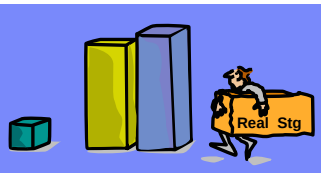
Performance Engineering & Tuning for WebSphere Version 6 & 7 on z/OS

July, 2009

John Hutchinson
IBM Washington Systems Center
hutchjm@us.ibm.com



ON DEMAND BUSINESS™

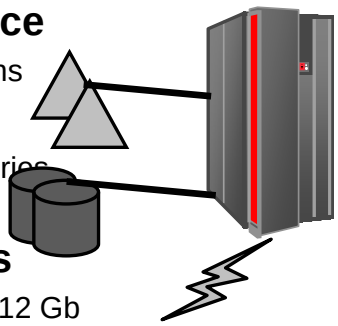
See Techdoc article PRS2494 for the latest copy of this presentation.

Agenda

- **Engineer for Performance**
  - Hardware Resources & Configuration
  - Software Levels: z/OS, WAS, & Java
  - Systems, Subsystems, & Security
  - System Topology - Client/Server placement
  - Application Server Configuration Options
- **Tune your Runtime**
  - Workload Manager Controls & Classification
  - Java Tuning
- **Monitoring Performance**
  - System and Application Monitors
  - Isolating Performance Problems
- **Appendix**
  - Tools & Documentation

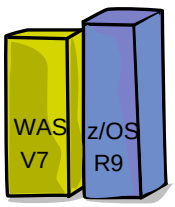
Hardware Configuration

- **System z & zSeries provides superior performance**
  - Cycle speed, Super scalar, IEEE FP, Crypto, New H/W instructions
- **zAAPs and zIIPs can reduce Costs (TCO)**
  - Application Assist Processors (zAAPs) on Systems z10, z9 & zSeries
  - Integrated Information Processors (zIIPs) on System z10 & z9
- **More Storage required than traditional workloads**
  - System z10 can have up to 1.5 Terabytes; System z9 can have 512 Gb
    - Minimum entry system 1.5 Gb (sandbox testing)
    - Real world Application Server, 1 Gb or more per servant region (JVM).
  - Paging is BAD!
- **Parallel Sysplex & Coupling Facility for Production**
  - RRS Logstreams, RACF, Error logs, DB2 data sharing
- **Cached DASD**
  - System Libraries, HFS/zFS, Application Data, Logs
- **OSA Express 2**
  - Gigabit Ethernet, 10 Gigabit Ethernet, 1000BASE-T Ethernet



Software Configuration - Latest Software Levels Best

- **z/OS 1.9**
  - WLM improvements for zAAPs, New LDAP server
  - CFRM, z/OS XML System Services
  - LE, XPLink, USS asynchronous socket read and write
- **z/OS 1.10**
  - HiperDispatch & Capacity Provisioning Manager
- **DB2 V.9**
  - LOB improvements, Index optimization, Multi-row fetch
- **WebSphere V 6.1**
  - Web Container, EJB improvements, Web services, Imbedded Messaging
- **WebSphere V 7**
  - JDK improvements
  - Servant/Controller communication optimizations
  - Codepath improvements
- **Java 1.4, 5.0, 6.0 SDK**
  - JIT & GC performance enhancements with every release
  - SDK 5.0 showing ~30% performance improvement over 1.4.2
  - SDK 6.0 showing ~60% performance improvement over 5.0



Software configuration - Why WAS on z/OS?

- **Mainframe** qualities of robustness - not *'Mainframe like'*

  Bold items help Performance:

  – **Hardware** - CPU, Storage, I/O Subsystem, Storage protect, MTTF
  – Operating System - Isolation, Recovery, Architecture
  – Virtualization – LPAR, IRD
  – **Optimizations** - Hyper-channel, Local TCP Stack Optimization
  – **Workload Management** – zWLM, IRD, Sysplex Distributor
  – GDPS or DR - Recovery based on capacity not box duplication
  – **Capacity planning** & Utilization - WLM & RMF reporting
  – Storage management – DFSMS, Backup, File sharing
  – **Sysplex distributor** - Client access distribution of TCP connections
  – **Scalability** - MQ shared queues, DB2 data sharing, etc.
  – Secure, Manageable environment

- Benefits of "just showing up" on z/OS (Mike Cox)

Optimization – z/OS exploitation

- **LOCALCOMM** (Path-length and latency avoidance)
  – Cross memory services to communicate between Servers rather than TCPIP
  – SSL avoidance, Security and WLM context propagated
  – Type-2 resource managers (IMS, CICS, MQ, DB2)
- **Thread affinity**
  – Dispatch stays on same thread if app. components in same server
  – Reduces communication costs
- **Common DataSpaces** used for shared memory
  – Avoids communication costs & allows for light weight serialization
- **Multi-system ENQ**
- **RRS** for transaction support
- **Encryption** - IBMJCECCA
- **IBM JDK** - zAAP exploitation, JZOS, JRIO, RACF

WebSphere for z/OS leverages zSeries architecture

What's new in WAS V6.1 Performance

z/OS and distributed common code base

- **Improved performance with Java 5 (SDK 1.5)**
  - ►Improved JIT compiled code efficiency
  - ►Improvements in Software Crypto performance
  - ►New memory allocation and garbage collection schemes
  - ►Java class cache in shared memory for faster startup time
- **Improved Web Container performance/scalability**
  - ►Caching enhancements
  - ►JSP engine improvements
- **EJB improvements**
  - ►Code path improvements
  - ►Higher performance access intent settings
  - ►Optimizations to persistence manager
  - ►Light weight Entity Beans

64-bit JVM mode available !

- **Improved Web services performance**
  - ►New XML parsing technology
  - ►Other web services improvements
- **Imbedded messaging**
  - ►Code path improvements
  - ►Option to use file system as message store
- **Misc.**
  - ►Finer grain authentication optimizations for data sources

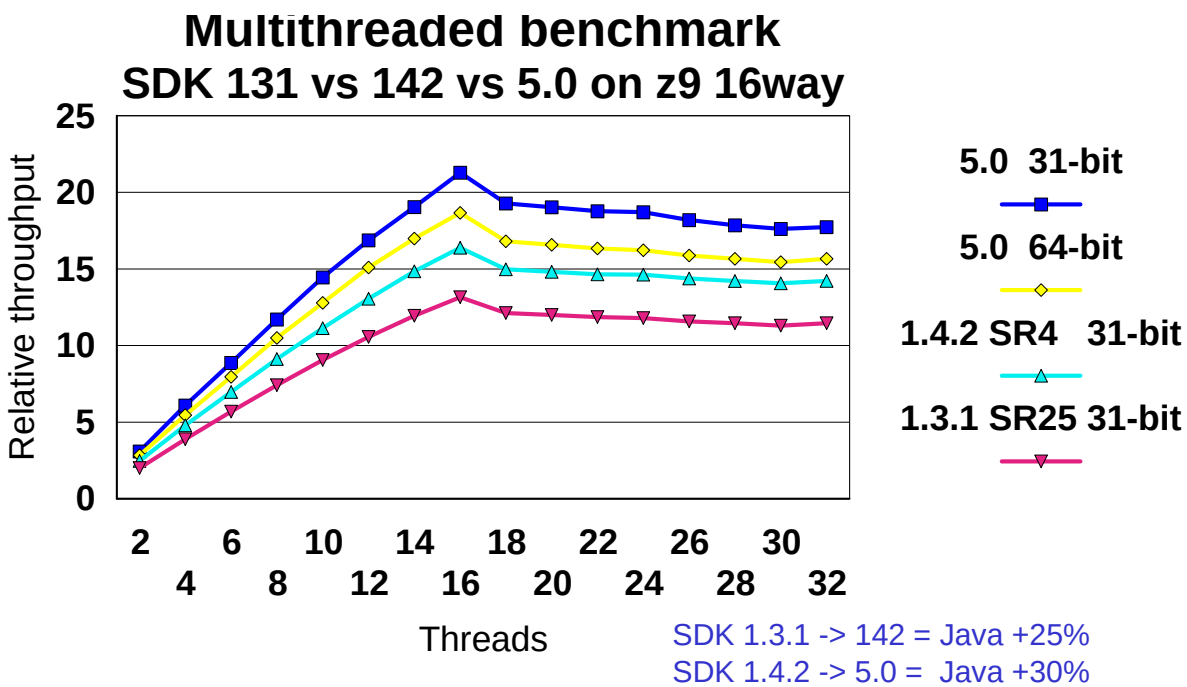
WAS V6.1 & V7 use the new IBM J9 JVM

(aka SDK 5.0, J2SE 5.0, J2RE 1.5.0;
SDK 6.0, or 1.6 for WAS V7)

Provide better Performance, Scalability, and Availability

- **Garbage collector enhancements**
  - – Incorporates for the first time generational garbage collection
- **Superior JIT (Just in time) compiler**
  - – Multiple optimization methods from application profiling to more intelligent and better code optimization algorithms
- **Asynchronous compilation**
  - – Compilation of Java methods proceeds on a background thread
    - Other application threads do not have to wait to execute the method
  - – Improves startup time of heavily multithreaded applications on SMPs
- **Compile-time optimizations to remove contention**
  - – escape analysis, lock coarsening, …
- **Fine-grained locking of VM data structures**

z/OS Java 5 SDK Performance



SDK 1.3.1 -> 142 = Java +25%
SDK 1.4.2 -> 5.0 = Java +30%
SDK 5.0 31-bit -> SDK 64-bit Java -12%
(but memory constraints relieved)

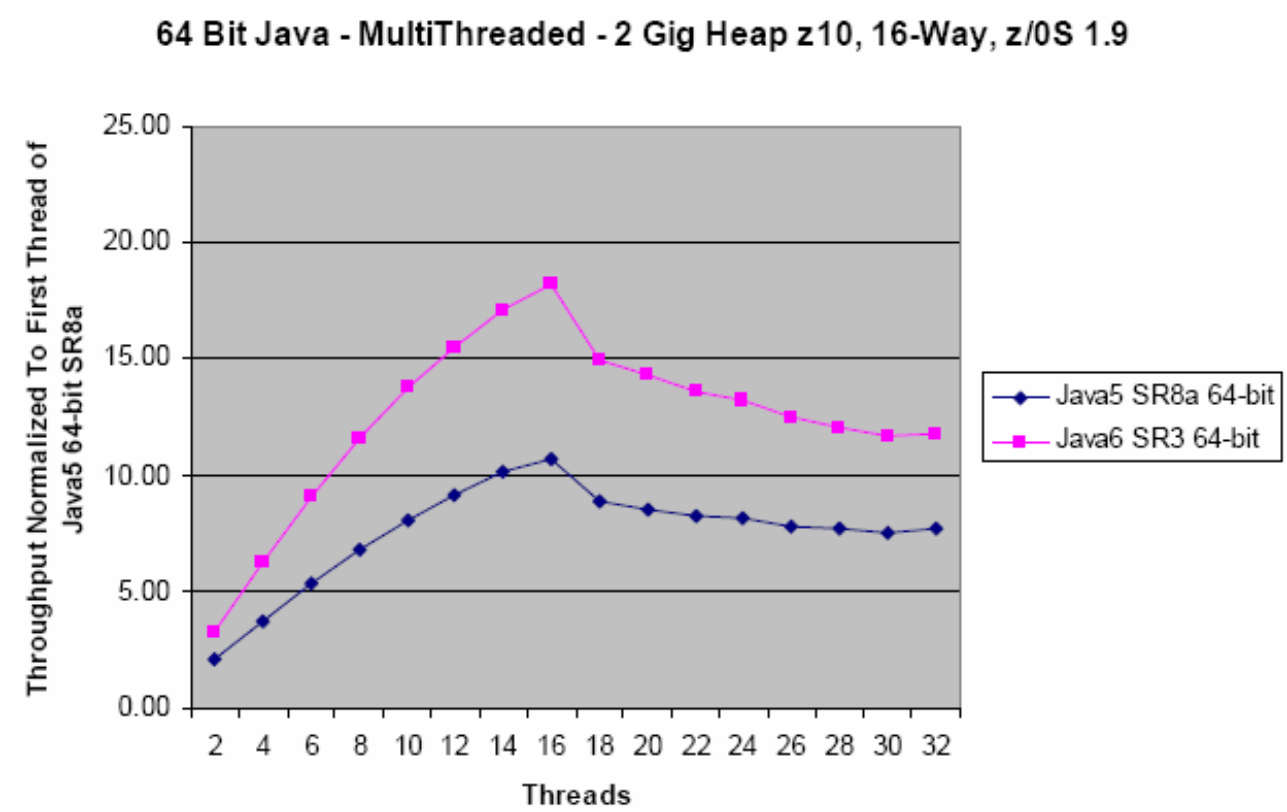
SDK6 Performance

What's new

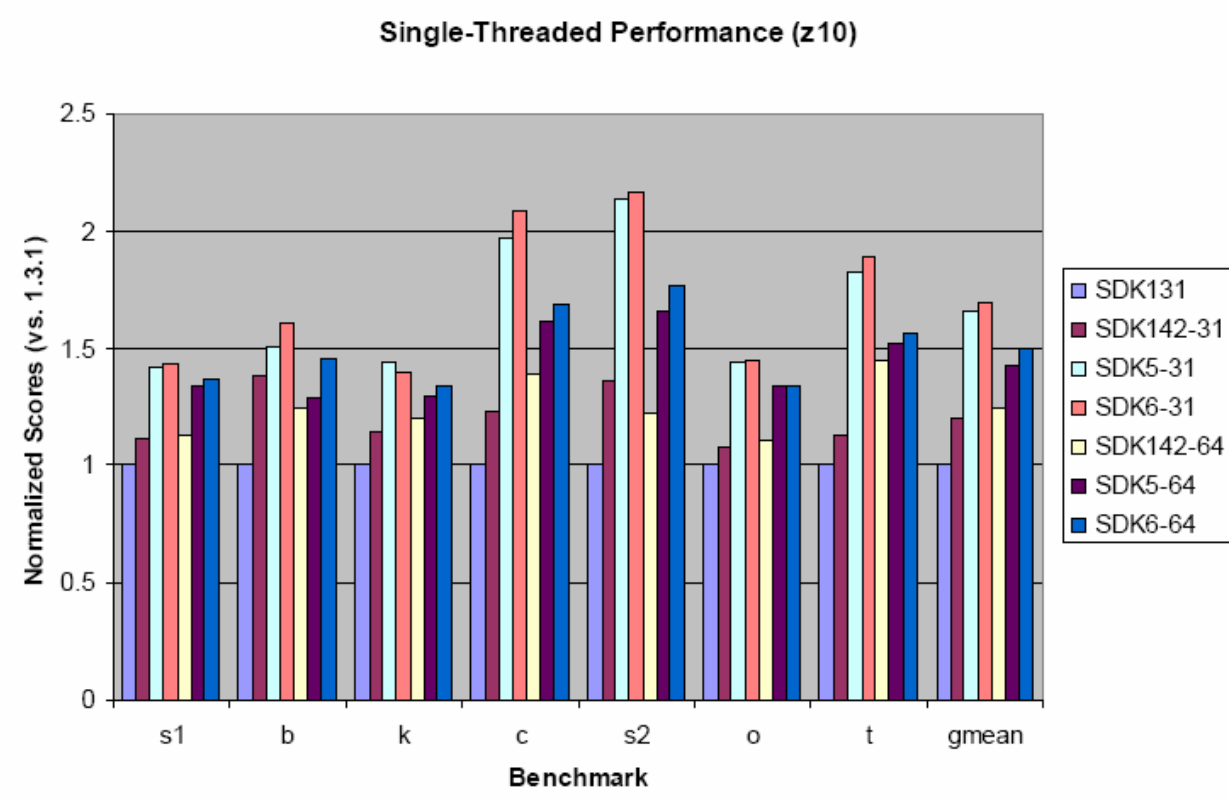
- Exploits z10 ISA features
- Multi-threaded performance improvements
  - Garbage collection improvements
  - Class library work
  - JIT improvements
- 64-bit SDK performance improvements in Java6 SR3
  - Compressed References (-Xcompressrefs)
- XML performance improvements
- Ahead-of-time JIT support for shared-classes

http://www.ibm.com/developerworks/java/jdk

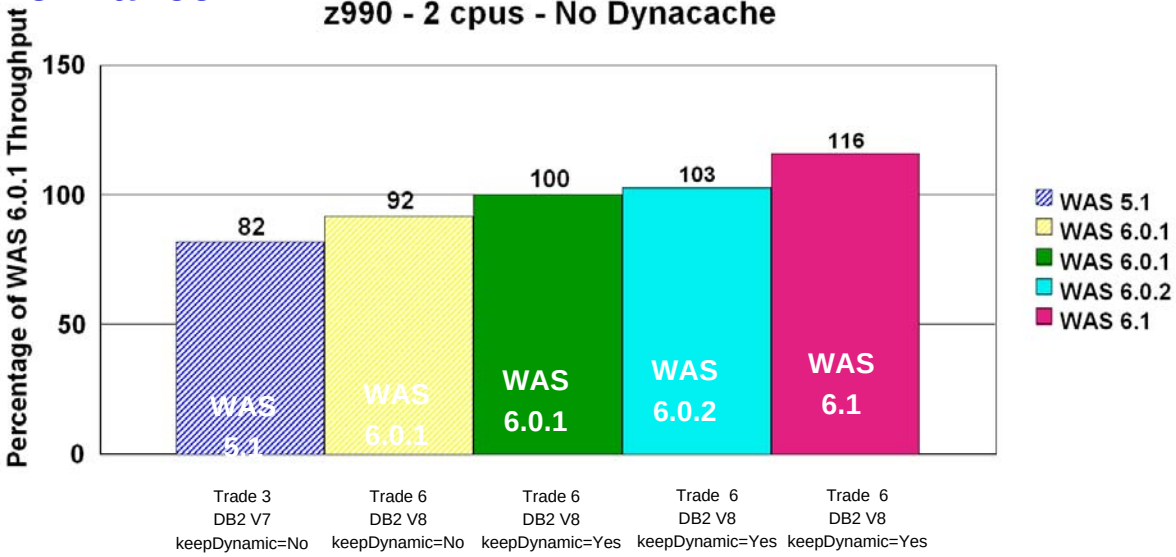
SDK Multi-Threaded Benchmark



SDK6 – Performance – Single Threaded Benchmarks



WAS Trade Performance

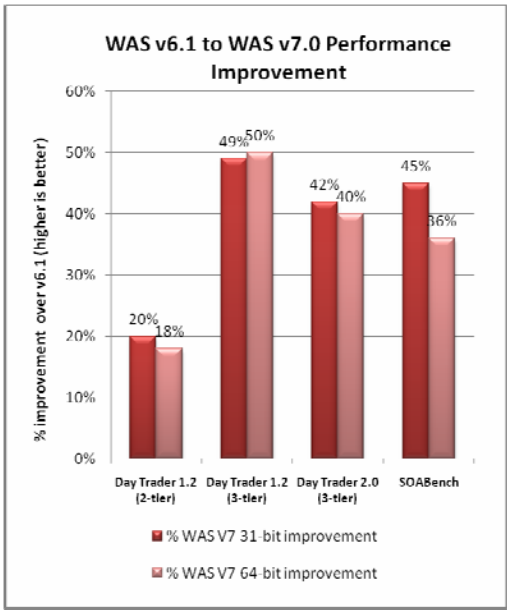


When the dynamic statement cache is active, and an application is run that is bound with **KEEPDYNAMIC(YES)**, DB2 retains a copy of both the prepared statement and the statement string. The prepared statement is cached locally for the application process.

12% Performance Improvement from WAS 5.1 to WAS 6.0.1
+9% from keepDynamic=YES (tuning)
3% Performance Improvement from WAS 6.0.1 to WAS 6.0.2
13% Performance Improvement from WAS 6.0.2 to WAS 6.1

"WebSphere Performance on z/OS" by Bob St. John, IBM at SHARE session 2567, February, 2007
http://ew.share.org/client_files/callpapers/attach/SHARE_in_Tampa_Bay/S2567BS073104.pdf.

WebSphere Application Server on z/OS V 7 Performance Improvements:



DayTrader 1.2 - WAS v6.1 to v7.0

- zWAS v7.0 performance is up 22% from v6.1 for 2-tier configuration
  - *JDK improvements*
  - *Servant/Controller communication optimizations*
  - *Codepath improvements throughout WAS v7.0*
- zWAS performance up 44% in 3-tier configuration

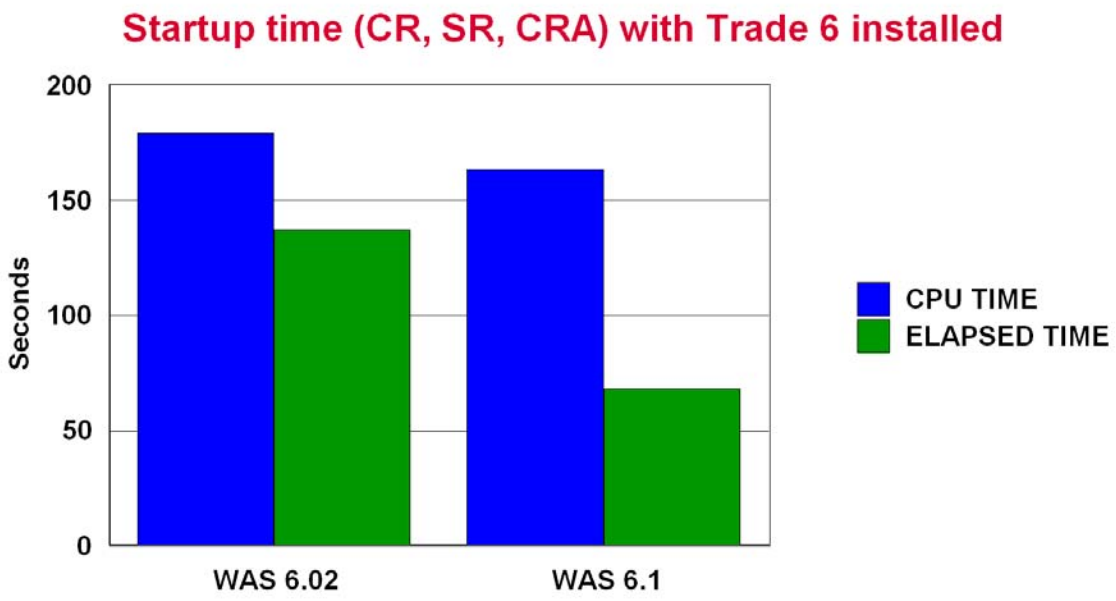
DayTrader 2.0 EJB3 - WAS v6.1 FeP to v7.0

- WAS v7.0 is 65% faster than v6.1 EJB3 FeP

SOABench - WAS v6.1+ WS Feature pack to v7.0

- zWAS v7.0 improved 25-50% for payload sizes ranging from 3kin3kout to 100kin100kout.
- Common payload 10kin10kout improved 45%

WAS Startup time



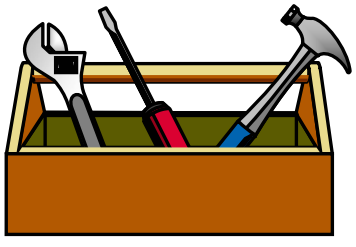
9% reduction in startup CPU time with WAS 6.1;
Add'l 18% reduction with WAS 7

50% reduction in startup elapsed time with WAS 6.1
Add'l 3% reduction with WAS 7

System Tuning

- z/OS or OS/390®
- Workload Manager
- UNIX System Services & HFS
- TCP/IP
- Language Environment (LE)
- System Logger & RRS
- Security & RACF®
- Java
- SMF
- GRS
- Library Search Order
- Other . . .
- Tracing & Logging - minimize as much as possible.

- Performance Tuning guidance in the WebSphere Application Server "InfoCenter"
  - Performance and Troubleshooting sections
  - "Performance Tuning and Monitoring" PDF



Tune for effective use of storage:

- **Need large servant regions** (set REGION=0M on proc)
  - **Biggest single affect on storage use**
    - Default SR heap (512 Mb requires ~700 meg)
    - Also affects GC time (server delays)
  - **Tune your Java heap size (often the biggest performance leverage item)**
    - See "Tuning the JVM Heap" (later)
  - **May have to tune # of Servant Regions and Threads**

- **Define more auxilary storage (Page packs)**
  - **Test systems with 1Gb may work with good paging resources**

- **"64-bit" Addressing available if needed**
  - **SDK in /<app_server_root>/java64/ (symlinks to /shared/zWebSphere/V6R1/java64/**
  - **Enable desired server(s) through AdminConsole (or WSADMIN.)**
  - **All regions in the server are updated – control, servant, adjunct.**
  - **Slight performance degradation, unless you NEED the extra Heap Size.**
  - **See WP100920 & WP101121 white papers on ibm.com/support/techdocs**

UNIX System Services & HFS Tuning

- **Make sure you allow enough sockets, etc.**
  - BPXPRMxx parms – some limits increased with z/OS 1.7
    - MAXFILEPROC (Impacts OMVS kernel storage, only set as high as needed)
      - Applies to all USS user processes (or set at user level using RACF)
    - MAXSOCKETS (At least as high as MAXFILEPROC - No Impact on OMVS kernel storage)
- **HFS (Hierarchical File System)**
  - **Product HFS** (/usr/lpp/WebSphere/...) - Mount Read/Only
  - **Configuration HFSes**
    - Separate HFS for each node - **make sure it is owned by the right system** (if sharable)
    - Can be shared for testing, sharable for fail-over
  - **File Caching:** Use SMF 92 records for tuning
- **zFS -** Supported by WAS V6.1 customization
  - Should improve performance when writing to a shared file system (not recom'd.)

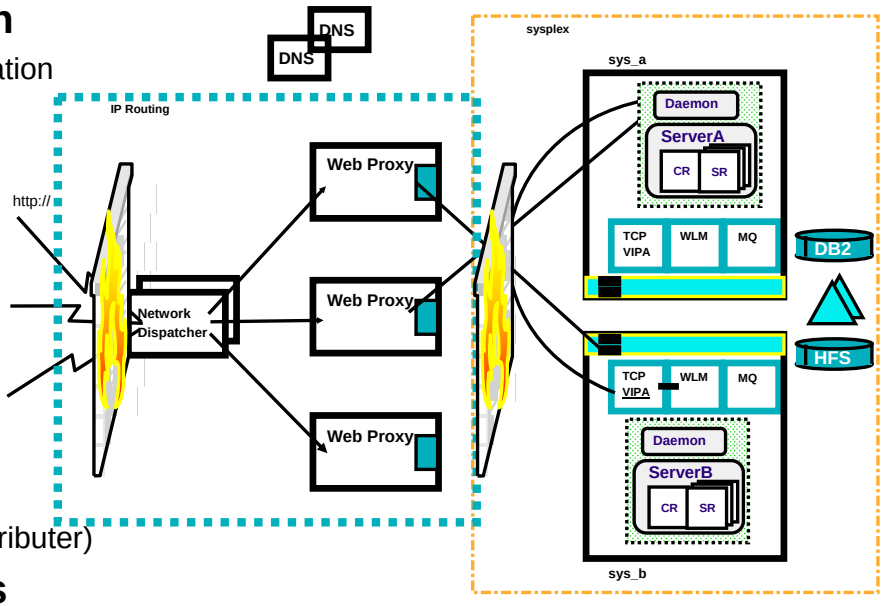
- **log4j recommendations**
  - Write log4j logstream to unshared zFS,
  - Write simple trace strings, Write without flush if possible,
  - Test before writing, and Write as seldom as possible.

Security is not "free" but can be tuned . . .

- **WebSphere runs with security off by default until V.6.1.**
- **SAF classes can be enabled or disabled to control security**
  - Disabled SAF classes: negligible overhead
  - Enabled SAF classes: number of profiles in class will affect performance
- **EJBROLE Class**
  - More EJBROLEs on a method will give you more access checks
  - Use GEJBROLEs to reduce the number of Profiles
- **Keep RACF classes and other info in memory**
  - RACLIST CBIND, EJBROLE, FACILITY, PTKDATA, SERVER, STARTED
  - Use VLF for ACEEs, GTS, and UID/GIDs
- **Disable SAF calls for successful HFS accesses**
  - Define the BPX.SAFFASTPATH facility class, or use the IRRSXT00 exit
- **Performance depends on your Repository Mechanism:**
  - 'Custom' *can* be better than RACF, which is better than LDAP
- **SSL security authentication**
  - Use IBM™ zSeries™ hardware assists to improve performance on z/OS
  - Reduce excessive SSL hand shakes for subsequent transactions.

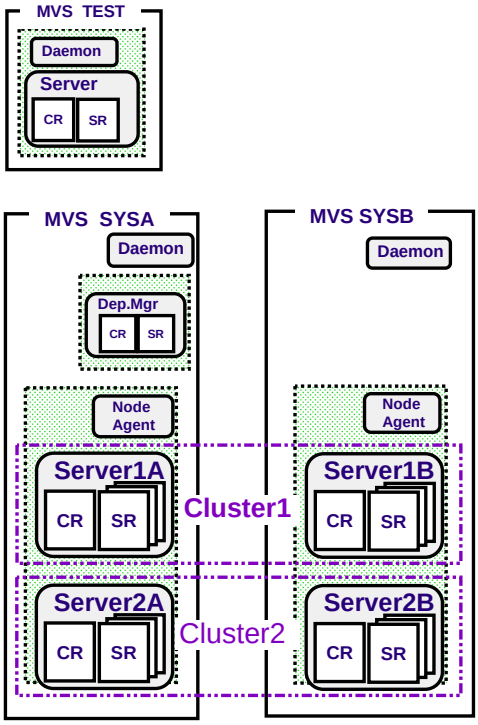
Server Topology Decisions

- **Client location**
  - Remote vs. Local
- **Server(s) location**
  - Number & Configuration
- **Web Tier**
  - Cache static objects
  - Firewalls for DMZ
  - Authentication
- **HTTP servers**
  - HTTP vs. IIOP
  - Reverse Proxies
  - Use HTTP transport
- **DNS**
- **D-VIPA** (Sysplex Distributer)
- **Database servers**



WAS Configuration Options

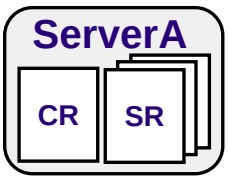
- **Base Application Server**
  - Easy to set up & useful for testing
  - Responsive to server & application changes
  - Not suited for production
    - no clustering, single-systems config.
- **Network Deployment (ND)**
  - Managed by Deployment Manager & Node Agents
  - Multiple Application Servers
    - Group multiple AppServers into Clusters
  - **ND Required for:**
    - Multi-systems configuration & Clustering
    - Horizontal scaling for increased throughput
    - Continuous availability & fail-over
    - Rolling upgrades for continuous operations



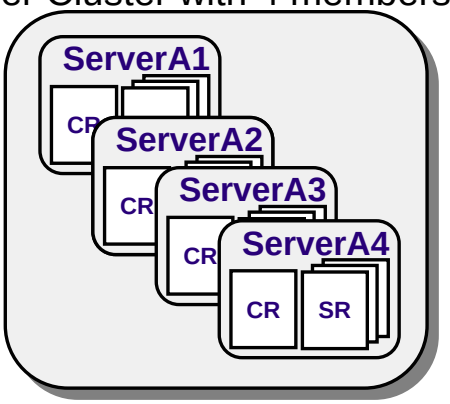
WAS for z/OS clustering:

- **Inner cluster - Server Instance**
  - Controller region - communication endpoint (HTTP, IIOP, MDB)
    - Performs work classification, security processing, queues to WLM
  - Servant region(s) - 1 or more address spaces (WLM managed)
    - JVM - Web & EJB container - where applications run
  - Isolated for availability & performance
  - Have identical runtime settings
  - Confined to a single z/OS system

Application Server:



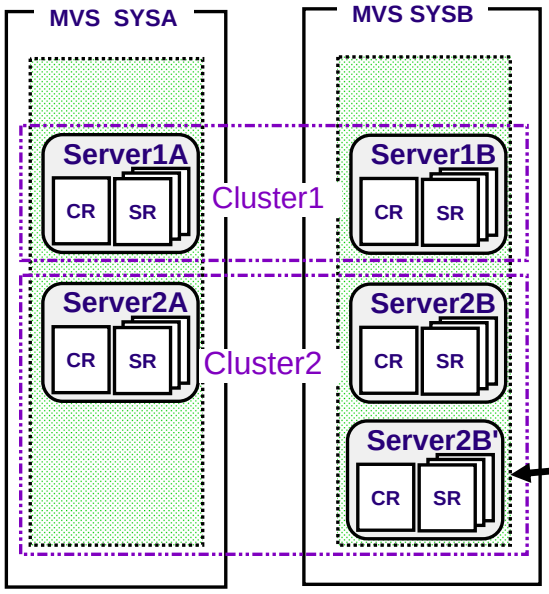
Server Cluster with 4 members:



- **Outer Cluster - Generic Server**
  - 1 or more server instances of a server.
  - All servers have the same applications
  - May have different runtime settings
  - May exist on multiple z/OS systems.
- **Cell consists of one or more clusters.**
  - Confined to one Parallel Sysplex

Clustered Servers

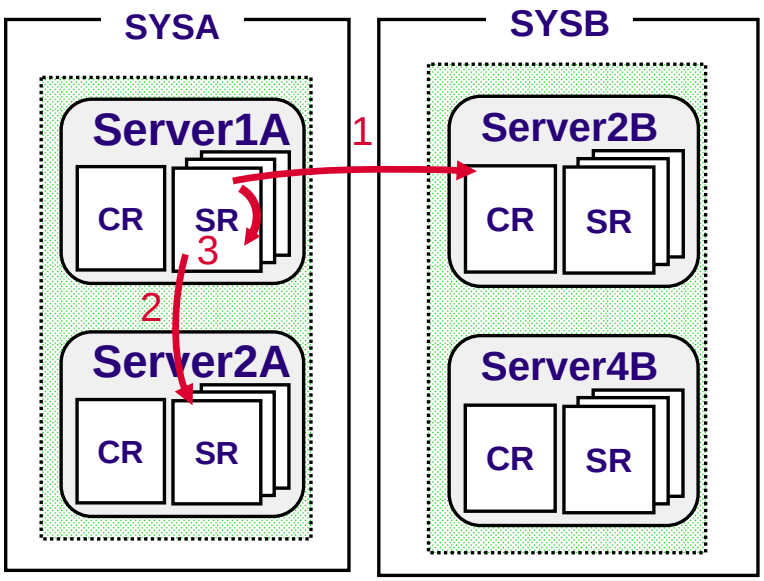
- **Horizontal (cross-LPAR) vs. Vertical (Same LPAR)**



- **Multiple instances of the same application server:**
  - Increases Availability:
    - Remove single point of failure
    - Allows rolling updates.
  - Can improve performance
    - On multiple systems (horizontal scaling)
- **However:**
  - Multiple instances on the same system normally won't improve performance.

Optimize application object flows

Deploy related applications in the same server:



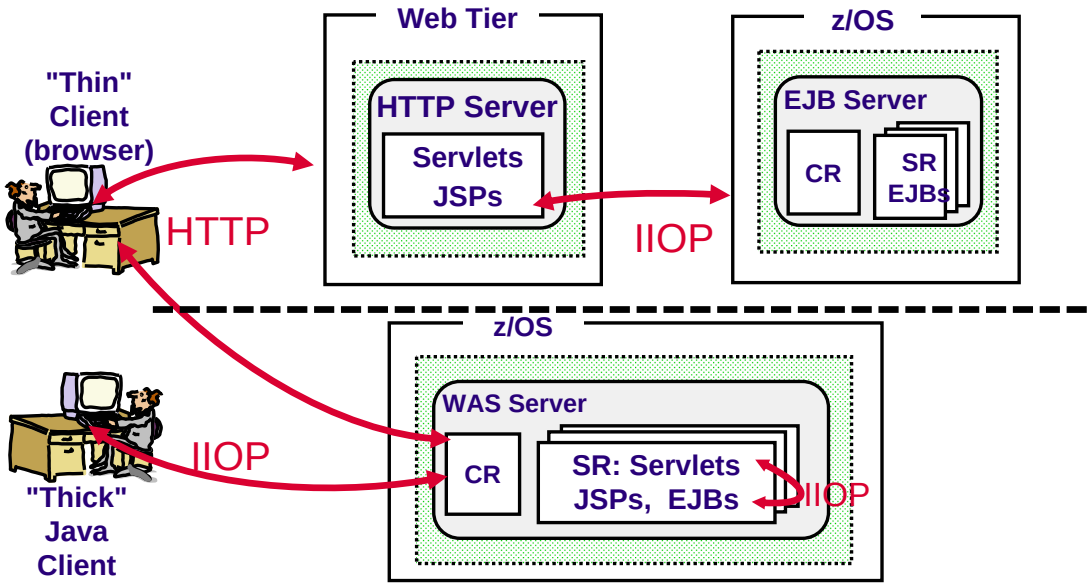
1 **Avoid application calls from one system to another**

2 **Provide a local replica of any required application server.**

3 **Deploy applications in the same server, because local calls are even faster.**

➔Use 'Pass by reference' (default ORB setting: "noLocalCopies")
– must be in same EAR file or use Server Class Loader

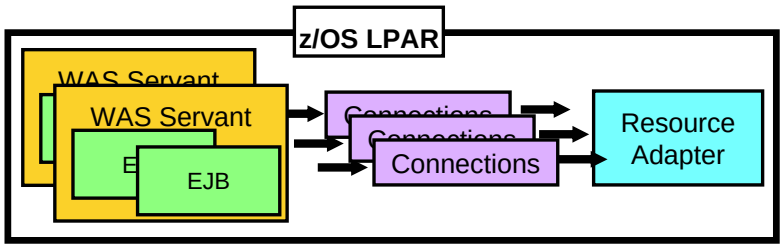
Reduce unnecessary IIOP Flows



- Avoid IIOP calls from one system to another - serialization/deserialization overhead can be excessive!

Connector Performance

(over-simplified)



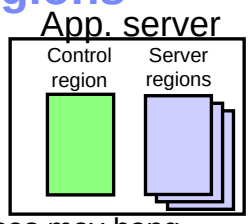
General:

- **Use Local Connections** over Remote
  - Avoids Network Delays
  - Requires less CPU resources
- **Use Pooled Connectors -** queuing model between connectors & resource adapters

- **DB2:** JDBC Type 2 vs. Type 4
  - Static SQLJ out-performs dynamic SQL
    - Can benefit greatly from dynamic statement caching in the database engine.
- **CICS:** Use TransGateway which uses EXCI
  - Monitor/Manage the number of Pipes, and Threads (& Servant Regions)
- **IMS:** Use Local Connect Option vs. MSC vs. Remote IMS Connect
- **MQ:** Use Binding Mode vs. Client mode
- **Optimized Local Adapters**

See "***WebSphere for z/OS Connectivity Architectural Choices***" SG24-6365

Replication - Managing the # of Servant Regions

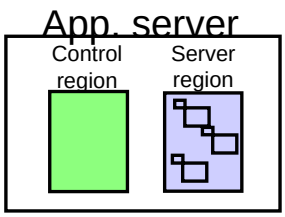


- **Adminconsole: Appl. Server >> "Server Instances"**
  - **Check "Multiple Instances Enabled"**
    - Otherwise, WLM will only start 1 servant region for this appserver
    - If checked, and Min/Max = 1, transactions from different service classes may hang.
  - **"Minimum number of Instances"**
    - Useful for avoiding delays to start up server regions
    - To keep work from coming in thru the protocol handler before SRs are ready, use protocol_accept_http_work_after_min_srs=nn
  - **"Maximum number of Instances"**
    - Useful for limiting excessive server regions during server instance ramp-up or if you have limited real storage . . .
  - **Caution!**
    - If you specify a maximum number of instances, WLM is restricted from starting more than this number of servant regions for this server instance.
      - The Maximum number must be >= number of service classes used by this application's transactions, or transactions will time out.
      - Account for default CB service class and enclaves that originate outside WebSphere.

Managing the number of Threads in the JVM

- **Workload Profile in ISC (adminconsole):**
  - AppServer > ORB Service > Advanced Settings > "Workload Profile"
    - ISOLATE (1 thread)
    - NORMAL (3 threads)
    - CPUBOUND (# of CPs-1, minimum of 3)
    - IOBOUND (Number of CPs*3, Min=5, Max=30)
    - LONGWAIT (40)
    - CUSTOM (V7): Set with servant_region_custom_thread_count, Min 1, Max 100

    See message BBOO0234I in the controller job log to check the number.



  - Allow for increased concurrency
  - WebSphere for z/OS doesn't need threads as placeholders for work
    - WLM queues are used for that
  - Plan for # of in and ready threads to be 2-3X the # of CPs
  - Experiment with # of threads, # of servants to optimize performance.
    - Too many servant regions take excessive storage
    - Too many threads in a JVM creates interference & more frequent GC.
    - Display # of threads: SDSF PS Panel, or MVS commands: D OMVS,PID= or Modify (F) <server>,DISPLAY,THREADS, then compare to JAVACORE dump

Classifying Work with WLM

- **Started Tasks**
- **OMVS work**
- **Transactions - changes with WAS 5.1**
  - 'CB' work
  - HTTP by URL
  - MDBs
  - IIOP
- **Resource managers:**
  - DB2
  - CICS
  - IMS
  - MQ
  - other
  - Network QoS

```
          WLM Subsystem Type Selection List for Rules
                                                          Default
Action  Type      Description                             Service
  __    CB        CB Class'n w/WLM Trans. CLASSes         CBCLASS
  __    CICS      Use Modify to enter YOUR rules
  __    DB2       Use Modify to enter YOUR rules
  __    DDF       Use Modify to enter YOUR rules          DB_DDF
  __    IMS       Use Modify to enter YOUR rules
  __    IWEB      IWEB rules                              IWEBFAST
  __    JES       Batch Classification Rule               BAT_MED
  __    OMVS      E_Biz Classification Rule               EBIZ_DEF
  __    STC       Started Task Classification Rule        OPS_DEF
  __    TSO       TSO Classification Rule                 TSO_DEF
```


WLM / STC - Classifying WebSphere Address Spaces

- **Controller Regions** (Daemon, Node Agent, Deployment Manager, App. Servers)
  - Classify as High Importance & High Velocity
- **Servant & Adjunct Regions**
  - Classify with velocity goal, high enough to get started quickly, lower than controllers
    - Work is actually classified under the application environment
- **Sample STC Classification Rules:**
  - Use Unique Report Classes to track important Started Tasks:

#	Qualifier type	Qualifier name	Starting position	Service Class	Report Class
1	TN	WSDM*		OPS_HI	RWSDMGR
1	TN	WSSR%%%		OPS_HI	RWSAPCR
1	TN	WSSR%%%S		OPS_MED	RWSAPSR
1	TN	WSSR%%%C		OPS_MED	RWSAPCRA

  - OPS_HI service class: Importance= 1, Velocity = 70
  - OPS_MED service class: Importance= 2, Velocity = 40

WLM / OMVS - Controller start-up Procedure

- **ApplyPTF step added to Contol Region Proc:**
  - applyPTF.sh checks to see if service has been applied to WebSphere and "update files" for the new service.
  - Classify server controller jobnames with WLM OMVS rules.

- **OMVS Classification rules - see the WAS InfoCenter for details**

```
   --------Qualifier--------                  -------Class--------
   Type       Name     Start                   Service     Report
                                   DEFAULTS:  EBIZ_DEF    ________
1  TN         T6*      ___                     EBIZ_HI     RPTACR
1  TN         WS*      ___                     EBIZ_HI     RPTACR
```


WLM/CB - Classifying WebSphere Transactions

- **Subsystem type = CB using the following criteria:**
  - **Generic Server name (CN)** - cluster transition name = the applenv name
  - **Server Instance name (SI)** - not useful because instances share work
  - **Userid assigned to the transaction (UI)** - usually not useful
  - **Transaction class (TC)** - assigned by "Workload Classification" xml document.
- **Percentage response time goal is recommended**
  - Example: 80% of trans less than 0.5 seconds (or high velocity default service class)
  - Response time goals better than Velocity goals in a true production environment.
    - Velocity goals need to be re-calibrated with environmental changes (CPU, workload)
  - Multi-period Goals may be used, but are not recommended.
  - Default is SYSOTHER (discretionary)
- **Other considerations:**
  - Requests that already have enclave tokens, run under these enclaves, and with the service class assigned for this enclave .
  - Control region maintains "internal queues" based on the service class:
    - A server region may switch queues if needed to, based on demand

Workload classification file

- **Set transaction class (TC) of inbound work - .xml file**
  - HTTP - host, port, URI
  - IIOP - application, module, component, and method name
  - MDB - message listener port, selector attribute
- **Adminconsole:** Environment >> Manage WebSphere Variables

  `wlm_classification_file = <path>/MDBClassMap.xml`
  - **See InfoCenter for details**

```
<?xml version="1.0" encoding="UTF-8"?>

  <InboundClassification type="iiop"
                         schema_version="1.0"
                         default_transaction_class="TCLASS1">
     <iiop_classification_info transaction_class="TCLASS2"
```

`CB Classification Rules:`

```
  Qualifier  Qualifier  Start          Service  Report
# type       name       position       Class    Class
- ---------  --------   --------       -------- --------
1 CN         WSPROD     1              CBMED    RWSPROD
2 . TC       . TCLASS1                 CBFAST   RWSPRD1
2 . TC       . TCLASS2                 CBMED    RWSPRD2
```

Examples:

a **www.ibm.com**:80/Webap1/myservlet => TCLASS1 => CBFAST (RWSPRD1)

b **www.ibm.com**:443/Webap1/myservlet => TCLASS2 => CBMED (RWSPRD2)

How is my Classification File working?

• Displaying Classification of Work Requests

- **See if classification scheme is classifying work as expected:**
  - MVS oper command: F <server>,DISPLAY,WORK,CLINFO

```
F H5SR01D,DISPLAY,WORK,CLINFO
BBOO0281I CLASSIFICATION COUNTERS FOR HTTP WORK
BBOO0282I CHECKED 27976, MATCHED 27976, USED 816, COST 4, DESC: HTTP Default
BBOO0282I CHECKED 27976, MATCHED 9053, USED 9053, COST 2, DESC: H5Servlets
BBOO0282I CHECKED 18923, MATCHED 9021, USED 9021, COST 3, DESC: H5EJBs
BBOO0282I CHECKED 9902, MATCHED 9086, USED 9086, COST 4, DESC: H5JSPs
BBOO0283I FOR HTTP WORK: TOTAL CLASSIFIED 27976, WEIGHTED TOTAL COST 84777
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,WORK,CLINFO
```

- **CHECKED** - Number of times the rule has been examined.
- **MATCHED** - Number of this times that this rule has been matched by the request.
- **USED** - Number of times that this rule has actually been used.
- **COST** - Number of compares that required to determine if this is the correct rule to use.
- **WEIGHTED COST** - Number of times each rule was used multiplied by the cost, or number of rule compares that were done, and adding up across all rules.
- Reduce the cost by re-arranging your Classification File.

WLM Classification Guidelines

- **Service Classes used to meet goals**
  - Do not let work default to discretionary goals.
  - Set realistic (achievable) goals.
  - Assumes there is displaceable work when resources constrained.
  - Use Velocity for Address Spaces, Response Time for enclaves.
  - Avoid multi-period service classes for disparate work in the same server.
  - Avoid too many Service Classes.
  - Understand difference between Business Trans & RMF Trans.
    - WID Quality of Service: Activity Properties can change Scope of Transaction and RMF numbers. (New Tran, Participate, Commit Before/After Tran …)
    - Changes Ratio of Business Tran to RMF Trans
- **Report Classes distinguish among items of interest**
  - Do not lump components together.
  - Use RMF RCPER(rc*) to show Resp. time distribution, Delay break-out, etc.

WLM key to Configuring for Availability

Application availability based on sysplex availability principles.

- **First Principle - "One" is a lonely number**
- **"Two" entities with failure isolation (Three are better!)**

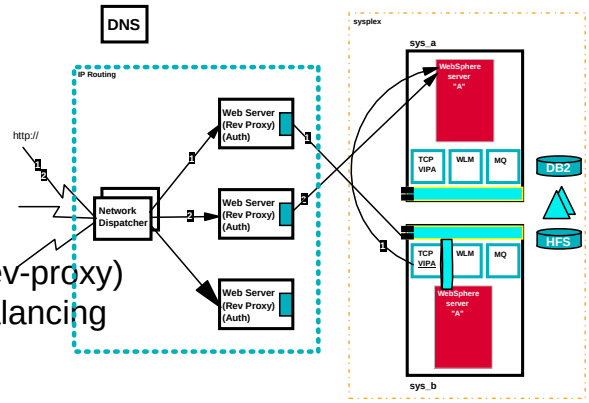
Application availability is dependent upon:

- **Sysplex components - SYSPLEX distributor, data sharing, etc.**
- **Non-sysplex components - Edge servers, DNSs, routers, etc.**
- **Configuration changes & Operational procedures - Service upgrades, Backups, etc.**

Ensure clients can always get to the server

Intelligent Routing:

- WLM-aware vs. Round-Robin
- Session Affinity within Server Instance and across Server Instances (Systems)
- Network Dispatcher (MNLB) load balances
- IHS or Web Server with WAS AE plug-in (rev-proxy)
- Sysplex Distributor good for TCP/IP load balancing



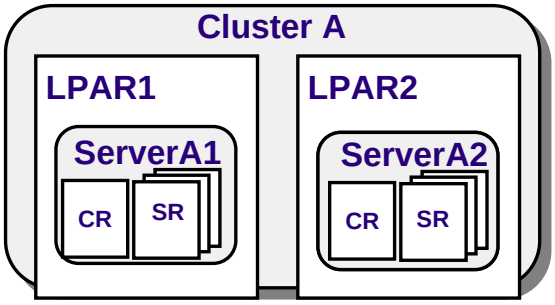
Workload Management & Availability

Multiple Server Instances provide Continuous Operation

- Cluster Horizontally across multiple LPARs (& Hardware Engines)
- Allows for Planned and Un-Planned Outages

Multiple Servants also allow for Continuity

- WLM will restart a Servant Region if one fails, or if killed by an operator (Cancel cmd, or SDSF 'K' action char.)
- Server Instances (Controller Regions) can be re-started by ARM (Automatic Restart Manager) or your System Automation Product
- Insulates from Garbage Collection interruptions.
- More Servants vs. More Threads
  (depends on many variables)



Distributing HTTP Requests on multiple Servants

- **WAS uses a "hot server" strategy to route HTTP requests**
  - Route to servant regions which had recently dispatched work with threads available.
  - "hot servers" have pages in memory, application methods and cache full of data.
  - HTTP requests with session affinity are routed to the servant region where the session object(s) reside.
- **However, this can cause imbalances in some situations:**
  - "Hot" servant regions can get over-loaded with work
  - GC and loss of a servant region can impact many sessions.
- **Distribute HTTP requests evenly across servant regions:**
  - Specify Adminconsole setting:
    - Servers > Applications servers > server_name > Server Infrastructure > Administration > Administration Services > Additional Properties > Custom Properties
    - Change 'WLMStatefulSession' to 'true'
  - Optimize the minimum and maximum number of servant regions.
    - May want to eliminate transaction class mapping.
  - Minimize the number of different service classes for these servers.

Java Tuning

- **Java level is reported in servant region joblog**
  - 5.0 SDK: JVM Build is J2RE 1.5.0 IBM J9 2.3 z/OS s390-31 j9vmmz3123ifx-20090225 (**JIT enabled**)
  - 6.0 SDK: JVM Build is J2RE 1.6.0 IBM J9 2.4 z/OS s390x-64 jvmmz6460-20081107_25433 (**JIT enabled, AOT enabled**)
  - Also indicates if the Just-in-Time and Ahead-of-Time Compilers are enabled.
- **Make sure the JIT is enabled**
  - Number of references (or loop iterations) before keeping JITed code in LE Heap:
    - 1.4.2 SDK: IBM_MIXED_MODE_THRESHOLD=nnn (default = 800-1107)
    - 5.0 SDK: IBM_JAVA_OPTIONS=-Xjit:count=<value> (defaults to progressive optimizations)
    - Recommendation: only change this if needed for benchmarking. Use default for production.
- **Turn off JRAS debugging support**
  - Turn off in adminconsole - Set *=all=disable
    - Note: you may be tracing and not know it if ras_trace_outputLocation=BUFFER
    - Verify by looking in SYSOUT dataset for trace setting
- **Other JVM Performance Options**
  - Most default values provide best performance.
- **Other tips: www.ibm.com/servers/eserver/zseries/software/java/**

5.0 JVM Heap & GC Tuning

- **Must be tailored to your Application & Workload**
  - Typically get 80% of maximum performance with 20% of the work by making good choices on a few key settings.
  - To get the best performance, you must know your applications memory allocation and runtime needs.
- **2 iterative tuning steps over a testing cycle:**
  - Step 1: Heap Size tuning
  - Step 2: GC Runtime Policy optimization
- **Key setting for the JVM: Heap Size (-Xms / -Xmx)**
  - Set min & max to values within your physical memory limitation,
  - Keep a large interval between GC's, and a low duration:
    - Typical low end bound on frequency of GC's is 10 sec
    - Typical high end bound on duration of GC's is 1-2 sec (GC should account for less than 2% of the time)
    - May also have to increase the number of Servant regions.

Runtime GC Policy settings for the J9 JVM

J9 Memory management has 4 configurable policies:

- **Optimize for Throughput** – flat heap collector focused on maximum throughput
  "I want my application to run to completion as quickly as possible."
  -Xgcpolicy:optthruput **(default)**
- **Optimize for Pause Time** – flat heap collector w/ concurrent mark & sweep to minimize GC pause time
  "My application requires good response time to unpredictable events."
  -Xgcpolicy:optavgpause
- **Generational Concurrent** – divides heap into "nursery" & "tenured" segments - fast collection for short lived objects. Max. throughput w/ minimal pause time
  "My application has a high allocation and death rate."
  -Xgcpolicy:gencon
- **Subpool** – flat heap technique to increase performance on MP systems, (> 8) Available on IBM pSeries™ and zSeries™
  "My application is running on big iron & high allocation rates on many threads."
  -Xgcpolicy:subpool

Notes on Fragmentation:

- Most Java Objects in the heap are moveable (not tied to a single space in memory)
- "Pinned objects" cannot be moved (permanently or temporarily.)
- J9 helps prevent fragmentation by moving pinned objects during compaction.

Tuning your Java heap: Collect verboseGC stats

```
<af type="nursery" id="35" timestamp="Thu Aug 11 21:47:11 2005" intervalms="10730.361">
   . . .
   . . .
   </gc>
     <tenured freebytes="189664320" totalbytes="268435456" percent="70" >
       <soa freebytes="187251000" totalbytes="265751552" percent="70" />
       <loa freebytes="2413320" totalbytes="2683904" percent="89" />
     </tenured>
     <time totalms="224.006" />      <time totalms="377.634" />
</af
```

- Adminconsole: Server >> Process >> Servant >> JVM >> check "GC Verbose"
- Results appear in server region's //SYSOUT DD file (or pipe to HFS file)
  - Don't specify JVM LOGFILE or output from multiple SRs will be meaningless.
- Key value: percent free storage after each GC in each area.
  - Run for a long time to make sure your application does not have a memory leak.
  - Steady state, this is your base Java heap requirement
    - Normally it's best to set JVM_MINHEAPSIZE=JVM_HEAPSIZE
- Key value: % of elapsed time spent in GC
  - "totalms = "**XXX"** (GC time) / intervalms ="**YYYY"** (time since last GC) **< 2%**
- Visualizers
  - See the ISA, APMT or GC Diagnostic tool on www.alphaworks.ibm.com/tech/

Profiling & Monitoring Tools for 5.0 SDK

- **New Features & Interfaces included in JVM:**
  - Monitoring Tool Interface (JVMTI) – replaces JVMDI for Profiling (JVMPI)
- **Garbage Collection - verbose "visualizers"**
  - **(Need to clean out extraneous messages from SYSOUT.)**
  - **ISA –** IBM Support Assistant
  - **EVTK** – IBM Solution Center
  - **PMAT** – IBM alphaWorks
- **Application Profilers**
  - Eclipse **TPTP** - Open source "Test and Performance Tools Platform" - Profiler from http://www.eclipse.org/tptp
  - **Jprobe** (Quest Software)
  - **Jprofiler** (ej-technologies)

Specialty Engines: zIIPs & zAAPs

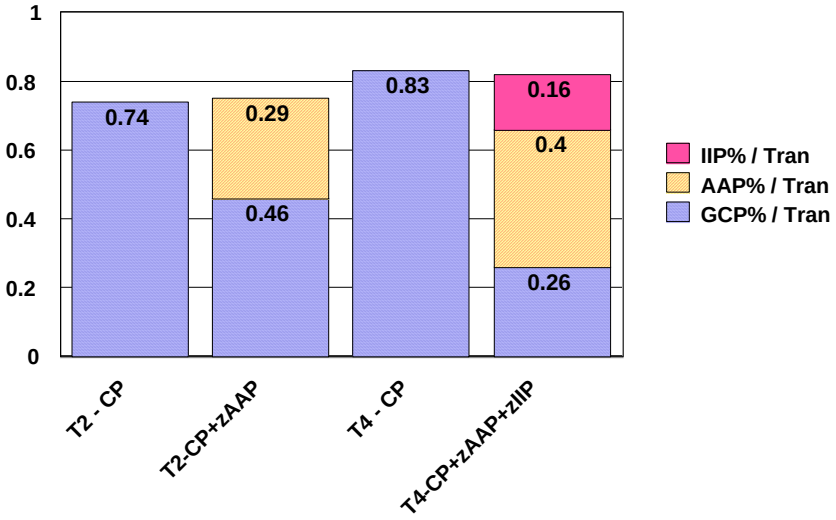
- **zIIP: zSeries Integrated Information Processor**
  - System z10 & z9 + z/OS 1.6 + DB2 for z/OS V8 + FMIDs JBB77S9(1.6) or JBB772S(1.8)
- **zAAP: zSeries Application Assist Processor**
  - System z + z/OS 1.6 + IBM SDK for Java 1.4 + PTF for APAR PQ86689
- **Not a performance boost** (except systems w/ sub-capacity GCPs)
  - Maybe helpful in reducing General Purpose CPs and associated License fees
- **Sub-capacity Processors receive extra benefits**

  (z10 BC, and z10EC*, z9 BC, and z9 EC)
  - Specialty engines run at full speed – may provide performance boost.
- **Estimation of Usage:**
  - z/OS Use RMF Workload Activity Report with `IEAOPTxx PROJECTCPU=YES`
- **See Techdocs:** TD103516, TD103460, and FLASH10432

Specialty Engines: zIIPs & zAAPs – which to use?

- **WSC Measurements of Trade 6 application using JDBC drivers:**
  - **Type 2** drivers provide superior performance (resp. time & CPU usage) and use zAAPs.
  - **Type 4** drivers can take advantage of both zIIPs and zAAPs with minimal degradation.

CPU Usage per Trade 6 Transaction (preliminary results) - by Processor Type:



Your results will vary – Very Application-dependent!

Tuning Session Management

- **Good practices for using HTTP Sessions** (InfoCenter)
  - Enable Security integration for securing HTTP sessions (use HTTPS)
  - Release HttpSession objects w/ javax.servlet.http.HttpSession.invalidate() when finished.
  - Avoid trying to save and reuse the HttpSession object outside of each servlet or JSP file.
  - Implement java.io.Serializable interface for new objects to be stored in the HTTP session.
  - The HTTPSession API does not dictate transactional behavior for sessions. (Use EJBs.)
  - Ensure the Java objects you add to a session are in the correct class path.
  - Avoid storing large object graphs in the HttpSession object.
  - Utilize Session Affinity to help achieve higher cache hits in the WebSphere App. Server.
  - Maximize use of session affinity and avoid breaking affinity.
  - Secure all of the pages (not just some) when applying security to servlets or JSP files that use sessions with security integration enabled, .
  - Use manual update and either the sync() method or time-based write in applications that read session data, and update infrequently.
  - Tune HTTP Session Management for memory-to-memory or Database session replication.
  - Use EJB session beans to access EJB entity beans.
  - Exploit connectionFactory caching for J2C connections.

More WLM & WebSphere Options

- **See Appendices**
  - Server Start-up Options
  - WebSphere Routing Options
  - Sysplex Distributor & WLM Routing Options
  - Capping the Resources used by WebSphere
  - WLM Tools
  - MVS Commands & Displays
  - Resources & References

Performance Monitoring & Debugging

- **Set Performance Expectations**
- **CPU resources**
  - Understand where the CPU time is spent & how to measure/account for it
- **Performance Monitors**
  - There are many from IBM and other vendors
- **Performance Problem Determination**
  - Response time delays
  - CPU delays
  - Memory useage

Set performance expectations

- **Request pre-sale capacity sizing estimate from your IBM Rep**
  - Fairly detailed input required
  - Estimate is rough, but getting more accurate
  - zPSG Version 2.3 tool for WAS 6 and WPS 6 available now
- **Use a client emulator program to test your application**
  - Determine your CPU cost per transaction
  - Determine your application environment response time
  - Determine your client response time (in a measurement environment)
- **After your application goes into production**
  - Keep key historical data for the WAS application environment
    - Transaction rate, response time, 90% resp time, appl %
  - Keep key historical data for WAS servant region proc
    - appl %
- **WAS application monitors can help keep historical data and detect problems**

Where is CPU Time Accounted?

- **Controller Region**
  - Communications End-point: Receives IIOP/HTTP/SSL request
  - Security authorization for IIOP requests
  - Classifies & Queues Request to WLM queue
- **Servant Region**
  - Selects work from WLM for a given Service Class
  - Some Java Garbage Collection
  - (plus any application created threads)
- **Enclaves**
  - J2EE Application code executes under an enclave (in JVM)
    - Includes JDBC & JNI calls & most Java Garbage Collection
    - Type 2 drivers - DB2 CPU time charged to the Enclave
    - Type 4 drivers - DB2 CPU time charged to the DDF address space
  - Use SDSF ENClaves panel, or RMF Monitor to display
- **Note: Difference between Reporting & Management Classes**
  - All work is Managed by WLM according to the CB-assigned Service Class
  - CPU time is Reported (Charged) to Enclave only if it is part of the transaction.
    - e.g., Garbage Collection is managed to the CB-assigned Service/Reporting Class, but CPU time charged to the servant region's STC-assigned Service/Reporting Class.

RMF Monitor 1 Workload Activity Report

- **Transactions/second**
  - AVG=MPL=AVG ENC = # of enclaves in the period
  - "Business Tran" may not = "WebSphere Tran"
- **Response times**
  - Actual R.T. ~= Execution R.T. (includes waiting on WLM queue)
  - QUEUED delays
- **CPU & Service Rates**
  - CPU service units, & Service/Sec.
  - APPL% = # of engines (CPs) in service (report) class
  - CPUsec/Tran = TCB sec/ENDED
- **Delays**
  - QMPL means waiting for Servant Region (WLM)

```
REPORT BY: ...  REPORTCLASS=RWSCTLRG  -   CONTROL REGION
    TRANSACTIONS    TRANS.-TIME  SS.TTT    --SERVICE RATES--
    AVG      1.00   ACTUAL            0    ABSRPTN     89615
    MPL      1.00   EXECUTION         0    TRX SERV    89615
    ENDED       0   QUEUED            0    TCB          39.9
    END/S    0.00   R/S AFFINITY      0    SRB           4.7
    AVG ENC  0.00   STD DEV           0    HST           0.0
    REM ENC  0.00                        APPL %       37.2

REPORT BY: ...  REPORTCLASS=RWSSRVRG  -   SERVANT REGIONS
    TRANSACTIONS    TRANS.-TIME  SS.TTT    --SERVICE RATES--
    AVG      2.00   ACTUAL            0    ABSRPTN    122075
    MPL      2.00   EXECUTION         0    TRX SERV   122075
    ENDED       0   QUEUED            0    TCB          11.0
    END/S    0.00   R/S AFFINITY      0    SRB           1.0
    AVG ENC  0.00   STD DEV           0    HST           0.0
    REM ENC  0.00                        APPL %       10.0

REPORT BY: ...  REPORTCLASS=RWSAP1ENC - WAS ENCLAVES (TRANSACTO
    TRANSACTIONS    TRANS.-TIME  SS.TTT    --SERVICE RATES--
    AVG    241.52   ACTUAL          276    ABSRPTN      115
    MPL    241.52   EXECUTION       272    TRX SERV     115
    ENDED  106717   QUEUED            4    TCB        255.5
    EXCTD       0   CONVERSION        0    IIT          0.0
    AV ENC 241.52   STD DEV          66    HST          0.0
    REM ENC  0.00                        APPL %       212.9

             EX    PERF   AVG    --USING%--  -- EXECUTION DELAYS
             VEL   INDX   ADRSP   CPU   I/O  TOTAL   CPU QMPL
    GOAL    40.0%
    ACTUALS 45.3%  .89    13.4   0.1   0.0   36.1  23.5 12.6
```


zIIPs & zAAPs CPU Accounting - RMF

Workload Activity Report: DDF & WAS Transactions:

```
WORKLOAD=DB_WKL     SERVICE CLASS=DDFTRD      PERIOD=1 IMPORTANCE=1

TRANSACTIONS   TRANS-TIME SS.TTT  ---SERVICE----  SERVICE TIMES  ---APPL %---
AVG     13.27  ACTUAL         10  IOC          0  CPU    585.4   CP     50.51
MPL     13.27  EXECUTION      10  CPU      16607K SRB      0.0   AAPCP   0.00
ENDED  790019  QUEUED          0  MSO          0  RCT      0.0   IIPCP   5.72
END/S 1316.70  R/S AFFIN       0  SRB          0  IIT      0.0
#SWAPS      0  INELIGIBLE      0  TOT      16607K HST      0.0   AAP     0.00
EXCTD       0  CONVERSION      0  /SEC     27678  AAP      0.0   IIP    47.05
AVG ENC 13.27  STD DEV        14                 IIP    282.3

 EX   PERF  AVG    ------ USING% ----- --EXEC. DELAYS %---    ---DLY%---
 VEL% INDX ADRSP   CPU  AAP  IIP  I/O  TOT  IIP  CPU QMPL    UNKN IDLE
 21.1  1.4  13.1   1.7  0.0  1.4  0.0 11.4  7.2  3.5  0.7    85.6  0.0
=============================================================================
WORKLOAD=WAS_WKL     SERVICE CLASS=TRADE       PERIOD=1 IMPORTANCE=4

TRANSACTIONS   TRANS-TIME SS.TTT  ---SERVICE----  SERVICE TIMES  ---APPL %---
AVG     18.28  ACTUAL         57  IOC          0  CPU    941.9   CP     13.01
MPL     18.28  EXECUTION      44  CPU      26720K SRB      0.0   AAPCP   2.89
ENDED  244689  QUEUED         13  MSO          0  RCT      0.0   IIPCP   0.00
END/S  407.82  R/S AFFIN       0  SRB          0  IIT      0.0
#SWAPS      0  INELIGIBLE      0  TOT      26720K HST      0.0   AAP   143.97
EXCTD       0  CONVERSION      0  /SEC     44533  AAP    863.8   IIP     0.00
AVG ENC 18.28  STD DEV        88                 IIP      0.0

RESP. TIME EX   PERF  AVG   ------ USING% ----- -- EXEC. DELAYS %--  ---DLY%--
.MM.SS.TTT VEL% INDX ADRSP   CPU  AAP  IIP  I/O  TOT QMPL  AAP  CPU  UNKN IDLE
.00.00.057 30.8  2.3  23.4   0.0  7.1  0.0  0.0 16.1  9.6  5.6  0.8  76.8  0.0
```

includes zAAP & zIIP-eligible transactions

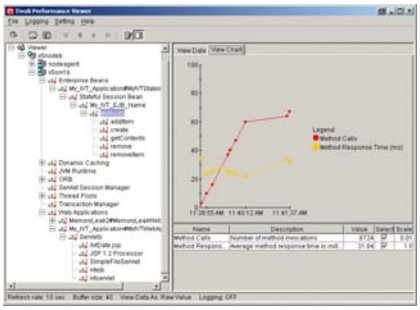
% time used by zIIP-eligible trans on CPs

% time running on zIIPs

% time used by zAAP-eligible trans on CPs

% time running on zAAPs

Performance Monitoring & Management



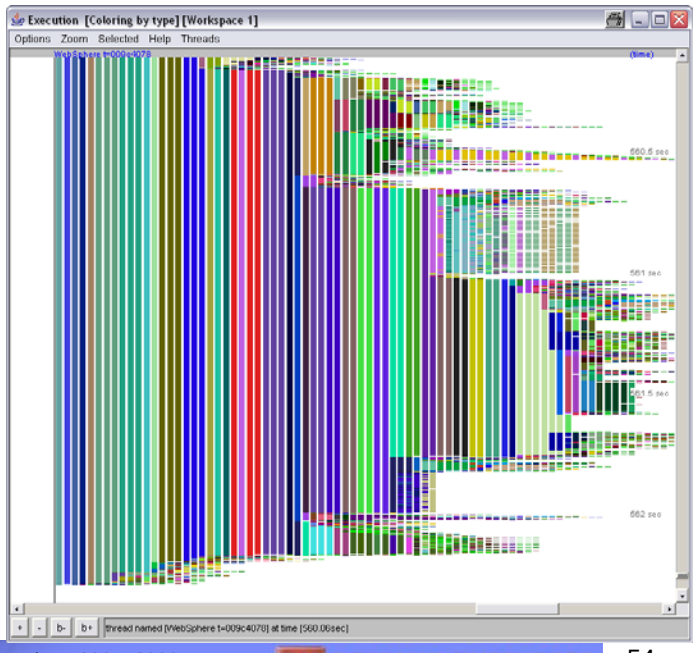
- **SMF/RMF on z/OS**
- **jinsightLive for System z - "Use Case" Profiler**
- **ITCAM for WebSphere** (IBM Tivoli Composite Application Monitor)
- **WebSphere Performance & Diagnostic Advisor** (integrated in WAS)
- **Tivoli Performance Viewer** (Integrated into AdminConsole for WAS V.6.1)
- **Tivoli Decision Support for z/OS** (SMF Records (120) moved to DB2)
- **CA Wily Technology Inc. Introscope** (PowerPack for WAS on z/OS)

& many others . . .

JinsightLive for IBM System z -"Use Case" Profiler

Zero in:
Right tool for the problem?

- http://www.alphaworks.ibm.com/tech/jinsightlive
- **Best way to see where time is going . . .**



- **Profile each use case**
  - Look for 'poor' choices - Repeated tasks that can be avoided
  - Profile in a production configuration (Data volume)
- Look for common patterns
  - Predominant use cases are visible
- 1st step in understanding use cases
  - Quicker than reading unfamiliar code.
- Will not find latching bottlenecks nor identify 'wait' time from 'CPU' time

Isolating problems

Zero in:
Right tool for the problem?

- **Time spent & CPU activity at method level:**
  - WSAM, and other application monitors
  - SMF 120 data (turn on activity records only for diagnostics)
    - General viewer: **www.ibm.com/software/webservers/appserv/zos_os390/** - "Trials & Betas"
    - Summary viewer: See PRS752 "Performance Summary Report for SMF 120 ..." on Techdocs
    - Overhead tolerable in many environments.

```
WSC SMF 120 Performance Summary V500 Date: Thu Oct 31 12:00:02 EDT 2003   SysID: SYSD
      - record subtypes: 1:Svr_Act. 3:Svr_Int. 5:EJB_Act. 6:EJB_Int. 7:Web_Act. 8:Web_Int.
SMF -Record Time     Server   Bean/WebAppName                  # of El.Time(mSec)   WLM_Encl_CPU_Time(uSec)
Numbr -Type hh:mm:ss Instance  Method/Servlet                 Calls    Avg     Max     Avg.     Max.    Min
1---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+----0----+---
  359 120.6 19:00:02 T5SRV1   MY_IVT_ApplicationMyIVTStatelessSession.jar
                              remove:                           5      1       2       758     1472     378
                              getContents:                      5      0       0       304      338     283
                              create:                           5     15      65     11177    31661     911
                              removeItem:java.lang.String       5      0       1       355      391     300
                              addItem:java.lang.String         15      0       0       330      609     284
  360 120.8 19:00:02 T5SRV1
                              ivtservlet                        3      1       1       845     1202     650
                              ivtejb                            3    115     301     62691   146527   18265
                              SimpleFileServlet                29     33     314      2544    18659    1712
                              JSP 1.2 Processor                 3   4041   12095   1414156 4234288    3747
                              /ivtDate.jsp                      3    141     420     60952   179122    1587
                             My_IVT_Application#MyIVTWebApp.war
```


WAS V7 - New SMF 120.9 Records

- **WebSphere for z/OS creates SMF 120 records.**
  - Issues with prior versions of WebSphere for z/OS:
    - Insufficient user/request information for Chargeback
    - Not extendable
    - Costly to record, Not dynamically controlled
- **WebSphere Version 7 introduces new subtype-9**
  - Dynamically enabled/disabled
  - Contains more information for chargeback
  - Show Bytes transferred, Elapsed Times, CP, zAAP, zIIP times
  - Low overhead
  - Extendable with user inserted sections
- **SMF Record Interpreter** available from the WebSphere Application Server for z/OS Web site at:
  - https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=zosos390
  - Select "**SMF Browser for WebSphere Application Server for z/OS V5 and V6**"

WAS V7 SMF Browser for SMF 120.9 Records

- Sample Output

```
 - Record subtypes: 1:Svr_Act. 3:Svr_Int. 5:EJB_Act. 6:EJB_Int. 7:Web_Act. 8:Web_Int. 9:Request
   -subtype 9 Sections: CPU:CPU, N:Network, Cl:Classification, S:Security, T:Timestamps, U:User

SMF -Record Time     Server   Bean/WebAppName  Bytes   Bytes  # of  El.Time  CPU_Time(uSec)
Numbr -Type hh:mm:ss Instance  Method/Servlet  toSvr   frSvr Calls   (msec)   Tot-CPU      zAAP
1---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+
    3 120.9  9:41:10 H1SR01B STC20577-HTTP  /                              3      2626      2192
         .9N ip addr=9.82.24.70 port=1603        210     675
          9CPU:Web ivtApp#ivt_app.war  /ivtserver                1        2      1699
          9CPU:Web ivtApp#ivt_app.war  //ivtAddition.jsp         1        0        43
    4 120.9  9:41:11 H1SR01B STC20577-HTTP  /                              4      3120      2467
         .9N ip addr=9.82.24.70 port=1604        201     567
          9CPU:EJB ivtApp::ivtEJB.jar::/create:                  1        0       219
          9CPU:Web ivtApp#ivt_app.war  /ivtejb                   1        2      1081
          9CPU:Web ivtApp#ivt_app.war  /ivtserver                1        1       554
          9CPU:EJB ivtApp::ivtEJB.jar::/remove:                  1        0       104
          9CPU:EJB ivtApp::ivtEJB.jar::/removeItem:java.lang     1        0        44
          9CPU:EJB ivtApp::ivtEJB.jar::/getContents:             1        0        41
          9CPU:EJB ivtApp::ivtEJB.jar::/addItem:java.lang.St     3        0       152

1---+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+----9----+

REQUEST Recs: Avg Bytes, TranCount & Times =     205     621   120        3      2231      1580

===SMF=120=V700===== End of Report ====== ====== End of Report =SMF=V700=JMH= 7July,2008
```


Isolating CPU problems

Zero in:
Right tool for the problem?

- **CPU usage at the detailed level:**
  - SMF 120 records provide CPU usage at the method level
  - CPU Time service (WSC program) can be used for your own detailed measurements
    - See PRS621 "CPU Time-used function for Java applications on z/OS" on Techdocs
    - See TD101339 "How-to find CPU TimeUsed in your WAS V5 for z/OS"
  - WAS V5 has method **`SMFJActivity.obtainTotalCpuTimeUsed()`** in pmi.jar

```
import com.ibm.ws390.sm.smf.SmfJActivity;
...
                    long startTime;
                    long stopTime;
                    long cpuTime;
                    startTime = SmfJActivity.obtainTotalCpuTimeUsed();

      < main Java code or method calls here >

                    stopTime = SmfJActivity.obtainTotalCpuTimeUsed();
                    cpuTime = stopTime - startTime;
                    System.out.println("CPU Time: " + cpuTime + " microseconds");
...
```

Isolating problems - Delays

Zero in:
Right tool for the problem?

- **WLM Delay Monitoring States:**
  - DISP - waiting for response from a distributed server
  - LOCL - waiting for session w/ server on the local system
  - SYSP - waiting for TCP/IP session establish w/ local system
  - REMT - waiting for TCP/IP session establish w/ remote system
  - SSLT* - waiting for SSL session in controller
  - REGT* - waiting for thread in controller
  - WORK* - waiting to register work in controller
  - OTHER - waiting for DNS or TCP/IP
  - TYP1 - EJB Collaborator
  - TYP2 - J2C Connector
  - TYP3 - RMI/IIOP
  - TYP4 - OTS call to RRS

* Added w/ APARs for WLM OW51848 & RMF OW52227

Adds up to 100%

```
          RMF V1R2   Work Manager Delays - WLMGL - Workload Activity Report

             RESP  ---------------------- STATE SAMPLES BREAKDOWN (%) -  ------STATE------
SUB    P     TIME  --ACTIVE-- READY IDLE  ------WAITING FOR---------- SWITCHED SAMPL(%)
TYPE         (%)    SUB  APPL             TYP4 REGT LOCL               LOCAL SYSPL REMOT
CB     BTE   0.0   26.9   0.0   0.0  0.0  65.4  3.8  3.8               0.0   0.0   0.0
CB     EXE   0.0    0.0   0.0   0.0  0.0   0.0  0.0  0.0               0.0   0.0   0.0

           EX   PERF   AVG   --USING%--  ----- EXECUTION DELAYS % -----
           VEL  INDX  ADRSP   CPU   I/O  TOTAL   CPU QMPL
 GOAL    40.0%
 ACTUALS 45.3%  .89   13.4   0.1   0.0   36.1  23.5 12.6
```


Displaying Threads & CPU Time Used:

`D OMVS,PID=nnnn,BRL`, or
SDSF – PS panel, D action character
(output goes to Syslog.)

```
BPXO040I 13.27.01 DISPLAY OMVS 506
USER     JOBNAME  ASID     PID       PPID STATE    START     CT_SECS
H2ASRU   H2SR01BS 0021 197284   84083363 HR---- 10.24.55 2231.967
  THREAD_ID        TCB@    PRI_JOB  USERNAME   ACC_TIME SC STATE
  1450F94000000000 008D6AD0                      36.296 IPT  YU
  . . .
  1451AE0000000037 008C90A8                        .089 WRT  JR V
  1451BD1000000038 008C5E88 WLM                  70.820 CLO  JR V
  1451CC2000000039 008C60D0 WLM                  70.586 CLO  JR V
  1451DB300000003A 008C62F0 WLM                  71.462 CLO  JR V
  1451EA400000003B 008C6510 WLM                 351.265 WRT  JR V
  1451F9500000003C 008C6730 WLM                  69.749 CLO  JR V
  145208600000003D 008C6950 WLM                 348.383 CLO  JR V
  145217700000003E 008C6B70 WLM                 348.176 CLO  JR V
  145226800000003F 008C6E00 WLM                  70.408 CLO  JR V
  145244A000000040 008C8098 WLM                  69.230 CLO  JR V
  1451A80E00000037 008C90B8                      .287PTC  JR V
```

Worker Threads

CPU Time (Secs)

Using JAVACORE Dumps to identify threads

In the javacore, thread 27070580 used 178 seconds out of a total of 1950 (9%) why?

```
3XMTHREADINFO "Thread-34"(TID:0x556C1600,sys_thread_t:0x554EADD8,state:CW
ID:0x27070580) prio=5
4XESTACKTRACE at java/lang/Object.wait(Native Method)
4XESTACKTRACE at java/lang/Object.wait(Object.java:231(Compiled Code))
4XESTACKTRACE at
com/ibm/tivoli/itcam/toolkit/ai/gccollector/Semaphore.waitForAndLock(Semaphore.j
   ava:69(Compiled Code))
. .
```

Thread in stack trace suggests it relates to ITCAM monitoring GC.

- Missing fix pack resulted in higher overhead for collecting performance data.
- After installing the required fixes this percentage dropped to 2-3%.

See *"Threads and excessive CPU consumption in WAS for z/OS"* Techdoc WP101474

IBM Support Assistant (ISA)

Free application simplifies & automates software support

- **Helps customers analyze & resolve questions and problems**
  - ibm.com/software/support/isa/
  - ibm.com/developerworks/websphere/techjournal/0906_supauth/0906_supauth.html

Java and WebSphere Troubleshooting Tools

- IBM Monitoring and Diagnostic Tools for Java
  - Health Center
  - Dump Analyzer
  - Garbage Collection and Memory Visualizer (GCMV)
- Memory Dump Diagnostic for Java (MDD4J)
- IBM Pattern Modeling and Analysis Tool for Java Garbage Collector (PMAT)
- IBM Thread and Monitor Dump Analyzer for Java
- Thread Analyzer
- WebSphere Application Server extensions for Dump Analyzer
- IBM Trace and Request Analyzer for WebSphere Application Server
- Database Connection Pool Analyzer for IBM WebSphere Application Server
- Log Analyzer
- Symptom Editor
- Visual Configuration Explorer

More Tools . . .

- **Workload simulators**
  - Rational Performance Tester
    - http://www.ibm.com/software/awdtools/tester/performance/
  - WebSphere Studio Workload Simulator
    - www.ibm.com/software/awdtools/studioworkloadsimulator/
  - MS Web Application Stress Tool - www.microsoft.com/technet/default.mspx
    - Search on 'Web Application Stress Tool'
  - Loadrunner - www.mercuryinteractive.com
  - Silk - www.segue.com
- **Java tools**
  - Javadump formatters – see appropriate IBM SDK, Java™ Diagnostics Guide
- **HTTP sniffers**
  - tcpmon - org.apache.axis.utils
  - ethereal - www.ethereal.com
- **MVS Sysprog tools**
  - MXI - www.mximvs.com/

Some Benchmark Experiences

The best tuned system cannot fix some application problems

- **Java Heap required by application**
  - Use JVM Verbose GC reports for information
  - Application requires much larger JVM heap
  - Application has a memory leak
- **Inefficient Application Code**
  - Application 'swallows' errors, making them invisible
  - Application use of storage and caching, caching algorithm used
  - Resource bundles or property files read for every transaction
  - Frequent, Verbose logging to Un-owned HFS
  - String handling, data conversions ASCII <->EBCDIC
  - Check WebSphere error log for errors and correct

Appendices

1. More WLM Options & Tools
2. Controlling WebSphere Workload license charges
3. MVS Commands & Displays
4. Resources & References

WLM Startup Options for WebSphere

- **Number of Servers:** *server_name* > Java and process management > Server instance.
  - Select the Multiple instances enabled field.
  - Minimum Number and Maximum Number of Instances.
- **protocol_accept_http_work_after_min_srs**
  - Wait for minimum number of servants ready before starting HTTP transport channels.

  True (1): HTTP transport channels start when minimum number of servants is ready for work.

  False (0): HTTP transport channels start when the controller starts.
- **protocol_accept_iiop_work_after_min_srs**

  True (1): IIOP transport channels start when minimum number of servants is ready for work.

  False (0): IIOP transport channels start when the controller starts.
- **wlm_servant_start_parallel** (New with WAS V7)

  1: After first servant is initialized, server starts remaining address spaces in parallel.

  0: Server starts all servant address spaces sequentially.

WLM Workload Distribution Options

- **Even distribution of HTTP requests:** *server_name* > Server Infrastucture/Administration > Administration services > Additional Properties > Custom properties > Check “WLMStatefulSession” property.

  (general property `wlm_stateful_session_placement_on` is ignored.)

- **server_use_wlm_to_queue_work**

  Specifies whether WLM is used for workload queuing.

  1: if you are using stateless application models.

  0: if you are using conversational application models

- **server_work_distribution_algorithm**

  This is only used if **`server_use_wlm_to_queue_work=false.`**

  0: Hot thread algorithm is used. (not recommended.)

  1: The round robin algorithm is used. This is the default.

More WLM Options for WebSphere

- **control_region_wlm_dispatch_timeout**
  - Limits the amount of time a client request waits on the WLM queue, as well as the time required for the application component to process the request.
- **protocol_iiop_local_propagate_wlm_enclave**
  - Propagate the WLM enclave associated with currently dispatched request on an outbound IIOP request made to another server on the same z/OS system.
- **control_region_timeout_save_last_servant**
  - Specifies whether the controller terminates the last available servant when a timeout situation occurs.

WLM WebSphere Routing Level algorithm

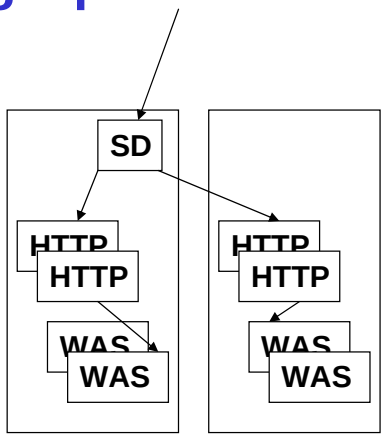
- New support in z/OS 1.9 uses displaceable CP capacity of systems as basis for routing work.
  - Function enabled on z/OS 1.6 and above with OA16486.
- In the past, WLM routing algorithm was round-robin.
- New IEAOPT parameter WASROUTINGLEVEL

  =1 Use the old Round-Robin routing algorithm. (the default)

  =0 Use LPAR capacity when making routing decisions.
  - "Over committed" systems shouldn't get additional work.
  - WLM will avoid systems that are in 'stress' (real storage shortage.)
  - May change routing recommendations compared to current behavior.
- Keep the same WASROUTINGLEVEL option on all systems of the sysplex.
- This applies only to the Daemon Routing IIOP requests.

Sysplex Distributor WLM Routing Options

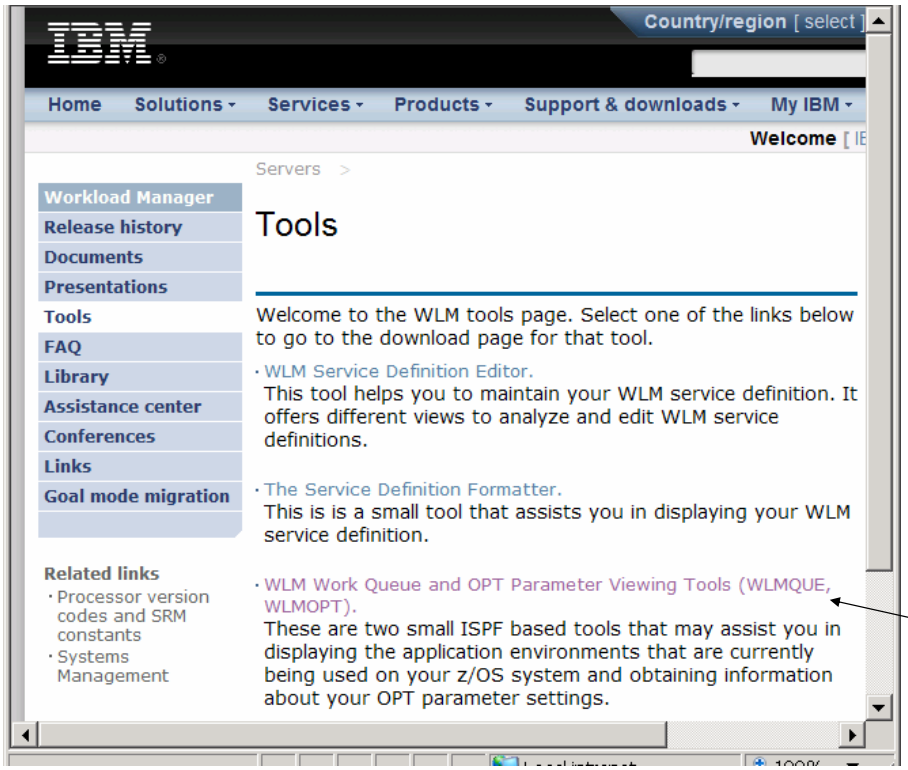
- **VIPADISTribute DISTMethod=**
  - **BASEWLM** – Route based on available GCP capacity.
  - **SERVERWLM** – Include zAAP/zIIPs in routing recommendations
  - **WEIGHTEDActive** – Balance requests proportional to connection weight.
  - **ROUNDROBIN** – (Ignore WLM routing.)
- **OPTLOCAL** (value) – Use local Server if Available & Healthy
  - Avoids traffic-routing through Sysplex Distributor.
  - Value=0: always use local connection (Req'd if ROUNDROBIN)
  - Value=1: use local connection unless server WLM weight=0
  - Value=2-16: multipliers to favor the local server's WLM weight



Recommendations (?)
- While it may seem good to route based on available capacity, Overhead is significantly reduced if the work stays on the local system.
- Stateful sessions reduce flexibility & may create imbalance

WLM Tools – WLMQUE & WLMOPT

http://www.ibm.com/servers/eserver/zseries/zos/wlm/tools



WLMQUE
WLMOPT

WLMQUE – Application Env. Monitor

```
Selection:  >HELP< >SAVE< >OVW< >ALL<
System: SYSB   Sysplex: WSCPLEX   Version: z/OS 010900  Time: 13:03:24

ApplEnv_ Type SubName_ WMAS Del Dyn NQ QLen Str Hav Unb Trm Min_ Max__
H2SR01   CB   H2SR01B  0054 No  Yes  1    1   0   1   0   0    9     9

WorkQue_ Del Wnt Hav ICnt QueIn_ QueOut QueLen QueTot__ Act_ Idl_
******** No    1   1    0    410    409      1   304120    1    6

SvAS Binding_ Ter Opr Btc Dem Have PEU_ ICnt WUQue____ Aff AffQue
0021 ******** No  No  No  No     9    9    0    216669   0      0
```

Key: ==========

```
 NQ       - Number of work queues (service classes)
 QLen     - Total number of currently queued requests
 QueIn    - Number of Requests inserted into work queue since last refresh
 QueOut   - Number of Requests taken from queue since last refresh
 QueLen   - Current queue length
 QueTot   - Total number of requests seen so far
 Binding  - Server class (work queue) from which the server AS selects work.
            - Shows dashes if the server address space is unbound
 Have     - Number of instances which can select work
 PEU      - Parallel execution units: number of defined server instances
              for the subsystem (for example: NUMTCB)
```

WLMOPT

```
                    WLM OPT Settings                                >SAVE<
System: SYSB      Version: z/OS 010900  OPT: 00  Time: not issued
OPT-Parameter:                Value:       Description:
ABNORMALTERM                     No Abnormal term. used in routing rec.
BLWLTRPCT                         5 CPU cap. to promote blocked work
BLWLINTHD                        60 Time blocked work waits for help
CCCAWMT                      490000 Alternate wait management time value
ZAAPAWMT                      12000 AWM time value for zAAPs
ZIIPAWMT                      12000 AWM time value for zIIPs
CNTCLIST                         No Clist commands count individually
CPENABLE                       0,0 LOW,HI thresh for % TPI int. x 100
DVIO                            Yes Specifies w/ directed VIO is active
ERV                          500,CB Enq res. CPU Service and DP
IFAHONORPRIORITY                Yes Specifies w/ zAAP work can run on CPs
IIPHONORPRIORITY                Yes Specifies w/ zIIP work can run on CPs
INITIMP                       0,FE INITIMP value and DP for initiators
MCCAFCTH                  400,800 LOW,HIGH central threshold
MCCFXEPR                         92 % of storage fixed within first 16MB
MCCFXTPR                         80 % of online storage fixed
PROJECTCPU                       No CPU projection for zAAPs and zIIPs
RCCFXTT                       66,72 Low,High Logical MPL threshold
RCCFXET                       82,88 Low,High Physical MPL threshold
RMPTTOM                       3000 SRM invocation interval
VARYCPU                         Yes VARYCPU is enabled
VARYCPUMIN                        1 VARYCPUMIN value
WASROUTINGLEVEL                   0 WebSphere Routing Level
```


Controlling WebSphere Workload License Charges

- **Limit WebSphere to a Maximum amount of MSUs**
  - Customer wants to Control the Budget for Software Pricing based on MSUs
  - Useful for **Getting Started SubCapacity** (GSLS) Pricing
  - Useful in Test or Development Environments;
  - Not in Production where Performance Matters!
- **Isolate an LPAR for WebSphere Work, and use:**
  - PR/SM Capping, or . . .
  - Group Capacity limits (4-hour moving Average)

MVS Modify <server> Command - Help

```
F <server>,HELP
THE COMMAND MODIFY MAY BE FOLLOWED BY ONE OF THE FOLLOWING KEYWORDS:
CANCEL - CANCEL THIS CONTROL REGION
TRACEALL - SET OVERALL TRACE LEVEL
TRACEBASIC - SET BASIC TRACE COMPONENTS
TRACEDETAIL - SET DETAILED TRACE COMPONENTS
TRACESPECIFIC - SET SPECIFIC TRACE POINTS
TRACEINIT - RESET TO INITIAL TRACE SETTINGS
TRACENONE - TURN OFF ALL TRACING
TRACETOSYSPRINT - SEND TRACE OUTPUT TO SYSPRINT (YES/NO)
DISPLAY - DISPLAY STATUS
TRACE_EXCLUDE_SPECIFIC - EXCLUDE SPECIFIC TRACE POINTS
JAVACORE - GENERATE JVM CORE DUMP
HEAPDUMP - GENERATE JVM HEAP DUMP
JAVATDUMP - GENERATE JVM TDUMP
TRACEJAVA - SET JAVA TRACE OPTIONS
TRACETOTRCFILE - SEND TRACE OUTPUT TO TRCFILE (YES/NO)
MDBSTATS - MDB DETAILED STATISTICS
PAUSELISTENERS - PAUSE THE COMMUNICATION LISTENERS
RESUMELISTENERS - RESUME THE COMMUNICATION LISTENERS
STACKTRACE - LOG JAVA THREAD STACK TRACEBACKS
TIMEOUTDUMPACTION - SET TIMEOUT DUMP ACTION
TIMEOUTDUMPACTIONSESSION - SET TIMEOUT DUMP ACTION SESSION
TIMEOUT_DELAY - SET TIMEOUT DELAY VALUE
WLM_MIN_MAX - RESET WLM MIN/MAX SERVANT SETTINGS
SMF - SET SMF120 OPTIONS
DPM - DISPATCH PROGRESS MONITOR
```


MVS Modify <server>,Display,Help

```
F <server>,DISPLAY,HELP
THE COMMAND DISPLAY, MAY BE FOLLOWED BY ONE OF THE FOLLOWING KEYWORDS:
SERVERS - DISPLAY ACTIVE CONTROL PROCESSES
SERVANTS - DISPLAY SERVANT PROCESSES OWNED BY THIS CONTROL PROCESS
LISTENERS - DISPLAY LISTENERS
CONNECTIONS - DISPLAY CONNECTION INFORMATION
TRACE - DISPLAY INFORMATION ABOUT TRACE SETTINGS
JVMHEAP - DISPLAY JVM HEAP STATISTICS
WORK - DISPLAY WORK ELEMENTS
ERRLOG - DISPLAY THE LAST 10 ENTRIES IN THE ERROR LOG
MODE - DISPLAY THE EXECUTION BITMODE
THREADS - DISPLAY THREAD STATUS  (WAS V7)
WLM - DISPLAY WLM SETTINGS
SMF - DISPLAY SMF120-9 SETTINGS AND STATUS
FRCA - DISPLAY FRCA INFORMATION
DPM - DISPLAY DISPATCH PROGRESS MONITOR SETTINGS
END OF OUTPUT FOR COMMAND DISPLAY,HELP
```

Display a list of all the keywords you can use with the modify `timeoutdumpacation` or `timeoutdumpactionsession` command:

```
f <server>,timeoutdumpactionsession=help
BBOO0178I MODIFY TIMEOUTDUMPACTIONSESSION= MAY BE FOLLOWED BY ONE OF
        THE FOLLOWING KEYWORDS:
BBOO0179I SVCDUMP - SVC DUMP
BBOO0179I JAVACORE - JAVA CORE DUMP
BBOO0179I NONE - NO DUMP
```

MVS Modify Command to Display Work

```
F <server_name>,DISPLAY,WORK,HELP
BBOO0178I THE COMMAND DISPLAY,WORK, MAY BE FOLLOWED BY ONE OF
THE FOLLOWING KEYWORDS:
BBOO0179I EJB - DISPLAY EJB REQUEST COUNT INFORMATION
BBOO0179I SERVLET - DISPLAY SERVLET REQUEST COUNT INFORMATION
BBOO0179I MDB - DISPLAY MDB REQUEST COUNT INFORMATION
BBOO0179I SIP - DISPLAY SIP REQUEST COUNT INFORMATION
BBOO0179I SUMMARY - DISPLAY SUMMARY REQUEST COUNT INFORMATION
BBOO0179I ALL - DISPLAY ALL REQUEST COUNT INFORMATION
BBOO0179I CLINFO - DISPLAY WORK CLASSIFICATION INFORMATION
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,WORK,HELP
```

WebSphere Operator Display Commands to determine:
Work, Queued or Active + Deltas provided between invocations:

```
F H2SR01B,DISPLAY,WORK
BBOO0255I TIME OF LAST WORK DISPLAY 2008/06/12 14:32:15.215714
BBOO0261I TOTAL REQUESTS TO SERVER     414120    (DELTA 316139)
BBOO0262I TOTAL CURRENT REQUESTS       9
BBOO0263I TOTAL REQUESTS IN DISPATCH   9
BBOO0268I TOTAL TIMED OUT REQUESTS     0    (DELTA 0)
BBOO0188I END OF OUTPUT FOR COMMAND DISPLAY,WORK
```


WLM Dynamic Application Environments

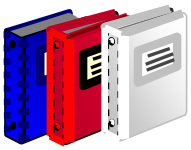
```
D WLM,DYNAPPL=*
IWM029I  12.57.17  WLM DISPLAY 590
  DYNAMIC APPL. ENVIRON. NAME      STATE     STATE DATA

  F6SR01                           AVAILABLE
  ATTRIBUTES: PROC=F6ASRA   SUBSYSTEM TYPE: CB
  SUBSYSTEM NAME: F6SR01A   NODENAME: F6CELL

  F6SR01ADJUNCT                    AVAILABLE
  ATTRIBUTES: PROC=F6CRAA   SUBSYSTEM TYPE: CB
  SUBSYSTEM NAME: F6SR01A   NODENAME: F6CELL
```

```
V WLM,DYNAPPL=F6SR01,RESUME | QUIESCE | REFRESH
```

Dynamic WLM Env's Started and Stopped Dynamically, but can be used to:

- QUIESCE - WLM stops the server address spaces.
- RESUME - WLM starts the server address spaces.
- REFRESH - WLM stops the server address spaces, and starts new ones.

Resources & References



WebSphere Application Server Information Center

- ibm.com/software/webservers/appserv/was/library/
  - Down load a copy onto your workstation

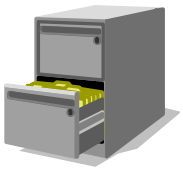
WebSphere for z/OS "home page"

ibm.com/software/webservers/appserv/zos_os390/

Redbooks: www.redbooks.ibm.com

- *Systems Programmer's Guide to: Workload Manager – SG24-6472*
- *Performance Monitoring & Best Practices for WAS on z/OS* - SG24-7269
- *Monitoring WebSphere Application Performance on z/OS* - SG24-6825
- *Writing Optimized Java Applications for OS/390* - SG24-6541
- *WebSphere for z/OS V6 Problem Determination* - SG24-6880
- *WebSphere V6 Scalability & Performance Handbook* - SG24-6392
- *WebSphere for z/OS to CICS & IMS Connectivity Performance* – REDP-3959

Build a library of WAS & Java for z/OS pubs



- Developers & Sysprogs need access to z/OS specific information
- Information is perishable and time sensitive
  - Out of date information is like no information or bad information.

Education

- **Courses by IBM Learning Services, ITSO, & WSC**

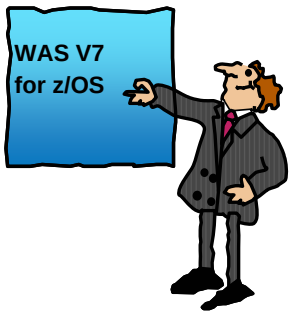
  www.ibm.com/services/learning/

  - ES685 - WAS V6 Implementation Workshop (4.5 Days)
  - OZ850 - "Maximizing WebSphere for z/OS V6 Performance" (4.5 Days)

  Wildfire Workshops:

  - WBSR7 - WebSphere V7 for z/OS Workshop "Gen 7" (2.5 days)
  - WSW07 - Security Workshop: WAS V7 for z/OS (2.5 Days)

- **Conferences & User Group Meetings**
  - SHARE & Regional User Groups
  - zSeries Expo
  - Large Systems z/OS Update
  - WebSphere Virtual Usergroup



WAS for z/OS Performance articles on Techdocs

White Papers:

- WP101206 Installing ITCAM V6.1 for WebSphere on z/OS
- WP101342 Understanding SMF Record Type 120, Subtype 9
- WP101374 WebSphere Application Server for z/OS V7 - Dispatch Timeout Improvements
- WP101138 WebSphere z/OS V6.1 - Hidden Gems and Little Known Features
- WP101121 The 64-bit Effect Five Different Ways to Look at Applications
- WP100678 Diagnosing Performance Problems with WebSphere Application Server on z/OS
- WP100558 Optimizing WebSphere for z/OS Performance
- WP100489 Mission: zAAP your costs Running WebSphere and Java on the zAAP
- WP100417 z/OS Performance: Capacity Planning Considerations for zAAP Processors
- WP100392 Exploiting web services in WebSphere for z/OS
- **WP101476 Value of Co-Location with WebSphere for z/OS**
- **WP101490 Introduction to Optimized Local Adapters**
- **WP101474 Threads and excessive CPU Consumption in WebSphere on z/OS**

ATS/WSC TechDocs - http://www.ibm.com/support/techdocs

More Performance articles on Techdocs

Hints & Tips:

- **TD104172 WSC Guidelines for a Healthy WebSphere Runtime on z/OS**
- TD103548 Capacity Planning for zAAP and zIIP Specialty Engines
- TD103036 Performance and tuning tips for WebSphere Application Server for z/OS
- TD102730 Classify the Application Control Region in WLM OMVS rules
- TD102454 How to find CPU Time Usage in your WebSphere V6 for z/OS java programs
- TD101645 Tivoli Performance Viewer Security
- TD101216 Tracing and Analyzing Java Garbage Collection in WebSphere for z/OS V5
- TD101199 Enabling the WSAD Application Profiler in a WAS V5 for z/OS Environment
- TD101152 Manage the Number of Servant Regions with WAS for z/OS V5 and WLM
- TD101151 How to Classify HTTP Transactions in WebSphere for z/OS V5

Presentations & Downloads:

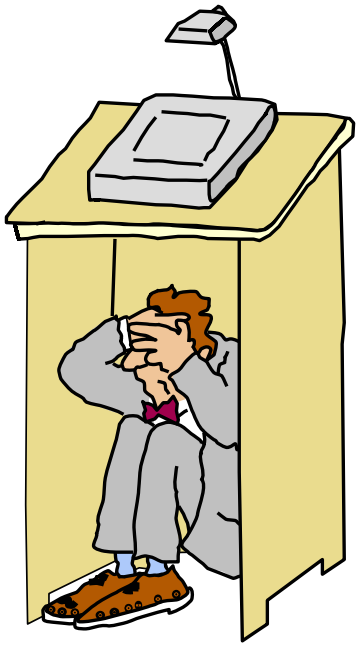
- PRS752 Performance Summary Report for SMF 120 records from WAS for z/OS
- PRS2494 Performance Engineering for WebSphere Application Server for z/OS
- **PRS3317 WLM Configuration & Advanced Topics for WAS on z/OS**

ATS/WSC TechDocs - http://www.ibm.com/support/techdocs

Other Resources . . .

- **Developer's Domain** (WebSphere & Java Best Practices, Help, Docs & Tools)
  - www-136.ibm.com/developerworks/websphere/
- **Java Specifications** (J2EE, EJB, JSP, Servlet, JNDI) Papers
  - java.sun.com/j2ee/docs/
- **Java Community Process**
  - jcp.org/
- **z/OS Home Page**
  - www.ibm.com/servers/eserver/zseries/zos
- **IBM Support Assistant (ISA) V.4.1**
  - http://www-01.ibm.com/software/support/isa/
  - http://www.ibm.com/developerworks/websphere/techjournal/0906_supauth/0906_supauth.html
- **Publications** on-line (view, print, order books)
  - **www.ibm.com/servers/eserver/zseries/zos/bkserv**
  - **www.ibm.com/servers/eserver/zseries/softcopy**

Questions?



Trademarks & Disclaimers

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries. For a complete list of IBM Trademarks, see www.ibm.com/legal/copytrade.shtml: e-business logo, eServer, IBM, IBM Logo, MVS, OS/390, S/390, Websphere, z/OS, zSeries

The following are trademarks or registered trademarks of other companies

Java and all Java-related trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries

UNIX is a registered trademark of The Open Group in the United States and other countries.

*** All other products may be trademarks or registered trademarks of their respective companies.**

NOTES:

- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
- IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.
- References in this document to IBM products or services do not imply that IBM intends to make them available in every country.
- Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.
- The information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.
- Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.