POWER7 Affinity and Performance Part 2

- Slides we did not cover last week in Part 1
- DeveloperWorks Guru of the Month & YouTube
  - Ten Top Techie Treats – information sources
- My Redbook Library
- Getting help from guru level tools
  - The Optimisers
  - The Advisors
- More Advanced Level and New stuff
  - Getting POWER7 to look more like POWER5/6
  - Working out “space capacity”
  - Physical VM placement
  - VM “defrag”
  - Scaled Throughput

Nigel Griffiths
IBM Power Systems
Advanced Technology Support, Europe

BLOG 6

Too High a Virtual Processor → Bad Side Effect

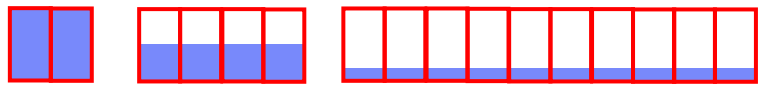
Many find the term: Virtual Processor (VP) confusing!

Doesn't "virtual" mean pretend or shared and are they free?

"Spreading Factor" is clearer

Too High a Virtual Processor → Bad Side Effect

- Entitled - guaranteed CPU cycles
- VP - number of CPUs the LPAR "sees"



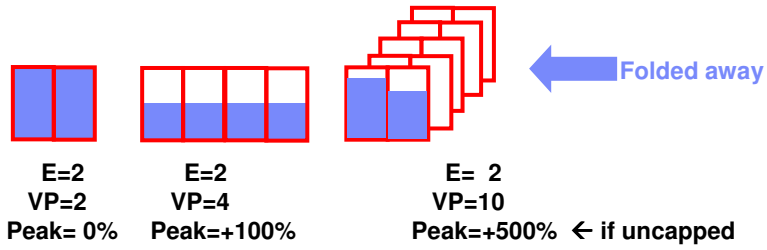
E=2
VP=2
Peak= 0%

E=2
VP=4
Peak=+100%

E= 2
VP=10
Peak=+500% ← **all uncapped**

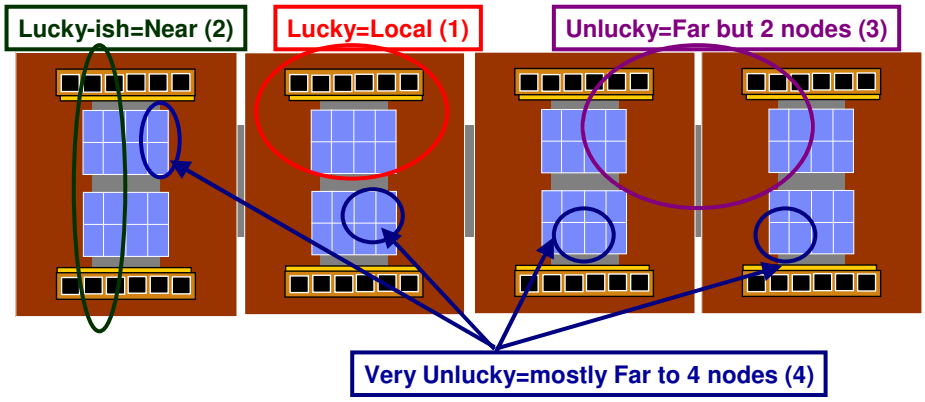
Too High a Virtual Processor → Bad Side Effect

- Entitled - guaranteed CPU cycles
- VP - number of CPUs the LPAR "sees"



Even small VP counts can be unlucky
Example 4 CEC Power 770/780

Virtual Processor=8 possible CPU-core allocation



Don't want to force the Hypervisor to use Very Unlucky

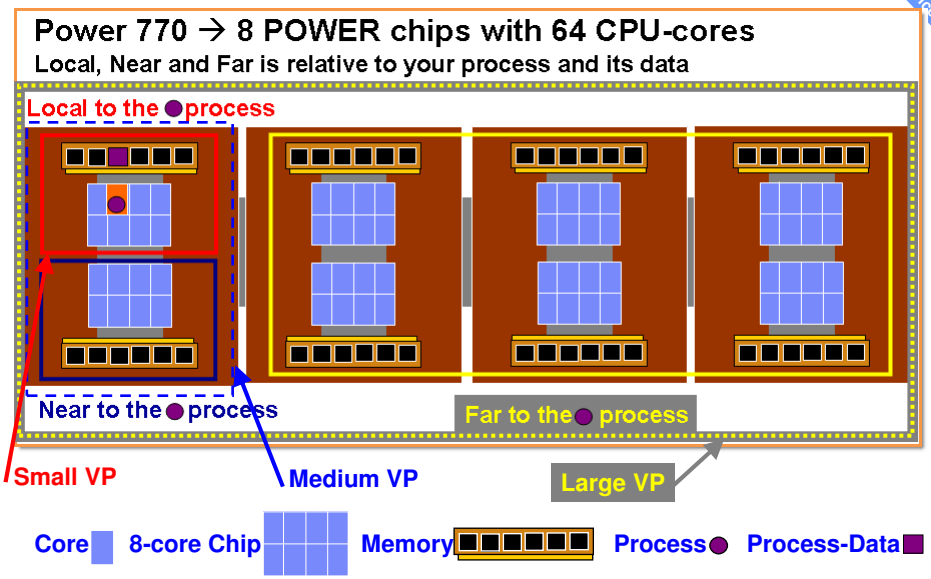
If Entitlement=16 and unused VP's are folded …

Is there an difference between VP=20, 32, 48, ... ?

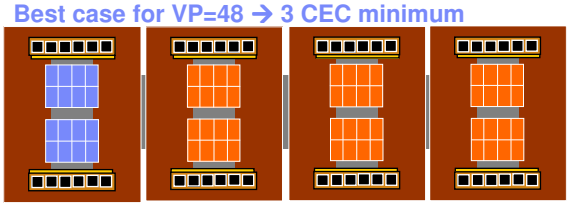
Four Strategies for E and VP

1. Maximum flexibility - "Virtual Processors are free, right!"
   - E as low as possible & VP as high as possible
   - Example: VP = 48 and E=4.8 (48/10)
2. It is production, so it must get what it needs
   - E ~ monitored average CPU & VP=E * 3
   - Example: E=16 and VP=48
3. It is vital production so it needs to peak & perform
   - E ~ regular peaks (as monitored) & VP a good handful of extra CPUs
   - Example: E=18 and VP=36
4. It is vital but limit the spread across the machine
   - E to cover the peaks and the minimum extra VP (+1 or +2)
   - Example: E=18 and VP=20

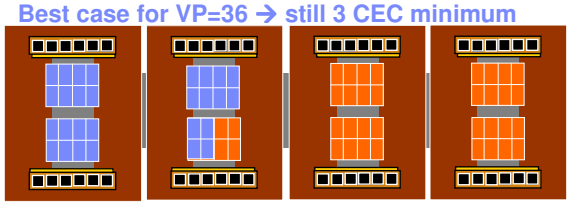
High VP
= more the VM is spread for no added value



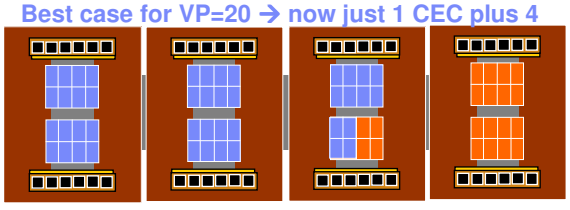
Best case for VP=48 → 3 CEC minimum



Best case for VP=36 → still 3 CEC minimum



Best case for VP=20 → now just 1 CEC plus 4



Back to the example virtual machine of Entitlement = 4.8 - 18

Allocated CPU-core

Unallocated CPU-core

This asks the question: Can we squeeze down to E=16 and VP=16 ?

Virtual Processor Conclusions

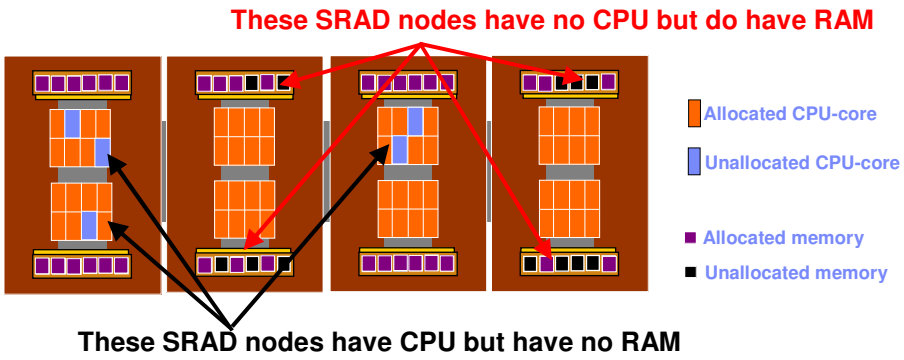
1. Smaller VP reduced VM spread across the box
2. Don't use the Maximum possible, E x 3, nor E +50%
3. Recommended:
   - Sensible Entitlement with VP = E + 1 or E + 2
   - But monitor your physical CPU use
4. Avoid setting just beyond a “natural” size
   - To avoid orphan resources

BLOG 7

VM placement also needs RAM

VM placement also needs RAM

- Easy to forget in VM placement needs CPU and RAM
- A worst case is …CPU & RAM islands



IBM POWER7 Performance © 2013 IBM Part 2

VM placement also needs RAM

The hypervisor has a set of rules for placement

- Memory minimum size is the LMB
  - Set on the HMC → ASMI → Performance Menu
  - Note: LPM needs source & target the same LMB
- Ideally, assigned CPUs have matching RAM
  - If you have fixed CPU:RAM ratio it is fairly likely
  - Also a VM CPUs are a bit sticky!
  - This means consistent over VM reboots

VM placement also needs RAM

- 20 VP + 64GB RAM → Minimum 3 SRADs

```
purple1:/# lssrad -av
REF1   SRAD        MEM       CPU
0
          0   21195.25      0-3 8-11 16-19 28-31 44-47 60-63 76-79
          3   10582.50      32-35 48-51 64-67
1
          1   20027.00      4-7 12-15 20-23 36-39 52-55 68-71
          2   11827.50      24-27 40-43 56-59 72-75
purple1:/#
```

Memory Balance looks good

- SRAD 0 - 7 VP with 21195.25 MB RAM = 3028 MB per CPU-core
- SRAD 1 - 6 VP with 20027.0 MB RAM = 3338 MB per CPU-core
- SRAD 2 - 4 VP with 11827.5 MB RAM = 2957 MB per CPU-core
- SRAD 3 - 3 VP with 10582.5 MB RAM = 3528 MB per CPU-core

- But it would be nice to use less SRADs …

IBM POWER7 Performance Part 2 © 2013 IBM

VM placement also needs RAM

- 16 VP + 64GB RAM → still 4 SRADs

```
purple1:/# lssrad -av
REF1   SRAD        MEM       CPU
0
          0   21195.25      0-3 8-11 16-19 32-35 48-51
          3   10582.50      28-31 44-47 60-63
1
          1   20027.06      4-7 12-15 20-23 36-39 52-55
          2   11827.50      24-27 40-43 56-59
purple1:/#
```

But then remembered 64 GB per Power 770 CEC = every byte that could force high SRADs.

```
purple1:/# lssrad -av
REF1   SRAD        MEM       CPU
0
          0   15871.44      0-15 20-23 32-35 44-47 56-59
1
          1    9835.50      16-19 24-27 36-39 48-51 60-63
          2    6100.50      28-31 40-43 52-55
purple1:/#
```

That dropped one SRAD. Perhaps no other empty POWER7 chip!

VM placement also needs RAM

What have we learnt?

1. You can't decide where the virtual machine goes in absolute physical terms
2. Existing virtual machines may not allow a perfectly layout balance
3. Given free resources the Hypervisor does something sensible

BAD VM placement

Bad example = hypervisor “no room to manoeuvre”

- VP=8 + RAM=32GB
  - SRAD 0 has 4.3 GB/CPU ☺
  - SRAD 1 has 1.5 GB/CPU ☹
  - It is small = limited damage
  - Don’t assume DPAR 1 CPU would remove it

```
# lssrad -av
REF1   SRAD        MEM      CPU
0
          0   30224.00      0-27
          1    1490.00     28-31
#
```

- Bigger example
  - 5 CPUs have no local RAM
  - Not a disaster


```
#lssrad -av
REF1   SRAD        MEM      CPU
0
          0   13677.50      0-11
          1   35808.00     12-43
          2    4731.00     44-47
          3       0.00     48-51
1
          4       0.00     52-59
          5       0.00     60-63
          6       0.00     64-67
          7       0.00     68-71
#
```

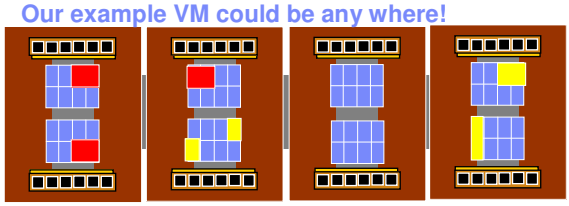
BLOG 8

Dynamic-LPAR changes - mess-up VM placement

Dynamic-LPAR changes - mess-up VM placement

- The hypervisor does the placement
- lssrad only gives **logical** view of 1 LPAR

Our example VM could be any where!



- We can't "see" the in-use or free machine physical resources
- Leads to unpredictable DLPAR changes

DLPAR shrink + explode + return

VP=16 RAM=32 GB

```
# lssrad -av
REF1  SRAD      MEM     CPU
0
      0  15824.31     0-7 12-15 20-23 28-31 36-39 44-47 8
                          56-59
1
      1  12325.50     8-11 16-19 24-27 32-35 40-43 52-55 6
      2   3610.50     48-51 60-63                        2
#
```

VP=16 RAM=32 GB

```
# lssrad -av
REF1  SRAD      MEM     CPU
0
      0  14828.31     28-31 36-39 44-47 56-59 64-67 8
                          72-75 80-83 88-91
      3      0.00
1
      1  14317.50     24-27 32-35 40-43 52-55 68-71 7
                          76-79 84-87
      2   2614.50     48-51                        1
#
```

VP=2 RAM=8 GB

```
# lssrad -av
REF1  SRAD      MEM     CPU
0
      0   4121.31     56-59
1
      1   4108.50     52-55
      2    124.50
#
```

VP=24 RAM=48 GB

```
# lssrad -av
REF1  SRAD      MEM     CPU
0
      0  19932.81     0-7 12-15 20-23 28-31 36-39 44-47
                          56-59 64-67 72-75 80-83 88-91
      3   3859.50
1
      1  21289.50     8-11 16-19 24-27 32-35 40-43
                          52-55 68-71 76-79 84-87
      2   2614.50     48-51 60-63 92-95
```

Lesson: might not get back to your starting position

DLPAR shrink again then slow grow

VP=2 RAM=4 GB

```
# lssrad -av
REF1  SRAD      MEM     CPU
0
      0   1382.31     56-59
      3      0.00
1
      1   1245.00     68-71
      2   1245.00
```

VP=2 RAM=8 GB

```
# lssrad -av
REF1  SRAD      MEM     CPU
0
      0   5366.31     56-59
      3      0.00
1
      1   1245.00     68-71
      2   1245.00
#
```

VP=2 RAM=5 GB

ADD 1 GB

```
# lssrad -av
REF1  SRAD      MEM     CPU
0
      0   2378.31     56-59
      3      0.00
1
      1   1245.00     68-71
      2   1245.00
```

VP=2 RAM=6 GB

```
# lssrad –av
REF1  SRAD      MEM     CPU
0
      0   3374.31     56-59
      3      0.00
1
      1   1245.00     68-71
      2   1245.00
```

Lesson: might get more unbalanced

DLPAR then new VM + grow

VP=2 RAM=8 GB

```
# lssrad -av
REF1  SRAD      MEM     CPU
0
      0   5366.31     56-59
      3     0.00
1
      1   1245.00     68-71
      2   1245.00
#
```

VP=16 RAM=32 GB

```
# lssrad -av
REF1  SRAD      MEM     CPU
0
      0  12836.31     0-7 12-15 20-23 28-31 36-39 44-47 56-59   8
      3      0.00
1
      1   1618.50     8-11 16-19 24-27 32-35 40-43 52-55 68-71  7
      2  17305.50     48-51                                     1
#
```

Start other virtual machine with most resources except what this one needs to grow →

VP=2 RAM=32 GB

```
# lssrad -av
REF1  SRAD      MEM     CPU
0
      0  12836.31     56-59
      3      0.00
1
      1   1618.50     68-71
      2  17305.50
#
```

Lesson: fully machine then DLPAR can result in ugly placement (no other option)

Dynamic-LPAR changes - mess-up VM placement

- This was a deliberately extreme test case
- It could have been far worse!
  - Sticky CPU-cores could be unstuck with more DLPAR
- In production placement may remain a mystery !!
- Observation: Having the luxury “played/researched!”
  - Seen odd decisions at 1st but later I could explain them
  - 16 CPU + 64GB → in 770 obviously two POWER7 CEC + its 64GB
  - But already had 4 VM running
  - Only one POWER7 unused so got 8
  - Then had to split 2nd 8 CPU across SRADs (6+2 or 7+1)
  - Later tried 8 CPU + 64GB – but probably at least 1 LMB is in use so to split memory into 2 SRADs

Dynamic-LPAR changes - mess-up VM placement

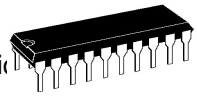
- Smaller changes or temporary boost are OK
  - Return to same placement (mostly)
- The warning is clear:
  Drastic DLPAR changes &
  starting new VM's which fill the box
  can lead to
  "sub-optimal" VM placement

BLOG 9

Firmware = Hypervisor improvements

Firmware – Experience based Maturity

- It's been 2 year since POWER7 rolled-out
- We have no firmware marketing budget
  - The Readme history has "one liners" only
  - If you know you can see the important ones but its well hi
  - Mostly vague statements
- Much learnt in
  - Real life applications scheduling patterns
  - Large VMs process and memory management
  - Performance with 10,000's of thread
  - Algorithms improved, tuning options added, new thresholds … **bugs fixed**



Look for "associate/associated" "performance"

Download

- **Low-End - ftp://ftp.boulder.ibm.com/software/server/firmware/AL-Firmware-Hist.html**
- **Mid-range - ftp://ftp.boulder.ibm.com/software/server/firmware/AM-Firmware-Hist.html**
- **High-End - ftp://ftp.boulder.ibm.com/software/server/firmware/AH-Firmware-Hist.html**

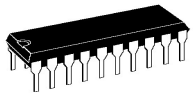
IBM
© 2013 IBM
POWER7 Performance
Part 2

Firmware – Now is the Time to upgrade

OLD STATEMENT from early 2012

1st years experience went into the latest 730 Firmware

- Minimum 720-101



Extra bonus for Capacity Upgrade on Demand users

- VM placed before ensuring spare capacity unused

I strongly encourage all **POWER7** users
to upgrade to firmware level 730
especially the larger machines

In 2013
Newer machines have 740, 750 & soon 760 Firmware
Please plan for upgrade 6 months & 1 year
after initial install & then yearly
- GA Firmware does not support CHARM
- Benefit from early RAS & performance fixes

Lessons

1. Placement: find out the layout of your box & VMs
2. SMT: Expect POWER7 SMT4 CPU to “look” different
3. Entitlement: set based on monitoring
4. Virtual Processor: set just a little larger
5. On Reboot: start the larger LPARs first
6. To unstick a LPAR: start it and stop at SMS with VP=1 & RAM=1GB, no slots, then stop it & start with regular profile
7. Drastic DLPAR can effect placement
8. Firmware Update to 730 (or 720_101 minimum)

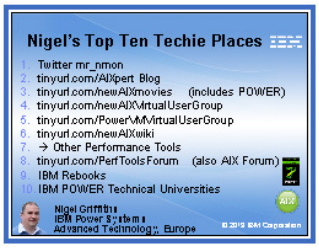
9. Not covered but for unfolding fixes upgrade to
   - AIX 6 TL06 + latest services pack
   - AIX 6 TL05 + latest services pack + fixes
     - APAR IZ94768, APAR IV01111, APAR IV06194
   - AIX 7 TL1 or AIX 6 TL7 are ok

Since I wrote this slide AIX 6 TL 7 & 8 and AIX 7 TL2 available

Now for something Completely Different

Now for something Completely Different

- A bit of fun for DeveloperWorks “Guru of the month”
  - nmon for AIX and Linux story
  - What I do now
  - 2nd half = Ten Top Techie Treats



- Only 5 minutes long
  - http://www.youtube.com/watch?v=w58MMb-XbwQ

Nigel’s Top Ten Techie Places

1. Twitter mr_nmon
2. tinyurl.com/AIXpert Blog
3. tinyurl.com/newAIXmovies (includes POWER)
4. tinyurl.com/newAIXVirtualUserGroup
5. tinyurl.com/PowerVMVirtualUserGroup
6. tinyurl.com/newAIXwiki
7. → Other Performance Tools
8. tinyurl.com/PerfToolsForum (also AIX Forum)
9. IBM Rebooks
10. IBM POWER Technical Universities

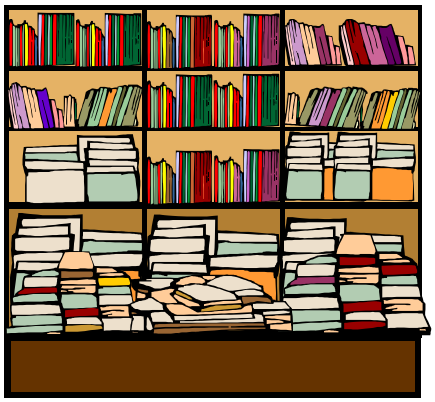
Nigel Griffiths
IBM Power Systems
Advanced Technology, Europe

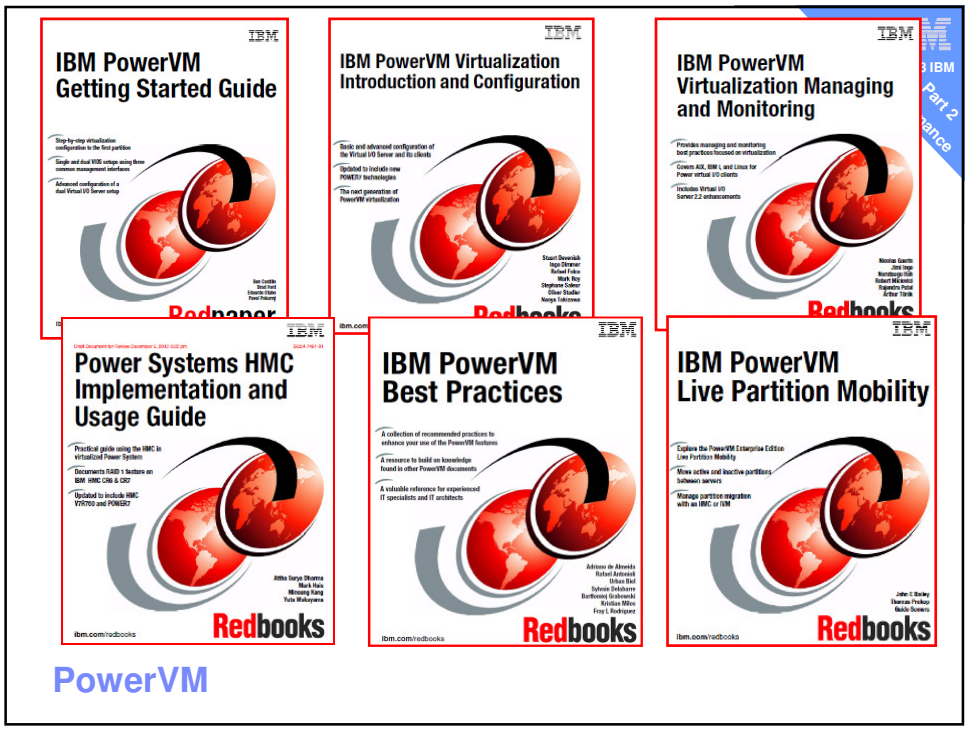
AIX Virtual User Group

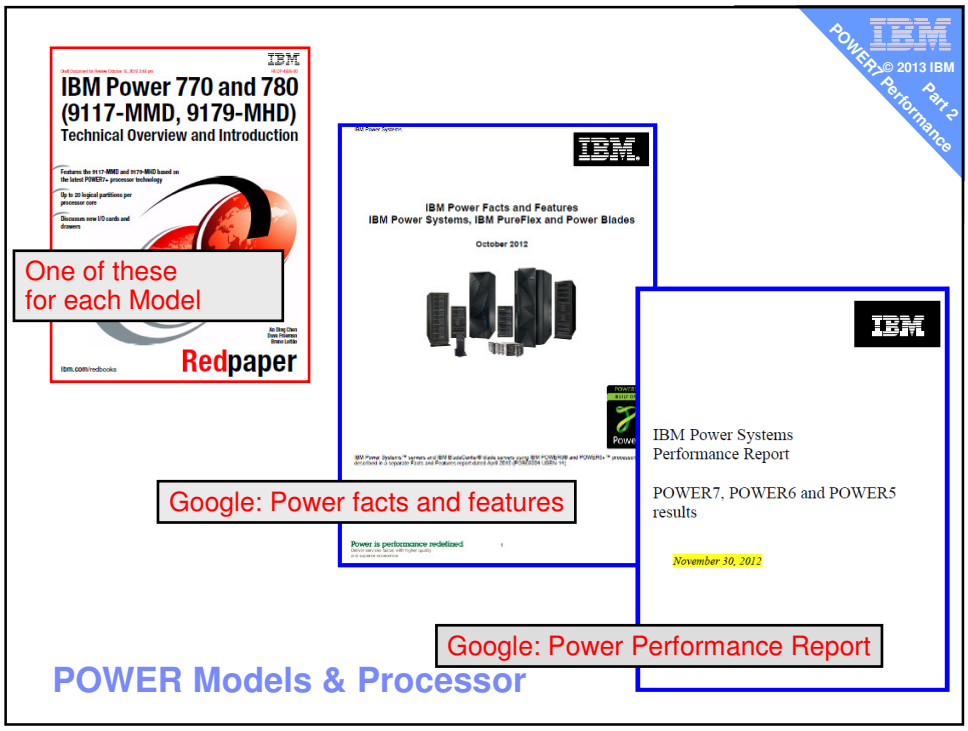
- Rosa Davidson → USA Advanced Technical Skills
- Very popular two part session on Entitlement & Virtual Processor setting on POWER7
  - **Highly recommended**
  - Part 1 last week & Part 2 on Thursday Jan 31st
  - Both will have downloadable Replays
  - She explains in details why you need to lower VP when you move to POWER7 plus the importance of Entitlement to reflect real expected CPU use
- Join the calls or get you Replays from tinyurl.com/newAIXVirtualUserGroup

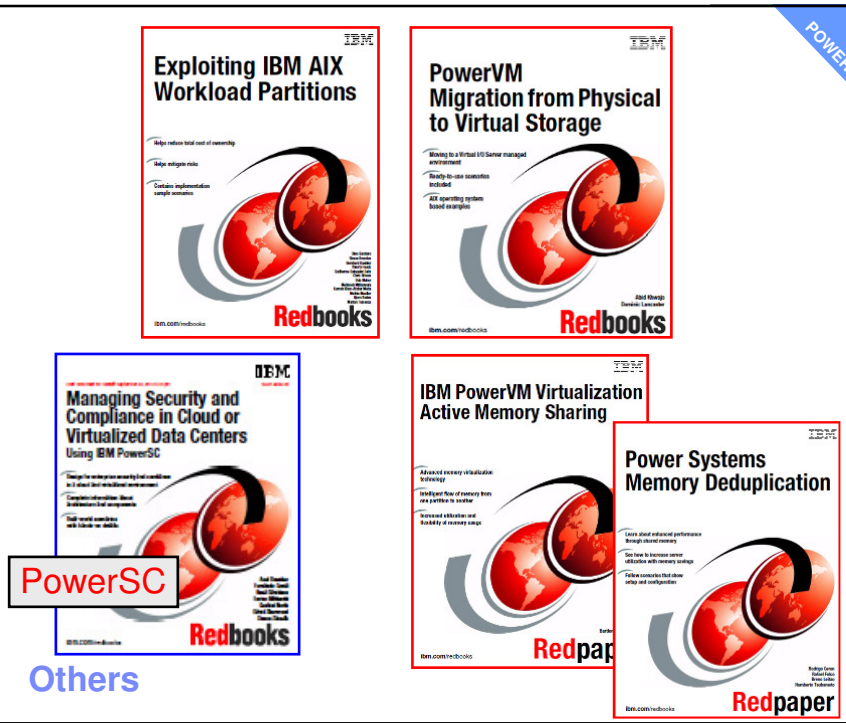
IBM © 2013 IBM POWER7 Performance Part 2

What is on my Techie Book Shelf?




IBM PowerVM Getting Started Guide
IBM PowerVM Virtualization Introduction and Configuration
IBM PowerVM Virtualization Managing and Monitoring
Power Systems HMC Implementation and Usage Guide
IBM PowerVM Best Practices
IBM PowerVM Live Partition Mobility
Redpaper
Redbooks
PowerVM


IBM Power 770 and 780 (9117-MMD, 9179-MHD) Technical Overview and Introduction
Redpaper
One of these for each Model
IBM Power Facts and Features
IBM Power Systems, IBM PureFlex and Power Blades
October 2012
Google: Power facts and features
IBM Power Systems Performance Report
POWER7, POWER6 and POWER5 results
November 30, 2012
Google: Power Performance Report
POWER Models & Processor

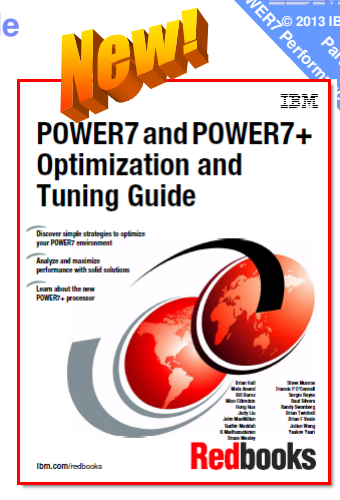


POWER7 Optimization & Tuning Guide

A single “first stop” definitive source for a wide variety of general information and guidance, referencing other more detailed sources on particular topics

Redbook SG24 8079

- Lots of guru level
  Advanced Technical content
- Section 3.2.3 – Physical LPAR placement using Affinity Groups - more later in the session



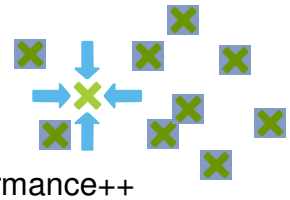
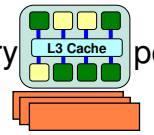
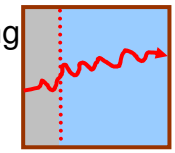
POWER Optimisers and Advisors

- A Quick Reminder only
- We have PowerVM sessions in plan for these in the next few weeks

Power Advisers & Optimisers

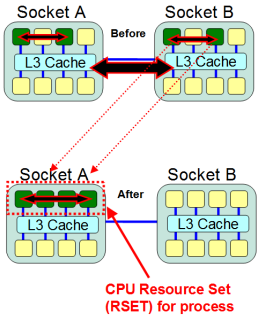
- A Quick Reminder only
- We have PowerVM sessions for these in the next few weeks

ASO/DSO in Operation Overview

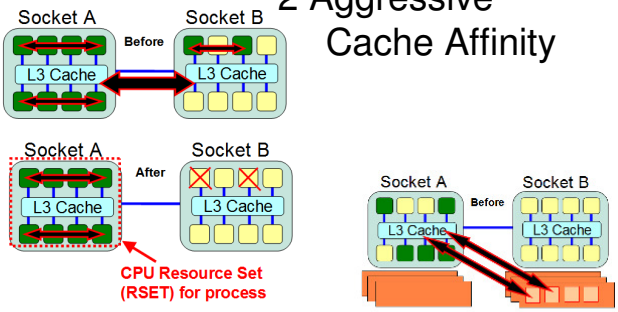
1. Once activated  requires no user involvement
2. Identifies & optimizes suitable workloads 
3. Improves cache & memory  performance++
4. Performs pre- & post-optimization monitoring 
5. Hibernates when not busy 

ASO Technical Information: Optimizations

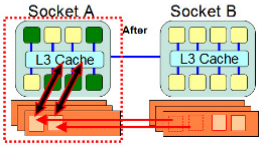
1 Cache Affinity



2 Aggressive Cache Affinity



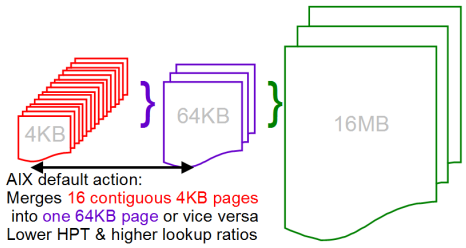
3 Memory Affinity



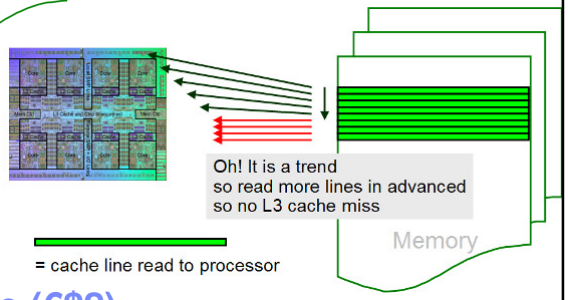
ASO is part of AIX

DSO 4 Technical Information: Optimizations

4 Dynamic migration to Large Pages (16MB)



5 Data Stream Pre-fetch



DSO separate package (€$£)
using the ASO framework

Active System Optimizer Summary

1. "Set & forget"

2. Advanced Autonomic Affinity Tuning
   - Low CPU impact with zero negative effects
   - High performance impact

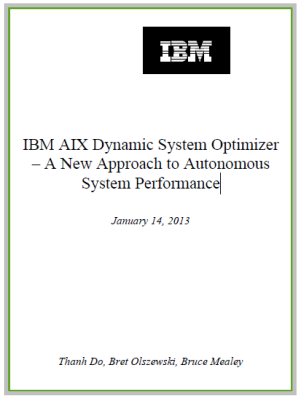
3. Particularly good for
   - Complex, multi-threaded, long running processes
   - Large CPU + RAM LPARs on larger machines
   - Reduced man-power & complex setup for large page use

Dynamic System Optimizer (DSO)

More Information:

- IBM AIX Dynamic System Optimizer Whitepaper
  A New Approach to Autonomous System Performance
- http://public.dhe.ibm.com/common/ssi/ecm/en/pow03093usen/POW03093USEN.PDF

- Covers ASO and DPO
- Has benchmark results from many workloads with results & graphs
- Includes installing DSO


IBM

IBM AIX Dynamic System Optimizer – A New Approach to Autonomous System Performance

January 14, 2013

Thanh Do, Bret Olszewski, Bruce Mealey

Power Advisors

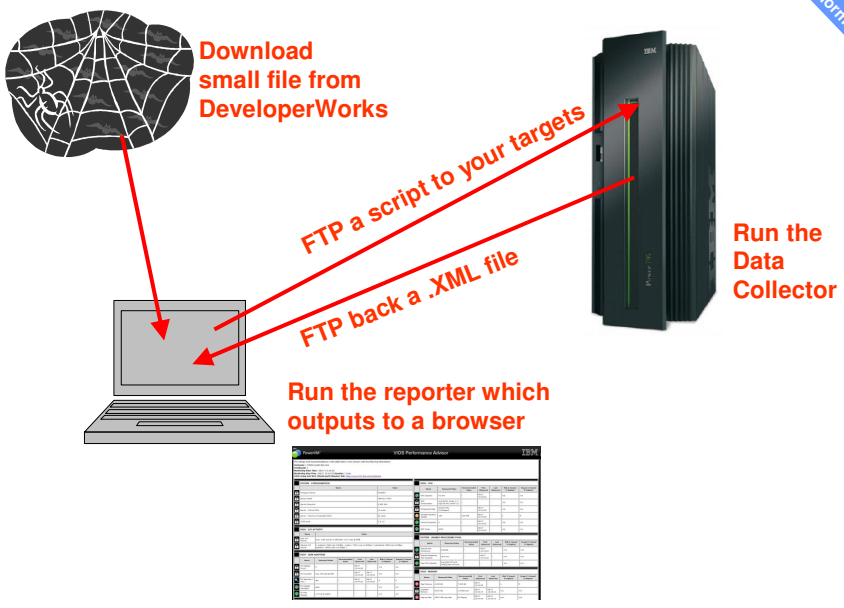
Would you like some advice on:

A. LPAR Performance
B. Java on POWER7 Performance
C. VIOS Performance

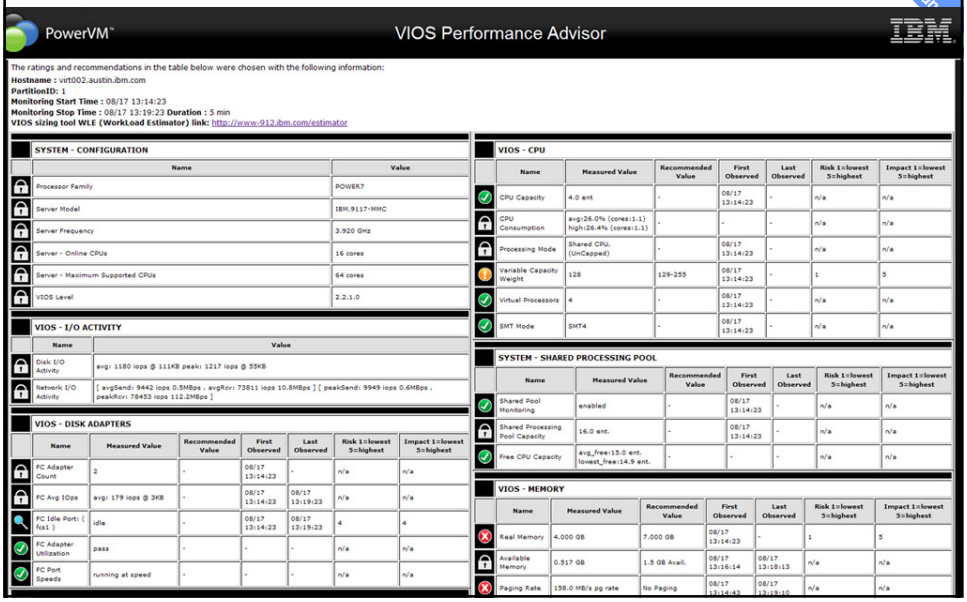
- Popular - 1000's of downloads in a few months
- Actively being development by AIX performance team
- Two movies from me tinyurl.com/newAIXmovies

IBM
POWER7 Performance Part 2
© 2013 IBM

Advisors on IBM DeveloperWorks



Example



PowerVM™ VIOS Performance Advisor IBM

The ratings and recommendations in the table below were chosen with the following information:
Hostname : virt002.austin.ibm.com
PartitionID: 1
Monitoring Start Time : 08/17 13:14:23
Monitoring Stop Time : 08/17 13:19:23 **Duration :** 5 min
VIOS sizing tool WLE (WorkLoad Estimator) link: http://www-912.ibm.com/estimator

SYSTEM - CONFIGURATION

Name	Value
Processor Family	POWER7
Server Model	IBM,9117-MMC
Server Frequency	3.920 Ghz
Server - Online CPUs	16 cores
Server - Maximum Supported CPUs	64 cores
VIOS Level	2.2.1.0

VIOS - I/O ACTIVITY

Name	Value
Disk I/O Activity	avg: 1180 iops @ 111KB peak: 1217 iops @ 55KB
Network I/O Activity	[avgSend: 9442 iops 0.5MBps , avgRcv: 73811 iops 10.8MBps] [peakSend: 9949 iops 0.6MBps , peakRcv: 78453 iops 112.2MBps]

VIOS - DISK ADAPTERS

Name	Measured Value	Recommended Value	First Observed	Last Observed	Risk 1=lowest 5=highest	Impact 1=lowest 5=highest
FC Adapter Count	2	-	08/17 13:14:23	-	n/a	n/a
FC Avg IOps	avg: 179 iops @ 3KB	-	08/17 13:14:23	08/17 13:19:23	n/a	n/a
FC Idle Port: (fcs1)	idle	-	08/17 13:14:23	08/17 13:19:23	4	4
FC Adapter Utilization	pass	-	-	-	n/a	n/a
FC Port Speeds	running at speed	-	-	-	n/a	n/a

VIOS - CPU

Name	Measured Value	Recommended Value	First Observed	Last Observed	Risk 1=lowest 5=highest	Impact 1=lowest 5=highest
CPU Capacity	4.0 ent	-	08/17 13:14:23	-	n/a	n/a
CPU Consumption	avg:26.0% (cores:1.1) high:26.4% (cores:1.1)	-	-	-	n/a	n/a
Processing Mode	Shared CPU. (UnCapped)	-	08/17 13:14:23	-	n/a	n/a
Variable Capacity Weight	128	129-255	08/17 13:14:23	-	1	5
Virtual Processors	4	-	08/17 13:14:23	-	n/a	n/a
SMT Mode	SMT4	-	08/17 13:14:23	-	n/a	n/a

SYSTEM - SHARED PROCESSING POOL

Name	Measured Value	Recommended Value	First Observed	Last Observed	Risk 1=lowest 5=highest	Impact 1=lowest 5=highest
Shared Pool Monitoring	enabled	-	08/17 13:14:23	-	n/a	n/a
Shared Processing Pool Capacity	16.0 ent.	-	08/17 13:14:23	-	n/a	n/a
Free CPU Capacity	avg_free:15.0 ent. lowest_free:14.9 ent.	-	-	-	n/a	n/a

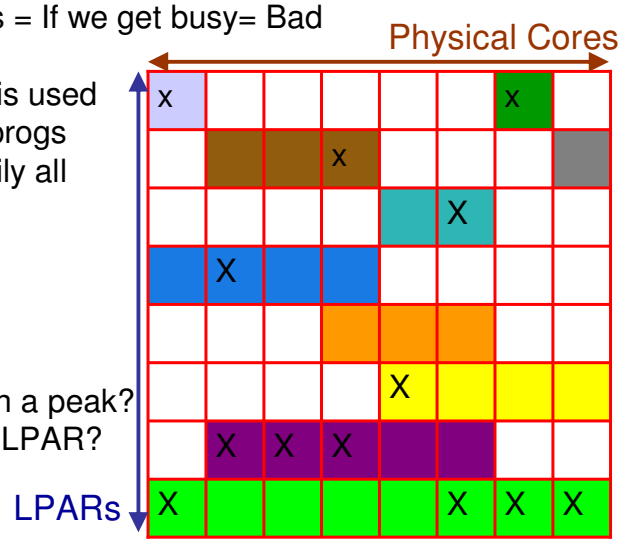
VIOS - MEMORY

Name	Measured Value	Recommended Value	First Observed	Last Observed	Risk 1=lowest 5=highest	Impact 1=lowest 5=highest
Real Memory	4.000 GB	7.000 GB	08/17 13:14:23	-	1	5
Available Memory	0.517 GB	1.5 GB Avail.	08/17 13:16:14	08/17 13:18:13	n/a	n/a
Paging Rate	158.0 MB/s pg rate	No Paging	08/17 13:14:43	08/17 13:19:10	n/a	n/a

Finding ‘Spare’ Capacity

Finding 'Spare' Capacity

- POWER7 spreads across cores to maximise performance
- 32 VP on 8 cores = If we get busy= Bad
- Here every core is used with a few busy progs but not necessarily all SMT threads
- lparstat app = 0
  No spare cores
- Can we cope with a peak?
- Can I add a new LPAR?



Finding Spare Capacity

- Low Entitlement, uncapped, high VP & low thread #'s
  = all physical CPUs rapidly get used
- Looks like 100% machine used no spare CPU in Pool
  = plenty of unused 2nd, 3rd, 4th threads
- User says
  "No spare CPUs in the pool"
  "You promised I could run 3 more VMs!!"
  "How can I measure the 'spare' capacity?"

Finding Spare Capacity

Answer:

- I depends!
- Any unused physical CPU time in the pool?

But if CPU empty …

- If my car uses all 4 cylinders driving to the local shops; How fast can I go on the Autobahn?
- We can't tell → Optimised for max perf. not min HW use
- Lots of island of overlapping spare capacity
- No single statistic can tell you
- A right pain if you have loads of VMs!

Finding Spare Capacity

Approach 1:

- Review every VM to raise E to peak use (efficiency) & lower VP (force higher SMT thread use)
- Then monitor CPU pool for unused CPUs
- Downside: man-power intensive + lots of VM changes

Approach 2:

- Fix a few large Virtual Machines (as above)
- Start removing 1 CPU each day until it hurts
  - Remove by Parking them in a dedicated CPU LPAR
- If performance dips → DLPAR a CPU back to the pool
- Parked CPUs = spare

Physical LPAR Placement

Finding out about Physical LPAR Placement

- lssrad –av gives you logical placement
  - All relative, starts REF1=0, SRAD=0
- 99% of the time pretty good
- 1% of the time “could do better” often due to history
  - Start large production LPAR after the small ones
  - Create large LPAR months later
  - Large DLPAR of LPM changes over months
- So how bad are my LPARs placed?
  - lssrad can show bad orphan CPU’s and Memory
  - There is no publically available tool

Finding out about Physical LPAR Placement

- After 12 years of logical CPU and memory
  - IBM does not want to encourage you to do this
  - Does not want you to worry about it
  - Does not spend man-days time physically laying out your LPARs – major step backwards to the previous century
- Placement tools
  - Only used to investigate/explain a performance issue

Given ½ a chance the Hypervisor will do the right thing

Finding out about Physical LPAR Placement

- There are internal IBM Support tools
  - Not documented, not public, not for users nor recommended
  - Only used in PMR problem diagnostics
  - Data extracted via machine dumps on the HMC
  - The data is largely binary format
- Two types
  - Full non-destructive live dump – exact details
  - Quick Summary dump – can get out of date

Don't ask me how to do this as I am not allowed to say

Example of Summary – Two CEC Power 770
Just for back ground information

Domain		Procs		Units	Memory			Proc Units		Memory		Ratio	
SEC	PRI	Total	Free	Free	Total	Free	LP	Tgt	Aloc	Tgt	Aloc		
0		1600	400	20	512	84						656	
	0	800	300	0	256	32						331	
							13	500	500	205	205		
	1	800	100	20	256	52						1625	
							1	200	200	32	32		
							5	50	50	16	16		
							13	100	100	51	51		
							14	80	80	16	16		
							15	50	50	16	16		
							16	50	50	16	16		
							18	50	50	16	16		
							20	50	50	16	16		
							21	50	50	16	16		
1		1600	100	0	512	248						7750	
	4	800	100	0	256	63						1968	
							3	100	100	16	16		
							4	100	100	16	16		
							11	200	200	64	64		
							12	200	200	72	72		
							17	100	100	2	2		<-
							19			16	16		
	5	800	0	0	256	185						0	
							17	800	800	14	14		<-
							24			51	51		

Notes:

- SEC = the CEC in the Power 770 – we have just two
- PRI = POWER7 chips with 8 cores each on this machine hence 800
- LPAR placement is good in this example
- "Procs" in 100ths of a CPU to avoid floating point maths in the kernel
- "Memory" is in LMB (Logical Memory Block) size chunks on this machine 128MB
  - This machine has 128 GB of memory
- "LP" is the LPAR number as seen on the HMC

IBM © 2013 IBM POWER7 Performance Part 2

Entitlement, Memory & Partition Placement

Shared CPU VM Partition Placement

- On VM start, Hypervisor tries to localise base on:
  - Primarily driven by Entitlement & Desired Memory +Page tables
  - Much less important Virtual Processor & Max Memory
  - If undersized Entitlement home CPUs shared with many VM
  - When VP scheduled=forces use of near/far SRADs
  - Max. Memory force Page table allocation (don't go massive)
  - Power 795 SPPL settings
  - Capacity-On-Demand CPU/RAM can help flexibility (fw730+)
  - Watch those internal boundaries
    - Core per chip, chips per CEC
    - RAM per chip, RAM per CEC

If you suspect bad placement! So what can you do?

1. Use lssrad –av to build a picture 
2. Restart the machine from cold = total rethink then start large & important LPARs first 
   - The Hypervisor does the right thing
   - Can be painful to schedule
3. Start LPAR with 0.1 CPU & 1GB RAM profile then restart the regular profile 
   - This gets the Hypervisor to rethink placement
4. Use DPO – needs 760+ firmware 
5. Use Affinity Group – needs 730+ firmware 
6. If you have bad performance – raise a PMR 

DPO
Dynamic Platform Optimiser

Platform = POWER7 machine

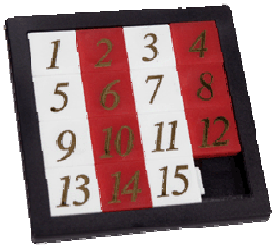
I call it the LPAR shuffler or VM defrag

IBM
POWER7 Performance
© 2013 IBM
Part 2

Dynamic Platform Optimizer (DPO)

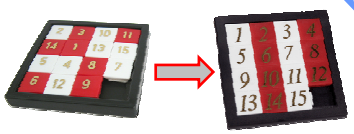
POWER Virtual Machines before DPO →

After DPO →



Cool right ☺

Dynamic Platform Optimizer (DPO)



Firmware 760+ mandatory
- Currently
  - Power 770/780 D = POWER7+
  - Power 795 = POWER7
- Plus new POWER7+ boxes coming soon
- Machine Properties →Capabilities →

Dynamic Platform Optimization Capable	True

HMC command: optmem & lsmemopt

1. lsmemopt -m $MACHINE -o currscore
   curr_sys_score=37
2. lsmemopt -m $MACHINE -o calcscore
   curr_sys_score=37,predicted_sys_score=92, ……..
3. optmem -m $MACHINE -o start -t affinity
   … takes a lot of time here
4. optmem -m $MACHINE -o stop -t affinity

- Can also DPO 1 or a set of LPARs or all except certain LPARs
- Hint to get the LPARNAME use: lssyscfg -r sys -F name

IBM POWER7 Performance Part 2
© 2013 IBM

Affinity Group LPAR Placement

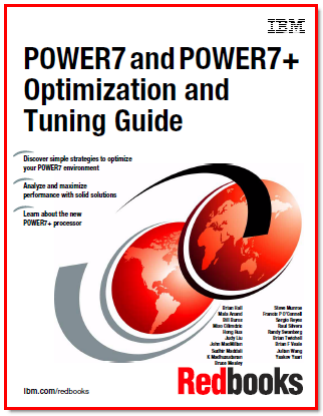
This is for awareness & not for you to try later today ☺

Affinity Group

- Firmware 730+ [latest level please!]
- Only to be used to address very specific performance issues i.e. you know more than the Hypervisor

- L3 Support should be involved
  = not simple

- Redbook SG24 8079
- Section 3.2.3 – Physical LPAR placement using Affinity Groups



Affinity Group

- Group id = 255, 254, 253, 252 …… 1
- Assuming you ensure all the cores/RAM in one group_id can't fit in one CEC/book
  - 255 is first CEC or Book
  - 254 next CEC or Book, etc.
- This is set with the HMC command (no GUI)
  - chsyscfg -r prof -m <system_name>
    -i name=<profile_name>
    lpar_name=<partition_name>,affinity_group_id=<group_id>
- Example:
  - chsyscfg -r prof -m myPower770
    -i name=normal
    lpar_name=LPAR_42,affinity_group_id=255

Affinity Group

- Treat each group_id = CEC/Book as a bucket
- You need to assign LPAR(s) with
  - Entitlement & VP
  - and memory sizes

  that fit and you have to allow for memory for
  - Page tables 1/64th
  - Hypervisor – tricky System Planning tool can help
  - DMA buffers for adapters – ditto
- You end up with a spreadsheet to write HMC script

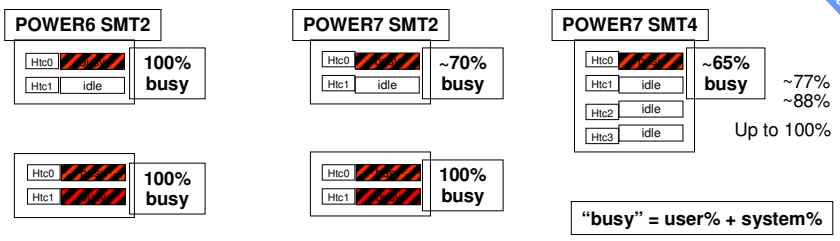
Affinity Group

- Apply the HMC script to the LPAR profiles
  - Can do this in advance with the LPARs running
- Shutdown the machine
- Cold start the machine
- Hypervisor places them the way you like

- Then get the Physical LPAR placement
  - With L3 Support (of course)

Utilisation Numbers have been fiddled with!

POWER6 vs POWER7 SMT Utilization



POWER7 SMT=2 70% & SMT=4 65% **tries to show potential spare capacity**

- Escaped most people attention
- VM goes 100% busy at entitlement & 100% from there on up to 10 x more CPU
- IMHO Utilisation 100% is confusing and just about meaningless now!

Reference Whitepapers on POWER7 SMT and Utilization

Simultaneous Multi-Threading on POWER7 Processors by Mark Funk

http://www.ibm.com/systems/resources/pwrsysperf_SMT4OnP7.pdf

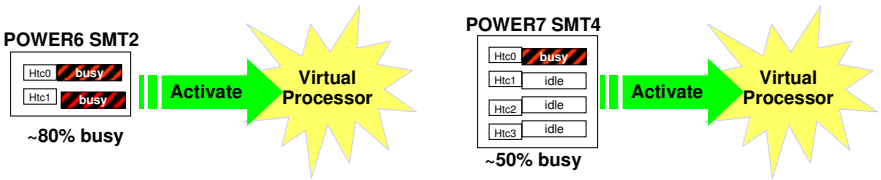
Processor Utilization in AIX by Saravanan Devendran

https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/Power%20Systems/page/Understanding%20CPU%20utilization%20on%20AIX

POWER6 vs POWER7 Virtual Processor Unfolding

Virtual Processor is activated at the utilization threshold below

- both systems report physical consumption of 1.0



In POWER5 & 6

- 1st & 2nd SMT threads are loaded to ~80% utilization then VP unfolded

In POWER7 – called Raw Throughput mode

- 1st threads (on each VP) are loaded ~50% utilization then VP unfolded
- Only then 2nd threads are used
- Once 2nd threads are loaded, only then more threads used

Why?

- Raw Throughput provides the highest per-thread throughput and best response times at the expense of activating more physical cores

IBM
© 2013 IBM
POWER7 Performance
Part 2

POWER7 Consumption: A Problem?

1. With excess VP's - POWER7 may activate more cores at lower utilization levels than POWER5 & 6
2. Customers may complain that the physical consumption (physc or pc) metric is equal to or even higher after migrations. They may also note that CPU capacity planning is more difficult in POWER7
3. Expect complaining POWER7 customer to also have significantly higher idle% percentages
4. If consolidating workloads they may also have many more VP's assigned to the POWER7 partition

Scaled Throughput

IBM
© 2013 IBM
POWER7 Performance
Part 2

Scaled Throughput?

POWER7 & POWER7+ with
AIX 6.1 TL08 & AIX 7.1 TL02

- It will dispatch more SMT threads to a VP core before unfolding additional VPs
- Considered a bit more like POWER6 unfolding but is a *generalization*, not a technical statement

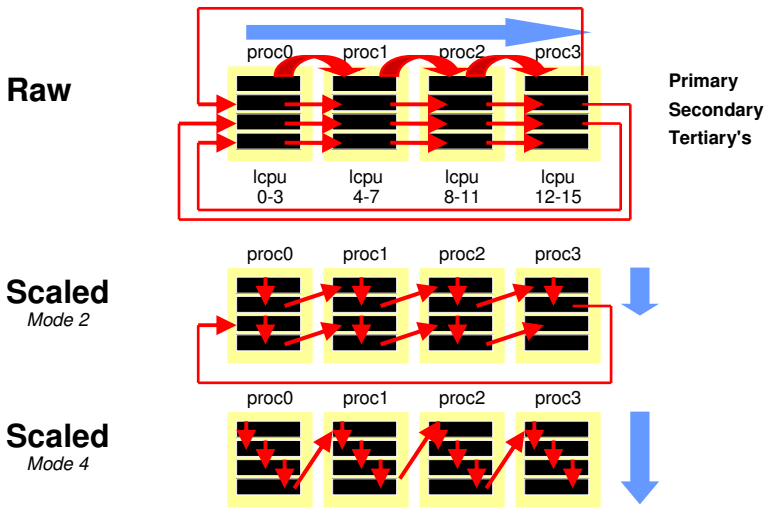
What is Scaled Throughput?

- **Raw** provides the highest per-thread throughput and best response times at the expense of activating more physical core

- **Scaled** provides the highest core throughput at the expense of per-thread response times and throughput.
  It also provides the highest system-wide throughput per VP because tertiary thread capacity is “not left on the table.”

IBM
© 2013 IBM
POWER7 Performance
Part 2

Raw vs Scaled Throughput



Scaled Throughput: Tuning

- Not restricted, but
  anyone experimenting without understanding may suffer significant performance impacts

- `schedo -p -o vpm_throughput_mode=`
  - `0` Legacy Raw mode (default)
  - `1` "Enhanced Raw" mode with a higher threshold than legacy
  - `2` Scaled mode, use primary and secondary SMT threads
  - `4` Scaled mode, use all four SMT threads

- Dynamic tunable

Scaled Throughput: Workloads

- **Workloads**
  - Workloads with many light-weight threads with short dispatch cycles and low IO (the same types of workloads that benefit well from SMT)
  - Customers who are easily meeting network & I/O SLA's may find the tradeoff between higher latencies & lower core consumption attractive
  - Customers who will not reduce over-allocated VPs & prefer to see behavior similar to POWER6
- **Performance**
  - *It depends,* we can't guarantee what a particular workload will do
  - Mode 1 may see little or no impact but higher per-core utilization
  - Workloads that do not benefit from SMT & use Mode 2 or Mode 4 could easily see double-digit per-thread performance degradation (higher latency, slower completion times)

That's All Folks!