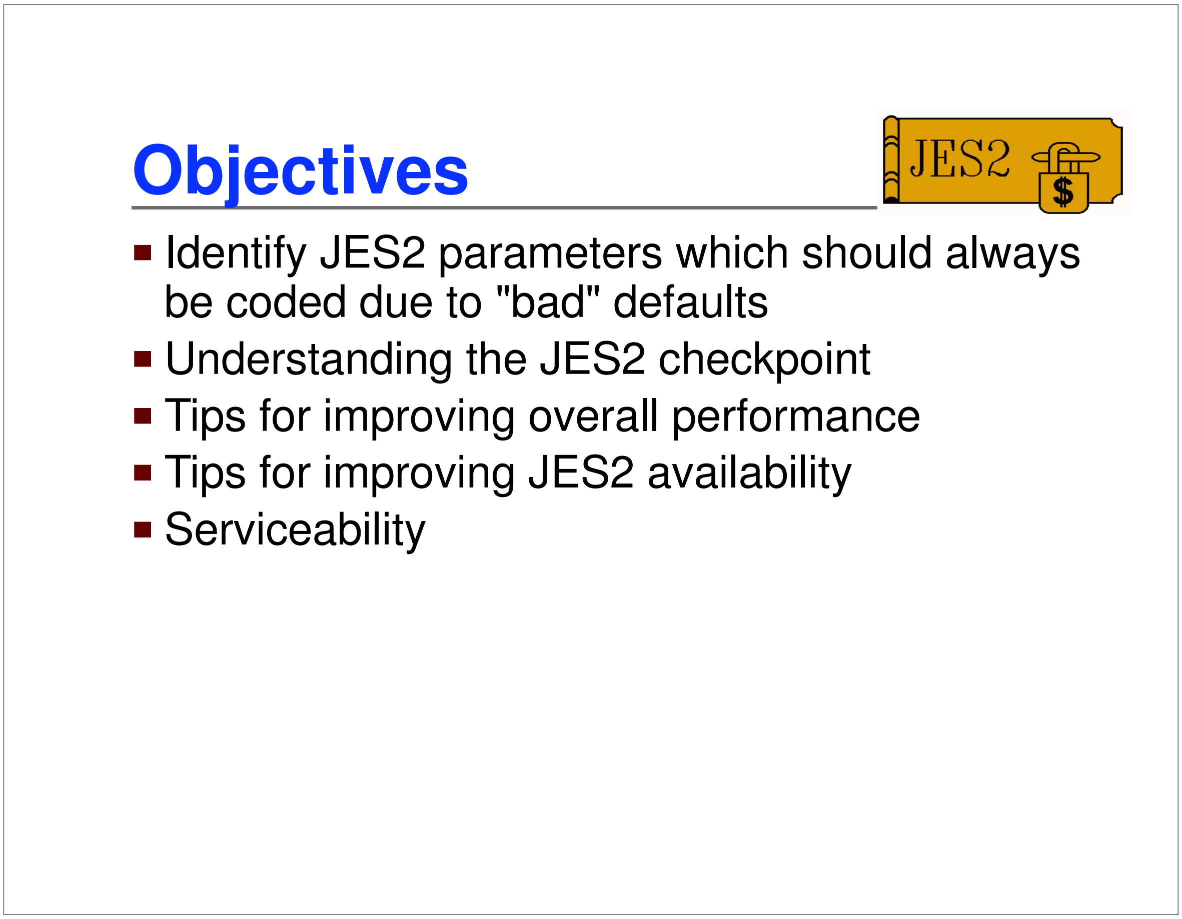
Configuration and Tuning Tips

SHARE 94, Session 2658

March 8, 2000

Chip Wood
JES2 Design/Development/Service
Poughkeepsie, NY
chipwood@us.ibm.com

Permission is granted to SHARE Inc. to publish this presentation in the SHARE proceedings. IBM retains its right to distribute copies of this presentation to whomever it chooses.

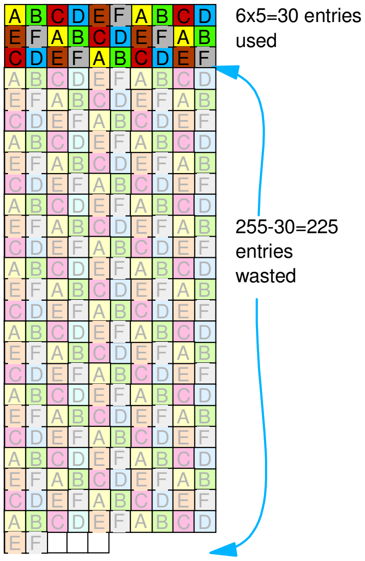


Objectives

- Identify JES2 parameters which should always be coded due to "bad" defaults
- Understanding the JES2 checkpoint
- Tips for improving overall performance
- Tips for improving JES2 availability
- Serviceability

"Bad" defaults (1)

BLOB



A	B	C	D	E	F	A	B	C	D
E	F	A	B	C	D	E	F	A	B
C	D	E	F	A	B	C	D	E	F
A	B	C	D	E	F	A	B	C	D
E	F	A	B	C	D	E	F	A	B
C	D	E	F	A	B	C	D	E	F
A	B	C	D	E	F	A	B	C	D
E	F	A	B	C	D	E	F	A	B
C	D	E	F	A	B	C	D	E	F
A	B	C	D	E	F	A	B	C	D
E	F	A	B	C	D	E	F	A	B
C	D	E	F	A	B	C	D	E	F
A	B	C	D	E	F	A	B	C	D
E	F	A	B	C	D	E	F	A	B
C	D	E	F	A	B	C	D	E	F
A	B	C	D	E	F	A	B	C	D
E	F	A	B	C	D	E	F	A	B
C	D	E	F	A	B	C	D	E	F
A	B	C	D	E	F	A	B	C	D
E	F	A	B	C	D	E	F	A	B
C	D	E	F	A	B	C	D	E	F
A	B	C	D	E	F	A	B	C	D
E	F	A	B	C	D	E	F	A	B
C	D	E	F	A	B	C	D	E	F
A	B	C	D	E	F	A	B	C	D
E	F								

6x5=30 entries used

255-30=225 entries wasted

- **SPOOLDEF TGBPERVL**
  - Number of track groups from each spool volume that can be obtained by jobs when CKPT not owned (default=**5**)
  - Too small a value and running jobs are forced to wait
    - ENQ **AWAITING SPOOL SPACE**
- Specify **TGBPERVL=255** to

SPOOLDEF

- Use **BUFSIZE=3992** (maximum)
- Best **TGSIZE=** value depends on the nature of the data the system processes
  - Small value is good when most jobs are relatively small (e-mail) - minimizes wasted space
    - Tradeoff: large jobs use LOTS of track groups
  - Large value is good when most jobs are relatively large (large print jobs) - better performance for those jobs
    - Tradeoff: small jobs can waste LOTS of space.
  - Default: **30**

© IBM Corporation 1998, 1999

SPOOLDEF

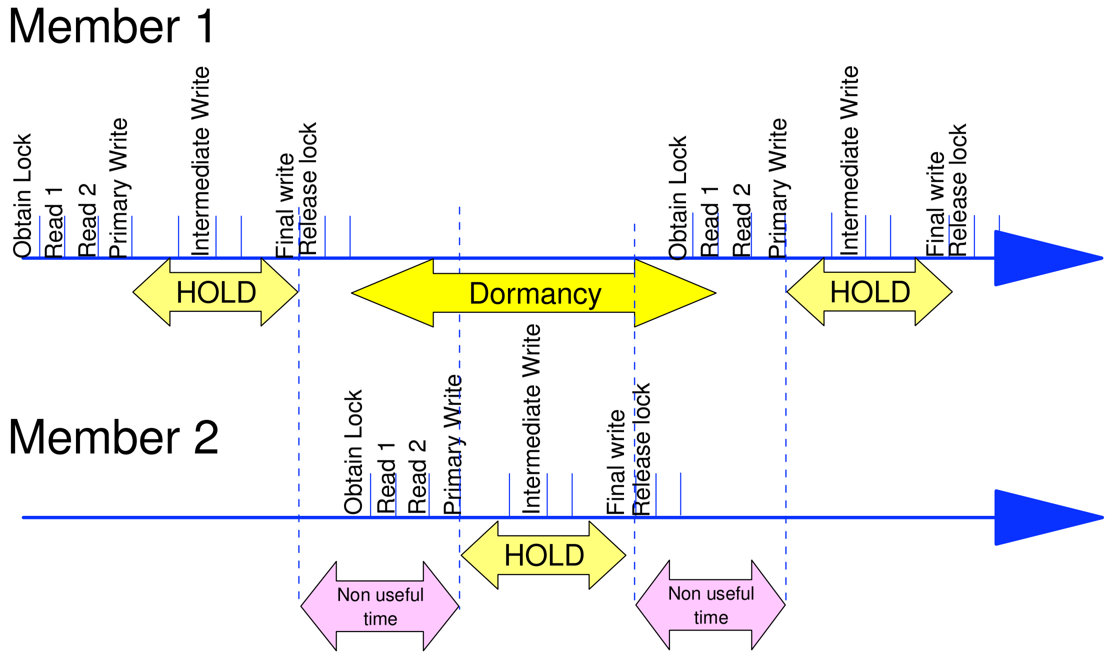
- Set **TRKCELL=** to a full track
  - **D/T 3380 -** 10 buffers per track (BUFSIZE=3992)
  - **D/T 3390 -** 12 buffers per track (BUFSIZE=3992)
  - Take advantage using **PRTnn TRKCELL=YES**
- Spool fencing tradeoff:
  - **FENCE=NO** - allows multiple parallel reads to occur on different volumes
  - **FENCE=YES** - limits the impact of a spool failure
- Avoid placing other data sets on the JES2 SPOOL volumes

© IBM Corporation 1998, 1999

"Bad" defaults (2)

- **JOBCLASS SWA=BELOW**
  - Introduced by **OY02574** in **1986**
  - **SWA=ABOVE** was "new function", so the default was set to be compatible with the "old way"
    - Applications needed to be checked to ensure they work when SWA=ABOVE
    - Most (not all) current applications work with SWA=ABOVE
  - Specify SWA=ABOVE for most job classes, use SWA=BELOW only when needed
- **JOBCLASS JOURNAL=YES**
  - Expensive performance when not needed

The Checkpoint Cycle



© IBM Corporation 1998, 1999

CKPTDEF parameters

- **CKPT on DASD or CF?**
  - CKPT1 - use either DASD or CF
    - Use CF in large MAS for locking fairness
  - CKPT2 - recommend using DASD due to persistent nature of the data
- **DUAL or DUPLEX mode?**
  - If CKPT on CF, must run in DUPLEX mode
  - If single member, run in DUPLEX mode
  - DUAL mode is ideal for small MAS

© IBM Corporation 1998, 1999

HOLD and DORMANCY

- **MASDEF HOLD** is the amount of time we'll hold the checkpoint
- **MASDEF DORMANCY=(*mindorm*,*maxdorm*)**
  - *mindorm* - the minimum amount of time we'll wait before accessing the checkpoint again
    - Allows other members to get checkpoint
  - *maxdorm* - the maximum amount of time we'll wait
    - Ensures that new work is seen and processed throughout MAS
- Checkpoint cycle of about 2 seconds is generally desirable (but may vary)
  - Single member - Set **HOLD=99999999**

Availability

- To minimize the impact of a checkpoint error:
  - Always run with 2 checkpoint datasets (**CKPT1** and **CKPT2**)
    - CKPT2 is cheap insurance against CKPT1 failure
    - Define on separate DASD/control unit/channel
  - Always define replacement checkpoint datasets (**NEWCKPT1** and **NEWCKPT2**)
  - Specify **OPVERIFY=NO** (default is **YES**) to automatically forward the checkpoint if there is an error.
    - Prevents checkpoint reconfiguration dialog WTOR from stopping all activity for long periods of time
    - Only used if specified on **all** members

© IBM Corporation 1998, 1999

Availability

- Use **MASDEF RESTART=YES,AUTOEMEM=YES** to automatically free checkpoint lock and requeue jobs in the event a JES2 member fails
  - **OW32320** changed processing so automatic freeing of lock always occurs
- Avoid placing other data sets on JES2 checkpoint volumes
- Update init deck after checkpoint forwarded

© IBM Corporation 1998, 1999

Availability

- Define backup JES2 proc and init deck to be used in case of a problem with the production versions.
  - **S** *proc-name*,**JOBNAME=JES2**
  - Test by starting secondary JES2 system
- Implement automation to initialize JES2 environment, but:
  - Define alternate **IEFSSNxx** or **COMMNDxx** to allow MVS to start without automatically restarting JES2
  - Ensure that automated responses to JES2 messages can be turned off
  - Don’t put a **$S** command in your init deck

© IBM Corporation 1998, 1999

Devices

- Spread JES2 devices across multiple MAS members to reduce impact of system loss
- Define devices on all members to allow them to be switched in an extended outage
  - Homogeneous JES2 definitions
  - Homogeneous HCD definitions
- Monitor devices (especially NJE, RJE, unattended devices) to ensure they remain ready
  - Use automation to restart them
  - Separate from IPL automation
- Define multiple transmitters and receivers on NJE lines

© IBM Corporation 1998, 1999

Monitoring JES2

- Use **$HASP050** and various **WARN=** parameters to monitor resource utilization
  - Default of **80%**
  - Avoid using **WARN=0** (no monitoring) or >90%
  - Define more resources than you need (at least slightly more) to deal with "spikes" and growth over time
- **$D JOBDEF**, **$D OUTDEF**, etc. can also be used to monitor resource utilization
- **$D PERFDATA** can be used to isolate PCEs with performance problems
  - Use **L=Z** to get complete output (prior to R7)

© IBM Corporation 1998, 1999

Monitoring JES2

- **$TRACE(17)** for checkpoint performance
- **$TRACE(20,30,31)** for **$#GET**, **$#POST**, and **$QGET** services
  - APARs **OW39179** and **OW39672**
  - These services are CPU-intensive while holding $QSUSE and may affect actual checkpoint HOLD times
  - **WS=** parameters on devices affect the performance of these services
    - Avoid speciying **P** (priority) in list
  - Size of queues affect these services - shorten queues by utilizing all 36 job/output classes

© IBM Corporation 1998, 1999

$D PERFDATA

- **$D PERFDATA(INITSTAT)**

```
$HASP660 STATISTICS FROM INITIALIZATION:
$HASP660 ROUTINE=MVSSTART,TIME=2.757341,CPU=0.001346,
$HASP660 ROUTINE=LOADINIT,TIME=0.077567,CPU=0.001112,
$HASP660 ROUTINE=IRMODCHK,TIME=0.000130,CPU=0.000129,
$HASP660 ROUTINE=IROPTS,TIME=0.046876,CPU=0.003310,
$HASP660 ROUTINE=IRSSI,TIME=0.722329,CPU=0.020674,
$HASP660 ROUTINE=IRSETUP,TIME=3.153794,CPU=0.060517,
$HASP660 ROUTINE=IRENF,TIME=0.048134,CPU=0.000549,
$HASP660 ROUTINE=IRPL,TIME=12.079740,CPU=1.397231,
$HASP660 ROUTINE=IRPOSTPL,TIME=2.141273,CPU=0.011836,
$HASP660 ROUTINE=IRDCTDCB,TIME=0.033853,CPU=0.003910,
$HASP660 ROUTINE=IRURDEV,TIME=0.000039,CPU=0.000037,
$HASP660 ROUTINE=IRNJE,TIME=8.486918,CPU=6.078407,
$HASP660 ROUTINE=IRRJE,TIME=1.314377,CPU=0.179421,
$HASP660 ROUTINE=IREMVS,TIME=0.287586,CPU=0.033267,
$HASP660 ROUTINE=IRDA,TIME=13.935658,CPU=0.632235,
$HASP660 ROUTINE=IRCSA,TIME=0.024407,CPU=0.000504,
$HASP660 ROUTINE=IRDCTCP,TIME=0.037114,CPU=0.000263,
$HASP660 ROUTINE=IRMVS,TIME=0.877603,CPU=0.003907,
$HASP660 ROUTINE=IRPCE,TIME=0.263991,CPU=0.005193,
$HASP660 ROUTINE=IRINFO,TIME=0.000023,CPU=0.000021,
$HASP660 ROUTINE=IRFINAL,TIME=0.000760,CPU=0.000551,
$HASP660 ROUTINE=WARMSTRT,TIME=0.366896,CPU=0.157284
```

$D PERFDATA

- **$D PERFDATA(QSUSE)**

```
$HASP660 $QSUSE UTILIZATION STATISTICS - INTERVAL=12:04.686250,
$HASP660 MOD=HASPJQS,SEQ=03330000,COUNT=8,AVGWAIT=0.738824,
$HASP660 MOD=HASPJQS,SEQ=46496600,COUNT=7,AVGWAIT=0.729787,
$HASP660 MOD=HASPXEQ,SEQ=03510000,COUNT=6,AVGWAIT=0.966112,
$HASP660 MOD=HASPJOS,SEQ=65460000,COUNT=5,AVGWAIT=1.007884,
$HASP660 MOD=HASPJOS,SEQ=26500900,COUNT=5,AVGWAIT=0.993899,
$HASP660 MOD=HASXDYNT,SEQ=04890000,COUNT=4,AVGWAIT=0.991853,
$HASP660 MOD=HASPJOS,SEQ=02055000,COUNT=4,AVGWAIT=1.014605,
$HASP660 MOD=HASPTRAK,SEQ=04479900,COUNT=4,AVGWAIT=1.017079,
$HASP660 MOD=HASPSPIN,SEQ=05463000,COUNT=2,AVGWAIT=1.226588,
$HASP660 MOD=HASPJOS,SEQ=75960000,COUNT=2,AVGWAIT=0.517071,
$HASP660 MOD=HASPMISC,SEQ=75282000,COUNT=1,AVGWAIT=0.210015,
$HASP660 MOD=HASPNJT,SEQ=17172000,COUNT=1,AVGWAIT=0.277836,
$HASP660 MOD=HASPJOS,SEQ=18635400,COUNT=1,AVGWAIT=0.239419,
$HASP660 MOD=HASPJOS,SEQ=23699300,COUNT=1,AVGWAIT=1.015909
```

- Details all $WAITs for checkpoint over interval, independent of PCE type
- **$T PERFDATA,RESET** to start new interval

$D PERFDATA

- **$D PERFDATA(PCESTAT)**

```
$HASP660 PCE PERFORMANCE STATISTICS - INTERVAL=18:09.243933,CPU=2.262742,
$HASP660 PCENAME=CKPT,TIME=2.650084,CPU=1.714688,CPU%=75.77,
$HASP660 QSUSE_TIME=0.005741,IOCOUNT=1306,CKPT_COUNT=761,
$HASP660   WAIT=CKPTW,INHIBIT=NO,MOD=HASPCKPT,SEQ=06730000
$HASP660    COUNT=363,AVGWAIT=0.100818,
$HASP660      POST=RESOURCE,COUNT=132,AVGWAIT=0.003424,
$HASP660      POST=HOLD,COUNT=1,AVGWAIT=0.533140,
$HASP660      POST=WORK,COUNT=230,AVGWAIT=0.154834,
$HASP660   WAIT=POST,XECB,MOD=HASPCKPT,SEQ=09186000
$HASP660    COUNT=18,AVGWAIT=0.000759,
$HASP660      POST=XECB,COUNT=18,AVGWAIT=0.000759,
$HASP660   WAIT=POST,XECB,MOD=HASPCKPT,SEQ=11848000
$HASP660    COUNT=1,AVGWAIT=0.001449,
$HASP660      POST=XECB,COUNT=1,AVGWAIT=0.001449,
...
```

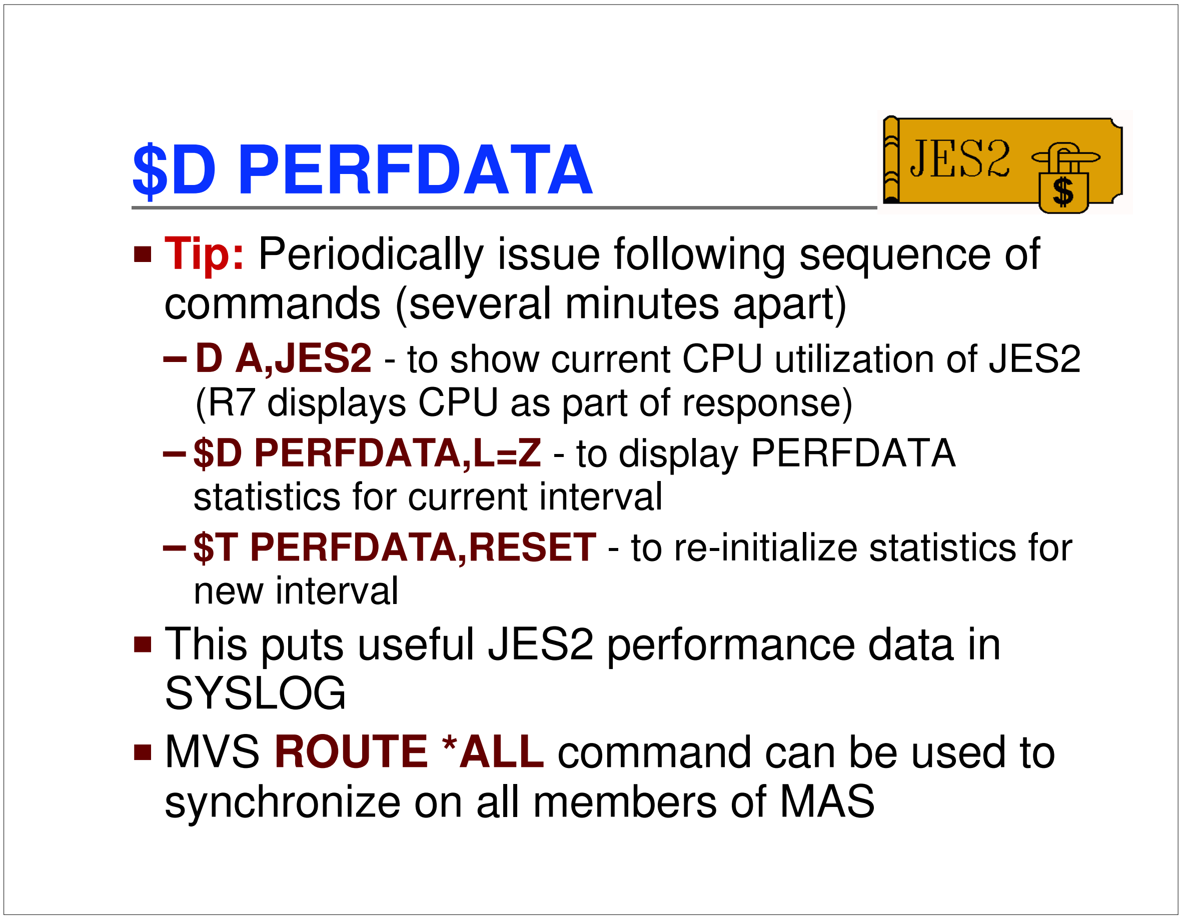
- **$D PERFDATA(PCESTAT), PCENAME=***xxxx* for specific PCE type
- Indicates CPU time and % of main task over interval
- Details all $WAITs done by PCE type, avg time, type of post

$D PERFDATA

- **$D PERFDATA(SAMPDATA)**

```
$HASP660 SERVICE CLASSES KNOWN TO JES2:
$HASP660 SRVCLASS(HOTPRIME)=(TOKEN=0F4A8000,REGISTERED,SYSTEMS=(AQFT,AQTS)),
$HASP660 SRVCLASS(NRPRIME)=(TOKEN=104A8000,REGISTERED,SYSTEMS=(AQFT,AQTS)),
...
$HASP660 SERVICE CLASS SAMPLING DATA:
$HASP660 SRVCLASS(15)=(SYS_QUEUE=0,SYS_INEL=18,SYS_LIMIT=0,
$HASP660 LOCAL_QUEUE=0,LOCAL_INEL=18),
$HASP660 SRVCLASS(16)=(SYS_QUEUE=0,SYS_INEL=2,SYS_LIMIT=0,
$HASP660 LOCAL_QUEUE=0,LOCAL_INEL=2)
$HASP660 REPORT CLASS SAMPLING DATA:
...
```

- Displays information about service classes known to JES2
- Represents exact sampling data passed to WLM
- Can be used to isolate problem to JES2 or WLM



$D PERFDATA

- **Tip:** Periodically issue following sequence of commands (several minutes apart)
  - **D A,JES2** - to show current CPU utilization of JES2 (R7 displays CPU as part of response)
  - **$D PERFDATA,L=Z** - to display PERFDATA statistics for current interval
  - **$T PERFDATA,RESET** - to re-initialize statistics for new interval
- This puts useful JES2 performance data in SYSLOG
- MVS **ROUTE *ALL** command can be used to synchronize on all members of MAS

© IBM Corporation 1998, 1999

PCEs and subtasks

- Define maximum PURGE, SPIN, OUTPUT, and CNVT PCEs (**PCEDEF xxxNUM**)
  - Default is only 2 or 3 of each type, maximum is **10**
  - Overall performance may be improved
  - Minimizes impact of the failure of a single PCE
  - In some environments, increasing or decreasing number of PSO and STAC PCEs can improve (*or degrade!*) overall performance
- Define maximum general purpose subtasks (**SUBTDEF GSUBNUM=10**)
  - This is the default!

© IBM Corporation 1998, 1999

Duplicate jobs

- Duplicate job processing is expensive and can lead to delays in running jobs
  - Consider using **JOBDEF DUPL_JOB=NODELAY**
    - Introduced in R3; allows jobs with same name to run simultaneously
  - When jobs must run serially, consider either
    - A dedicated job class with **JOBCLASS(x) XEQCOUNT=(LIMIT=1)**
    - A job scheduling package
- **Don’t change without fully understanding your workload!**

Serviceability (and more "bad" defaults)

- Don’t use **RECVOPTS** to prevent getting dumps - use DAE instead
  - **RECVOPTS(MAIN) COUNT=2,INTV=24**
    - Number of failures (**COUNT**) within interval (**INTV**) at which operator becomes involved
    - Default changed from **2**/**24** to **2**/**1** in OS/390 R8
    - **Decrease interval** or **increase count** to get more dumps automatically
  - **RECVOPTS(xxx) COUNT=2,INTV=24**
    - Number of failures within interval at which JES2 stops taking dumps
    - **Decrease interval** or **increase count** to get more dumps automatically

© IBM Corporation 1998, 1999

Serviceability

- **DEBUG** is a tradeoff
  - **DEBUG=YES** provides the best diagnostics but can impact performace (e.g. **CKPT=YES**) or availability (e.g. **BERT=YES**)
    - Normally performance or availability impacts are not severe
  - **DEBUG=NO** has minimum impact to performance and availability, but prevents meaningful diagnostics for certain errors
  - Avoid specifying **DEBUG=YES** or **DEBUG=NO;** that affects ALL JES2 debug options
    - Use **DEBUG=(xxxx=YES/NO)**
    - Default is a mix of YES and NO

Miscellaneous

- Specify **REGION=0** for JES2 and monitor JES2 storage usage
  - JES2 uses lots of UNBACKED virtual storage
- Avoid init statements using a subscript of '*' (for example NODE(*))
  - Unnecessary overhead processing undefined elements, causes slow initialization which can extend an outage
- ***Keep up to date on maintenance***
  - Latest RSU level if possible
  - HIPERs
- ***Read the PSP bucket***