IBM Education Assistance for z/OS V2R2

Items: JES2 Growth: Grow from 400K to 1M Active jobs
JES2 Growth: Grow checkpoint without cold start
JES2 Growth: Dynamic Checkpoint tuning
Element/Component: JES2

Material current as of May 2015

Agenda

- Trademarks
- Presentation Objectives
- Overview
- Usage & Invocation
- Interactions & Dependencies
- Migration & Coexistence Considerations
- Installation
- Presentation Summary
- Appendix

Trademarks

- See url http://www.ibm.com/legal/copytrade.shtml for a list of trademarks.

Presentation Objectives

- Support growing the number of jobs and output JES2 can support
  - Includes performance enhancements
  - Improved system management
  - Improved RAS
- Changes include
  - SYSOUT work selection enhancements
  - Updates for duplicate job processing
  - Reworking checkpoint on I/O processing
    - Includes using 64 bit storage
  - Simplify managing the JES2 checkpoint data sets
    - Size and configuration
  - Tuning MASDEF HOLD and DORMANCY automatically
  - Numerous RAS enhancements

Overview – Grow from 400k to 1M Active Jobs

- Problem Statement / Need Addressed
  - Ability to support larger numbers of jobs resident on spool at one time
  - Deal with performance problems related to larger number of jobs
- Solution
  - Update limits on job, JOE, and BERT numbers
  - SYSOUT work selection improvements (beyond what 2.1 did)
  - Better management of duplicate jobs
- Benefit / Value
  - Support larger workloads with improved performance

Usage & Invocation – Grow from 400k to 1M Active Jobs

- External updated for this support
  - New $ACTIVATE level (z22) to support new functions
    - Level z2 is not supported in this release
  - SYSOUT work selection performance extended to local devices
    - New option to activate support
  - Updated limits for major checkpoint items
    - JOBNUM limit now 1,000,000
    - JOENUM limit now 2,500,000
    - BERTNUM limit now 2,500,000
  - Increased BLOB size
    - Cache of available track groups increased from 256 to 1024
  - New command to display duplicate job families
    - $D DUPJOB

New $ACTIVATE level

- `$ACTIVATE,LEVEL=z22`
  - Requires SPOOLDEF CYL_MANAGED=ALLOWED be set
  - Enables a number z22 function
    - Larger limits
    - Job execution zones (dependent job control)
    - Larger SPOOL BLOB size (256 updated to 1024)
    - Checkpoint extend/alter processing
- Same management options same as when z11 mode was introduced
  - $ACTIVATE supports `LEVEL=z11` or `LEVEL=z22`
  - OPTSDEF COLD_START_MODE= Z11 or Z22
  - S JES2,PARM=(UNACT) warm starts in z11 mode
  - $D ACTIVATE displays current mode/what is needed for other mode
- Checkpoint size increases for BLOB size and new ZJC data area
  - About 6 tracks depending on settings

Work Selection Performance

- JES2 SYSOUT work selection process for the SAPI applications extended to other devices (local printers and punches and offload)
  - SYSOUT work selection is the process of selecting SYSOUT output groups ($#GET)
- These enhancements address both:
  - active work selection – assigning work to a device (GET)
  - passive work selection – waking up the device when work with appropriate characteristics becomes available (POST)
- Active SYSOUT work selection uses JOE index added in V2R1
- In V2R2, improvements have been made to the JOE index management. In particular, JES2 will now periodically remove unused JOE index nodes.

Work Selection Performance

- To enable active work selection optimization (GET):

```
$T OUTDEF,WS_OPT=YES
```

  - Activates both SAPI and local device support
  - This is the same option introduced in V2R1.
  - This type of optimization has MAS scope and persists until it is explicitly disabled.

- To enable passive work selection optimization (POST) for local devices:

```
$T OUTDEF,LDEV_OPT=YES
```

  - This type of optimization has a member scope and is only active until this member is restarted.

- All three optimization types are independent and can be enabled in any combination.
  - `LDEV_OPT=, SAPI_OPT=, WS_OPT=`

Work Selection Performance - Cautionary statement

- The implemented algorithms are sensitive to the SYSOUT selection criteria configured for a device (keyword WS=), so “mileage can vary”
- If for some reason optimization causes undesirable results, it can be turned off at any time:

```
$T OUTDEF,LDEV_OPT=NO
```

  - to disable passive work selection optimization (POST)

```
$T OUTDEF,WS_OPT=NO
```

  - to disable active work selection optimization (GET)

Work Selection Performance - Cautionary statement (cont.)

- Optimized path for active SYSOUT work selection (GET) uses index structure over JOEs
- This JOE index requires maintenance, which has its own CPU overhead
- It is unlikely but possible that in environment where majority of job SYSOUT is never processed (never printed or selected by SAPI applications), the overhead of index maintenance may not be offset by the performance improvements of work selection.
  In this case, using old path (WS_OPT=NO) may be preferable.

Duplicate Job Processing Performance

- By default, only one job with a given name can be active at a time
  - DUPL_JOB= on JOBDEF and JOBCLASS can alter this
  - Does not apply to jobs in a job group
- Additional overhead is associated with tracking these jobs
  - Impacts statistics reported to WLM for how many jobs available to run
- Internal changes made to reduce overhead of processing
- New command added to display duplicate job information

```
$D DUPJOB(jobname)
```

  - Output displays
    - Count of jobs with specified name
    - Indicator if job is active that will limit other jobs

Duplicate Job Processing Performance

- Example command

```
$ddupjob
$HASP734 DUPJOB(IBMUSERG)  NUMBER=3,ACTIVE=YES
$HASP734 DUPJOB(IBMUSERQ)  NUMBER=2,ACTIVE=NO
```

- In this display
  - 3 jobs are in the execution phase named IBMUSERG
    - At least one is executing in a class with DUPL_JOB=DELAY
  - 2 jobs are in the execution phase names IBMUSERQ
    - Either no jobs are executing or
    - Jobs are executing in a class with DUPL_JOB=NODELAY
- If JOBDEF DUPL_JOB=NODELAY then you will never see ACTIVE=YES
  - The tables are still maintained and duplicates can be displayed
  - Little overhead in this case
- Can also be used to detect runaway program submitting multiple job with the same name

64 bit CKPT Processing

- To support increased limits, checkpoint processing re-worked
  - No impact to instorage control blocks (JQE, JOE, etc)
  - I/O area moved to 64 bit storage
  - Most DASD checkpoint I/O processing moved to new subtask
    - CF processing already in a subtask
  - Numerous performance and RAS enhancements
- DASD (as with existing CF) I/O sensitive to being “ripped out”
  - More so since multiple physical I/Os used to do one logical I/O
    - Takes advantage of multiple paths to CKPT DASD
  - If possible, at least ABEND JES2 before crashing a system
- Larger checkpoint size implies
  - Larger checkpoint versions
    - More data space than previous releases
  - More virtual storage usage
    - Even with I/O area and other CBs moved to 64 bit

Overview – Grow checkpoint without cold start

- Problem Statement / Need Addressed
  - Reconfiguration dialog overly complex for simple changes
  - Cannot alter the size of a checkpoint data set without deleting it
- Solution
  - New fast path checkpoint reconfiguration process
  - Support dynamic changing of checkpoint size
    - Support ALTER processing on CF
    - Support extending a DASD data set into adjacent free space
- Benefit / Value
  - Simplified operations
  - Continuous operations

Usage & Invocation – Grow checkpoint without cold start

- New options on the CKPTDEF command

```
$T CKPTDEF,CKPTn=(DSNAME|VOLSER|STRNAME,INUSE)
```

  - Runs fast path reconfiguration if needed
    - Not needed for DSNAME/VOLSER/STRNAME if INUSE=NO
  - Only one change at a time allowed
    - CKPT1 or CKPT2
    - DSN/VOL or STRNAME or INUSE

- New information on $D CKPTSPACE command
  - Number of tracks in a DASD data set (TRACKS=*xxxx*)
  - Current and maximum size of a CF structure (SIZE=(*cur*,*max*))
    - Size is in 1K blocks

- New options to increase the size of the checkpoint
  - SPACE=(TRK|CYL|MAX,nnnn) for CKPT on DASD
  - SIZE=nnnn for CKPT on CF

- CF structure deleted when changing INUSE to NO

Reconfiguring the Checkpoint Data Sets

- Use $T CKPTDEF command to update CKPT specification
  - Uses the standard reconfiguration process to make changes
    - Works with down level members in the MAS
  - Will make the changes requested with no WTORs
    - Creates data set if it does not exist
  - Note, change is NOT reflected in the command response
    - Processing occurs asynchronously
- Sample command:

```
$T CKPTDEF,CKPT1=(VOL=CKPTPK)
$HASP829 CKPTDEF
$HASP829 CKPTDEF  CKPT1=(DSNAME=SYS1.JESCKPT1,VOLSER=SPOOL1,
$HASP829          INUSE=YES,VOLATILE=NO),
   :
$HASP285 JES2 CHECKPOINT RECONFIGURATION STARTING
$HASP233 REASON FOR JES2 CHECKPOINT RECONFIGURATION IS OPERATOR
         REQUESTED SET COMMAND
$HASP285 JES2 CHECKPOINT RECONFIGURATION STARTED - DRIVEN BY
         MEMBER IBM1
$HASP280 JES2 CKPT1 DATA SET (SYS1.JESCKPT1 ON CKPTPK) IS NOW IN USE
$HASP255 JES2 CHECKPOINT RECONFIGURATION COMPLETE
```

Reconfiguring the Checkpoint Data Sets

- If there is a problem, the reconfiguration fails
  - No WTORs asking for alternate actions
- Sample command:

```
$TCKPTDEF,CKPT1=VOL=UNKNOW
$HASP829 CKPTDEF
$HASP829 CKPTDEF  CKPT1=(DSNAME=SYS1.JESCKPT1,VOLSER=CKPTPK,
$HASP829          INUSE=YES,VOLATILE=NO),
   :
$HASP285 JES2 CHECKPOINT RECONFIGURATION STARTING
$HASP233 REASON FOR JES2 CHECKPOINT RECONFIGURATION IS OPERATOR
         REQUESTED SET COMMAND
$HASP285 JES2 CHECKPOINT RECONFIGURATION STARTED - DRIVEN BY
         MEMBER IBM1
$HASP424 MEMBER IBM1 -- UNKNOW IS NOT MOUNTED

$HASP255 JES2 CHECKPOINT RECONFIGURATION FAILED
```

Reconfiguring the Checkpoint Data Sets

- If CKPTn is INUSE=NO, then change can occur without reconfiguration
  - Changes appear in the command output
  - No actual allocation occurs, the checkpoint is not in use
- Sample command:

```
$DCKPTDEF
$HASP829 CKPTDEF
$HASP829 CKPTDEF  CKPT1=(DSNAME=SYS1.JESCKPT1,VOLSER=CKPTPK,
$HASP829          INUSE=NO),CKPT2=(DSNAME=SYS1.JESCKPT2,
   :
$TCKPTDEF,CKPT1=VOL=SPOOL1
$HASP829 CKPTDEF
$HASP829 CKPTDEF  CKPT1=(DSNAME=SYS1.JESCKPT1,VOLSER=SPOOL1,
$HASP829          INUSE=NO),CKPT2=(DSNAME=SYS1.JESCKPT2,
   :
```

Reconfiguring the Checkpoint Data Sets

- This is NOT a replacement for the reconfiguration dialog
  - I/O errors still use the traditional format
    - Still need to specify NEWCKPTn and OPVERIFY=NO
  - Does not interrupt hung I/O processing
    - CKPT PCE must be running to notice a change is needed
  - Presumes you know what you are doing
    - A typo in the data set name or volume will create the data set
- Traditional dialog still exists for all the reasons described above
- Examples used DASD data sets but also works for CF structures

Altering the Size of a CKPT

- New operand on CKPTn for DASD data set
  - SPACE=(TRK|CYL|MAX,*nnnn*)
  - Similar in function to SPACE= on SPOOL(*vvvvvv*)
  - MAX is the largest CKPT or all the free space, whichever is smaller
  - Data set must be INUSE=YES and allocated
    - Not CKPT2 in MODE=DUPLEX, DUPLEX=NO
  - Must be adjacent space AFTER the current data set
  - Must be $ACTIVATEd at z22 level

- Sample commands:

```
$DCKPTSPACE,CKPT1
$HASP852 CKPTSPACE  CKPT1=(CAPACITY=1788,UNUSED=1566,TRACKS=150)
$TCKPTDEF,CKPT1=SPACE=(TRK,160)
$HASP829 CKPTDEF
$HASP829 CKPTDEF  CKPT1=(DSNAME=SYS1.JESCKPT1,VOLSER=CKPTPK,
$HASP829          INUSE=YES,VOLATILE=NO),
   :
$HASP740 Volume CKPTPK Data set SYS1.JESCKPT1 Extend successful.
$DCKPTSPACE,CKPT1
$HASP852 CKPTSPACE  CKPT1=(CAPACITY=1908,UNUSED=1686,TRACKS=160)
```

Altering the Size of a CKPT

- New operand on CKPTn for CF structure
  - SIZE=*nnnn*|MAX
  - Units is 1K blocks (same a specification in policy)
  - MAX is the largest CKPT or all the policy limit, whichever is smaller
  - Only valid if current size is less than maximum size
    - Specify both INITSIZE and SIZE when defining structure
  - ALTER processing does round the size specified
  - Must be $ACTIVATEd at z22 level
- Sample commands:

```
$DCKPTSPACE,CKPT1
$HASP852 CKPTSPACE  CKPT1=(CAPACITY=617,UNUSED=391,SIZE=(4096,10240))
$TCKPTDEF,CKPT1=SIZE=5000
$HASP829 CKPTDEF
$HASP829 CKPTDEF  CKPT1=(STRNAME=SPOOLCKPT1,INUSE=YES,
$HASP829          VOLATILE=YES),CKPT2=(DSNAME=SYS1.JESCKPT2,
   :
$HASP739 CKPT1 structure SPOOLCKPT1 size altered to 5120
$DCKPTSPACE,CKPT1
$HASP852 CKPTSPACE  CKPT1=(CAPACITY=857,UNUSED=631,SIZE=(5120,10240))
```

Altering the Size of a CKPT

- JES2 CKPT structures are considered persistent
  - They are not deleted when JES2 goes down
  - Only way to delete them is to “force” them
- A new policy with a new size cannot fully active when structure exists
  - Policy change is “pending”
- CKPT reconfiguration altered to force structure when setting INUSE=NO
  - Both new fast path and traditional processing
  - Allows simple way to get a new structure with desired size
  - Does NOT work when forwarding the checkpoint
- NOTE: Ensure that the INITSIZE in the new policy for the structure is at least large enough to hold the current usage
  - This will build a new structure that is INITSIZE in size
  - If not big enough, cannot bring it INUSE

Altering the Size of a CKPT

- Sample commands:

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=CTTEST2
IXC511I START ADMINISTRATIVE POLICY CTTEST2 FOR CFRM ACCEPTED
IXC512I POLICY CHANGE IN PROGRESS FOR CFRM
TO MAKE CTTEST2 POLICY ACTIVE.
1 POLICY CHANGE(S) PENDING.
$DCKPTSPACE,CKPT1
$HASP852 CKPTSPACE  CKPT1=(CAPACITY=857,UNUSED=631,SIZE=(5120,10240))
$TCKPTDEF,CKPT1=INUSE=NO
$HASP829 CKPTDEF
$HASP829 CKPTDEF  CKPT1=(STRNAME=SPOOLCKPT1,INUSE=YES,
$HASP829          VOLATILE=YES),CKPT2=(DSNAME=SYS1.JESCKPT2,
   :
$HASP285 JES2 CHECKPOINT RECONFIGURATION STARTING
   :
$HASP280 JES2 CKPT1 DATA SET (STRNAME SPOOLCKPT1) IS NO LONGER IN USE
IXC513I COMPLETED POLICY CHANGE FOR CFRM.
CTTEST2 POLICY IS ACTIVE.
$HASP255 JES2 CHECKPOINT RECONFIGURATION COMPLETE
$TCKPTDEF,CKPT1=INUSE=YES
$HASP829 CKPTDEF
$HASP829 CKPTDEF  CKPT1=(STRNAME=SPOOLCKPT1,INUSE=NO),
   :
$HASP285 JES2 CHECKPOINT RECONFIGURATION STARTING
   :
$HASP280 JES2 CKPT1 DATA SET (STRNAME SPOOLCKPT1) IS NOW IN USE
$DCKPTSPACE,CKPT1
$HASP852 CKPTSPACE  CKPT1=(CAPACITY=617,UNUSED=391,SIZE=(4096,12288))
```

Overview – Dynamic Checkpoint Tuning

- Problem Statement / Need Addressed
  - JES2 checkpoint tuning key to good JES2 throughput
    - MASDEF HOLD= and DORMANCY= main parameters
  - Properly managing these settings depends on many factors
    - Workload, overhead, size of MAS, etc
  - Often not updated until performance is actually a problem
- Solution
  - Let JES2 manage the setting of the values
  - Adjusted based on workload, MAS size, etc
  - Set and forget
- Benefit / Value
  - Simplified operations
  - Better overall performance

Usage & Invocation – Dynamic Checkpoint Tuning

- New function – automatic checkpoint cycle management
  - Adjust HOLD= and DORMANCY= based on workload
  - Can display, but not set, current values
    - SDSF MAS display shows what members are using
  - Adjusts as member join and leaves MAS
- To enable automatic checkpoint cycle management:

```
$T MASDEF,CYCLEMGT=AUTO
```

  - This setting has MAS scope and persists until it is explicitly disabled.
- The function can also be enabled via MASDEF initialization statement
- The function can be turned off at any time via:

```
$TMASDEF,CYCLEMGT=MANUAL
```

  - All members will revert to HOLD and DORMANCY settings they had before automatic management had been turned on.

RAS Enhancements

- JES2 internal CTRACEs moved to 64 bit buffers
  - Increased size of buffers and thus depth of trace
- New internal CTRACE for checkpoint operations
- New DEBUG option to drive regular job queue verifications

  ```
  DEBUG QVERIFY=YES
  ```

  - High overhead but timely detection of queue errors
  - Very useful for debugging job/output/bert queue errors in test environment
  - NOT activated by DEBUG=YES
- Updated data in $D PERFDATA(CKPTSTAT)
- More granularity in WAIT data for $D PERFDATA(PCESTAT)
- Better selective $TRACE and filtering for processing that uses subtasks
- Many IPCS updates, new LOGRECs, etc.

Interactions & Dependencies

- $ACTIVATE LEVEL=Z22 requires all MAS members to be running z/OS 2.2 JES2
  - Pre-z/OS V2R2 members cannot enter a MAS at level Z22
- Increasing size of a CKPT requires activate to Z22 mode
- Automatic CKPT tuning requires all MAS members are at z/OS V2R2 JES2
- Increased JOB/SYSOUT/BERT limits require Z22 mode checkpoint

Migration & Coexistence Considerations

- Migrating from JES2 z/OS V1R13 or z/OS V2R1
  - Must $ACTIVATE to z11 mode prior to starting JES2 z/OS V2R2
  - APAR OA41740 needed on z/OS V1R13, or z/OS V2R1 member to coexist in MAS with z/OS V2R2
  - APAR is required for fall back as well
    - Some new data structures created by z/OS V2R2 JES2 will result in problems for prior releases if OA41740 is not installed.

Installation

- To enable ALTER processing for a CKPT data set on CF
  - Define a CFRM policy that has an INITSIZE and SIZE for the structure used for CKPT
  - For example:

```
STRUCTURE NAME(SPOOLCKPT1)    INITSIZE(4000) SIZE(12000)
```

- Note you could enable ALLOWAUTOALT and FULLTHRESHOLD
  - Not recommended due to how JES2 uses CKPT
  - Usage is based on parameter settings not workload
  - Better to manage size using JES2 commands

Presentation Summary

- Support growing the number of jobs and output JES2 can support
  - Includes performance enhancements
  - Improved system management
  - Improved RAS
- Changes include
  - SYSOUT work selection enhancements
  - Updates for duplicate job processing
  - Reworking checkpoint on I/O processing
    - Includes using 64 bit storage
  - Simplify managing the JES2 checkpoint data sets
    - Size and configuration
  - Tuning MASDEF HOLD and DORMANCY automatically
  - Numerous RAS enhancements

Appendix

- Publications
  - *z/OS V2R2.0 JES Application Programming* – SA32-0987
  - *z/OS V2R2.0 JES2 Commands* – SA32-0990
  - *z/OS V2R2.0 JES2 Initialization and Tuning Guide* – SA32-0991
  - *z/OS V2R2.0 JES2 Initialization and Tuning Reference* – SA32-0992
  - *z/OS V2R2.0 JES2 Installation Exits* – SA32-0995
  - *z/OS V2R2.0 JES2 Macros* – SA32-0996
  - *z/OS V2R2.0 JES2 Messages* – SA32-0989
  - *z/OS V2R2.0 MVS JCL Reference* - SA23-1385
  - *z/OS V2R2.0 MVS Using the Subsystem Interface* – SA38-0679