Session P20

Will HyperPAV be the End of IOSQ Time?

Andries de Jong / Dr. Gilbert Houtekamer - IntelliMagic

IBM System Storage and Storage Networking SYMPOSIUM
20-24 Aug 2007
Las Vegas, NV

About IntelliMagic

IntelliMagic Product Portfolio:

- **RMF Magic:**
  Enabling z/OS Storage Performance Best Practices
- **Disk Magic:**
  Disk Subsystem Performance Modeling Tool for zSeries, iSeries, Open
- **Batch Magic:**
  z/OS Tape/Virtual Tape Analysis and sizing
- **Capacity Magic:**
  Disk configuration planning tool for IBM Disk Subsystems

IntelliMagic Services:
Consulting and educational services such as the 4 day class: “z/OS I/O Architecture & Performance Analysis”.
Presentation Overview

- Multiple Allegiance
- Why must I/O be kept waiting?
- PAV operation
  - Static PAV
  - Dynamic PAV
  - HyperPAV
- Differences between Dynamic and HyperPAV
- Sizing alias requirements
- Implementation recommendations
  - How many aliases do YOU need for YOUR workload
- Tracking PAV usage in production
  - RMF measurements about (Hyper)PAV

Multiple Allegiance
zSeries I/O Queuing

How it works:
- Historically, MVS issues one I/O at a time to a logical device.
- Logical device is busy while I/O is being executed.
- Other I/Os will be put in I/O Supervisor Queue (IOSQ).

Device busy can be computed as:
\[
\text{busy} = \text{I/O rate} \times \text{service time}
\]
And IOSQ wait time is approx.:
\[
\text{wait} = \frac{\text{service time}}{1 - \text{busy}}
\]

Multiple I/Os to One z/OS Device

- A z/OS device (volser) is a logical, not a physical disk.
  - Amount of space in a RAID array.
- I/O to one z/OS device number is handled from cache or is serviced by (one of) multiple disks.
- ‘Device busy’ does not have to occur when multiple:
  - Read I/Os can be processed from cache.
  - Read misses can be served from free disk.
  - Write operations do not have extent conflict.
Shared Disks and Multiple Allegiance

- Each z/OS system maintains its own I/O Queue for each device
- More than one system can issue I/O to the same logical device at the same time
- Without Multiple Allegiance you will get
  - Pending delay: device busy
- With Multiple Allegiance the disk subsystem will accept one I/O per z/OS host
  - Multiple Allegiance allows for concurrent I/O execution, but
  - Conflicts may occur; on write extents or with read miss

How Multiple Allegiance Works

Multiple Allegiance:
- Separate queue in each z/OS system
- Disk Subsystem (DSS) accepts one I/O from each system
  - This implies separate ‘bookkeeping’ for each attached z/OS system (pathgroup)
  - Concurrent I/O without the need for an alias
But You Still May Have to Wait

Conversely, the I/O from multiple z/OS devices could be mapped to one physical disk.
Actual mapping depends on physical and logical volume size, and on RAID scheme.

Many z/OS 3390 images (e.g. 9 GByte each)
Devices in RAID rank (e.g. 73 GByte each)
Wait for Physical Disk

- Wait for physical disk access as part of:
  - Read miss
  - Destaging (random, sequential)
  - Read-ahead (sequential)
- Only read-miss has direct response time impact
- Likelihood of delay depends on I/O density to physical disk; not on logical volume size!
- Physical disk busy does not result in RMF ‘device busy’, but in disconnect delay time

Wait Because of Extent Conflict

- Write operations need to maintain consistency
- Only one update is allowed to each ‘entity’ on disk, e.g. a data set
- DFSMS defines size of entity as ‘extent’ in channel program
- ‘Extent’ is a cylinder/head range on disk: Devno – CC,HH₁ : CC,HH₂
- Disk Subsystem serializes writes to same extent
- Extent conflicts are independent of 3390 volume size (Mod 3, 9, 27, or 54)!
“Define Extent” CCW Contains Extent Definition

Formatted GTF Trace

Usually the extent is a (portion of) a data set

Wait for Logical Device

- Must wait when device is ‘reserved’ by another system
  - Reported by RMF as ‘reserve delay’
  - Very few applications still require a ‘reserve’
Parallel Access Volumes

- Parallel Access Volumes allow z/OS to start more than 1 I/O to a logical volume
- ‘Alias’ addresses are used to start the additional I/Os
- Disk Subsystem may need to queue request internally
  - Extent conflicts
  - Physical device busy

With sufficient alias addresses, any 3390 volume size can be used!
Static PAV

- One or more aliases for each base address
- Introduced with OS/390 v1.3
- Guarantees availability of one or more additional addresses for busy devices
  - Delay probability much lower, roughly \((1 - \text{device busy})^2\)
- Issues
  - Occasionally more aliases may be required for a busy device, i.e. IOSQ cannot be completely eliminated
  - Cannot automatically follow workload changes, e.g. online vs batch volumes
  - May run into system addressing limits

Dynamic PAV

- Introduced with OS/390 v2.7
- Pool of alias addresses per SSID (LCU) that can be assigned dynamically to base addresses in SSID (LCU) in need of more parallelism
- Reassignment requested and coordinated by WLM (workload manager) based on
  - Workload performance needs
  - Overall I/O efficiency
- Disk Subsystem owns alias – base mapping and is point of control
- Initial state in IOCP
WLM Assigns Aliases

- WLM manages pool of alias addresses, and re-assigns if:
  - A workload is not meeting its goals
  - Overall efficiency can be improved
- WLM detects that service class is missing goals because its I/Os are suffering IOSQ delay:
  - Each z/OS in the sysplex must have same assignments!
  - One WLM in sysplex will handle alias changes, and communicate those to all systems.

WLM Dynamic Alias Management

(A) Goal algorithm assigns aliases to base device that (1) is experiencing IOSQ delay and (2) is serving a workload that is missing its customer-specified WLM goal. Alias is taken from donor device that is serving less important work, or that has no need for it.

- Example shows that a device not meeting its WLM goals receives additional aliases.
- WLM continues to add aliases when the service requirements are not met (also when it does not help; that is why you may see very many).
WLM Dynamic Alias Management

(B) Efficiency algorithm assigns aliases to base devices that are experiencing IOSQ delay to improve overall system activity.

- Alias is taken from donor device that has no queuing, and that is not serving important work.
  - Alias are assigned to devices with the most queuing.
    - Could be devices with very little queuing!
  - Efficiency algorithm is executed once every minute, goal algorithm every 30 seconds.

Dynamic PAV Alias

Same relationship between alias and base address for all attached servers (pathgroups)

Example shows:
- SYSA with 2 I/Os to 1012 and 2 I/Os to 1013
- SYSB with 2 to 1013

Each DSS base and each DSS alias address provides I/O execution thread per pathgroup.
Monitoring Alias Re-assignment

Chart shows the number of devices that had aliases added or removed (hence minimum value of 2) during each RMF Interval over a 1 day period.

Note that 1 RMF Interval has about 30 executions of the goal algorithm!
So changes are rare for this disk subsystem.

HyperPAV

- Alias is assigned for duration of I/O request
  - Any alias in the pool will do
  - Does not require coordination with other systems
- Assigned based on WLM I/O priority
  - But not to reserved volumes
- Requirements:
  - FICON,
  - z/OS 1.6 or later
  - DS8000 (2.4 or later)

See also: “Understanding the Performance Implications of HyperPAV” by Dr. Pat Artis
http://www.perfassoc.com/jsc/pdf/papers/HyperPAV_slides_06.pdf
HyperPAV Alias

Relationship between alias and base address only for one system (pathgroup)

Example shows
- SYSA with 2 I/Os to 1012
- SYSB with 2 I/Os to 1013

Base addresses 1000-107F
Alias addresses 1080-10FF

HyperPAV Does Not Use Fixed Alias-Base Mapping

- No need for coordination between z/OS systems!
# Comparing Dynamic and HyperPAV

<table>
<thead>
<tr>
<th></th>
<th>Dynamic PAV</th>
<th>HyperPAV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managed at</td>
<td>LSS</td>
<td>LSS</td>
</tr>
<tr>
<td>How dynamic</td>
<td>Reassignment takes minutes</td>
<td>Per I/O request</td>
</tr>
<tr>
<td>Alias assignment</td>
<td>Fixed alias to base assignment, until changed by WLM</td>
<td>Assigned for duration of I/O request</td>
</tr>
<tr>
<td>Alias coordination</td>
<td>Alias assignment must be the same on all servers; change requires execution of protocols between all servers and DSS</td>
<td>Alias assignment has single system scope</td>
</tr>
<tr>
<td>Alias sizing</td>
<td>Based on maximum number of active devices</td>
<td>Based on maximum number of concurrent I/O operations /LPAR</td>
</tr>
<tr>
<td>Alias status</td>
<td>Always assigned to a base</td>
<td>Assigned only while I/O in progress</td>
</tr>
<tr>
<td>Multiple Sysplexes</td>
<td>Cannot coordinate across Sysplexes</td>
<td>Number of Sysplexes does not matter</td>
</tr>
</tbody>
</table>

## How Many Alias Addresses?
How Many Aliases?

In most cases 1 alias will be enough to eliminate queuing
Compare single-server queue with dual-server queue for same work:

Three 3390-9’s vs One 3390-27 with 2 aliases

Even at 80% utilization, queue time is very limited: PAVs can drive utilization higher!
PAV and Large Volumes (3390-27/54)

3390-9 and 3390-27 without PAV:
- Large volume makes contention more likely
- Only one address, more load

3390-27 with PAV:
- More alias addresses can be assigned as needed
- Performance will be better than 3 * 3390-9

• High priority I/Os can overtake low priority I/Os!
• Read hits can overtake read-misses

So, How Many Do We Need?

• Little’s Law:
  - Number in system = arrival rate * service time

So if the I/O rate is 10,000 and the service time is 5 msec, then the number of active devices =
  - 10,000 * 0.005 (5 ms) = 50

So on average 50 devices are busy at any one time, while the subsystem may have 4000 device addresses defined!

Real workload example with 5 large subsystems (peak 30,000 I/Os per second)
Size Aliases on # Concurrent I/Os

Dynamic PAV
- PAV algorithm can assign only 1 (efficiency) or 2 (goal) aliases per minute,
- Workload will be more dynamic than that, so we need to over-assign PAV addresses
- Utilization target for alias should not be 100% but maybe 25%, so need 4 times as many

HyperPAV
- Alias assignment at I/O level, so “only” need to size for peak of load from any one system
- Peak value concurrent I/Os will be higher than average, allow for at least factor 2
- Workload bursts can even require more aliases

Monitoring Concurrent I/Os

IORate*Service (Concurrent Active I/Os)
Limits to Concurrency

- Many aliases can be assigned, but...
  - They share a pool of alias addresses in the LCU
  - They share a set of channels with all other I/O activity to the subsystem
  - Each logical device maps to a limited number of back-end HDDs, even with RAID-5
    - ... and in many cases the device gets I/O from more than 1 host
- It is therefore not clear that there is much value in more than 4-8 aliases to any device

Possible Rules to Follow

- Use RMF Magic
  - Example on the left
  - Conservative sizing based on device capacity AND activity
- Assign aliases based on concurrent I/Os
  - Dynamic PAV: 4-8 aliases per concurrent I/O (for all systems)
  - HyperPAV: 2-3 aliases (for busiest system)
- Assign aliases based on use of large volumes (‘fear factor’)
  - 1 alias for 3390-9
  - 2 aliases for 3390-27
  - 3 aliases for 3390-54
Other Observations

- PAV and HyperPAV are managed at LCU level
  - LCU balance is important for best performance
  - Larger LCUs (more device addresses) simplify balancing

- PAV and HyperPAV do not solve extent conflicts
  - Concurrent updates do require serialization

- z9 109 provides second set of subchannel addresses
  - Can be alternative if disk subsystem does not support HyperPAV
  - Or if the Workload does not need HyperPAV responsiveness

- Be sure to plan for growth
  - Leave addresses to add capacity and/or aliases during the lease term

(Hyper)PAV and Logical Volume Size Recommendations

- Use PAV to eliminate IOSQ delay
  - Dynamic PAV is more effective than static PAV
  - HyperPAV is more dynamic than Dynamic PAV

- Use all 256 addresses in LCU (rather than many ‘small’ LCUs)
  - Allows for better balancing and more dynamic PAV assignment

- Migrate aggressively to large logical volumes
  - Remaining contention will be at extent level, which is largely independent of volume size

- Risks
  - If secondary device throughput is limited, PAV may increase exposure
  - Can be mitigated with striping, especially for sequential I/O
RMF HyperPAV Information

- New HyperPAV reporting fields are (74.1 and 78.3)
  - Number of aliases that are in use
  - Maximum queue length for HyperPAV (should be small)
  - HyperPAV wait rate, i.e. how often out of free aliases
  - HyperPAV request rate
HyperPAV - Summary

- Allows data centers to use very large volumes without performance concerns

- Improves PAV responsiveness
  - Unlike Dynamic PAV, HyperPAV can respond instantaneously to workload changes

- Minimal changes to z/Architecture and z/OS
  - Easy to implement, low risk

References

- IBM Publication SG24-6990-02
  ABCs of z/OS System Programming Volume 10
  (http://www.redbooks.ibm.com/redpieces/pdfs/sg246990.pdf)

- Dr. Pat Artis
  Understanding the Performance Implications of HyperPAV
  (http://www.perfassoc.com/jsc/pdf/papers/HyperPAV_slides_06.pdf)

- IBM System Journal, 2003, Volume 42, Number 2
  A. S. Meritt, J. A. Staubi, K. M. Trowell, G. Whistance, and H. M. Yudenfriend
  z/OS support for the IBM TotalStorage Enterprise Storage Server
Questions?