SMT and MFC for z/VM Understanding and Using

Velocity Software Inc.
196-D Castro Street
Mountain View CA 94041
650-964-8867

Velocity Software GmbH
Max-Joseph-Str. 5
D-68167 Mannheim
Germany
+49 (0)621 373844

Barton Robinson,
barton@velocitysoftware.com

If you can't measure it, I'm just not interested….

Topics

Point of discussion: Is SMT adding capacity?

Hardware instrumentation: MainFrame Cache

- **IBM Z Architecture**
- **Understanding MFC (smf 113)**
- **CPI,**
- **RNI**
- **Z13 vs z14/z15**

SMT Overview

VELOCITY SOFTWARE

MFC Terms

Terms

- L1 / L2 Cache: Core level cache
- L3 Cache: Chip level
- L4 Cache: Node/Drawer Cache
- Local vs Remote (L4R)
- MFC: Mainframe Cache
- CPI: Cycles per instruction
- RNI: Relative nest intensity

Instruction components

- L1 Load: Data, Instruction
- DAT: Direct Address Translation
- Execution

VELOCITY SOFTWARE

Hardware Architecture

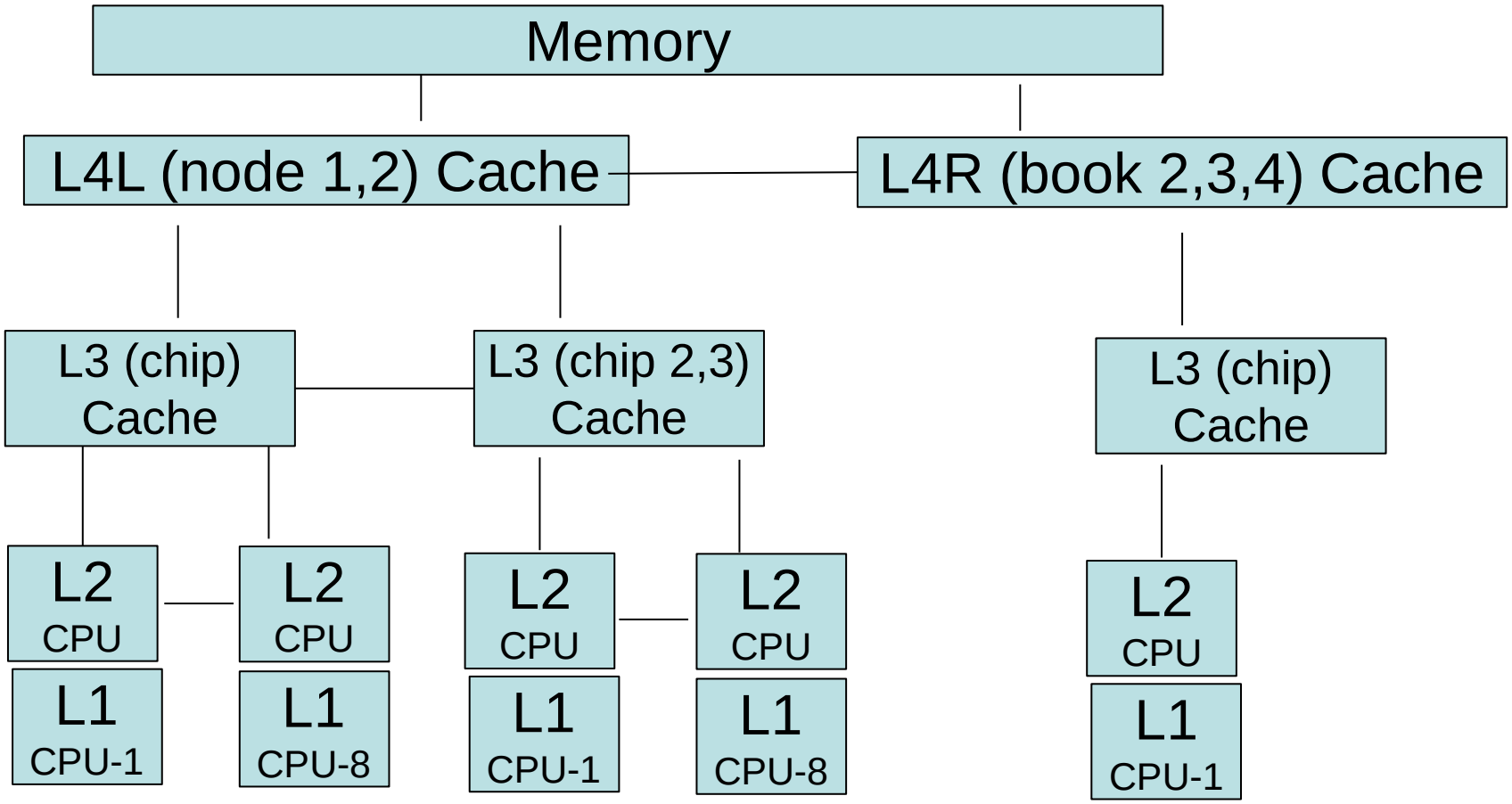
For instruction to execute, L1 cache includes

- Data (at possibly two locations)
- Instruction (and possibly branch location)

If data/instruction not in cache, must load from:

- L2 cache (1 cycle)
- L3 cache
- L4 cache (local, remote)
- Memory
- Architecture changes on every CPU model

Z (z13) Architecture



Cores, Chips (multiple cores), node/book (Multiple chips)

What is the CPU Measurement Facility

- Hardware instrumentation
- Statistics by virtual CPU / thread in z/VM LPAR
- 5.13 Monitor records (PRCMFC) (Basic, Extended)
- "Extended" different for z10,196,EC12, z13/14/15/16
- Shows cycles used, instructions executed and thus CPI Ratio

```
Report: ESAMFC          MainFrame Cache Analysis Re
Monitor initialized: 02/27/15 at 20:00:00
-------------------------------------------------
                <CPU Busy> <-------Processor------>
                <percent>  Speed/<-Rate/Sec->
Time      CPU Totl User   Hertz Cycles Instr Ratio
-------- --- ---- ----   ----- ------ ----- -----
20:01:00   0  0.7  0.4   4196M  30.8M 8313K 3.709
```

BC12…

VELOCITY SOFTWARE

What is the CPU Measurement Facility (Basic)

CPI: Cycles per Instruction

Report: ESAMFCA MainFrame Cache Hit Analysis
Monitor initialized: 12/10/14 at 07:44:37

		<CPU Busy>		<-------Processor------->			
		<percent>		Speed/<-Rate/Sec->			CPI
Time	CPU	Totl	User	Hertz	Cycles	Instr	Ratio
07:48:35	0	20.8	18.4	5504M	1121M	193M	5.807
	1	21.6	19.6	5504M	1161M	221M	5.264
	2	24.4	22.5	5504M	1300M	319M	4.078
	3	22.4	19.7	5504M	1248M	265M	4.711
	4	19.6	17.6	5504M	1102M	194M	5.683
	5	20.4	18.6	5504M	1144M	225M	5.087
	6	23.9	22.0	5504M	1341M	341M	3.935
	7	17.6	15.4	5504M	949M	160M	5.927
	8	18.5	16.5	5504M	1005M	194M	5.195
	9	22.5	20.6	5504M	1259M	347M	3.629
System:		212	191	5504M	10.8G	2457M	4.733

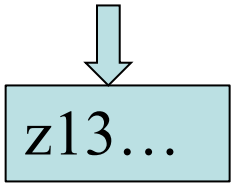
EC12…

Why you should be interested in MFC?

```
Report: ESAMFC          MainFrame Cache Analysis Rep
-------------------------------------------------------
                <CPU Busy> <-------Processor------>
                <percent>  Speed/<-Rate/Sec->
Time      CPU  Totl  User  Hertz  Cycles  Instr  Ratio
--------  ---  ----  ----  -----  ------  -----  -----
14:05:32    0  92.9  64.6  5000M   4642M  1818M  2.554
            1  92.7  64.5  5000M   4630M  1817M  2.548
            2  93.0  64.7  5000M   4646M  1827M  2.544
            3  93.1  64.9  5000M   4654M  1831M  2.541
            4  92.9  64.8  5000M   4641M  1836M  2.528
            5  92.6  64.6  5000M   4630M  1826M  2.536
               ----  ----  -----  ------  -----  -----
System:         557   388  5000M   25.9G  10.2G  2.542
-------------------------------------------------------
14:06:02    0  67.7  50.9  5000M   3389M  2052M  1.652
            1  67.8  51.4  5000M   3389M  2111M  1.605
            2  69.0  52.4  5000M   3450M  2150M  1.605
            3  67.2  50.6  5000M   3359M  2018M  1.664
            4  60.8  44.5  5000M   3042M  1625M  1.872
            5  70.1  53.8  5000M   3506M  2325M  1.508
               ----  ----  -----  ------  -----  -----
System:         403   304  5000M   18.8G  11.4G  1.640
```

1830 mip cpus (at 100%)

2828 Mip cpus (at 100%) Doing 10% more work



VELOCITY SOFTWARE

Why you should be interested – what is a MIP?

```
Report: ESAMFC          MainFrame Cache Analysis Rep
-------------------------------------------------
                <CPU Busy> <-------Processor------>
                <percent>  Speed/<-Rate/Sec->
Time       CPU Totl User   Hertz Cycles Instr Ratio
-------- --- ---- ----   ----- ------ ----- -----
14:05:32    0 92.9 64.6   5000M  4642M 1818M 2.554
---- ----   ----- ------ ----- -----
System:        557  388   5000M  25.9G 10.2G 2.542
-------------------------------------------------
14:06:02    0 67.7 50.9   5000M  3389M 2052M 1.652
---- ----   ----- ------ ----- -----
System:        403  304   5000M  18.8G 11.4G 1.640
```

1830 mips (at 100%)

2828 Mips (at 100%)

Workload changes,

- Cache benefit is better
- Cycles per instruction then improves
- Doing 10% more work at 20% lower utilization

Capacity benefits of SMT depend on workloads

- Cache benefit drops if high dispatch rate
- Address translation time increases as cache benefit drops

VELOCITY SOFTWARE

Understanding Hardware Metrics (ESAMFCA)

Reported in "per 100 instructions"

- 0.7% miss is 7 misses per 1,000 instructions
- .02 MEM equates to one memory ref per 5,000 instructions
- Example (z15) is low utilization
- ESAMFCA

	--Processor------>			<----------Rate per 100 Instructions					
	<-Rate/Sec->		CPI	**L1**	<---Data source read from---->				
Time	Cycles	Instr	Ratio	**MISS**	L2	L3	L4L	L4R	MEM
09:28:00	488M	384M	1.272	**0.701**	0.590	0.074	0.008	0.000	0.029
09:29:00	485M	385M	1.260	0.701	0.594	0.074	0.007	0.000	0.027
09:30:00	537M	434M	1.238	0.686	0.584	0.070	0.007	0.000	0.024
09:31:00	524M	403M	1.301	0.726	0.608	0.076	0.010	0.000	0.032
09:32:00	535M	420M	1.273	0.720	0.570	0.114	0.009	0.000	0.027
09:33:00	618M	522M	1.184	0.671	0.584	0.061	0.006	0.000	0.020

VELOCITY SOFTWARE

Processor cache comparison

Clock speed 5,500 to 5,000 (10% slower?)

- How can z13 be faster than EC12?

Cache sizes – EC12

- L1: 64k Instruction, 96k Data
- L2: 1MB Instruction, 1MB Data (private, cpu)
- L3: 48MB (Chip, shared 6 CPUs)
- L4: 384MB (Book, shared over 20 CPUs)

Cache Sizes – z13

- L1: 96K Instruction, 128K Data
- L2: 2MB Instruction, 2MB data
- L3: 64MB (Chip, Shared over 8 CPUS)
- L4: 480MB + 224M NIC (per node)

Processor cache comparison

Cache sizes – z15

- L1: 128k Instruction, 128k Data
- L2: 4MB Instruction, 4MB Data (private, cpu)
- L3: 256MB (Chip, shared 12 CPUs)
- L4: 960MB (Book, shared over 20 CPUs)

Cache Sizes – z16

- L1: 128K Instruction, 128K Data
- L2: 32MB unified
- L3: 224MB (Chip, Shared over 8 CPUS)
- L4: 1.75GB (per drawer)

Measure of cache value:

- RNI – how far away is the data?
- (Relative Nest Intensity)

IBM RNI calculations (per John Burg, WSC)

- **Z16 RNI =**
  4.3*(0.45*L3P + 1.3*L4LP + 5.0*L4RP + 6.1*MEMP) / 100
- **Z15/15s RNI =**
  2.9 (0.45*L3P + 1.5*L4LP + 3.2*L4RP + 6.5*MEMP)/100
- **Z14/14s RNI =**
  2.4 (0.4*L3P + 1.5*L4LP + 3.2*L4rp + 7.0*MEMP)/100
- **z13 RNI =**
  2.6 (0.4*L3P + 1.6*L4LP + 3.5*L4RP + 7.5*MEMP)/100
- **zEC12 RNI =**
  2.3 (0.4*L3P + 1.2*L4LP + 2.7*L4RP + 8.2*MEMP)/100

Smaller is better, less time loading L1 cache

Higher means more opportunity for SMT?

Cycle requirement per source

Z15, Based on RNI calculations (per John Burg)

- **Level 3 = 2.9*.45 = 1.3 cycles**
- **Level 4L = 2.9 * 1.5 = 4.3 cycles**
- **Level 4R = 2.9 * 3.2 = 9 cycles**
- **Memory = 2.9 * 6.5 = 19 cycles**

A lot of cycles can be wasted,

- RNI is a measure

What happens to RNI when add 2nd thread?

- (Yes, gets larger)

VELOCITY SOFTWARE

Address Translation (DAT)

On z13, one DAT per core

- Address translation performed on every address
- Translated addresses stored in TLB (translation lookaside buffer)

MFC data provides cycles waiting for DAT

During address translation, core cycles wasted

Z14+ has 4 DATs / core

VELOCITY SOFTWARE

Hardware metric: TLB Analysis – z13

DAT Translation: 30% of the cycles for ONE thread

- Two threads on one core leaves very little for real work

```
Report: ESAMFC        MainFrame Cache Magnitudes Report  ZMAP 4.2.4
-------------------------------------------------------------------------------
             <CPU Busy> <----------> <-Translation Lookaside buffer(TLB)->
             <percent>  Speed/       <cycles/Miss><Writs/Sec>  CPU Cycles
Time      CPU Totl User  Hertz  Ratio  Instr  Data Instr  Data  Cost  Lost
-------- --- ---- ----  -----  -----  ----- ----- ----- ----- ----- -----
07:45:01   0 25.9 24.4  5000M  1.704    159   742  473K  244K 19.77  257M
           1 35.9 34.7  5000M  1.491    138   731  530K  249K 14.17  255M
           2 15.8 13.9  5000M  2.868    206   826  419K  245K 36.30  289M
           3 16.6 15.4  5000M  2.508    212   825  411K  247K 34.90  291M
          23 18.1 17.0  5000M  2.144    197   815  412K  229K 29.44  268M
          24 21.4 19.9  5000M  1.865    114   533  598K  302K 21.35  229M
          25 26.2 24.9  5000M  1.742     98   503  736K  346K 18.71  246M
          26 12.9 11.6  5000M  2.050    154   631  378K  214K 29.92  194M
          27 13.1 11.9  5000M  1.987    156   630  378K  217K 29.64  195M
             ---- ----  -----  -----   ----- ----- ----- ----- ----- ----
System:       514  476  5000M  2.257    176   724   14M 7641K 30.69 7917M
```

One Thread, 30.69%

TLB Analysis – Should SMT be Enabled?

Evaluate other data points:

- z/VM Linux workloads issue: VERY HIGH dispatch
- Why z14 should be great….
- Don't enable SMT if one thread is consuming your DAT

```
Report: ESAMFC          MainFrame Cache Magnitudes Report
-------------------------------------------------------------------
     <CPU Busy> <---- <-Translation Lookaside buffer(TLB)->
     <percent>  Speed <cycles/Miss> <Writs/Sec> CPU  Cycles
 ##  Totl User  Hertz   Instr Data  Instr Data   Cost  Lost
 --- ---- ----  -----   ----- ----- ----- ----- ----- -----
 Mem1 907  874  5504M      54   232  117M   36M 29.55 14.8G
 Mem2 1188 1140 5000M     147   364   30M   26M 23.62 14.0G
 VLB4 1703 1366 5000M     185   567   66M   46M 44.59 38.2G
 z13N  216  212 5000M     192   598 3084K 1802K 15.94 1669M
 TCPN  892  757 5000M     217   947   32M   17M 51.46 23.0G
 MTRN  947  868 5000M     265  1283   33M   17M 65.25 30.8G ←
```

VELOCITY SOFTWARE

TLB Problem, z14/z15 Advantages

z13 (z/VM) Problem:

- z/VM does NOT support large pages, needs 256 times TLB
- Linux with java/websphere has VERY high dispatch (30k/sec?)
- Address translation (DAT) required for all parts of instruction
- Some times no cycles left after address translation….

z14 / z15

- The fix
  - If one dat per core is the bottleneck, put on 4…(“Quad TLLB”)
- Wait for DAT still degrades performance…
- Z14 –SMT is better
- Z15 just as good

TLB Analysis – Should SMT be Enabled?

Z14 is better:

- (4) x DAT gives a lot of cycles back.
- 12% DAT cycles (SMT) vs. 30% z13, NO SMT….

ESAMFC MainFrame Cache Magnitudes Rate ZMAP 5.1.0

initialized: 04/08/19 at 19:00:00 on 39064/08/19 19:00:00

	<CPU Busy>		<-------Processor------>				<-Translation Lookaside buffer(TLB)->					
	<percent>		Speed/<-Rate/Sec->				<cycles/Miss>		<Writs/Sec>		CPU Cycles	
CPU	Totl	User	Hertz	Cycles	Instr	Ratio	Instr	Data	Instr	Data	Cost	Lost
0	29.5	28.0	5208M	1535M	822M	1.867	177	284	243K	364K	9.55	147M
1	26.8	25.3	5208M	1399M	748M	1.871	178	294	248K	359K	10.71	150M
2	37.3	35.1	5208M	1945M	877M	2.219	135	210	446K	818K	11.91	232M
3	36.9	34.8	5208M	1925M	914M	2.107	136	212	449K	821K	12.19	235M
4	22.6	20.9	5208M	1181M	530M	2.228	158	263	316K	445K	14.18	167M
5	23.4	21.8	5208M	1219M	590M	2.066	156	260	316K	449K	13.63	166M
6	23.9	21.5	5208M	1248M	615M	2.030	170	284	236K	364K	11.49	143M
7	26.9	25.5	5208M	1402M	730M	1.921	166	265	237K	391K	10.19	143M
8	31.2	29.5	5208M	1628M	792M	2.055	163	257	338K	507K	11.39	185M
9	32.9	31.3	5208M	1715M	878M	1.954	159	247	326K	508K	10.34	177M
10	20.9	19.4	5208M	1093M	504M	2.171	166	276	257K	391K	13.79	151M
11	23.4	22.1	5208M	1223M	658M	1.859	162	265	247K	401K	11.95	146M
12	22.3	20.5	5208M	1162M	526M	2.209	173	302	321K	443K	16.32	190M

VELOCITY SOFTWARE

Improving Cache Value

Affinity

- Dispatching virtual machine on same core uses L1/L2 cache
- Keeping work on same CHIP maintains L3 cache
- Keeping work on same drawer maintains L4 cache
- “steals” from one thread to another delayed

What impacts (pollutes) cache?

- High dispatch rate with new process loaded
- High number virtual CPUs accessing same data

Nesting Steals – Affinity working?

z13, 60 IFLs, LPAR: 14 IFLs (SMT Enabled)

Report: **ESAPLDV** Processor Local Dispatch Vector Activity

Time	CPU	<VMDBK **Steals**	Moves/sec> To Master	Dispatcher Long Paths	<-CPU Steals fr <-From Nesting Same	**NL1**	NL2
19:47:00	0	7442.9	16.5	**34163.5**	3034	4408	0
	1	5854.5	0	29842.1	2313	3542	0
	2	5363.9	0	23112.3	2466	2898	0
	25	5900.3	0	25649.6	847	5053	0
	26	7022.2	0	28863.4	1035	5987	0
	27	5907.7	0	25927.4	799	5109	0
		--	------	-------	----	----	----
System:		161948	16.5	757754.4	67K	95K	0

Steals: vmdblks moved to different processor (20% of time)

Dispatcher Long paths:

- vmdblks dispatched (30K/Sec/CPU)
- NL1: Different chip (L3) (check affinity)
- NL2: Different book (L4) No NL2, smaller lpars better?

VELOCITY SOFTWARE

Nesting Steals – Affinity working?

z14, LPAR: 14 IFLs / 28 threads (SMT Enabled)

Report: ESAPLDV Processor Local Dispatch Vector Activity

	<-CPU Steals from Other CPUs->								
		<VMDBK Dispatcher		<-From Nesting Levels (/sec)->					
Time	CPU	Steals	Long Paths	Same	NL1	**NL2**	NL3	NL4	NL5
21:25:00	0	924.1	8387.1	632	42.9	249	0	0	0
	1	855.0	7697.4	597	39.3	219	0	0	0
	2	1047.0	7640.9	781	45.6	220	0	0	0
	3	931.6	7081.6	691	41.2	199	0	0	0
	4	935.7	7552.7	682	40.0	213	0	0	0
	5	851.6	7651.8	619	37.8	195	0	0	0
System:		21363	139331.4	9917	1292	10K	0	0	0

- NL2: Different cluster / book (L4) LPAR too big?

Z14 – Why does NL2 happen?

- 10 cores / chip
- 2-3 chips per cluster
- 2 clusters per drawer
- 4 drawers max
- CPU Order just bad luck (ESALPAR)?

<---------CPU------->			
Type	**Count**	**Ded**	**shared**
CP	**11**	**0**	**11**
IFL	**37**	**6**	**31**
ICF	**4**	**3**	**1**
ZIIP	**4**	**0**	**4**

CPU	Percent	Type
1	1.037	CP
2	0.884	CP
3	1.163	CP
4	0.838	CP
5	1.870	CP
6	0.463	IFL
8	0.459	IFL
9	1.678	CP
11	2.314	ZII
12	0.398	ZII
13	0.859	ZII
14	0.844	ZII
18	0.983	IFL
24	0.795	IFL
26	0.441	IFL
27	0.722	IFL

CPU	Percent	Type
28	0.384	IFL
29	1.102	CP
32	0.416	IFL
33	2.155	CP
34	2.169	CP
38	0.234	IFL
40	1.337	IFL
41	0.777	IFL
42	0.551	IFL
43	0.793	IFL
45	0.765	IFL
46	0.850	IFL
48	0.002	IFL
49	0.002	IFL
50	0.002	IFL
51	0.002	IFL

CPU	Percent	Type
52	0.003	IFL
53	0.002	IFL
54	0.675	IFL
55	0.605	IFL
57	0.855	IFL
58	0.683	IFL
59	0.702	IFL
60	0.743	IFL
61	0.770	IFL
62	1.200	IFL
63	1.495	IFL
65	1.115	CP
66	1.644	IFL
67	1.572	IFL
68	1.586	IFL
74	0.235	IFL
75	1.727	CP

VELOCITY SOFTWARE

Impact of NL2

Moving virtual machine to different nesting level

LPAR IFLs are across two drawers

IFLs on wrong drawer: Of Drawer memory reads

		<CPU Busy>		<-L4 Cache->		<Sourced from Memory / Sec>			
		<percent>				On	On	Off	**Off**
Time	CPU	Totl	User	OnBook	**OffBk**	Chip	Book	Book	**Drawer**
21:25:02	0	88.4	74.5	535093	72532	197648	398002	659588	0
	1	88.6	76.9	548073	71167	187255	386960	590812	0
…									
	7	89.0	77.7	561408	70493	202484	417522	637009	0
	8	89.9	77.7	594979	76359	208859	427118	650971	0
	9	89.0	77.9	591668	74152	207873	423139	647480	0
	10	12.9	11.8	94593	39694	14984	15844	32611	**137067**
	11	12.7	11.8	96285	38869	16900	17382	36391	133331
	12	13.7	12.7	87874	58750	13900	14098	29191	135545

VELOCITY SOFTWARE

SMT Topics

- **SMT Theory**
- **Data Validation, Capture ratios**
- **Capacity Planning – what does SMT add?**
- **Chargeback – What are metrics?**

SMT is about using unused cycles

If one thread

- Cycles wasted waiting for L1/L2 cache update
- Cycles wasted waiting for DAT (address translation)

If two threads

- Wasted cycles could be used by alternate thread
- **If contention for cache / dat, work takes longer**
- Is there an increase in capacity?
- What is performance impact?

SMT Theory

SMT Objective:

- Increase capacity at cost of performance (response time)
- Better core utilization (more cycles for real work)

In theory: Processor cycles are sitting idle

- To execute instruction, L1 cache is populated (data, instruction)
- Cycles wasted while L1 cache loaded from L2,L3,L4,Memory
- SMT uses “wasted” cycles for another “thread”

In practice

- Two threads share one core – and cache
- More processes share core – and cache
- Cache has more contention
- Core has contention
- But **wasted cycles are now being used**

Cycle requirement per source

What happens to RNI when add 2nd thread?

- Average CPI went from 1.25 to 1.40
- Average RNI went from .55 to .66 (cache contention)

Report: **ESAMFCA** MainFrame Cache Magnitudes R

Time	CPU	<CPU Busy> <percent> Totl	User	<-------Processor------> Speed/<-Rate/Sec-> Hertz	Cycles	Instr	**CPI Ratio**	**RNI From Burg**	
09:47:00	0	10.9	10.6	5208M	569M	454M	1.254	0.53	
09:48:00	0	11.9	11.6	5208M	621M	523M	1.187	0.42	
09:49:00	0	9.3	9.0	5208M	487M	385M	1.265	0.56	
09:50:00	0	9.5	9.2	5208M	497M	391M	1.270	0.54	
09:51:00	0	9.5	9.1	5208M	497M	380M	1.309	0.65	
09:52:00	0	10.0	9.5	5208M	520M	373M	1.394	0.62	←SMT Enabled
09:53:00	0	11.2	10.8	5208M	587M	448M	1.312	0.48	
09:54:00	0	9.8	9.3	5208M	512M	365M	1.403	0.68	
09:55:00	0	10.5	10.0	5208M	550M	390M	1.411	0.66	
09:56:00	0	10.0	9.4	5208M	521M	366M	1.422	0.75	
09:57:00	0	11.1	10.5	5208M	577M	421M	1.372	0.67	

VELOCITY SOFTWARE

CPU Measurement Facility with SMT

- Cycles by thread (total cycles used for both work, wait)
- Shows cycles used, instructions executed (thread CPI)
- Core CPI went down.
- Meaningful is Instructions per second – total (3.33G)
- Real cycles per instruction: 88% of 5208M / 3330M (1.37)

```
Report: ESAMFC          MainFrame Cache Magnitudes
Monitor initialized: 06/17/20 at 21:23:09 on 390
-------------------------------------------------
                <CPU Busy><-------Processor------>
                <percent> Speed/<-Rate/Sec->
Time      CPU Totl User Hertz Cycles Instr Ratio
-------- --- ---- ---- ----- ------ ----- -----
21:25:02   0 88.4 74.5 5208M  4607M 1652M 2.789 ->1.37
           1 88.6 76.9 5208M  4617M 1678M 2.752
```

Back to – What is a cpu second?

- We charge for CPU seconds?
- Is it consistent? No!
- How much does it vary (in instructions per second)
- Dependent on workload (cache residency)
- If more contention for cache, more time waiting

SMT Data Points

System data points / Hardware perspective

- Core time allocated to LPAR
- Thread busy / thread idle (potential capacity)
- Instructions per second per core
- Cycles per instruction (low is good)
- Impact of LPAR definition

User data points

- Core time, thread time
- Change in thread time (response time)
- Change in cycles consumed (capacity)
- Does the data agree?

SMT on z/VM has challenges

- Why is SAP / Oracle better for SMT?
- (30% ITR improvement with SMT in one Prod LPAR)
- Why would z/OS do better with SMT?

Dispatching 30,000 times per second on one thread

- How long is task on CPU? (< 30 microseconds)
- (30 microseconds -> 15,000 cycles, 5k instructions?)
- How long does data remain in L1/L2 cache?
- The more references further out, the worse things get

Relative Nest Intensity – RNI (John Burg, WSC)

- Provides relative wait times
- Smaller means less time waiting for cache to be loaded

SMT – When to use it?

SMT Announced on z13 without much guidance

Some installations said "good stuff"

- **Oracle, SAP workloads**

Others said "not so good…."

- **Java, Websphere workloads**

The question is why?

And why is z14 (and z15) so much better?

Does SMT provide more capacity?

Which approach is designed for the higher volume of traffic? Which road is faster?

**Illustrative numbers only*

Measurement:

- "person miles"?
- Per minute?

Add lanes and?

Does SMT provide contention?

Which approach is designed for the higher volume of traffic? Which road is faster?

**Illustrative numbers only*

Not always more….

VELOCITY SOFTWARE

Capture Ratios for chargeback

If multiple data sources for same "thing":

- Should they agree?
- If they don't, who is right?

Metrics that agree (single thread):

- LPAR Assigned time (source HMC/SYTCUP)
- z/VM CPU utilization (source z/VM SYTPRP)
- User data (virtual machine data) plus system overhead
- Linux system metrics via snmp (vsi mib)
- Linux process metrics via snmp (vsi mib)

Objective is to know where the resources go

- Can you capture 100%?
- How much fudge factor?
- Which metrics are impacted by SMT???

Capture Ratios – LPAR / HMC (ESALPAR)

		<----Logical Processor--->		
VCPU Addr	CPU Type	<---%Assigned--> Total	Ovhd	Emul
zVM 0	IFL	15.7	0.5	15.2
1	IFL	18.8	0.5	18.3
2	IFL	20.7	0.4	20.3
3	IFL	25.1	0.4	24.7
4	IFL	27.2	0.4	26.8
5	IFL	38.4	0.4	38.0
6	IFL	64.8	0.6	64.3
7	IFL	1.1	0.2	0.9
8	IFL	0.8	0.0	0.7
Total	IFL	212.6	3.3	209.3

Physical CPU Management time:

CPU	Percent	Type
140	0.468	IFL
141	0.623	IFL
142	0.606	IFL
143	0.506	IFL
144	0.488	IFL
145	0.449	IFL
146	0.323	IFL
148	0.632	IFL
149	0.263	IFL
150	0.909	IFL
151	0.968	IFL
152	0.940	IFL

Start at CEC level with 100%

LPAR provides (SYTCUP monitor record) for each VCPU

- System (Physical) overhead – not assigned (SYTCUG)
- LPAR (Logical) overhead – assigned to LPARs
- Emulation time – Time LPARs operate (209.3)

VELOCITY SOFTWARE

Capture Ratios – z/VM (NON SMT)

Report: **ESACPUU** CPU Utilization Report

CPU	CPU Type	Total util	Emul time	User ovrhd	Sys ovrhd	Idle time	Steal time
0	IFL	14.9	12.0	1.3	1.6	84.3	0.7
1	IFL	17.9	16.0	1.5	0.5	81.3	0.8
2	IFL	20.0	18.1	1.4	0.5	79.3	0.6
3	IFL	24.4	22.5	1.5	0.4	75.0	0.6
4	IFL	26.5	24.6	1.4	0.5	72.9	0.6
5	IFL	37.7	35.5	1.7	0.6	61.7	0.6
6	IFL	64.0	60.4	2.8	0.8	35.2	0.8
7	IFL	0.7	0.1	0.1	0.5	99.0	0.3
8	IFL	0.7	0.6	0.0	0.1	99.2	0.1
		206.9	189.8	11.6	5.4	688.0	5.1

Report: **ESAUSP2** User data

UserID /Class	<---CPU time--> <(Percent)> Total	Virt	T:V Rat
11:06:00	201.4	**189.8**	1.1
Servers	0.06	0.02	2.6
ZVPS	1.32	1.27	1.0
Linux	199.6	188.2	1.1
IBMStuf	0.17	0.13	1.3
TheUsers	0.23	0.16	1.5

z/VM provides capture ratio of 100.0%

- System overhead – not assigned to users
- User overhead – assigned to users
- Emulation time – user work

User data (ESAUSP2) from USEACT / USELOF

VELOCITY SOFTWARE

Capture Ratios – System

CEC (900%)

- Physical Management time (5%)
- IDLE Time (about… 680%)
- LPAR Assigned Time (212.6% in example)
  - LPAR Management (3.3%)
  - Emulation Time (209.3% in example)

z/VM Time

- Uncaptured (209.3 – 206.9 = 2.4%)
- System Overhead time (5.4%)
- User Overhead (11.6%)
- User Emulation Time (189.8%)
- 189% attributed to users -> capture ratio 100%

Capture Ratios – z/VM – NO SMT

ESACAPT Logical Partition Analysis

<----Logical Processor--->					<---CPU (percentages---->				Capture%
VCPU Addr	CPU Type	<---%Assigned--> Total	Ovhd	Emul	Total util	Emul time	User ovrhd	Sys ovrhd	LPAR
0	IFL	15.7	0.5	15.2	14.9	12.0	1.3	1.6	0.98
1	IFL	18.8	0.5	18.3	17.9	16.0	1.5	0.5	0.98
2	IFL	20.7	0.4	20.3	20.0	18.1	1.4	0.5	0.98
3	IFL	25.1	0.4	24.7	24.4	22.5	1.5	0.4	0.99
4	IFL	27.2	0.4	26.8	26.5	24.6	1.4	0.5	0.99
5	IFL	38.4	0.4	38.0	37.7	35.5	1.7	0.6	0.99
6	IFL	64.8	0.6	64.3	64.0	60.4	2.8	0.8	1.00
7	IFL	1.1	0.2	0.9	0.7	0.1	0.1	0.5	0.76
8	IFL	0.8	0.0	0.7	0.7	0.6	0.0	0.1	0.95
Total	IFL	212.6	3.3	209.3	206.9	189.8	11.6	5.4 6	0.99

Compare LPAR (SYTCUP) to z/VM (SYTPRP): Capture 99%

- CPU by CPU comparison accurate
- Some scheduling time likely lost

VELOCITY SOFTWARE

Capture Ratios – Linux

Report: ESALNXV LINUX Virtual Processor Analysis Report

Node/ Name	VM ServerID	Node GroupID	<Linux Pct CPU> Total	Syst	User	<Process Data> Total	Syst	User	Capture Ratio
09:28:00									
lxbmq001	LXQMQ001	TheUsers	0.6	0.2	0.3	0.5	0.2	0.3	0.912
lxbpc001	LXQPC001	TheUsers	2.2	0.9	1.3	2.2	1.0	1.3	1.011
lxbsb001	LXQSB001	TheUsers	2.3	0.7	1.6	2.3	0.7	1.6	0.993
lxvmb101	PXVMQ101	TheUsers	1.6	0.5	1.1	1.8	0.6	1.2	1.082
lxvmb102	PXVMQ102	TheUsers	1.7	0.5	1.2	1.6	0.5	1.0	0.922

Capture ratio concept for Linux process table

- Add all the processes, compare to system
- Much more difficult problem than z/VM

Charge back model is NOT 100%

Data requires "fudge factor"

- PRSM Overhead: 1% ?
- LPAR Overhead: 3%?
- LPAR Capture ratio: 1% (capture ratio 99%)
- z/VM System overhead
- z/VM virtual machine overhead
- Virtual machine real work – this is what we charge for

What does SMT do?

Processor utilization – NO SMT

- Numbers agree and make sense
- Can capture virtual machine resources and believe it
- Have value for overheads

SMT Challenges

- Virtual machines share the CPU / core
- The more they share, the slower they go (how slow?)
- Numbers likely not repeatable based on workload
- How much added capacity with SMT for YOUR workload?
- How do you charge?
- (You MUST charge for consumption)

*Processor **Capacity Planning** Concepts*

Processor utilization – **what level is target**?

- Performance –what level of performance required?
- What level of performance management required? Available?
- Capacity Planning – what utilization level is needed financially?

Customer targets

- Target based on performance?
- 80+% hardware utilization plus requires management
- 50% CPU minimizes CPU queue – better performance tradeoff
- Higher utilization is better financially

Capacity planning objective:

- Provide resources to get work done in timely fashion
- Meeting appropriate financial and performance objectives

Processor Measurement Concepts - Utilization

What is "CPU Utilization"? Need to agree on this first?

All zVPS numbers are measured in CPU Seconds

- Percent is always based on CPU seconds divided by wall clock
- What is a CPU second if two threads with SMT?

Impacts measurements of

- LPAR (percent of processor assigned to partition)
- z/VM Virtual Machines (percent of "thread" assigned to virtual machine)
- Linux processes (percent of vcpu)

BUT DO WE AGREE ON WHAT IS IMPORTANT?

- Is it processor utilization?
- Or work completed?

Capacity Benefits?

SMT Adds how much capacity?

- How much more throughput?
- Workload dependencies
- How to predict

Z13/14/15/16 have larger cache sizes

- How long does cache last when 30,000 dispatches / second / processor?
- How much does enabling SMT impact cache?

How much used capacity at CEC level?

- Total IFL (Assigned) Utilization (ESALPARS / ESALPMGS)
- Totals by Processor type
- Shared processor total busy

```
<---------CPU-------> <-Shared Processor busy->
Type Count Ded shared  Total  Logical Ovhd Mgmt
---- ----- --- ------ ------ -------- ---- ----
CP      11   0     11  892.1    865.2 11.2 15.7
IFL     37   6     31 2466.7   2412.0 30.9 23.8  ← 80% utilization
```

z/VM: One core, Two threads

- "assigned" 933.7% - 4.1%
- Two threads not always both active -> thread idle time
- Subtract 138% thread idle (not really excess capacity)
- -> (933%-4)*2 – 138% = **1720**% Thread time (z/VM time)

			<-------Logical Partition------->						
			Virt	CPU	<%Assigned>		<-Thread->		
Time	Name	Nbr	CPUs	Type	Total	Ovhd	Idle	cnt	
21:25:00	Totals:	00	27	CP	876.3	11.2			
	Totals:	00	54	IFL	2443	30.9			
	ZVMQAXX	0B	14	IFL	933.7	4.1	138.1	2	←

How much used capacity z/VM LPAR?

- Total IFL Utilization (ESACPUU) 1,709%, (capture 99%+)
- User billable Traditional: (1499+36) 1,535% ??

Report: ESACPUU CPU Utilization Report

	<----Load---->					<--------CPU (percentages)-------->					
	<-Users->		Tran		CPU	Total	Emul	User	Sys	Idle	Steal
Time	Actv	In Q	/sec	CPU	Type	util	time	ovrhd	ovrhd	time	time
21:25:00	194	399	0.5	0	IFL	88.4	74.5	1.7	12.2	10.5	1.1
				1	IFL	88.6	76.9	1.9	9.8	10.3	1.1
				2	IFL	89.2	77.7	2.4	9.1	9.7	1.1
				3	IFL	89.2	77.7	1.5	10.1	9.6	1.2
				4	IFL	89.6	78.0	1.7	9.9	9.4	1.0
				5	IFL	89.1	77.7	2.3	9.1	9.9	1.1
				22	IFL	67.2	58.5	1.5	7.1	11.6	21.2
				23	IFL	66.8	58.4	1.4	6.9	12.0	21.3
				24	IFL	74.9	66.4	1.5	7.0	13.9	11.1
				25	IFL	74.4	66.3	1.6	6.5	14.4	11.2
				26	IFL	76.4	68.3	1.3	6.8	12.6	10.9
				27	IFL	75.6	68.2	1.6	5.8	13.4	11.0
System:						1709	1499	36.2	173.6	332.2	759.1

VELOCITY SOFTWARE

Processor Measurements SMT

```
Report: ESAUSR5      User SMT CPU Consumption Analysis
-------------------------------------------------------
         <----Raw CPU Seconds Consumed (Total)---->
UserID   <Traditional> <MT-Equivalent> <MT Prorated>
/Class   Total  Virt   Total  Virtual  Total Virtual
-------- ----- -----   -----  -------  ----- -------
10:32:00 660.4  641.7  476.0    462.5  432.0   420.0
 ***User Class Analysis***
TheUsers 660.2  641.6  475.9    462.4  431.9   419.9
 ***CPU POOL User Analysis***
DB2      15.63  15.42  12.13    11.97  12.23   12.09
EEMSCSP   9.03   8.97   6.91     6.87   6.59    6.55
IIB      498.7  488.6  360.4    353.2  321.8   315.4
```

ESAUSR5/ESAUSP5 show SMT user data (raw, percents)

Three CPU measures

- Traditional: Time assigned and dispatched on a thread
- MT-Equivalent: Time Would take if non-SMT (Performance)
- MT Prorated: Cycles really used (Capacity, Chargeback) (Estimated)

What if some workloads perform better non-smt?

- Should you have a "performance LPAR"?
- SMT ALWAYS degrades single task response time

VELOCITY SOFTWARE

SMT Not Always a good thing?

```
Report: ESAUSP5       User SMT CPU Consumption Analysi
Monitor initialized: 06/17/20 at 21:23:09 on 3906 ser
-------------------------------------------------------
          <------CPU Percent Consumed    (Total)---->
UserID    <Traditional> <MT-Equivalent> <MT Prorated>
/Class    Total  Virt    Total  Virtual  Total Virtual
--------  -----  -----   -----  -------  ----- -------
21:25:00   1535   1499    1051     1026   1192    1163
```

Workload helped by SMT? Is Monitor user data valid?

- 1535 percent “thread time” (validated against cpu busy)
- 1192 percent core time
- “would be” time 1051,
- Used 1192 percent, could have been 1051.
- Based on user data, less capacity because of SMT? (13%)

- But hardware said 933% assigned, and that data is validated
- And still there is thread idle, account for that?

VELOCITY SOFTWARE

Processor Measurements SMT Validity

```
<------CPU Percent Consumed    (Total)---->
UserID    <Traditional> <MT-Equivalent> <MT Prorated>
/Class    Total  Virt    Total  Virtual  Total Virtual
-------- ----- -----    -----  -------  ----- -------
21:25:00  1535   1499    1051     1026   1192    1163
```

ESAUSR5/ESAUSP5 show SMT user data

- Traditional: Thread time (response time)
- Equivalent: Time Would take if non-SMT
  - (PERFORMANCE ratio 1051 / 1535) – 50% slower
- Prorated: Cycles really used (approximately, prorated)
  - (Capacity, Chargeback)
  - Want to charge for 933% (physical assigned time to LPAR)
  - Prorated metrics are too high (1192 / 933)

Processor Measurements Data Valid?

ESAUSP5:

CPU percent Consumption

- **Total all user**
- **By user**
- **By Class**

```
Report: ESAUSP5       User SMT CPU Consumption Analysis
Monitor initialized: 06/17/20 at 21:23:09 on 3906 seri
-------------------------------------------------------
          <------CPU Percent Consumed    (Total)---->
UserID    <Traditional> <MT-Equivalent> <MT Prorated>
/Class    Total  Virt   Total  Virtual  Total Virtual
-------- ----- -----   -----  -------  ----- -------
21:25:00  1535   1499   1051     1026   1192     1163
 ***User Class Analysis***
Servers    0.04   0.00   0.03     0.00   0.03     0.00
ZVPS       2.38   1.57   1.66     1.04   2.14     1.37
TheUsers   1532   1497   1049     1025   1189     1162
```

LPAR Assigned Time: 933.7 %

z/VM Thread assigned time: 1720 %

User time: (1499+36=1535)

- Traditional measurements valid,
- 100% capture ratio
- IBM SMT prorated numbers 30% off?
- Watch for "Velocity Prorates in next"

```
                <Processor>  Captur
                Utilization   Ratio
Time            Total Virt.  (pct)
--------        ----- -----  ------
21:25:00         1709  1499  100.00
21:26:00         1642  1438  100.00
21:27:00         1641  1381  100.01
21:28:00         1639  1329   99.99
21:29:00         1561  1332  100.00
21:30:00         1528  1305   99.99
**********************
Average:         1629  1389  100.00
```

VELOCITY SOFTWARE

Processor Measurements SMT Validity

ESALPARS	ASSIGNED		ESAUSP5	THREAD		MT-PRORATED	
ZVMQA00	933.7	4.1	21:25:00	1535	1499	1192	1163
ZVMQA00	897.6	4.2	21:26:00	1477	1438	1146	1116
ZVMQA00	908.8	5.6	21:27:00	1431	1381	1115	1076
ZVMQA00	905.1	5.9	21:28:00	1382	1329	1077	1035
ZVMQA00	883.2	7.4	21:29:00	1379	1332	1081	1044
ZVMQA00	873.5	8.2	21:30:00	1350	1305	1061	1025
ZVMQA00	894.9	7.0	21:31:00	1445	1402	1129	1095
ZVMQA00	915.2	4.8	21:32:00	1469	1427	1141	1107
ZVMQA00	901.2	5.3	21:33:00	1413	1364	1097	1058
ZVMQA00	917.3	6.2	21:34:00	1452	1405	1134	1097
ZVMQA00	906.4	6.2	21:35:00	1430	1383	1117	1080
ZVMQA00	923.1	6.5	21:36:00	1454	1406	1137	1099

Compare assigned time to thread time to "prorated"

- **Target is assigned time, maybe subtract thread idle**
- **The Velocity Prorated will be in next release**

SMT Prorate minute by minute

Compute ESALPARS Assigned, subtract thread idle

Prorate against ESAUSP5 total, get "new" prorate interval by interval (.56 - .59)

```
ratio:  0.563289902
ratio:  0.562660799
ratio:  0.583018868
ratio:  0.603183792
ratio:  0.577411168
ratio:  0.578814815
ratio:  0.560761246
ratio:  0.578012253
ratio:  0.591224345
ratio:  0.575172176
ratio:  0.576713287
ratio:  0.576925722
```

Chargeback for SMT Step 1

Start with ESALPARS:

- Assigned time to LPAR (1900, Dedicated)
- Thread idle
- Time to be charged: (1900 * 2 – 1471) / 2 = 1164%
- Thread time for comparison: 1164% * 2 = 2329%

```
Report: ESALPARS
Monitor 12/12/22
-------------------------------------------------------
          <------ical Partition------->
                    Virt CPU  <%Assigned>  <-Thread->
Time      Name      CPUs Type Total  Ovhd  Idle   cnt
--------  ------- ---- ---- -----  ----  ------ ---
22:02:00  Totals:   16  CP  126.4   1.4
          Totals:   64 IFL  282.2   4.0
          VMP1O3    19 IFL   1901   0.1   1471    2
```

Chargeback for SMT Step 2

Compare to ESACPUU to validate capture ratio

- LPAR measurement thread time: 2329%
- Time to be charged: 1164%
- CPU (thread time): 2296% **(98.5% capture)**
- User thread time: 2207+39 = 2246%
- Time to be charged from ESALPARS: 1164%

Report: ESACPUU

		<--------CPU (percentages			
		Total	Emul	User	Sys
Time	CPU	util	time	ovrhd	ovrhd
22:02:00	0	58.1	55.9	0.9	1.3
	1	63.8	61.8	0.9	1.1
	37	57.1	54.8	1.1	1.2
System:		2296	2207	39.0	50.4

Chargeback for SMT Step 3

Calculate prorate factor: ESAUSP5

- Thread time: 2246% (100% capture ratio within z/VM)
- IBM "prorated time" 1623% is incorrect
- Time to be charged from ESALPARS: 1164%
- Prorate using "traditional" 1164 / 2246 = .51
- Charge back factor .51 against traditional times (.51-.53)

Report: ESAUSP5 User SMT CPU Consumption Analysis

	<------CPU Percent Consumed				(Total)---->	
UserID	<Traditional>		<MT-Equivalent>		<MT Prorated>	
/Class	Total	Virt	Total	Virtual	Total	Virtual
22:02:00	2246	2207	1643	1614	1623	1597
Top User Analysis						
xxxDBLP5	436.1	427.9	320.3	314.2	316.4	310.9
xxQDBLP1	350.3	346.5	256.7	254.0	266.4	263.6
xxDDBLP1	314.8	309.0	224.2	220.1	203.8	200.1
xxQDBLP3	282.8	280.1	213.8	211.7	235.8	233.6
xxDDBLP3	248.7	244.2	182.8	179.5	186.9	183.6

Expectations of SMT?

IBM Monitor data "MT Prorated" is incorrect

IBM Monitor data "MT-Equivalent" not validated

Need validated prorate factor

Low utilization:

- **Capacity not really an issue**
- **Response time should not change**

High utilization – Intense workloads (SAP, Oracle)

- **Capacity should see improvements**
- **Cache utilized well (dedicate engines….)**

High Utilization – polling workload (was, db2)

- **Cache competition very very high**
- **Response time WILL get worse**
- **Capacity may drop? – validate with MFC….**

VELOCITY SOFTWARE

SMT Capacity Planning

- Your capacity improvements are “dependent”
- Enhancements to capacity measurable
- Evaluate each LPAR for SMT value (CPI)
- Evaluate each server for SMT impact

SMT Chargeback

- IBM provides bogus metrics at user chargeback
- Likely results in overcharging
- Develop an added prorate metric