POWER7 Affinity & Performance
Part 1

Nigel Griffiths
IBM Power Systems
Advanced Technology Support, Europe

© 2012 IBM Corporation

tinyurl.com/AIXpert

Local, Near & Far Memory – Blog Article Series
1. Large Power7 boxes more local memory
2. Virtual Machine CPU & Memory Lay Out
3. Scheduling processes to SMT & Virtual Processors
4. Aggressive Intelligent Threads
5. Low Entitlement has a Bad Side Effect
6. Too High a Virtual Processor number has a Bad Side Effect
7. VM placement also needs RAM
8. Dynamic LPAR changes can mess up your placement
9. Firmware Updates for better Affinity
10. Final of the table by Model
11. Why Local+Far on Lower End machines?
   – POWER7 Affinity Nine Conclusions
12. I have a 10 core POWER7 chip, eh!
   – Plus AIX Virtual Processor Folding is Misunderstood
   – Many others: nmon, Systems Director, VIOS, …
The SMP curve & Cache Boundaries

- Recap – we all know this!

Well Known SMP Curve

- Standard large machine don’t scale perfectly linearly
- Distances & comm’s & sharing caches
- Localising your VM can give you a “free boost”

- Not to scale and drawn by hand!
Power 795

- **CPU books**: 1, 2, 4, 6, 8
- **Cores**: 64, 128, 192, 256
- **rPerf**: 372.27, 2978.16

### Calculation:
- \(372.27 \times 8 = 2978.16\)
- **Interpretation**: It is linear.
- **Conclusion**: So 256 rPerf is 8 x 32 rPerf.

- **One LPAR per CPU book**:

### Observations:
- Most LPARs are down in this area: 1 to 32 cores.

---

**POWER 2012 IBM**
Well Known Whole Core Boost

VM of 1.05 is slower than VM of 1.0 as the L2 cache is then split across more cores

- Not to scale and drawn by hand!

Some LPARs are down in this area → 1 to 8 cores

11% performance boost
POWER7

- 8 cores per chip
- Also 4 core & 6 core chips
- Increases the strain on the memory bus

POWER6

- 2 cores per chip
Reminders about rPerf for sizing

- Relative Performance measurement
- For comparing POWER machines and “bangs per buck”
- But a common source of misconception

rPerf for Sizing

- The assumptions have been forgotten = causes serious Sizing issues
- POWER6 10 CPU VM with rPerf=100
- POWER7 10 CPU VM with rPerf=150

Which is true?
- A. So application is 50% faster
- B. Utilisation will drop by 33%
- C. Batch will finish 33% quicker
- D. All of the above
- E. None of the above
rPerf for Sizing

- The assumptions have been forgotten = causes serious Sizing issues

- POWER6 10 CPU VM with rPerf=100
- POWER7 10 CPU VM with rPerf=150

Which is true?

A. So application is 50% faster
B. Utilisation will drop by 33%
C. Batch will finish 33% quicker
D. All of the above
E. None of the above

↵These are speed statements but rPerf is all about Throughput. Also comes with many assumptions …

rPerf for Sizing - Ten Golden Rules

1. Highly threaded workloads – CPUs x SMT x 2
2. Well tuned system – retuned from scratch
3. Full Spec RAM – no empty slots + lots of GBs
4. No Disk Issues – 100’s of disks, no bottlenecks
5. No Network Issues – tuned to zero bottlenecks
6. Current App. software – not previous generation
7. Latest AIX6/7 – latest TL + all service packs
8. Large LPARs – no micro-partitions
9. Firmware – latest
10. Bug Free – user willing to fix

AND the workload can be proved to be the same before and after = same number of transactions, dialogue steps, same size database, same SQL, record batch processed per minute/hour.
BLUNTEST WARNING

To get the POWER7 rPerf throughput, we have to use SMT4

Which can mean transaction take slightly longer!

If using worst case (P6 highest to P7 lowest GHz)
POWER6 5 GHz to
POWER7 3 GHz then this means similar or slightly lower thread speed and lower core speed

If POWER7 at higher GHz we may get to similar or slightly better thread speed slightly faster core speed lots of cores but higher throughput of work

How are machines build out of POWER7 Chips?

Strength of Power Systems is same chips & technology from bottom to top
### Bandwidth and Buses

- **New POWER7+ and Power 795 book**
  - Four POWER7 chips directly
  - Eight links to the other CPU books

- **Lower End = Far memory**
  - Two POWER7 chips directly connected using AB
  - XYZ not used

- **Memory Buses**
  - 3 XYZ bus
  - 2 AB bus
  - 8 AB bus

### CPU & Memory Affinity by POWER7 Model

<table>
<thead>
<tr>
<th>Model/RAM Access</th>
<th>POWER7 Chips/CPU</th>
<th>Local</th>
<th>Near</th>
<th>Far</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power7 blades</td>
<td>1</td>
<td>Same Chip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power7 blades</td>
<td>2</td>
<td>Same Chip</td>
<td>Other Chip</td>
<td></td>
</tr>
<tr>
<td>Power 710</td>
<td>1</td>
<td>Same Chip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power 730</td>
<td>2</td>
<td>Same Chip</td>
<td>Other Chip</td>
<td></td>
</tr>
<tr>
<td>Power 720</td>
<td>1</td>
<td>Same Chip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Power 740</td>
<td>2</td>
<td>Same Chip</td>
<td>Other Chip</td>
<td></td>
</tr>
<tr>
<td>Power 750/755</td>
<td>1 to 4</td>
<td>Same Chip</td>
<td>Other Chip</td>
<td></td>
</tr>
<tr>
<td>Power 770/780</td>
<td>2 to 8 / 64</td>
<td>Same Chip</td>
<td>Other 1 Chip but same CEC</td>
<td>Different CEC</td>
</tr>
<tr>
<td>Power 770/780 POWER7+ (dual core sockets)</td>
<td>4 to 16 / 128</td>
<td>Same Chip</td>
<td>Other 3 Chips but same CEC</td>
<td>Different CEC</td>
</tr>
<tr>
<td>Power 795</td>
<td>4 to 32 / 256</td>
<td>Same Chip</td>
<td>Other Chip but same CPU Book</td>
<td>Different CPU Book</td>
</tr>
</tbody>
</table>
Bandwidth and busses

- **Local memory**
  - Memory 68GB/s per memory controller (P7 has 2)
  - Power 770/780/795 uses both – the rest uses one
- **Near memory bus → XYZ Intra-node**
  - Only on 770/780 two chips* & 795 four chips, 8 byte wide
  - ~40 - 50 GB/s depends on the model
- **Far memory bus → AB Inter-node**
  - Power710-750 between chips = 4 byte wide (reduced cost)
  - Power 770/780 CEC & 795 book = 8 byte wide
  - 23 – 26 GB/s depends on the model

* POWER7+ models use the 4 core per chip

---

POWER7 mounting

- **Power 70x.x, 710-755**
  - Single Chip Organic
  - 1 Memory Controller = 68 GB/s
  - 4 Byte AB memory buses between chips = 23 GB/s each
- **Power 770-795**
  - Single Chip Glass Ceramic
  - 2 Memory Controllers total 136 GB/s
  - 8 Byte XYZ memory buses in the node
  - 8 Byte AB busses between nodes
Perspective & Perception

<table>
<thead>
<tr>
<th>Distance</th>
<th>Local</th>
<th>Good</th>
<th>Blisteringly fast</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Near</td>
<td>Bad</td>
<td>Excellent</td>
</tr>
<tr>
<td></td>
<td>Far</td>
<td>Ugly</td>
<td>Good</td>
</tr>
</tbody>
</table>

How does this effect me?

- Memory better if VM is:
  - Inside the 8 core POWER7 size
  - Inside the 16 core Power 770/780 CEC drawer *
  - Inside the 32 core Power 795 CPU book

- Note the VM size is Virtual Processor not Entitlement
- Used to determine the SRADs …

* 32 core POWER7+ Power 780
SRAD – eh!

- Scheduler Resource Affinity Domains
  - Groups of efficient CPU + memory
  - Hypervisor decides what you get
  - AIX works within SRADs to place processes with fastest RAM
  - Hierarchy of resources
    - Whole machine
    - CEC / Book
    - POWER7 chips
      - Cores
      - Threads

In POWER Logical Virtual Machines

- So how can we investigate
  - lssrad -av
  - mpstat –d
  - topas –M
  - svmon …

- Lets see some examples
**Issrad -av**

<table>
<thead>
<tr>
<th>REF</th>
<th>SRAD</th>
<th>MEM</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>29224.00</td>
<td>0-27</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2490.00</td>
<td>28-31</td>
</tr>
</tbody>
</table>

**Issrad -av**

- Only options that make sense!
- REF1
  - backplane, CEC drawer, or CPU book
  - Why REF1? = Reference !!!!
- SRAD → CPU+RAM group
- MEM → Megabytes!
- CPU
  - Logical CPU number
  - Assuming SMT=4

---

**Issrad -av**

If your process running here
- This is your Local memory &
- This is your Near memory &
- Memory in a different REF is Far memory

Example from Power 750 Local + Far
Issrad -av

<table>
<thead>
<tr>
<th>REF</th>
<th>SRAD</th>
<th>MEM</th>
<th>CPU</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td>29224.00</td>
<td>0-27</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>2490.00</td>
<td>28-31</td>
</tr>
</tbody>
</table>

28 Logical CPUs = 7 Physical CPU-Core

Total memory here= 31714 MB
but HMC 32 GB= 32768 MB
~3% less

BLOG 2

Investigating commands
Scheduler Resource Affinity Domains
Example: Power 770/780 → 8 POWER chips with 64 CPU-cores

REF 0
SRAD 1 → 2.5GB RAM + 1 core
SRAD 0 → 29GB RAM + 7 cores

But can we tell which CEC drawer or chip? No we can’t – logical resources

Local, Near & Far - relative to a process’ home
Memory is allocated on the processes home SRAD (if possible)

Local to the process
Near to the process
Far to the process

Power 770 → 8 POWER chips with 64 CPU-cores
Local, Near and Far is relative to your process and its data
Local, Near & Far - relative to a process’ home

If a process is schedule away from it’s home SRAD

Dedicated CPU 256 way

Power 795 + 4TB RAM

# lssrad -av
REF1 SRAD MEM CPU
0 0 94341.00 0 4 8 12 16 20 24 28
1 94711.00 32 36 40 44 48 52 56 60
2 94711.00 64 68 72 76 80 84 88 92
3 94711.00 96 100 104 108 112 116 120 124
4 94711.00 128 132 136 140 144 148 152 156
5 94695.00 160 164 168 172 176 180 184 188
6 94695.00 192 196 200 204 208 212 216 220
7 94695.00 224 228 232 236 240 244 248 252
8 94695.00 256 260 264 268 272 276 280 284
9 94695.00 288 292 296 300 304 308 312 316
10 94695.00 320 324 328 332 336 340 344 348
11 94695.00 352 356 360 364 368 372 376 380
12 94695.00 384 388 392 396 400 404 408 412
13 94695.00 416 420 424 428 432 436 440 444
14 94695.00 448 452 456 460 464 468 472 476
15 94695.00 480 484 488 492 496 500 504 508
16 93970.94 512 516 520 524 528 532 536 540
17 45421.00 544 548 552 556 560 564 568 572
18 94695.00 576 580 584 588 592 596 600 604
19 94695.00 608 612 616 620 624 628 632 636
20 94695.00 640 644 648 652 656 660 664 668
21 94695.00 672 676 680 684 688 692 696 700
22 94695.00 704 708 712 716 720 724 728 732
23 94695.00 736 740 744 748 752 756 760 764
24 94695.00 776 780 784 788 792 796 800 804
25 94695.00 808 812 816 820 824 828 832 836
26 94695.00 836 840 844 848 852 856 860 864
27 94695.00 864 868 872 876 880 884 888 892
28 94896.00 896 900 904 908 912 916 920 924
29 94880.00 928 932 936 940 944 948 952 956
30 94896.00 960 964 968 972 976 980 984 988
31 94309.00 992 996 1000 1004 1008 1012 1016 1020

Note: SMT=1 CPU numbers by first Logical CPU number 0, 4, 8, 12, ...
```
mpstat -d

mpstat -d 1 6
```

System configuration:
- lcpu=8
- ent=1.0
- mode=Uncapped

<table>
<thead>
<tr>
<th>CPU</th>
<th>s0rd</th>
<th>s1rd</th>
<th>s2rd</th>
<th>s3rd</th>
<th>s4rd</th>
<th>s5rd</th>
<th>s0hrd</th>
<th>s1hrd</th>
<th>s2hrd</th>
<th>s3hrd</th>
<th>s4hrd</th>
<th>s5hrd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>162</td>
<td>70</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>99.5</td>
<td>0.5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>1</td>
<td>42</td>
<td>28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>ALL</td>
<td>210</td>
<td>104</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>99.6</td>
<td>0.4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

```

s[0-5]rd = % of thread re-dispatches within a scheduling affinity domain

s[3-5]hrd % of thread dispatches on this logical processor

"-" → intelligent threads in action
mpstat –d 1 999 (thread dispatch = memory access)

Thread dispatches
- S3hrd → Local
- S4hrd → Near
- S5hrd → Far

Relative to home SRAD if away from home then Home memory access is also Local, Near, Far

Note: vlcs highlights how often used

# mpstat –d 1 999

<table>
<thead>
<tr>
<th>cpu</th>
<th>S3hrd</th>
<th>S4hrd</th>
<th>S5hrd</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>100.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>17</td>
<td>80.0</td>
<td>0.0</td>
<td>20.0</td>
</tr>
</tbody>
</table>

Topas –M or topas and the hit M or nmon then ~ then M

Yes it is a bit of a mess to highlight some things
### SRAD Memory View

<table>
<thead>
<tr>
<th></th>
<th>TOTALMEM</th>
<th>FREE</th>
<th>FILECACHE</th>
<th>HOMEThRES</th>
<th>CPUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>24.5G</td>
<td>17.7G</td>
<td>9.9G</td>
<td>0.0G</td>
<td>3.5G</td>
</tr>
<tr>
<td>1</td>
<td>17.9G</td>
<td>32.6G</td>
<td>63.6G</td>
<td>217.7G</td>
<td>5</td>
</tr>
</tbody>
</table>

- **Home processes/threads**: Limited room
- **File system cache MB (numperm)**: Limited room
- **CPUs**: Limited room

You know this from lssrad.
In CPU Busy Order

- **SMT=4**: 1 physical core
- **All logical CPUs in use?**
- **Some Near memory access**
- **Most work on 1st SMT**
- **Move the cursor here to order the Logical CPUs otherwise ordered on busy CPU**

**Good**

**Excellent**

**Blisteringly Fast**

**Very little use so 100% not that important**
svmon – just for reference

- Report global affinity domains more detail than lssrad:
  - `svmon -G -O affinity=on,unit=MB`

- Report memory statistics
  - `svmon -P [PID] -O threadaffinity=on and -O affinity=detail`

---

```
# svmon  -G -O
unit=auto,timestamp=on,pgsz=on,affinity=detail

Unit: auto                                           Timestamp: 21:18:40
--------------------------------------------------------------------------------------
size       inuse free         pin     virtual  available   mmode
memory        64.0G       4.57G       59.4G       3.28G       4.53G      59.4G     Ded
pg space    512.00M     203.60M
work        pers clnt other
pin           1.42G          0K          0K 1.86G
in use        4.37G          0K     204.76M
PageSize PoolSize inuse pgsp pin     virtual
s    4 KB         - 2.75G     203.60M       2.45G       2.71G
Domain affinity        used
  3     136447
  0     44655
  2     21703
  1     31757
m     64 KB         - 1.82G          0K     847.12M       1.82G
Domain affinity        used
  3     13280
  0     16208
  2     11328
  1     13136
```

Missing! File Cache heading
on my beta AIX version
BLOG 3

Process Thread Scheduling to SMT Threads

Scheduling to SMT

- **POWER5 & 6**
  - SMT=2 and modes on or off

- **POWER7**
  - Up to SMT=4 and modes 1, 2, 4
  - And intelligent threads – auto switching mode
    - 1 runnable thread $\rightarrow$ SMT=1 mode
    - 2 runnable threads $\rightarrow$ SMT=2 mode
    - 3 or 4 runnable threads $\rightarrow$ SMT=4 mode

- A physical processor core is running one LPAR at a time
  - So all four SMT’s in one LPAR at a time
  - AIX schedules work on these are four logical processors via run queues
Power7 12 execution units

- 2 integer units
- 2 load-store units
- 4 double-precision floating-point units
- 1 branch unit
- 1 condition register unit
- 1 vector unit
- 1 decimal floating-point unit

Four people in a small caravan
Works fine provided they don’t all want to do the same thing at the same time

Two can sit but only one can cook
So they have to take turns
### Simultaneous Multi-Threading (SMT)

- Over all more gets done
- Individual threads go a bit slower
- Good for throughput of many transactions
- Response time a little longer than SMT=1

#### Example

- Virtual machine (LPAR)
- Entitlement of 1.5 (Uncapped)
- Virtual processor count of 4
- AIX is set to SMT=4 on our POWER7 machine
- So each CPU-cores has four SMT threads
  - = 16 logical CPUs

Tested on Firmware 730 & AIX 7.1 TL1
One busy (running 100% of the time) program

Where does the 2nd program go?

Some times you see SMT2 used but then return to SMT1

POWER7 is proactive moving to SMT1
Third busy program?

Four busy program?

What is the Utilisation?

Physical core Utilisation? Logical processor Utilisation? How much head room is there? 100% 25% meaningless We don’t know! If enough threads & SMT friendly, ... guess 30% to 60%
One more busy program – where?

Sixth busy program – where?

Eight busy programs

Ninth busy program – where?

No SMT=3, 1st core goes SMT4 runs four threads, so 2nd core goes SMT=1 Note: now we have SMT=4 + SMT=1 and SMT=2
Ten busy programs

Eventually 16 programs

Where does the 17th go?

Classic time sharing on the logical CPUs
Advanced points in using POWER7 SMT=4

1. With 4 programs, 8 programs or 16 programs
   ALL physical cores were 100% busy

2. If you don't have enough runnable processes
   (run queue), you can't use SMT=4 and
   you don't get the full POWER7 rPerf

3. Not enough processes (or process threads)
   – Tune app or middleware to use more
   – Reduce your VP count! NOT the Entitlement
   – Get the users to work faster!!

HOWEVER

- Above was for Spinning processes = 100% busy

- Typical workloads 100’s of processes taking factions
  of a second, so harder to determine if it needs more
  CPU resources

- AIX uses % thresholds to determine when to switch:
  – On more cores or
  – On more SMT threads

- These % were subject to fixes for AIX6 TL5 +TL6
Comparing POWER6 and 7

Regular question: Is POWER7 broken?
– “fake” 8 program steady workload

We see POWER Intelligent threads working & POWER7 is working very well
POWER7 Aggressive Intelligent Threads

- POWER6 in SMT=2 not bothered which thread

- POWER7 moves process threads to SMT1 (or 2) proactively and switches to SMT=1 (or 2) mode for higher efficiency

- In other stats POWER6 & 7 both using ~2.5 CPUs but on POWER7 its obvious we can remove a CPU

- You CAN’T find the 2.5 on the previous graphs as you can’t average logical CPUs intermixing on the physical CPU. \( (70+10+0+0)/4=20\% \)

BLOG 5

Low Entitlement ➔ Bad Side Effect
Our VM runs until E consumed
Then must yield to make sure other VM get their E

10 millisecond dispatch wheel cycle in which every VM gets its Entitlement

Other VMs running either stopped as E used up or no more work to do
All VM's have run to their Entitlement or ran out of work & yielded the CPU

10 millisecond dispatch wheel cycle in which every VM gets its Entitlement

Time

4 milliseconds left, divided between still running VMs according to weight factor and run again

10 millisecond dispatch wheel cycle in which every VM gets its Entitlement

Time
10 millisecond dispatch wheel cycle in which every VM gets its Entitlement

- Time

1 millisecond left divided between VMs again ...

Dispatch wheel cycle complete
- Start again running VMs to get their full Entitlement
Our VM runs in three parts within 10 millisecond window

Our VM runs in three parts but other VM have used the CPU caches removing out VM's cache lines. So it will have many cache misses.
As our VM runs it needs to warm up the CPU caches each time – this means it not running at full speed for a while.

Higher Entitlement lets it run continuously with less cache warm up time.
Lessons

- Don't forget 10 milliseconds is a long time on a CPU
  - At 4 GHz = 400,000,000 instructions (assuming 1 op/cycle)
  - Illustration was grossly simplified by factor 10 or more

- Get the Entitlement "about right"

Next Week

POWER7 Affinity and Performance
Part 2 “same time, same channel”

- 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
  - Physical VM placement
  - VM “defrag”
  - Working out “space capacity”
  - Getting POWER7 to look more like POWER5/6

Nigel Griffiths
IBM Power Systems
Advanced Technology Support, Europe