL

10 z/OSMF zEvent
BY ROBERT VAUPEL AND QAIS NOORSHAMS

13 z/OSMF workflows
BY LI BO, HONG MEI ZHAO, AND HONG LIANG ZHAO

15 Five reasons why PTF UI90034 will make you love z/OSMF
BY MATTHEW HUNGER AND MICHAEL YOUNG

16 Contributing to the Open Mainframe Project’s analytics solution
BY EDRIAN IRIZARRY AND CHRIS BROOKER

19 Wrangling resources
BY ANTHONY GIORGIO AND JUSTIN MCCOY

21 Is life in the even faster lane for you?
BY DAVID HERR, GUS KASSIMIS, AND JERRY STEVENS

23 Taking the direct path with Shared Memory Communications – Direct Memory Access
BY DAVID HERR, RANDALL KUNKEL, AND JERRY STEVENS

26 Stil on IBM HTTP Server powered by Domino?
BY MARNA WALLE

28 PFA extends its cognitive capabilities
BY KARLA K. ARNOLD

30 Find z/OS IT information quickly
BY GEOFFREY K. SMITH AND JODI EVERDON

32 MFA to the rescue!
BY ROSS COOPER, BRUCE WELLS, AND SCOTT WOOLLEY

34 How to customize TSO/E LOGON using parmlib
BY ERIC ROSENFELD

35 Three steps to business continuity with GDPS/Active-Active Sites
BY PIERRE CASSIER, ERIC CHABERT, AND STEVE WALL

38 Data migration made easy
BY GLENN WILCOCK

40 Too good NOT to share
BY MICHAEL GIERLACH, RANDALL KUNKEL, AND DHANANJAY PATEL

43 Is your PKI ready for SHA1’s retirement?
BY WAI CHOI AND BOB GENSLER

46 Let’s configure z/13, z/13s, and LinuxONE with HCD
BY SIGRUN ELSNER, MARCO SELIG, AND MICHAEL GROETZNER

48 z/OS reporting enhancements for Mobile Workload Pricing
BY PETER MAILAND, HORST SINRAM, AND DR. IRENE STAHL

52 New LPAR capping options
BY KARIN GENTHER, HORST SINRAM, AND GUENTER VATER

55 Handing over manually-activated processor capacity to z/OS CPM for later deactivation
BY DANIELA IMMEL AND PETRA GNECH

56 Tired of logging on to multiple TSO/E sessions to perform identical tasks?
BY RITA BEISEL, SHERI DEGROODT, AND BOB PETTI

59 Hurry up and HyperSwap
BY DONNA STENGER AND TABOR POWELSON

62 Access and manage z/OS data sets with Cloud REST API
HE JIANG JIA, JIAN GUO CHEN, AND XIAO ZHEN ZHU

64 Whittling down dumps
BY GIRIJA VARANASI, PURVI PATEL, AND RALPH SHARPE

66 RMF keeps a close eye on zFS performance
BY KARIN GENTHER AND PETER MAILAND

68 Some changes in z/OS V2R2 that you need to know before you migrate
BY SHIGEKI KIMURA

70 Thin provisioning for z/OS
BY NICK CLAYTON, LISA GUNDY, AND CAROL MELLGREN

72 2015 US and Canada IBM Master the Mainframe contest
BY TROY CRUTCHER

74 Our contributors
Becoming a Hybrid Cloud hero in three easy steps

BY FRANK DE GILIO

Becoming a hybrid cloud hero is easier than you think. As cloud moves from being merely another set of servers to a new set of services that can provide some cool capability, we can return IBM® z Systems™ to a level of prominence in the data center.

The interesting thing about services is that application developers, as users of the services, have no need to know or be concerned about the underlying infrastructure of a service.

All they care about is the ability of a service to perform some task or provide some data quickly, reliably, and securely at low cost. In this world of services, all of the arguing over servers compared with System z® can come to an end. The solution that CIOs have yearned for is here.

Imagine a world where the CIO wants to get rid of x86 servers in favor of the much more economical System z® solution; it is no longer science fiction. The power and capabilities of z Systems are made-to-order for enterprises that are trying to take advantage of an API economy. To move to a services API, all you need to do is follow these three easy steps:

Step 1: Look at cloud differently
As soon as you see cloud as a method for deploying services rather than servers, you can start playing in a league that leverages the differences and benefits of your approach. Applications that rely on services do not care about the underlying infrastructure. Creating new instances of a service can be as simple as creating a new task or address space on z/OS®. This takes advantage of the associated benefits of an inherent infrastructure of isolation and protection for and from other instances.

Unlike other cloud models that require a complete stack in a distributed environment, playing in this world does not require having to manage your system differently from how you already have been. Businesses that have created z/OS® services have demonstrated that they can save money and gain interest from clients who are traditionally focused only on distributed solutions. Utility and business services that run on z/OS® can take advantage of the capabilities of the hardware and operating system to do things that are either difficult or expensive in a distributed space.

Step 2: Look at applications differently
We tend to think of applications as a path of code running in the data center that leads the user through a set of activities to a result. Today, that path is much closer to the user and will probably only partially leverage the content that you create.

Instead of trying to force an entire path, today you have to create consumable services that can be called from cloud applications. These services are smaller and are built around the components of the value chain that makes up the traditional applications. The neat thing is that these services are hiding in your current applications. Bringing them to life can be easier than you think.

For example, consider an HR application that provides all of the information about a specific employee. The application might provide all the details of an employee based on simple search terms. This application could be the basis for a number of services that would not only provide data, but can also provide information without exposing the underlying data. You might start with an initial set of services focused on providing information related to employees such as:

- **Find_employee_by_name** — This service could return a list of employees by name.
- **Find_employees_management_chain** — This service could return the chain of managers up from the employee.
- **Find_employees_contact_information** — This service could provide email, phone number, web presence, and social media links.

There is nothing groundbreaking about these services. In fact, they can easily be derived from existing code. The value is not in the uniqueness of the service, but rather in the exposing of the capability through a services API in a more granular form.

Having access to such services lets an application retrieve or access small pieces of data that it needs rather than pulling a great deal of information; most of which is superfluous and ignored. The services are efficient and powerful and can be used by multiple applications. In fact, you might want to make services like Find_employees_management_chain even more granular. Each of these services can easily be derived from the existing application.

After these common data extraction services are created, you can start to build services that use the data to provide information that you do not want to expose to an application. For example, you might want to have a service like Is_Eligible_for_ProgramA which would determine if a person is eligible for ProgramA.
It doesn’t matter what ProgramA is or which of the ProgramA criteria the individual fits. In fact, all of the details that qualify someone for ProgramA are hidden from the caller. The only thing you know is whether a person qualifies or not. The neat thing about such a service is that it is abstract and has a fair amount of smarts while not exposing the actual data. Rather, the service exposes information that has been derived based on the data. In this case, the information is an answer to a fairly simple question, but you can imagine much more complex answers based on the myriad of data that already exists in the enterprise.

After you embrace the idea of creating services the next question becomes: "How difficult is it to create a REST-based service from existing 3270 applications? It turns out that creating a REST-based implementation from existing 3270-based code should be fairly simple. This is especially true if you have separated the presentation logic from the display logic. Any code that has been generated since the development of Web programs is fair game to become services.

Of course, you don’t need to have a ton of web programs to create services on z/OS. Any piece of code can be turned into a service, even if you have comimgled the business logic and the user interface. In fact, the user interface calls might help you define appropriate services. The key is to separate that particular piece of code into a service and replace the 3270 calls with web-based code.

**Step 3: Look at operations differently**

One of the biggest changes of the move to cloud is the rise of DevOps (a compound of development and operations). DevOps refines the role of operations. In the past, development has created code and made it available to operations, which is focused on making the code available and keeping it available.
In the DevOps world, development is responsible for creating a function and ensuring its availability, while operations is focused on ensuring that the infrastructure (everything surrounding the code) is available. This gives z/OS a huge advantage because of the inherent capabilities of the platform. It requires a different focus on part of operations. Herein lies the challenge for IT people who have been managing operations the same way for decades.

Moving to a services-based model can affect operations in other ways as well. One of the key differences is in the way they will collect money for operations. The established model is based on a “tax” approach. The tax approach splits or shares the cost of processing distributed across all of the users of the system. This means you pay a share of the cost regardless of the amount you are using the system. This has two drawbacks:

- If you are using less of the system, you are unfairly charged based on your consumption.
- If the number of people using the system drops, your tax increases even though you aren’t getting any additional value from the system.

The tax model has been in place since the creation of the data center back in the ‘60s.

The cloud model (of charging) is based on usage. When a service is created and the cost for a single invocation of that service is established, you can charge for the number of times a service is used. Thus, each consumer is charged based on usage. This, of course, requires more than a technical change. Accounting methods have to change. The culture that is based on the tax approach must be replaced or modified. You are probably thinking that this is a big hurdle to overcome. The easiest way to start is to be able to show exactly what is consumed. This, of course, is easier with z/OS because of the tools already at your disposal. By cutting the right SMF records, you can document who used which resource and when. SMF becomes the key to a facilitating a usage-based model.

As development focuses on keeping their services more available, operations has to focus more on managing the lifecycle of service instances. You will have to create and remove instances of the service. You will also have to be able to monitor and modify existing services.

Fortunately, there are tools to help manage those instances. The latest version of z/OSMF provides methods that you can use to create and manage instances of a service. The latest version of z/OS Connect provides a simple way to connect endpoints for services. These two tools let you connect to existing services catalogs that are employed in your environment. These two tools are the z/OS service contact points to the larger world.

**Hybrid cloud is essential to success**

The IT world has been completely changed by cloud. As businesses focus on combining a number of different services into an app that runs in a web or mobile environment, the need for enterprise-level services becomes essential to making your company stand out. Enterprises can win in this new high-speed agile world if they can leverage their existing assets in a new way. When you provide those existing business assets as services to cloud applications, you become the enterprise’s hybrid cloud hero.

The interesting thing about services is that application developers, as users of the services, have no need to know or be concerned about the underlying infrastructure of a service.
Connecting System z with the Cloud

Developing a multi-tenant notification service on Bluemix using z/OSMF zEvent

BY QAIS NOORSHAMS, ACHIM STÄBLER, AND ROBERT VAUPEL

The IBM Bluemix® cloud-based platform is an important aspect of the new IBM z/OS Management Facility (z/OSMF) zEvent innovation. z/OSMF zEvent is a mobile notification and monitoring app to alert system administrators of critical events in their z/OS environment. The Representational State Transfer (REST) zEvent API on Bluemix is a central part of this process; it connects z/OS installations with the mobile device to deliver important notifications.

Design
The goal of the zEvent API is to provide a simple mechanism to send push notifications from a z/OS installation to a mobile device. To that end, the zEvent design combines client management aspects with uniform access to proprietary push services that are abstracted from the client. The API is tied to the zEvent app that is available in app stores such as Apple App Store or Google Play.

The following list describes the main design principles and strengths of z/OSMF zEvent:

• Multi-tenancy
Multiple clients can use the zEvent API because it hides specific push service information, such as the push service credentials. Furthermore, a reservation mechanism prevents a client from sending messages to the device of another client. After a client uses the device to send a message, that client is only able to send messages to this device.

• Hiding end-to-end connectivity
Using third-party push services is critical in z/OS environments. They can reveal important organizational information about a client, and show details about which device is receiving notifications from what system. The zEvent approach hides the end-to-end connection information from the third-parties because the message is routed through neutral IBM servers.

• Transparency and security
To deliver the notifications, the zEvent API relies on the push services of Apple (APNS), and Google (GCM). For the client, the usage is transparent because the correct target is determined automatically by using a push token (the address of the device) for the specific app.

The payload and actual content of the notification is not accessible through the process; it can be encrypted on the host long before it leaves the system, and the API receives the message. The goal of the zEvent API is to provide a simple mechanism to send push notifications from a z/OS installation to a mobile device.

RESTful API
The zEvent API base URL ibm-zevent.mybluemix.net provides the following methods to accept JSON payloads, based on the order in which they are used:

• POST /clients
This method is the starting point and allows the client to be registered. The input is the name and email of a client. If the registration is successful, the method returns the new client credentials (the client ID and client secret), required to send messages. The client information is stored in a back-end Cloudant® database.

• POST /messages?clientId=<id>&clientSecret=<secret>
After a client is registered, this method is used to send messages to the device. The sending of messages is limited to the zEvent mobile app and requires a specific payload (that is, the push token that identifies the device and information of the event or chart that is sent to the device).

• DELETE /associations/<token>?clientId=<id>&clientSecret=<secret>
A mobile device is identified by its push token. A push token that is used by one client will be reserved for this client for security reasons, and ensures that a client sends messages to only the organization’s devices. This reservation is stored as an association in the Cloudant database and can be released with this method.

RESTful API
The zEvent API base URL ibm-zevent.mybluemix.net provides the following methods to accept JSON payloads, based on the order in which they are used:

• POST /clients
This method is the starting point and allows the client to be registered. The input is the name and email of a client. If the registration is successful, the method returns the new client
credentials (the client ID and client secret), required to send messages. The client information is stored in a back-end Cloudant database.

- **POST /messages?clientId=<id>&clientSecret=<secret>**
  After a client is registered, this method is used to send messages to the device. The sending of messages is limited to the zEvent mobile app and requires a specific payload (that is, the push token that identifies the device and information of the event or chart that is sent to the device).

- **DELETE /associations/<token>?clientId=<id>&clientSecret=<secret>**
  A mobile device is identified by its push token. A push token that is used by one client will be reserved for this client for security reasons, and ensures that a client sends messages to only the organization’s devices. This reservation is stored as an association in the Cloudant database and can be released with this method.

**Registration and notification process**
The API uses only a few methods to send messages. The process is shown in Figure 1.

1. First the client administrator requires credentials for the zEvent API.
   a. The process begins when an administrator of a client registers at the zEvent API using the method /clients.
   b. The response of the API contains the API credentials (client ID and client secret).
   c. The credentials are stored on the z/OS system.

2. Then, the zEvent app is configured to receive notifications
   a. A mobile device that should receive notifications registers for push notifications at the push service using the zEvent app.
   b. The push service replies with a push token (that is, address) that identifies the app for this device.
   c. The push token is stored on the z/OS system, which is configured to send certain messages to this device.

3. Finally, the message is sent using Bluemix to the device.
   a. If an important event occurs, the z/OS system uses the credentials to authenticate with the zEvent API and sends the message to the device with the specific push token using the method or messages. The message is contained in the payload and may be encrypted.
   b. The zEvent API checks if the credentials are valid and if the push token is not already reserved by another client. Then, the push token is used to identify the correct push service (APNS or GCM).
   c. Finally, the message is sent to the device and is displayed as a push notification. The zEvent app can be used to decrypt and read the message and to retrieve further information.

**Implementation and deployment**
The zEvent API is implemented as a stateless RESTful Bluemix application that consumes and produces JSON. It uses the Node.js buildpack as a basis for its run time. The following services fulfill the zEvent API architectural and operational requirements:

- Monitoring and Analytics service detects exceptions and other hiccups, and reviews logs and traces.
- Auto-scaling service scales out horizontally with increasing load.
- The Cloudant NoSQL database persists push tokens and client metadata.

---

The goal of the zEvent API is to provide a simple mechanism to send push notifications from a z/OS installation to a mobile device.
Figure 2 displays how the zEvent API appears on the dashboard on Bluemix.

zEvent API is a Cloud Foundry application. Configuration properties are kept in the environment, and user-provided service instances provide push certificates and service credentials.

The zEvent API uses the Express® framework to implement the HTTP interface, and third-party implementations to communicate with vendor push notification services. Swagger is used to provide API documentation to developers.

The zEvent API implementation has the following characteristics:

- Mocha and Chai JavaScript test frameworks are selected for a test-driven approach (total coverage is greater than 99 percent and close to 100 percent).
- Proxyquire and SinonJS are used to test all components in isolation.
- SuperAgent is used to test integration.
- Istanbul is used to create code coverage reports.
- Dependencies are managed with Node Package Manager (npm) through package.json and Shrinkwrapping.
- Task automation is done with npm.

The development and production environments are organized into spaces with the same code in both environments; it is only the configuration that distinguishes them. With a blue-green deployment implementation, you can deploy to production frequently and without service interruption.

The release process automates the following steps:

- Validate that all unit tests pass.
- Perform a blue-green deployment.
- Validate that smoke and integration tests pass on the newly deployed version.
- Switch production over to the new version.
- Let the previous version continue to run as a hot-standby.

This solution requires a load balancer to manage the two locations, the appropriate mechanisms for the deployment, and the back-end database synchronization.

## References and resources

For more information on z/OSMF zEvent, refer to the article, “z/OSMF zEvent, a rapid notification function to get immediately informed on critical system events” (page 10).

For more information on the Bluemix services and frameworks, refer to the following websites:

- Auto-scaling service: console.ng.bluemix.net/catalog/services/auto-scaling/
- Cloudant NoSQL database: console.ng.bluemix.net/catalog/services/cloudant-nosql-db/
- Monitoring and analytics service: console.ng.bluemix.net/catalog/services/monitoring-and-analytics/
- Node.js buildpack: console.ng.bluemix.net/catalog/starters/sdk-for-nodejs/
- Blue-green deployment – Jez Humble and David Farley: Continuous Delivery (Chapter 1), Addison Wesley
- Chai: chaijs.com/
- Cloud foundry: cloudfoundry.org
- Express: expressjs.com/
- Grunt: gruntjs.com/
- Gulp: gulpjs.com/
- Istanbul: gotwarlost.github.io/istanbul/
- npm: npmjs.com/
- Mocha: mochajs.org/
- Proxyquire: github.com/thlorenz/proxyquire
- Shrinkwrapping: docs.npmjs.com/cli/shrinkwrap
- Sinon.JS: sinonjs.org/
- SuperAgent: visionmedia.github.io/superagent/
- Swagger: swagger.io

## Limitations

Although this method of release is already in use, there are some aspects to consider:

- The deployment of the zEvent API is limited to a single data center region. While this deployment is sufficient for a data center that is always available, certain locations might experience load peaks and other influences that can affect the services in the data center. A more robust solution might be to deploy the API in multiple regions; thus, in multiple data centers.

Figure 2. Screen capture of the dashboard on Bluemix
Presenting at customer sites is an educational experience for both the customer and the presenter. Oftentimes during a presentation, the audience is distracted by people who enter the room. The reason for this distraction is that something occurred in their z/OS environment that requires the immediate attention of a specific system administrator who can resolve the problem they are experiencing on z/OS.

What if it is possible to notify the correct people of such an event all the time without the need to search for them? What if they were able to check the details about the incident immediately no matter where they are? This scenario is possible by directing event notifications to the mobile phones of the correct system administrator. With this application, they can log on to monitoring environments on their host systems. In today’s business world, it is truer than ever that the time between the occurrence of a critical event and the time a resolution is required is so little that a rapid notification that helps to solve the problem is crucial. A rapid notification to solve a potential problem is critical in today’s world.

The solution
IBM z/OSMF zEvent is the solution to the problem. It is a notification and monitoring app that you can use to manage IBM z/OS systems on a mobile device. It helps you react quickly to critical system messages, process failures, or resource shortages.

zEvent receives and displays system events and charts that are sent by push services. It provides system administrators with essential information that they can use to assess and respond to situations. For example, based on the information that is provided by the zEvent app, the administrator can decide whether a system event requires immediate attention.

A notification can be enriched with context information. With this information, which is coupled with zEvent’s integrated interfaces to z/OS tools, such as IBM Resource Measurement Facility (RMF) and IBM z/OS Management Facility (z/OSMF) Resource Monitoring, Job status, and Workflow Management, you can drill down and pinpoint the system message source. Using these monitoring facilities, you can then decide whether a direct reaction at your workspace is required.

The app is focused on z/OS systems and is tailored to functions provided by z/OS, which is a prerequisite to exploit the full capabilities of this app.

Architecture and components
zEvent consists of the following parts:

- The mobile app that is available for iOS and Android mobile phones
- The host component that is an integral part of z/OSMF 2.2
- The push services that use Bluemix cloud services and exploit Apple Push Notification Services and Google Cloud Messaging Services.

Mobile app and registration to use push services
You can download the mobile app from the Apple app store for iOS mobile devices and from Google Play store for Android mobile devices. After you install the application, register the mobile phone to receive push notifications. The push token must then be sent to you as the z/OSMF system administrator to complete the registration of the new device and user ID on the z/OS system. You can also add the new user ID to one or multiple user groups, like z/OS or CICS administrators who receive specific event notifications related to their job tasks.

You can ease the registration process by preparing and providing a QR code, which contains the project ID and email address where the registration is sent. With the zEvent mobile app, you can both generate and scan QR codes.
Host components
On z/OS, the user administration, as well as the application interface to push notifications, are an integral part of z/OSMF. In addition, you can find samples on the zEvent website that help to push event notifications out of Message Processing Facility (MPF) exits, and from RMF.

The z/OSMF notification API is a Representational State Transfer (REST) web service that can be used by any program on z/OS. In addition, a sample REXX stub that is called zEvent is available to use the push mechanism out of REXX procedures.

Bluemix and push services
A Bluemix-based API is provided as a uniform interface to the push services to send notifications to the device. You can encrypt the content of the notification on the host by using a standard AES encryption algorithm, which ensures that no third-party component is able to read the content. The use of Bluemix and the push services is described in detail in the article “Connecting System z with the Cloud – Developing a multi-tenant notification service on Bluemix that uses the z/OSMF zEvent as example” on page 30.

Pushing and receiving notifications
After you register the mobile device and set up the event notification on z/OS, push notifications are sent to the mobile device whenever the specified situation occurs. zEvent supports the following types of notifications:

1. Message-like event notifications: Such events can be pushed out of MPF exits and they are derived from z/OS system messages. The idea is to push only critical messages that require the attention of that mobile user.

Another possibility is to set up a monitor on z/OS, which extracts information continuously from RMF and tests for specific situations that occur on the host and require attention of a system programmer. For example, suppose that a defined capacity limit is approaching or was reached, and that the capping of the z/OS system affects important work on the system. All such information can be extracted and evaluated by the Monitor III batch function, which can also be downloaded from the zEvent website.

2. The second set of events are charts that provide regular status information about resources, workloads, or a set of jobs running on z/OS.

For the example in notification method 1, it would be possible to send the MSU consumption for the last ten RMF intervals continuously to the mobile phone, or to display the performance index of the most important service classes after the defined capacity limit is reached.

Figure 1 shows the notification path from the z/OS system to the mobile phones. The events are generated by MPF exits or z/OSMF and passed to the z/OSMF notification API where they are encrypted and sent to Bluemix. From there, they are passed on to the push services, and then to the mobile phones where the messages are decrypted, the user is notified, and then the information is displayed.

Notifications are events that are similar to system messages or charts, which contain information about resources or workloads running on the host system.

Accessing z/OS for additional information
After you receive an event notification, you can get back to the z/OS system and retrieve additional information about the event. For this purpose, you must first set up the host access by defining the logon credentials in the Connection set-up area of the zEvent app. You specify a user ID and password, or only a user ID if the password is entered every time that the connection is opened. If the IT organization uses Multi-Factor Authorization to obtain a pass ticket for logon, it is also possible to pass this to zEvent.
For a secure connection to the z/OS system, it might be necessary to set up a VPN session first; refer to Figure 2. For example, if the system is accessed from home, it is possible to link directly from a specific event to a monitoring function that is integrated in zEvent. With zEvent, you can use the following monitoring functions:

- **z/OSMF Resource Monitoring (RM)**, to retrieve dashboards that are related to the event
- **RMF Monitor III web monitoring interface**, to either receive dashboards or complete reports for further investigation
- **z/OSMF Job log display**, to obtain information for specific jobs
- **z/OSMF Workflow content**, to monitor the progress of a z/OSMF workflow.

To link from an event to the monitoring functions, click the event. A pop-up window opens that offers the possible links. When you select the monitoring function, the app establishes a connection to the host and retrieves the information from the host monitor.

In our previous example, you could define a link from an event that is passed to the mobile phone when a capacity limit is reached to z/OSMF RM, which then displays dashboard “Performance Index” for the service classes running on the system.

**Resources and references**

To learn more, watch the zEvent video available on YouTube: [https://youtu.be/5Xa7zMCATzE](https://youtu.be/5Xa7zMCATzE)


The host samples include an ISPF interface for the user administration in case you are not using z/OSMF, examples for MPF exits, and the Monitor III batch function that you can set up as a monitor to retrieve information from RMF Monitor III and create events that are sent to mobile devices.

With zEvent, you can link directly from an event to a monitoring function that retrieves specific information to analyze a potential problem.
IBM z/OSMF workflows can guide the activities of anyone at your installation who is responsible for managing the configuration of the IBM z/OS system. Workflows provide a framework for configuration activities in the form of structured procedures.

z/OSMF workflows can simplify the completion of complex tasks and reduce errors in the following ways:

- Help users to perform tasks that they are unfamiliar with or perform rarely
- Ensure that all steps in a task are performed in the right order and only when their dependencies are met
- Ensure that all steps are completed, even if steps have been delegated to a number of different colleagues
- Monitor and track progress toward the completion of the task
- Provide a history (an audit trail) of the steps that are performed for a task
- Perform the same tasks on multiple systems, such as when you migrate your software to a new release

The basic unit of a workflow is a step. A step can:

- Provide instructions for completing a task, or start a wizard that updates and submits jobs, executes shell scripts, and runs REXX execs
- Define dependencies on other steps
- Be assigned to an individual or a specific role, such as systems programmer, or security administrator

To accomplish a task using a workflow:

1. The workflow owner creates the workflow instance with the Create Workflow action.
2. The workflow owner assigns steps to the appropriate users. A z/OSMF notification informs each user when a step is assigned to him or her.
3. The assignee accepts the assigned steps.
4. The assignee checks if the steps are ready to be performed.
5. The assignee performs the assigned steps.
6. The workflow owner checks the status of the steps and reviews the history.

**Use workflow automation to improve efficiency**

The workflow engine can run a single step, or an entire workflow, automatically. You specify automation in the workflow template. The workflow engine might require certain system variables. The values for the system variables can be:

- Supplied manually, by the user
- Discovered by the workflow engine. Discovery plays the role of a bridge between the workflow engine and z/OS.

The workflow template provides conditions that can control the flow of a workflow instance. The conditions can be variable values or the state of the workflow steps. A condition might specify that if the value of an expression is true, the workflow engine should run the step; otherwise, the workflow engine should skip the step and run the next step. When the conditions are satisfied, the workflow engine can automatically run every step of the workflow instance.

For example, a vendor provides a workflow to install a plug-in on z/OS. The plug-in installs two components, A and B. The workflow template consists of these steps:

1. Present installation instructions.
2. Verify whether component A is installed on z/OS by using a workflow discovery function.
3. Install component A if it is not installed.
4. Check whether component B is installed on z/OS by using a workflow discovery function.
5. Obtain the system variables that are required to install component B, by using a discovery function.
6. Install component B.
7. Verify the installation.

The user selects the template and creates the workflow instance. After reading the installation instructions, the user performs the second step. If component A is installed on z/OS, the expression value of the third step is false, so the workflow...
engine skips that step and performs step 4. If component B is not installed, the expression value of step 5 is true, so the workflow engine performs this step to obtain system variables on z/OS and set them to the variables of the workflow instance. Next, the workflow engine performs step 6 to install component B. The final step verifies whether the plug-in is installed successfully.

The whole process is completed automatically.

For the vendor, installing or upgrading their products is easier and the usability of their products is improved. For the user, this means saving time and effort, as well as lowering the probability of errors during the manual installation process.

Develop your application based on workflow REST API services
With z/OSMF, you can use Representational State Transfer (REST) APIs, which are public APIs that your application can use to work with system resources and extract system data. As with implementations of REST services on other operating systems, the z/OSMF APIs allow for services that are independent of language and platform, stateless, scalable, and easily parsed.

REST APIs add wings to workflow automation, because workflows can run on different systems on z/OS. When you call REST APIs to run workflow instances, you do not have to log on to the z/OSMF user interface, because this process is performed in the workflow template. The application can send a request to the workflow engine to create a workflow instance. After the workflow instance is created on the system, the application receives a success message that contains the new workflow instance’s location. The application can send a request to start the workflow instance, which then runs the workflow. When the workflow processing has completed or failed, the system sends a message to your application.

The workflow engine can also monitor the workflow on different systems and remove or cancel the unused workflow by using a REST API. For example, the workflow engine can monitor the workflow instance’s completion percentage, can check which step of the workflow instance is running, and can receive error messages to guide you in resolving problems. By using REST APIs, the workflow engine can run one or more jobs automatically on different systems.

Who is using workflows
IBM DB2® uses z/OSMF to automate the process of migrating DB2 subsystems to DB2 11 for z/OS. You can create a workflow definition and variable one time and reuse it across many different DB2 subsystems.

With the z/OSMF migration workflow, you can migrate to z/OS V2R1 by using an interactive, step-by-step process. Two workflows are available, one for the migration from z/OS V1R13 to V2R1 and one for the migration from z/OS V1R12 to V21R2. Similarly, workflows for migration to V2R2 are available. The workflows are available here:

Five reasons why PTF UI90034 will make you love z/OSMF

BY MATTHEW HUNGER AND MICHAEL YOUNG

IBM z/OS Management Facility Version 2 Release 1 is updated with the new PTF UI90034. This PTF adds features that were previously available only with z/OSMF V2R2, which greatly increases the function and usability of z/OSMF V2R1. In addition, the PTF incorporates a complete overhaul of the installation and configuration process, making the configuration of new z/OSMF instances a breeze. Here are just a few of the reasons why you’ll love z/OSMF after installing UI90034:

1. Configuration through parmlib
Finally, you can configure z/OSMF with a parmlib member, just as you configure many other z/OS software products. In member IZUPRMxx, using familiar parmlib syntax, you can specify which plug-ins your z/OSMF instance should deploy, and easily configure your z/OSMF server host name, as well as many other settings. Using IZUPRMxx replaces the old system of multiple UNIX configuration files and scripts, and consolidates the z/OSMF configuration into a single, easily modifiable location. Which leads us to the next reason you’ll love UI90034.

2. Goodbye UNIX scripts!
Previously, z/OSMF required administrators to run various z/OS UNIX shell scripts for installation and maintenance. The scripts are no longer required! Most configuration is handled automatically at server startup by a new z/OSMF started task procedure. Those few tasks that can’t be automated are now supplied as sample JCL jobs. For example, creating and mounting the z/OSMF file system used to be part of the izusetup.sh script. Now there’s a sample job for that, IZUMKFS. In addition, the improved security setup eliminates the use of REXX execs or shell scripts. In their place is an IZUSEC sample security job for the core configuration, as well as other sample jobs for plug-in security.

3. New user interface
The new z/OSMF user interface incorporates modern web UI design features that you can use with all modern web browsers. Now, when you update your browser, you won’t encounter compatibility messages. Plus, it looks great!

4. Multi-sysplex Incident Log
The Incident Log task now provides multi-sysplex support that you can use to manage the incidents for all of the sysplexes in your enterprise from a single z/OSMF instance. By using secondary z/OSMF instances, your remote sysplexes can report to your primary z/OSMF any issues that have occurred. This gives you more control than ever when monitoring your complex, multi-sysplex environments.

5. Cool new sysplex graphical displays
As z/OSMF gains more multi-sysplex functionality, understanding your installation’s topology becomes more important than ever. The z/OSMF Systems display can now show you and your administrators a graphical view of your sysplex topology. You can quickly visualize the layout of your sysplexes, groups, and CPCs with helpful color coding.

We hope this list has inspired you to apply PTF UI90034. We think you’ll love the improvements that it brings. To learn more about how to migrate your system, see http://ibm.biz/ptf-ui90034.
Contributing to the Open Mainframe Project’s IT analytics solution

BY EDRIAN IRIZARRY AND CHRIS BROOKER

This article introduces you to the Anomaly Detection Engine for Linux Logs (ADE), an IT analytics solution that has been contributed to the Open Mainframe Project. We will use ADE to create a model of what is “normal” from a set of historical logs. Then, we will analyze another set of logs, looking for anomalies. ADE uses various statistical methods to build the models and surface anomalies.

Downloading and building
To download and build, follow these steps:

1. To make a new directory in your home directory for the source code, enter these commands in the terminal window:
   
   mkdir lab-src
   cd lab-src

   Figure 1. Creating a new directory - an example

2. To clone the repository, enter the following command:

   git clone https://github.com/openmainframeproject/ade.git

   This command downloads the latest stable code from the GitHub repository.

3. To switch to the root directory of the source tree, enter the following command, using the directory name that was created in step 2:

   cd ade-directory

4. To make the required binaries, using the build instructions packaged with the source, enter this command:

   mvn clean package

   The build process prints a large amount of output to the screen. When it completes successfully, the system issues a BUILD SUCCESS message.

5. To copy the collection of binaries to a new location and unpack them for use, enter these commands:

   mkdir ~/lab-bin
   cp ade/ade-assembly/target/filename.tgz ~/lab-bin
   cd ~/lab-bin ‘tar –zxvf filename.tgz’

   Figure 2. Sample build process output

6. Download and unzip Apache Derby from this website:

   https://db.apache.org/derby/releases/release-10.12.1.1.cgi

7. To customize the bin/env.sh file, follow these steps:

   a. Open bin/env.sh in an editor such as vi.
   b. Update the DB_CLIENT_PATH variable to point to the path of the derby.jar file.

   export DB_CLIENT_PATH=/user/local/derby/db-derby-10.11.1.1-bin/lib/derbyclient.jar

   Figure 3. Updating the DB_CLIENT_PATH variable - an example

8. To customize the conf/setup.props file, follow these steps:

   a. Create a directory for the lab output. For example, enter this command: mkdir /home/student/lab-output
   b. In an editor, update the analysisOutputPath value to point to the directory you just created:

      /home/student/lab-output
Running the analytics
To run the analytics, follow these steps:

1. Change to the appropriate starting directory:
   `cd ~/lab-bin/starting-directory`

2. Create the database structure that is required by the analytics function:
   `bin/controldb create`

   The system displays a few lines that summarize what is happening. The last line ends with the following:
   `ControlDB succeeded. Total time elapsed: 00:04:05 (hh:mm:ss)`

3. To prime the database with the historical logs, enter this command:
   `bin/upload -o Linux -d ~/lab-historical-logs/`

   The upload process now runs through all of the logs in the archive file provided. It parses, summarizes, and creates metadata for all of the valid messages in the archive. Depending on the data, this will include creating database entries for days (periods), intervals (smaller windows of time within days), and unique systems. Each system is added to the model group named `default`.

4. Training is done against groups of Linux systems.
   To train the default model group, enter the following:
   `bin/train default date`

   At this point, all of the data loaded into the database is used to build a model of “normal” for the given group of systems. This process is typically quick and finishes with a message similar to the following:
   `Train succeeded. Total time elapsed: 00:00:12 (hh:mm:ss)`

5. To analyze the new Linux logs, enter the following:
   `bin/analyze -o Linux -d ~/lab-new-logs/`

   This process takes the most time to complete. However, you can jump to the output directory almost immediately and watch the output accumulate.

---

Figure 4. Switching to the output directory - an example
The output flows to ~/lab-output, where you can browse it.

To sum up

By completing these instructions, you have downloaded, built, and deployed the Anomaly Detection Engine for Linux Logs (ADE), which is part of the Open Mainframe Project. You can stop here and use this function to help find anomalies within your Linux system logs or you can choose to further enhance this function with new features—for example, new parsers to allow new log types to be analyzed, or a new scorer to change the anomaly detection. The possibilities are numerous!

References

For more information, visit the following websites.

- Open Mainframe Project: openmainframeproject.org
- ADE: openmainframeproject.github.io/ade/
Dynamic Resource Management with KVM for IBM z Systems

Protecting business objectives
Consider the following scenario: a business has a hypervisor with a single workload. This is a high-importance workload that drives the hypervisor at a constant CPU utilization rate of 25%. A second, lower importance workload is then started, which consumes as much CPU as it can. The result is that the first workload is given only half the CPU resources it was previously consuming, causing significant CPU delay.

When zHPM is installed on the hypervisor and allowed to automatically manage CPU resources, a different scenario occurs. When the second workload is started and begins to consume all the available CPU cycles, zHPM detects that the high-importance workload has begun to miss its goals. If the situation continues, zHPM redistributes CPU shares from the second workload to the first one. This can help the high-importance workload recover and continue to meet its goals. Figure 1 shows CPU utilization before and after setting up automatic management of resources.

Gold, Silver, and Bronze
To illustrate how zHPM manages workloads to meet business goals, consider the following example. A user defines three workload resource groups, and names them Gold, Silver, and Bronze, as shown in Figure 2. The most important business functions run on virtual servers belonging to the Gold workload resource group, with a business importance value of “highest.” Less important business functions are in the Silver workload resource group, with “medium” business importance. The least important business functions are in the Bronze workload resource group, with “low” business importance.

CPU Critical service class
Sometimes your business functions include workloads that have sudden spikes in CPU utilization. These workloads could possibly encounter high CPU delays due to contention with other workloads.

To avoid this problem, with zHPM, you can designate a service class as “CPU Critical.” This means that the virtual servers within the service class in question are closely monitored for sharp jumps in hypervisor CPU delay. If spikes are detected, zHPM immediately makes large adjustments to the CPU shares for the designated virtual servers. zHPM aggressively reacts to the workload spike and improves the workload performance towards its goal.
However, you must be careful with this setting. Only apply “CPU Critical” to a limited amount of workloads that are expected to have spikes in CPU utilization. If too many service classes are marked as “CPU Critical,” and they experience a sudden sharp spike in CPU delay, zHPM might not be able to make any effective changes. This is because any changes will be zero sum, meaning that some virtual servers are delayed to help others meet their goals.

**zHPM interfaces**

You can interact with zHPM in two ways. The first is the command line interface (CLI), which is accessed from the KVM hypervisor console. The zhpm command communicates with the zhpmd service, so that you can manage zHPM functions. For example, to display a list of all the running virtual servers on the system, use the command `zhpm vs-display`.

The second way to interact with zHPM is by using REST APIs. zHPM “listens” for HTTP requests, and performs actions on the user’s behalf. For example, to list all the workload resource groups on a hypervisor, GET request to the URL `/zhpm/wsapi/v1/workload-resource-groups`. This returns a JSON encoded string that lists every workload resource group. Figure 3 shows sample JSON output.

You can also manage your virtual servers by using APIs. If you wanted to move the virtual server VS1 to the Silver workload, you would use the following URL: `/zhpm/wsapi/v1/workload-resource-groups/{wrg-resource-id}/actions`

zHPM performs the action on your behalf, and responds with an appropriate return code.

The zHPM API is a very powerful tool, because you can use it to monitor and manipulate your workloads using whatever custom tooling you want. Whether it is enabled by using a Python script or a web application, you can manipulate zHPM to serve your business needs.

**Meeting your goals**

zHPM can assist you by making efficient use of resources and maintaining the highest possible throughput. It also works to ensure overall system responsiveness, and ultimately to meet service level agreements.

For more details on the various zHPM commands and APIs, see *KVM for IBM z Systems System Administration*, SC27-8236.

---

**Figure 3. Sample JSON output**

```
"workload-resource-groups": [  
   {  
      "wrg-info": {  
         "resource-uri": "/zhpm/wsapi/v1/workload-resource-groups/2da7caa6-1491-4062-90fa-cffe10c6ef5f",  
         "resource-id": "2da7caa6-1491-4062-90fa-cffe10c6ef5f",  
         "name": "Gold",  
         "description": "High priority work"  
      },  
      "wrg-info": {  
         "resource-uri": "/zhpm/wsapi/v1/workload-resource-groups/f1e5b6b3-088e-4850-874b-0655c5bd3af6",  
         "resource-id": "f1e5b6b3-088e-4850-874b-0655c5bd3af6",  
         "name": "zHPM Default Workload Resource Group",  
         "description": "zHPM Generated Default Workload Resource Group"  
      }  
   ]
```

---

```
"wrg-info": {  
   "resource-uri": "/zhpm/wsapi/v1/workload-resource-groups/sf5ef752-4437-4ew9-9ee8-496bc802f56a",  
   "resource-id": "sf5ef752-4437-4ew9-9ee8-496bc802f56a",  
   "name": "Silver",  
   "description": "Medium priority work"  
},  
"wrg-info": {  
   "resource-uri": "/zhpm/wsapi/v1/workload-resource-groups/c7cc0638-8c3d-8390-7679b2b3f72",  
   "resource-id": "c7cc0638-8c3d-8390-7679b2b3f72",  
   "name": "Bronze",  
   "description": "Low priority work"  
}
```
Is life in the even faster lane for you?

Enhancements to z/OS SMC-R Applicability Tools

BY DAVID HERR, GUS KASSIMIS, AND JERRY STEVENS

In a previous article, we discussed the SMC Applicability Tool (SMCAT) and its ability to project the potential benefits of Shared Memory Communications over Remote Direct Memory Access – SMC-R (see “Is life in the fast lane for you?” in z/OS Hot Topics Newsletter Issue 29, August 2015, GA32-0892-02).

Since the time that article was published, there have been several enhancements to the SCMCAT tool worth sharing.

SMCAT support for SMC-D

A new variation of Shared Memory Communications was introduced that significantly optimizes communications across LPARs in the same CPC.

This technology is called Shared Memory Communications – Direct Memory Access or SMC-D and is available on the z13™ and z13s processors. You can now configure SCMCAT to monitor TCP workload for a list of destination IP addresses or subnets and produce a report that can be used to project the potential benefits of SMC-D.

SMCAT requires no additional updates to support SMC-D. The configuration and activation of SCMCAT for SMC-D is the same as it is for SMC-R with the possible exception of the configured list of IP addresses or subnets to monitor. For SMC-D, the list of configured IP addresses and subnets would reflect TCP connection destinations that are either within the same CPC today or reflects z/OS instances that could be redeployed in the future. SMC-D can support intra-CPC traffic patterns that today flow over HiperSockets™ or OSA. You can find more information about SMC-D in the article “Taking the direct path with Shared Memory Communications – Direct Memory Access” on page 22.

The SMCAT monitoring report

The following enhancements for the SCMCAT tool are available in the following PTFs:

- z/OS V1R13: PTF UI31050
- z/OS V2R1: PTF UI31054
- z/OS V2R2: PTF UI31055

The SCMCAT monitoring report is enhanced to provide the application message-received sizes (see Figure 1). This enhancement, in addition to the application message-send sizes already provided, gives a full picture of the data as it moves over the monitored TCP connections. With this data, you can more accurately determine the potential benefits of SMC (SMC-R and/or SMC-D) when compared to using TCP/IP.

Figure 1. Sample SCMCAT report

To make this determination even easier, the SCMCAT monitoring report now includes Summary Report Export Area. This area of the report is an unformatted summary of all of your reported application send-and-receive data. The SMC protocol provides for improved latency for all message sizes. However, if you also have a significant amount of larger message sizes (8 KB or larger), you also have an opportunity for network related CPU savings. The Summary Report Export Area of the report (see Figure 2) provides another view that you can use to quickly visualize your network traffic. This export area also makes it easier for IBM to assist you with evaluating your results.
You can get a detailed analysis of these savings by sending the output of your report, including the SMCAT Summary Report Export Area, to David Herr at dherr@nc.rr.com.

SMCAT can show you why you need to get into the even faster lane. For more information about SMC-R, SMC-D and SMCAT see: ibm.com/software/network/commserver/SMCR

---

A new variation of Shared Memory Communications was introduced that significantly optimizes communications across LPARs in the same CPC.

---

<table>
<thead>
<tr>
<th>Connections meeting direct connectivity requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>20% of all TCP connections can use SMC (eligible)</td>
</tr>
<tr>
<td>95% of eligible connections are well-suited for SMC</td>
</tr>
<tr>
<td>18% of all TCP traffic (segments) is well-suited for SMC</td>
</tr>
<tr>
<td>20% of outbound traffic (segments) is well-suited for SMC</td>
</tr>
<tr>
<td>16% of inbound traffic (segments) is well-suited for SMC</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interval Details:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Total TCP Connections:</td>
<td>120</td>
</tr>
<tr>
<td>Total SMC eligible connections:</td>
<td>74</td>
</tr>
<tr>
<td>Total SMC well-suited connections:</td>
<td>22</td>
</tr>
<tr>
<td>Total outbound traffic (in segments):</td>
<td>110990</td>
</tr>
<tr>
<td>SMC well-suited outbound traffic (in segments):</td>
<td>22890</td>
</tr>
<tr>
<td>Total inbound traffic (in segments):</td>
<td>100990</td>
</tr>
<tr>
<td>SMC well-suited inbound traffic (in segments):</td>
<td>16590</td>
</tr>
</tbody>
</table>

Application send sizes used for well-suited connections:

<table>
<thead>
<tr>
<th>Size</th>
<th># sends</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1560 (&lt;1500)</td>
<td>2000</td>
<td>39%</td>
</tr>
<tr>
<td>4K (&gt;1500 and &lt;=4K)</td>
<td>1500</td>
<td>29%</td>
</tr>
<tr>
<td>8K (&gt;4K and &lt;= 8K)</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>16K (&gt;8K and &lt;= 16K)</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>32K (&gt;16K and &lt;= 32K)</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>64K (&gt;32K and &lt;= 64K)</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>256K (&gt;64K and &lt;= 256K)</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>&gt;256K</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>

Application receive sizes used for well-suited connections:

<table>
<thead>
<tr>
<th>Size</th>
<th># sends</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1560 (&lt;1500)</td>
<td>1500</td>
<td>32%</td>
</tr>
<tr>
<td>4K (&gt;1500 and &lt;=4K)</td>
<td>1250</td>
<td>27%</td>
</tr>
<tr>
<td>8K (&gt;4K and &lt;= 8K)</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>16K (&gt;8K and &lt;= 16K)</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>32K (&gt;16K and &lt;= 32K)</td>
<td>250</td>
<td>5%</td>
</tr>
</tbody>
</table>

Figure 2. Sample SMCAT Summary Report Export Area

---

How much of my TCP workload can benefit from SMC?

What kind of CPU savings can I expect from SMC?

This report is repeated for indirect IP connections (different subnets)
Taking the direct path with Shared Memory Communications - Direct Memory Access

BY DAVID HERR, RANDALL KUNKEL, AND JERRY STEVENS

Have you ever noticed a perfectly suitable walkway constructed using commercial-grade materials between two buildings and nearby, there’s a well-worn path that’s been formed by the majority of the pedestrian traffic going between the same buildings? What’s going on here? Why aren’t people using the constructed walkway?

Turns out, the old adage that “time is money” can be applied here. Consider a very busy college campus, commercial center, or government complex with lots of busy foot traffic. As people go about their daily tasks, they tend naturally to find the most efficient practices, including their paths or routes between buildings. Some of the pedestrians take this same path several times a day. One path that’s perfectly suitable might be a longer route or be “out of the way”, while the well-worn path is the most direct path. It’s this direct approach that represents savings.

When the walkway was built originally, why didn’t the design follow the most direct or efficient path? Often, such a design was not feasible at that time. Typically, this was due to some obstacle, say a tree or fence, that was in the way, or possibly something was missing that prevented the most direct path from being taken. At some later point, something changes, which makes the direct path possible — perhaps the obstacle was removed or the situation was resolved. Suddenly, people can now start taking the most direct path, saving time and money.

This simple analogy can often be applied to technology. This is how technology evolves. In our case, the “buildings” are unique IBM z/OS instances—logical partitions (LPARs) or guest virtual machines—running within IBM System z (the “complex”) and the “pedestrians” are network packets carrying application data (network traffic).

Today, several existing z/OS options for communication paths (“walkways”) are also constructed out of perfectly suitable materials. The most common communication paths are Ethernet (OSA), FICON® (CTC), and HiperSockets. While each of these technology options provides a communication path, HiperSockets provides the most optimal and direct path because it was designed specifically for intraprocessor communication. So, what has changed and how can we improve upon a technology like HiperSockets?

Introducing SMC-D and ISM

z/OS V2R2 has just delivered (through a PTF) a new protocol called Shared Memory Communications – Direct Memory Access (SMC-D). SMC-D combines with new IBM z13 or IBM z13s technology called internal shared memory (ISM).

ISM is provided with new z13 system firmware and is supported as a virtual Peripheral Component Interconnect (PCI) adapter. ISM is configured and managed just like existing PCI-based adapters, but ISM does not require any additional physical networking hardware. Similarly to Remote Direct Memory Access (RDMA), ISM enables “internal” direct memory access across z/OS instances (LPARs or z/VM® guests) within a central processor complex (CPC).

Figure 1 illustrates the concepts of SMC-D with ISM architecture.
SMC-D and ISM combine to form the most direct and efficient path between two z/OS instances for application data communication. This model allows two applications running in unique z/OS instances to communicate transparently and directly with each other. SMC-D is transparent to applications that exploit TCP sockets (TCP/IP). SMC-D avoids TCP/IP processing in the data path, but preserves such TCP/IP Quality of Services (QoS) as security, load balancing, and filtering for establishing connections.

SMC-D is a new variation of the existing Shared Memory Communications -RDMA (SMC-R). SMC-R exploits RDMA over Converged Ethernet using the IBM 10GbE RoCE Express feature, which provides highly optimized, cross-system communication. You can use both forms of SMC concurrently. The protocol dynamically selects the most direct path. When both z/OS instances are on the same CPC, SMC-D is used.

The following list summarizes the performance of SMC-D and ISM compared to TCP/IP and HiperSockets:

- **Request/response summary for workloads with 1k/1k – 4k/4k payloads:**
  - Latency: Up to 48 percent reduction in latency
  - Throughput: Up to 91 percent increase in throughput
  - CPU cost: Up to 47 percent reduction in network-related CPU cost

- **Request/response summary for workloads with 8k/8k – 32k/32k payloads:**
  - Latency: Up to 82 percent reduction in latency
  - Throughput: Up to 475 percent (~6x) increase in throughput
  - CPU cost: Up to 82 percent reduction in network-related CPU cost

- **Streaming workload:**
  - Latency: Up to 89 percent reduction in latency
  - Throughput: Up to 800 percent (~9x) increase in throughput
  - CPU cost: Up to 89 percent reduction in network-related CPU cost

ISM FIDs are not configured in z/OS. Instead, after ISM FIDs are defined in HCD and configured online to a z/OS LPAR, when the associated IP network adapter—Open Systems Adapter (OSA) or HiperSockets—is started, TCP/IP dynamically discovers and uses the ISM FID.

The SMC protocol uses TCP/IP to establish connections. This means an IP network is required to connect your peer hosts. With SMC-D, your IP network can be provided internally with HiperSockets or shared OSA or externally with separate OSAs (Ethernet). SMC architecture preserves your IP network topology and security. To be eligible to connect over SMC, you must be eligible to connect over IP, with both hosts having direct access to the same layer 2 network and IP subnet.

ISM and IP connectivity concepts
ISM is supported as a PCI device. PCI devices are configured with PCI function IDs (FIDs) and virtual function numbers (VFs), which lets multiple users (z/OS instances) share the same PCI physical resource. Because ISM is a logical resource, an ISM channel ID (CHID) is shared. The ISM CHID forms an internal “ISM network”, similar to internal queued direct (IQD) communication. When you configure multiple FIDs and VFs for the same ISM CHID, it allows a set of z/OS instances that are using the defined FIDs to communicate over this ISM network. You can configure another set of z/OS instances to use another ISM CHID, which isolates the second set of z/OS instances from the first. The z13 supports up to 32 ISM CHIDs and up to 255 FIDs per ISM CHID. You can also isolate z/OS instances on the same ISM CHID by using VLANs.

ISM FIDs are not configured in z/OS. Instead, after ISM FIDs are defined in HCD and configured online to a z/OS LPAR, when the associated IP network adapter—Open Systems Adapter (OSA) or HiperSockets—is started, TCP/IP dynamically discovers and uses the ISM FID.
Getting started with SMC-D and ISM

So what’s required to take the direct path? Ensure your system meets the following SMC-D and ISM requirements:

- ISM support requires z13 GA2 with driver level 27 or z13s.
- SMC-D support is provided in z/OS V2R2 with PTFs; see the preventive service planning (PSP) information.

What about configuring and enabling SMC-D?

1. Follow these steps:
   a. Define an ISM CHID with enough FIDs and VFs for each z/OS TCP/IP stack that will communicate by using this ISM CHID. z/OS requires a unique FID for each TCP/IP stack. The ISM CHID should have a single PNET ID (that is, each FID uses or defines the same ID).
   b. Define a matching PNET ID in your OSA (OSD) or HiperSockets (IQD) HCD configuration.

2. In your TCP/IP profile, enable SMC-D with an SMCD parameter on your Global Configuration statement.

After your ISM FIDs are “configured” to each z/OS instance, you can use the following z/OS display commands to verify the status of your devices, and to verify that you correctly configured the expected PNet IDs:

- `D M=CHP(NN)` where NN is the IQD or OSD CHPID
- `D PCIE,PFID=NNN` where NNN is the ISM FID value

After you complete the verification process, you can start your associated OSA or HiperSockets interfaces. Your ISM FIDs are dynamically discovered and started. As new and eligible TCP connections are started, your application workloads begin exploiting SMC-D. You can verify SMC-D and ISM status and usage with information that is provided by NetStat.

For more information about SMC-D and ISM and the benefits of this new technology, see: [ibm.com/software/network/commserver/SMC](http://ibm.com/software/network/commserver/SMC)

So, are you ready to take the most direct path?

SMC-D and ISM combine to form the most direct and efficient path between two z/OS instances for application data communication.
Still on IBM HTTP Server powered by Domino?

BY MARN A WALL E

If you’ve been thinking of migrating to IBM z/OS Version 2 Release 2, you need to know the most important (and possibly the largest) migration action in that release. This migration action might require some work on your current release, as well as some work after you start z/OS V2R2. In the worst of cases, it might even require application changes. What is that migration action? It is the replacement of the IBM HTTP Server Powered by Domino® with the IBM HTTP Server Powered by Apache (IHSA). To be clear, we are talking about the HTTP Server that is provided within the z/OS operating system itself. This is a z/OS V2R2 migration action, and has no relation to other HTTP servers that you might be running from other products, such as IBM WebSphere® software, which embeds their own HTTP servers.

Although some z/OS customers are not affected by this replacement, some z/OS customers are. What follows is a suggested migration path that you can follow to remove IBM HTTP Server Powered by Domino and add IHSA. This migration path provides the minimum risk and allows you to make changes in discrete steps.

Are you affected?
The first item is to know whether you are affected by the removal. Many people know right away, others are unsure. In either case, on every system, you should be certain whether you are using the IBM HTTP Server Powered by Domino. To that end, IBM provides a simple IBM Health Checker for z/OS health check to find out whether you are running the IBM HTTP Server Powered by Domino. This health check provides the minimum risk and allows you to make changes in discrete steps.

I’m impacted, now what?
If your system is at z/OS V1R13 or V2R1, install the IBM Ported Tools V1R3 HTTP Server (5655-M23). This provides IHSA at the V8.5.5 level (also known as the Apache 2.2 level). Your task in this time frame is to move your z/OS V1R13 or V2R1 system from IBM HTTP Server Powered by Domino to this IHSA V8.5.5 level. There are several aids to help you do this:

- Use the IBM Redbooks® publication, IBM HTTP Server on z/OS: Migrating from Domino-powered to Apache-powered, REDP-4987. Read this book carefully. Not all Domino-powered functions are available in the Apache-powered HTTP server; that is where the dreaded application changes might rear its ugly head. redbooks.ibm.com/abstracts/redp4987.html
- Use the IBM developerWorks® IBM HTTP Server forum, which contains many common questions that might help you. See: ibm.biz/Bd4v3q
- Use the tools on the z/OS UNIX Tools page. To find the list of tools, click the Tips tab at: ibm.com/systems/z/os/zos/features/unix/bpxa1ty2.html
  The IHS Configuration Migration Tool (ihsCMT) generates an approximation of an equivalent HTTP configuration file for IHSA.

You can run Domino and Apache HTTP servers side by side on z/OS V1R13 or V2R1. In fact, if you are using the z/OS Infoprint Server Infoprint Central function, you must keep the IBM HTTP Server Powered by Domino running on your current system, because there is no PTF available for z/OS V1R13 or z/OS V2R1 to supply the IHSA support. The z/OS Infoprint Server Infoprint Central function support for IHSA is found in z/OS V2R2.

So far, so good. I have IHSA V8.5.5 running pre-z/OS V2R2
Next, migrate from the IHSA V8.5.5 you have on z/OS V1R13 or z/OS V2R1 to the IHSA included in z/OS V2R2. The IHSA level included in z/OS V2R2 is the V9.0 level (also known as the Apache 2.4 level). You have a choice at this point: you can decide whether you want to continue to run IHSA V8.5.5 (provided from the IBM Ported Tools product) on z/OS V2R2, in addition to the IHSA V9.0 level, or whether you want to run only the IHSA V9.0 level with z/OS V2R2 right from the start.
Some users feel most comfortable running IHSA V8.5.5 at the beginning of their migration to z/OS V2R2, and finishing their IHSA migration to V9.0 at a later time, after z/OS V2R2 is deployed. Other users feel that the migration from IHSA V8.5.5 to IHSA 9.0 does not entail much work, and choose to move to IHSA V9.0 when z/OS V2R2 is deployed. It is your choice, but there is a deadline to be aware of; after the end of service for z/OS V2R1 (planned for September 2018), you cannot run IHSA V8.5.5 on z/OS V2R2.

If you chose to run IHSA V8.5.5 with z/OS V2R2, you are not able to order the IBM Ported Tools V1R3 HTTP Server (which is IHSA V8.5.5) with your z/OS V2R2 ServerPac; that feature cannot be ordered because z/OS contains a later level (IHSA V9.0). The work to make your existing IHSA V8.5.5 available on z/OS V2R2 is work that you will need to do. Details on how to perform this task are described in techdoc: ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10857

Another consideration before you decide to run IHSA V8.5.5 on z/OS V2R2 is that the Infoprint Server Infoprint Central function requires the fix for APAR OA48654 to support IHSA V8.5.5.

The differences between IHSA V8.5.5 and IHSA V9.0 are much smaller than the differences between IBM HTTP Server Powered by Domino and IHSA V8.5.5. You can learn about the differences between IHSA V8.5.5 and IHSA V9.0 ("Apache HTTP Server Version 2.4", “Upgrading to 2.4 to 2.2") at: publib.boulder.ibm.com/httpserv/manual24/upgrading.html

You can find a general process on how to move from one IHSA level to another at: ibm.com/support/knowledgecenter/SSEQTJ_8.5.5/com.ibm.websphere.ihs.doc/ihs/cihs_upgrading2.html

This documentation is a general process, and does not provide specifics about the differences between levels.

You are almost there!

After your system is at z/OS V2R2, you need to accommodate z/OS functions that now use IHSA V9.0. As expected, the z/OS Infoprint Server Infoprint Central function requires that you complete some tasks to do to configure it to use IHSA. Other z/OS elements affected are PKI and Library Server. Find all the IHSA tasks necessary for these elements in z/OS V2R2 Migration, GA32-0889 and in z/OS V2R2 Migration Workflow, if you use IBM z/OSMF Workflow.

Home free!

When you get this far, you are now well positioned on z/OS to take advantage of new functions that were not available in the Domino-powered HTTP server are now at your fingertips. Enjoy your new HTTP server and z/OS V2R2!
PFA extends its cognitive capabilities

BY KARLA K. ARNDT

Predictive Failure Analysis (PFA) has been in the forefront of cognitive computing since IBM z/OS V1R10, when it released its first health check to predict the potential for common storage exhaustion. To improve z/OS availability and resiliency, PFA continues to extend its capabilities to detect other potential problems based on your systems’ behavior.

In z/OS V2R2, PFA delivered an often-requested health check, which predicts the potential for private, virtual storage exhaustion in the near future; it alerts you, so that you can take action before your system is affected. In addition PFA has added enhancements that make it more usable and dynamic for existing PFA checks.

Predicting storage exhaustion

PFA extends capabilities to predict storage exhaustion in z/OS V2R2 with the PFA_PRIVATE_STORAGE_EXHAUSTION check. This check predicts future usage of private storage locations below the 2 GB bar. These storage locations are denoted by PFA as follows: Private User Region (USER), Private System Region (AUTH), USER and AUTH combined (BELOW), Extended Private User Region (EUSER), Extended LSQA/SWA/229/230 (EAUTH), and EUSER and EAUTH combined (ABOVE).

PFA is focused on predicting private, virtual storage exhaustion for address spaces that will affect your system as a whole. It is important that the address spaces that are selected are fairly “persistent”, and that use enough storage to have the potential for exhaustion. PFA recognizes there is a fine line between tracking too much data (for example, files that store too much historical data become large, and processing them consumes significant resources), and tracking too little data (which reduces the accuracy of the predictions and might eliminate the tracking of an important address space). Therefore, PFA has created a way to dynamically choose which address spaces qualify for data collection and for modeling.

PFA determines that an address space qualifies for data collection if it is “persistent” and is using “enough” storage to have the potential for exhaustion. If an address spaces starts within the first hour after IPL, or has been active for the time specified in the configurable COLLUPTIME parameter, it is considered persistent. An address space is considered to be using enough storage if it is using a configurable percentage of the current capacity for any given storage location as defined by the configurable usage parameter, COLL% (20% by default). By using these parameters, PFA allows an address space to start at any time, and be dynamically selected for data collection when it is considered persistent and is using enough storage.

PFA also recognizes that it is not beneficial to create predictions for storage locations that are not increasing, or that are not using enough storage to have the potential for exhaustion even if historical data is being collected. Therefore, PFA postpones creating predictions for the historical data until a configurable percentage of the current capacity is used, which is defined by the MOD% parameter (40% by default).

Even with these parameters, it is not necessary to model all locations that have reached MOD% of capacity. For example, a location that is using 65% of capacity, but whose usage is stable, is not in danger of storage exhaustion. PFA further selects locations to model by focusing on the top 15 address spaces per storage location that have increased their usage the most in the last hour. These locations must also have at least one hour of historical data collected to produce a trend. By default, PFA creates predictions every 720 minutes, but will model more frequently when it detects instability that would
cause the existing model to become stale. The definition of total exhaustion is when a storage location is going to use 100% or more of capacity. But, sometimes it’s more efficient to be notified if the prediction is higher or lower than that value. PFA has provided the configurable parameter, COMP%, which you can use to define your exhaustion point.

If you want exceptions to occur sooner, such as when the storage location is predicted to reach only 95% of capacity, modify this value to be 95. Conversely, if you want exceptions to occur when the storage location is predicted to reach 110% of capacity, modify this value to be 110.

The closer the address space is to exhausting storage, the more critical it is to take action. This new check uses dynamic severities when issuing the alerts for the potential for exhaustion. By default, if the exhaustion is predicted to occur within the next 3 hours, a critical eventual action WTO is issued. If the time to exhaustion is predicted to be between 3 - 5 hours from now, an eventual action WTO is issued. If it’s going to be more than 5 hours to exhaustion, an informational WTO is issued. These WTO severities are based on time and are configurable by using the E_HIGH, E_MED, E_LOW, and E_NONE parameters.

As is true for the common storage exhaustion check, the private, virtual storage check cannot detect all issues caused by fragmentation or by fast increases that are on a machine-time scale or faster than one collection interval.

New features for existing checks
PFA is committed to continuous improvement and as enhancements are created initially for new checks, those enhancements are applied to existing checks where applicable. For example, dynamic severity alerts based on time to exhaustion were applied to the existing PFA_COMMON_STORAGE_USAGE check.

Also, prior to V2R2, the PFA_JES_SPOOL_USAGE check could only detect abnormal usage of track groups for address spaces that started within an hour after IPL. Address spaces could be specifically excluded, but address spaces that started after an hour after IPL were never tracked. PFA applied the algorithm for dynamic selection based on the configurable active duration parameter (COLLUPTIME) to this check, too. That is, this check now collects historical data for any address space that has been active for COLLUPTIME (6 hours by default for this check). Address spaces tracked by this check can now start at any time.

In V1R12 and later, you could exclude address spaces that might cause undesirable exceptions or skew results. However, you were not able to force PFA to track specific address spaces. With V2R2, you can track the specific address spaces, which are more useful for checks that do not support dynamic selection for data collection. This includes the PFA_MESSAGE_ARRIVAL_RATE, PFA_SMF_ARRIVAL_RATE, and PFA_ENQUEUE_REQUEST_RATE checks. Now, if your system does not start all of its important address spaces in the first hour after IPL, or if the workloads that use the address spaces aren’t active until later, you can use the new INCLUDED_JOBS support file to force PFA to track them after they start.

Ease of use
Installation and setup are now much easier with only one configuration file, /etc/PFA.ini. The need to run a shell script every release was eliminated. PFA does installation validation when PFA starts, issues detailed messages if startup problems are found, and won’t start until they are fixed. You can define the directory in which to store the PFA files.

To reduce the amount of DASD for PFA's files, some checks’ file structures were modified, which also reduces their data processing. Also, PFA now keeps a maximum of 30 exception directories per check, and keeps only those that are newer than the number of days specified in the configuration value EXCDIRDAYS (90 days by default). All old directories are deleted when PFA starts and when a new directory is being created for an exception.

For checks that have STDDEV or STDDEVLOW configuration parameters, tuning assistance is now provided directly on the exception reports. When an exception occurs, PFA calculates the highest STDDEV or STDDEVLOW that would have been required to avoid the exception condition for the data used in this comparison. After the exception has been investigated, if the condition is not considered an abnormal condition for your system, set the STDDEV or STDDEVLOW parameter to the value indicated to avoid the exception in the future when the data is similar.

With PFA, you can define a time-of-day to model for each check to avoid modeling during peak periods in a stable environment. For example, if you want to force predictions to be created at 5:00 AM and then not again until 6:00 PM when the data is stable, set the new configuration parameter FORCEMODEL(5:00) and MODELINT(780) persistently. PFA will model at 5:00 AM and, at most, 13 hours later. At any time, if PFA detects that the data is changing substantially, it will create a model even if 13 hours has not yet passed so that your predictions are not stale.

PFA now provides more data on the exception reports for the PFA_COMMON_STORAGE_USAGE check by displaying the current usage in "SYSTEM" and additional owner gone information.

With these enhancements, PFA is improving and expanding its cognitive capabilities to improve your systems’ availability and resiliency.

More information on these enhancements and other features of PFA can be found in z/OS Problem Management, G325-2564.
Find z/OS IT information quickly

BY GEOFFREY K. SMITH AND JODI EVERDON

In today’s IT environment, many employees work across job roles. For example, a system administrator who was once highly skilled in a few areas, might now need to be skilled in many areas. Finding the correct information quickly to do your job is important which is why z/OS provides you with information choices.

Easy search for PDF
While some people might believe that PDF files (which can be referred to as manuals, books, or hardcopy) are obsolete, others prefer, and might even require PDF files. A common problem with using PDF files for a large library is the inability to search across the entire collection of PDF files. IBM z/OS V2R2 addresses and eliminates the search dilemma by providing you with a PDF index for the entire z/OS release. Now, even in an isolated IT environment, you can easily download the z/OS library and its index to search every file.

How to use PDF index
1. If you don’t already have it, download Adobe Acrobat Reader. You need the full reader, not the plug-in version for web browsers.
2. Go to the PDF Download section of the z/OS Internet Library and select z/OS V2R2.
   Hint: If you ever need to find z/OS information, search for “zFavorites” by using your favorite search engine.
3. Click the link to download all of the PDF files. The compressed file contains all the PDF files, an HTML index listing the titles, and an Adobe Acrobat index of the entire collection.
4. Extract the package, open the PDF subdirectory, and double-click the PDF file for the z/OS collection. As shown in Figure 1, this step opens the advanced search dialog for Adobe Acrobat Reader with the preselected index for the collection.

Now, you are ready to type in a search argument to get the list of PDF books for your search. When you click a document title, the window shows the search results in context (similar to BookManager).

Speaking of BookManager
Many people told IBM, “I want BookManager” but support for it was removed from z/OS in V2R1 and cannot be restored. If you are a BookManager fan, there is another option you might consider. An experimental, proof-of-concept, z/OS V2R2 search-scope catalog has been added to the z/OS product documentation in IBM Knowledge Center. The search scope catalog includes methods for searching by element, feature, and even by book.

How to use the V2R2 search scope catalog
1. Go to ibm.biz/Bd4Yau
2. Use the menu to choose the element or feature (also known as the shelf) that you want to search.
3. Enter the search term. You can search a book or all the content in the element or feature.

Figure 2 shows an example of a search for the SMFLLMxx parmlib member in z/OS V2R2 MVS Initialization and Tuning Guide, SA23-1379.

Search engines
Many people use a search engine to find information. You enter the text you want to find and see what results are returned. Sometimes you get an exact match, but other times you are forced to wade through hundreds of matches.
When you order z/OS V2R2, the order includes the IBM Knowledge Center for z/OS (KC4z). You can add to and maintain information in the KC4z, which makes it a wise choice when you want a comprehensive documentation repository for your products. For more information, see IBM Knowledge Center for z/OS Configuration and User Guide, SC27-6805.

You can review all your z/OS information choices by going to the IBM z/OS Internet Library at: ibm.com/systems/z/os/zos/library/bkserv

Figure 2. z/OS V2R2 search scope catalog
MFA to the rescue!

BY ROSS COOPER, BRUCE WELLS, AND SCOTT WOOLLEY

In a world filled with ever present threats, how can we keep our valued treasures out of the hands of those who seek to do us harm? Once upon a time, passwords would fully shield us, but their strength is waning. Just in time... here comes multi-factor authentication (MFA) to the rescue!

Passwords are old news
Unfortunately, we can be pretty lousy at selecting strong passwords. Who can blame us when choosing a new password seems like an inconvenient and repetitive task? Faced with this dull activity, many of us take the same flawed approaches such as picking our favorite superhero, using our pet’s name, or selecting patterns from our keyboard. Then, after we come up with a password, we often make the mistake of writing the password down or reusing it on many other systems. These mistakes just increase the attack surface if one of our systems is compromised. Combine these poor password habits with malware, keyboard logging software, and offline password database cracking, and it’s easy to see why protecting systems with only passwords is falling out of favor.

Strength in numbers
One way to outperform passwords and raise the assurance level of mission-critical systems is to require multifactor authentication (MFA). An MFA system requires that a user provide at least two different authentication factor types, each from a separate authentication multi-factor category, when entering the system. The authentication factor categories are:

- Something you know (such as a password or PIN code).
- Something you have (such as an ID badge or cryptographic token device).
- Something you are (such as a fingerprint or retinal scan).

By requiring multiple authentication factors, your account cannot be compromised even if one of your authentication factors is discovered. For example, if your token device is lost, your PIN code still protects your account. Or, if your key strokes are captured by malware and your PIN code is compromised, the token device is still required to log on. This extra protection that MFA provides can make the difference between a secure system and a compromised system.

A new hero appears
IBM Multi-Factor Authentication for z/OS (IBM MFA), with RACF, provides a way for IBM z/OS systems to authenticate selected user IDs by using multiple authentication factors. Security administrators use RACF commands to add MFA data to a user ID and to define which MFA factors are accepted on the system. When an MFA-provisioned user logs on to a z/OS application, RACF detects that the user requires MFA authentication and calls IBM MFA. Additionally, RACF logs the MFA authentication event to SMF.

Starting with z/OS V2R1, with the PTF for APAR OA48359 installed, IBM MFA and the RACF MFA infrastructure are available and support the RSA SecurID multi-factor authentication method.

Getting down to business
After SMP/E-installing the IBM MFA product, follow the steps in IBM Multi-Factor Authentication for z/OS Installation and Customization, SC27-8447, to configure IBM MFA and provision RACF users.

1. Modify the SYSEXEC and PROCLIB members.
2. APF-authorize the SAZFLOAD data set.
3. Add AZFSTCMN to the SCHEDxx member.
4. Configure the AZF started task.
5. Define the RSA SecurID factor to RACF:
   RDEFINE MFADEF FACTOR.AZFSIDP1
6. Define the R_factor authorization profile and permit the MFA administrator:
   RDEFINE FACILITY IRR.RFACTOR.MFADEF.AZFSIDP1
7. Allocate the RSA node secret data set.
8. Allocate the RSA configuration file.
9. Set SecurID parameters.
10. Start the MFA AZF task.
11. Assign the RSA SecurID factor to your user IDs by using the MFA keyword of the ALTUSER command:
    ALTUSER userid MFA(FACTOR(AZFSIDP1) ACTIVE TAGS(SIDUSERID:RSA_userid))
12. Activate MFA authentication for your MFA-defined users by activating the MFADEF class:
    SETROPTS CLASSACT(MFADEF)
Using your new powers

Now that you configured RACF to enforce RSA SecurID authentication for this certain subset of user IDs, how do these users log on to the system? The answer is, “Well, it depends.” In this case, logon processing depends on the token type and the application being logged on to.

RSA Authentication Manager provides several types of tokens, such as a physical hardware device or a software program running on a workstation. Some tokens have PIN-entry keys while others do not.

With a PIN-entry key, you type in your PIN to obtain a token code that incorporates the PIN into its key generation algorithm, thus representing two factors with a single string. You then type in the token code as your “password” in the logon dialog box presented by the application.

In the absence of a PIN-entry key, you type both your token code and your PIN into the logon dialog box. If the dialog box supports long passwords (password phrases), you simply enter your PIN followed by your token code (without spaces) in the password field.

Not all applications support long passwords or password phrases. In these cases, you must enter the token code as the password and the PIN as the new password. In many cases, the application requires you to enter the PIN twice to confirm it.

RACF recognizes that the user is required to use MFA and sends the password data to IBM MFA for authentication. If the credentials are incorrect, the password revoke count is incremented, and if this happens enough times in succession (as configured with your SETROPTS password revoke value), the user is revoked in RACF.

Of course, certain types of users (system programmers, for example) must always be allowed to log on even if there are problems with the MFA started task or the RSA server. In this case, the user should be defined in RACF with the “password-fallback” option. If MFA processing is unavailable for any reason, the user is still able to log on with their RACF password and must, therefore, continue to remember and maintain their password.

Additionally, the RACF audit trail indicates whether the user authenticated with a password, password phrase, PassTicket, or MFA. This logging happens automatically with the PTF for APAR OA48359 installed. This feature further improves your ability to demonstrate compliance with your security policy.

Roll the credits

Rest a little easier knowing that you have MFA on your side, but we’ve only just scratched the surface of our new found powers. For more information about the IBM MFA solution, see these resources:

- IBM Multi-Factor Authentication for z/OS Installation and Customization, SC27-8447
- IBM Multi-Factor Authentication for z/OS User’s Guide, SC27-8448
How to customize TSO/E LOGON using parmlib

BY ERIC ROSENFIELD

The seemingly simple TSO/E LOGON command has a number of options that can be set by an administrator that greatly affects how this command acts. This article discusses how the LOGON process can be tailored using a parmlib member.

Customizing logon controls

An IKJTSOxx parmlib member provides you with the ability to customize TSO/E processing. The LOGON statement in member IKJTSOxx controls logon options that are processed when TSO/E starts. You can update the options dynamically by using the TSO/E PARMLIB user command or the SET IKJTSO console command.

You can specify the options PASSPHRASE, VERIFYAPPL, LOGONHERE, and PASSWORDPREPROMPT. By default, these options are not enabled.

LOGON PASSPHRASE (ON | OFF)

When this option is ON, users can log on with either a password or password phrase as long as they have one defined. With the latest RACF support, users can be defined as phrase-only users without a password. Because phrases must be longer than passwords, TSO/E determines which authenticator is entered by length and processes accordingly. If this option is set OFF, only passwords are allowed for authentication. The preferred method is to set this option ON.

LOGON VERIFYAPPL (ON | OFF)

When this option is ON, the application ID is sent to RACF when the user ID is being verified. With this method, you can restrict access to a specific TSO/E application based on the definitions in a RACF application profile. If this is set OFF, no application ID is provided to RACF when the user ID is verified and all TSO/E instances are accessible with a valid ID and password.

LOGON LOGONHERE (ON | OFF)

When this option is ON, users can reconnect to an existing session even when they have not disconnected from the current session. A user can issue a LOGON command from a terminal when they have an active session on another. If the option is set OFF, the user receives the following message when they attempt to reconnect to an active session:

IKJ5641I TSOLOGON RECONNECT REJECTED -USERID userid IN USE

The latest logon option

A new option is available for all supported releases in PTFs for APAR OA44855. When the option is active, the user is prompted for the ID and password, and the values are validated before the full-screen logon panel is displayed. If validation fails, the command is terminated, with no indication of which supplied information, ID or password, was the cause of the failure.

PASSWORDPREPROMPT (ON | OFF)

When a valid ID and password is entered, the logon panel is displayed and logon processing continues.

Implementing this feature protects system information. It disables the ability to enumerate valid user IDs based on failure messages because it can only be known whether the user ID exists if the password is known. Another benefit of this feature is that characteristics of the user ID are only displayed if the user ID is authenticated. This prevents collection of additional information about the system (for example, logon procedures, account numbers, or commands run at logon) without a valid user ID and password.

The preferred method is to set this option ON.

Conclusion

The IKJTSOxx parmlib member controls useful options of LOGON processing. By taking advantage of these options, you can aid in enforcing usage and security policies.
You have a business-critical workload running on a z/OS system that you want to deploy on multiple sites so that if an outage occurs, you can quickly and transparently switch that workload from one site to another.

With GDPS®/Active-Active (GDPS/A-A) Sites, you can switch business-critical workloads from the active site to the standby site, which can be hundreds of kilometers away, in a matter of seconds.

Transactions that update the data being accessed by business-critical workloads are routed to the active site. During software replication, those transactions are copied to the standby site. If a problem exists with the active site that affects those update transactions, GDPS/A-A Sites can automatically switch the routing of incoming update transactions to the standby site. Transactions that do not update the business-critical data, but which query it, can be routed to either the active or the standby site, or to both.

**GDPS/A-A Sites workloads**

Many products talk in terms of workloads, but the term workload can have a different meaning for each product. It is worth spending a few moments to consider what workload means in the context of GDPS/A-A Sites.

Transactions that update the data associated with the business-critical workloads are routed to the active site, and those updates are then replicated on the standby site. This set of transactions is defined to GDPS/A-A Sites as an update workload.

If the business-critical workload includes transactions that access, but do not update the same business data, this set of transactions can be defined as a separate query workload, which is explicitly associated with its corresponding Update workload. With GDPS/A-A Sites, query workload transactions can be routed to either the active or standby sites, or both, depending on what is most appropriate.

Once started, the availability of a GDPS/A-A Sites workload is monitored. If a problem is detected, that workload can be automatically switched to an alternative site, or the system operator is alerted. A combination of IBM products is used to manage and monitor the following tasks:

- Starting and stopping the workload’s business application environments on the different sites
- Replication of updates to data made by the workload between the sites
- Incoming network connections to a workload

**Defining your GDPS/A-A Sites workload**

Each GDPS/A-A workload is defined in terms of these factors:
• The applications that execute the business-critical transactions that need to be protected using GDPS/A-A Sites

• The data that is accessed by those applications, which needs to be replicated between the two sites

• One or more dedicated connections or transports that are assigned to those applications, over which those applications’ transactions are initiated.

Step 1. Identify the applications that make up your business-critical workload

You need to identify the software elements that make up your business-critical workload. You also need to indicate to GDPS/A-A Sites the z/OS address spaces in which those software elements run.

Those applications can run in one or more z/OS address spaces. Because GDPS/A-A Sites is based on z/OS parallel sysplex technology, the workload can span multiple z/OS LPARs that are running in the same parallel sysplex.

After you identify the address spaces in which the business-critical applications run, use IBM Tivoli® System Automation for z/OS to define them to GDPS/A-A Sites. Tivoli System Automation performs the following roles for GDPS/A-A Sites:

• Acts as a repository for GDPS/A-A policy information.

• Monitors the status of the software elements that are defined as part of a GDPS/A-A workload and shares that status information with the other components of GDPS/A-A Sites.

After the workload is defined, use the GDPS/A-A Sites web application to start or stop all the elements of the workload by using the GDPS/A-A Sites web application. All that you have to do is click a button.

Tivoli System Automation for z/OS passes information about the status of address spaces and systems to GDPS/A-A Sites. If a problem exists with the workload on the active site, you can take the necessary steps to switch processing for that workload to the standby site.

Step 2. Identify the data accessed by your business-critical workload

This step is arguably the most critical part of the GDPS/A-A Sites investigation and planning.

A. You must identify exactly what data is updated or accessed by the GDPS/A-A Sites workload.

B. Because that data must be replicated, you must define the data to the IBM product responsible for data replication.

Which product you use to define and manage replication of the application data depends on the nature of the application.

With GDPS/A-A Sites, you can use the following data replication products:

• IBM InfoSphere® Data Replication for IMS™ for z/OS
• IBM InfoSphere Data Replication for VSAM for z/OS
• IBM InfoSphere Data Replication for DB2 for z/OS

Restriction: If not all the necessary application data is defined to the replication software, the data on the two sites will not be consistent.

GDPS/A-A Sites monitors the replication operation to ensure that the replication engines are running. It also monitors the time that is taken to copy updates to the standby site, which is known as the workload latency.

If the latency of an update workload process exceeds a user-specified threshold, GDPS/A-A Sites can automatically take action. GDPS/A-A Sites can also notify the operator of the problem so they can take appropriate action. If a query workload process that is associated with the update workload has a latency problem, GDPS/A-A Sites can automatically adjust the routing of the query workload to prevent incoming query workload transactions from accessing stale data on the standby system.

Step 3. How do client applications connect to your business-critical workload?

GDPS/A-A Sites controls and monitors the connections over which your business-critical workloads are initiated. If a connectivity problem prevents the initiation of new connections to the active site or to the standby site (such a problem would prevent a successful switch to the standby site), GDPS/A-A Sites can take action where possible or inform the operator that GDPS/A-A Sites cannot automatically resolve the problem.

You have to identify the network connections through which the transactions that make up your GDPS/A-A Sites workload arrive. With GDPS/A-A Sites, you can use the following connectivity methods:

• TCP/IP (for example web-based technologies such as HTTP, Web Services, JDBC)

• SNA (typically used for old-fashioned “green screen” IBM 3270 transactions)

Multiple TCP/IP ports, or in the case of SNA, multiple VTAM® APPLIDs, can be associated with a GDPS/A-A workload. All the connections over which the workload can be accessed must be included in the workload definitions. If transactions that update the business-critical workload’s data arrive over a connection that is not defined as part of the workload, updates made by transactions arriving over that connection might compromise the consistency of the data that is accessed by the workload.
With GDPS/Active-Active (GDPS/A-A) Sites, you can switch business-critical workloads from the active site to the standby site, which can be hundreds of kilometers away, in a matter of seconds.

You must define the way in which client applications access your business-critical workload to GDPS/A-A Sites. GDPS/A-A Sites monitoring and control of connectivity to its workloads consists of two elements:

- A Server/Application State Protocol (SASP) conformant TCP/IP router (the Server/Application State Protocol is documented in RFC 4678)
- IBM Multi-site Workload Lifeline

After you define connectivity to the GDPS/A-A workload, IBM Multi-site Workload Lifeline monitors the status of the connections that are associated with the workload, and transmits this status to the SASP router. Based on the information that is passed to it by IBM Multi-site Workload Lifeline, the SASP router decides where to route incoming connection requests for the workload: the Active site for Update workloads, and one, none or both sites for Query workloads. IBM Multi-site Workload Lifeline shares information about the status of the connections with the other components of GDPS/A-A Sites.

If IBM Multi-site Workload Lifeline detects a connectivity problem on the Active site which requires that the workload be moved to the Standby site, GDPS/A-A Sites informs the operator of any replication or application issues that would prevent a successful switch to the Standby site.

For planned and unplanned network or system outages, GDPS/A-A Sites provides a mechanism to seamlessly switch business-critical z/OS workloads from one site to another with minimal operator intervention, with near continuous availability and minimal operator intervention.
Data migration made easy

BY GLENN WILCOCK

Whether you just bought a new disk system, defined a set of extra-large disk volumes, or need to merge data from one storage group to another, it can feel overwhelming to face the task of migrating data from its existing location to a new one. IBM z/OS V2R2 DFSMSshsm provides a MOVE option on the MIGRATE command so that you can migrate data in a few simple steps.

By using the MIGRATE command, you can move unreferenced online data to less expensive offline storage. The new MOVE option enables you to use the MIGRATE function to simply move online data from its existing location to a different online location, as defined by the SMS ACS routines. When you specify MOVE for a VOLUME or STORAGE GROUP, DFSMSshsm processes every data set on the specified volume or within the specified storage group.

For example, supposed that you have new EAVs in a storage group, and you want to move existing data off smaller volumes to these larger EAVs. You can do this in two simple steps:

1. Set the SMS status of the existing smaller volumes to DISNEW. (This prevents SMS from selecting these volumes for any new allocations).
2. For each of the smaller volumes, issue the MIGRATE VOLUME(volser) MOVE command.

For each data set on the specified volume, DFSMSshsm runs the DFSMSdss COPY with DELETE command by using standard ACS allocation to perform the move. As shown in Figure 1, because the EAVs will have the most free space, they will be automatically selected as the targets for the allocations. This can also be done at the data-set and storage-group level.

As another example, suppose that you want to consolidate the number of storage groups that you have in your environment. Do this task by following these basic steps:

1. Update the SMS ACS routines to ensure that new data allocations are targeted to one of the storage groups that will persist.
2. Set the SMS status of the volumes to be drained to DISNEW.
3. Issue the MIGRATE STORAGEGROUP(sgname) MOVE command for each storage group that is to be removed.

DFSMShsm automatically processes all the data sets within each of the specified storage groups, and moves them to a newly assigned storage group.

Figure 2 shows how this function can be used to move data in a variety of ways.

Because a data set must be closed to move it, only closed data sets can be processed. By default, any data set that is not closed is skipped. Because certain data types, such as IBM DB2 objects, are expected to always be open, DFSMS provides a method to process these types of data sets, also.

You can use the SERIALIZATION ERROR...
EXIT option of the SMS management class to indicate that the data assigned to a management class, are associated with DB2, IBM CICS, or IBM zFS. If one of these types is specified, before moving these data sets, DFSMSdss first attempts to close them.

To do so, DFSMSdss invokes DB2 or CICS to close the data sets, or invokes zFS to unmount the file system. If the action is successful, (there are no outstanding transactions that would prevent the close operation), DFSMSdss moves the data set. After the data set is moved, DFSMSdss invokes DB2 or CICS to open the data set, or invokes zFS to mount the file system.

This function is further optimized by using the supporting FlashCopy® feature to move data very rapidly. With the FlashCopy feature, control is returned to the user immediately after a FlashCopy relationship is established rather than waiting for all of the data to be copied to the target location. The actual data movement is performed by the disk control unit in the background so the user can immediately access to the data at the new location.

The TRANSITION COPY TECHNIQUE option of the SMS management class indicates which type of data movement technique to use. The default is to use standard I/O. For large data sets, standard I/O can take a few minutes, many minutes, or even longer. Because the data set is deleted after the data is moved, using standard I/O ensures that the original data set is not deleted until all of the data is physically moved to the new location.

With the FlashCopy feature, because the data set is deleted before the control unit completes the background copy, the default is that you can only use the FlashCopy feature if the data-set change indicator is off and there is a current backup copy for the data. This ensures that if a physical error occurs during the FlashCopy operation, a valid backup copy exists for recovery.

Because the data-set change indicator is always on for DB2, CICS, and zFS data sets that are closed by DFSMSdss, you can use the PRESERVE MIRROR REQUIRED (PMR) option as another copy method.

When you specify the PRESERVE MIRROR REQUIRED (PMR) option, it indicates that the FlashCopy operation must be done concurrently at the local site and at a Metro Mirror remote site.

Because there are two simultaneous copies of the background copy, FlashCopy can be used with integrity. If a physical error occurs at the local site, a failover operation is done at the remote site to ensure the integrity of the data set being moved. If you start your data migration by using the PRESERVE MIRROR REQUIRED (PMR) option during an off-peak period, you can complete the move with minimal downtime.

Try out this new feature and see how much simpler it can make data migration for you!
IBM System z 10GbE RoCE Express Virtualization

BY MICHAEL GIERLACH, RANDALL KUNKEL, AND DHANANJAY PATEL

In 2013, IBM unveiled support for Remote Direct Memory Access (RDMA) over Converged Ethernet, or RoCE (pronounced “Rocky”), on the IBM System zEC12 (zBC12). RDMA allows for the direct access of a remote peer’s memory without involving either operating system. You can exploit RDMA protocols with your existing 10 Gigabit Ethernet (10GbE) switches without special switch features. RoCE packets are processed just like any other Ethernet packet.

New kid on the block
IBM System z support for RoCE uses the newest kid on the block, the IBM 10GbE RoCE Express feature. The RoCE Express feature might not be imposing physically, weighing a whopping half-pound and measuring just seven inches, but it packs a real punch by providing an optimized network interconnection for System z communications. Request/Response (RR) workloads can take advantage of the low latency, high-bandwidth networking that RoCE provides, while streaming workloads can take advantage of significantly reduced networking CPU costs. IBM z/OS uses the RoCE Express feature to provide an RDMA-based solution called Shared Memory Communications over RDMA (SMC-R). SMC-R is a sockets-based solution that provides transparent access to RoCE for TCP socket applications over standard 10GbE Ethernet.

Learning to share
Like many youngsters, when RoCE was new to System z it had trouble with the concept of sharing. Basically, only one LPAR could use a RoCE Express feature at a time. Fast forward two years, and RoCE is bigger, stronger, and has been adopted on the IBM System z13 and z13s. RoCE is now able to share, or virtualize, its resources across multiple LPARs or IBM z/VM guest virtual machines, up to a total of 31 virtual servers at a time. The sharing is achieved by using standardized PCIe virtualization (SR-IOV) technology. A single z13 can house up to 16 physical RoCE Express features on the CPC. RoCE might not be old enough to do the math, but this allows you to define up to 496 z/OS instances or virtual functions (VFs) for SMC-R communications. With RoCE virtualization support, you can extend SMC-R capability to additional workloads, over more LPARs, with fewer physical features. Each virtual function is identified by a function ID, or FID. RoCE FIDs support dynamic I/O that your administrators can use to dynamically provision FIDs by adding or removing FIDs from different LPARs.

When exploiting SMC-R, there are best practice recommendations for achieving high availability (HA). To eliminate a single point of failure, users should provision two unique FIDs, each associated with a different physical adapter, for each z/OS instance and unique physical network. Attachment to a unique physical network is identified with a physical network ID (PNet ID). PNet IDs are configured in Hardware Configuration Definition (HCD) per physical port and dynamically learned by z/OS. Using multiple FIDs allows SMC-R traffic to continue if the RoCE Express feature encounters a problem. While the RoCE Express feature recuperates, traffic dynamically moves to the second feature to preserve the TCP connections that are using SMC-R. After the feature is working correctly again, the TCP connections are dynamically redistributed and rebalanced across the two features. Figure 2 shows two z/OS LPARs sharing two RoCE Express features in a fully redundant manner.
Virtualization value points
The RoCE Express feature is not only resilient, but with 496 possible instances, RoCE is highly scalable on the z13. With RoCE virtualization, you can extend the benefits of RDMA technology to more of your application workloads by using SMC-R.

Just how well does SMC-R perform compared to existing TCP/IP and OSA-Express QDIO processing?

- Network latency for z/OS TCP/IP-based Online Transaction Processing (OLTP) workloads are reduced by up to 80 percent.
- Network-related CPU consumption for z/OS TCP/IP-based workloads with bulk or streaming data patterns are reduced by up to 60 percent with an increase in network throughput by up to 60 percent.

Additional performance studies have shown that overall RoCE performance in terms of throughput, response time and CPU usage remains consistent even as multiple VFs are added to share the RoCE Express feature. The bandwidth sharing offered by RoCE virtualization on the z13 also is very efficient. Clearly, the RoCE Express is too good NOT to share! When RoCE is exploited by z/OS with SMC-R, the combined solutions provide two key value points:

- Improved latency that can potentially provide improved transaction rates for latency-sensitive transactional-based workloads.
- Lower CPU cost for workloads that transfer bulk payloads (for example, workloads used for analytics, streaming, FTP, big data, data replication, and web services).

SMC-R preserves the critical qualities of services (load balancing, security, isolation, reuse of IP topology, and so on) that are required by System z clusters in data center networks without requiring any application or middleware changes. In addition, SMC-R introduces minimal operational changes to existing TCP/IP networks.

After you enable SMC-R, you should immediately experience the benefits, and over the long term these benefits can be extended as you expand your exploitation of RDMA technology on System z.

RoCE virtualization provides these additional benefits:

- Fewer physical RoCE Express features are needed to provide redundancy for the SMC-R workloads, and these workloads can be distributed across multiple z/OS instances.
- The bandwidth is effectively doubled, because SMC-R can now use both 10GbE physical ports on the RoCE Express feature.

Configuring and enabling Shared Memory Communications over Remote Direct Memory Access on System z
So what is required to take advantage of SMC-R and RoCE virtualization?

- You must use a System z13 or a z13s.
- You must configure at least one IBM 10GbE RoCE Express feature in the HCD.
- You must apply the PTFs for z/OS CS APARs OA44576 and PI12223.

Figure 2. z/OS Shared RoCE virtualization System View
SMC-R uses TCP/IP for connection establishment, so an IP network is required to connect your peer hosts. With SMC-R your IP network is provided externally with OSA (Ethernet) in QDIO mode (OSD). SMC-R architecture preserves your IP network topology and security. To connect over SMC-R, two hosts must be eligible to connect over IP with direct access to the same Layer 2 network and IP subnet.

Your 10GbE RoCE Express features must be associated with the OSA adapter that is used for your IP network. This association occurs by defining a PNet ID for RoCE and then defining the same PNet ID for OSA. PNet IDs are defined in your I/O definition using HCD, and are dynamically learned by z/OS.

The following are the HCD definition steps:

1. Configure multiple FIDs and VFs for each physical adapter (PCHID) based on your anticipated needs for sharing the adapter. Best practices for HA suggest that, for each TCP/IP stack, you should provision two unique FIDs, each associated with a different physical adapter, for each PNet ID. Configure a unique FID/VF for each physical feature for each z/OS TCP/IP stack that requires access to the feature.

In the following HCD/I/OCDs example, two physical adapters (PCHID 100 and PCHID 200) are defined, and two FIDs are defined for each adapter. All adapters and FIDs reside in the PNet ID NETA. These definitions correspond to the configuration shown in Figure 3.

In the following HCD/I/OCDs example, two physical adapters (PCHID 100 and PCHID 200) are defined, and two FIDs are defined for each adapter. All adapters and FIDs reside in the PNet ID NETA. These definitions correspond to the configuration shown in Figure 3.

FUNCTION FID=1,PCHID=100,VF=10,PART=({LP3},{LP1,LP5}),PNETID=(NETA,NETA),TYPE=ROCE
FUNCTION FID=2,PCHID=100,VF=11,PART=({LP4},{LP2,LP6}),PNETID=(NETA,NETA),TYPE=ROCE
FUNCTION FID=16,PCHID=200,VF=22,PART=({LP3},{LP1,LP5}),PNETID=(NETA,NETA),TYPE=ROCE
FUNCTION FID=17,PCHID=200,VF=23,PART=({LP4},{LP2,LP6}),PNETID=(NETA,NETA),TYPE=ROCE

Figure 3. Configuration definitions

2. Define a matching PNet ID in your OSA (OSD) HCD configuration.

Complete the final configuration steps:

1. In your TCP/IP profile, specify the SMCR parameter on your GLOBALCONFIG statement. Define PCIe function (PFID) values on the SMCR parameter, where the PFID value matches a FID value coded in the HCD. (In the preceding examples, valid PFID values are 1, 2, 16, and 17).

Also on the GLOBALCONFIG statement, specify the PORTNUM keyword to identify which port number you assigned to this TCP/IP stack. For example:

GLOBALCONFIG SMCR PFID 1 PORTNUM 1 PFID 16 PORTNUM 1

2. Verify that an IP subnet mask is configured on the IPAQENET INTERFACE statements. Generally, no other changes are required to your OSA interface statements, because the SMCR parameter is enabled by default.

After your RoCE Express FIDs are “configured on” to each z/OS instance, you can start your associated OSA interfaces and your RoCE Express FIDs are then dynamically started. As new and eligible TCP/IP connections start, they immediately begin exploiting SMC-R.

Is Shared Memory Communications over Remote Direct Memory Access beneficial to you?

Customers who have multiple CPCs in a single site, or an extended LAN among multiple sites, with z/OS centric workloads (such as IBM DB2 or IBM CICS) are natural candidates that can benefit from RoCE Express and SMC-R. You can use the Shared Memory Communications Applicability Tool (SMCAT) to assess whether your specific application workloads might be applicable to SMC-R, or to estimate the potential level of benefits to your network from SMC-R. (See “Is life in the even faster lane for you?” on page 21.)

If you have additional questions about SMC-R, RoCE Express, RoCE virtualization, or about using SMCAT, explore the FAQ and other reference materials provided at the following website:
//www-01.ibm.com/software/network/commserver/SMC/
Is your PKI ready for SHA1’s retirement?

BY WAI CHOI AND BOB GENSLER

SHA1-based signatures have served us well, but the algorithm’s time is nearing its end. Technological advances in processors and networks have made it much easier to forge these signatures, rendering them less secure. Aware of this limitation, certain web browser providers announced plans to withdraw support for non-root certificates that are signed with a SHA1-based algorithm. When that happens, other applications quickly follow, and certificates that are using SHA1-based signatures soon become obsolete.

z/OS PKI Services administrators must anticipate these changes and prepare the certificate authorities (CAs) that they control in order to issue certificates that browsers and applications can accept when SHA1-signature support is removed.

If you are a z/OS PKI Services administrator, how do you prepare for such a change?

Getting started

You have to update your z/OS PKI Services configuration to use a new signature algorithm. If you have several levels of certificate authorities, you might have to change the configurations for each instance z/OS PKI Services in each level, in the correct order. Depending on the strength of the keys that are used by the CA certificate, you might have to renew or rekey the CA certificate as part of this process.

Use the following steps to get started.

1. Determine how many certificate authorities are involved for your installation.
   You might have several levels of certificate authorities, where one acts as a root CA and others act as intermediate CAs. For example:
   
   root CA → intermediate CA 1 → intermediate CA 2

   To update all of the z/OS PKI Services instances in this example, start from the head of the chain. Make your updates in the following order: 1) root CA; 2) intermediate CA 1; 3) intermediate CA 2.

2. Determine whether you need to renew or rekey the CA certificate for each PKI Services instance.
   For the root CA, you can skip this step because trust in it is not based on the hash that is used in its signature. But if you want to change anyway, you can follow the steps indicated in the table for renew or rekey under the column ‘For a root CA’.

   For an intermediate CA, you can skip this step if it already has a SHA2-based signature. Otherwise, you can follow the steps indicated in the table for renew or rekey under the column ‘For an intermediate CA’.

   Whether to renew or to rekey the CA certificate depends on the CA certificate’s validity period and its key strength:

   • If the certificate’s key pair uses a strong key size (2048 bits or above), and less than half of the certificate’s validity period has yet to pass, the renew option might be better. You do not have to distribute the new CA certificate to applications that validate certificates that are issued from this CA because their copies of the original CA certificate validates both old and new certificates that are issued from that CA.

   • If the certificate’s key pair does not use a strong key size, use the rekey option. This option gives you the opportunity to use a key pair with a stronger size, but it requires that you distribute the new CA certificate to all of the applications that must validate its newly issued certificates. These applications must retain the original CA certificate so that older certificates that were issued by it can still be validated.
### To renew a CA certificate:

<table>
<thead>
<tr>
<th>For an intermediate CA</th>
<th>For a root CA (which has stronger key 2048 bits or above)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Create a certificate request (CSR) for the issuing CA by using the RACDCERT GENREQ command:</td>
<td>1. Verify that this instance of z/OS PKI Services has issued certificates for all approved certificate requests.</td>
</tr>
<tr>
<td>RACDCERT CERTAUTH GENREQ(LABEL('CA-cert-label')) DSN(output-dataset)</td>
<td>RACDCERT CERTAUTH GENREQ(LABEL('current-CA-cert-label')) WITHLABEL('new-CA-cert-label' SIZE(2048 or above))</td>
</tr>
<tr>
<td>The CSR is stored in output-dataset.</td>
<td></td>
</tr>
<tr>
<td>2. Send this CSR to the certificate authority that issued your original CA certificate. Request that the certificate be signed with a SHA2-based signature.</td>
<td>2. Shut down the z/OS PKI Services instance for that CA.</td>
</tr>
<tr>
<td>3. When you receive the renewed CA issuing certificate, store the new certificate in a data set.</td>
<td>3. Generate a temporary, self-signed certificate for the issuing CA certificate by using the RACDCERT GENREQ command:</td>
</tr>
<tr>
<td>4. Verify that this instance of z/OS PKI Services has issued certificates for all approved certificate requests.</td>
<td>RACDCERT CERTAUTH GENREQ(LABEL('new-CA-cert-label')) DSN(output-dataset)</td>
</tr>
<tr>
<td>The CSR is stored in output-dataset.</td>
<td>The CSR is stored in output-dataset.</td>
</tr>
<tr>
<td>5. Shut down the z/OS PKI Services instance for that CA.</td>
<td>3. Send this CSR to the certificate authority that issued your original CA certificate. Request that the certificate be signed with a SHA2-based signature.</td>
</tr>
<tr>
<td>6. Add the new certificate to RACF by using the RACDCERT ADD command. Add the certificate without a label. It replaces the original CA certificate:</td>
<td>4. When you receive the renewed CA certificate, store the new certificate in a data set.</td>
</tr>
<tr>
<td>RACDCERT CERTAUTH ADD (dataset_with_new_CA_cert)</td>
<td>4. Roll over the original CA certificate to the new one:</td>
</tr>
<tr>
<td>For a root CA</td>
<td></td>
</tr>
<tr>
<td>1. Verify that this instance of z/OS PKI Services has issued certificates for all approved certificate requests.</td>
<td></td>
</tr>
<tr>
<td>2. Shut down the z/OS PKI Services instance for that CA.</td>
<td></td>
</tr>
<tr>
<td>3. Generate a temporary, self-signed certificate for the issued CA certificate by using the RACDCERT REKEY command:</td>
<td></td>
</tr>
<tr>
<td>RACDCERT CERTAUTH REKEY(LABEL('current-CA-cert-label')) WITHLABEL('new-CA-cert-label')</td>
<td></td>
</tr>
<tr>
<td>4. When you receive the renewed CA certificate, store the new certificate in a data set.</td>
<td>4. Roll over the original CA certificate to the new one:</td>
</tr>
<tr>
<td>5. Verify that this instance of z/OS PKI Services has issued certificates for all approved certificate requests.</td>
<td>RACDCERT CERTAUTH ROLLOVER(LABEL('current-CA-cert-label')) NEWLABEL('new-CA-cert-label')</td>
</tr>
<tr>
<td>6. Shut down the z/OS PKI Services instance for that CA.</td>
<td></td>
</tr>
<tr>
<td>7. Add the new certificate to RACF by using the RACDCERT ADD command. Add the certificate without a label. It replaces the temporary self-signed certificate: RACDCERT CERTAUTH ADD(dataset_with_new_CA_cert)</td>
<td></td>
</tr>
</tbody>
</table>
To rekey a CA certificate (continuation):

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.</td>
<td>Roll over the original issuing CA certificate to the new one that was added in the previous step by using the RACDCERT ROLLOVER command:</td>
</tr>
<tr>
<td></td>
<td>RACDCERT CERTAUTH ROLLOVER('current-CA-cert-label') NEWLABEL('new-CA-cert-label')</td>
</tr>
<tr>
<td>3.</td>
<td>Update the configuration for each instance of z/OS PKI Services.</td>
</tr>
<tr>
<td>1.</td>
<td>Verify that this instance of z/OS PKI Services has issued certificates for all approved certificate requests.</td>
</tr>
<tr>
<td>2.</td>
<td>Shut down the z/OS PKI Services instance for that CA, if it is not already shut down.</td>
</tr>
<tr>
<td>3.</td>
<td>Update the z/OS Services configuration file pkiserv.conf that is used by that CA to begin by using a SHA2-based signing algorithm. This option is called SigAlg1 and is found in the [CertPolicy] section of the configuration file. Change the SigAlg1 keyword value to a SHA2-based signature algorithm name. For example, suppose that you wanted to start using the SHA256 signature algorithm in place of the original SHA1 algorithm. If your configuration file looked like this:</td>
</tr>
</tbody>
</table>
|      | [CertPolicy]
|      | SigAlg1=sha-1WithRSAEncryption |
|      | It now looks like this: |
|      | [CertPolicy]
|      | SigAlg1=sha-256WithRSAEncryption |
| 4.   | Restart the z/OS PKI Services instance for that CA. |

**You’re done!**

**Note:**
Do not create a new instance of z/OS PKI Services just for the purpose of the SHA2-based signature algorithm change.

Creating a new instance of z/OS PKI Services is not an easier approach than reconfiguring your existing z/OS PKI Services instances, or easier than renewing or rekeying your existing issuing certificates. After you create this instance, the old instance still needs to be active to create certificate revocation lists (CRLs) and to respond to OCSP requests. Instead of being a simple fix, you now have more z/OS PKI Services instances to administer and manage.

---

SHA1-based signatures have served us well, but the algorithm’s time is nearing its end.
Let’s configure z13, z13s, and LinuxONE with HCD

BY SIGRUN ELSNER, MARCO SELIG, AND MICHAEL GROETZNER

The IBM z13 server found its little brother in the new IBM z13s server. Both servers support new PCIe adapter types. Additionally, IBM introduced an enterprise and business solution with IBM LinuxONE™ Emperor (Emperor™) and IBM LinuxONE Rockhopper™ (Rockhopper).

IBM Hardware Configuration Definition (HCD) was enhanced to configure the I/O for these new servers. HCD definition support for the z13 and z13s is available for z/OS with the PTFs for APAR OA46010. The IBM LinuxONE (LinuxONE) server definition support is available with the PTFs for APAR OA49573.

The new PCIe adapters differ from the well-known adapters and how to define them with HCD. The article describes what to consider when introducing LinuxONE in your z Systems environment.

The new PCIe adapters

You can use two new PCIe adapters: a physical adapter for Regional Crypto Enablement (RCE) and a virtual adapter for Internal Shared Memory Communication (ISM).

You can configure the RCE adapter similarly to the RoCE adapter for the zEnterprise EC12 (zEC12). RCE features cryptography algorithms and equipment from external providers to meet cryptographic requirements of local governments. The adapter is not available through IBM, but IBM certification of interface compliance is required.

When communicating among different partitions of a server, the new virtual ISM adapter uses IQD or OSA adapters in reducing TCP/IP usage. You can define up to 32 adapters. Each represents a virtual network that can host up to 255 virtual functions. An adapter is defined as a PCIe function type of ISM with a virtual channel ID (VCHID) in the range of 0X’7C0’ to 0X’7FF’. This range is shared with the virtual channel IDs for IQD channels.

Each VCHID represents a separate network and all virtual functions that refer to one VCHID must specify the same physical network ID (PNETID). This PNETID determines whether communication between ISM functions and channel adapters is legitimate.

To support the full set of virtual functions of the ISM adapter, the ranges of PCIe function IDs and virtual function numbers were increased to 4-hexadecimal digits and 3-decimal digits, respectively. See Figure 1.

The new z13s processor

The z13s server announcement includes an update to the z13 server. In HCD, you define the server as a 2965 type with model N10 or N20. This server has I/O capabilities that are similar to the z13. The maximum number of supported CS5 coupling adapters is 16 instead of 32. The server supports 3-channel subsystems, 3-subchannel set, and you can define up to 40 partitions.

Introducing LinuxONE servers

IBM introduced two servers with the LinuxONE announcement: IBM LinuxONE Emperor and IBM LinuxONE Rockhopper. If you define an Emperor with HCD, select the 2964 type with a model L30, L63, L96, LC9, or LE1. If you define a Rockhopper, select the 2965 type with a model L10 or L20.

The LinuxONE system has similar capabilities to z13 and z13s, except for the support of coupling adapters. The LinuxONE servers host Linux and cannot run z/OS and dynamically activate changes. Dynamic activation for z/VM is supported in HCD.
Why would you define the servers in your z/OS IODF if you cannot run z/OS on the LinuxONE? The answer is that you might share your I/O with these servers. To configure your entire server environment, you can rely solely on HCD.

References
smartphones and other mobile devices are rapidly becoming ubiquitous and are being used for everything from buying theatre tickets to getting driving directions. The mobile applications that support these devices can generate lots of transactions that refer to lots of data. z/OS provides a powerful, secure, reliable, and cost effective platform for managing these mobile transactions and data.

IBM Workload Manager for z/OS (WLM) and IBM z/OS Resource Measurement Facility™ (RMF™), when used with the IBM Mobile Workload Pricing option, now give you an easy way to save money on your mobile transactions in z/OS.

Mobile Workload Pricing is a pricing option that offers a discount on processor service that is consumed by eligible software products for transactions that originated on a mobile device. To take advantage of this discount, you need a process, agreed upon by you and IBM, to identify mobile-sourced transactions and report on their consumption.

Up to now, this type of reporting required product-specific tooling from high-volume, transaction-level accounting data, which created a significant overhead. Now, you can simply identify mobile transactions in the WLM service definition, and WLM automatically performs all the data gathering and reporting for you. Together with your favorite performance monitor, such as RMF, you can easily extract your mobile consumption from low-volume SMF records. And you can do that in real time for any kind of work, including CICS and IMS transaction work (with exploiting levels of CICS and IMS), which opens up new possibilities for capacity planning and charge-back.

Identifying mobile work
In your WLM classification rules, you can classify transactions as MOBILE using a new attribute called Reporting Attribute.

Figure 1 shows classification rules for a CICS banking application. The first rule assigns all banking transactions to service class CICSFAST.

And, the mobile attribute is applicable to all IBM-supplied subsystem types, including enclave work like DB2, distributed data facility (DDF), or WebSphere, and CICS or IMS work.

CICS and IMS, which are specifically suited for identifying mobile transactions:
- For CICS, Connection Type (CT) and Transaction Class (TC)
- For IMS, Connection Type (CT) and Client Transaction Name (CTN)

Aggregating and reporting mobile consumption at the service and report class level
WLM aggregates the total and the mobile processor consumption for all service and report classes and reports the results in the Workload Activity Collection service, one of the WLM APIs for performance monitors. That’s the point where RMF comes in. The new processor consumption measurements...
that are supplied by WLM are gathered by RMF Monitor I and stored in SMF record type 72 subtype 3 records in the Service and Report Class Period Data section.

The RMF Postprocessor Workload Activity report provides the new measurements as Transactions APPL%. For the transaction service class CICSFAST, it might look like the following example:

<table>
<thead>
<tr>
<th>REPORT CLASS=BTRAD</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSACTIONS APPL%:</td>
</tr>
<tr>
<td>TOTAL: CP 74.32 AAP/IIP ON CP 0.00 AAP/IIP 0.00</td>
</tr>
<tr>
<td>MOBILE: CP 24.84 AAP/IIP ON CP 0.00 AAP/IIP 0.00</td>
</tr>
</tbody>
</table>

You see total and mobile processor consumption on general-purpose processors (CP), on offload engines (AAP/IIP), and crossover from offload engines (AAP/IIP ON CP).

The data is reported for all service and report classes, including CICS and IMS transaction service and report classes that previously did not report any processor consumption data. So, this is the first time you get processor consumption data for CICS and IMS transactions from low-volume SMF records, which is invaluable for capacity planning and charge-back purposes.

The new values for the report classes BTRAD and BDIRECT might look like the following example:

<table>
<thead>
<tr>
<th>REPORT CLASS=BDIRECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSACTIONS APPL%:</td>
</tr>
<tr>
<td>TOTAL: CP 111.59 AAP/IIP ON CP 0.00 AAP/IIP 0.00</td>
</tr>
<tr>
<td>MOBILE: CP 74.32 AAP/IIP ON CP 0.00 AAP/IIP 0.00</td>
</tr>
</tbody>
</table>

Compared to the overall consumption for transaction service class CICSFAST, 40% of the CICS transaction consumption can be charged back to the traditional banking division (BTRAD), and 60% to the direct banking division (BDIRECT).

You even can see the region overhead of CICS and IMS regions. The data for the CICS region service class might look like the following example:

<table>
<thead>
<tr>
<th>---APPL %---</th>
</tr>
</thead>
<tbody>
<tr>
<td>CP 221.98</td>
</tr>
<tr>
<td>TRANSACTIONS APPL%:</td>
</tr>
<tr>
<td>TOTAL: CP 36.06 AAP/IIP ON CP 0.00 AAP/IIP 0.00</td>
</tr>
<tr>
<td>MOBILE: CP 0.00 AAP/IIP ON CP 0.00 AAP/IIP 0.00</td>
</tr>
</tbody>
</table>

The APPL% CP value contains overall region consumption, including transaction service. In contrast, the Transactions APPL% value is region-only service, excluding transaction service, that is, the region overhead.

RMF also helps you to visualize all the new data using new Postprocessor overview conditions. But, is there any easy way to get these figures into meaningful charts?

Join our journey through RMF

Let’s take a look at some of the new RMF Postprocessor overview conditions for SMF record type 72.3, which are documented in z/OS Resource Measurement Facility User’s Guide, SC34-2664:

- **TAPPLCP**: Total percentage of general-purpose processor time used by transactions
- **MAAPLCP**: Percentage of general-purpose processors used by transactions classified with reporting attribute MOBILE

Both overview conditions can be used with various qualifiers, such as service class or report class period. To summarize the data at policy level, you use the qualifier POLICY.

We used the following overview control statements for our RMF Postprocessor job:

```
OVW(TOTAL(TAPPLCP(POLICY)))
OVW(MOBILE(MAAPLCP(POLICY)))
```

This created an RMF overview report:

```
RMF OVERVIEW REPORT
z/OS V2R2 SYSTEM ID SYS1 START 03/03/2016-04.26.43 INTERVAL 00.00.59
RPT VERSION V2R2 RMF END 03/03/2016-07.56.59 CYCLE 1.000
SECONDS
NUMBER OF INTERVALS 211 TOTAL LENGTH OF INTERVALS 03.30.16
DATE | TIME | INT | TOTAL | MOBILE
03/03 | 04.26.43 | 00.00.16 | 217.2 | 95.3
03/03 | 04.27.00 | 00.00.59 | 236.3 | 104.8
03/03 | 04.28.00 | 00.00.59 | 217.7 | 95.9
03/03 | 04.29.00 | 00.01.00 | 260.9 | 115.6
03/03 | 04.30.00 | 00.00.59 | 260.4 | 115.2
```

Now you can use the RMF Spreadsheet Reporter tool. The Spreadsheet Reporter tool extracts performance measurements from SMF records, produces Postprocessor Report Listings and Overview Records reports, and converts this Postprocessor output into spreadsheets for graphical presentation.

To use this tool, follow these steps:

1. Create a text file with the overview control statements on your workstation.
2. In the Spreadsheet Reporter tool, define a system by clicking Define > System from the menu bar. Specify the overview control statement file in the OVW text field.
3. Define your SMF data set. Click Define > SMF Data from the menu bar.
4. Ensure that your system is active on the Systems tab. Select your SMF data set as input on the Resources tab and create a working set by clicking Create > Working Set.
The Spreadsheet Reporter tool creates the Postprocessor job to create the overview record, uploads it to the host, submits the job, downloads the overview record to the workstation, and converts it into spreadsheet format.

Now, let’s browse the available Excel spreadsheets, included with the Spreadsheet Reporter tool by selecting the All Resources > Local > Spreadsheet resources on the Resources tab. Select Open RMF Overview Spreadsheets.

On the starting page, click Select Overview Working Set and process data and select your recently created Working Set. The working set data is loaded into the spreadsheet.

On each chart, you can decide which data from the overview record you want to display, using the Chart Options button. You can also select the chart type that you prefer, such as line chart or stacked array chart.

From there, you can feed it directly into the IBM Sub-Capacity Reporting Tool (SCRT) and get your mobile discount without the need to gather and analyze gigabytes of transaction-level data. Furthermore, the long-term average total and mobile consumption is available in real time, so you can make informed decisions about adapting your capacity within the bounds of your budget.

Enabling the z/OS Reporting Enhancements for Mobile Workload Pricing

To enable the z/OS reporting enhancements for Mobile Workload Pricing, apply the PTFs for these APARs for z/OS V2R1 and z/OS V2R2:

- WLM: APAR OA47042
- RMF: APAR OA48466
- z/OSMF: APAR PI47638
- IMS Transaction Management System: Available in IMS 14, APARs PI46933 and PI51948
- CICS Transaction Server: Available in CICS 5.3
- SCRT and the billing system: Available in SCRT 23.13.0

For more information

See Announcement of IBM Mobile Workload Pricing, Sub-Capacity Reporting Tool (SCRT) and Mobile/zCAP Workload Reporting Tool (MWRT).

IBM Workload Manager for z/OS (WLM) and IBM z/OS Resource Measurement Facility (RMF), when used with the IBM Mobile Workload Pricing option, now give you an easy way to save money on your mobile transactions in z/OS.
**New LPAR capping options**

*BY KARIN GENTHER, HORST SINRAM, AND GUENTER VATER*

Capping limits the processor consumption of an IBM z Systems logical partition (LPAR), especially LPARs that are running the IBM z/OS operating system.

Capping can be controlled by the PR/SM™ hypervisor, or by the IBM z/OS Workload Management (WLM) component.

First, let’s review some reasons where you might want to use the capping feature:

- Capping might be implemented for pricing reasons, such as to limit the cost with usage-based software pricing models. These pricing models are primarily based on the four-hour rolling average (4HRA) MSU consumption on general-purpose processors. By using capping in this way, the system can run freely while the 4HRA is below the defined limit. You do not have to limit the specialty processor consumption for software pricing reasons. MSU based capping is controlled by WLM.

  However, if your contract specifies a certain MSU number for one or more systems, you would want to permanently enforce that MSU limit regardless of other configuration changes.

- Capping might be required for technical reasons. For example, there might be a requirement to isolate a partition, or a group of partitions, against others. Or, the amount of capacity that is available to a partition should be limited for capacity-based workload routing reasons. When you cap for technical reasons you want to cap always, independently of the 4HRA, and you might want to cap the consumption on specialty processors, such as the zIIP processors, too.

In 2Q16, two new capping options are available to support those requirements:

- WLM absolute MSU capping is available with the PTFs UA81256 (HBB77A0) and UA81257 (HBB790) for z/OS V2R2 and V2R1.

- Absolute capping of LPAR groups is available on the IBM z13 (z13) with the GA2 firmware level and on the IBM z13s (z13s).

Figure 1 summarizes the available capping types.

---

**WLM absolute MSU capping**

In the past, WLM allowed only capping that was based on a four-hour rolling average MSU limit. WLM did not isolate an LPAR by capping it permanently. WLM now provides the option AbsMsuCapping=Yes in its parmlib member IEAOPTxx to always enforce the capping limit. WLM instructs the hypervisor to implement a permanent capping based on the WLM capping limit.

What is the consequence when capping is in effect permanently? When an LPAR is capped by this method, you cannot exceed the defined capping limit, not even for a short time period. For general-purpose processors, this is comparable to LPAR initial capping or LPAR absolute capping.
The main difference to LPAR initial capping is that the limit is defined in absolute MSU instead of relative weight. Therefore, the limit does not change if the configuration changes. It is independent of the weight settings of other LPARs, and does not change if other LPARs are activated, or the server model changes through On/Off Capacity on Demand (CoD) or Capacity Backup (CBU). Because WLM instructs the hypervisor to implement the capping, the capped partition is similar to initial capped partitions.

LPAR absolute capping is close to WLM absolute MSU capping. The main differences are that the WLM limit is defined in MSU instead of fractional number of processors, and that the MSU number that the LPAR is capped to does not change when the physical processor model changes. WLM absolute MSU capping applies only to general-purpose processors.

If the LPAR that is capped with WLM absolute MSU capping is a member of an LPAR capacity group, this LPAR is also capped permanently to its group share. This means that all group members that use this local setting are not allowed to use more than their share at any time.

Because the unused capacity of low utilized group members is made available to other group members, the group limit can be achieved, even if some group members are low utilized.

WLM uses always the minimum of all capping algorithms as the resulting limit. If you defined both a capacity limit and a group limit, the minimum of these limits is enforced by using the capping mode specified by the AbsMsuCapping option.

**Absolute capping of LPAR groups**
The z13 firmware introduces a group capping function that is independent from WLM. Because absolute capping of LPAR groups is an extension to LPAR absolute capping, its limit is also defined on the Hardware Management Console (HMC) as a fractional number of processors. The z13 hypervisor enforces that the LPARs, which belong to the LPAR group, do not use more CPU capacity than defined by the group limit at any time.

For general-purpose processors, absolute capping of LPAR groups is similar to WLM Group Capping with AbsMsuCapping option on all group members. However, the limit is defined in fractional number of processors instead of MSUs.

WLM recognizes this limit in its calculations, but is not involved in the enforcement. The LPAR capacity that is returned by WLM interfaces accepts this limit and WLM algorithms for workload routing account for the limit.

To use the LPAR Group Absolute Capping, apply PTF UA90994 for z/OS V2R2 or PTF UA90993 for z/OS V2R1.

The column CAPPING DEF in the tabular part of this report, as shown in Figure 2, displays the hardware capping options for all reported partitions. Each DEF value is a string where each position of the string is either Y (Yes) or N (No), and has the following meaning:

- The first character (Y or N) indicates whether or not initial capping is set.
- The second character (Y or N) indicates whether or not an LPAR absolute capping limit (maximum number of processors) is defined.
- The third character (Y or N) indicates whether or not an LPAR group absolute capping limit (maximum number of processors) is defined.
- An "*" following a Y or N indicates whether or not there is a change in this interval.

The LPAR group absolute capping values, which are defined at the HMC, are visualized in a new subsection of the RMF Postprocessor CPU Activity report. This ‘Hardware Group Report’ lists the LPAR groups, its members, and the absolute capping limits in numbers of CPUs, as shown in the following example:
New overview conditions (HGCP, HGCIIP, HGCICF, and HGCIFL) are available for the values in this Hardware Group Report. The overview report shows the LPAR group absolute capping limits per processor type. The overview conditions must be specified by using a group identifier that specifies the LPAR group name.

In addition, the RMF Monitor II OPT settings report lists the value of the AbsMsuCapping option that is in effect on a partition.

The capping information is also available in the SMF 70, subtype 1 records. You can find that information in the following sections:

- **CPU Control section:**
  Field SMF70HFF provides an indication as to whether WLM absolute MSU capping was activated.

- **PR/SM Partition Data section:**
  Field SMF70HWGr_Name provides the LPAR group name.

- **PR/SM Logical Processor Data section:**
  Field SMF70HWGr_Cap_Limit contains the LPAR group absolute capping limits for the individual processor types, and field SMF70VPF indicates whether there was a change to the LPAR group settings during the interval.

All of this capping information is available in the Monitor III CPC report and can be accessed by using the HTTP API of the RMF Distributed Data Server. If you prefer to work with a web browser-based interface, you can use the RMF Data Portal or the z/OSMF Resource Monitoring plug-in to view the capping values. To enable the RMF reporting of the capping options, install PTF UA80441 for z/OS V2R2 and PTF UA80443 for V2R1.

Capping can be controlled by the PR/SM hypervisor, or by the IBM z/OS Workload Management (WLM) component.
Handing over manually-activated processor capacity to z/OS CPM for later deactivation

BY DANIELA IMMEL AND PETRA GNECH

If you use IBM z/OS Capacity Provisioning Manager (CPM) for automatic processor-capacity adaption to current workload situations, sometimes you might have to activate new processor capacity manually. This could be the case if the capacity that was authorized by the provisioning policy for additional activation is insufficient.

For example, Bob is a system administrator at a computer center that uses CPM for automatic processor-capacity adaption to current workload situations. CPM decisions are based on its policy. The policy defines when, in which situations, how much, and what kind of processor capacity might be activated. Bob knows that the workload on one of the CPM-managed IBM z Systems CPCs (CPC XY2) is higher than usual. CPM policy allows a limited number of additional MSUs to be activated. Bob decides to proactively activate additional capacity. The manually activated capacity remains untouched by CPM, therefore, Bob is required to deactivate the capacity himself.

Beginning with z/OS V2R1, you can allow CPM to deactivate this processor capacity according to its provisioning policy.

Figure 1 displays the record report of CPC XY2 and the additional general purpose capacity that was manually activated.

Bob activates additional capacity by using the CPM command, ACTIVATE RESOURCE:

A R CPC=XY2 MODEL=730

The z/OS console displays the following CPM message:
CP04103I A change of the manually activated resources has been detected for CPC XY2. The base levels for provisioning management are now 4 CP, capacity level 0, 0 zAAP and 0 zIIP

Figure 2 displays the record report where the additional general purpose capacity is managed by CPM.

Bob passes this capacity to CPM for further management by using the CPM command, MANAGE RESOURCE:

M R CPC=XY2 MODEL=726 KEEPTIME=120

The z/OS console displays the following CPM message:
CP04407I Management for CPC XY2 to model 726 started. Managed resources remain active for at least 120 minutes

Figure 2. CPC XY2 record report – showing CPM managed capacity

Later, Bob wants CPM to deactivate the general-purpose capacity to model 726. After a keep time of 120 minutes, CPM deactivates all additional activated capacity if the workload situation and the policy does not prevent the deactivation. Bob is done. He ensured that the system meets its service level agreements and that the additional capacity is freed. When CPM is done with the deactivation, the system runs as usual.

Options of the MANAGE RESOURCE command

You can use the parameter MODEL on the MANAGE RESOURCE command when you want to manage general-purpose capacity. To transfer special processor capacity to CPM, use the command parameters ZIIP or ZAAP instead of MODEL, followed by the number of temporary processors that are to be managed by CPM.

You can set a special value for the KEEPTIME parameter instead of defining a number of minutes. When you specify KEEPTIME=RECORD on the MANAGE RESOURCE command, CPM leaves the capacity active until the current activation period of the On/Off Capacity on Demand (CoD) record ends. This period usually ends 24 hours after the first activation of the currently active capacity. CPM deactivates all additional capacity at the end of this period if the workload situation and the policy does not prevent deactivation.

For more information, see z/OS MVS Capacity Provisioning User’s Guide, SC34-2661.

Beginning with z/OS V2R1, you can allow CPM to deactivate this processor capacity according to its provisioning policy.
Tired of logging on to multiple TSO/E sessions to perform identical tasks?

BY RITA BEISEL, SHERI DEGROOT, AND BOB PETTI

A frequent grievance among z/OS clients is that ICSF (Integrated Cryptographic Service Facility) requires logging on to a TSO/E user ID and using IBM ISPF (Interactive System Productivity Facility) panels to perform basic administrative functions, such as deactivating a cryptographic coprocessor or disabling I/O updates to key data sets (KDS).

For a single system, this might not seem like such a hardship, but for a large sysplex environment, administrators balk at the need to log on to multiple TSO/E sessions to perform identical tasks on each.

With Cryptographic Support for z/OS V1R13 - z/OS V2R2 (also known as FMID HCR77B1 or web deliverable 15), you can manage certain administrative tasks by using two new operator console commands, DISPLAY ICSF and SETICSF. The commands include functions that were previously only available by using ICSF ISPF panels or the ICSF installation options data sets and also provide new functionality and information.

**Tip:** These commands are also available on z/OS V1R13 with the PTF for APAR OA47380 installed.

The DISPLAY ICSF command returns information about your system, such as the cryptographic device firmware level, the number of requests that are active on each device, KDS format, sysplex communication levels, and the active ICSF code change date. These commands can be processed on all systems in your sysplex, if requested.

Additionally, with the SETICSF command, you can enable or disable TKDS updates, which was previously limited to a CKDS and a PKDS. The SETICSF OPTIONS parameter, with the RPSEC keyword, provides a configuration setting not available in the installation options data set.

**DISPLAY ICSF command**
The DISPLAY ICSF command displays the following information:

- The status for available cryptographic devices
- Certain ICSF options
- Information pertaining to active KDS
- The status of the master key registers for the available cryptographic devices
- The systems that are available to participate in commands with a SYSPLEX scope and current ICSF code level

**Tip:** When you add the SYSPLEX=YES keyword to the DISPLAY ICSF command, it increases the scope of the command to include all the participating members of the sysplex.

**Examples**
Issue the DISPLAY ICSF,CARDS command to display information about the cryptographic devices available on the system or sysplex:

D ICSF,CARDS,SYSPLEX=NO

CSFM668I 09.35.33 ICSF CARDS 292

ACTIVE DOMAIN = 003
CRYPTO EXPRESS5 COPROCESSOR 5C36
STATUS=Active SERIAL#=99EA6059 LEVEL=5.0.45
REQUESTS ACTIVE=0002
CRYPTO EXPRESS5 COPROCESSOR 5P40
STATUS=Active SERIAL#=8706099 LEVEL=02.09
C LiC=0742
REQUESTS ACTIVE=0001
CRYPTO EXPRESS5 ACCELERATOR 5A42
STATUS=Active
REQUESTS ACTIVE=0000

Issue the DISPLAY ICSF,KDS command to display information about the active -KDS on the system or sysplex:

D ICSF,KDS

CSFM668I 14.38.31 ICSF KDS 040

CKDS RACFDRVR.SHERID.CKDSPLX
FORMAT=KDSR COMM LVL=3 SYSPLEX=Y MKVPs=DES AES
PKDS RACFDRVR.SHERID.PKDSPLX
FORMAT=KDSR COMM LVL=3 SYSPLEX=Y MKVPs=RSA ECC
TKDS RACFDRVR.SHERID.TKDSPLX
FORMAT=KDSR COMM LVL=3 SYSPLEX=Y MKVPs=P11
Issue the `DISPLAY ICSF,MKS` command to display information about the master key on the system or sysplex:

```
D ICSF, MKS
CSFM668I 09.45.18  ICSF MKS 852
SYSNAME: SYSA  DOMAIN: 003  CPC Name: R01
```

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>SERIAL#</th>
<th>STATUS</th>
<th>AES</th>
<th>DES</th>
<th>ECC</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA P11</td>
<td>99EA6059</td>
<td>Active</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>5P39</td>
<td>97006054</td>
<td>Active</td>
<td></td>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>

Issue the `DISPLAY ICSF,LIST` command to display the members of a sysplex who are eligible to participate in the `DISPLAY ICSF` and `SETICSF` commands:

```
D ICSF, LIST
CSFM668I 14.57.29 ICSF LIST 984
Systems supporting SETICSF and DISPLAY ICSF commands:
  SYSA HCR77B1 DOMAIN = 003
  SYSB HCR77B1 DOMAIN = 003
```

Issue the `DISPLAY ICSF,OPTIONS` command to display the values for certain ICSF installation options:

```
D ICSF,OPTIONS
SYSA CSFM668I 11.36.35 ICSF OPTIONS 995
SYSNAME = SYSA ICSF LEVEL = HCR77B1
LATEST ICSF CODE CHANGE = 01/09/15
  Refdate update interval in Days/HH.MM.SS = 030/00.00.00
  Refdate update period in Days/HH.MM.SS = 000/01.00.00
  MASTERKCVLEN = display 3 digits
```

### SETICSF command

Use the `SETICSF` command to complete the following tasks:

- Activate, deactivate, or restart a cryptographic device or a list or range of cryptographic devices
- Change a subset of ICSF’s installation options
- Enable or disable updates to a KDS

**Tip:** When you add the `SYSPLEX=YES` keyword to the `SETICSF` command, it increases the scope of the command to include all the participating members of the sysplex. The output of `SETICSF` commands, when you specify `SYSPLEX=YES`, includes information about the execution of the command on the other participating members of the sysplex.

### Examples

Issue the `SETICSF DEACTIVATE` command to deactivate the specified device or devices. Because the last active device was deactivated, messages are issued about the change in availability of the master keys.

```
SETICSF DEACTIVATE, INDEX=42
CSFM134I CRYPTOGRAPHIC FEATURE IS INACTIVE.
CRYPTO EXPRESS5 COPROCESSOR 5C42, SERIAL NUMBER 99EA6055 RSN=Deactivated.
CSFM505I CRYPTOGRAPHY - THERE ARE NO ACTIVE CRYPTOGRAPHIC COPROCESSORS.
*CSFM131E CRYPTOGRAPHY - DES SERVICES ARE NOT AVAILABLE.
*CSFM131E CRYPTOGRAPHY - RSA SERVICES ARE NOT AVAILABLE.
```
Issue the SETICSF ACTIVATE command to activate the specified device or devices. Because this was the first device that is activated, messages are issued about the change of master key availability.

SETICSF ACTIVATE,SN=99EA6055,SYSPLEX=y
CSFM136I CRYPTO EXPRESS5 COPROCESSOR 5C42, SN 99EA6055
STATUS CHANGED FROM Deactivated TO Initializing stage 1.
CSFM111I CRYPTOGRAPHIC FEATURE IS ACTIVE. CRYPTO
EXPRESS5 COPROCESSOR 5C42, SERIAL NUMBER 99EA6055.
CSFM400I CRYPTOGRAPHY - SERVICES ARE NOW AVAILABLE.
CSFM130I CRYPTOGRAPHY - DES SERVICES ARE AVAILABLE.
CSFM130I CRYPTOGRAPHY - RSA SERVICES ARE AVAILABLE.
CSFM130I CRYPTOGRAPHY - ECC SERVICES ARE AVAILABLE.
CSFM127I CRYPTOGRAPHY - AES SERVICES ARE AVAILABLE.
CSFM667I 11.08.29 SETICSF Activate 559 CRYPTO
EXPRESS5 COPROCESSOR 5C42
SERIAL#=99EA6055 SYA Active
Systems without a compatible ICSF level for commands started: SYSB

Issue the SETICSF DISABLE command to disable updates for the specified key data set. The valid KDS specifications are CKDS, PKDS, or TKDS:

SETICSF DISABLE,CKDS
CSFM667I 11.17.33 SETICSF IO Cntl 1 566
I/O DISABLED ON SY1 FOR CKDS
CSF.CKDS.VARLEN2

Issue the SETICSF ENABLE command to enable updates for the specified key data set. The valid KDS specifications are CKDS, PKDS, or TKDS:

SETICSF ENABLE,CKDS
CSFM667I 11.18.01 SETICSF IO Cntl 1 569
I/O ENABLED ON SY1 FOR CKDS
CSF.CKDS.VARLEN

Issue the SETICSF OPTIONS command to change the value of an ICSF option. You can specify the MKCVLEN, RISEC, and RPSEC options:

SETICSF OPTIONS,RISEC=600
CSFM667I 11.21.43 SETICSF Options 572
Reference date interval in seconds:
00000600

For more information about these new console commands, see z/OS Cryptographic Services Integrated Cryptographic Service Facility System Programmer’s Guide, SC14-7507.

A frequent grievance among z/OS clients is that ICSF (Integrated Cryptographic Service Facility) requires logging on to a TSO/E user ID and using IBM ISPF (Interactive System Productivity Facility) panels to perform basic administrative functions, such as deactivating a cryptographic coprocessor or disabling I/O updates to key data sets (KDS).
N
o matter the I/O configuration, hardware failures can occur that might impact a common hardware component. By default, the IBM z/OS I/O supervisor (IOS) recovers such errors on a device-by-device basis, and recovery processing removes the path from a single device. Other I/O requests on the problematic path to different devices on the same logical control unit (LCU) might encounter the same errors. Even though devices have multiple paths, a single path error might cause an application I/O request to time out because of recovery actions that z/OS takes on the failing path for each device.

By enabling IOS path recovery options, the system can act more proactively to remove a failing path sooner. The limited recovery time option can also aid in triggering an unplanned HyperSwap sooner.

Let’s take a closer look at how to use IOS recovery options to limit recovery time and how to determine appropriate error thresholds that result in IOS taking a path recovery action.

**Limiting recovery time to trigger a HyperSwap sooner**

When hardware problems happen, we don’t want IOS recovery actions to delay application I/O requests for too long. The limited recovery time option (LIMITED_RECTIME) can help reduce the impact time and trigger a HyperSwap sooner when all paths to a device are affected.

If a device becomes unresponsive and leads to MIH timeouts, dynamic pathing validation occurs. Dynamic pathing validation involves issuing Sense Path Group ID (SNID) channel commands down each path that is configured for use by the device. If the device still does not respond to the SNIDs, path validation will likely take a long time before it fails. Without the LIMITED_RECTIME setting, a SNID from dynamic pathing validation can be retried multiple times after each timeout. Setting a LIMITED_RECTIME value limits the SNID timeout value and avoids retries. In any environment, LIMITED_RECTIME causes a problematic path to be removed sooner.

This is beneficial in a HyperSwap environment because, if all paths to the device are unresponsive, we want path validation to fail quickly and trigger an unplanned HyperSwap sooner.

For example, let’s say we have eight paths that are configured to each DASD with the MIH time set to 30 seconds. A situation arises that causes a DASD to be unresponsive on all paths. Such a failure would likely first appear as a 30-second MIH timeout to a device, which would trigger dynamic pathing validation for each path to the device. Here is where the limited recovery time option comes in. If LIMITED_RECTIME=2, dynamic pathing validation would include only a single SNID command on each path, timed at 2 seconds each. After dynamic pathing validation times out for all eight paths, the device would be boxed. In a HyperSwap environment, the boxed device triggers an unplanned HyperSwap, which swaps to the secondary storage volumes, and application I/O can proceed.

Figure 1 illustrates the effect of the LIMITED_RECTIME parameter on the overall recovery time in a HyperSwap environment.
In the figure, the maximum I/O impact time is calculated as the sum of the following values:

**a:** The MIH timeout value

**b:** The time for dynamic pathing validation to complete:

- Without LIMITED_RECTIME: \( b = \text{number of paths} \times \text{at least one 15-second timeout with multiple retries} \)
- With LIMITED_RECTIME: \( b = \text{number of paths} \times \text{LIMITED_RECTIME value} \)

**c:** The maximum time impact on application I/O requests while a HyperSwap completes after it has been triggered.

In our example, if no LIMITED_RECTIME value is set, the value of \( b \) would be \( 8 \times 15 \), or 120 seconds, times some number of retries. However, if the LIMITED_RECTIME value is set to 2, the value of \( b \) is \( 8 \times 2 \), or 16 seconds, with no retries.

In a HyperSwap environment, if you want to minimize the dynamic pathing validation time, \( b \), shown in Figure 1, ensure that you set LIMITED_RECTIME to a low value, such as 2 seconds.

### Using path recovery options to reduce recovery time

Many devices can share common hardware components. When a common component fails, the impact affects multiple devices, even if the devices share a common hardware component. By using a PATH_SCOPE value of CU, along with the PATH_THRESHOLD and PATH_INTERVAL parameters, you can reduce the amount of time and the number of errors that it would take for IOS to remove a path to a logical control unit.

When you specify PATH_SCOPE=CU, for certain types of errors, IOS removes the path from all devices that are defined to a logical control unit. Additionally, for certain intermittent channel-path errors, the system collects statistics, and if the number of errors reaches or exceeds the threshold over the specified interval, IOS removes the path from all devices that are defined to the LCU. (For a more detailed explanation of the errors that the system monitors, see “Usage notes for RECOVERY” for the IECIOSxx parmlib member in z/OS MVS Initialization and Tuning Reference, SA23-1380.)

By using the PATH_SCOPE=CU parameter along with the PATH_INTERVAL and PATH_THRESHOLD options, you can reduce the elapsed time that it takes for IOS to remove a problematic channel path to a logical control unit.

For example, assume that the following IOS recovery parameters are in effect:

- PATH_SCOPE=CU
- PATH_INTERVAL=1
- PATH_THRESHOLD=3

This combination of parameters means that the system would take a path offline to an LCU if three errors are detected for that path to that LCU within 1 minute. If three errors for the same path for one LCU are detected within 1 minute, it suggests that the path is experiencing a problem such that it would be advisable to take the path offline until the problem can be corrected.

The PATH_INTERVAL value is the number of consecutive minutes during which the PATH_THRESHOLD value must be met to cause IOS to remove the path. So, a PATH_INTERVAL value of more than 1 minute makes it less likely that path removal will occur. With path recovery options, the system will not take offline the last path to a device.

### Using system commands to manage IOS recovery

In addition to setting recovery options in the IECIOSxx member of parmlib, you can use the SETIOS command to dynamically set or change the options. For example, to limit DASD recovery time to 2 seconds, issue the following command:

```
SETIOS RECOVERY,LIMITED_RECTIME=2,DEV=DASD
```

The D IOS,RECOVERY command displays the recovery option settings. For example:

```
D IOS,RECOVERY
IOS103I 11.40.45 RECOVERY OPTIONS 367
LIMITED RECOVERY TIME IS 2 SECONDS
LIMITED RECOVERY IS REQUESTED FOR DASD
PATH RECOVERY SCOPE IS BY DEVICE
DCCF IS SET TO MESSAGE
```

The D M=DEV(dddd,(pp)) command provides information about why a path was taken offline. For example, the following command displays information about the path to device 1014 on channel path 40:
D M=DEV(1014,(40))
IEE174I 14.22.18 DISPLAY M 221
DEVICE 01014 STATUS=ONLINE
CHP 40
ENTRY LINK ADDRESS 05
PATH ONLINE N
CHP PHYSICALLY ONLINE Y
PATH OPERATIONAL Y
MANAGED N
CU NUMBER 1000
DESTINATION CU LOGICAL ADDRESS = 00

... PATH OFFLINE DUE TO THE FOLLOWING REASON(S):
BY OPERATOR
PATH RECOVERY
CONTROL UNIT INITIATED RECONFIGURATION
CONFIGURATION MANAGER

If you specified PATH_SCOPE=CU, and the path reaches the threshold, the system varies the path offline for all devices on the logical control unit. The following message is displayed: IOS102I PATH RECOVERY INITIATED FOR PATH pp ON CU cccc REASON=txt.

After the paths are taken offline, ensure that the problem that caused the path to be removed is fixed. Then, issue the VARY PATH(dddd,pp),ONLINE command to bring the path online to one device first. If all seems fine, issue the V CU(cunumber,pp),ONLINE command to bring the path online to the other devices on that LCU. If the entire channel path was taken offline, issue the CF CHP(xx),ONLINE command.

By enabling IOS path recovery options, the system can act more proactively to remove a failing path sooner. The limited recovery time option can also aid in triggering an unplanned HyperSwap sooner.

zFavorites

Go to the zFavorites website, where you will find the latest information for all of your z Systems needs, including product documentation, software, ISV development, marketing information, education, links to downloads, and much, much more!

ibm.com/systems/z/os/zos/library/zfavorites

Scan with your phone’s QR code reader
Suppose that you want to create applications on an IBM z/OS system with web technologies, or you want to develop an application that is language and platform-independent, but can reach to z/OS data.

The z/OS data set and file REST interface gives you the ability to access and manage z/OS data sets in this way.

The z/OS data set and file Representational State Transfer (REST) interface is part of IBM z/OS Management Facility (z/OSMF), which was introduced as “the new face of z/OS” with the intent to simplify and modernize z/OS. As a part of z/OSMF strategy to make z/OS more reachable, the z/OS data set and file REST interface services were introduced in z/OSMF version 2.1 with PTF UI16044 and were continually enhanced in z/OSMF version 2.2. You can now use following services that are related to data set operations:

• List data sets that match a specific filter
• List partitioned data set members
• Retrieve the contents of a sequential data set or partitioned data set member
• Write data to a sequential data set or partitioned data set member
• Create a sequential data set or partitioned data set
• Delete a sequential data set or partitioned data set

You can learn more about these services in IBM z/OS Management Facility Programming Guide, SA32-1066, especially if you are also interested in using the z/OS data set and file REST interface services with UNIX files and directories.

List data sets matching a specific filter

When your application needs to access a data set on z/OS, the first thing you might want to know is whether a specific data set exists, or a serial of data sets exist. With the z/OS data set and file REST interface, you can use the following URL with an HTTP GET method: /zosmf/restfiles/ds/?dslevel=<filter-criteria>

The parameter dslevel is used to identify the data set to be listed. For example, you can specify dslevel=SYS1.SAMPLIB to identify whether SYS1.SAMPLIB is on your system, or more generally, you might want to specify the dslevel parameter as your user ID, for example, dslevel=IBMUSER, to see how many data sets contain your user ID as a high-level qualifier.

Optionally, if you want to know the attributes of the data set, such as record format, record length, and block size, you can specify the customer header X-IBM-Attributes=base in the request; all the attributes available on the ISPF 3.4 panel are returned.

List partitioned data set members

When you are working with a partitioned data set, you might want to know how many members there are and what they are named. You can list members of a specific partitioned data set by issuing an HTTPS GET method with the following URL path: /zosmf/restfiles/ds/<data-set-name>/member

The <data-set-name> parameter specifies the data set that you are working on. You can also filter the result by specifying the parameter pattern=<filter-criteria> for your data set; thus, only those members whose names meet the specified criteria can be returned. Optionally, if you want to know the attributes of the members, you can specify the custom header X-IBM-Attributes=base in the request.

Retrieve the contents of a sequential data set or partitioned data set member

After you determine that the specific data set or member exists, the next step is to retrieve the contents of the data set or member. To obtain these results, issue an HTTPS GET method with the following URL paths:

/zosmf/restfiles/ds/<data-set-name>
/zosmf/restfiles/ds/<data-set-name>(<member-name>)

If you have some data sets that are not cataloged, you can specify the volume in which they reside on in the URL path like this:

/zosmf/restfiles/ds/-(<volser>)/<data-set-name>
/zosmf/restfiles/ds/-(<volser>)/<data-set-name>(<member-name>)

The custom header X-IBM-Data-Type indicates whether data conversion is to be performed on the returned data. You can set the value as text, binary, or record:
A value of text means that the data is converted from EBCDIC to the character set specified on the “Content-Type” header on the request, and a newline (NL) character from the response charset is inserted between logical records.

A value of binary means that the data is not converted but returned as is.

A value of record means that the data is not converted but each logical record would be preceded by the 4-byte big endian record length of the record that follows.

Write data to a sequential data set or partitioned data set member

You can modify the data set with your own data by using the HTTPS PUT method with the following URL paths:

/zosmf/restfiles/ds/<data-set-name>
/zosmf/restfiles/ds/<data-set-name>(<member-name>)
/zosmf/restfiles/ds/-(<volser>)/<data-set-name>
/zosmf/restfiles/ds/-(<volser>)/<data-set-name>(<member-name>)

The URL paths of this service are the same as those used when retrieving data from data set or member except that the HTTP PUT method is used instead of the HTTP GET method. You can also use the X-IBM-Data-Type header in this reverse data process.

Tip: If you write data to a member that does not exist in the data set, the member is created.

Create a sequential data set or partitioned data set

To create your own data sets on z/OS, use an HTTPS POST method with the following URL path:

/zosmf/restfiles/ds/<data-set-name>

Specify a JSON object with the following fields in the request body:

```json
{
  "volser" : "<volume>",
  "unit" : "<device type>",
  "dsorg" : "<data set organization>",
  "alcunit" : "<unit of space allocation>",
  "primary" : "<primary space allocation>",
  "secondary" : "<secondary space allocation>",
  "dirblk" : "<number of directory blocks>",
  "avgblk" : "<average block>",
  "recfm" : "<record format>",
  "blksize" : "<block size>",
  " Irecl" : "<record length>",
  "storclass" : "<storage class>",
  "mgntclass" : "<management class>",
  " dataclass" : "<data class>
}
```

All the properties in the preceding JSON document have corresponding options in the ISPF 3.2 panel that is traditionally used to allocate a data set, which means you can also specify a storage class, management class, and data class to manage the data set that you are allocating.

Delete a sequential data set or partitioned data set

If you are authorized, you can delete data sets or members that are not used anymore. Issue HTTPS DELETE with the following URL paths:

/zosmf/restfiles/ds/<data-set-name>
/zosmf/restfiles/ds/<data-set-name>(<member-name>)
/zosmf/restfiles/ds/-(<volser>)/<data-set-name>
/zosmf/restfiles/ds/-(<volser>)/<data-set-name>(<member-name>)

The z/OS data set and file REST interface provides another way to access data sets on z/OS and simplifies the development of your applications on z/OS. If you have specific services that you think might benefit from using the data set and file REST interface, please visit the IBM developerWorks® Request for Enhancement (RFE) site to send us your request! IBM RFE Community: https://www.ibm.com/developerworks/rfe/
Whittling down dumps

BY GIRIJA VARANASI, PURVI PATEL, AND RALPH SHARPE

Are your supervisor call (SVC) dumps larger than life? In the past, the amount of data captured depended on the storage area in which the data resided. The SVC dump allowed specification of these areas with parameters such as CSA (common system area), SQA (system queue area), LSQA (local system queue area), RGN (address space Private Area), and a few others. The system does not have the information to decide what data is useful for an application’s problem diagnosis. That is left up to the requester of the SVC dump.

In the beginning of 64-bit addressability support, VSM provided High Virtual (HV), which was private. Because no application “is an island,” the HV evolution continued with mechanisms to share the HV storage. First came HV shared, which was complex and cumbersome. Then came HV common, which was simpler and easier to use. Applications took advantage of that. Guess what’s taking longer to capture those gigabytes of data? Even with the advent of HV storage, the dumping interface remained unchanged in an effort to keep things simple. Under the covers, below-the-bar relationships were simply expanded to encapsulate the introduced HV areas.

VSM introduced memory objects (MOMBs) to represent the HV storage. When a MOMB is created, a dump attribute can be associated with it. Specifying SVCDUMPTRGN=YES (or DUMP=LIKERGN, as appropriate) with an HV private MOMB would lead to the storage being dumped when RGN was requested. Otherwise, the storage would have no dump attribute associated with it (SVCDUMPTRGN=NO, or DUMP=NO, as appropriate), and the application would have to maintain the addresses that would need to be supplied to the LIST64 parameter if the storage is to be dumped. For the “uncommon” HV common, there are dump attributes of DUMP=LIKECSA and DUMP=LIKESQA. It is uncommon because “likeness” has little to do with the actual nature (like memory “map” location, or fixed versus pageable attributes) of the storage, compared with the usage for below-the-bar areas. The associations began to result in dumping virtually unlimited amounts of data.

“Houston, we have a problem!”

Before the 64-bit (HV) era, dumping everything used to be less of an issue. The amount of irrelevant data was not a burden worth addressing. Times have changed! But the traditional interface would force applications to:

• Get the storage without dump-able attributes, plus

• Track that storage as lists of address ranges, to be supplied with the dump request; yet another cumbersome, and likely error-prone, procedure.

A two-pronged approach

The most pressing issue appeared to be the use of HV common storage. Some applications need storage that is available from any address space, but that storage does not need to be dumped all the time. For this, we present a tactical solution that could also be made available in the IBM Service stream for earlier releases.

The general problem to handle both HV common and private storage needed a strategic solution to be exploited by applications.

Tactical solution

To address the HV common storage volume issue, SDUMP introduced three new, “never-before-seen” SDATA filter options, namely HCSAByAsid, HCSANoOwner, and HCSASysOwner. Unlike parameters to be used by applications on the SDUMPX macro, they are to be used by an installation using the operator console commands (CHNGDUMP, DUMP and/or SLIP). These options are independent from each other, as well as the CSA option; therefore, they can be combined (see Figure 1). For compatibility, when only the CSA option is specified, all of the above-and-below the bar CSA (LIKECSA) storage is included in the dump. However, when these options are specified with the CSA option, the amount of high virtual CSA storage captured in the dump is reduced. The following figures describe how these new HV CSA SDATA options affect the CSA storage capture in an SDUMP:

<table>
<thead>
<tr>
<th>Specified SDATA option or options</th>
<th>CSA storage that is included in the dump</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSA</td>
<td>All above the bar and below the bar CSA storage.</td>
</tr>
<tr>
<td>CSA, HCSAByAsid, HCSANoOwner, HCSASysOwner</td>
<td>All below the bar CSA storage, high virtual CSA storage that is owned by the ASIDs included in the dump, high virtual CSA storage that is no longer owned by an active ASID, and high virtual CSA storage that belongs to the SYSTEM.</td>
</tr>
<tr>
<td>HCSAByAsid, HCSANoOwner, HCSASysOwner</td>
<td>The dump does not include high virtual CSA storage that is owned by the ASIDs that are excluded from the dump.</td>
</tr>
<tr>
<td>(neither CSA nor any of this HCSAxxxx options)</td>
<td>All high virtual CSA storage that is owned by the ASIDs included in the dump, high virtual CSA storage that is no longer owned by an active ASID, and high virtual CSA storage that belongs to the SYSTEM.</td>
</tr>
<tr>
<td>Non below the bar CSA storage is included in the dump.</td>
<td>No below the bar CSA storage is included in the dump.</td>
</tr>
</tbody>
</table>

Figure 1. New SDATA filter options
Strategic solution
Thankfully, VSM provided a feature that SDUMPX did not exploit at its inception. An 8-byte memory token can be associated with one, or more than one, MOMB. The token value can be supplied by the application or generated by VSM for the application to use. Wouldn’t it be nice if one could then provide that token with the dump request? That would be a superior way to account for storage to be dumped, rather than trying to maintain lists of storage address ranges. So, SDUMP will now allow applications to exploit this capability for their dumps. Introducing:

The SDUMPX UTOKEN Parameter
In z/OS V2R2, SDUMPX was enhanced to accept those tokens using the UTOKEN parameter. VSM advertises the use of their MOMB token for storage management. We propose that it also be used as a “dump-able attribute.” The SDUMPX interface allows the flexible use of tokens for both HV common and HV private storage area dumping. Applications may use only one token, or several separate tokens, for multiple-use cases.

Consider the following scenarios:

Application A uses HV common because any address space in the system may use its services. When a problem occurs in one of those address spaces, the SVC dump would benefit from having that HV common included.

Application B gets HV private within address spaces that invoke its services. When a problem occurs, that storage is interesting, but the other HV private areas are not.

The SDUMPX parameter allows you to specify:

- A token, plus
- Whether that token should apply to HV common versus HV private storage, plus
- Which ASID, taking part in the dump, that HV private token applies to

References
To find out more, see these z/OS publications:

z/OS MVS Programming: Authorized Assembler Services Reference, Volume 3 (LLACOPY-SDUMPX), SA23-1374

z/OS MVS System Commands, SA38-0666

Are your supervisor call (SVC) dumps larger than life?
RMF keeps a close eye on zFS performance

BY KARIN GENTHER AND PETER MAILAND

One way to monitor the performance of IBM z/OS File System (zFS) in a sysplex is to use the set of QUERY commands provided by zFS. Keeping an eye on zFS performance got easier in IBM z/OS V1R7 when IBM Resource Measurement Facility (RMF) introduced two Monitor III reports that collected data about the z/OS File System component of the IBM z/OS Distributed File Service base element. Those reports were redesigned and a new added in V2R2 when zFS evolved into a sysplex-aware shared file system.

The three Monitor III sysplex reports, ZFSOVW, ZFSFS, and the new report, ZFSKN, show important system-wide measurements, drill down into detailed numbers, reference data from previous intervals, and accumulate data from multiple ranges.

ZFSOVW: zFS overview report
ZFSOVW, the zFS overview report, provides information about various aspects of the response times of zFS requests. It contains percentages of time that requests had to wait, and it reports on the important hit ratios for each cache that is present in the zFS implementation.

To view the details about the behavior of any of the caches, position the cursor on any number in the corresponding block and press the ENTER key. A pop-up window displays all available measurements on request rates, hit and delay percentages, and storage utilization.

ZFSFS: zFS file system report
Because zFS has not supported multi-file system aggregates since z/OS V2R1, the term “zFS file system” is now synonymous with the term “zFS aggregate.” To combine the metrics for the multi-file system aggregates and the zFS file system, RMF redesigned the zFS file system report, ZFSFS. The report displays the following statistics about each file system:

- Its name and owner
- Connected systems
- Space utilization
- A selection of usage characteristics

A pop-up window triggered by cursor-sensitivity shows additional in-depth measurements.

When you invoke the ZFSFS report for the first time, you will see measurements for all zFS file systems from all systems that are defined in the sysplex. Use the options for the Monitor III report to tailor the amount of displayed data. To invoke the options, enter the RO command while ZFSFS is displayed. In the options window, narrow the scope of the ZFSFS data by selecting a file system name, ending with a wildcard (*) if needed. To choose the correct name, look at the list of available zFS file systems in the lower part of the panel. You can also specify whether the report is to contain summary data only for the sysplex or contain detailed data from each system.

Figure 1 shows an example of a tailored report.
**ZFSKN: zFS kernel report**

The new ZFSKN report provides the most basic measurements of overall zFS performance. It displays those measurements from all systems in one view. For example:

<table>
<thead>
<tr>
<th>System</th>
<th>Avg Response Time 10%</th>
<th>Avg Response Time 50%</th>
<th>Avg Response Time 90%</th>
<th>User Cache Request Rate</th>
<th>Vnode Cache Request Rate</th>
<th>Metadata Cache Request Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>SYSE</td>
<td>36.3</td>
<td>2.9</td>
<td>17.8</td>
<td>0.033</td>
<td>26.43</td>
<td>99.3</td>
</tr>
<tr>
<td>SYSF</td>
<td>106</td>
<td>0</td>
<td>69.6</td>
<td>1.033</td>
<td>88.0</td>
<td>1800</td>
</tr>
</tbody>
</table>

This window displays information about request rates to locally and remotely owned file systems. It also displays information about requests that require XCF calls and on-average response times.

Last, but not least, RMF integrates the zFS performance metrics with the RMF Distributed Data Server and its HTTP API. Use the RMF Data Portal to view complete zFS reports (see Figure 2), sort them if necessary, or pick a single metric.

The new zFS performance metrics are integrated into the z/OSMF Resource Monitoring plug-in.

For more information about the zFS sysplex reports, see z/OS RMF Report Analysis, SC34-2665.

The three Monitor III sysplex reports, ZFSOVW, ZFSFS, and the new report, ZFSKN, show important system-wide measurements, drill down into detailed numbers, reference data from previous intervals, and accumulate data from multiple ranges.
Some changes in z/OS V2R2 that you need to know before you migrate

Advice from Professor Kimura

Now it’s time to start the migration to z/OS V2R2! Having some insights on changes and enhancements can help you prepare your system for a successful migration.

Exploitation of the new IEAVTABX_EXIT by Fault Analyzer
Prior to z/OS V2R2, Fault Analyzer (FA) required a dump capture exit, IDIXDCAP, to be installed in the IEAVTABX installation exit list. This exit is installed by the IDITABX USERMOD. Typically, the IEAVTABX exit process is only called if the job step allocated a dump by using the SYSMDUMP, SYSUDUMP, or SYSABEND Ddbname statement. However, the FA IDITABD USERMOD permits the IEAVTABX-defined exits, including IDIXDCAP, to be invoked regardless of any JCL dump DD statement specification.

In z/OS V2R2, a new dynamic exit called IEAVTABX_EXIT receives control for all ABDUMP dump requests. This enhancement is included in the PTF for APAR OA48457. Its interface is similar to the existing IEAVTABX interface, but the difference is that the dynamic exits are always called regardless of either of these conditions:

• Whether or not an ABDUMP is taken

• Whether or not an ABDUMP DD statement was specified in the job’s JCL steps.

This new function was exploited by FA. For z/OS V2R2, the FA dump capture exit, IDIXDCAP, is installed during system initialization as the IEAVTABX_EXIT dynamic exit routine by including the following statement in a PROGxx parmlib member:

EXIT ADD EXITNAME(IEAVTABX_EXIT) MODNAME(IDIXDCAP)

To activate the IDIXDCAP exit before the next IPL, issue the operator command: SET PROG=xx

IDIXDCAP is invoked for all abends where RTM will drive ABDUMP processing, regardless of whether the job step has allocated a dump by using the SYSMDUMP, SYSUDUMP, or SYSABEND Ddbname statement.

Restriction: With the z/OS V2R2 PTF for APAR OA48457 installed and the appropriate FA PTF described below installed, the USERMODs IDITABX and IDITABD are now not applicable to z/OS V2R2 or later. This will help reduce the number of usermods that need to be installed with z/OS V2R2.

Install the following FA PTFs to prevent a IDI0095W message from being issued whenever IDIXDCAP is invoked by using the dynamic exit interface:

• V13 APAR PI46839 (PTF UI31227)

• V12 APAR PI46838 (PTF UI31231)

• V11 APAR PI46837 (PTF UI31220)


Increase of JES2 virtual storage allocation
The functions in JES2 such as Job Execution Control (JEC), Checkpoint Extend and Alter processing, and Larger limits (JQEs, JOEs, BERTs) requires the new z22 checkpoint mode in z/OS V2R2. To support the new mode, the virtual storage requirement was increased in z/OS V2R2 JES2 regardless of whether you are using z11 or z22 checkpoint mode.

z/OS V2R1 JES2 is the last release level that will support z2 checkpoint mode. z/OS V2R2 JES2 can run in either z11 or z22 checkpoint mode.

The default mode on a JES2 cold start is controlled by the COLD_START_MODE option on the JES2PARM OPTSDEF statement, which is for checkpoint mode z22, by default.

Tip: Set the COLD_START_MODE value to the checkpoint mode that JES2 is expected to be running in so that an unplanned cold start does not result an unplanned activation. When you issue the $ACTIVATE command to dynamically change the checkpoint mode, the message $HASP260 includes this recommendation if it finds the inconsistency. JES2 31-bit virtual storage allocation was increased by about 200 MB and the largest increase was observed in SP132 Key1
(CTENT: Checkpoint Table Entry) to support the increased limits for JQEs, JOEs, and BERTs. The virtual storage size is the same regardless of whether you are running in z11 or z22 mode. As for the JES2 64-bit virtual storage allocation, the checkpoint I/O area (about 800 MB) was moved in z/OS V2R2 to a 64-bit storage location from a 31-bit storage.

**Tip:** The `$JDDETAILS(STORAGE)` command reports the JES2 31-bit virtual storage allocation size in your configuration. Also, beginning in z/OS V2R2, SDSF (DA, I, INIT, NS, and ST) provides a new Job Memory (JM) panel that you can use to obtain the memory map for a specific address space, including the 64-bit virtual storage.

You must check the REGION size of the JES2 procedure, if specified, to prevent an ABEND878 situation during JES2 initialization. Additionally, you must check the maximum possible size of extended private storage in your system because, for example, a large ECSA allocation might not tolerate the increase of JES2 EPVT allocation in z/OS V2R2.

For more information, check out my blog on the IBM z Systems Development Blog site: [https://ibm.biz/Bd4ZVR](https://ibm.biz/Bd4ZVR)
Thin provisioning for z/OS

BY NICK CLAYTON, LISA GUNDY, AND CAROL MELLGREN

The main advantage to thin provisioning implementations is the trade-off between performance and space-efficiency. The way users benefit from the trade-off depends on the granularity of capacity allocation.

Thin provisioning is now available for count key data (CKD) devices on the IBM DS8000® and supported for use in z/OS environments. This article reviews the concepts and benefits of thin provisioning, and provides an overview of the implementation for z/OS.

Thin provisioning concepts

The main advantage to thin provisioning implementations is the trade-off between performance and space-efficiency. The way users benefit from the trade-off depends on the granularity of capacity allocation.

Allocation on Demand is where the storage system allocates physical space to a volume only when data is written. Typically, the volume is divided into parts called extents. When each extent is first written, it is allocated on the backend storage. The size of the extent is often a compromise between capacity efficiency (smaller is better), and performance or metadata overheads. Smaller extents can reduce performance and increase memory requirements for metadata.

The ratio of the logical capacity to the physical capacity of a storage system is called the over-provisioning ratio. When the logical capacity is larger than the physical capacity, a storage administrator can provide volumes to users without purchasing all of the backing storage. An administrator can delay, even avoid purchasing this capacity if it is not consumed by the user.

The risk of over-provisioning is that users could consume all the available physical storage in a short space of time, which would result in access loss for any volume where additional storage was needed. Typically, when thin provisioning is first implemented, an over-provisioning ratio of 1:1 is used. Then, when the behavior of the environment is understood, a ratio such as 2:1 or higher might be used.

Thin provisioning on DS8000

The IBM DS8870 and prior DS8000 models provided a capability known as Track space-efficient (TSE) volumes. These were extremely space efficient models with an effective extent size of one track, but the metadata overheads and performance impact meant that active workloads with significant sequential write activity did not provide sufficient performance. TSE devices were designed to be used as FlashCopy targets providing a space-efficient FlashCopy capability.

In Release 8.1, the IBM DS8880 introduced a change to the virtualization layer. Small extent support provides the basis for an efficient thin provisioning implementation for z/OS, giving the option to use 21 cylinders (instead of 1113 cylinders) as the unit of allocation in a storage pool. These Extent space-efficient (ESE) devices provide a much higher performance replacement for TSE volumes and space-efficient FlashCopy although with a larger allocation unit and hence slightly less capacity efficiency.

To prevent out-of-space conditions, notifications are sent to the host operating system when the storage pool reaches certain thresholds. There are three thresholds defined:

- A user settable threshold (that is, 20 percent free space)
- 15 percent remaining physical space
- No available free space

The DS8000 provides interfaces to release space on a thinly provisioning volume either at a full volume or extent level. On distributed systems, these can be exploited by host software to release space when files are deleted.

In the initial release, space release is not allowed on a volume that is the source of a copy services relationship. However, FlashCopy can optionally perform a space release of a FlashCopy target when the relationship is established or withdrawn and when an initial copy of Metro Mirror, Global Copy or Global Mirror is performed space will be released on the secondary to reflect the unallocated space on the primary.

Thin provisioning for z/OS

z/OS environments have long been regarded as providing very efficient use of storage such that thin provisioning would present minimal benefits. However, this assumption is often false in environments with multiple DFSMS storage groups,
or multiple individual systems and sysplexes, which share a common storage system. When storage administrators allocate volumes on request, to project or application specific storage groups based on projected demand, this is little different from behavior on distributed systems.

In distributed systems, we often ask how “thin provisioning friendly” a particular operating system or filesystem is to understand how data is distributed over the available logical volume capacity; hence, how close the allocated space will be to the space actually used by the server. z/OS can be regarded as thin provisioning-friendly, because data sets are allocated in the free space nearest to the start of the volume and the allocations are generally spread evenly over all volumes in a storage group. Because of this, the higher-order cylinders in each volume are typically not used unless the entire storage group becomes full.

Thin provisioning can also provide benefits for replication function such as Metro Mirror and Global Mirror. If the space is not allocated on the primary device, it does not need to be copied when the volume is first replicated and when resynchronizing or sending data to the secondary with Global Copy the bitmaps for unallocated extents do not have to be individually scanned.

Figure 2. Extent space efficient volume optimization with replication

The z/OS Global Mirror (zGM) System Data Mover (SDM) software is not currently thin provisioning aware and so while thin provisioning can be used on a zGM primary device the secondary device will always become fully allocated when the initial synchronization is performed. IBM recommends that devices that could become zGM secondary devices are always fully provisioned.

Thin provisioning can also be used for FlashCopy target devices to provide a high performance space-efficient FlashCopy. Extents on the FlashCopy targets are only allocated when writes occur to the source which result in a copy of data to protect the previous version in the FlashCopy. As well as providing short term FlashCopies for backup this can also be used with a “Golden Copy” concept where several FlashCopies are taken from a read only source to provide multiple space-efficient instances of the same data for testing or other purposes.

z/OS Storage software and thin provisioning

Upon an ICKDSF initialization for a volume (INIT), if the device is an ESE volume, a space release command will be issued to the device to be initialized.

This will ensure always starting with a “clean slate” with no leftover allocated extents for a newly initialized volume. Unlike the original space-efficient support on DS8000, no new keywords will be required on replication commands (for example, FlashCopy Establish) in order to use thin provisioned volumes as target devices so thin provisioning will be transparent to replication management software.

When DFSMSdss is used to move data, it is actually a copy operation followed by a delete of the source. In the case where FlashCopy is used to perform the copy portion of the operation, the delete is performed as soon as the FlashCopy Establish is complete. With space efficient volumes, an out-of-space condition during background copy results in the loss of the data, since the source has been deleted and the target becomes unavailable. To prevent this loss, for DFSMSdss move (copy with delete) operations, the FlashCopy Establish pre-allocates the extents for the target tracks. This ensures that, even if the storage pool runs out of space, the copy operation can complete without risk of data loss.

As mentioned above, the storage control sends alerts to all connected hosts when certain physical space consumption thresholds are reached. These alerts are captured by the host and highlighted messages are issued to the system console. With multiple threshold values, including the user settable threshold, this provides plenty of opportunity for user intervention, or automation, to take action to relieve space constraints long before running out of physical storage.

In addition to the alerts from the DS8000, z/OS Version 2 Release 2 introduced storage group thresholds. This new function tracks storage consumption at the storage group level and can also provide a mechanism to manage over-provisioned volumes by setting the new track and cylinder managed space thresholds and monitoring new messages IGD400I and IGD401I.

If thin provisioned volumes are placed in a storage group separate from full provisioned volumes, then the thin provisioned volumes can actually be over-provisioned, and an alert is raised when more physical storage is required for that storage group. For instance, if the desired ratio of virtual storage to physical storage is 2:1, volumes can be configured as 1 TB volumes, but only 500 GB per volume can be consumed before running out of space. The threshold can be set to 20 percent, so when half of the physical storage (200 GB per volume) is consumed, an alert is raised. If no relief is provided, another alert will be raised at 30 percent, and so on.

Future direction

The initial release of thin provisioning on the DS8000 provides significant benefits to clients for use as host volumes and with space-efficient FlashCopy. Additionally, IBM plans to provide software that will enable any extents on a volume that are allocated but only contain free space to be freed on the backend storage and the space returned to the pool. This will provide further efficiency benefits.
2015 US and Canada IBM Master the Mainframe contest

The winners announced

BY TROY CRUTCHER

It was another successful season for the 2015 US and Canada Master the Mainframe contest. Total registration reached more than 4,800 students, all vying for the top prize spots. To achieve the top spot, students had to pass three parts of the contest.

Part 1: Breaking the ice
The students download emulators to access the contest operating system: z/OS, z/VM, or Linux on z Systems. Because the contest is “no experience necessary,” we walk them through the operating systems and show them how to access the systems. All 2,000 “Part 1” winners received a custom Master the Mainframe t-shirt to sport around campus in.

Part 2: Practical experience
Next, students dive into what each OS can do. The students work with JCL and SDSF, setting permissions by using RACF, and try z/TPF. The students are exposed to Bluemix, programming languages such as Assembler, C, COBOL, and Java. The 75 “Part 2” winners received IBM awards in the form of sweatshirts, messenger bags, and sports bottles.

Part 3: Real-world challenge
Finally, students worked on a real-world project in which they were required to sort and manipulate real world census data in a way that could benefit real companies. In this challenge, 44 students competed for the top five spots. A major component of the judging process was to run each student’s program. The judges looked for unique ways in which the students created basic reports and analysis reports. The first-place student created a completely interactive GUI application to show the data. The five winners received awards that included an Apple iPad Air 2, along with bragging rights!

To achieve the top spot, students had to pass three parts of the contest.
Let’s hear from our winners

First Place:  
Ari Kenney, Lake Brantley High School  
Ari participated in the contest previously. He plans to attend a university to study computer science and political science.  
“Participating in this contest for the past four years has increased my understanding of the mainframe and solidified my intention to study and work in computer science. The variety of problems presented appeals to many skills taught in high school computer science classes and acts as a great supplement to what I’ve learned in school.”

Second Place:  
Henry Liu, Algonquin Regional High School  
Henry also participated in the contest previously. He plans to attend a university to study computer science and artificial intelligence.  
“Doing Master the Mainframe for four years has really familiarized me with the industrial and practical side of computing. When I first competed, I didn’t really like mainframes because I thought their text-only interface directly translated into a cumbersome machine, but four years later I’ve come to realize just how immensely powerful and practical mainframes can be.”

Third Place:  
Stephen Solis-Reyes, University of Western Ontario  
Stephen is in his fourth year at the University of Western Ontario studying computer science. This was his first year competing; the competition was a requirement for one of his courses.  
“Before this contest, I had never worked with or really thought about the mainframe. But I’m now seriously considering mainframe technology as a possible career path after I graduate.”

Fourth Place:  
Michael Johnson, West Texas A&M University  
Michael is new to the Master the Mainframe Contest this year. He is studying computer science at West Texas A&M University. Originally a math major, he decided to give computer science a try and he loved it.  
“My experience with the mainframe has opened my eyes to a whole other area of computing that is fun and interesting. Before I knew about the mainframe, I really wasn’t sure about my career goals. I know I would enjoy working with the mainframe in the future.”

Fifth Place:  
Kenneth Vahl, Mohawk College  
Kenneth is in his second semester at Mohawk College where he studies software development. This is his first year participating in the Master the Mainframe Contest.  
“Before the contest, I wasn’t even aware that working with mainframes was a possible career path. Now I’m considering working with a mainframe as a possibility after I graduate.  
These students have so much to be proud of. The unique programming skills that these winners possess will surely allow them to become valuable to enterprise companies long into the future.”
Free COBOL classes for all!

The COBOL programming language supports numerous Fortune 500 businesses every day, making proficiency with it a much-sought-after IT skill in high demand.

Interskill Learning, a premier online mainframe training provider, and affiliate of Arrow, an IBM Global Training Provider, is offering their COBOL e-learning curriculum at no cost from now until October 31, 2016. This exclusive, first-time offering is open to all IBM z Systems clients, and to all educators and students who are currently investigating, teaching, or studying enterprise computing.

Boost your technical career today! Register for this no-cost offering and learn about the COBOL language, the structure, and basic programming techniques as you progress through six courses and 27 hours of self-paced, structure training, complete with assessments to track your personal progress.

Hurry up and register today for this free, limited-time offer.

For more information, visit: http://ibm.biz/FreeCOBOL

P.S. Tell all of your friends!
Our contributors

Karla Arndt is a Senior Software Engineer and the team leader for Predictive Failure Analysis and Runtime Diagnostics. She has published several articles and holds several patents in the areas of predictive technologies, autonomic computing, and referential integrity.

Rita Beisel is a technical writer at IBM. She is currently writing for Integrated Cryptographic Services Facility (ICSF), Trusted Key Entry (TKE), Encryption Facility for z/OS, and Cryptographic Services System Secure Sockets Layer Programming (SSL). She has written for RACF, z/VM, DB2, DFSMS, and MVS in the past. She also likes to do interface design, usability testing, and surveys.

Li Bo joined IBM in June 2012 and has been working in z/OSMF for three years. He is responsible for designing and developing z/OSMF workflows, including automation features.

Chris Brooker is a Senior Software Engineer currently leading the IBM zAware team. He started working on the mainframe for IBM in 2001 after receiving his M.S. degree in Computer Science and Software Development from Marist College.

Pierre Cassier is a senior certified IT specialist based in Montpellier, France, working for the GDPS test team. He has worked with IBM z for 40 years, as system programmer for a large customer, and then for IBM as an IBM z performance expert. He has worked with GDPS Active/Active since the inception of the product.

Eric Chabert is a senior certified IT specialist based in Montpellier, France, working for the GDPS/Active-Active Test team. He has worked with IBM z for over 25 years, with a variety of products, but principally with CICS, DB2, problem determination tools, and GDPS Active/Active.

Jian Guo Chen is an Advisory Software Engineer with 12 years of experience on the mainframe. He currently works on the z/OS Management Facility developer team in IBM development lab in Beijing, China.

Wai Choi, CISSP*, is a Senior Software Engineer for IBM in Poughkeepsie. She works on digital certificate support in RACF, PKI services and Kerberos. Wai actively participates in conferences and forum on certificate-related topics.

Nick Clayton works for IBM systems development as the solution architect for the DS8000. Nick graduated from Trinity College Cambridge in 1994 with a degree in Mathematics and has been with IBM for 17 years.

Ross Cooper, CISSP*, is a senior software engineer for the z/OS security design and development team where he has worked on the RACF, PKI, Kerberos, and ICSF products.

Troy Crutcher is the Academic Initiative Master the Mainframe project manager. He is focused on getting enterprise systems skills to university professors and their students, thus allowing for a smooth transition into the industry.

Frank De Gilio is the Chief Architect for Cloud in the IBM Systems Group. Frank is a Distinguished Engineer in the World Wide Client Centers. Frank helps clients use IBM’s latest technology to modernize their IT environments. His latest focus has been on making z/OS the premier Cloud Platform for high level Cloud Services.

Sheri DeGroodt is a Function Tester currently working on Integrated Cryptographic Services Facility (ICSF) in Poughkeepsie, NY. Before joining the ICSF test team, she worked primarily in RACF Function Test.

Sigrun Elsner is a software developer at IBM Boeblingen and has been a member of the HCD development team since 2002.

Jodi Everdon is an Advisory Software Engineer at IBM Poughkeepsie, NY. She is the lead writer for the IBM Wave for z/VM Hallmark Project. Previously, Jodi was the Global Information Planner for z/OS, Executive Editor of z/OS Hot Topics, and a writer for many z/OS elements and z/OS Problem Management.

Bob Gensler is a security software developer for z/OS RACF and PKI Services. His 28 years of experience include: operating system and software infrastructure development for AIX and the pSeries HPC platforms.

Karin Genther has been with IBM at the Boeblingen lab for 14 years. She holds a degree in Computer Science from the University in Darmstadt, Germany. During her IBM career, she has worked in System Automation and RMF.

Michael Gierlach is an Advisory Programmer in z/OS Communications Server in Research Triangle Park. Mike has worked on VTAM, TCP/IP and z/OS Communications Server for 25 years, most recently on the new Shared Memory Communications enhancements to z/OS Communications Server.

Anthony Giorgio is an Advisory Software Engineer with 15 years of experience on the mainframe. He currently works on the Hypervisor Performance Manager team in IBM Poughkeepsie, NY. Anthony holds a BS and MS in Computer Science from the NYU Polytechnic School of Engineering. Anthony can be reached on Twitter at @a_giorgio

Petra Gnech is a software engineer for z/OS Capacity Management at IBM Boeblingen Lab in Germany. Petra holds a degree in Computer Science from the
advanced technical college of Furtwangen, Germany. Petra has been with IBM for 22 years.

Michael Groetzner is an advisory IT specialist for z/OS programming at IBM Germany. His expertise includes z/OS Capacity Provisioning and Hardware Configuration Definition (HCD). Michael holds a degree in Computer Science from the University of Cooperative Education in Stuttgart, Germany.

Lisa Gundy works for IBM systems development as an architect for DFSMS.

David Herr is a Senior Software Engineer in IBM Systems Unit Enterprise Networking Solutions development team. He is the performance analyst and a developer for z/OS Communications Server, focusing on TCP/IP technologies, High Availability functions, and zEnterprise networking.

Matthew Hunger is a z/OSMF Client Support representative, also known as “Level 2.” As you read this, he’ll be celebrating his second anniversary with IBM. Matt is using IBM Design Thinking skills to help shape the future of z/OSMF.

Daniela Immel is a software engineer at IBM Germany. After 6 years in z Systems firmware development, she joined the z/OS Capacity Provisioning development team in 2010.

Edrian Irizary joined IBM in 2015, and is a UI developer on IBM zAware. He has worked on IBM’s biggest open source contribution to the Open Mainframe Project. He has also worked on a side project that created a proof of concept mobile app for IBM’s Runtime Diagnostics component.

Gus Kassimis is a Senior Technical Staff Member with IBM Systems Unit Enterprise Networking Solutions architecture, strategy, and design team. He is a designer for the z/OS Communications Server, focusing on TCP/IP technologies, High Availability functions, and zEnterprise networking.

Shigeki Kimura is the technical lead for z/OS migration at IBM Japan. He has expertise in and deep knowledge of z/OS release-to-release migration and has participated in key reviews of z/OS Migration book since 2006. He has been contributing articles to z/OS Hot Topics since 2010. Shigeki has been with IBM for 30 years.

Randall Kunkel is a Senior Software Engineer in IBM Systems Unit Enterprise Networking Solutions development team. He is a developer for z/OS Communications Server, focusing on TCP/IP technologies, SNA HPR/EE, and zEnterprise networking.

Peter Mailand is a software engineer with IBM Systems in Germany. He joined IBM 1996 as a member of the RMF development team and has 20 years of experience in the performance monitoring area.

Justin McCoy is an Advisory Software Engineer with 13 years of experience at IBM. He’s sculpting a simplified world through software and art and is currently focused on building cloud infrastructure on the z platform. Justin can be reached on Twitter at @mccoyjus

Carol Mellgren is a Distinguished Engineer working for IBM systems development.

Qais Noorshams is a Software Engineer working on the zEvent project since early 2015. He received his Master’s degree and his Ph.D. degree in computer science in 2010 and his interests broadly include performance-related topics and future-oriented technologies.

Dan Patel is a Senior Software Engineer with the IBM Systems Unit Enterprise Networking Solutions z/OS Communications Server team, focusing on performance. He has over 30 years of experience with z Systems, IBM operating systems, and network communications.

Purvi Patel is a Service Aids lead at IBM, Poughkeepsie. She joined IBM in 1999 and has worked in IOS and Service Aids components.

Bob Petti has been a member of the ICSF development and test team for 10 years. Prior to ICSF, he worked on other z/OS security and directory components such as LDAP, SSL, DCE, and zFS. Bob can be reached at bobpetti@us.ibm.com

Tabor Powelson is a software engineer in Poughkeepsie, NY, working in z/OS I/O Supervisor (IOS) Level 2. He joined IBM in 2009 after receiving his B.S. degree in Software Engineering and Mathematics from Clarkson University.

Eric Rosenfeld, CISSP, joined IBM as a software engineer in 1985 and has been developing security products most of his career. He is currently a designer and developer for TSO/E.

Marco Selig joined the HCD team in 2015 as an IT specialist shortly after obtaining his PhD in physics.

Ralph Sharpe is a software engineer at IBM. Since joining IBM in 1977, he has had roles in 3081 engineering, MVS performance and design areas, and settled in the MVS operating system programming area being responsible for development and service programming of Dumping services.

Horst Sinram is a Senior Technical Staff Member in the IBM Germany Research and Development lab in Boeblingen, Germany. His responsibilities include z/OS Workload Management and Capacity Management. Horst is a frequent speaker at major conferences and also works frequently with clients world-wide.

Geoffrey K. Smith is a Senior Software Engineer at IBM Poughkeepsie, NY. He has worked at IBM for 34 years. He is the lead information architect and information strategist for z/OS documentation. Using his direct involvement with customers, Geoff directed the design and development of IBM Library Server, Softcopy Librarian, and, most recently, IBM Knowledge Center for z/OS.

Achim Staebler is a cloud and mobile expert focusing on all aspects of application development. He is an experienced lab services consultant with deep knowledge in the JEE and Javascript world. Achim holds a Diploma of Computer Sciences from the University of Stuttgart and is passionate about front-end tools, DevOps, and PaaS (platform as a service).
Dr. Irene Stahl is a software engineer with the z/OS Workload Management team in IBM Boeblingen. After receiving her doctorate in Computer Science, she has worked for over 20 years in various z Systems projects, the last 8 years in z/OS Workload Management.

Donna Stenger works for the IBM Systems and Technology Group in Poughkeepsie, NY. She is an Advisory Software Engineer in the I/O Supervisor component of z/OS. She joined IBM in 1989 after receiving her M.S. degree in Mathematics from SUNY Oneonta. Her current role is as an I/O Supervisor Level 2 team member. She specializes in the HCD component area and has worked on testing various HCD / IOS scenarios.

Jerry Stevens is a Senior Technical Staff Member with IBM Systems Unit Enterprise Networking Solutions z/OS architecture, strategy and design team, focusing on communications hardware architecture, protocols and related solutions. He has over 30 years of experience with System z, IBM operating systems and network communications.

Girija Varanasi is a software programmer in the Service Aids Development team at IBM Poughkeepsie, NY. Since joining IBM in 1997, she has worked in MSYS for Setup, Supervisor, and Allocation components.

Guenter Vater is a software engineer with IBM Systems in Germany. He joined IBM in 1997 as a member of the WLM development team and has almost 20 years of experience in the workload management with focus on capping technologies.

Robert Vaupel has over twenty seven years with IBM Germany after acquiring a Master’s degree in computer science at the University of Karlsruhe. During his time with IBM he worked on various performance and system management related topics, like Resource Measurement Facility, and z/OS Workload Management Development. Robert is System Technical Staff Member and is responsible for future developments of z/OS WLM and Mobile extensions for Systems Management functions.

Steve Wall is a senior certified IT specialist based in Montpellier, France, working for the GDPS/Active-Active Development team. He has 30 years of experience of working with IBM z, with CICS TS Development, in the z/OS Benchmark Center in the Montpellier and with GDPS/Active-Active Development.

Marna Walle is an IBM Senior Technical Staff Member in z/OS System Install at IBM Poughkeepsie. Her current responsibilities include z/OS release migration and new installation technologies of the z/OS platform. Marna is a frequent speaker on z/OS installation and exploitation topics at System z Technical University, SHARE, and other customer conferences.

Bruce Wells, CISSP, is a senior software engineer for the RACF design and development group, where he has worked for the past 25 years. He has extensive experience in both the z/OS and z/VM versions of the product, and his recent contributions include enhancements to password security, the RACF Remote Sharing function, and multi-factor authentication.

Glenn Wilcock is a Senior Technical Staff Member with the IBM Systems Unit in Tucson, AZ. He is the lead architect for z/OS DFSMShsm, which provides automated, policy-driven space management and backup/recovery for enterprise systems data. Glenn has 25 years of experience developing storage software solutions.

Scott Woolley, CISSP, is a seasoned IBMer who continues to wrestle with security in and out of the office. He embraces the challenge of making the world a little safer and a little happier, even just one KLOC and one person at a time.

Mike Young is the team lead of z/OS Management Facility Client Support, also known as “Level 2.” He’s been involved with z/OSMF since its first release and is on a mission to improve its serviceability with every PTF.
I wandered lonely as a cloud
That floats on high o'er vales and hills, ...

And then my heart with pleasure fills,
And dances with the daffodils.

– William Wordsworth
IBM z/OS Platform for Apache Spark publications available

Apache Spark is an open source cluster computing framework optimized for extremely fast large-scale data processing. See the following:

IBM z/OS Platform for Apache Spark Administrator’s Guide, SC27-8451

IBM z/OS Platform for Apache Spark Installation and Customization Guide, SC27-8449

IBM z/OS Platform for Apache Spark Solutions Guide, SC27-8452

IBM z/OS Platform for Apache Spark User’s Guide, SC27-8450