Why did my job run so long?
Speeding Performance by Understanding the Cause

John Baker
MVS Solutions
jbaker@mvssol.com
Agenda

• Where is my application spending its time?
  – CPU time, I/O time, wait (queue) times

• What am I waiting for?
  – Various flavors of queue time
  – What can/should I do about delays?

• Real world comparison
  – Stay tuned! 😊

• Q/A

• Conclusions and wrap up
Distribution of Elapsed Time

Elapsed time = CPU time + I/O time + wait times

- CPU time = TCB + SRB
- I/O = IOSQ + PEND + CON + DISC
- Wait (queue) times
  - Initiator
  - Allocation (ENQ contention)
  - System services (HSM recall)
  - CPU Delay
  - LPAR dispatch
  - ...

ThruPut Manager
Sample Job A

<table>
<thead>
<tr>
<th>JOB</th>
<th>RUNTM</th>
<th>CPUTM</th>
<th>IOTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBA</td>
<td>4:23:53</td>
<td>0:48:21</td>
<td>0:09:12</td>
</tr>
</tbody>
</table>

- Elapsed time over 4 hours
- CPU time almost 1 hour
- I/O time under 10 minutes

= Focus on CPU time
Reducing CPU time

• Recompile
  – Many improvements in OS updates

• Tune application
  – Application Performance Tools
    • (e.g. Strobe, FreezeFrame)
  – Identify CPU use by area of source code
  – Make friends with your developer 😊
Sample Job B

<table>
<thead>
<tr>
<th>JOB</th>
<th>RUNTM</th>
<th>CPUTM</th>
<th>IOTIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOBB</td>
<td>2:41:30</td>
<td>0:21:20</td>
<td>1:37:44</td>
</tr>
</tbody>
</table>

• Elapsed time under 3 hours
• CPU time 20 minutes
• I/O time over 1.5 hours

= Focus on I/O time
Reducing I/O time

• Identify patterns
  – sequential vs random; read vs write

• Buffers
  – For VSAM consider NSR vs LSR
  – Give SORT memory – but not too much!

• Block size
  – System-determined generally works well – but check!
  – Half track for sequential; No smaller than 2K for random

• Compression
  – zEDC sounds promising!

• Include Storage Subsystem in your capacity planning
Wait/Queue time
High Utilization = High wait time

- At high utilization levels, wait time is much greater than service time
Flavors of Queue (wait) Time

- Wait for “server”: initiator / CICS AOR / IMS MPR
- CPU delay (wait for logical CPU)
- I/O delay (iosq, pend, disconnect)
- Capping delay (LPAR capped vs actual delay)
- Resource Group maximum enforced
- Wait for LPAR (logical CPU) to be dispatched
  - PR/SM weight
  - Demand from other LPARs
  - CPC/CEC capacity
Initiator Queue

• SMF: R723CQDT
• TOTAL queue time (divide for average per job)

• Just start more inits?
• Not necessarily a good idea

• “Tuning to reduce the number of simultaneously active address spaces to the proper number needed to support a workload can reduce RNI and improve performance”
Automated Initiators: Less is More

- Concurrency based on performance and utilization
CPU Delay

- Wait for logical CPU
- SMF: R723CCDE
- Work is ready to run but is delayed access to CPU
- Related to Service Class / goal / importance
  - Dispatching priority

- There is almost always some CPU delay
  - Tolerance is subjective
  - Are goals/SLA's being met?

- Priorities are relative – overloading leads to thrashing
- Consider discretionary for MTTW
Utilization vs CPU delays

At 100% busy, throughput degrades significantly

50% delay may be acceptable
I/O Delay

• IOSQ:
  – HyperPAV

• Pend:
  – CMR = overloaded controller
  – DB = volume contention (reserve?)
  – Any remaining = likely channels

• Disconnect
  – Random read misses
  – Synchronous remote copy
Revisit: sample Job B

- Disconnect time of 0:40:31 = 40% of total 1:37:44
- 40:31 (2431 seconds) divided by 9239646 I/O’s…
- = .263 ms average disconnect time

- Likely not unreasonable for random reads (consider SSD)
- Could also be replicated writes
- Become familiar with your typical application response times
Capping Delay

• Possible when caps present

• SMF70NSW
  – WLM caps the logical CPUs
  – Delays LPAR dispatch

• SMF70NCA
  – Work is actually delayed for CPU due to capping

• Consider TM automation 😊
Capping vs Delay

MSU Demand vs R4HA

LPAR is capped (SMF70NSW)

Work is delayed SMF70NCA
Capping can impact all Workloads

Batch is not the only workload suffering. Even the most critical workloads are unable to meet their goal.

Machine CPU reaches 100%

Capping begins
Resource Group (max)

• Also a form of capping (same WLM algorithms)

• Pro: Useful to control “problem” applications
• Con: Static. Not flexible

• R723CCCA
  – Resource Group maximum enforced
  – Will override Service Class goals
LPAR Dispatch Delay

- Ratio of logical processor busy to physical processor busy
- Not always as obvious but very common!
- Term “Short CPs” introduced by Kathy Walsh (IBM WSC)
  - Share, Aug. 2004
  - https://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1077
- MXG: PLCPRDYQ
- Improved with Hiperdispatch and IRD
• More initiators and/or higher dispatching priority will not resolve this problem
What does this look like in the real world?

Let’s take a trip to the deli 😊
How many in the store at one time?

INITIATORS
Who’s next in line?

Dispatching Priority
(Service Class)
How long til I can give my order?

Logical Processor (CP) Busy
How long til I’m done!

Physical Processor (CP) Busy
Forcing one step only stresses the next.
Balance
About MVS Solutions

• MVS Solutions Inc.
  – Installed in over 200 datacenters worldwide
  – IBM Partner in Development

• ThruPut Manager
  – Automated Workload Balancing
  – Automated Batch Prioritization
  – Automated Capacity Management

Contact me: jbaker@mvssol.com

systemz@cmg.org

Join our Blog at www.thruputmanager.com
42nd International Performance & Capacity Conference

SAVE the DATE
November 7-10, 2016
Hyatt Regency | La Jolla, CA

www.cmg.org