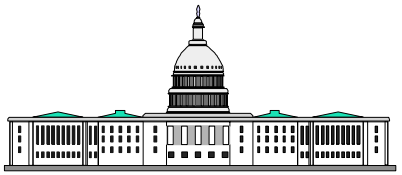
# zAAP - What it Can Do for You!

**Share, Winter, 2006**
**Session: 2825**
**Project: MVS CP**

*Kathy Walsh*
*IBM*
*Washington Systems Center*
*Advanced Technical Support*



# Trademarks and Disclaimers

AIX*
AIX 5L*
BladeCenter
Chipkill
DB2
DB2 Universal Database
Domino
Enterprise Storage Server*
e-business logo*
GDPS*
Geographically Dispersed Parallel Sysplex
HiperSockets
IBM*
IBM eServer
IBM logo*
iSeries
Lotus*
OnForever*
Parallel Sysplex*
POWER
POWER5
Predictive Failure Analysis*
pSeries*
S/390
ServerProven*
Tivoli*
TotalStorage*
TotalStorage Proven
Virtualization Engine
X-Architecture
xSeries*
z/OS*
z/VM*
zSeries*
System z9*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Intel is a trademark of Intel Corporation in the Unites States, other countries or both.
Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc., in the United States and other countries
Linux is a registered trademark of Linux Torvalds in the United States other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries.
Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

* 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.

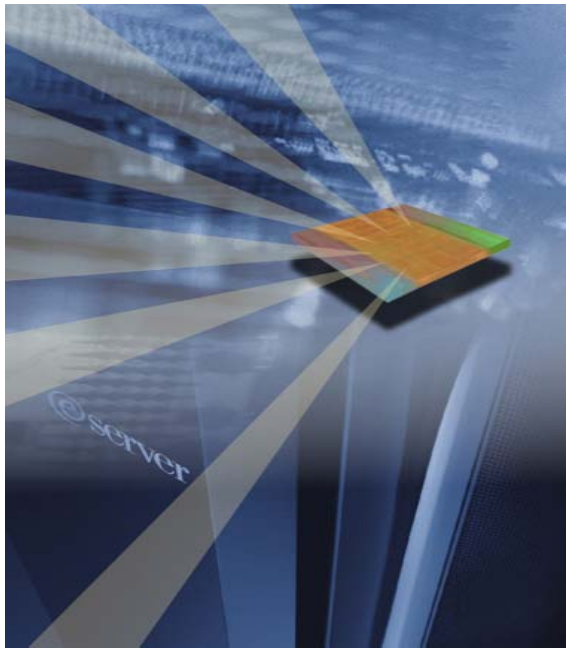
# Agenda

- Introducing the zAAP
- Technical Overview
- Capacity Planning Information
- WSC Experiences

# The New zSeries Application Assist Processor (zAAP)

***New specialty assist processor dedicated exclusively to execution of Java workloads under z/OS® – e.g. WebSphere®, CICS, IMS, DB2***

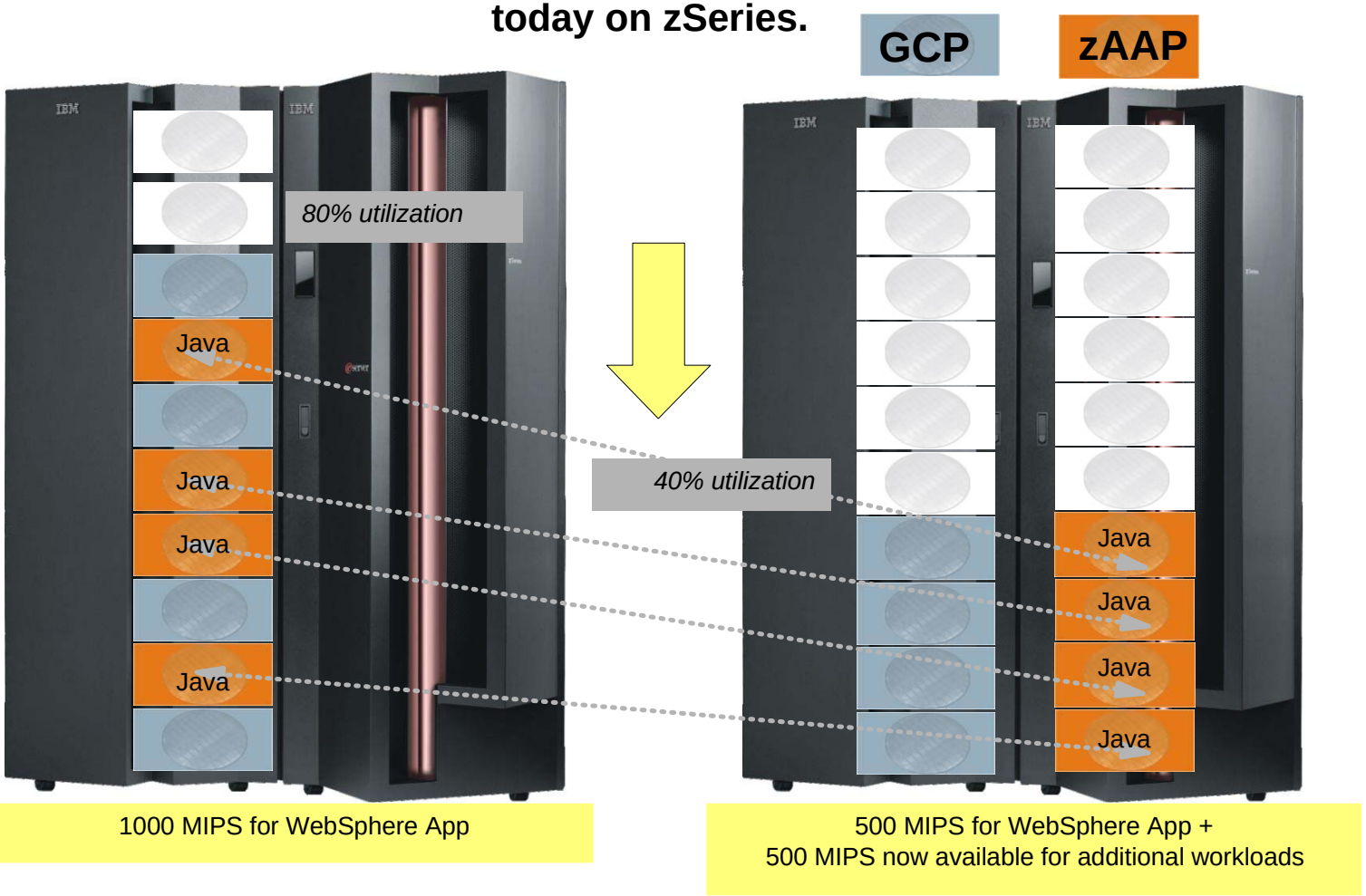
- Available on IBM System z9 and @server zSeries z990 and z890
- Used by workloads with Java cycles, e.g. WebSphere, DB2®
  - Executes Java code with no changes to applications
- Attractively priced, much lower than standard CPs
- Significantly lower maintenance costs than standard CPs
- Traditional IBM zSeries software charges unaffected
- Sub-capacity eligible IBM software charges can be reduced
- Up to 1 zAAP per general purpose processor in a CEC




***Objective: Enable integration of new Java based Web applications with core z/OS backend database environment for high performance, reliability, availability, security, and lower total cost of ownership***

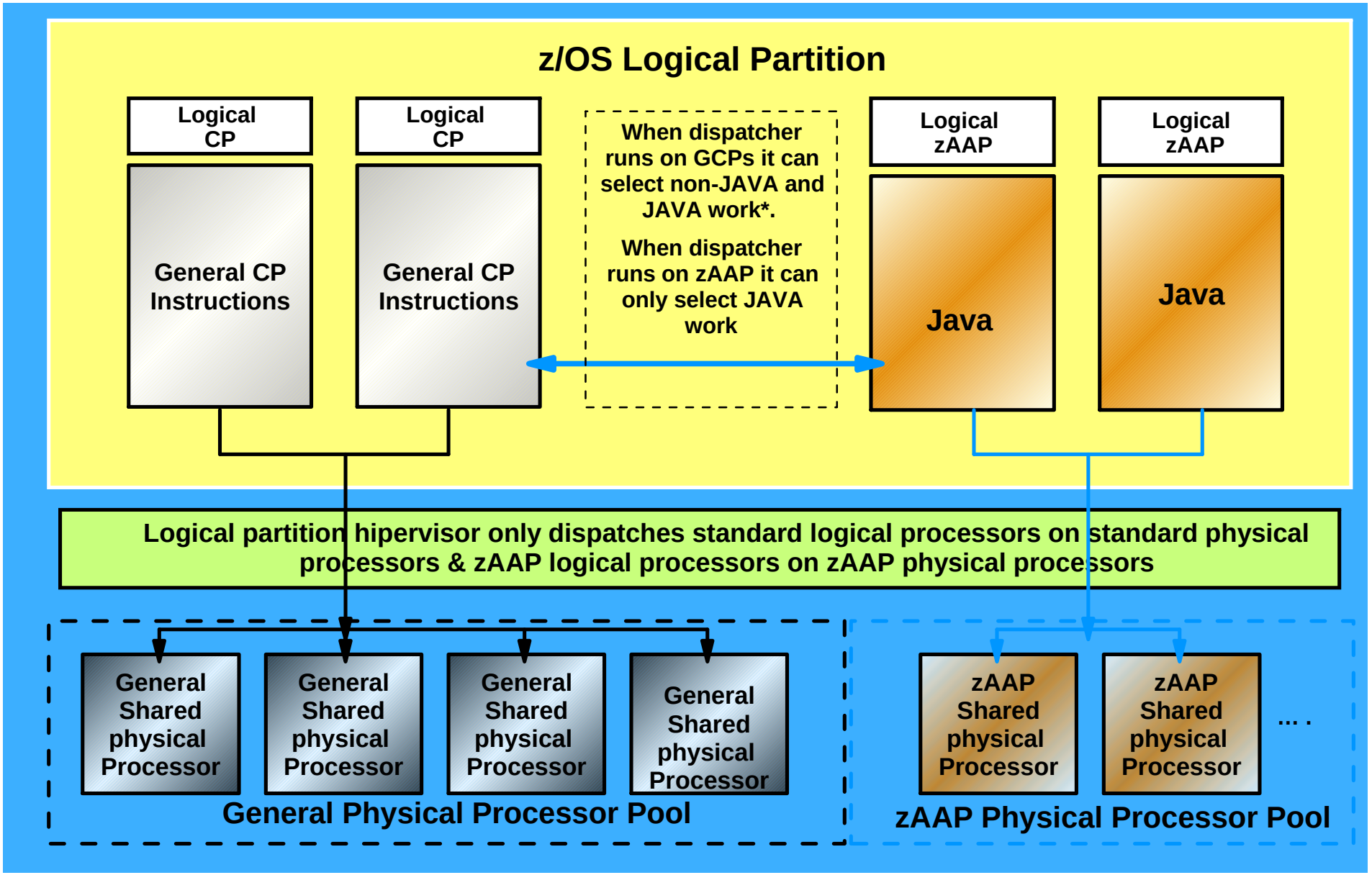
# zAAP Overview: A Simplified Example…

**Consider a WebSphere Application that is transactional in nature and requires 1000 MIPS today on zSeries.**




***In this example, with zAAP, we can reduce the standard CP capacity requirement for the Application to 500 MIPS or a 50% reduction.*** ** For illustrative purposes only*

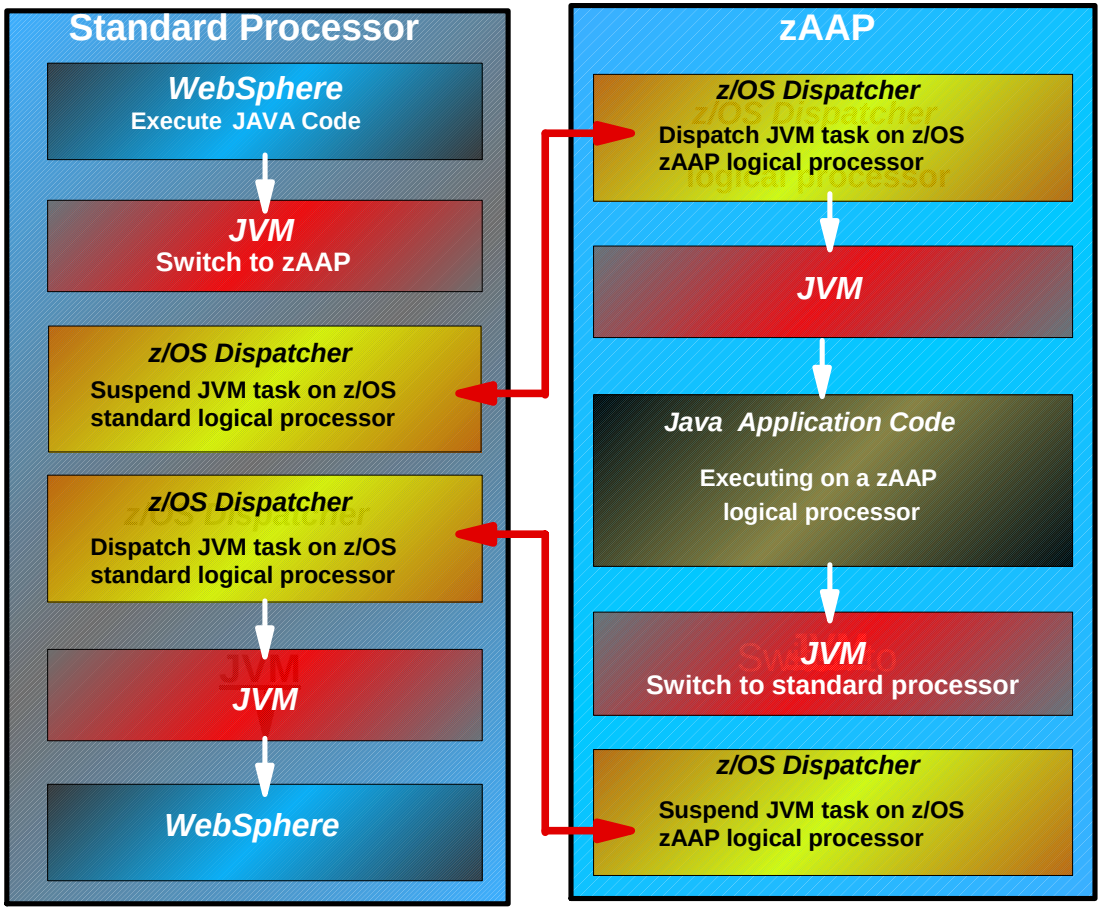
# zAAP Technical Overview: *z/OS zAAP Partition*



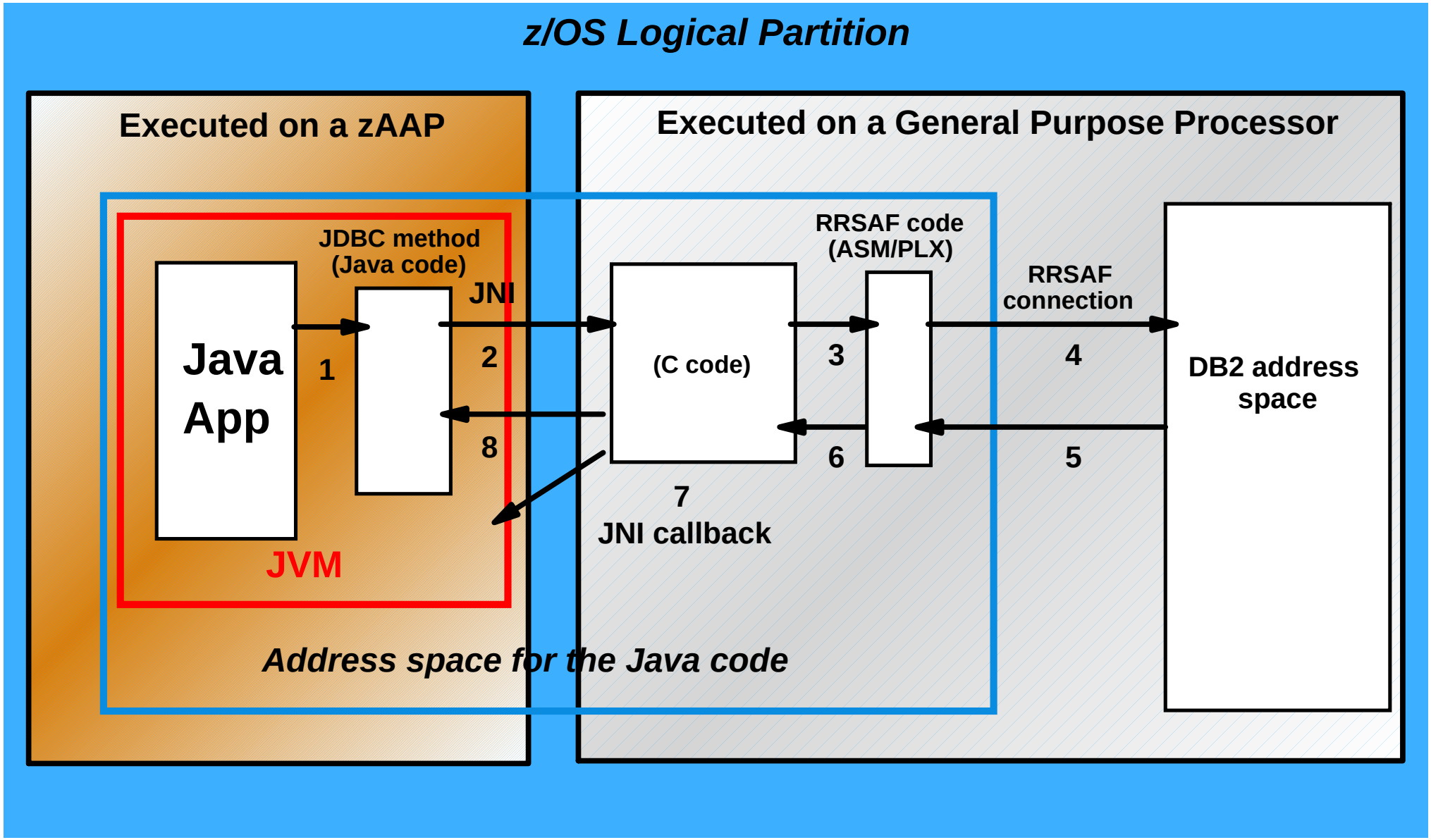

* Subject to installation controls

# zAAP Workflow: Executing Java under IBM JVM control

- IBM JVM, parts of LE runtime, and z/OS Supervisor are needed to support JVM execution on zAAPs
- IBM JVM communicates to z/OS dispatcher when Java code is to be executed
  - When Java is to be executed, the work unit is "eligible" to be dispatched on a zAAP
  - zAAP ineligible work is only dispatched on standard processors
  - There are installation controls to limit the use of standard processors to execute zAAP eligible work

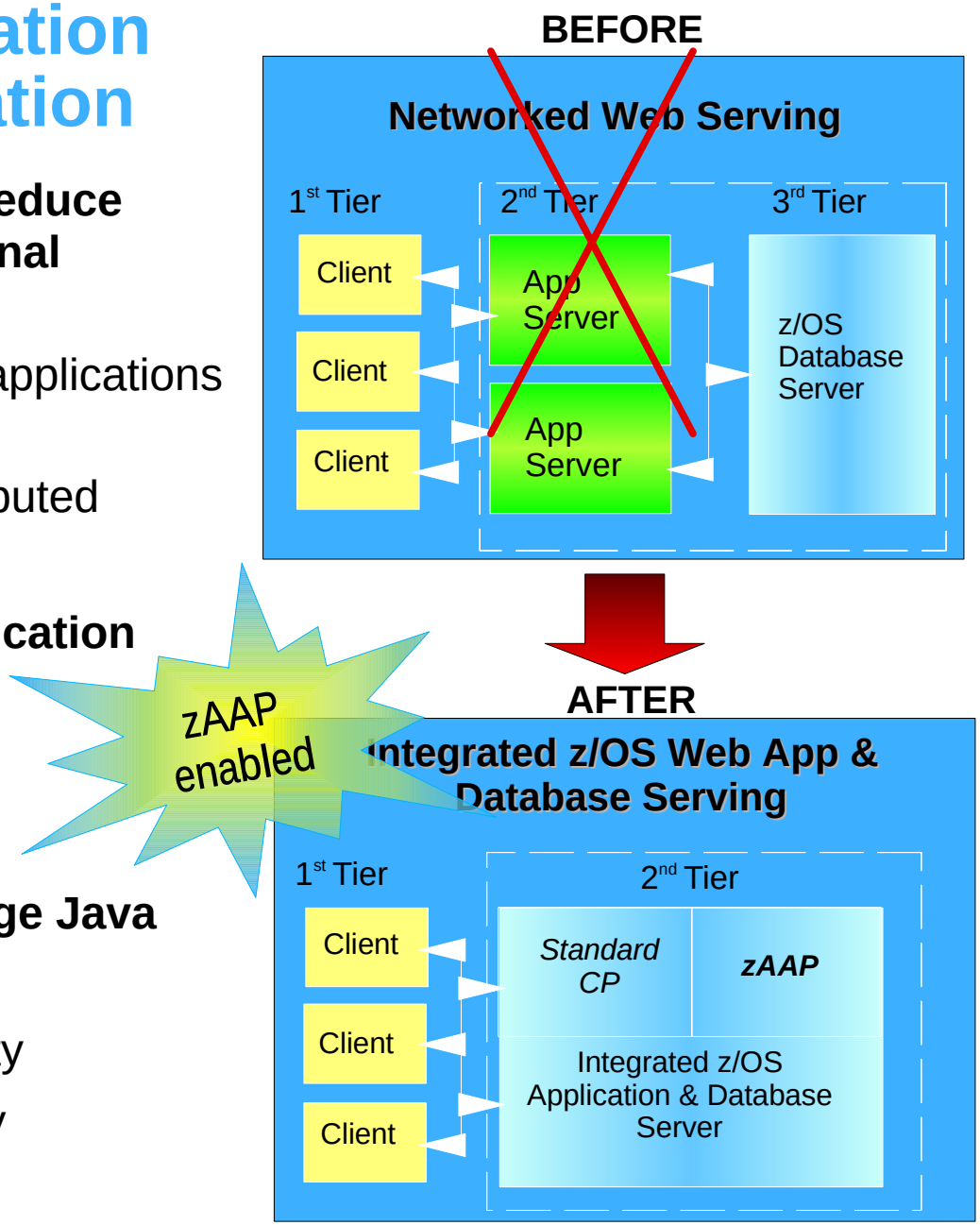


# zAAP Integration at Work: Java App calling DB2



# Increased e-business Integration and Infrastructure Simplification

- **zAAPs can help consolidate, simplify and reduce server infrastructure and improve operational efficiencies.**
  - Enables strategic integration of e-business applications with mission-critical database workloads
  - Potential operational advantages over distributed multi-tier solutions
- **Can eliminates separate tier to handle application server workload**
  - Remove one hardware tier
  - Remove one TCP/IP stack
- **Leverage core zSeries strengths and manage Java Workloads automatically with z/OS**
  - Security
  - Workload Manager (WLM)
  - Availability
  - Scalability
  - Flexibility



# How zAAPs Differ from General Purpose Processors

- **Some zAAPs Limitations**
  - zAAPs cannot be IPLed
  - zAAPs only execute z/Architecture™ mode instructions
  - zAAPs do not support all manual operator controls
    - No: PSW Restart, LOAD or LOAD derivatives (load from file, CDROM, Server)
  - zAAPs don't respond to SIGP requests unless enabled by a z/OS which supports zAAPs
  - Additional architecture differences are anticipated in future implementations
    - e.g., Java specific performance enhancements
- **The z/OS design accommodates processor differences for zAAPs:**
  - No I/O interrupts
  - No Clock Comparator interrupts
  - No affinity scheduling

# Requirements for zAAP Exploitation

- **Available on z990, z890, IBM System z9 and follow-on models only**
- **z/OS and z/OS.e exclusive**
- **Prerequisites:**
  - z/OS 1.6 (or z/OS.e 1.6)
  - IBM SDK for z/OS, Java 2 Technology Edition, V1.4 with PTF for APAR PQ86689
  - Processor Resource/Systems Manager (PR/SM) must be enabled
- **Subsystems and Apps using SDK 1.4 exploit zAAPs automatically:**
  - WAS 5.1
  - CICS® /TS 2.3
  - DB2 V7, DB2 V8
  - IMS™ V7, IMS™ V8, IMS™ V9
  - WebSphere WBI for z/OS
- **zAAPs must be jointly configured with general purpose processors within z/OS LPARs**
  - Number of zAAPs may not exceed the number of permanently purchased CPs (including z990 unassigned CPs or z890 Downgrade - Record Only CPs) on a given machine model.

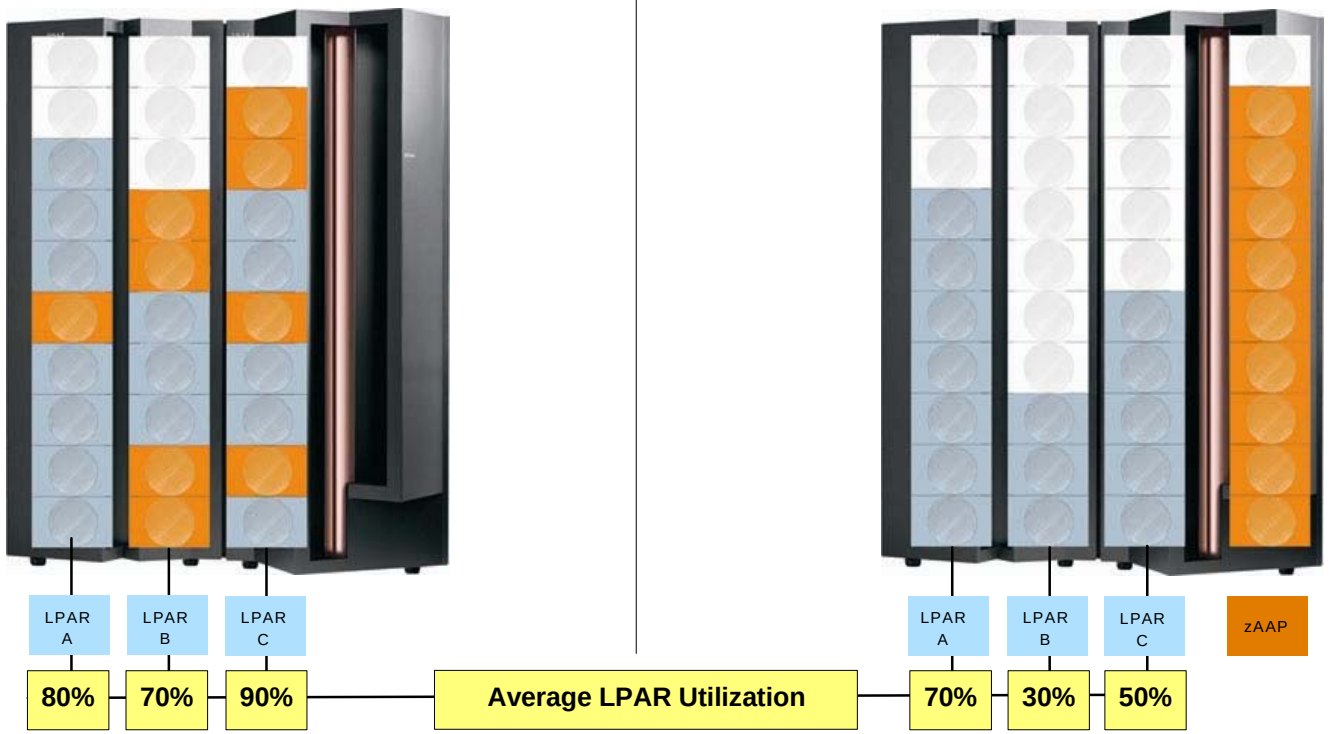
# Potential zAAP Impact on Full Capacity Pricing Scheme

***BEFORE zAAP:***

- Machine Type: 2084-B16
- Rated @ 647 MSUs
- Full-Capacity EWLC @ 647 MSUs
- Average Prime Shift Machine Utilization = 80%

***AFTER zAAP:***

- Machine Type: 2084-B16
- Rated @ 647 MSUs
- Java cycles executed on zAAPs
- Full Capacity EWLC remains @ 647 MSUs
- Average Prime Shift Machine Utilization = 50%



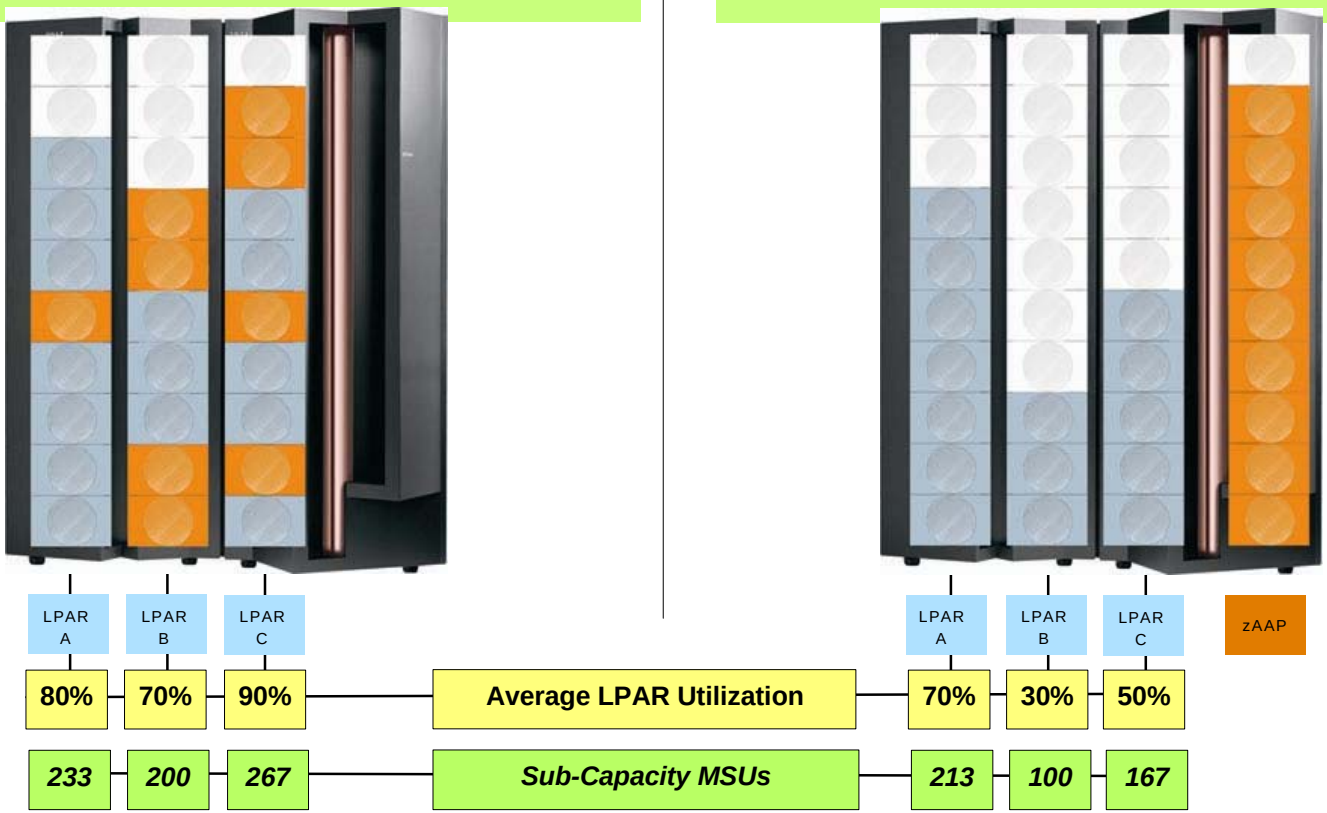
# Potential zAAP Impact on Sub-Capacity Pricing Scheme

***BEFORE zAAP:***

- Machine Type: 2084-B16
- Rated @ 647 MSUs
- Sub-Capacity Pricing based on;
  - LPAR A rolling 4hr avg @ 233 MSUs
  - LPAR B rolling 4hr avg @200 MSUs
  - LPAR C rolling 4hr avg @ 267 MSUs
- Rolling 4hr avg of Machine = 547 MSUs
- Average Prime Shift Machine Utilization = 80%

***AFTER zAAP:***

- Machine Type: 2084-B16
- Rated @ 647 MSUs
- Java cycles executed on zAAPs
- New Sub-Capacity Pricing on *reduced* rolling 4hr avg
  - LPAR A rolling 4hr avg @ 213 MSUs
  - LPAR B rolling 4hr avg @ 100 MSUs
  - LPAR C rolling 4hr avg @ 167 MSUs
- New Rolling 4hr avg of Machine = 480 MSUs
- Average Prime Shift Machine Utilization = 50%



# How Do I Know if zAAP is Right for My Workloads?

- **Instrumented in IBM SDK 1.3.1 and IBM SDK 1.4.1**
  - Available with Excel Summary Workbook
  - Runs in test environment
  - Gathers usage information on % of Java workload which can execute on a zAAP
  - Useful in predicting number of zAAPs necessary for optimum configuration
- ***z/OS Performance: Capacity Planning Considerations for zAAP* White Paper**
  - Describes the zAAP Projection Tool
  - Describes the prototype measurements
  - Describes Capacity Planning Methodology
- **Size 390**
  - Provides special assistance for the sizing methodology described in White Paper
  - May help with sizing consolidation of distributed Java workloads onto zSeries and zAAPs
- **Information is on the web**
  - http://www-1.ibm.com/servers/eserver/zseries/zaap/gettingstarted/

# Subsystem and Minimum Java Levels for zAAP Estimation

| Subsystem Version | IBM SDK for OS/390, Java 2 Technology Edition, V1.3.1 at PTF level UQ94379 (SR24 or later) | IBM SDK for z/OS, Java 2 Technology Edition, V1.4, at PTF level UQ88783 or later |
|---|---|---|
| WAS V4*, WAS V5.0* | X | |
| **WAS 5.1**** | | X |
| **IMS™ V7**** | X | X |
| **IMS V8**** | X | X |
| **IMS V9**** | | X |
| CICS® 2.2* | X | |
| **CICS 2.3**** | | X |
| **DB2® V7**** | X | X |
| **DB2 V8**** | X | X |

*** Configurations which support zAAPs**

**** zAAP projection tool can be used to assist in capacity planning**

# Technical Overview

# Enabling the zAAP

- **Enable zAAP processing at the JVM level through use of the -Xifa: options**
- **If there is no zAAP hardware present at the first call all additional JVM / zAAP calls will be stopped**
  - Use the -Xifa:force option to prevent this
- **To activate the projection tool, the -Xifa:project keyword must be specified in the JVM profile**
  - The output is directed to STDERR

| Keyword | Explanation |
|---|---|
| -Xifa:on | Default - allows Java work to run on zAAPs if available |
| -Xifa:force | JVM is to continue issuing switch calls even when zAAPs are unavailable |
| -Xifa:off | Disable switching support |
| -Xifa:projectn | Creates the zAAP projection tool data. n is the number of minutes in the projection interval |

# For IBM System z9 Processors
## Multiple ICF Pools Managed Independently

- **PU Pool - Physical PUs to dispatch to online logical PUs**
- **z9-109 with 10 CPs, 1 ICF, 2 IFLs, and 3 zAAPs online**
  - CP Pool contains 10 engines
  - ICF pool contains 1 ICF
  - IFL pool contains 2 IFLs
  - zAAP pool contains 3 zAAPs
    - zAAP weight can be independently set
- **z990 with 10 CPs, 1 ICF, 2 IFLs, and 3 zAAPs**
  - CP Pool contains 10 engines
  - **Specialty pool** contains 6 engines - ICFs, IFLs, zAAPs
  - zAAP LPAR weight is set equal to the initial z/OS LPAR weight
- **z890 with 2 CPs, and 1 zAAP and either ICF or IFL**
  - CP Pool contains 2 engines
  - Speciality pool contains 2 engines - zAAP and a ICF/IFL
  - zAAP LPAR weight is set equal to the initial z/OS LPAR weight

# For @server zSeries z890 and z990 processors

## Single Shared ICF Pool Considerations

- **zAAPs, CFs, and Linux partitions all use ICF CPs which are managed out of a single pool of capacity**
  - Managed independently from the General CP pool
- **zAAPs will acquire their characteristics from the z/OS partitions using the zAAPs**
  - If z/OS uses dedicated CPs, the zAAPs defined to the partition will be dedicated
  - If z/OS uses shared CPs, the zAAPs defined to the partition will use shared CPs and the weight given to the zAPPs will be equal to the z/OS partitions weight
- **The ICF pool partition weights need to be updated to reflect the introduction of the zAAP**

# Updating CF and Linux Partition Weights with a zAAP

| | Base Case | | Add a zAAP | | | |
|---|---|---|---|---|---|---|
| **Hardware** | **Partition** | **4 GCPs, 1 ICF** | **Partition** | **4 GCPs, 1 ICF, 1 zAAP** | **Partition** | **4 GCPs, 1 ICF, 1 zAAP** |
| **GCP Pool** | MVSA | 750 | MVSA | 750 | MVSA | 750 |
| | MVSB | 250 GCP pool = 1000 | MVSB | 250 GCP pool = 1000 | MVSB | 250 GCP pool = 1000 |
| **ICF Pool** | ICF1 | 95 | ICF1 | 95 | ICF1 | 712 |
| | ICF2 | 5 | ICF2 | 5 | ICF2 | 38 |
| | | ICF pool = 100 | MVSA | 750 ICF pool = 850 | MVSA | 750 ICF pool = 1500 |
| **LPAR Weights** | MVSA | 75% | MVSA | 75% | MVSA | 75% |
| | MVSB | 25% | MVSB | 25% | MVSB | 25% |
| **Allowed Resource** | ICF1 | 95% | ICF1 | 22% | ICF1 | 95% |
| | ICF2 | 5% | ICF2 | 1% | ICF2 | 5% |
| | | | MVSA | 176% (100%) | MVSA | 100% |

# Setting zAAP Weights

- **Always start with the zAAP weight (fixed value, with no direct installation control)**
- **Sum up the partitions which will have shared zAAPs**
- **Total share / # of shared zAAPs gives the weight of 1 CP in the ICF pool**
- **Use this value to reset the weight of the others to the same ratio you had before the zAAP**

| 2 SHARED zAAPs in the ICF Pool, and 1 IFL in the ICF Pool | | | | | |
|---|---|---|---|---|---|
| | Weight | Shared zAAP /IFLs | Fair Share | New Weight | Share of zAAP / IFLs |
| ZOS1 | 300 | 2 | | | 75% (1.5 zAAPs) |
| ZOS2 | 300 | 0 | | | 0% |
| ZOS3 | 100 | 1 | | | 25% (0.5 zAAPs) |
| IFL1 | 25 | 1 | 50% | 100 | 50% |
| IFL2 | 25 | 1 | 50% | 100 | 50% |

**zAAP weight = 400**
**shared zAAPs = 2**
**IFL CPs = 1**

**1 CP of weight = 400 / 2 = 200**
**IFL1 = 200 (1 CP share) * 50% (previous share) = 100**
**IFL2 = 200 (1 CP share) * 50% (previous share) = 100**

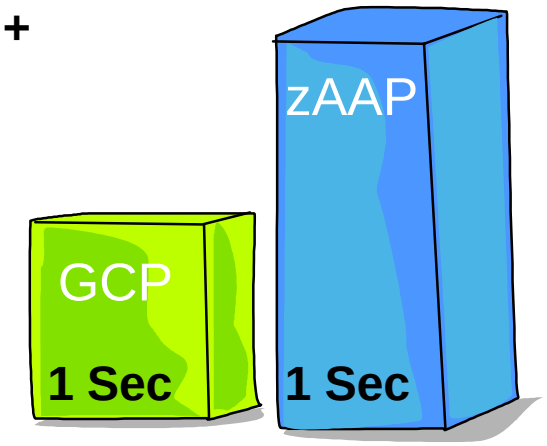
# z890 Considerations

- **z890s have 28 different capacity settings**
- **z890 have 4 PUs characterizable as GCP, IFL, ICF, and now zAAPs**
- **zAAPs on z890 run at full speed of the appropriate n-way**
- **Will need to understand CPU normalization**

Example: zAAP is 2 times the speed of the GCP

**Execution Time = GCP seconds + (zAAP seconds * normalization factor / 256)**

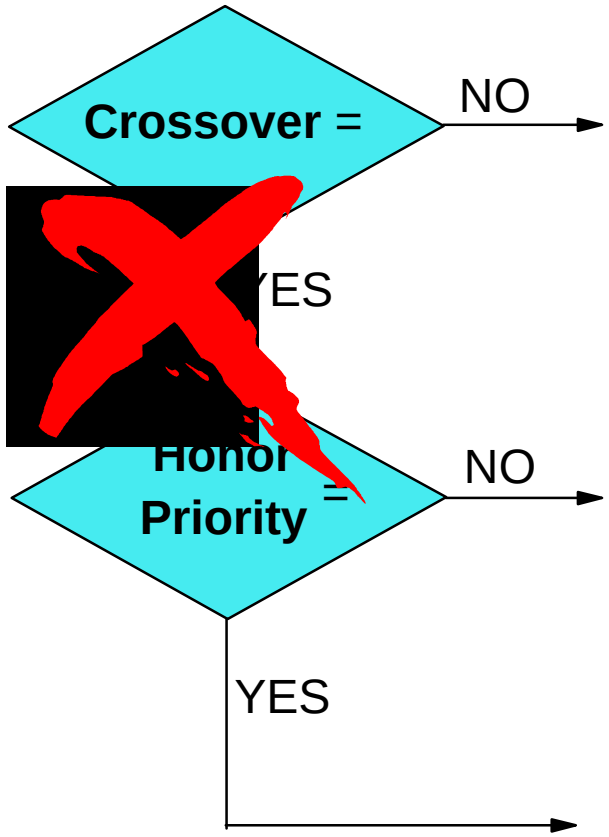
**Execution Time = 1 second + (1 second * (2)) = 3 seconds**




- **Normalization factor used is in RMF 72 subtype 3 record, R723NFFI**
- **If zAAP and GCP are the same speed the normalization factor is 1**

# New SYS1.PARMLIB Options

- IFACROSSOVER = YES | NO
- IFAHONORPRIORITY = YES | NO




No zAAP eligible work can run on a GCP
**Done when software license charges are paramount**
**HonorPriority setting is irrelevant when zAAP are online**
**HonorPriority setting used when last zAAP goes offline**

Have the GCP select zAAP eligible work when the GCP is not busy
**JAVA work runs below non-JAVA discretionary work on GCP**
**JAVA work runs at priority on zAAP**

Have the GCP select GCP work and zAAP eligible work in dispatch priority order
**GCPs and zAAPs will compete to process the JAVA work**
**Default Setting**

GCP - General CP
IFA - zAAP CPs

# zAAP IEAOPTxx parameters change with OA14131

## ★ Why the change?

- **Java eligible work tends to be high priority (e.g. Websphere, CICS)**
- **GCPs tend to outnumber zAAPs (3:1 or 4:1)**
  - With Crossover=YES, GCP tend to take most of the zAAP work hurting the effectiveness of the solution
  - With Crossover=NO, spikes in Java workloads can saturate the zAAP, introducing response time issues when available GCP capacity exists
- **Need to provide a change which allows zAAPs to process all available work, and indicate to GCP when "they need help"**

# IFA IEAOPTxx Planned Changes

| XO | HP | IFA NH | Description |
|---|---|---|---|
| **Y** | **Y** | **Y** | Standard processors run both zAAP eligible and non zAAP eligible work in priority order |
| **Y** | **Y** | N | Standard processors run only non-zAAP eligible work in priority order and run zAAP eligible work as if it has a priority less than non zaap eligible discretionary work |
| **Y** | **N** | **Y** | Standard processors run non zAAP eligible work in priority order and run zAAP eligible work as if it has a priority less than non zaap eligible discretionary work |
| **Y** | **N** | **N** | Standard processors run non zAAP eligible work in priority order and run zAAP eligible work as if it has a priority less than non zaap eligible discretionary work |
| **N** | **Y** | **Y** | Standard processors run both zAAP eligible and non zAAP eligible work in priority order |
| **N** | **N** | **Y** | Standard processors do not run zAAP eligible workloads |
| **N** | **Y** | **N** | Standard processors do not run zAAP eligible workloads |
| **N** | **N** | **N** | Standard processors do not run zAAP eligible workloads |

**XO: Crossover**

**HP: HonorPriority**

**IFA NH: IFA Needs Help (determined by Alternate Wait State Mgmt)**

*** zAAP Processors run all work in priority order**

# RMF Support for zAAPs

- **In RMF and SMF documentation zAAPs are called IFAs**
- **RMF support described in APAR OA05731**
- **RMF Monitor I**
  - CPU Activity, Including Partition Report
  - Workload Activity Report
- **RMF Monitor III**
  - Meaning of CPU UTIL% in Sysinfo and Workflow/Exception report changes: Only regular CPs are included and no IFAs
  - CPC Capacity Report
  - Enclave Report
- **All RMF reports where CPU activity or delay is measured, or CPU activity and delay influences job workflow, the zAAP CPU activity is included**
- **New fields in the RMF product section is added to indicate the presence of zAAPs**
  - SMFxxPRF, bit 4 indicates IFA processors are available

# zAAP CPU Time Reporting

```
              z/OS V1R6                SYSTEM ID SYSD              DATE 09/09/2004
                                     RPT VERSION V1R5 RMF        TIME 18.55.00

CPU  2084   MODEL  314
---CPU---  ONLINE TIME   LPAR BUSY       MVS BUSY      CPU SERIAL  I/O TOTAL
NUM  TYPE  PERCENTAGE    TIME PERC       TIME PERC     NUMBER      INTERRUPT RATE
 0   CP    100.00         61.07           61.30         043A6A       4.04
 1   CP    100.00         77.84           78.20         043A6A       5.51
CP   TOTAL/AVERAGE        69.45           69.75                      9.55
 2   IFA   100.00         24.75           24.86         043A6A
 3   IFA    33.23         13.73           14.64         043A6A
IFA  AVERAGE               22.00           22.31
```

```
                                P A R T I T I O N  D A T A  R E P O R T
MVS PARTITION NAME                    OOSP4          NUMBER OF PHYSICAL PROCESSORS          16
IMAGE CAPACITY                          497                        CP                        14
NUMBER OF CONFIGURED PARTITIONS          30                        ICF                        2
--------- PARTITION DATA ----------------  -- LOGICAL  -- AVERAGE PROCESSOR UTILIZATION PERCENTAGES --
                  ----MSU----  -CAPPING--  PROCESSOR-    LOGICAL PROCESSORS  --- PHYSICAL PROCESSORS ---
NAME       S   WGT  DEF   ACT  DEF   WLM%  NUM   TYPE    EFFECTIVE   TOTAL  LPAR MGMT  EFFECTIVE  TOTAL
OOSP4      A    30    0    69  NO     0.0  2.0   CP        69.07     69.45      0.05       9.87   9.92
OOSPA      A    10    0     3  NO     0.0    2   CP         2.40      2.67      0.04       0.34   0.38
*PHYSICAL*                                                                     4.34                4.34
                                                                             ------     ------ ------
  TOTAL                                                                        5.27      31.07  36.34

OOSP4      A    30                        2   ICF       21.79     22.00      0.14      14.52  14.66
*PHYSICAL*                                                                     2.56                2.56
                                                                             ------     ------ ------
  TOTAL                                                                        2.71      14.52  17.22
```

# zAAP CPU Times - RMF 72 Workload Activity

| TRANSACTIONS | | TRANS.-TIME | | SS.TTT | --DASD I/O-- | | ---SERVICE---- | | --SERVICE TIMES-- | |
|---|---|---|---|---|---|---|---|---|---|---|
| AVG | 1.00 | ACTUAL | | 0 | SSCHRT | 0.1 | IOC | 3 | TCB | 532.9 |
| MPL | 1.00 | EXECUTION | | 0 | RESP | 1.3 | CPU | 11059K | SRB | 0.0 |
| ENDED | 0 | QUEUED | | 0 | CONN | 0.8 | MSO | 1252M | RCT | 0.0 |
| END/S | 0.00 | R/S AFFINITY | | 0 | DISC | 0.3 | SRB | 219 | IIT | 0.0 |
| #SWAPS | 0 | INELIGIBLE | | 0 | Q+PEND | 0.1 | TOT | 1263M | HST | 0.0 |
| EXCTD | 0 | CONVERSION | | 0 | IOSQ | 0.0 | /SEC | 4210K | IFA | 57.6 |
| AVG ENC | 0.00 | STD DEV | | 0 | | | | | APPL% CP | 158.4 |
| REM ENC | 0.00 | | | | | | ABSRPTN | 4210K | APPL% IFACP | 0.0 |
| MS ENC | 0.00 | | | | | | TRX SERV | 4210K | APPL% IFA | 19.2 |

| | RESP TIME | EX | PERF | AVG | --- USING% --- | | ----- | ----- | EXECUTION DELAYS % | |
|---|---|---|---|---|---|---|---|---|---|---|
| SYSTEM | | VEL% | INDX | ADRSP | CPU | IFA | I/O | TOT | CPU | IFA |
| SYSD | N/A | 89.0 | 0.6 | 1.0 | 79.5 | 9.5 | 0.0 | 11.0 | 7.6 | 3.4 |

| | |
|---|---|
| TCB | - Includes GCP seconds and normalized IFA seconds |
| SRB | - Includes GCP seconds and normalized IFA seconds |
| IFA | - Includes only IFA seconds |
| APPL% CP | - Includes only non-IFA (GCP) time |
| APPL% IFACP | - Time spent on a GCP which was IFA eligible (Subset of APPL% CP) |
| APPL% IFA | - Time spent on the IFA |

# zAAP CPU Times - SMF 30 records

- **zAAP CPU time is not included in SMF30CPT**
- **New fields for zAAP timings**

| FIELD | Description |
|---|---|
| SMF30_TIME_ON_IFA | CPU time spent on IFA |
| SMF30_ENCLAVE_TIME_ON_IFA | Enclave time spent on IFA |
| SMF30_DEP_ENCLAVE_TIME_ON_IFA | Dependent enclave time spent on IFA |
| SMF30_TIME_IFA_ON_CP | CPU time spent running IFA eligible work on a GCP (already in SMF30CPT) |
| SMF30_ENCLAVE_TIME_IFA_ON_CP | IFA Enclave time spent on a GCP (already included in SMF30CPT) |
| SMF30_DEP_ENCLAVE_TIME_ON_CP | IFA dependent enclave time spent on a GCP (already in SMF30CPT) |

- **Done to ensure proper billing for new Java workloads**
  - zAAPs are assist processors and have different cost structures, (lower cost, don't carry IBM software charges, and have lower maintenance costs)
  - If zAAP time was in SMF30CPT you would mix CPU seconds with different cost structures
  - Current billing programs do not have to be changed

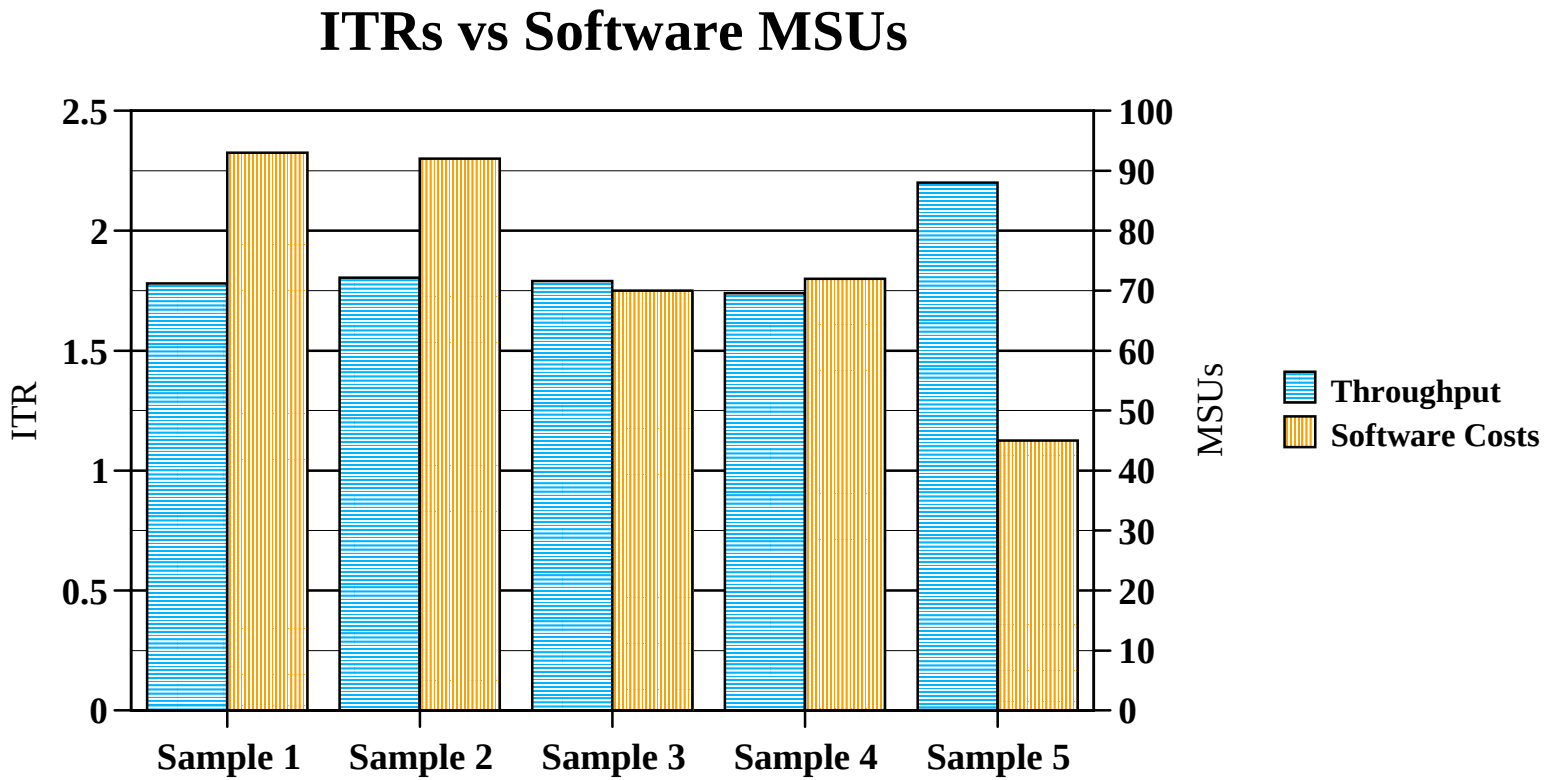
# WSC Experiences

# WSC Experiences

- **Installed z/OS 1.6 as part of the ESP**
- **Ordered and installed 1 zAAP**
  - 2084-315 and 1 zAAP (Eventually added a second zAAP)
- **Series of benchmarks**
  - Ran a Java program (JSP) which calculated Pi to user defined positions (it was used as a soaker program for the zAAP
  - Ran CICS 2.3, with Java transactions which called a DB2 V7 database
  - Ran USS Java workload which called DB2
- **Methodology**
  - Ran the workloads with SDK 1.3 and z/OS 1.5 to get a baseline and create the trace data needed for the estimation tool
  - Ran the workloads on z/OS 1.6 varying the IEAOPTxx parameters for IFACROSSOVER and IFAHONORPRIORITY
  - Measured:
    - Response Time
    - CPU per tran
    - Transaction Count
    - Various RMF Appl% values
    - zAAP busy

# CICS Results

| Test Case | GCPs | zAAPs | JVMs | CrossOver | HonorPriority | Other |
|---|---|---|---|---|---|---|
| Sample 1 | 2 | 0 | 2 | NO | YES | |
| Sample 2 | 2 | 1 | 2 | YES | YES | |
| Sample 3 | 2 | 1 | 2 | NO | YES | |
| Sample 4 | 2 | 2 | 2 | NO | YES | |
| Sample 5 | 2 | 2 | 2 | NO | YES | changed app to reduce EXEC CICS LINKS, and increase business logic |



# WSC Results

- **Importance of switch rate on performance and throughput**
  - **CICS workload had an extremely high switching rate due to the method of calling DB2 and using the result set**
    - **JNI call backs to set variables in the JVM environment**
  - **Highlighted the need to review the switch rate when evaluating applications**
    - **Introduction of a new metric - Microseconds per switch**
    - **Amount of Java execution time per switch**
  - **Updated the jni call back functions to identify additional switch points which were not productive**
    - **May reduce the need for microseconds per switch metric**
  - **High switch rates can influence the accuracy of the estimation tool**
    - **On each switch 2 TIMEUSED macros are used to get the CPU time**
    - **One of the macro invocations is included in the zAAP offloadable time**

# WSC Results

- **Original results have been invalidated by changes introduced via APAR OA14131**
- **See WSC FLASH for more information**
  **http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10432**

# WSC Results

- **Need to review and understand the impact of zAAP CPs on monitor's utilization numbers**

```
SDSF DA SYSD  SYSD     PAG    0 SIO   147 CPU  68/ 66  LINE 1-10 (10)
COMMAND INPUT ===>                                     SCROLL ===> CSR
NP   JOBNAME  SrvClass DP C StepName Workload JobID      CPU%    SIO Server
     KMWDL    BATLOW   F3 L STEP1    BAT_WKL  JOB27402   0.00   0.00 NO
     KMW7L    BATLOW   F3 L STEP1    BAT_WKL  JOB27405   0.00   0.00 NO
     KMW8L    BATLOW   F3 L STEP1    BAT_WKL  JOB27406   0.00   0.00 NO
     KMW9L    BATLOW   F3 L STEP1    BAT_WKL  JOB27407   0.00   0.00 NO
```

  - **System had 2 GCPs and 1 zAAP, system was 66% busy but work was not running.... why**
  - **SDSF incorporated the zAAP CP into CPU% field. So 2GCPs were 100% busy and the zAAP was 0% busy so the CPU% was 200% / 300% = 66%**
  - **New Support: SDSF took APAR PQ93310 to add additional zAAP CPU timing fields**

```
GCP-Time zAAP-Time zACP-Time GCP-Use% zAAP-Use%
    7.17      5.69      0.00    99.42    100.00
```

# WSC Results

- **IFAHONORPRIORITY and IFACROSSOVER parmlib settings are recorded in the field R723MFLG, offset 1**
  - **If Bit 0 is ON, CROSSOVER=YES is set**
  - **If Bit 1 is ON, HONORPRIORITY=YES is set**
- **New CICS support for Garbage Collection**
  - **PQ94293**
  - **Externalizes the MAX_RESETS_TO_GC= parm**
  - **Set in the JVM profile, and allows the setting of a value of something other than 100 uses before CICS issues the SystemGC() call**
- **New SMF Support**
  - **OA09118**
    - **Provides the SRM Coefficients in SMF 30 records to allow Service Units to be used to calculate CPU time**
  - **OA10901**
    - **Provides the zAAP normalization factor, scaled by 256, in the SMF 30 record**

# WSC Results

- **OA09113 - HIPER**
  - **CICS/Java transactions experienced increased response times and reduced throughput when running with zAAP processors as compared to no zAAP environment**
  - **CICS performance statistics indicate higher DISPWAIT time. Delays occurred dispatching CICS tasks on general purpose CPs after POST was issued on zAAP and all CPs were in NO WORK WAITs**
- **IFACrossover=No specified but APPL% IFACP time recorded**
  - **Due to significant local lock contention in application**
  - **Lock processing is not JAVA execution so the work ran on the GCP**
- **Update IEFACTRT exit to add information on zAAP time**
- **Message IEF374I and IEF376I includes normalized IFA time**

**IEF374I STEP/CICS /STOP 2004224.1105 CPU 238MIN 13.05SEC**

  - **CPU times should be the same if the work runs on a system with or without zAAPs**
  - **Can't use messages to determine how much CPU time was spent on the zAAP**

# WSC Results

- **Need to evaluate and update any performance management reports which report utilization**
- **Update capacity planning methods for zAAPs**
- **Update capture ratio calculations to include zAAP times**

$$\text{System CR} = \frac{(\Sigma \text{ Service Class APPL\%}) / \#\text{ Logical CPs} + (\Sigma \text{ Service Class APPL\% IFA}) / \#\text{ Logical IFAs}}{\text{LPAR Busy} + \text{IFA Busy}}$$

$$\text{zAAP CR} = \frac{(\Sigma \text{ Service Class APPL\% IFA}) / \#\text{ Logical IFAs}}{\text{IFA Busy}}$$

$$\text{GCP CR} = \frac{(\Sigma \text{ Service Class APPL\%}) / \#\text{ Logical CPs}}{\text{LPAR Busy}}$$

# zAAP Summary

- **zAAP …an industry first**
  - Only specialized processing units for Java Code today
  - Supported by IBM Middleware such as WebSphere, CICS, DB2…
  - Helps reduce demands on general purpose processors – make them available for other work
- **zAAPs for e-business Integration and Infrastructure Simplification**
  - Integrate Java technology-based applications with mission-critical data
  - Helps reduce infrastructure complexity for multi-tier applications
- **zAAPs Provide Investment Flexibility**
  - Extend the value of existing zSeries investments and lowers total cost of ownership
  - Cost-effective, specialized Java execution environment
  - Low Total Cost of Acquisition ($125K USD per zAAP)
  - Helps reduce Total Cost of Ownership (software and maintenance Savings)