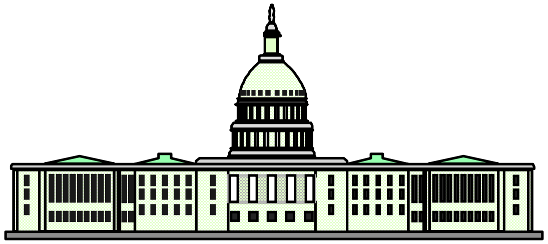
Planning for zAAPs

Walt Caprice
Washington Systems Center
IBM Corp.

zSTSU 2004
Session Z5020



Agenda

- What is a zAAP and how does it work
- How can I do capacity planning for zAAP
- What tuning options are available
  - How do they effect performance
  - How can they effect the capacity plan
- How do I know if an application is a good candidate

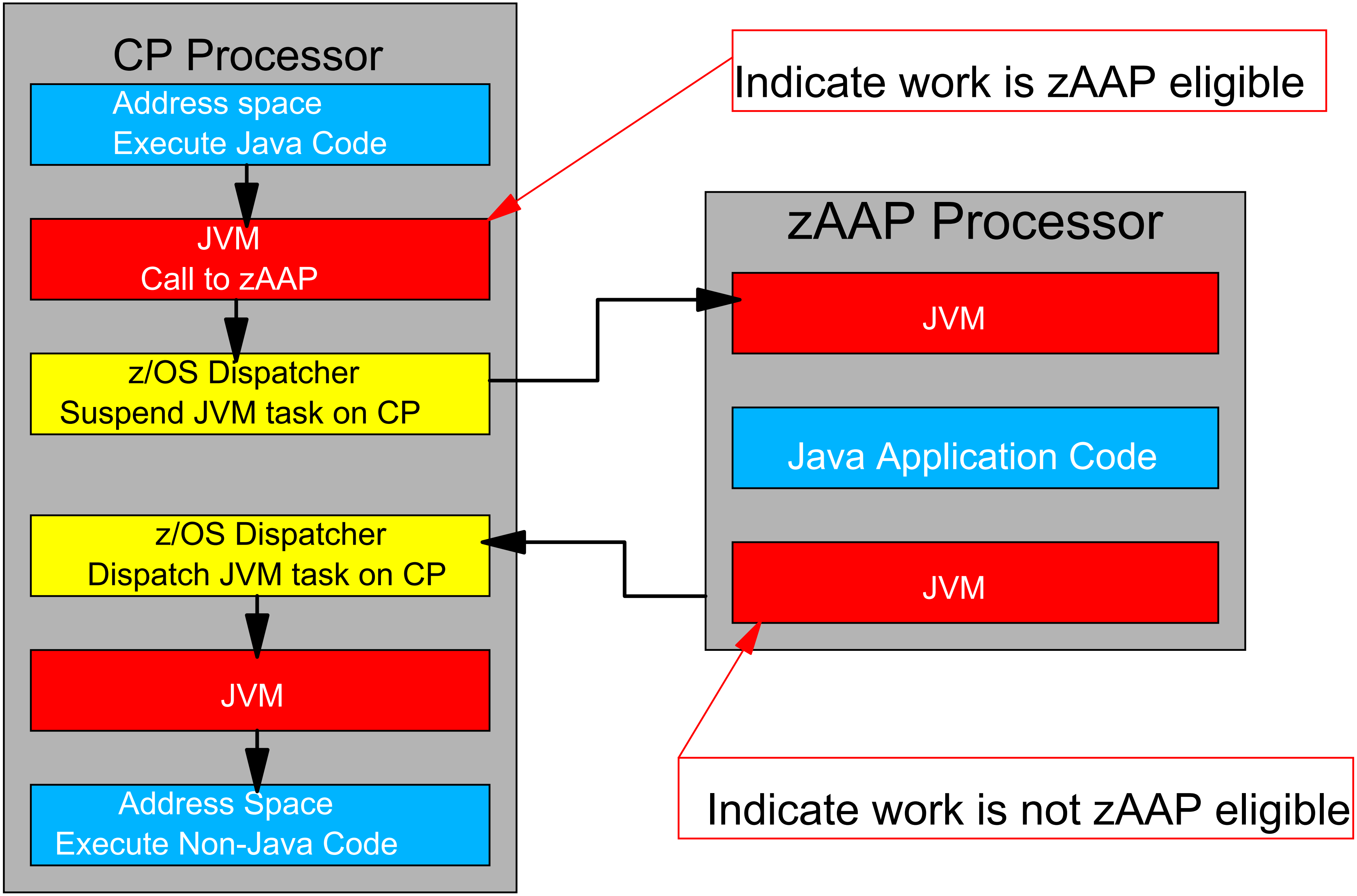
What is a zAAP?

- New processor type dedicated exclusively for Java workloads under z/OS
  - z/OS 1.6 required (available 9/04)
  - SDK 1.4 required
  - Enabled by innovative zSeries PR/SM virtualization
  - CPs and zAAPs in the same z/OS LPAR
  - Only Java work directed to zAAPs by z/OS 1.6
  - z990/z890 and future zSeries servers only
  - No IBM software charges on zAAP capacity

zAAP Characteristics

- Can run ONLY Java code using SDK 1.4 and above
- Users can manage the use of CPs, Java code runs only on:
  - A CP
  - A zAAP
  - Both
- zAAPs operate at:
  - Same speed as general purpose CPs on z990s
  - Potentially faster then general purpose CPs on z890s
- Do not affect overall MSU rating of CEC or LPAR
- Subsystems exploiting zAAPs include:
  - WAS 5.1
  - CICS/TS 2.3
  - DB2 V8
  - IMS V8
  - WebSphere WBI for z/OS
- Transparent to all IBM and ISV Java programming executing on JVM 1.4 and above

How Does a zAAP Work



Update Capacity Planning for zAAP Usage

- Prior to the zAAP, CPU capacity planning for z/OS dealt with only one processor type
  - RMF reported processor utilization as well as CPU consumption by service class

- We now have two types of processors which are used by z/OS, how much do I need and which type?
  - RMF in Version 1.6 of z/OS will report:
    - General purpose processor utilization as well as CPU consumption by service class
    - zAAP utilization as well as consumption by service class
    - CPU time run on general purpose CP but eligible to run on a zAAP

RMF Reports

```
                                                        C P U  A C T I V I T Y

          z/OS V1R6                  SYSTEM ID SYSD              DATE 07/24/2004
                                     RPT VERSION V1R5 RMF        TIME 22.45.00
CPU  2084   MODEL  315
---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        92.99           93.71         043A6A      48.37
 1   CP    100.00        93.04           93.75         043A6A      49.17
CP   TOTAL/AVERAGE       93.02           93.73                     97.53
 2   IFA   100.00        14.63           15.07         043A6A
IFA  AVERAGE             14.63           15.07
```

RMF support for zAAP

RMF Reports

```
REPORT BY:  POLICY=WLMPOL                          REPORT CLASS=RIYOT123
                                                   DESCRIPTION =Report class for R2.3 region

TRANSACTIONS     TRANS.-TIME  HHH.MM.SS.TTT   --DASD I/O--   ---SERVICE----   --SERVICE TIMES--
AVG       1.00   ACTUAL                   0   SSCHRT    0.1  IOC          2   TCB         562.7
MPL       1.00   EXECUTION                0   RESP      1.3  CPU     11677K   SRB           0.0
ENDED        0   QUEUED                   0   CONN      0.7  MSO      1331M   RCT           0.0
END/S     0.00   R/S AFFINITY             0   DISC      0.5  SRB        123   IIT           0.0
#SWAPS       0   INELIGIBLE               0   Q+PEND    0.1  TOT      1343M   HST           0.0
EXCTD        0   CONVERSION               0   IOSQ      0.0  /SEC     4476K   IFA          18.3
AVG ENC   0.00   STD DEV                  0                                   APPL% CP    181.5
REM ENC   0.00                                               ABSRPTN  4476K   APPL% IFACP   0.2
MS ENC    0.00                                               TRX SERV 4476K   APPL% IFA     6.1
```

% of time on zAAP

% General CP Busy eligible for zAAP
subset of APPL% CP

% General CP Busy including Java Time

Update Capacity Planning Prior to z/OS 1.6

- Evaluate applicability of a zAAP in the future
- No RMF information available yet
- Need to know how much time workload is eligible to run on a zAAP

New Estimation Process

- Amount of time eligible to execute on zAAP
  - An instrumented version of the 1.3.1 JVM is available to capture this information at the address space level
    - Available at https://www6.software.ibm.com/dl/zosjava2/zosjava2-p
    - Whitepaper available on Techdocs contain complete installation directions for WebSphere for z/OS Version 5 available at http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100431
  - Estimation data written to STDERR at regular intervals

New Estimation Process

- Download the instrumented JVM and estimation tool
- Install the instrumented JVM
- Update the JVM option files to use instrumented JVM
- Run Test
  - Results will be sensitive to the contents of the test, make it as real as possible.
- Download the STDERR file from the address space to a PC
- Process the flat file using the IBM supplied Excel workbook

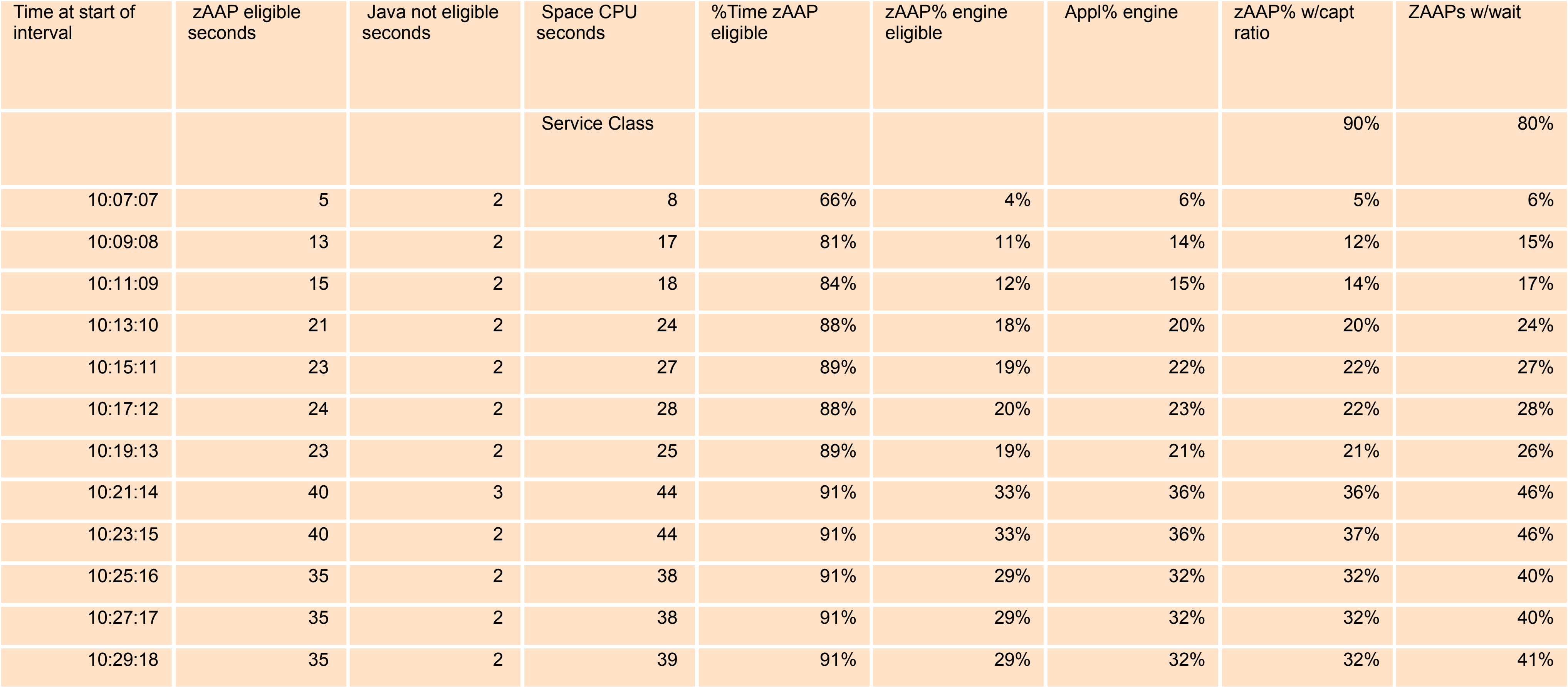
Where's the Data?

```
<WRC1.50397458> Interval at: 10:07:07 Switches To/From IFA: 242857 Java IFA: 18.745860 sec. Java Standard CPU 0.861216
 sec. Interval address space CPU: 23.120377 sec.
 <WRC1.50397458> Interval at: 10:09:08 Switches To/From IFA: 587782 Java IFA: 5.083801 sec. Java Standard CPU 2.023745
 sec. Interval address space CPU: 7.706499 sec.
 <WRC1.50397458> Interval at: 10:11:09 Switches To/From IFA: 613917 Java IFA: 13.415511 sec. Java Standard CPU 2.094004
 sec. Interval address space CPU: 16.643484 sec.
```

- Ugly
  - No SMF record
  - No log file
  - Print image

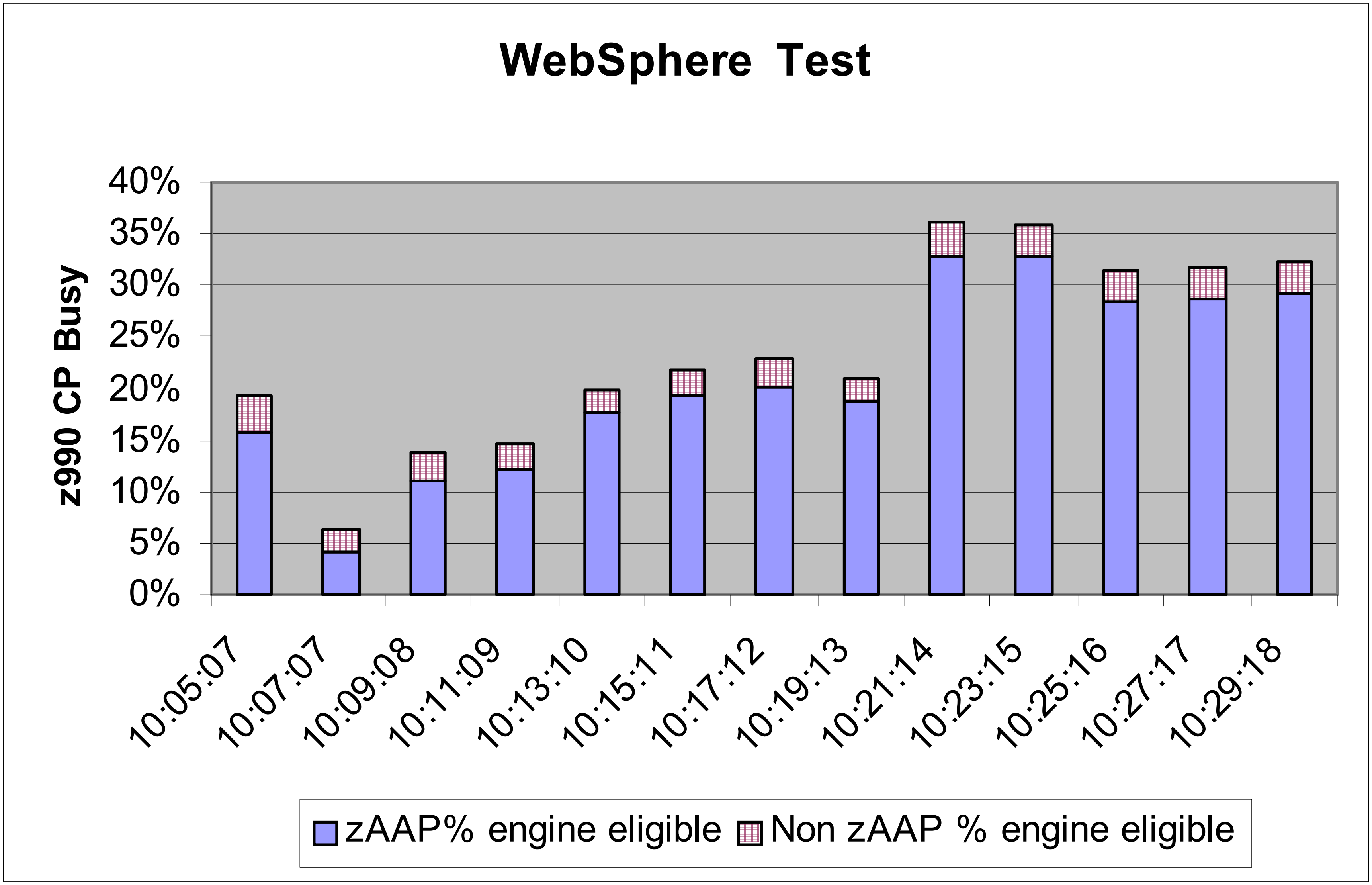
How Can I Process Ugly Data?

- IBM provides an Excel workbook to help process the output of the instrumented JVM
  - Available at https://www6.software.ibm.com/dl/zosjava2/zosjava2-p



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
			Service Class				90%	80%
10:07:07	5	2	8	66%	4%	6%	5%	6%
10:09:08	13	2	17	81%	11%	14%	12%	15%
10:11:09	15	2	18	84%	12%	15%	14%	17%
10:13:10	21	2	24	88%	18%	20%	20%	24%
10:15:11	23	2	27	89%	19%	22%	22%	27%
10:17:12	24	2	28	88%	20%	23%	22%	28%
10:19:13	23	2	25	89%	19%	21%	21%	26%
10:21:14	40	3	44	91%	33%	36%	36%	46%
10:23:15	40	2	44	91%	33%	36%	37%	46%
10:25:16	35	2	38	91%	29%	32%	32%	40%
10:27:17	35	2	38	91%	29%	32%	32%	40%
10:29:18	35	2	39	91%	29%	32%	32%	41%

Use the Capability of a Spreadsheet



What is the Objective for the zAAP

- Reduce software charge with possible reduction in performance

- Reduce the software charge and allow excess general purpose CP capacity to support Java workload

- Reduce software charge while maintaining current performance

What are the Tuning Options?

- Specified in IEAOPTxx member of parmlib
  - IFACrossOver=YES|No
  - IFAHonorPriority=YES|No
- Can have an impact on the:
  - Performance of Java work
  - The capacity plan for the zAAP

IFACrossOver

- Controls where Java work can run
  - No - Java work can ONLY run on the zAAP, it can not run on a general purpose CP unless there are no zAAPs available
    - Provides maximum offload of Java based work to the zAAP
    - Java work can be queued waiting for a zAAP while general purpose CP is available
  - Yes - Java work may execute on both general purpose CPs and zAAP
    - Java based work may run on a general purpose CP reducing the amount of work run on the zAAP
    - Java work will run if any processor is available

IFAHonorPriority

- Meaningless if IFACrossOver = No

- IFACrossOver = Yes, IFAHonorPriority = Yes
  - Work is processed by general purpose CP and zAAP in priority sequence
  - The general purpose CP will select both zAAP and non-zAAP work in priority sequence
    - Provides best performance for high priority work regardless of workload type
    - May reduce the amount of Java based work running of zAAP
      - Priority of Java work
      - Utilization of general purpose CPs
      - Ratio of general purpose CPs to zAAPs

IFAHonorPriority

- IFACrossOver = Yes, IFAHonorPriority = No
  - zAAP will process Java work in priority sequence
  - General purpose CPs will process Java work below non-Java discretionary work
    - Should help keep zAAP busier
    - Java work may wait for a general purpose CP when lower priority work is running
    - Ability to use Whitespace

How to Set the Options

- IFACrossOver=Yes, IFAHonor Priority=Yes
  - Provides best performance for high priority work regardless of workload type
  - Probably reduces the estimated amount of work running on the zAAP

- IFACrossOver=Yes, IFAHonor Priority=No
  - Allows all processors to run ready work, but gives the zAAP better chance of running Java work
  - Helps achieve estimated offload of Java work

- IFACrossOver=No
  - Provides maximum offload of Java work onto the zAAP
  - May cause Java work to queue to a zAAP when a general purpose CP is available

Application Considerations

- Application selection is probably needed
  - Not all Java applications will be a good candidate for zAAP

- Use of zAAP controlled at the JVM level
  - Controls include:
    - On
    - Off
    - Force
  - Are all applications running under the same JVM good candidates?

Application Selection

- Maximize the amount of time on the zAAP
  - Java intensive application - good candidate
  - Java trivial application - bad candidate
    - Estimation tool provides guidance

Time at start of interval	zAAP eligible seconds	Java not eligible seconds	Space CPU seconds	%Time zAAP eligible	Duration
			Service Class		
10:05:22	137	102	240	57%	120
10:07:22	167	197	365	46%	121
10:09:23	164	199	364	45%	121
10:11:24	163	197	361	45%	121
10:13:25	162	194	357	45%	121

Application Selection

- Minimize the cost of getting on the zAAP
  - Software switching rate
    - Low switch rate minimizes the software cost
    - High switch rate increases the software cost
    - Estimation tool provides guidance

Time at start of interval	Switch rate	zAAP eligible seconds	Java not eligible seconds	Space CPU seconds
7:04:20	7,806	18	11	51
7:09:21	2,835	9	5	28
7:14:22	3,022	8	5	27
7:19:23	2,906	15	5	32
7:24:25	2,512	7	4	24
7:29:26	2,687	7	5	23

Application Selection

- Java eligible time per switch is probably a good indicator of application applicability for a zAAP
  - ►Value can be calculated using the estimation tool

Time at start of interval	Switch rate	zAAP eligible seconds	Duration	zAAP eligible microseconds per switch
10:05:22	19,084	137	120	60
10:07:22	31,681	167	121	44
10:09:23	32,913	164	121	41
10:11:24	29,986	163	121	45
10:13:25	28,798	162	121	46
10:15:26	29,763	161	121	45

Summary

- zAAP has the ability to offload Java cycles from the general purpose CPs
- Estimation tool is available to help understand the potential
- Amount of actual offloaded cycles can be affected by zAAP options selected
- Estimation tool can help identify good Java candidates
- More information to follow as we learn more about zAAPs

Questions

What About......