UNIVERSITÉ DU MAINFRAME

Business Integration avec Java et zAAP

Isabelle Ringwald
IT specialist zSeries
iringwald@fr.ibm.com

Paris, 3-4 mai 2006

Agenda

- **Qu'est-ce que le zAAP ?**
- TCO
- Capacity planning
- Mise en oeuvre et suivi des performances

IBM eServer **z890/z990** et IBM System **z9 109** – zSeries Application Assist Processor (**zAAP**) –

2004 : Nouveau Processeur Spécialisé

Dédié exclusivement à l'exécution de Java sous z/OS

Sans facturation logiciel et à coût hardware réduit

- Disponible uniquement sur les serveurs **z890**, **z990** et **z9**
- Conçu pour l'exécution du code Java z/OS
- Configuration commerciale : pas plus d'1 zAAP par CP standard dans une machine
- Prix attractif : 125K€ par moteur quelque soit le serveur (Mainframe Charter)
- Coûts Software non affectés par le zAAP

➔ Objectif : baisser le TCO pour capter les nouvelles applications sur les zSeries

Exploitation du zAAP

- Disponible uniquement sur les serveurs **z890**, **z990** et **z9**
- Fonctionne dans les **z890** à pleine capacité
- Ne peut pas fonctionner tout seul
- Configuration technique : une partition peut être définie avec plus ou moins de zAAPs que de CPs standards
  ➔ définitions PR/SM
- **Les applications ne nécessitent pas d'adaptation pour supporter le zAAP**
- Prérequis :
  - **z/OS 1.6** (ou z/OS.e 1.6)
  - La JVM fourni par IBM SDK for z/OS, Java2 Technology Edition V1.4 (avec PTF pour APAR PQ86689)
- **Sous-systèmes et Applications exploitant le zAAP**
  - **WAS 5.1**
  - CICS® /TS 2.3
  - DB2 V8
  - IMS™ V8
  - WebSphere WBI for z/OS
- Support de la fonction CUoD sur z890, z990, z9, des fonctions CIU, CBU sur z9 uniquement
- 3 paramètres système : un dans la JVM, deux dans z/OS

En quoi les zAAPs diffèrent des CPs standards

- **Quelques limitations des zAAPs**
  - ➢zAAPs ne peuvent pas être IPLed
  - ➢zAAPs exécutent uniquement des instructions z/Architecture™
  - ➢zAAPs ne supportent pas toutes les commandes de contrôle opérateur
    - ➢No: PSW Restart, LOAD or LOAD derivates (load from file, CDROM, Server)
  - ➢zAAPs ne répond pas aux demandes SIGP (sauf si émis par un z/OS qui supporte les zAAPs)
  - ➢D'autres différences d'architecture dans les futures implémentations
    - e.g., Java specific performance enhancements
- **Différences de design pour les zAAPs:**
  - ➢No I/O interrupts
  - ➢No Clock Comparator interrupts
  - ➢No affinity scheduling

Agenda

- Qu'est-ce que le zAAP ?
- **TCO**
- Capacity planning
- Mise en oeuvre et suivi des performances

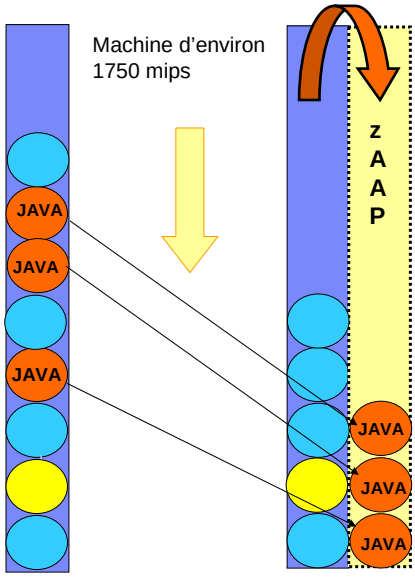
Exemple : z990-304 ou z9-703

Considérons un Batch ou une Application Websphere qui nécessite 800 mips sur un zSeries

CPU : chargée à environ 80%
(soit environ 1400 Mips utilisés)

Appli Websphere ou Batch : 47% de la CPU
(soit environ 800 Mips)

➔**70% de code Java**
(soit environ 560 Mips elligible au zAAP)



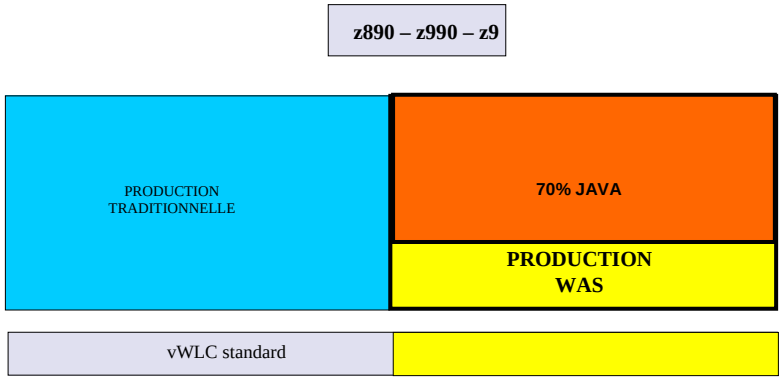
CPU : chargée à environ 50%
(soit environ 840 Mips utilisés)
➔ Machine d'environ 1000 Mips

➔560 Mips disponibles pour de la charge additionnelle

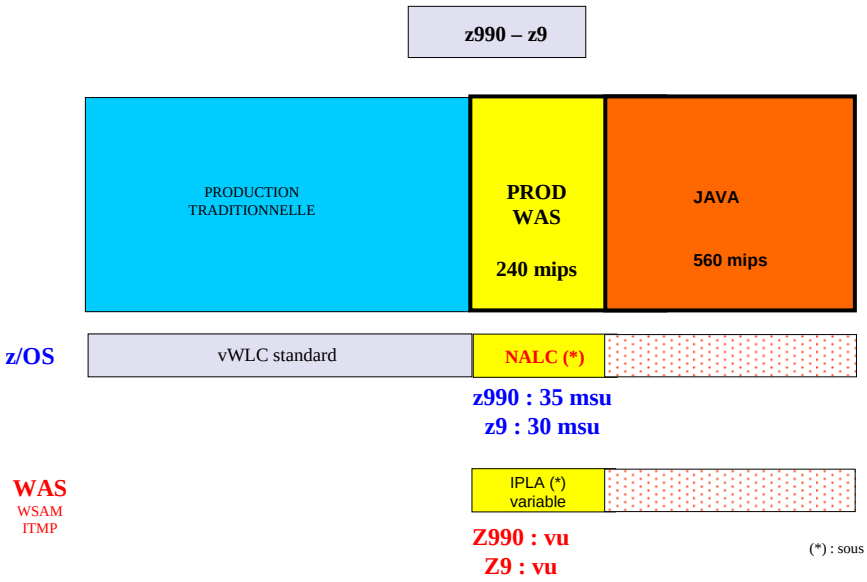
➔240 Mips des applis Websphere ou Batch restent sur CPs standards

Dans cet exemple, avec un zAAP, on peut réduire le besoin en CP standard de 560 mips

Exemple Coût Software sans zAAP



Exemple : Coût Software avec zAAP



(*) : sous réserve accord IBM

IBM eServer **z890/z990** et IBM System **z9 109**
zAAP

CONCLUSION :

Faites l'Etude zAAP !

- Faites faire une évaluation du code Java par les équipes techniques IBM
- Pensez à l'offre Interim zAAP si vous n'avez pas encore les pre-requis
- Calculer le TCO : **z990** et **z9**

➔ N'oubliez pas la différenciation zSeries !

Agenda

- Qu'est-ce que le zAAP ?
- TCO
- **Capacity planning**
- Mise en oeuvre et suivi des performances

Comment évaluer l'apport du zAAP pour mes applications ?

- **Information sur le web**
  - http://www-1.ibm.com/servers/eserver/zseries/zaap/gettingstarted/
  - White Paper : z/OS Performance - Capacity Planning Considerations for zAAP
    - ✓ Décrit le zAAP Projection Tool
    - ✓ Décrit les prises de mesures du prototype
    - ✓ Décrit la méthodologie du Capacity Planning
- **Redbooks**
  - SG24-6386 zSeries Application Assist Processor (zAAP) Implementation

- **zAAP Projection Tool**
  - Permet d'évaluer la charge Java des applications tournant déjà sous z/OS

- **Size 390**
  - Fournit une assistance spécilisée pour la méthodologie décrite dans le White
  - Aide à l'évaluation du sizing lors de consolidation de'applications Java d'environnements distribués vers les zSeries et les zAAPS

- **Equipe spécialisée projets Websphere zSeries :**
  - Architectes et spécialistes : IBM Software et IBM System & Technology Groups, IBM Global Service, PSSC Montpellier

Le zAAP Projection Tool

- **"zAAP Projection Tool for Java 2 Technology Edition"**
  - Disponible avec le SDK 1.3.1 SR 24 et inclus dans le SDK 1.4.x
    - Faire tourner dans un environnement de test de préférence
    - Récolte les informations sur le % de code Java dans les applications qui peuvent s'exécuter dans un zAAP
    - Les données servent d'input à un Workbook Excel qui donne une synthèse des résultats
    - Très utile pour prévoir le nombre de zAAPs nécessaires à une configuration optimale

- **Step 1 – Récolte des données**

```
IFA Projection data for system id=<SYSE.50594238> Starting at: 23:27:33 - Current address space CPU: 0.008068 sec.                    lookup 2
<SYSE.50594238> Interval at: 23:32:33 Switches To/From IFA: 3717251 Java IFA: 19.86 sec. Java Standard CPU 20.37 sec. Interval address space CPU: 50.08 sec.
<SYSE.50594238> Interval at: 23:37:33 Switches To/From IFA: 3903114 Java IFA: 20.85 sec. Java Standard CPU 21.39 sec. Interval address space CPU: 52.58 sec.
<SYSE.50594238> Interval at: 23:42:33 Switches To/From IFA: 4176332 Java IFA: 22.31 sec. Java Standard CPU 22.89 sec. Interval address space CPU: 56.26 sec.
<SYSE.50594238> Interval at: 23:47:33 Switches To/From IFA: 3842225 Java IFA: 20.53 sec. Java Standard CPU 21.06 sec. Interval address space CPU: 51.76 sec.
<SYSE.50594238> Interval at: 23:52:33 Switches To/From IFA: 3573269 Java IFA: 19.09 sec. Java Standard CPU 19.59 sec. Interval address space CPU: 48.14 sec.
<SYSE.50594238> Interval at: 23:57:33 Switches To/From IFA: 3859131 Java IFA: 20.62 sec. Java Standard CPU 21.16 sec. Interval address space CPU: 51.99 sec.
<SYSE.50594238> Interval at: 00:02:33 Switches To/From IFA: 3743357 Java IFA: 20 sec. Java Standard CPU 20.53 sec. Interval address space CPU: 50.43 sec.
<SYSE.50594238> Interval at: 00:07:33 Switches To/From IFA: 3668490 Java IFA: 19.6 sec. Java Standard CPU 20.12 sec. Interval address space CPU: 49.42 sec.
<SYSE.50594238> Interval at: 00:12:33 Switches To/From IFA: 4035339 Java IFA: 21.56 sec. Java Standard CPU 22.13 sec. Interval address space CPU: 54.36 sec.
```


Le zAAP Projection Tool

- **Step 2 – Analyse des données à partir du Workbook**

A1 fx Time at start of interval

	A	E	F	G	H	J	L	M	N
1	Time at start of interval	zAAP eligible seconds	Java not eligible seconds	Space CPU seconds	%Time zAAP eligible	zAAP% engine eligible	Appl% engine	zAAP% w/capt ratio	ZAAPs w/wait
2		Go to Inventory		Service Class				90%	80%
3	16:04:33	466	35	907	51%	12%	23%	13%	16%
5	16:04:33	30	1	36	85%	10%	12%	11%	14%
6	16:09:34	26	3	36	72%	9%	12%	9%	12%
7	16:14:35	27	3	37	72%	9%	12%	10%	12%
8	16:19:36	18	3	27	64%	6%	9%	6%	8%
9	16:24:37	18	3	28	64%	6%	9%	7%	8%
10	16:29:41	22	2	32	69%	7%	11%	8%	10%
11	16:34:42	52	3	76	68%	17%	25%	19%	24%
12	16:39:43	70	5	105	67%	23%	35%	26%	32%
13	16:44:44	76	5	111	68%	25%	37%	28%	35%
14	16:49:45	5	0	122	4%	2%	40%	2%	2%
15	16:54:49	0	0	87	0%	0%	26%	0%	0%
16	17:00:17	51	3	103	49%	17%	34%	19%	23%

Agenda

- Qu'est-ce que le zAAP ?
- TCO
- Capacity planning
- **Mise en oeuvre et suivi des performances**

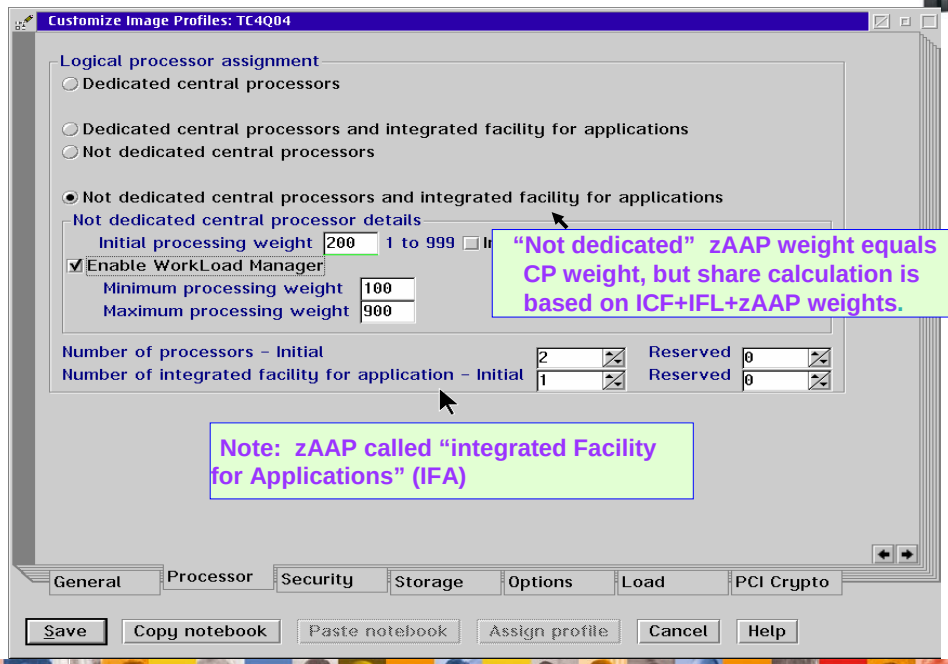
zAAP : z/OS pre-requisites

- **z/OS 1.6**

- **Enhanced RMF™ Reports (to include zAAP Usage) ➔ APAR OA05731**
  - Standard Processors: Reporting as today
  - Timing enhancements for zAAPs

- **New SDSF display columns : ➔ APAR PQ93310 et PK066116**
  - L'APAR PQ93310 apporte les nouvelles colonnes SDSF suivantes (à définir) :
    - * GCPTIME
    - * ZAAPTIME
    - * ZAAPCPTM
    - * GCPUSE
    - * ZAAPUSE
  - L'APAR PK066116 apporte la nouvelle colonne SDSF suivante (à définir) :
    - * SZAAP

PR/SM LPAR Configuration Panel



Gestion du poids : z990 (pas de zAAP)

- **Exemple étudié**
  - Ajout d'un ICF **dédiée** et d'un IFL partage entre deux LPAR LINUX

Nom LPAR	Poids	%SHARE = (Poids-LPAR-zOS) / Poids Total zOS	Capacité PP = %SHARE*#PP Shared du Pool
ZOS1	500	50.00%	5
ZOS2	200	20.00%	2
ZOS3	300	30.00%	3
TOTAL-ZOS	**1000**	**100.00%**	**10**
Nom LPAR ICF/IFL	**Poids**	**%SHARE = (Poids-LPAR-ICF/IFL) / Poids Total ICF/IFL**	**Capacité PP = %SHARE*#PP Shared du Pool**
ICF1	DED		1
LINUX1	50	50.00%	0.5
LINUX2	50	50.00%	0.5
TOTAL-ICF/IFL	**100**	**100.00%**	**1**

Le Pool ICF/IFL a
1 PP dédié
1 PP Shared

Gestion du poids : z990 **Ajout zAAP**

- **Exemple étudié :**
  - Configuration de départ : 10 CPs standards, 1ICF dédié, 1 IFL partagé
  - Ajout d'1 zAAP partagé par zOS2 et zOS3
  - Le Pool ICF/IFL/**zAAP** est constitué d'un moteur DEDIEE (ICF) et de **2 moteurs PARTAGES (#PP shared du pool)**
  - **Les deux moteurs sont MAINTENANT partagés entre :**
    - **LINUX1 / LINUX2**
    - **zOS2 / zOS3**
  - **Le poids TOTAL (ICF/IFL/zAAP)**
    - **Devient 600**
    - **LINUX1 et LINUX2 perdent du poids !**
    - **Les zAAPs ont hérités du poids des LPAR zOS1 et zOS2**
  - **On ne peut pas changer les poids des zAAPs**
  - **Il faut revoir ceux des IFLs (LINUX)**

Nom LPAR	Poids	%SHARE = (Poids-LPAR-zOS) / Poids Total zOS	Capacité PP = %SHARE*#PP Shared du Pool
ZOS1	500	50,00%	5
ZOS2	200	20,00%	2
ZOS3	300	30,00%	3
TOTAL-ZOS	**1000**	**100,00%**	**10**
Nom LPAR ICF/IFL/zAAP	**Poids**	**%SHARE = (Poids-LPAR-ICF/IFL) / Poids Total ICF/IFL**	**Capacité PP = %SHARE*#PP Shared du Pool**
ICF1	DED		1
LINUX1	*50*	*8,33%*	*0,17*
LINUX2	*50*	*8,33%*	*0,17*
zOS2	200	33,33%	0,67
zOS3	300	50,00%	1
TOTAL-ICF/IFL:zAAP	600	100,00%	2

zAAP – Gestion du poids z990 – Modifications

- **Exemple étudié**
  - On veut revenir à 50% des moteurs pour les LPAR LINUX
  - Le poids total zAAP est de 500, il y a 1 PP zAAP, donc UN PP doit avoir un poids de 500
  - Pour avoir la moitié d'un PP, les LINUX doivent avoir un poids de 250 (il y a 1 IFL)

Nom LPAR	Poids	%SHARE = (Poids-LPAR-zOS) / Poids Total zOS	Capacité PP = %SHARE*#PP Shared du Pool
ZOS1	500	50,00%	5
ZOS2	200	20,00%	2
ZOS3	300	30,00%	3
TOTAL-ZOS	**1000**	**100,00%**	**10**
Nom LPAR ICF/IFL/zAAP	**Poids**	**%SHARE = (Poids-LPAR-ICF/IFL) / Poids Total ICF/IFL**	**Capacité PP = %SHARE*#PP Shared du Pool**
ICF1	DED		1
LINUX1	*50*	*8,33%*	*0,17*
LINUX2	*50*	*8,33%*	*0,17*
zOS2	200	33,33%	0,67
zOS3	300	50,00%	1
TOTAL-ICF/IFL:zAAP	600	100,00%	2

Nom LPAR	Poids	%SHARE = (Poids-LPAR-zOS) / Poids Total zOS	Capacité PP = %SHARE*#PP Shared du Pool
ZOS1	500	50,00%	5
ZOS2	200	20,00%	2
ZOS3	300	30,00%	3
TOTAL-ZOS	**1000**	**100,00%**	**10**
Nom LPAR ICF/IFL/zAAP	**Poids**	**%SHARE = (Poids-LPAR-ICF/IFL) / Poids Total ICF/IFL**	**Capacité PP = %SHARE*#PP Shared du Pool**
ICF1	DED		1
LINUX1	*250*	*25,00%*	*0,50*
LINUX2	*250*	*25,00%*	*0,50*
zOS2	200	20,00%	0,40
zOS3	300	30,00%	0,6
TOTAL-ICF/IFL:zAAP	1000	100,00%	2

Gestion des poids : z9

Moteurs spécialisés : pools séparés

Exemple: z9-109 10 CPs, 1 ICF, 2 IFLs, 3 zAAPs

- **LPAR Share = Pool PUs x (LPAR Pool Weight)/(Total Pool Weight)**
  - Can't exceed number of Online Logical Processors dispatched from the pool
  - z/OS LPAR – Separate "Initial" weights for CPs and zAAPs
- **Pool PUs (Physical) – CP =10, zAAP = 3, IFL = 2, ICF =1**
- **Total Pool Weights – CP = 1000, zAAP = 200, IFL= 400, ICF= 100**

LPAR Name	LPAR Weight	Shared Logical PUs On				PU Share			
		CP	zAAP	IFL	ICF	CP	zAAP	IFL	ICF
ZOS1	250c / 100z	10	2	NA	NA	2.5	1.5	NA	NA
ZOS2	750c / 100z	10	3	NA	NA	7.5	1.5	NA	NA
CF1	50 - ICF	0	NA	NA	1	0	NA	NA	.5
CF2	50 - ICF	0	NA	NA	1	0	NA	NA	.5
ZVM1	100 - IFL	0	NA	2	NA	0	NA	.5	NA
LINUX1	300 - IFL	0	NA	2	NA	0	NA	1.5	NA
Pool Weight >		1000	200	400	100				
Total PUs (Physical) >						10	3	2	1


CP


zAAP


IFL


ICF

- **No interaction among specialty engine weights**

Enabling the zAAPs : JVM parm

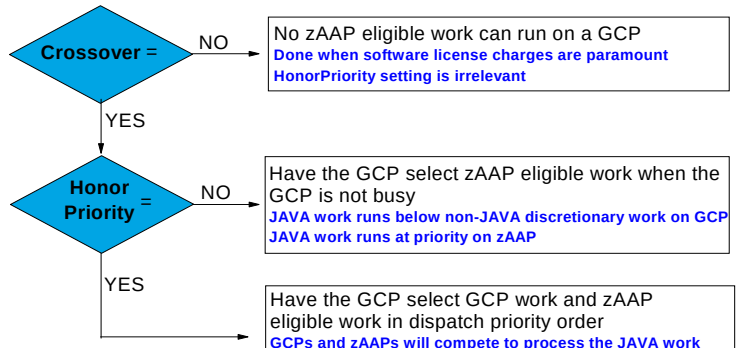
- **Enable zAAP processing at the JVM level through use of the -Xifa: options**

- 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 DFHJVMERR

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

zAAP Configuration Execution Options : original sys1.parmlib Option

- New SYS1.PARMLIB Options in IEAOPTxx
  - IFACROSSOVER = YES | NO
  - IFAHONORPRIORITY = YES | NO



GCP - General CP
IFA - zAAP CPs

zAAP Configuration execution Options

- **zAAPs are Configured via the Normal PR/SM™ Logical Partition Image Profile**

- **Java Application code can be executed under different options set in IEAOPTxx**
  - *Option 1 - **IFACrossover** = Yes + **IFAHONOR_PRIORITY** = Yes ➔ Java on CP by Priority*
  - *Option 2 - **IFACrossover** = Yes + **IFAHONOR_PRIORITY** = No ➔ Java Discretionary Crossover on CP*
  - *Option 3 - **IFACrossover** = No + **IFACrossover** = No ➔ No Java Crossover = no Java on CP*
  - *The selected switching option can be dynamically changed by a SET OPT command*

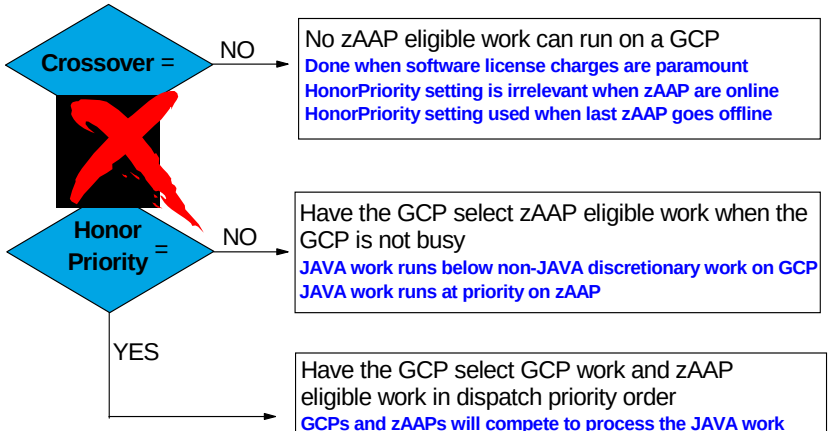
- **Enhanced RMF™ Reports (to include zAAP Usage)** **APAR OA05731**
  - Standard Processors: Reporting as today
  - Timing enhancements for zAAPs
- **Enhanced SMF Records (to include zAAP Usage)**
  - Type 30 & Type 72 (New fields for zAAP time and zAAP eligible on a CP)

zAAP Configuration : new sys1.parmlib Option with APAR 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"**

- **New SYS1.PARMLIB Options**
  - IFACROSSOVER = YES | NO
  - IFAHONORPRIORITY = YES | NO
  - independent parameters



GCP - General CP
IFA - zAAP CPs

zAAP Operator Interface

- **D M=CPU**

```
IEE174I 17.43.46 DISPLAY M
   PROCESSOR STATUS
     ID  CPU                 SERIA
   00  +                   12345A2084
   01  +                   12345A2084
   02  +                   12345A2084
   03  -                   12345A2084
   04  -                   12345A2084
   05  -                   12345A2084
   06  +A                  12345A2084
   07  -A                  12345A2084
   08  NA                  12345A2084


 . . .
 . . .
 A         ASSIST PROCESSOR
```

- **CF CPU(*nn*),OFFLINE | ONLINE**
- **zAAPs are not WLM (IRD) managed, so there is no *+AW* or *–AW* status**
- **SDSF DA reflects zAAP usage**
  - ➔ **DA panel : address space service time on CP, 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

RMF™ Reporting

- **RMF supports zAAP processors by extending the**
  - Postprocessor *CPU activity report*
  - Postprocessor *Workload report*
  - Monitor III *Enclave report (pop-up panel for IFA Using and Delay samples)*
- **The Internals**
  - Distinguishes between standard CP and zAAP processors where necessary
  - Collects and reports about *zAAP service times*
  - Collects and reports about *zAAP using and delay states* for service and report class periods
- **zAAP support is shipped as SPE APAR OA05371**

zAAP Workload Reporting Samples

The Resource Consumption Section of the WLMGL report

```
TRANSACTIONS    TRANS.-TIME  HHH.MM.SS.TTT   --DASD I/O--   ----SERVICE----   --SERVICE TIMES--   PAGE-IN RATES    ----STORAGE----
AVG      4.42   ACTUAL              8.142   SSCHRT   6.5   IOC      56040    TCB         271.3   SINGLE     0.0   AVG      298.08
MPL      4.39   EXECUTION           8.142   RESP    53.6   CPU       1680K   SRB           9.2   BLOCK      0.0   TOTAL   1309.04
ENDED    1879   QUEUED                 0   CONN    20.0   MSO       2938K   RCT           4.4   SHARED     0.0   CENTRAL 1309.04
END/SEC  1.04   R/S AFFINITY           0   DISC     3.2   SRB      56695    IIT           2.0   HSP        0.0   EXPAND     0.00
#SWAPS   3154   INELIGIBLE             0   Q+PEND  25.7   TOT       4731K   HST           3.4   HSP MISS   0.0
EXCTD       0   CONVERSION             0   IOSQ     4.7   /SEC      2626    IFA          20.1   EXP SNGL   0.0   SHARED     0.00
AVG ENC  0.32   STD DEV             8.431                                   APPL% CP     15.0   EXP BLK    0.0
REM ENC  0.12                                             ABSRPTN    598    APPL% IFACP   0.2   EXP SHR    0.0
MS ENC   0.01                                             TRX SERV   594    APPL% IFA     1.1
```

IFA	IFA Service Time (in seconds)
APPL% CP	% of CPU time used by transactions running on regular CPs
APPL% IFACP	% of CPU time used by IFA transactions executed on regular CPs
APPL% IFA	% of CPU time on IFA processors used by IFA transactions

Note: If no IFAs/zAAPs configured, N/A is shown for the new fields.

zAAP CPU Activity Report Samples

```
CPU  2084   MODEL  316

---CPU---  ONLINE TIME   LPAR BUSY      MVS BUSY      CPU SERIAL   I/O TOTAL          % I/O INTERRUPTS
NUM  TYPE  PERCENTAGE    TIME PERC      TIME PERC     NUMBER       INTERRUPT RATE     HANDLED VIA TPI

 0   CP    100.00        69.41          69.41         011511       58.67               0.00
 1   CP    100.00        70.75          70.75         111511       233.6               0.00
 2   CP    100.00        68.40          68.40         211511       254.2               0.00
 3   CP    100.00        63.64          63.64         311511       63.49               0.00
 4   CP    100.00        67.74          67.74         411511       1380                0.01
CP TOTAL/AVERAGE         67.99          67.99                      1990                0.01

 8   IFA   100.00        39.41          39.41         811511
 9   IFA   100.00        40.75          40.75         911511
IFA AVERAGE              40.08          40.08
```

NEW

- A new TYPE column indicates whether the processor belongs to the pool of regular CPs or IFAs
- The last two columns are only available for regular CPs
- A TOTAL/AVERAGE line is printed per pool

RMF Partition Data Report Sample

```
MVS PARTITION NAME                    LP1                    NUMBER OF PHYSICAL PROCESSORS          16
IMAGE CAPACITY                        167                                CP                          8
NUMBER OF CONFIGURED PARTITIONS         6                                ICF                         8
WAIT COMPLETION                        NO
DISPATCH INTERVAL                   DYNAMIC

--------- PARTITION DATA -----------------  -- LOGICAL PARTITION PROCESSOR DATA --  -- AVERAGE PROCESSOR UTILIZA
                  ----MSU----  -CAPPING--  PROCESSOR-  ----DISPATCH TIME DATA----  LOGICAL PROCESSORS  --- PHYS
NAME       S   WGT  DEF   ACT  DEF   WLM%  NUM   TYPE   EFFECTIVE       TOTAL      EFFECTIVE    TOTAL  LPAR MGM
LP1        A    50    0   167  NO     0.0    5   CP    00.29.26.356  00.29.27.505      99.94    100.0      0.02
LP2        A    50    0    33  NO     0.0    1   CP    00.05.53.443  00.05.53.501      99.98    100.0      0.00
LP4        A    50    0   268  NO     0.0    8   CP    00.47.08.000  00.47.08.008      100.0    100.0      0.00
*PHYSICAL*                                             00.00.00.237                                          0.00
                                                       ------------  ------------                          ------
  TOTAL                                                01.22.27.801  01.22.29.251                          0.03

ICF2       A    75                         8   ICF    00.04.53.443  00.04.53.501      99.98    100.0      0.00
IFL4       A    25                         3   ICF    00.24.08.000  00.24.08.008      100.0    100.0      0.00
LP1        A    50                         6   ICF    00.09.26.356  00.09.27.505      99.94    100.0      0.02
*PHYSICAL*                                             00.00.00.237                                          0.00
                                                       ------------  ------------                          ------
  TOTAL                                                01.22.27.801  01.22.29.251                          0.03
```

Partition Name & Weight match indicates IFA(s)

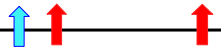
The ICF block contains logical processors of type ICF,IFL,IFA

zAAP Workload Reporting Samples

The WLMGL Goal and Actuals Section

```
RESPONSE TIME GOAL:   00.00.01.000 AVG

ACTUALS:  RESPONSE    EX    PERF   AVG    ----USING%----  ---------- EXECUTION DELAYS % --------- ---DLY%-- -CRYPTO%- ---CNT%--    %
            TIME      VEL%  INDX  ADRSP    CPU  IFA  I/O   TOT  CPU  IFA  I/O  AUX  AUX SWIN      UNKN IDLE  USG  DLY  USG  DLY QUIE
        HH.MM.SS.TTT                                                          VIO PRIV
*ALL    00.00.01.854  31.8   1.1    5.7    3.6  2.2  2.6  13.4  8.5  4.3  0.3  0.2  0.1  0.1      58.1 22.7  1.1  3.1  0.0  0.0  0.0
LP1     00.00.01.999  30.5   1.3    2.1    3.5  2.1  2.3  13.5  9.0  4.2  0.2  0.1  0.0  0.1      60.3 20.5  0.2  1.1  0.0  0.0  0.0
LP2     00.00.01.001  49.4   1.0    1.9    3.3  2.0  3.1   6.8  1.8  4.3  0.3  0.1  0.0  0.0      63.1 24.0  1.4  4.1  0.0  0.0  0.0
LP4     00.00.01.003  24.1   1.0    1.8    3.9  2.3  2.3  10.1 14.8  4.3  0.3  0.1  0.4  0.1      50.3 23.7  0.3  0.3  0.0  0.0  0.0
```



New Using and Delay States

IFA Using	**Work is found executing on an IFA**
IFA Delay	**Work is delayed for an IFA**
IFA on CP Using	**Contained in either CPU using samples or IFA using samples based on the dispatching priorities on selecting work from IFA work queue** **(for work is eligible to run on IFAs but executed on regular CPs)**

zAAP zSeries Application Assist Processor

Questions ?