JES2 Common Problems

Matthew Hunger
mlhunger@us.ibm.com
IBM – Poughkeepsie, NY

Wednesday, October 28th 2020

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

- IBM®
- MVS™
- Redbooks®
- RETAIN®
- 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.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- SET and Secure Electronic Transaction are trademarks owned by SET Secure Electronic Transaction LLC.
- 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.

Overview

- What are the recurring questions that the service team keeps answering? We’re here to spread awareness of some scenarios which seem to keep popping up among our JES2 customers…
    - How does JES2 handle duplicate Jobs processing?
    - What is automatic checkpoint tuning?
    - How do I collect JES2 PERFDATA?
    - How do I respond to a $HASP263 or $HASP292 message?
    - How do I set up and use JES2EDS?

Duplicate Job Processing

- JES2 CPU consumption can increase due to duplicate jobs (batch jobs with the same JOBNAME) in the MAS if the job queues are structured in certain ways
- Duplicate jobs in and of themselves will not cause increased CPU consumption
- JES2 does not have an intrinsic problem with duplicate jobs, and there is no “limit” of duplicates above which we would expect performance degradation
- So why do customers keep encountering high CPU in JES2 when they submit large amounts of duplicate jobs?

Duplicate Job Processing

- How does JES2 handle duplicate jobs?
  - JOBDEF and/or JOBCLASS(x) DUPL_JOB parameter sets behavior:
    - NODELAY - multiple duplicates can execute at the same time
    - DELAY - only 1 job with a given name can be in execution at any time
  - JES2 must track which jobs are in execution and hold back duplicates when DELAY is coded
    - This has an associated CPU cost within $QGET

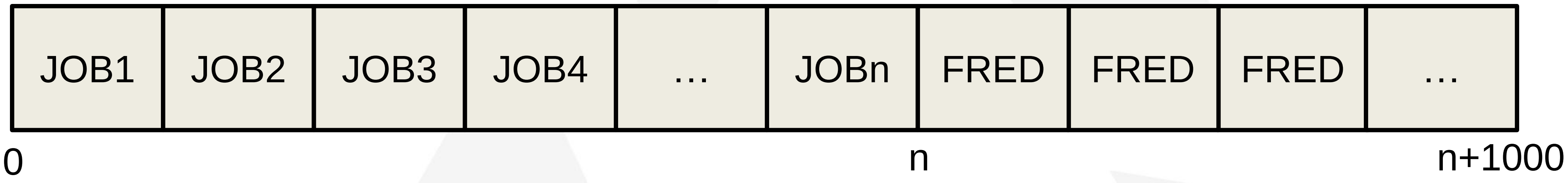
Duplicate Job Processing

- $QGET routine selects jobs to be placed into execution.
  - Works sequentially on a job queue – the more jobs we need to traverse to find a selectable job, the more CPU used
  - When DUPL_JOB=DELAY is coded, $QGET needs to skip over duplicate jobs if we know one of those jobs is currently in execution
  - Your queue composition matters
  - Other things that affect job selection:
    - SCHENV, SYSAFF, JOBCLASS XEQCOUNT, service class goals, available inits
    - All criteria needs to be checked in $QGET and this has an overhead cost
    - There are no job queue limits but everything has a cost

Session 25911

Duplicate Job Processing

- Scenario 1:
  - Job Queue for class A
  - DUPL_JOB=DELAY

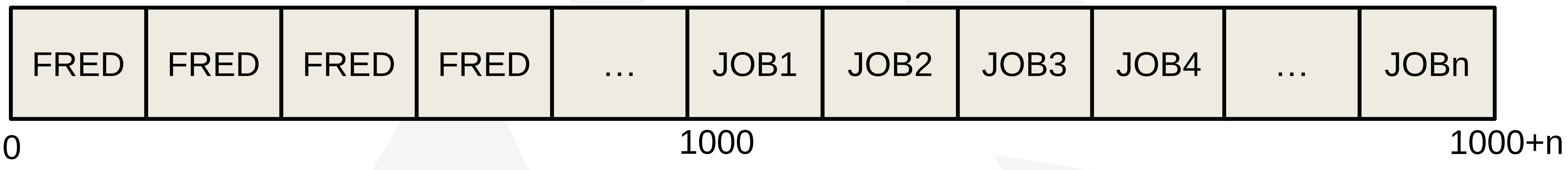


JOB1	JOB2	JOB3	JOB4	…	JOBn	FRED	FRED	FRED	…

- $QGET can choose non-duplicates right from the front of the queue
- Common when lots of dup jobs are dumped into a currently used job class

Duplicate Job Processing

- Scenario 2:
  - Job Queue for class A
  - DUPL_JOB=DELAY



FRED	FRED	FRED	FRED	…	JOB1	JOB2	JOB3	JOB4	…	JOBn

0

1000

1000+n

- Only 1 FRED job can execute at a time
- $QGET still traverses all FRED jobs before it can select a non-duplicate
- We'll constantly be running the chain looking for the next JOBn to run!

Duplicate Job Processing

- Scenario 3:
  - Two job classes: Class A for duplicates and Class B for non-duplicates

JOBCLASS(A)

FRED	FRED	FRED	FRED	…

JOBCLASS(B)

JOB1	JOB2	JOB3	JOB4	…

- $QGET is minimized because we will generally be pulling from the front of the queue
- Best case scenario to minimize duplicate job impact to non-duplicate work
- **Isolate your duplicate jobs!**

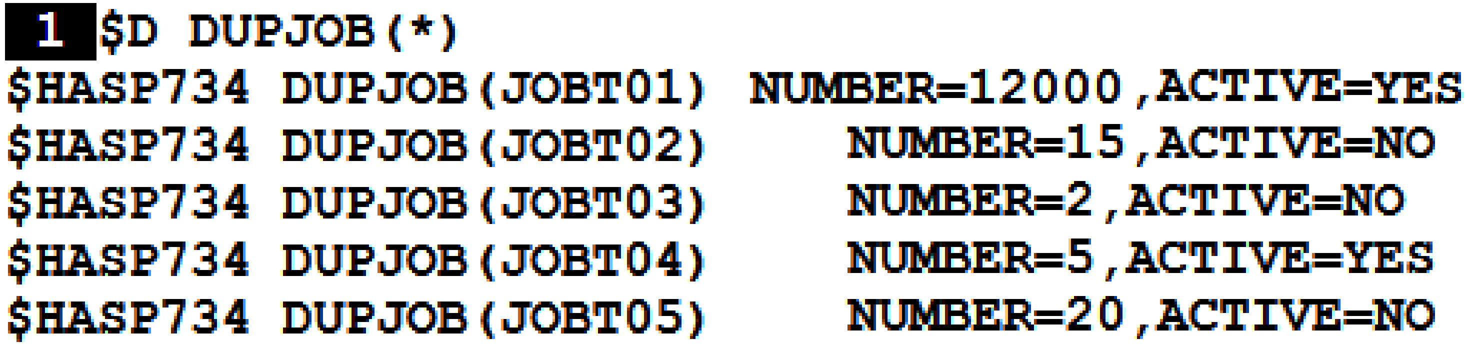
Session 25911

Duplicate Job Processing

- **My JES2 CPU usage is higher than normal. How can I tell if it is because of duplicate job processing?**
- This will manifest as CPU usage within the JES2 EXEC PCE
  - The EXEC PCE is in charge of putting jobs into and out of execution
  - JES2 PERFDATA will show where JES2 is using its CPU
    - We will demonstrate PERFDATA gathering a bit later

Duplicate Job Processing

- **How can I tell how many duplicate jobs are in my system?**
  - $D DUPJOB(*) or $D DUPJOB(nnnnnnnn)
  - Will display number of duplicates for all duplicate job families or a given job



```
 1 $D DUPJOB(*)
$HASP734 DUPJOB(JOBT01) NUMBER=12000,ACTIVE=YES
$HASP734 DUPJOB(JOBT02)     NUMBER=15,ACTIVE=NO
$HASP734 DUPJOB(JOBT03)     NUMBER=2,ACTIVE=NO
$HASP734 DUPJOB(JOBT04)     NUMBER=5,ACTIVE=YES
$HASP734 DUPJOB(JOBT05)     NUMBER=20,ACTIVE=NO
```

  - Here we see there are 12000 instances of JOBT01 and one of those jobs is currently executing (ACTIVE=YES)

Duplicate Job Processing

- **How can I get out of a high CPU condition?**
  - Use $D DUPJOB and PERFDATA to confirm a dup job issue
  - Cancel/purge the offending jobs ($CJ,P)
  - Rethink how you are submitting these jobs before you attempt them again
    - Should they run in a separated job class?
    - Can we restructure our job queues to reduce $QGET overhead?
- **Updates to $QGET tracing in V2R4**
  - New fields in JES2 trace 31 to give information on
    - Number of jobs skipped for duplicates
    - Number of jobs skipped for class limits
    - Number of times EXIT49 was called
    - Additional exit timing information

Session 25911

JES2 Automatic Checkpoint Cycle Tuning

- The health of the CKPT cycle is crucial to JES2
- Customers can manually tune the CKPT cycle values for each of their members
    - MASDEF HOLD
        - Maximum time a MAS member can hold the ckpt lock
    - MASDEF DORMANCY
        - Minimum and max time a member can wait before requesting the ckpt lock
- In z/OS V2R2, we added automatic checkpoint tuning
- JES2 determines the optimum hold and dorm values for each member based on workload “pain”

JES2 Automatic Checkpoint Cycle Tuning

- Controlled by the MASDEF CYCLEMGT setting
  - Set in init deck or via $T MASDEF,CYCLEMGT=AUTO
- In AUTO mode, HOLD and DORMANCY cannot be changed by command
- $D MASDEF,HOLD,DORMANCY can still be used to monitor these parameters as they are changed by JES2
- All MAS members must be at V2R2 or higher for AUTO mode
- Toggling between AUTO and MANUAL is quick and easy
  - If you don't like the results, returning to manual is as simple as $T MASDEF,CYCLEMGT=MANUAL
- We'll revert to the previous HOLD/DORM right away

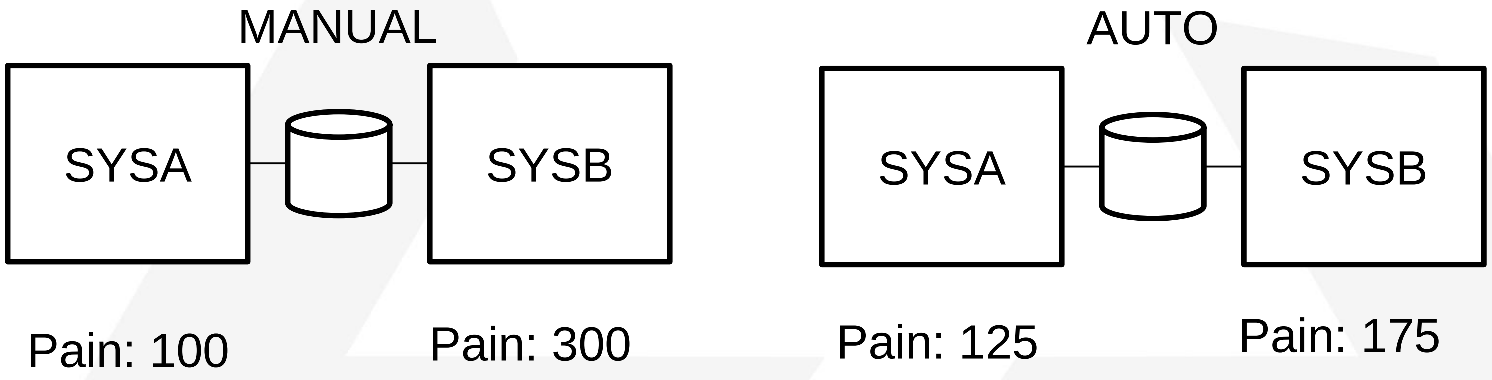
Session 25911

JES2 Automatic Checkpoint Cycle Tuning

- **So what is pain anyway?**
  - It is a measure of how much work is waiting for the checkpoint and how long it had to wait
  - Pain is reported in JES2 PERFDATA under CKPTSTAT heading
  - “Pain rate” multiplied by time a PCE spent waiting for CKPT ($QSUSE)
  - Some PCEs have a higher base pain rate
  - Pain means nothing in and of itself; it is only a metric for comparison
  - The goal of AUTO mode is to minimize pain **in the MAS**

JES2 Automatic Checkpoint Cycle Tuning

- **CYCLEMGT=AUTO caveats…**
  - AUTO will optimize the CKPT cycle for the entire MAS, not just for a specific member or application
  - This could be undesirable depending on your workload distribution
  - High pain might be tolerable on systems running lower priority work



MAS Pain went down – but SYSA Pain went up

Session 25911

JES2 Automatic Checkpoint Cycle Tuning

- **Not all JES2 workload is created equal**
  - Prior to V2R4, SAPI workload may generate less pain than other JES2 work
  - A member processing SAPI workload with relatively little other JES2 work may not generate enough pain to compete with other members
  - Reorganizing your JES2 workload can improve overall pain levels
  - SAPI workload is being more heavily weighted starting in V2R4
  - The AUTO algorithm is always being improved
  - Your mileage may vary, but switching back to MANUAL is painless
    - Make sure everything passes the eyeball test – don’t rely only on pain numbers

Session 25911

JES2 Perfdata – Overview

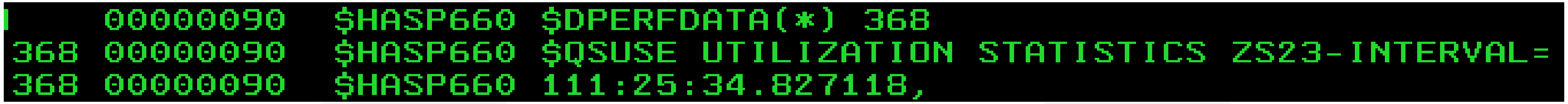
- JES2 is constantly collecting performance data in the background
  - No performance cost for gathering JES2 perfdata
- A summarized break down of CPU usage in various JES2 sub-components as well as things like CKPT cycle measurements
- Performance or high CPU issue? PERFDATA is a good place to start
- Likely to be requested by support if you open a service request
- Easy to collect – only requires a few console commands

JES2 Perfdata – Overview

- Always reset your perfdata interval before you begin data collection
  - $T PERFDATA(*),RESET
- To get a meaningful sample, collect data for 5-10 minutes (or as long as possible if the condition lasts for a shorter amount of time)
- To “collect” the perfdata, just use a display command to write to syslog
  - $D PERFDATA(*),L=Z
- It’s always best to collect more than one sample if possible
- Also, it’s good practice to collect samples under normal conditions for comparison

JES2 Perfdata – Tips and Tricks

- The perfdata interval – amount of time covered by the displayed statistics
- Perfdata is always being collected, so this is the time since the last refresh or JES2 restart
- If your system has been up for a while and you issue a perfdata display without doing a reset first, you’ll probably see something like this:

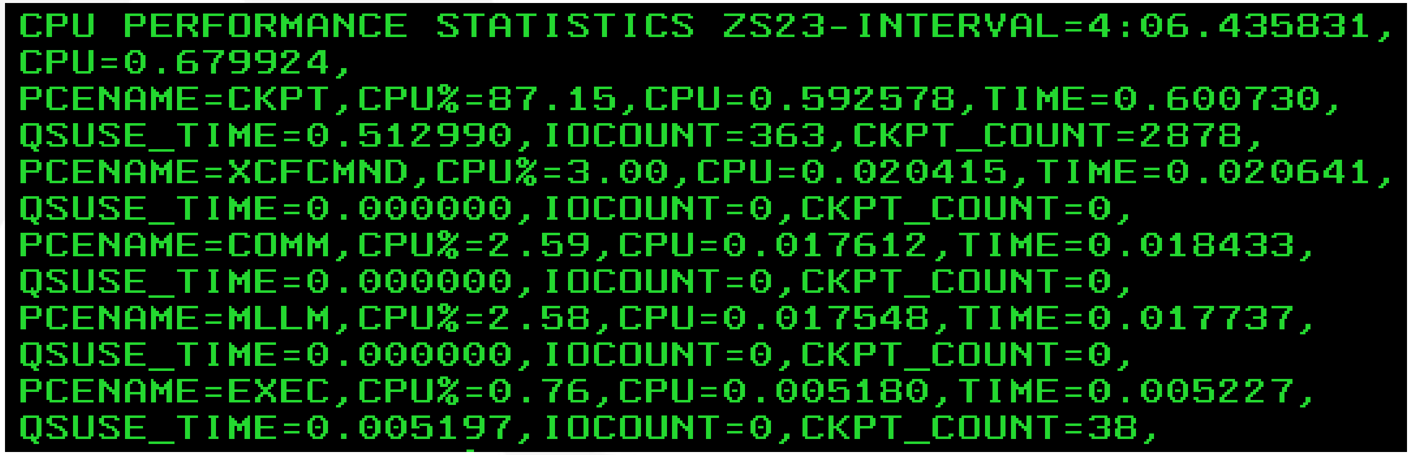


```
     00000090  $HASP660 $DPERFDATA(*) 368
368 00000090  $HASP660 $QSUSE UTILIZATION STATISTICS ZS23-INTERVAL=
368 00000090  $HASP660 111:25:34.827118,
```

- This means that the displayed data covers over 111 hours of JES2 activity. This is likely not useful data if your condition only lasted a few minutes

JES2 Perfdata – Tips and Tricks

- What does perfdata show us?
- CPU performance statistics
- High CPU usage in JES2 and EXEC PCE is largest consumer? Check your duplicate jobs.
- High usage in COMM? There may be a long running command
- HOPE PCE is very active? A SAPI application may be making a lot of requests

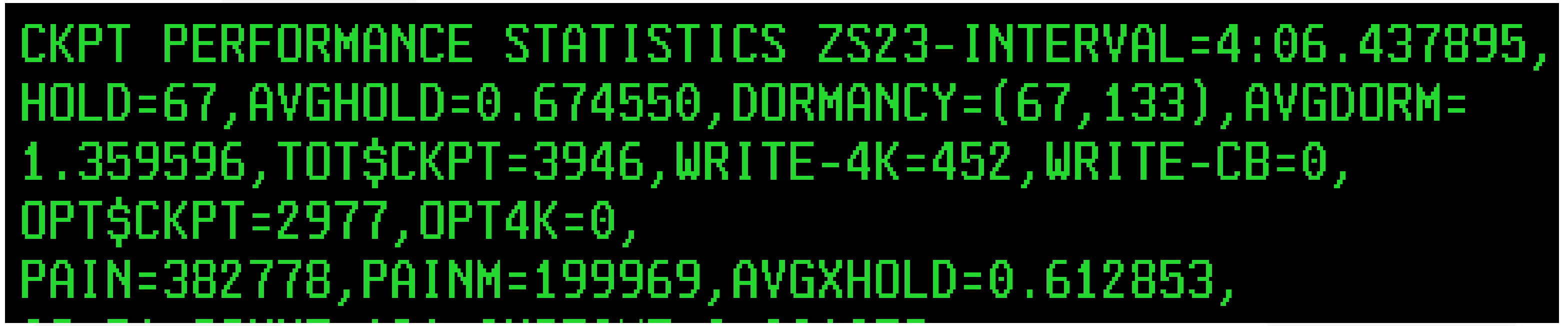


```
CPU PERFORMANCE STATISTICS ZS23-INTERVAL=4:06.435831,
CPU=0.679924,
PCENAME=CKPT,CPU%=87.15,CPU=0.592578,TIME=0.600730,
QSUSE_TIME=0.512990,IOCOUNT=363,CKPT_COUNT=2878,
PCENAME=XCFCMND,CPU%=3.00,CPU=0.020415,TIME=0.020641,
QSUSE_TIME=0.000000,IOCOUNT=0,CKPT_COUNT=0,
PCENAME=COMM,CPU%=2.59,CPU=0.017612,TIME=0.018433,
QSUSE_TIME=0.000000,IOCOUNT=0,CKPT_COUNT=0,
PCENAME=MLLM,CPU%=2.58,CPU=0.017548,TIME=0.017737,
QSUSE_TIME=0.000000,IOCOUNT=0,CKPT_COUNT=0,
PCENAME=EXEC,CPU%=0.76,CPU=0.005180,TIME=0.005227,
QSUSE_TIME=0.005197,IOCOUNT=0,CKPT_COUNT=38,
```

This system is mostly idle. Majority of CPU used by JES2 is for overhead of the CKPT cycle over a 4-minute interval

JES2 Perfdata – Tips and Tricks

- Checkpoint performance statistics
- Current hold and dorm settings as well as the actual average hold and average dormancy over the interval
- Total pain across the MAS as well as pain generated on this member (PAINM)
- AVGXHOLD – average excessive hold amount in seconds

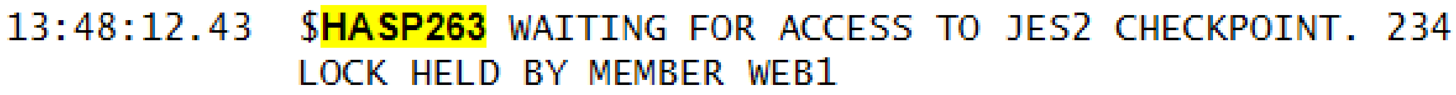


```
CKPT PERFORMANCE STATISTICS ZS23-INTERVAL=4:06.437895,
HOLD=67,AVGHOLD=0.674550,DORMANCY=(67,133),AVGDORM=
1.359596,TOT$CKPT=3946,WRITE-4K=452,WRITE-CB=0,
OPT$CKPT=2977,OPT4K=0,
PAIN=382778,PAINM=199969,AVGXHOLD=0.612853,
```

JES2 Perfdata – Usage Example

- JES2 CPU usage spikes / performance degrades every day at 2:00AM when our archiver program runs. What’s going on?
- Reset perfdata statistics when the issue occurs
    - Can use automated commands or operator input
- Gather 2-3 samples over 5-10 minutes during the condition (best case)
- Get a console dump of JES2
- When the condition clears, get another set of perfdata samples for comparison
- Getting good perfdata is a must-gather for IBM to diagnose performance issues

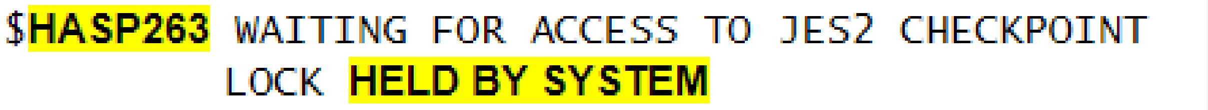
$HASP263



```
13:48:12.43   $HASP263 WAITING FOR ACCESS TO JES2 CHECKPOINT. 234
              LOCK HELD BY MEMBER WEB1
```

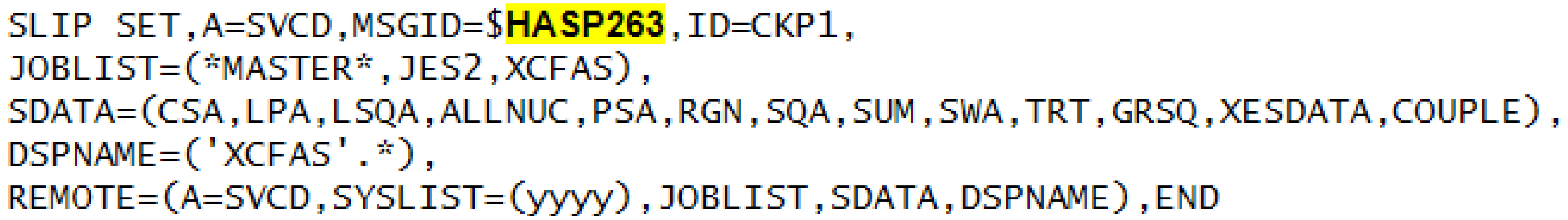
- First flavor of $HASP263 message – member issuing the message waited for the lock longer than the MASDEF LOCKOUT, which is held by another member of the MAS
- The system issuing the message is a victim in this case
- Need diagnostics of WEB1 in order to determine why it won’t give up the lock
- SLIP dump of MSGID $HASP263 is not good enough
- We require a dump of the system holding the lock, not the system issuing the message
  - Can set a MSGID SLIP to trigger remote dump of all other systems in the MAS
  - Generally, we won’t know who will be responsible until the message is issued
  - SLIP would need to be set on all members if we don’t know who will be affected

$HASP263



```
$HASP263 WAITING FOR ACCESS TO JES2 CHECKPOINT
         LOCK HELD BY SYSTEM
```

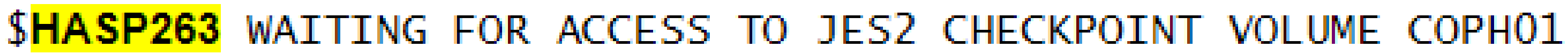
- Second flavor of $HASP263 message – member issuing the message waited for the lock longer than the MASDEF LOCKOUT, but we are unable to determine where the lock is held
- The system issuing the message is still a victim – but the culprit has changed
- This message indicates that our system did not get a response from XCF when we asked who held the lock
- Again, MSGID SLIP for $HASP263 is not good enough, need remote dump of all other members including XCF and it’s associated dataspace



```
SLIP SET,A=SVCD,MSGID=$HASP263,ID=CKP1,
JOBLIST=(*MASTER*,JES2,XCFAS),
SDATA=(CSA,LPA,LSQA,ALLNUC,PSA,RGN,SQA,SUM,SWA,TRT,GRSQ,XESDATA,COUPLE),
DSPNAME=('XCFAS'.*),
REMOTE=(A=SVCD,SYSLIST=(yyyy),JOBLIST,SDATA,DSPNAME),END
```

Session 25911

$HASP263



```
$HASP263 WAITING FOR ACCESS TO JES2 CHECKPOINT VOLUME COPH01
```

- Third flavor of $HASP263 message – member issuing the message cannot get access to the DASD volume which houses the CKPT (MASDEF LOCKOUT exceeded)
- Likely that some other application is obtaining a lock on the volume
- We recommend keeping the CKPT dataset on its own dedicated volume
- Can also occur if a mirroring/backup application is trying to back up the CKPT volume
- Also, possible I/O problem, other messages may be issued

$HASP263

- $HASP263 is often accompanied by other messages on other MAS members which more directly indicate the problem
  - **$HASP9203** – LONG PCE DISPATCH
    - This member of the MAS is busy doing some kind of JES2 processing
  - **$HASP9207** – JES CKPT LOCK HELD
    - This member is holding the lock longer than it should
  - **$HASP9211** – JES2 MAIN TASK NOT RUNNING
    - Something is preventing JES2 from being dispatched by MVS
    - Most often a lack of available CPU
  - **$HASP9213** – LONG JES2 COMMAND PROCESSING
    - JES2 is busy trying to process an operator command

$HASP263 – Summary

- The MAS member issuing $HASP263 is a victim
- A dump of the member issuing $HASP263 is not enough to find root cause
- You must gather a dump of the system holding the JES2 CKPT lock
    - Either trigger remote dumps on other MAS members when you see $HASP263
            - OR
    - MSGID SLIPs for related messages on the systems holding the lock
        - $HASP9203, $HASP9207, $HASP9211, $HASP9213

JES2EDS – What is it?

- JES2 V2R3 introduced the JES2 Email Delivery Service – JES2EDS
- Ability to notify a TSO user or email account when a job completes
- Works using the //NOTIFY= statement
- Multiple software pre-requisites
    - OMVS, TCPIP, zOSMF
    - JES2 uses the z/OS Web Enablement Toolkit to communicate with zOSMF
    - zOSMF accesses its configured SMTP server to actually send the emails
- Support team receives many questions regarding setup

JES2EDS – Setup FAQs

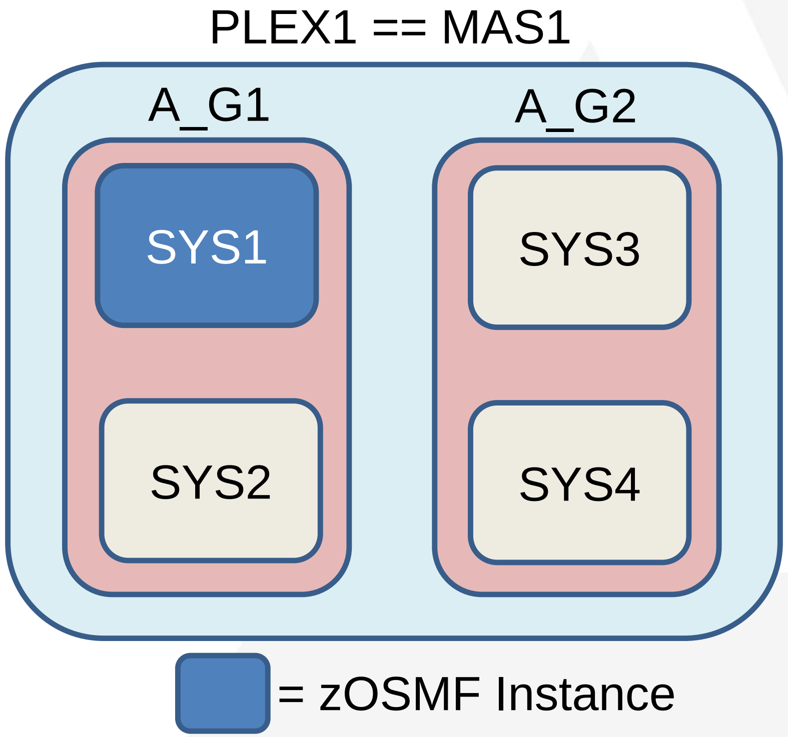
- Setup guide - https://ibm.biz/BdZSVT
- Assign a user ID to the JES2 EDS address space
    - Specified either by creating a profile in the STARTED class that matches the JES2 EDS address space name, or by adding an entry in the started procedures table (ICHRIN03)
    - For ease of setup, best for associated USERID to be JES2 subsystem user ID
- Keyring/Cert creation is straight forward for a default configuration
- A few small caveats:
    - The keyring must be named JES2EDS, there is no way to reconfigure this
    - Opening the keyring uses r_datalib SAF callable service GetRingInfo
    - Top Secret Security is missing this implementation without **PTF SO06741**
    - You must apply this PTF to use JES2EDS with TSS

JES2EDS – Setup FAQs

- Step 6: “*Add the user identifier that is configured in step 1 to the z/OSMF configuration as a valid z/OSMF user*” – what does this really mean?
- The userID associated with JES2EDS needs to be given the same authorizations as a normal zOSMF end user.
- Easiest way to do this is to add EDS userID to the IZUUSER RACF group
- No groups? Treat EDS userID as if it were a new user to zOSMF
  - zOSMF security requirements: https://ibm.biz/BdzK2Q

JES2EDS – Setup FAQs

- How do I tell JES2EDS which zOSMF server to communicate with?
  - JES2EDS finds the zOSMF server automatically – not configurable
  - The zOSMF server needs to be accessible to JES2EDS in the same zOSMF AUTOSTART_GROUP (set in IZUPRM) to be discoverable



- One zOSMF instance in the plex, running on SYS1, in autostart group A_G1
- EDS services available on SYS1 and SYS2
- JES2 in SYS3 and SYS4 will not be able to connect to the zOSMF instance
- Manually start a zOSMF instance in A_G2 or merge into one autostart group

JES2EDS – Setup FAQs

- JES2EDS ASID is starting but I don’t want to use it.
  - The ASID will be started when a job with NOTIFY= keyword completes
  - Or at JES2 start if there is an email stored on SPOOL waiting to be sent
  - There is currently no “clean” way to shut down only the JES2EDS ASID without shutting down JES2 entirely
  - There is no toggle to disable JES2EDS features
  - If you do not want JES2EDS ASID starting on your system, you will need to restrict the //NOTIFY= statement
  - Email queues can be purged with $PS($EDSQnnn)
  - You can display your queues with $DS($EDSQ*)
  - $PS($EDSQnnn),ONERROR – only purge messages with a delivery error

Questions?