Road Rage against the machine: Understanding – and dealing with – z14 traffic patterns

John Baker, Sr. Performance Consultant
• Understanding and dealing with z14 traffic patterns
  • Interpreting delays and utilizations

• PR/SM, processor weights and Hiperdispatch
  • Overview, implications, and measurement
  • How to set your weights

• Common sources of delays in Z Systems
We hate to wait
What does ‘busy’ mean?

• Wikipedia: “In equipment and tool rental companies, utilization is the primary method by which asset performance is measured and business success determined”

• Dictionary.com: “To put to use; turn to profitable account”
CPU utilization ("busy")

- Techopedia: CPU Utilization refers to “a computer’s usage of processing resources, or the amount of work handled by a CPU”

- Consider:
  - Average across time
  - Logical vs physical
  - Average across multiple components
Average across time

- John-ipedia: CPU Utilization is a measure of usage over time

80% busy simply means 100% busy for 80% of the time
Logical vs Physical
Average across components

**CEC 1**

- One CPU dispatched for 900 seconds, in a 900 second RMF interval, is 100% busy
- A second CPU dispatched for 450 seconds, in the same time interval, is 50% busy

**CEC 2**

- Both CPU’s are dispatched for 675 seconds each, in the same 900 second RMF interval.
- Each CPU is 75% busy.

- The average CPU utilization of both CEC’s is 75%
  - Will performance be equal?
z14: How we got here
PR/SM, processor weights and Hiperdispatch
Distributing resources on Z hardware: PR/SM

- Weight is guaranteed minimum
  - May be exceeded if other LPARs do not use their full share
- Logical CPs (LCP) limit share

<table>
<thead>
<tr>
<th>LPAR 1</th>
<th>LPAR 2</th>
<th>LPAR 3</th>
<th>LPAR 4</th>
<th>LPAR 5</th>
<th>LPAR 6</th>
<th>LPAR 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEIGHT=350</td>
<td>WEIGHT=100</td>
<td>WEIGHT=100</td>
<td>WEIGHT=100</td>
<td>WEIGHT=200</td>
<td>WEIGHT=100</td>
<td>WEIGHT=50</td>
</tr>
<tr>
<td>LCP = 7</td>
<td>LCP = 3</td>
<td>LCP = 3</td>
<td>LCP = 3</td>
<td>LCP = 4</td>
<td>LCP = 3</td>
<td>LCP = 2</td>
</tr>
</tbody>
</table>

350/1000 = 35% Share
100/1000 = 10% Share
100/1000 = 10% Share
100/1000 = 10% Share
200/1000 = 20% Share
100/1000 = 10% Share
50/1000 = 5% Share

* * * * * * * * * * * * * * *
15 physical CPs

LPAR Share of CPU Pool

- LPAR1
- LPAR2
- LPAR3
- LPAR4
- LPAR5
- LPAR6
- LPAR7
Horizontal vs Vertical alignment

- A method of “aligning” physical CPUs with LPARs
  - Goal is to take advantage of newer hardware design
  - Default as of z/OS 1.13

<table>
<thead>
<tr>
<th>HIPERDISPATCH = NO</th>
<th>HIPERDISPATCH = YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Horizontal” alignment</td>
<td>“Vertical” alignment</td>
</tr>
<tr>
<td>% share distributed across all available physical CPUs</td>
<td>% share distributed by physical CP</td>
</tr>
<tr>
<td>All CPs distributed among all LPARs</td>
<td>LCPs become “Vertical High, Vertical Medium, Vertical Low”</td>
</tr>
<tr>
<td>Utilization of all CP’s tends to be even</td>
<td>Unused (Vertical Low) CPs are “parked”</td>
</tr>
</tbody>
</table>
Why is Hiperdispatch a good idea?

- Dispatching on the same, physical cores (rather than any available core) dramatically improves your probability of cache hits, thus improving application performance and overall system throughput.
Weight distribution with Hiperdispatch

- **LPAR 1 example:**
  - 35% share = .35 * 15 CP = **5.25 CPs** (also defined 7 LCPs)
  - 1.0 = VH, .5-.99 = VM, remainder LCPs = VL *
  - **First, every LPAR receives one VM = ‘.5 CPs’**
    - 5.25 - .5 = 4.75 remaining
    - 4.75 – 4 (four ‘whole’ CPs = VH) = .75 remaining
    - .75 = one more VM
    - One VL to complete 7 LCPs as defined

- = 4 * VH, 2 * VM, 1 * VL
Logical/Physical Association

- LPARs and logical processors associated with drawers, nodes and physical processors
Measuring Hiperdispatch
RNI: Polarity matters
Common Sources of Delays
Common sources of road rage (wait time)

- Wait for “server”: initiator / CICS AOR / IMS MPR
- CPU delay (wait for logical or physical CPU)
  - Also PR/SM dispatch delay from weight enforcement or capping
- I/O delay (iosq, pend, disconnect)
- Capping delay (LPAR capped vs actual delay)
- Resource Group maximum enforced
Input queue

- Wait for “server”: INITIATOR / CICS AOR / IMS MPR

- Start more servers?
  - More logical processors...
    - More physical processors...

- Consider where you want your largest queues to reside
  - Sometimes, less is more

- “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”
CPU Delay

- Wait for logical CPU
- 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
WLM Samples: Using vs Delay
I/O Delay components

• **IOSQ:**
  • HyperPAV / SuperPAV

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

• **Disconnect**
  • Random read misses
  • Synchronous remote copy
I/O Response Distribution
Capping Delay

• Possible when caps present

• SMF70NSW
  • WLM caps the logical CPUs
  • Delays LPAR dispatch (‘sleep vs awake cycle’)

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

• Very useful measurements, particularly when using ISV capping tools

• At the Workload/Service Class levels (72) this will show up as CPU delay
LPAR Capped vs Work actually delayed

- LPAR is capped
- Work is delayed
WLM Resource Group (Max)

- Also a form of capping (same WLM ‘sleep/awake) patterns)

- Pro: Useful to control “problem” applications

- Con: Static. Not flexible
  - Will override Service Class goals

- Note: this ‘capping delay’ is unrelated to LPAR caps (DC/GC)
Resource Group Max Enforced
Questions?
Thank you for listening
John.Baker@intellimagic.com

IntelliMagic Vision for z/OS

End-to-End z/OS Infrastructure
AI-driven Analytics for
Performance and Capacity Management

IntelliMagic.com