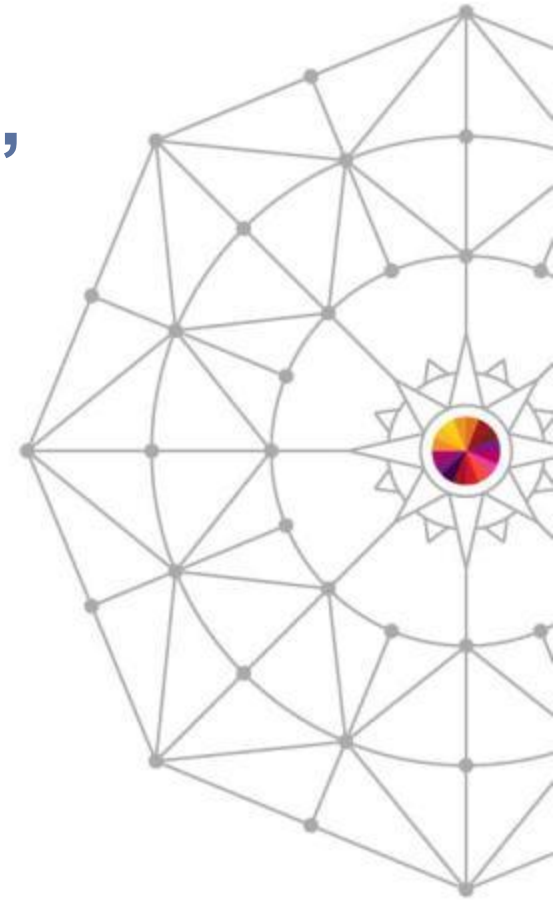
#SHAREorg

SHARE
Technology · Connections · Results

Automation for IMS: Why It's Needed, Who Benefits, and What Is the Impact?

Duane Wente
BMC Software

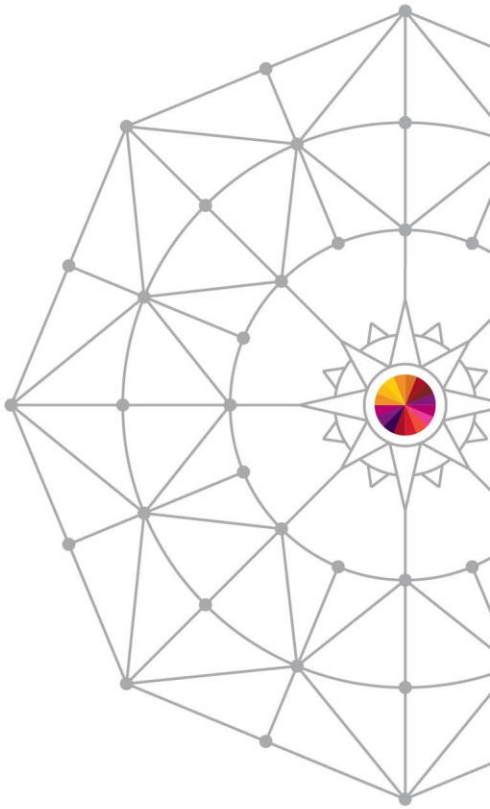
Monday, March 10, 2014, 11:00am
Session: 14771



SHARE in Anaheim

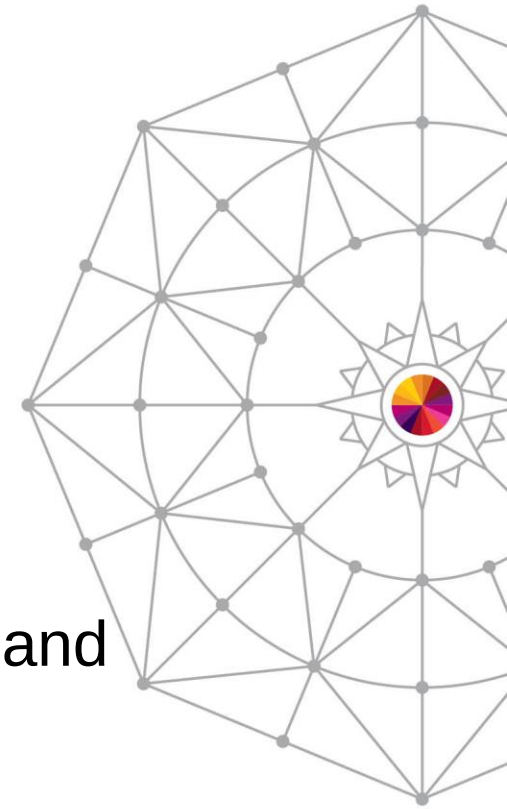
Agenda

- Policy driven database management
- Dynamic application optimization
- Tight integration changes work effort



Need for Automation - Why

- Growing quantity of IMS data
- Fewer IMS experts
- New engineers supporting IMS
  - Windows fluent
  - Green screen avoiders
- Need to capture 46 years of IMS knowledge and pass it on quickly
- Constant pressure to reduce cost



DBA Requirements – Maintain Database Health

Take care of the databases

Number of databases to manage

Available window to implement changes

Lead time required to implement changes

Lower cost

Taking care of the databases

Availability

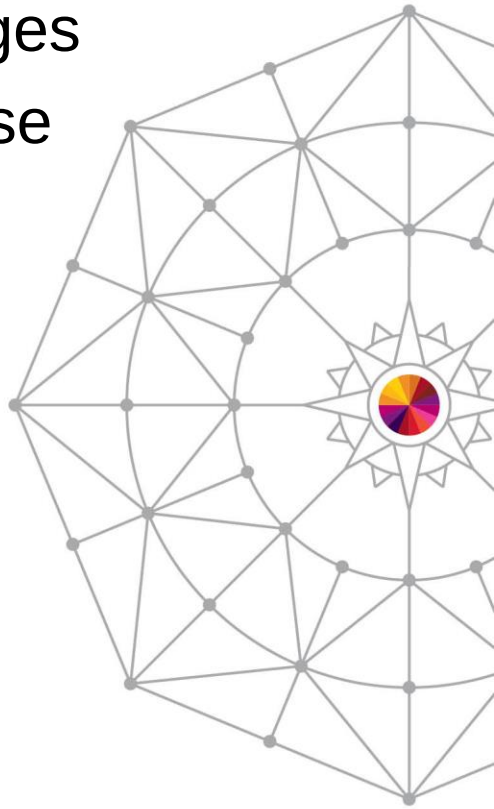
Performance

Recoverability

Minimal outage database change - Availability

The Problem

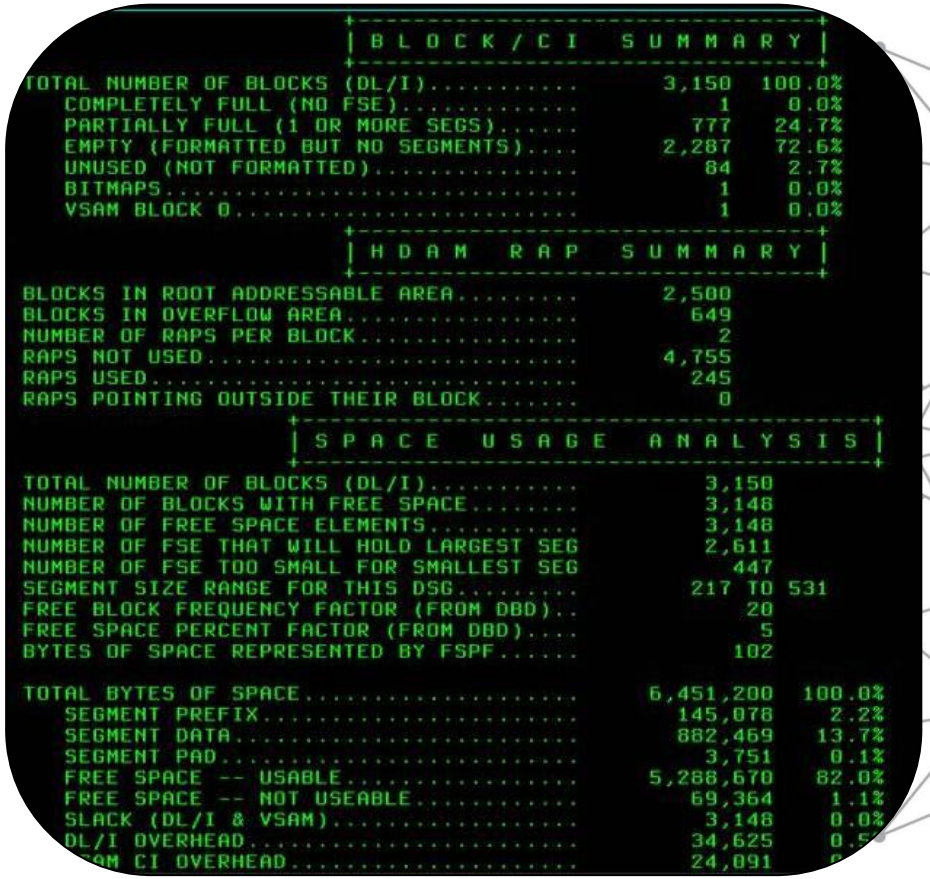
- Constant pressure to reduce number of outages
- Conflicts with the need to change the database
- Applications want additional availability
- DBA's/Systems want to maintain the change window



- The solution – Online Database Change
  - Minimal outage to the application
  - Change capture and apply technology
  - BMP coexistence
  - Resource management

Existing process for managing databases – metrics based

- Track multiple data points
- Correlate these data points
- Collect data
- Analyze data

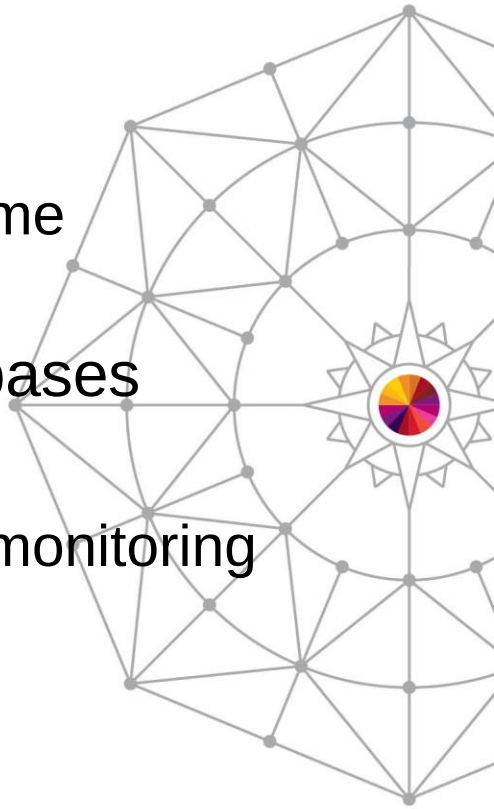


```
                    +-----------------------------+
                    | B L O C K / C I   S U M M A R Y |
                    +-----------------------------+
TOTAL NUMBER OF BLOCKS (DL/I).....        3,150  100.0%
   COMPLETELY FULL (NO FSE).......            1    0.0%
   PARTIALLY FULL (1 OR MORE SEGS).....     777   24.7%
   EMPTY (FORMATTED BUT NO SEGMENTS)....  2,287   72.6%
   UNUSED (NOT FORMATTED).............       84    2.7%
   BITMAPS............................        1    0.0%
   VSAM BLOCK 0.......................        1    0.0%
                    +-----------------------------+
                    | H D A M   R A P   S U M M A R Y |
                    +-----------------------------+
BLOCKS IN ROOT ADDRESSABLE AREA.....      2,500
BLOCKS IN OVERFLOW AREA.............        649
NUMBER OF RAPS PER BLOCK............          2
RAPS NOT USED.......................      4,755
RAPS USED...........................        245
RAPS POINTING OUTSIDE THEIR BLOCK...          0
                    +-----------------------------------+
                    | S P A C E   U S A G E   A N A L Y S I S |
                    +-----------------------------------+
TOTAL NUMBER OF BLOCKS (DL/I).....        3,150
NUMBER OF BLOCKS WITH FREE SPACE....      3,148
NUMBER OF FREE SPACE ELEMENTS.......      3,148
NUMBER OF FSE THAT WILL HOLD LARGEST SEG  2,611
NUMBER OF FSE TOO SMALL FOR SMALLEST SEG    447
SEGMENT SIZE RANGE FOR THIS DSG.....  217 TO 531
FREE BLOCK FREQUENCY FACTOR (FROM DBD)..     20
FREE SPACE PERCENT FACTOR (FROM DBD)....      5
BYTES OF SPACE REPRESENTED BY FSPF.....     102

TOTAL BYTES OF SPACE................  6,451,200  100.0%
   SEGMENT PREFIX...................    145,078    2.2%
   SEGMENT DATA.....................    882,469   13.7%
   SEGMENT PAD......................      3,751    0.1%
   FREE SPACE -- USABLE.............  5,288,670   82.0%
   FREE SPACE -- NOT USEABLE........     69,364    1.1%
   SLACK (DL/I & VSAM)..............      3,148    0.0%
   DL/I OVERHEAD....................     34,625    0.5%
   VSAM CI OVERHEAD.................     24,091
```

Consider policy based database management

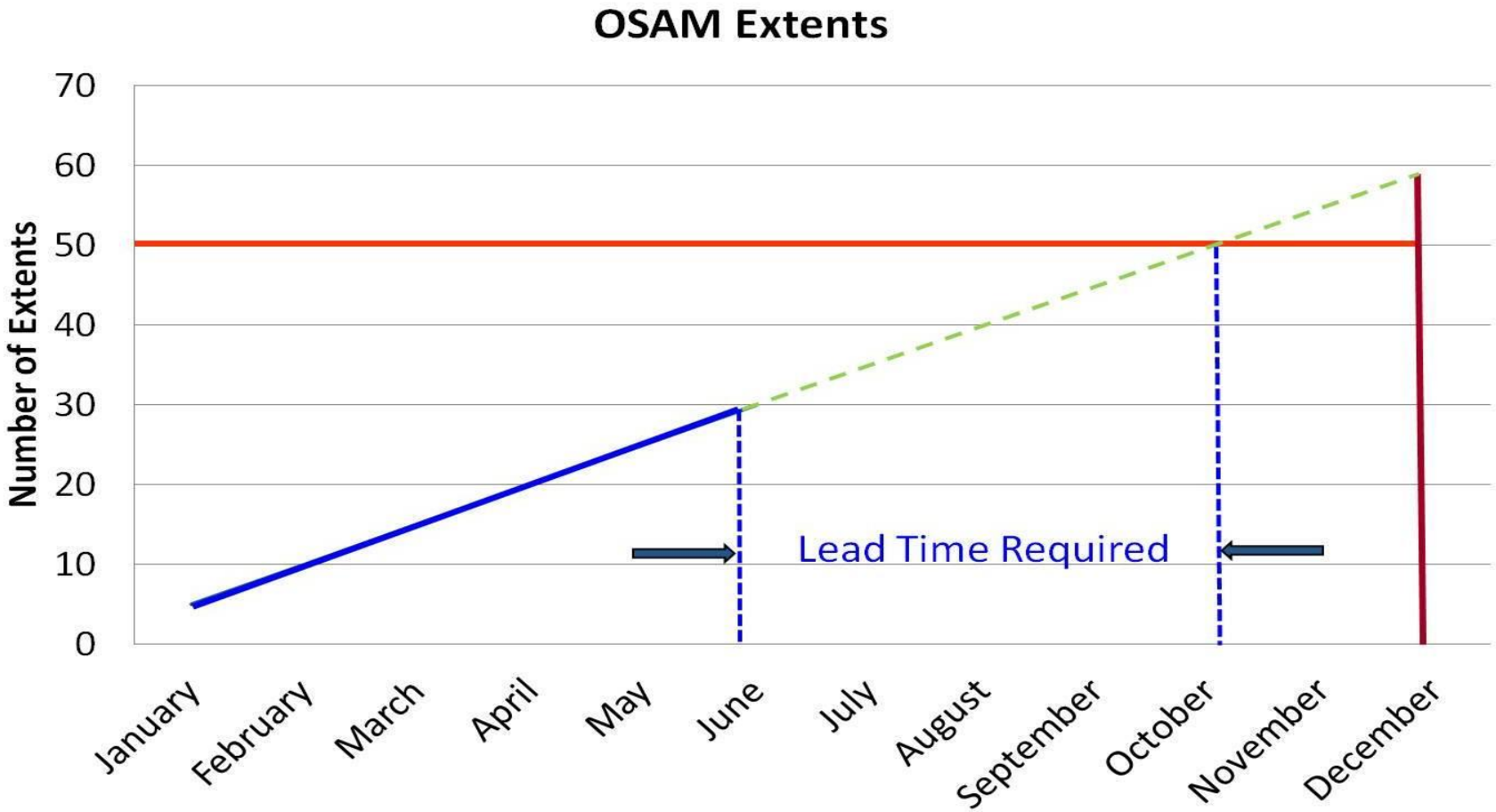
- Lead time required to implement a change
  - Database reorg may need 2 week lead time
  - Database change may need a 4 month lead time
- How frequently do you need to monitor databases
  - DEDB's may need to be monitored every hour
  - Database storing historical data once a week monitoring



Taking care of your databases - Space

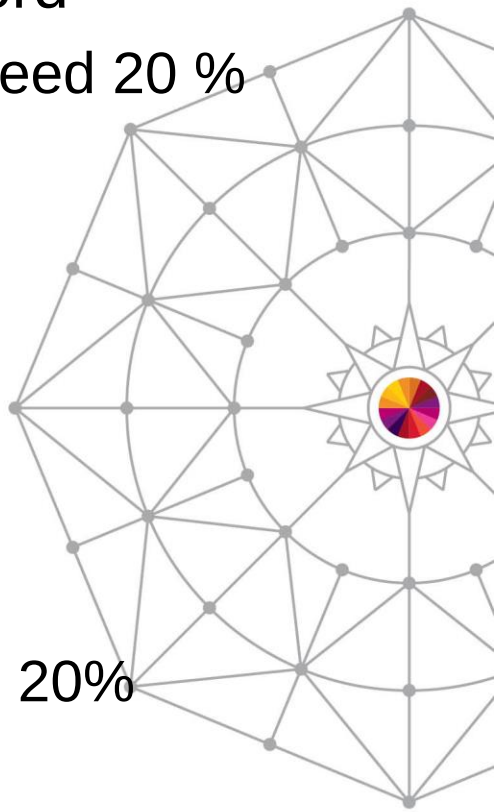
- My databases should have at least “x” free space
  - As example – all databases should have 20% freespace
- My database data sets should not be bigger than “Y” GB
  - As example – all data sets should be less than 3.5 GB
- My database data sets should not have more than “Z” extents
  - As example – all data sets should have less than 50 extents

Putting it together

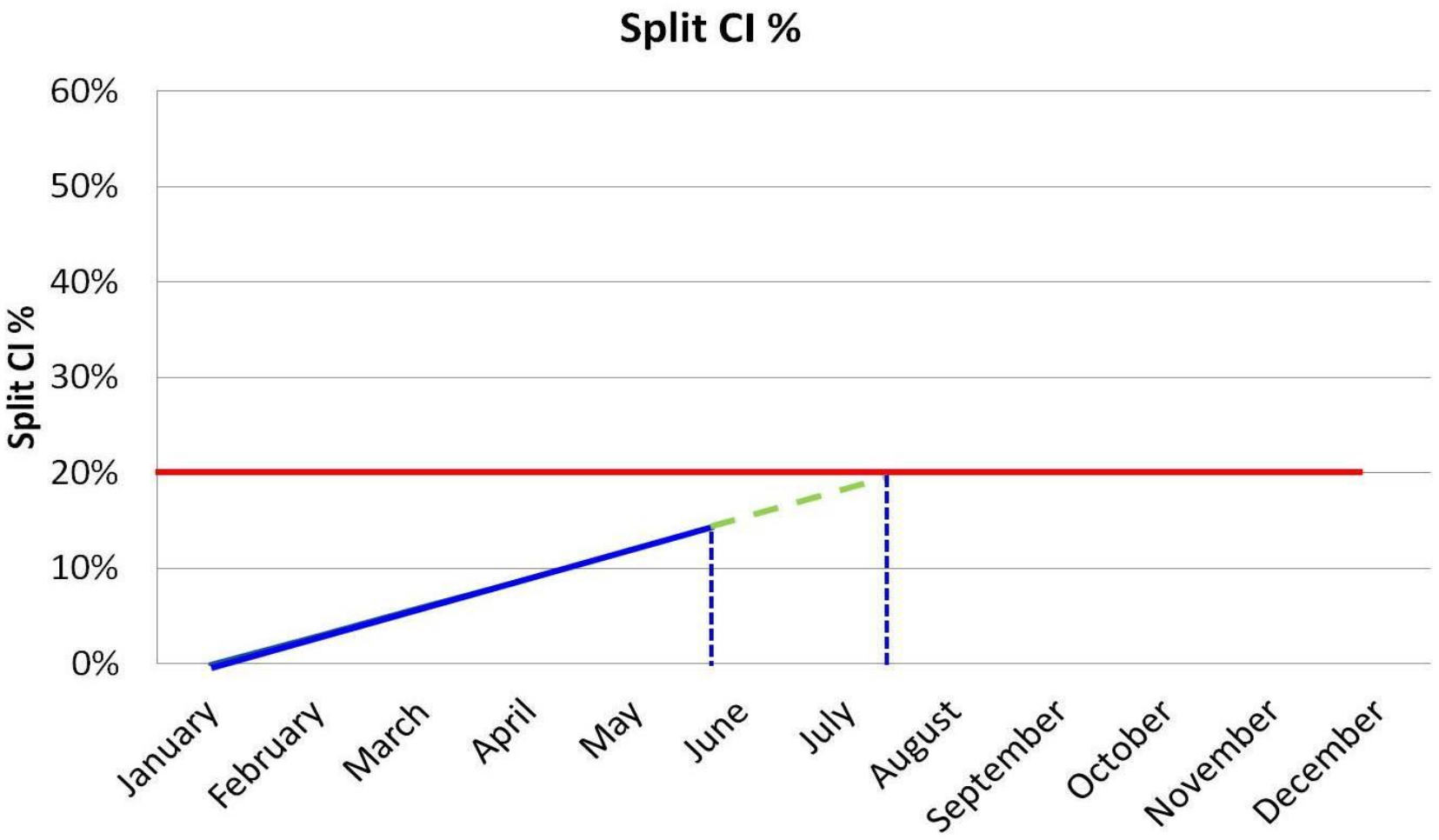


Taking care of your databases - Performance

- How many I/Os do you need to retrieve a record
  - As example - The growth in I/O should not exceed 20 %
- How many CI/CA splits do I have
  - As example – The % of split CIs should not exceed 20 %
- How are my randomizing parameters
  - As example – The parameters should be within 20% of optimal

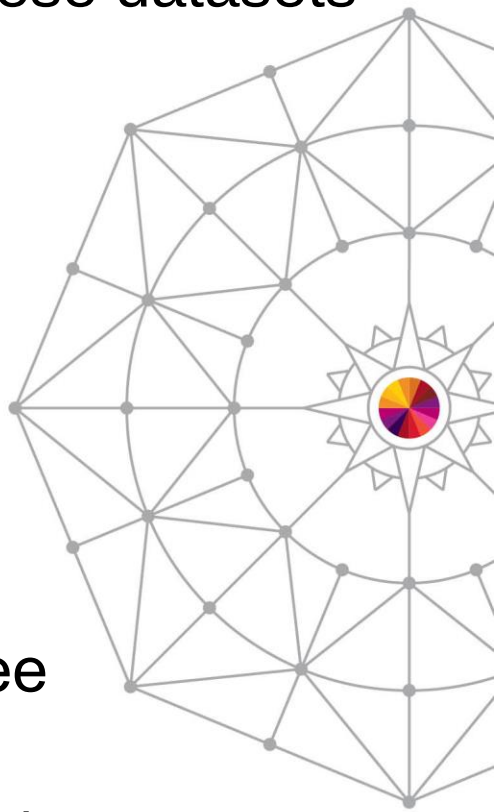


Same Concept for Performance Parameters



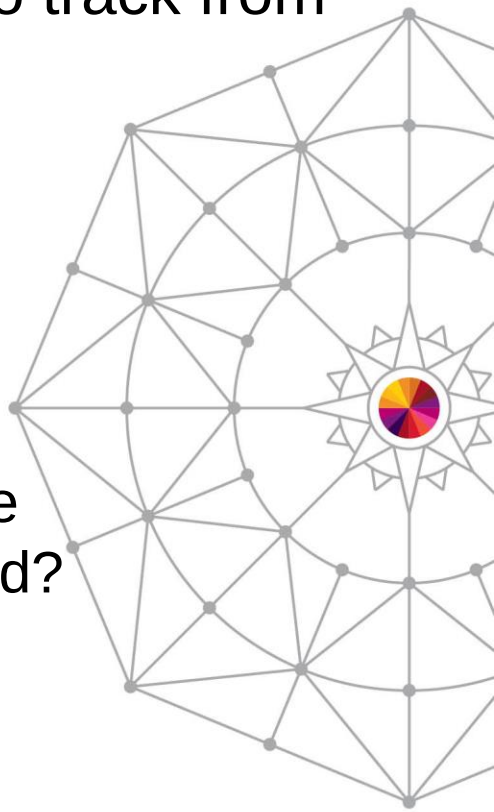
Taking care of your databases - Recoverability

- RECONS – IMS recovery revolves around these datasets
  - Monitor the health of the RECONs
- My RECONs should have less than “X” % CI/CA splits
  - As example – The % of split CIs should not exceed 20 %
- My RECONS should have “Y” % allocated free space
  - As example – The allocated free space should be 15 % or more



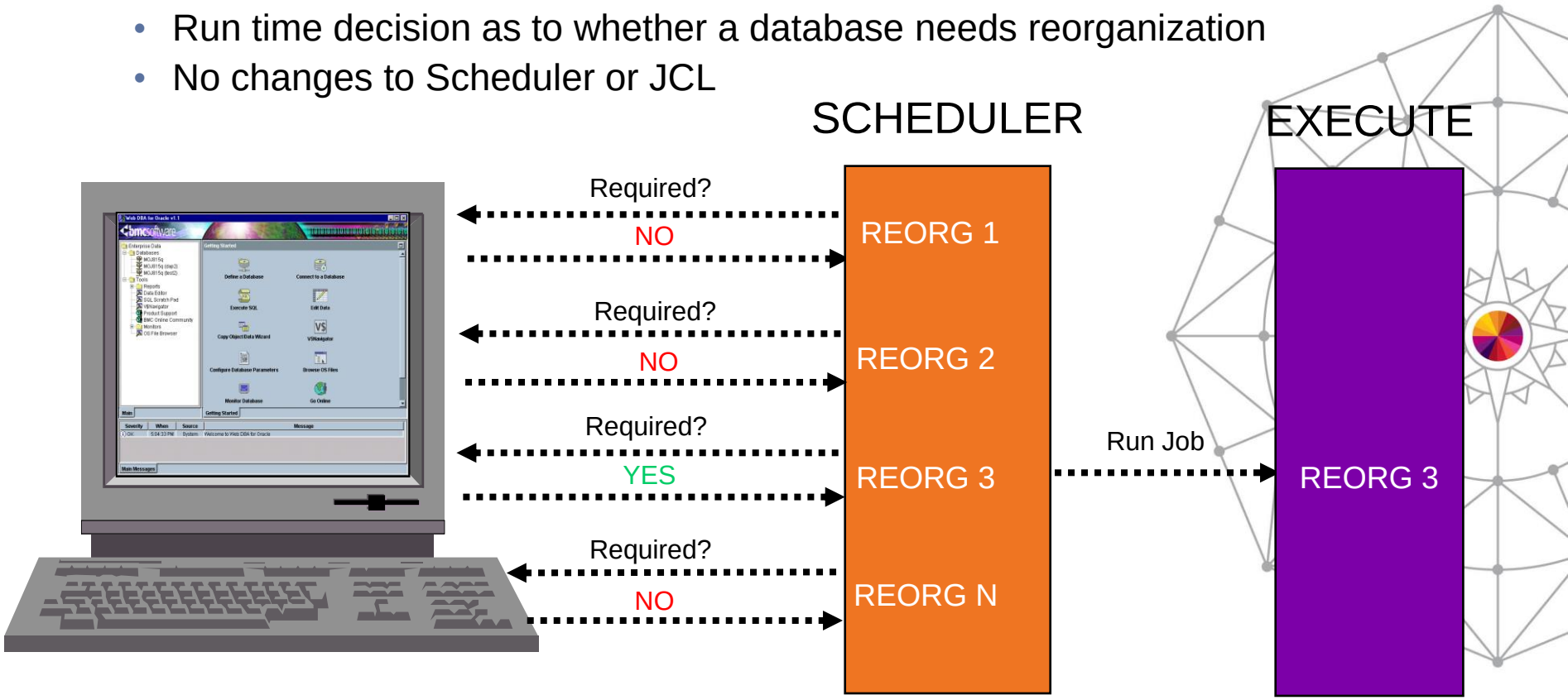
Taking care of your databases - Recoverability

- Recovery Conditions – select the conditions to track from the RECON
  - As example – database marked as IC needed
- Recovery assets - can I perform a successful recovery
  - As example – Are all my image copies, change accum datasets and IMS log datasets cataloged?
- Manage the CA & DBDS groups
  - As example – Take an image copy when CA dataset size is too large



Lower Cost - Conditional Reorganization

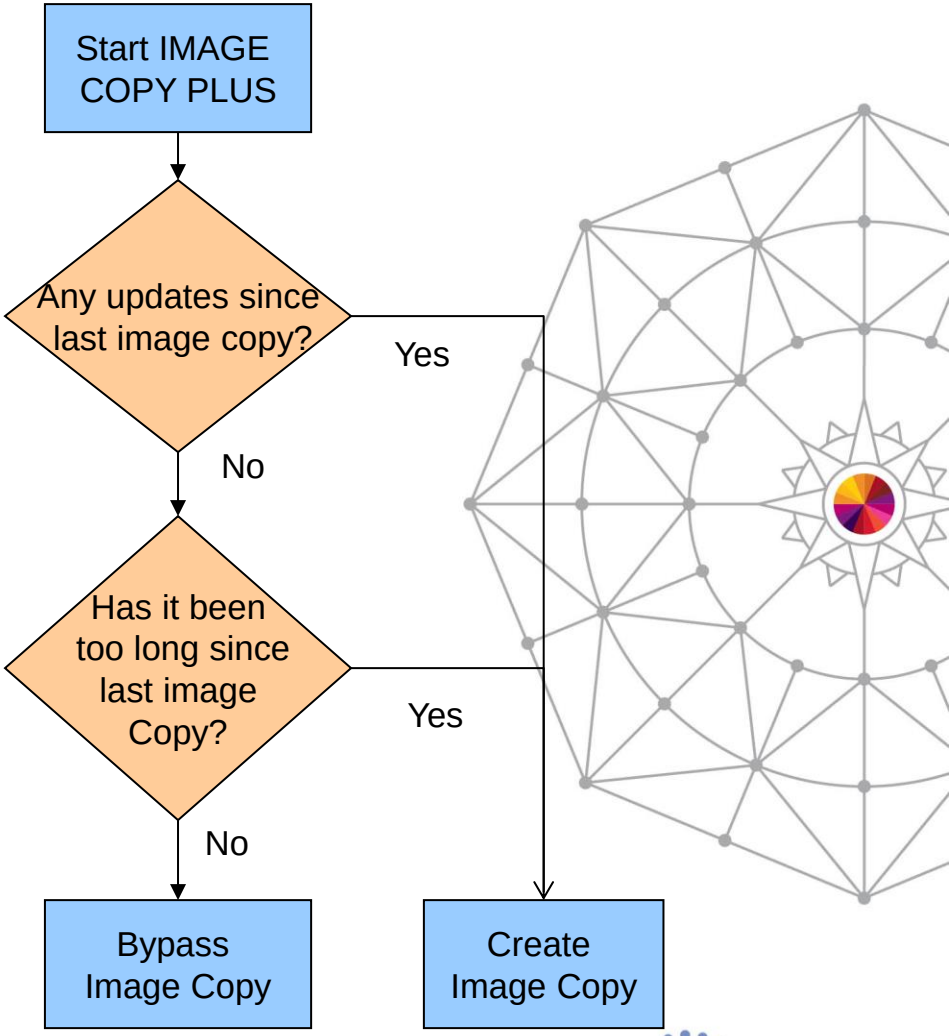
- The Problem – Database reorganizations that do not need to run
- The Solution – Conditional Reorganization
  - Run time decision as to whether a database needs reorganization
  - No changes to Scheduler or JCL



Reorganizes only the databases that need to be reorganized

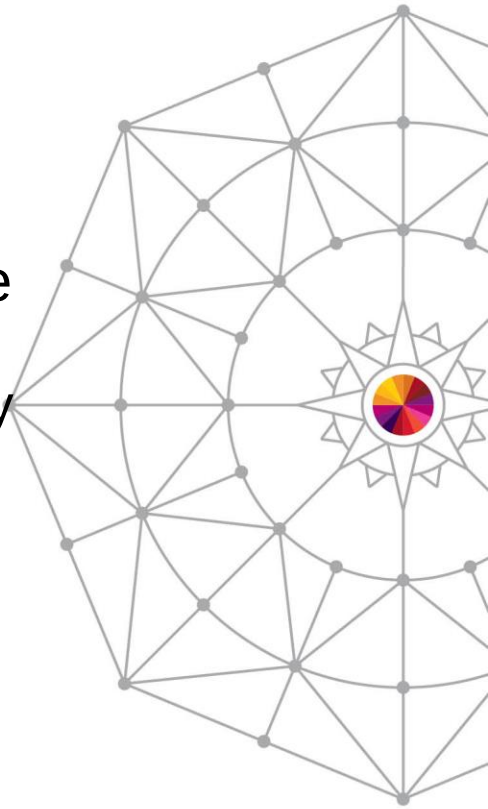
Lower Cost- Conditional Image Copy

- The Problem -
  - Am I taking too many batch image copies
  - Can I save money without changing the scheduler
- The Solution –
  - Conditional Image Copy



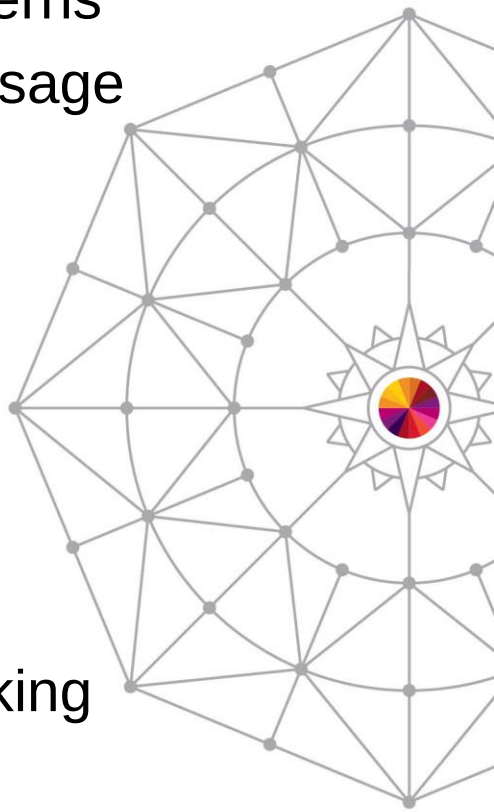
Policy based database management - Summary

- You decide what you need
  - Lead time
  - Monitoring frequency
  - Database Thresholds
- You are presented with a list of objects that violate the policy
  - Smaller number of databases that you need to worry about
  - Enough lead time to implement your changes
- A tool that automates this process will ensure:
  - You can manage your databases proactively
  - No database falls through the cracks
  - ISV Tools available to help with your automation process



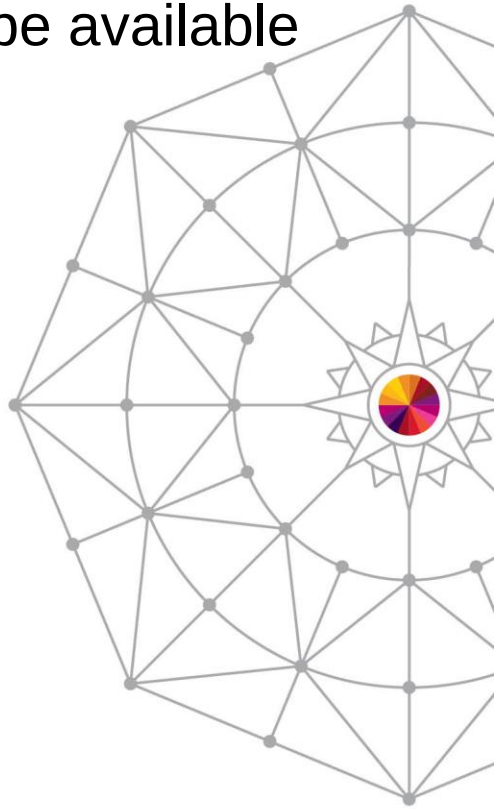
Application Program Tuning

- Peak usage occurs more and more during batch windows
  - Mobile devices are driving different usage patterns
  - Research shows that the time of day of peak usage has changed
- Volume of data is increasing
  - Amount of data in IMS continues to grow
- You need to improve throughput
  - The time available to process the data is shrinking
  - The amount of data to process is increasing



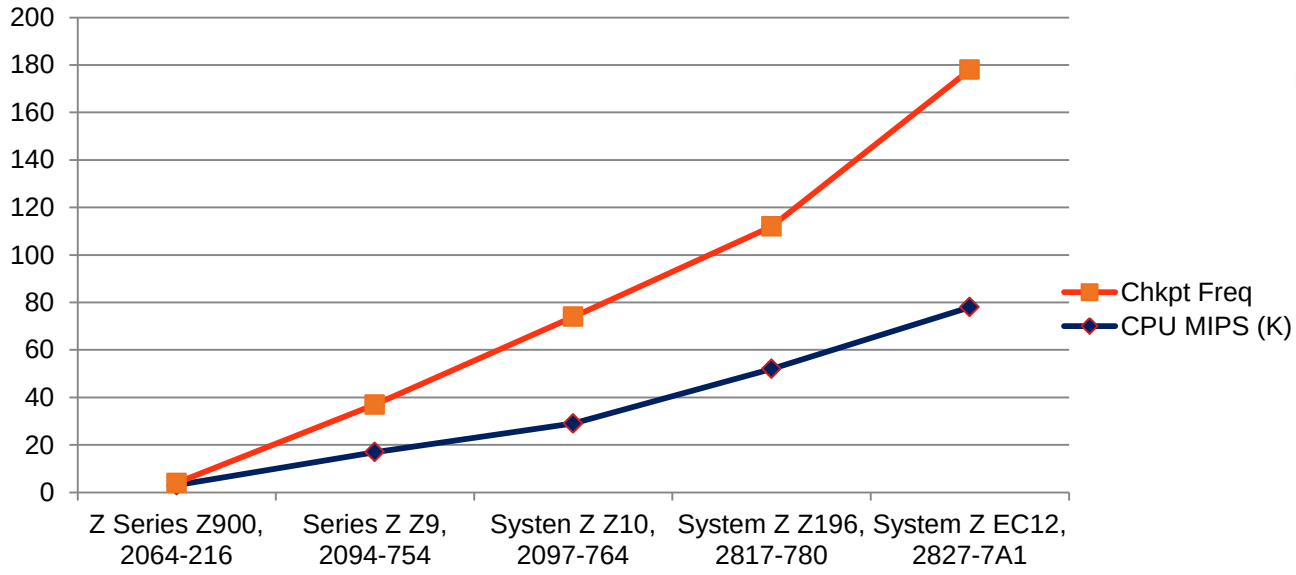
Requirements for potential solutions

- Changing application programs might not be feasible
  - People familiar with the applications might not be available
- The solution needs to be scalable – lots of application programs
  - Policy based deployment e.g. Optimize all jobs starting with PAY*
- JCL changes will probably be frowned on
  - Dynamic implementation of improvements



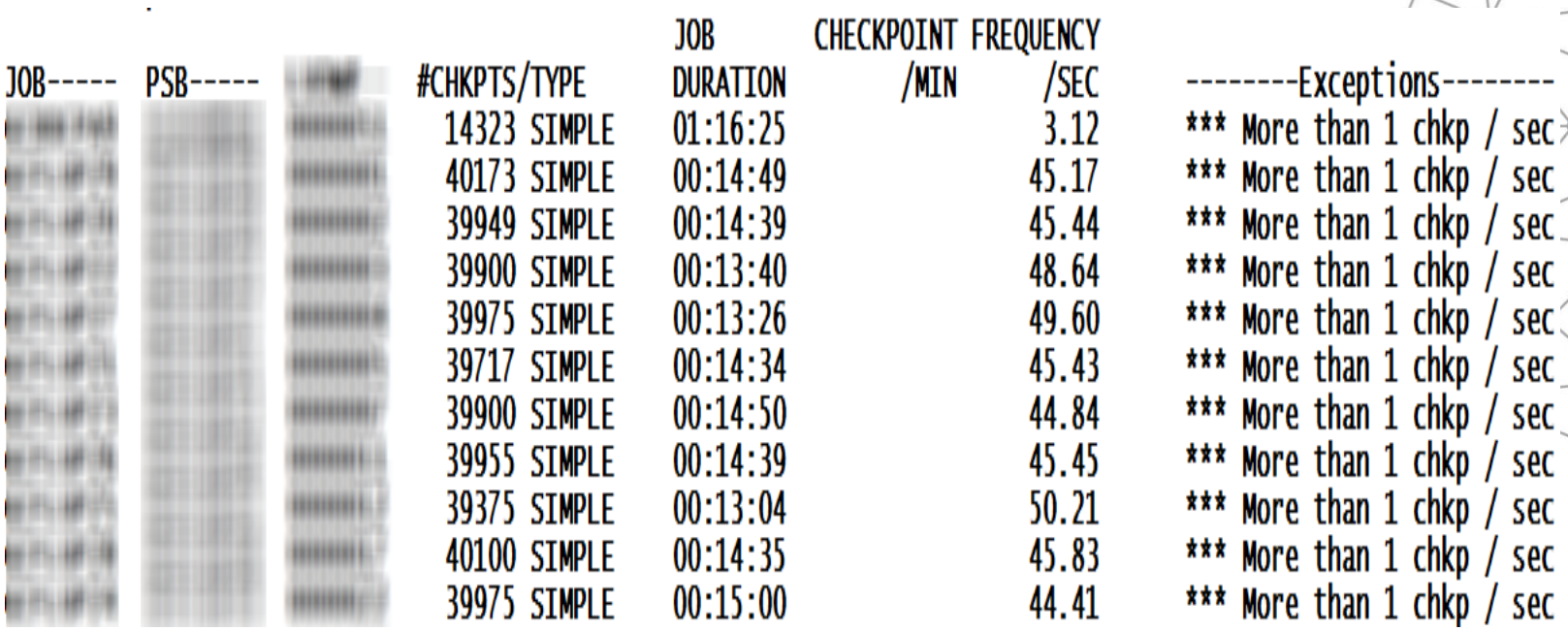
Hidden BMP Overhead Costs

- CPU MIPS rates Increase
- Checkpoint intervals decrease
- Excessive checkpoints adds to overhead costs



Problem Visibility

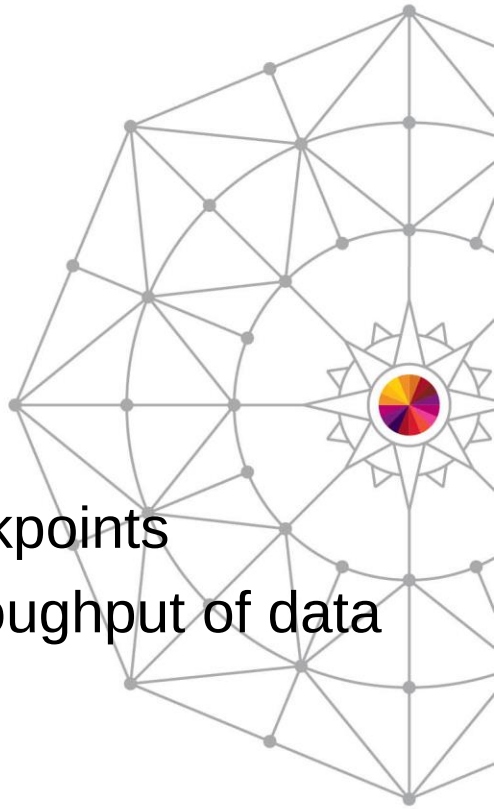
- BMP's run to completion
  - Out of Sight, Out of Mind
- BMP's run in the same time frame
- I don't want to change the application



JOB-----	PSB-----	[illegible]	#CHKPTS/TYPE	JOB DURATION	CHECKPOINT FREQUENCY /MIN	/SEC	--------Exceptions--------
[illegible]	[illegible]	[illegible]	14323 SIMPLE	01:16:25		3.12	*** More than 1 chkp / sec
			40173 SIMPLE	00:14:49		45.17	*** More than 1 chkp / sec
			39949 SIMPLE	00:14:39		45.44	*** More than 1 chkp / sec
			39900 SIMPLE	00:13:40		48.64	*** More than 1 chkp / sec
			39975 SIMPLE	00:13:26		49.60	*** More than 1 chkp / sec
			39717 SIMPLE	00:14:34		45.43	*** More than 1 chkp / sec
			39900 SIMPLE	00:14:50		44.84	*** More than 1 chkp / sec
			39955 SIMPLE	00:14:39		45.45	*** More than 1 chkp / sec
			39375 SIMPLE	00:13:04		50.21	*** More than 1 chkp / sec
			40100 SIMPLE	00:14:35		45.83	*** More than 1 chkp / sec
			39975 SIMPLE	00:15:00		44.41	*** More than 1 chkp / sec

Application Programs - Checkpoint Pacing

- The Problem – IMS checkpoint processing
  - Required, necessary evil
  - Extremely expensive – 100% overhead
  - Removing excessive checkpoint activity can provide significant run time improvements



- The Solution - Checkpoint Pacing
  - CPU Reduction – removes unnecessary checkpoints
  - Elapsed time Reduction – allow increased throughput of data
  - Policy based deployment

Application Programs - Buffer Tuning

- The Problem
  - DL/I Batch jobs usually run with a one size fits all buffer definition
  - It is not customized to volume of data
  - It is not customized to individual job call patterns

- The solution – Dynamic Application Tuning
  - Implement dynamic buffer tuning based on call volume and call pattern
  - Implement OSAM sequential buffering
  - Implement enhanced I/O techniques where possible
  - Policy based deployment
  - Delivers significant CPU and elapsed time savings

Application Programs – BMP Deadlock Reduction

- The Problem
  - BMP jobs abending with U0777
  - Issues with scheduler restart

- The solution – Dynamic Deadlock Reduction
  - Implement a reattach solution
  - No scheduler requirements
  - Does not terminate the BMP, but delays reattach until most conflicts are circumvented
  - Operational savings
  - FTE Savings

Summary

- Why? Use policy based database management
  - Consistent - no matter how many databases
  - Effective - “the right work at the right time”
- Who? DBA’s, Applications
- What? Online Change, Database Management, Recovery, and Batch Optimization

Duane Wente
duane_wente@bmc.com