Morten Buur Rasmussen
Lab Services Europe

IBM

IBM i Virtualization Performance

© 2017 IBM Corporation

Goal of presentation

Virtualization is a large area.

Virtualization for IBM i is a large area

Virtualization performance is a large area

So for the hour the goal is to give the audience a good base understanding of virtualization and performance aspects related to that.

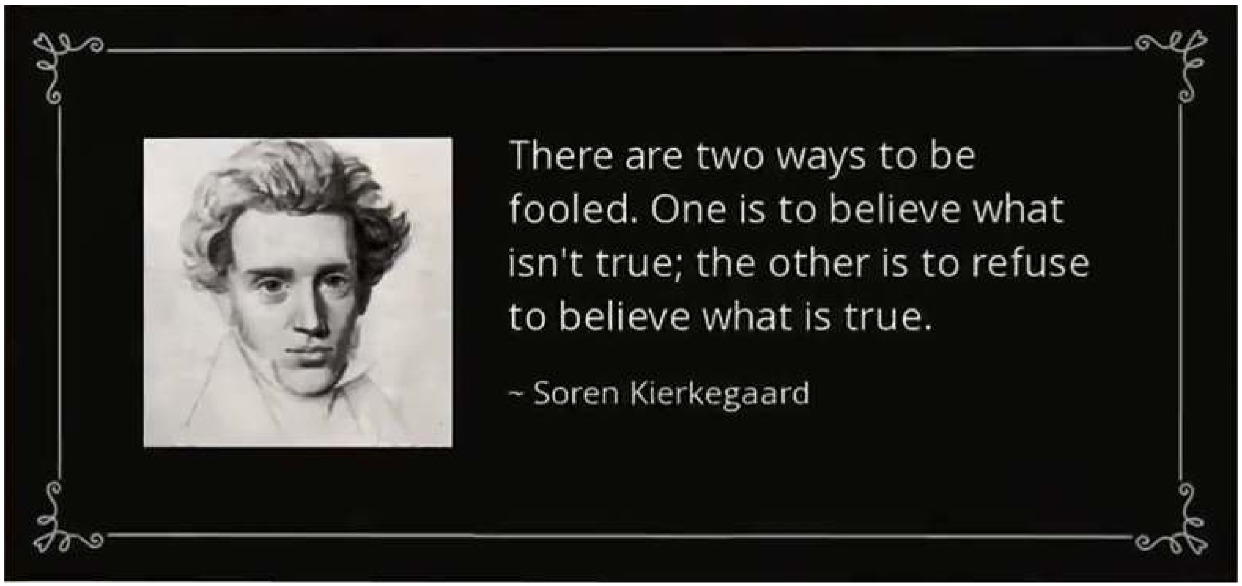
It will also include some practical examples to illustrate why its important for anybody having interest in IBM i performance.

Performance Disclaimer

“it depends …”

Performance information and recommendations in this presentation are based on measurements, analysis, and projections in different customer environments for specific performance workloads.
Your results may vary significantly and are dependent on the application and configuration.
This information is provided along with general recommendations for you to better understand system performance.
Information is provided *AS IS* without warranty of any kind.

Performance, like many other aspects of life :o)



Session objectives

- Introduction to Virtualization
- Why is virtualization performance important for IBM i?
- VIO Server
- CPU and memory affinity
- Does other LPAR's using the resources
- Virtualization tips on IBM i
- Examples of virtualization performance issues

What is Virtualization

Oxford Dictionary has Three Explanations:

1. Almost as described, but not completely or according to strict definition.

2. Computing: not physically existing as such but made by software to appear to do so.

3. Optics : relating to the points at which rays would meet if produced backwards.

To expand little further:

Logical representation of resources not restricted by physical location

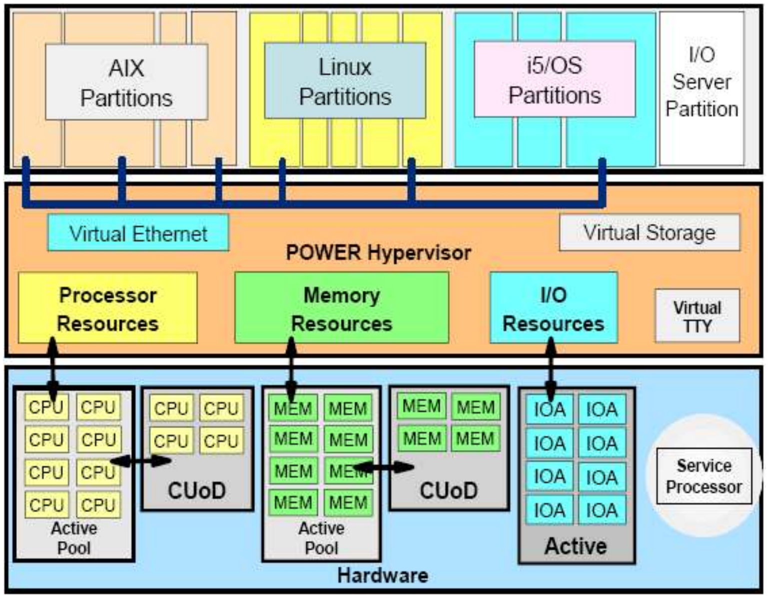
Dynamically change and adjust resources across the infrastructure

Being able to expand and share resources

Leverage resources and drive utilization up

Virtualization technologies on POWER systems

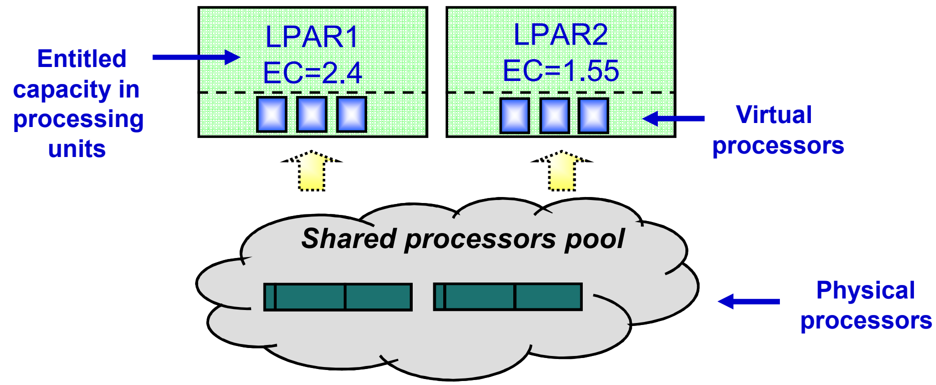
- Multiple OSs
- Multiple LPARs
- Dynamic LPAR
- Micro-partitioning
- Virtual CPUs
- Multiple shared pools
- Shared dedicated processor
- IVE
- SMT
- Virtual LANs
- Virtual I/O
- IVM
- COD
- Live Partition Mobility
- Active Memory Sharing
- Active Memory Expansion*
- Shared Storage Pools
- Suspend/Resume*
- Active Memory Deduplication*



* POWER7 systems only

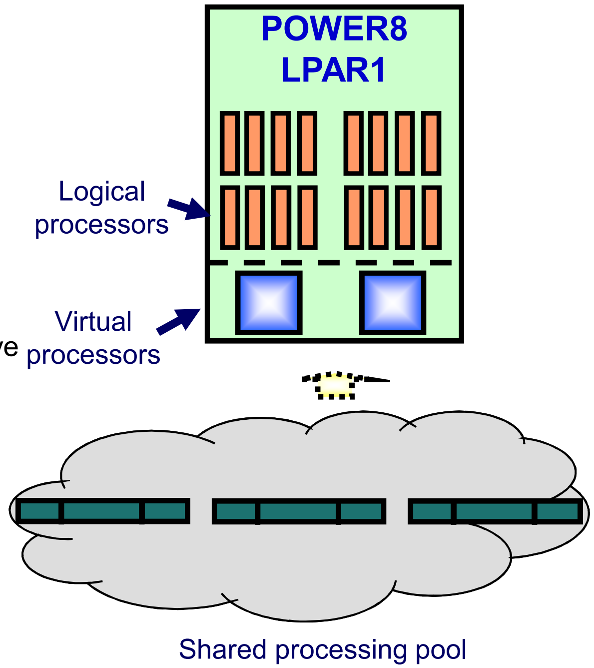
Micro-Partitioning: Shared processors

- Processor capacity is assigned in processing units from the shared processor pool:
  - Partition's guaranteed amount is its entitled capacity (EC).
  - Each partition is configured with a percentage of execution dispatch time for each 10ms timeslice (dispatch window) by default. This can be changed to 50ms dispatch window in Power8.
- Each virtual processor provides access to a single physical processor in the pool.



Simultaneous multithreading and Micro-Partitioning

- Simultaneous multithreading can be used with Micro-Partitions.
- With simultaneous multithreading, each virtual processor runs two threads (POWER6), four threads (POWER7) or eight threads (POWER8)
  - Each thread is called a logical processor.
- Even the smallest partition will always have access to a FULL CPU.
  - Minimum EC is 0.05 = 1/2ms per 10ms
- LPAR1 example:
  - 1.6 processing units
  - Two virtual processors
  - Simultaneous multithreading enabled
    - 16 logical processors



Each partition always have ONE FULL CPU

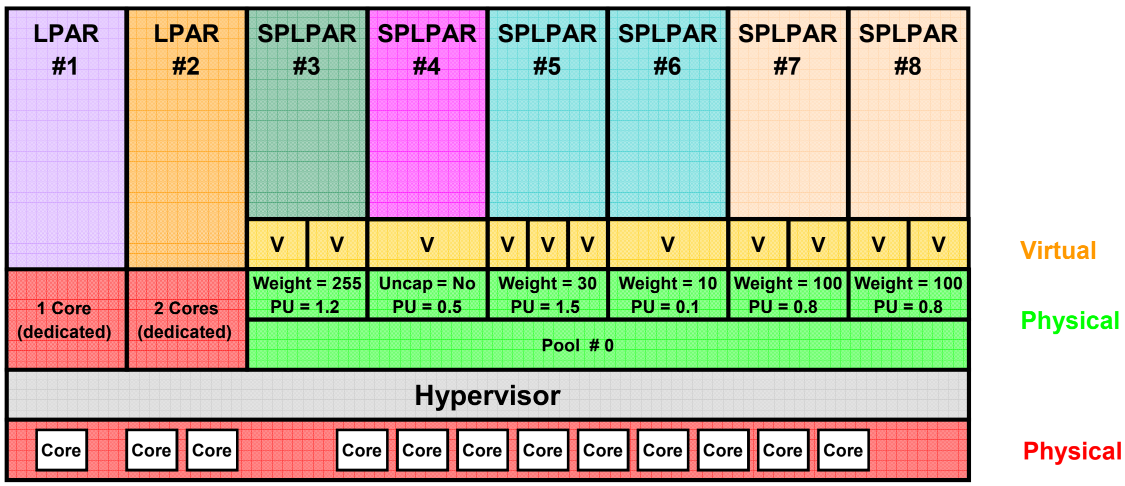
- Duration from 1/2ms to 10ms

LPAR configuration

- One LPAR can have dedicated or shared processors
  – Dedicated processors are reserved for the LPAR and can not be used by other LPAR’s, unless the settings is donating back to the Shared Processor Pool
- One LPAR can have capped or uncapped processors
  – Capped processors are limited by the EC (Entitled Capacity)
  – Uncapped processors are limited by the VP (Virtual Processors and Shared Processor Pool)

Distribution of extra processing cycles

Excess processing cycles are distributed based upon a weighting factor

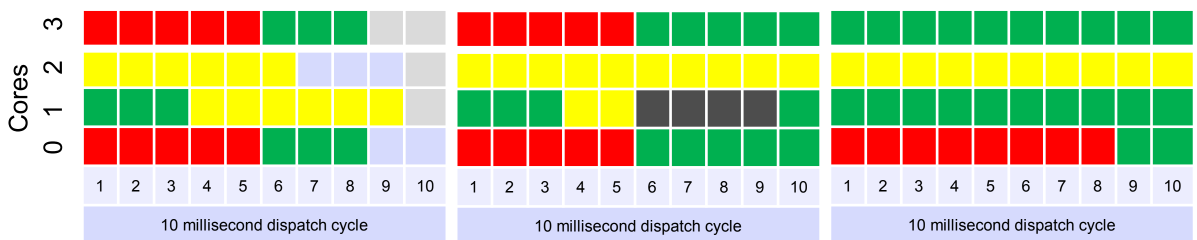


Learning points: (1) Capped LPARs are limited to their PU setting and cannot access extra cycles

(2) Uncapped LPARs have a *weight* factor which is a share based mechanism for the distribution of excess processor cycles.

Hypervisor dispatch model – Simplified version 1/2

	Virtual Processors	Processing Units	Units per VP	Uncapped?
LPAR1	2	1.0	0.5	No
LPAR2	2	1.2	0.6	Yes
LPAR3	3	0.9	0.3	Yes



Initial entitled capacity
LPAR3 uses all and requests more. LPAR1 and LPAR2 do not uses their entitlement.

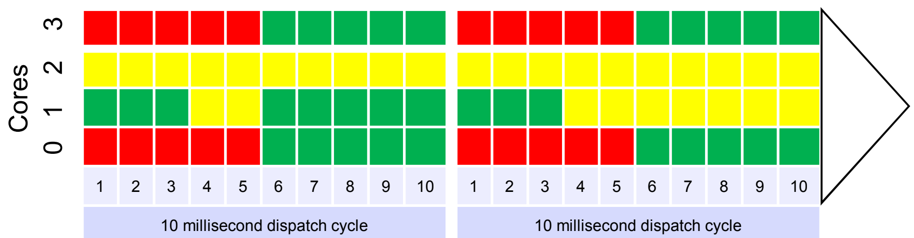
LPAR3 gets all available.
LPAR1 and LPAR2 are still not using their entitlement, so they will donate back to the
SPP (Shared Processor Pool)

LPAR3 gets all available.
LPAR1 and LPAR2 uses all cycles

Learning point: The hypervisor automatically adjusts allocations based on each shared LPAR's use of cycles during the previous dispatch cycle.

Hypervisor dispatch model – Simplified version 2/2

	Virtual Processors	Processing Units	Units per VP	Uncapped?
LPAR1	2	1.0	0.5	No
LPAR2	2	1.2	0.6	Yes
LPAR3	3	0.9	0.3	Yes



LPAR3, LPAR1 and LPAR2 uses all of their entitlement.

LPAR3, LPAR1 and LPAR2 uses all of their entitlement. The remaining cycles will be distributed to LPAR2 and LPAR3 based on weight, default128.
128/(2*128) * 0.9 = 0.4

Uncapped partition weight value simplified example

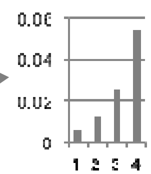
- The Weight value is used to calculate the portion of available processing capacity for partitions simultaneously requesting additional processing capacity above entitlement
  - It is also used to determine the contending partitions start order for the available capacity
  - Other factors can also impact evaluation, such as frequency of unfolding and virtual processor cap
  - ***If there are no contention, the weight value have no impact***
- Use sufficient spread between Weight values

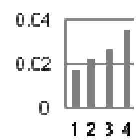
	Partition weight value distribution			
	Current		Recommended	
VIOS	255	**Diff**	**255**	**Diff**
Critical	192	25%	**~127**	**50%**
Important	128	33%	**~64**	**50%**
UAT/Dev	120	6%	**~32**	**50%**
Other	100	17%	**~16**	**50%**
Other	64	3%	**~8**	**50%**

In this simplified example, four partitions compete for one available 10ms time cycle, illustrating how the spread between weight values impact each partitions allowance.

			Available processing capacity	
	Sum of weights	465		1.0
	Weight values	Proportion	Fraction	Partition allowance
1	*30*	0.065	**6/100**	*0.06*
2	*60*	0.129	**13/100**	*0.13*
3	*120*	0.258	**26/100**	*0.26*
4	*255*	0.548	**55/100**	*0.55*
		1.0	1	1.0

			Available processing capacity	
	Sum of weights	739		1.0
	Weight values	Proportion	Fraction	Partition allowance
1	*128*	0.173	**17/100**	*0.17*
2	*164*	0.222	**22/100**	*0.22*
3	*192*	0.260	**26/100**	*0.26*
4	*255*	0.345	**35/100**	*0.35*
		1.0	1	1.0





Change processor multitasking

- Change system value QPRCMLTTSK to 0
  - Will set off the multitasking
- Change system value QPRCMLTTSK to 1 – SMT mode or 2 – System controlled
- CALL PGM(QWCCHGPR) PARM(X'00000002')
  - Will set the system to **SMT2**
- CALL PGM(QWCCHGPR) PARM(X'00000004')
  - Will set the system to **SMT4**
- CALL PGM(QWCCHGPR) PARM(X'00000008')
  - Will set the system to **SMT8**
- CALL PGM(QWCCHGPR) PARM(X'00000000')
  - Will set the system to default level (SMT4 for P8 7.1, SMT8 for P8 7.2 and 7.3)
- Retrieve current SMT value via API QWCRTVPR
  - Sample of a CL Program:
    - http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/apis/qwcrtvpr.htm

Retrieve processor multitasking information API

- Retrieve Processor Multitasking Information (QWCRTVPR) API

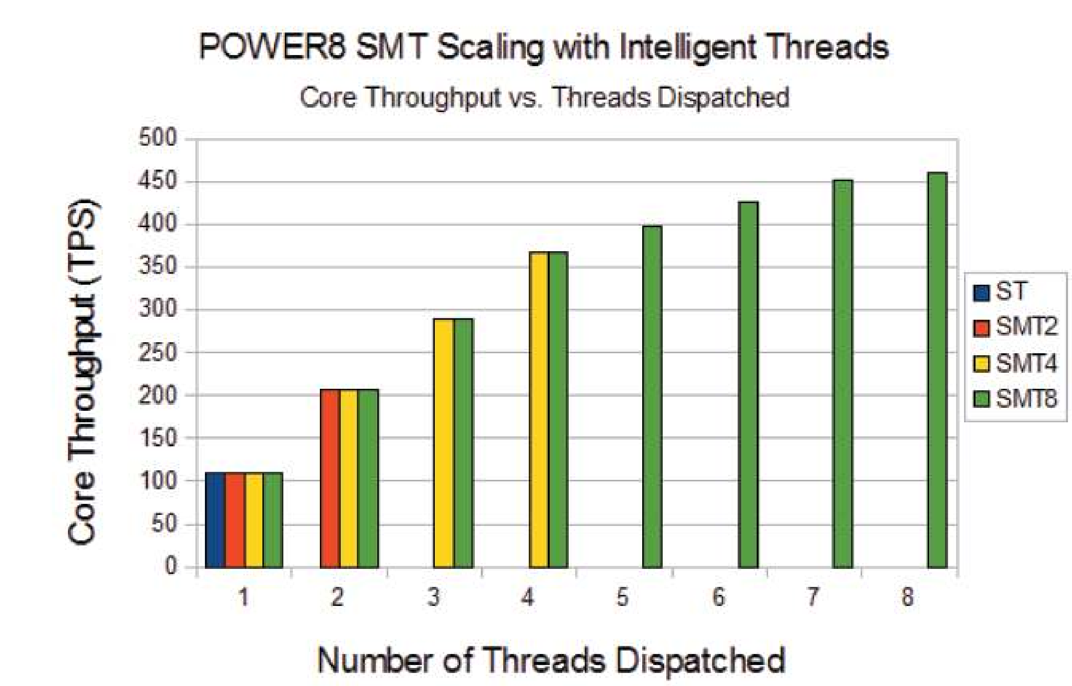
- Check the Knowledge Center for CL command to show SMT level:
  - http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_73/apis/qwcrtvpr.htm

Example

Note: By using the code examples, you agree to the terms of the Code license and disclaimer information.

```
/**********************************************************************/
/* This program calls the QWCRTVPR API to retrieve the maximum       */
/*   number of secondary threads per processor and sends a CPF9898 */
/*   message to the display with the result.                       */
/*                                                                 */
/* To compile program:                                             */
/* CRTCLPGM OBJ(QGPL/RTVPRCXMP) SRCFILE(QGPL/QCLSRC)               */
/*                                                                 */
/* To invoke program:                                              */
/* CALL QGPL/RTVPRCXMP                                             */
/**********************************************************************/
PGM
DCL VAR(&MAXTHD) TYPE(*INT) LEN(4)
DCL VAR(&MAXTHDC) TYPE(*CHAR) LEN(4)
DCL VAR(&IDX) TYPE(*INT) LEN(2)
DCL VAR(&ERR) TYPE(*CHAR) LEN(8) VALUE(X'0000000000000000')
DCL VAR(&MSGDTA) TYPE(*CHAR) LEN(52) +
  VALUE('The maximum secondary threads per processor is:     ')
```

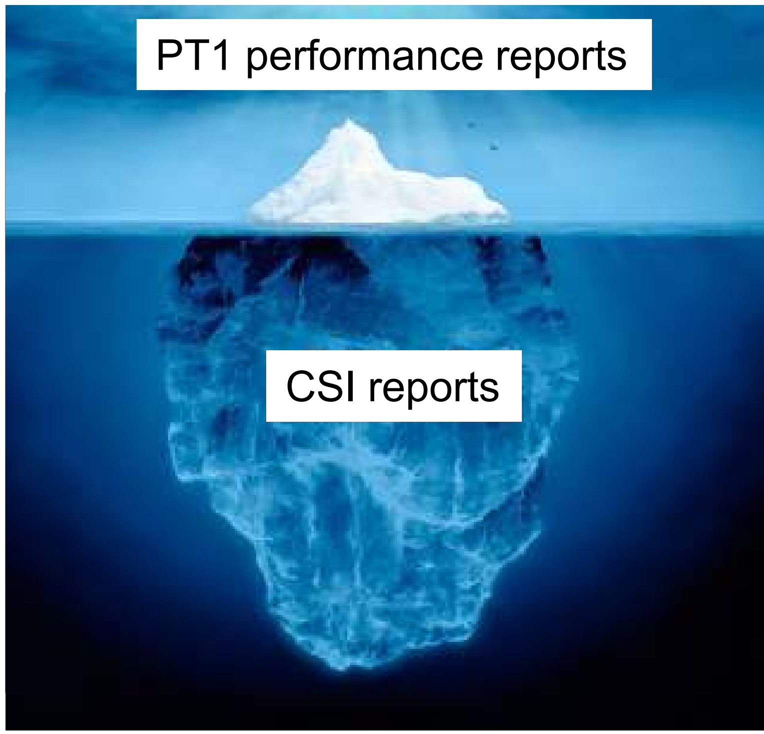
Intelligent threading - remember not related to multithreaded jobs



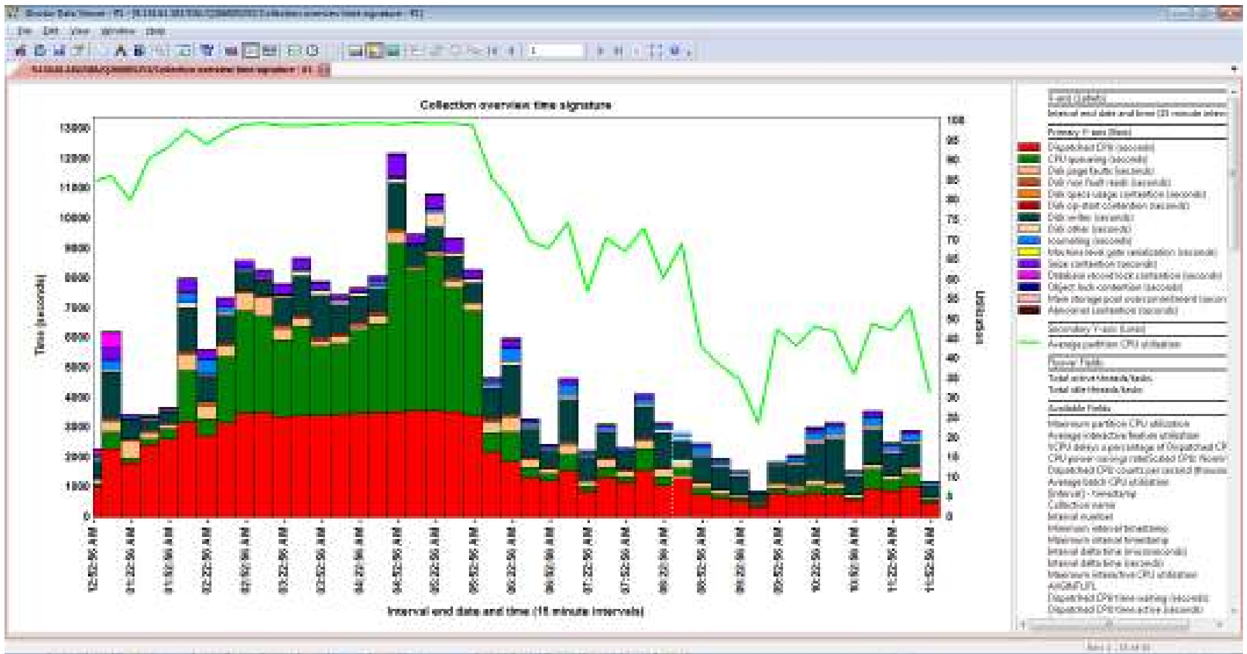
Multiple Execution Units or “Pipes” on POWER

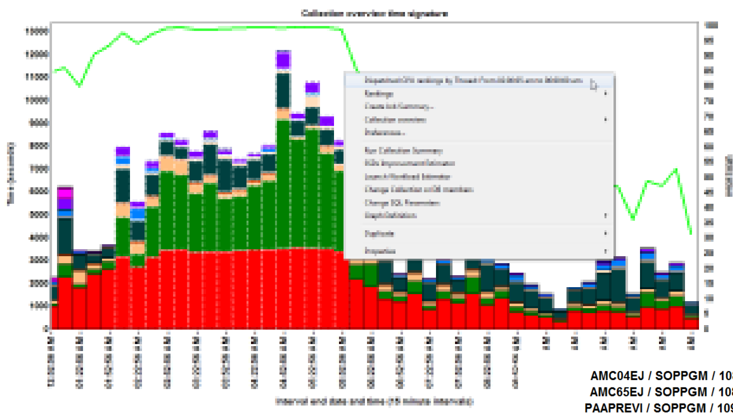
- The POWER7 processor has a set of 12 execution units
  - 2 fixed-point units
  - 2 load/store units
  - 4 double-precision floating-point units
  - 1 each of: vector unit supporting VSX, decimal floating-point unit, branch unit, condition register unit
- The POWER8 processor has a set of 16 execution units
  - 2 fixed-point units
  - 2 load/store units
  - 2 instruction fetch units
  - 4 double-precision floating-point units
  - 2 vector unit supporting VSX
  - 1 each of: Cryptographic Unit, decimal floating-point unit, condition register unit, branch register unit

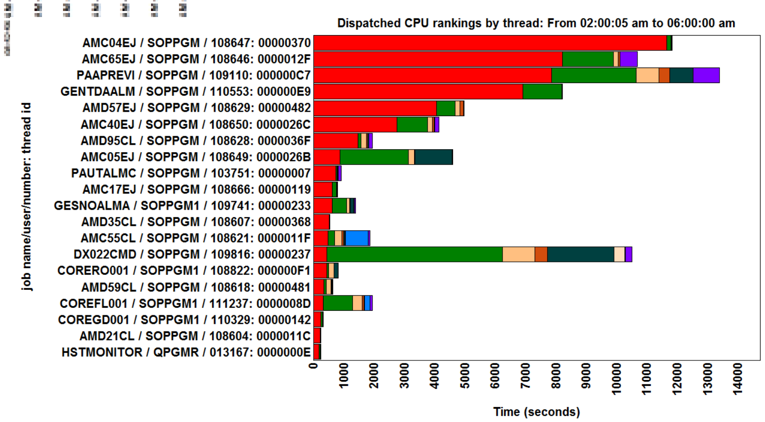
PT1 performance reports compared to Collection Service Investigator (CSI) reports/graphs (iDoctor)



Practical example with SPLPAR P8 – SMT4 – V7R1- EC 0.26/VP 1 – processor sharing on – capped Yes

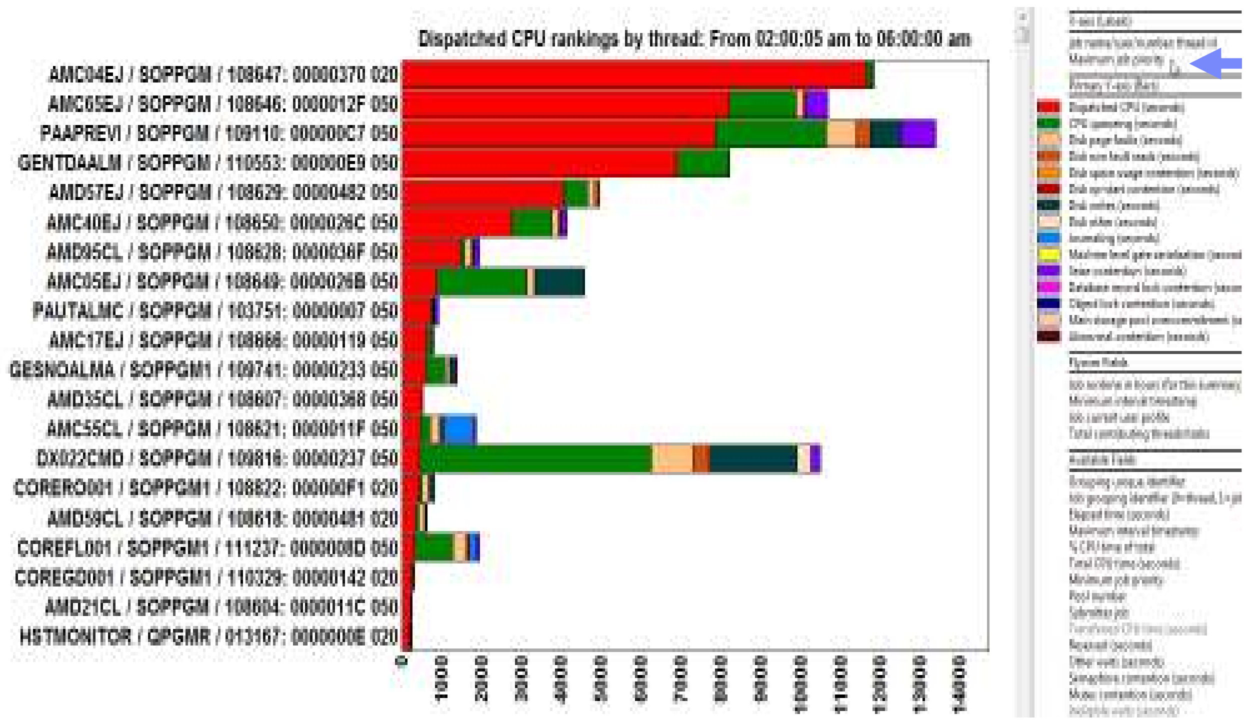




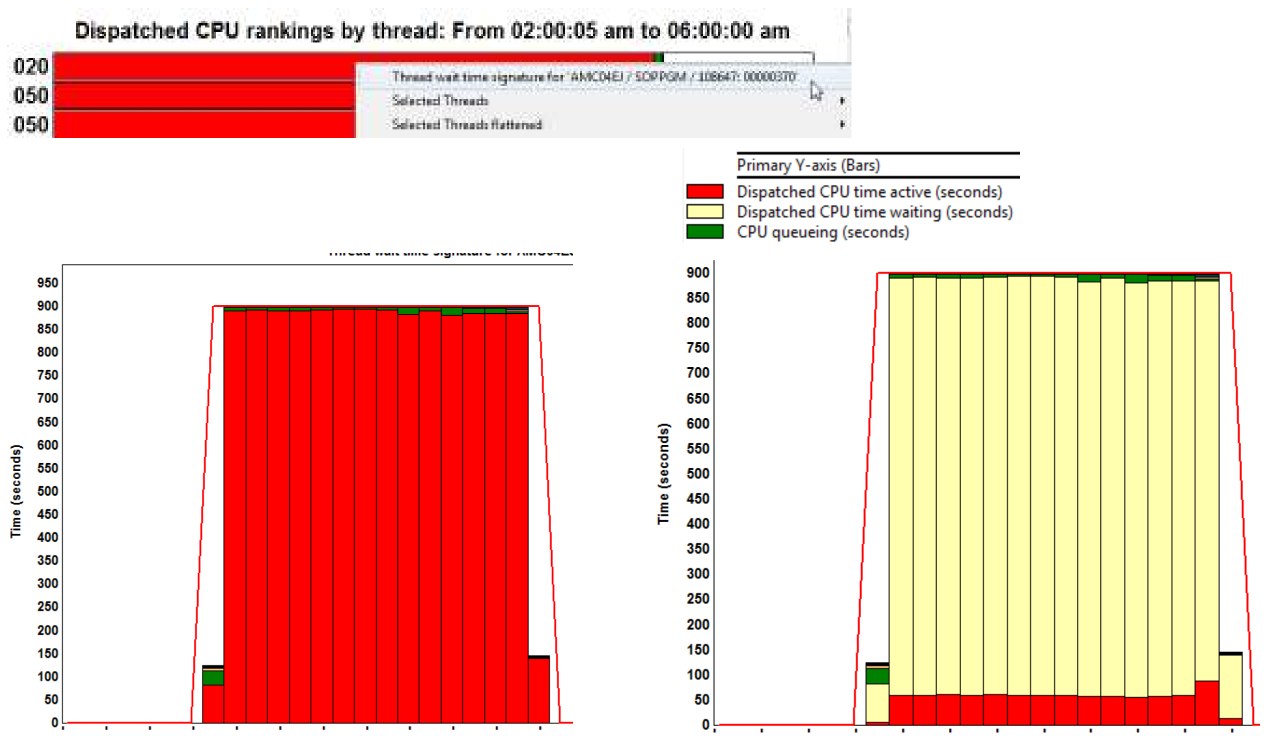


Dispatched CPU rankings by thread: From 02:00:05 am to 06:00:00 am
AMC04EJ / SOPPGM / 108647: 00000370
AMC65EJ / SOPPGM / 108646: 0000012F
PAAPREVI / SOPPGM / 109110: 000000C7
GENTDAALM / SOPPGM / 110553: 000000E9
AMD57EJ / SOPPGM / 108629: 00000482
AMC40EJ / SOPPGM / 108650: 0000026C
AMD95CL / SOPPGM / 108628: 0000036F
AMC05EJ / SOPPGM / 108649: 0000026B
PAUTALMC / SOPPGM / 103751: 00000007
AMC17EJ / SOPPGM / 108666: 00000119
GESNOALMA / SOPPGM1 / 109741: 00000233
AMD35CL / SOPPGM / 108607: 00000368
AMC55CL / SOPPGM / 108621: 0000011F
DX022CMD / SOPPGM / 109816: 00000237
CORERO001 / SOPPGM1 / 108822: 000000F1
AMD59CL / SOPPGM / 108618: 00000481
COREFL001 / SOPPGM1 / 111237: 0000008D
COREGD001 / SOPPGM1 / 110329: 00000142
AMD21CL / SOPPGM / 108604: 0000011C
HSTMONITOR / QPGMR / 013167: 0000000E
job name/user/number: thread id
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
13000
14000
Time (seconds)

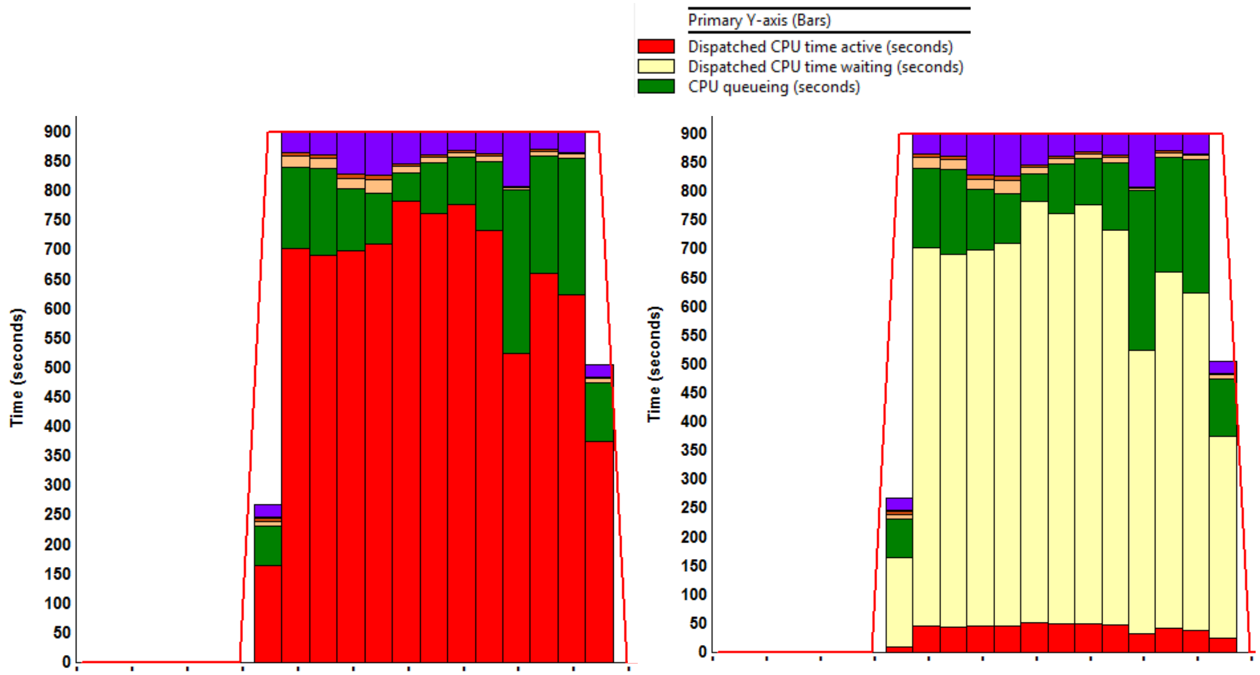
Run priority 20 gives much less CPU queuing than priority 50



Dispatched CPU is just to a VP = waiting and active



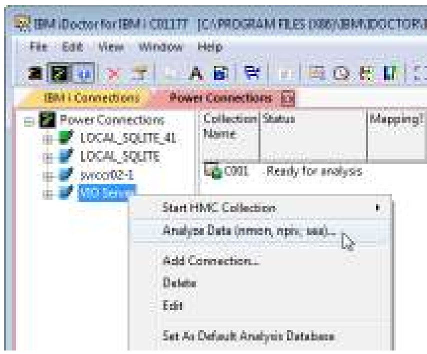
Similar graphs for a job with run priority 50

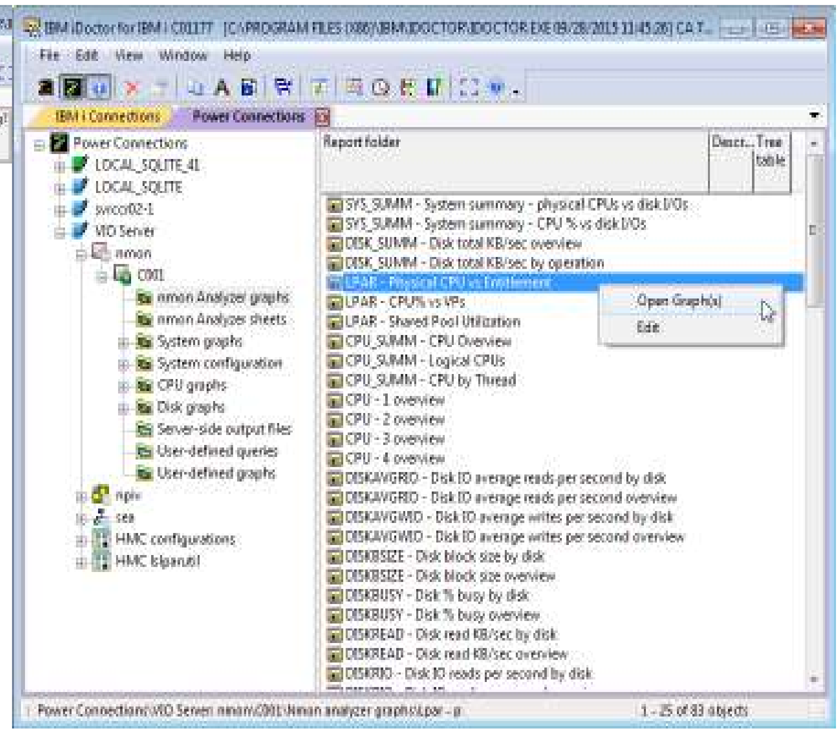


VIO Server performance

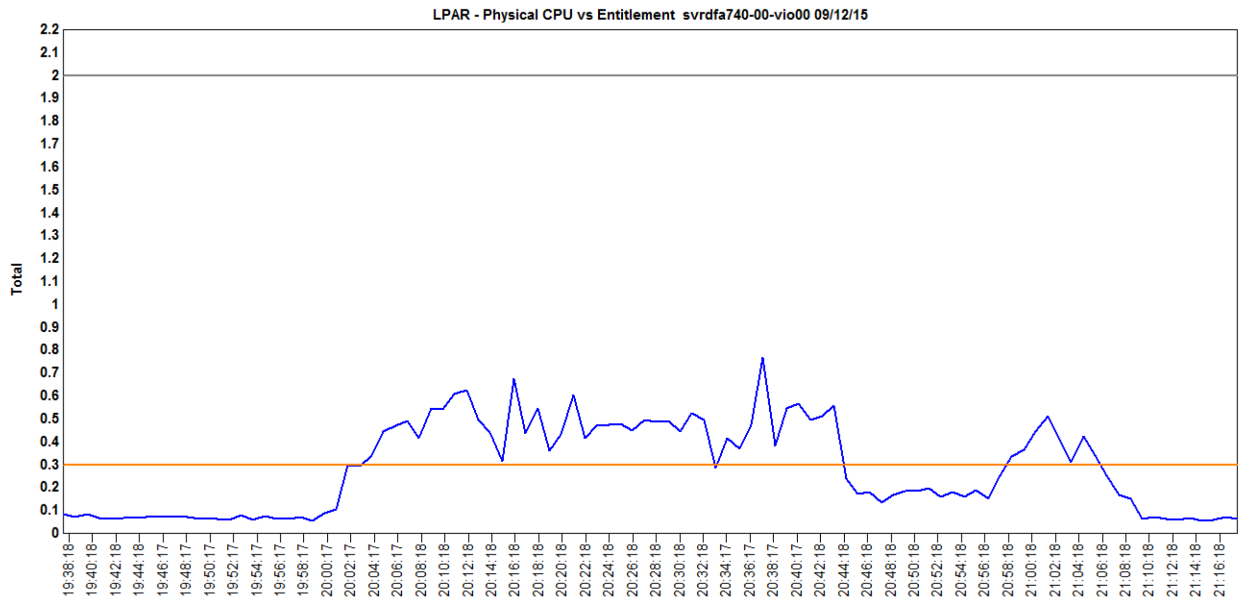
- You will need to look at the VIO Server, as **network** and **storage** can run through this partition
- For best performance insight, you will need to make use of NMON (Nigel Griffiths Monitor). It can be started on the VIOS like this:
  - topas_nmon -X -s 60 -c 480 -o /home/perf/nmon/vios01 -O -t -E -V -P -M -^ -L –A
- Some free tools are available for analysis of the data:
  - NMON Analyser
  - iDoctor/VIOS Investigator
- iDoctor can be downloaded here:
  - https://www-912.ibm.com/i_dir/idoctor.nsf/downloadoptions.html

iDoctor – VIOS Investigator





iDoctor – VIOS Investigator – VIOS over entitlement



VIOS - part command (originally VIOS Advisor)

- Provides performance reports with suggestions for making configurational changes to the environment, and helps to identify areas for further investigation.
- The reports are based on the key performance metrics of various partition resources that are collected from the Virtual I/O Server
- More info here:

https://www-01.ibm.com/support/knowledgecenter/POWER7/p7hcg/part.htm

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 13:25:13	-	n/a	n/a
	CPU Consumption	avg:27.1% (cores:1.1) high:27.4% (cores:1.1)	-	-	-	n/a	n/a
	Processing Mode	Shared CPU, (UnCapped)	-	08/17 13:25:13	-	n/a	n/a
	Variable Capacity Weight	128	129-255	08/17 13:25:13	-	1	5
	Virtual Processors	4	-	08/17 13:25:13	-	n/a	n/a
	SMT Mode	SMT4	-	08/17 13:25:13	-	n/a	n/a

Power Virtualization Performance (PowerVP)

Real and virtual resources

Individual VMs

System view

Real-time information

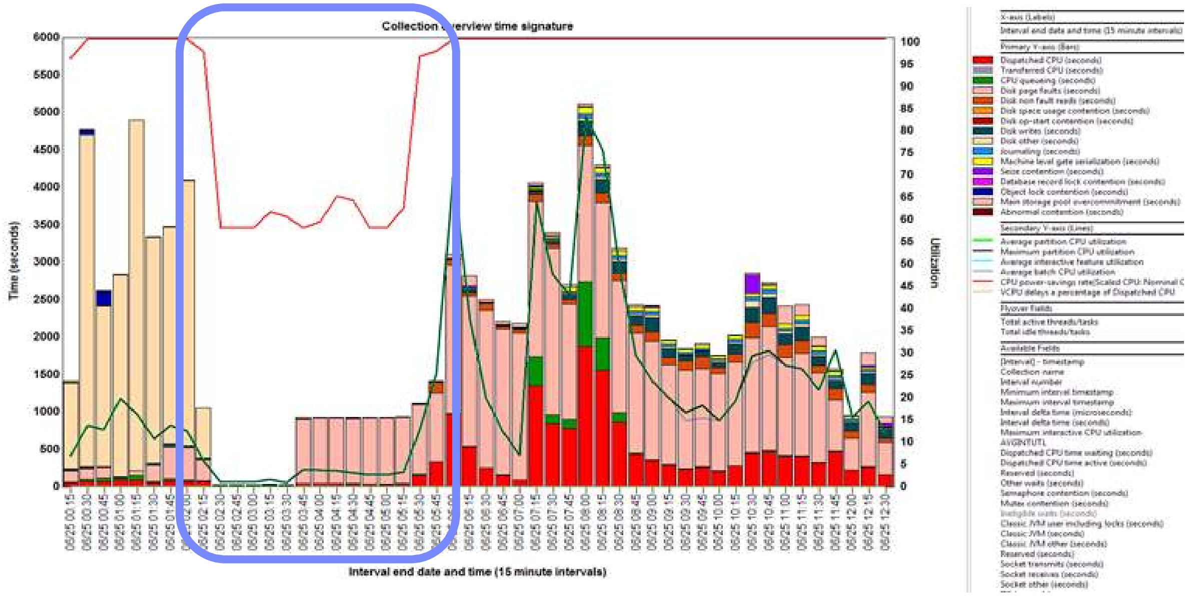
Replay saved data

AIX, Linux, IBM i

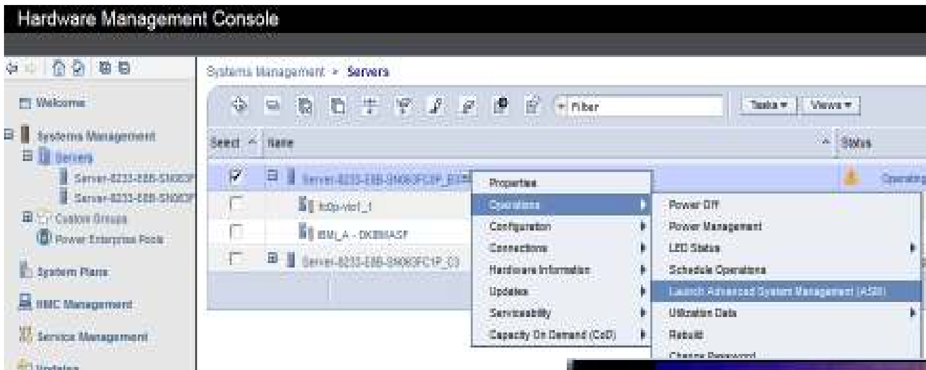
IBM i performance tips

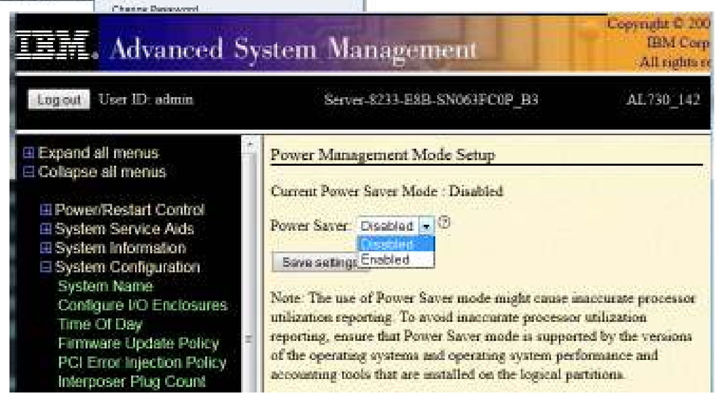
Power saving mode is per default active on P8

Running single jobs, as overnight can be effected by running on limited CPU resources. Use ASMI interface to deactivate if needed.

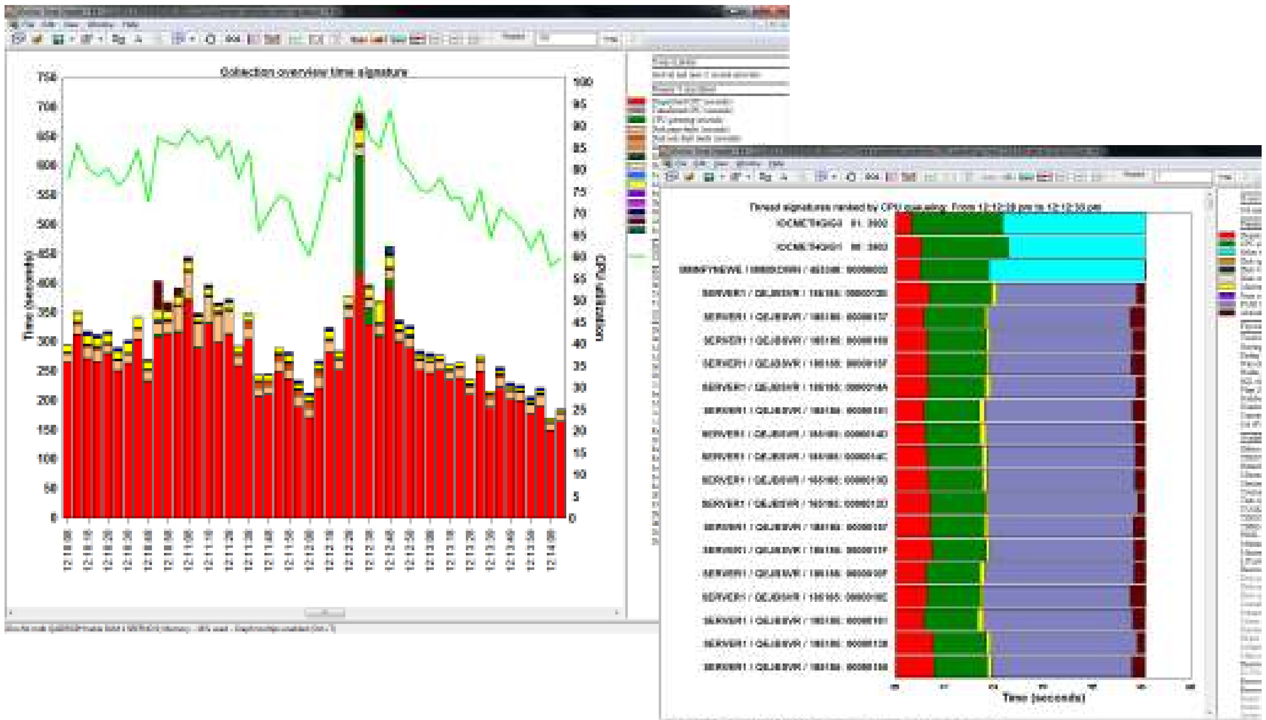


Disable power saving via ASMI

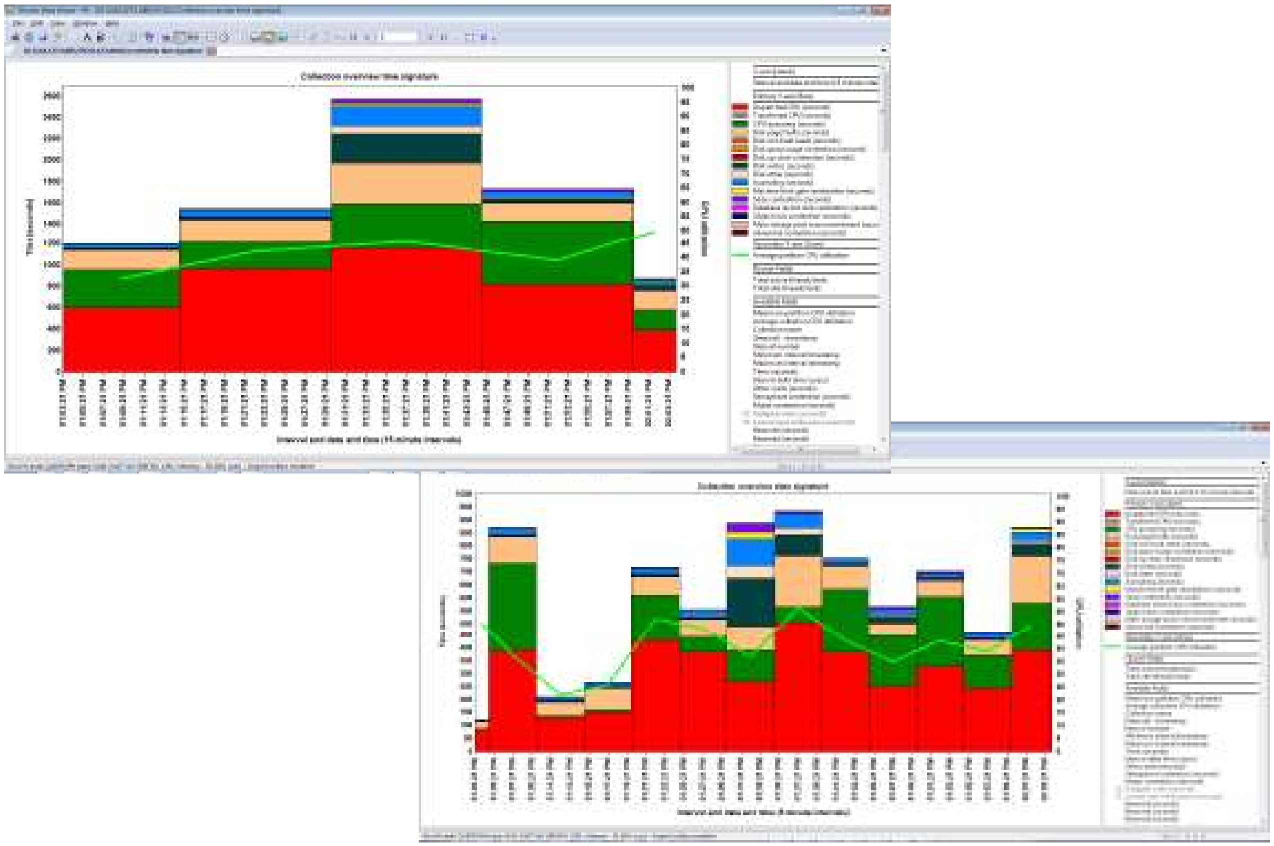




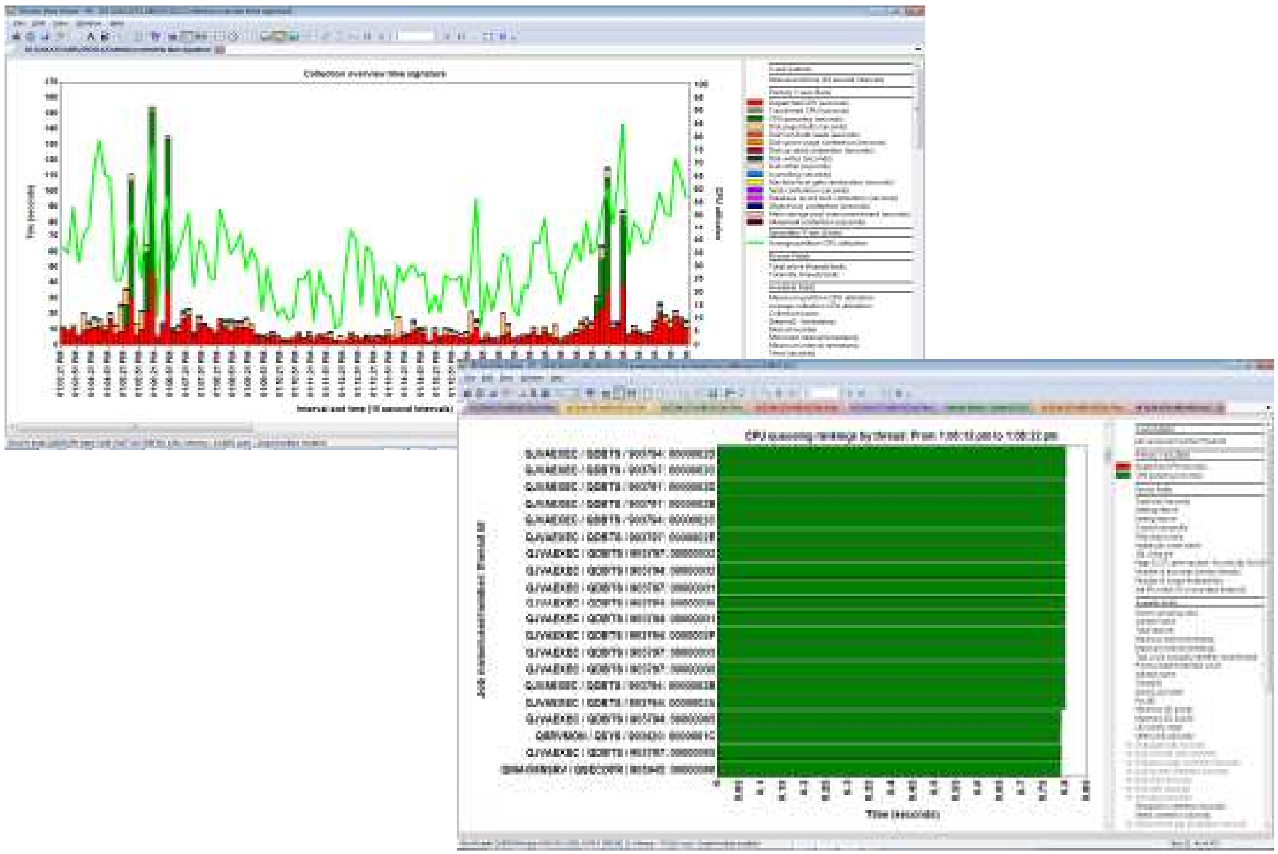
P7 and P8 default setting can cause CPU queuing for JVM



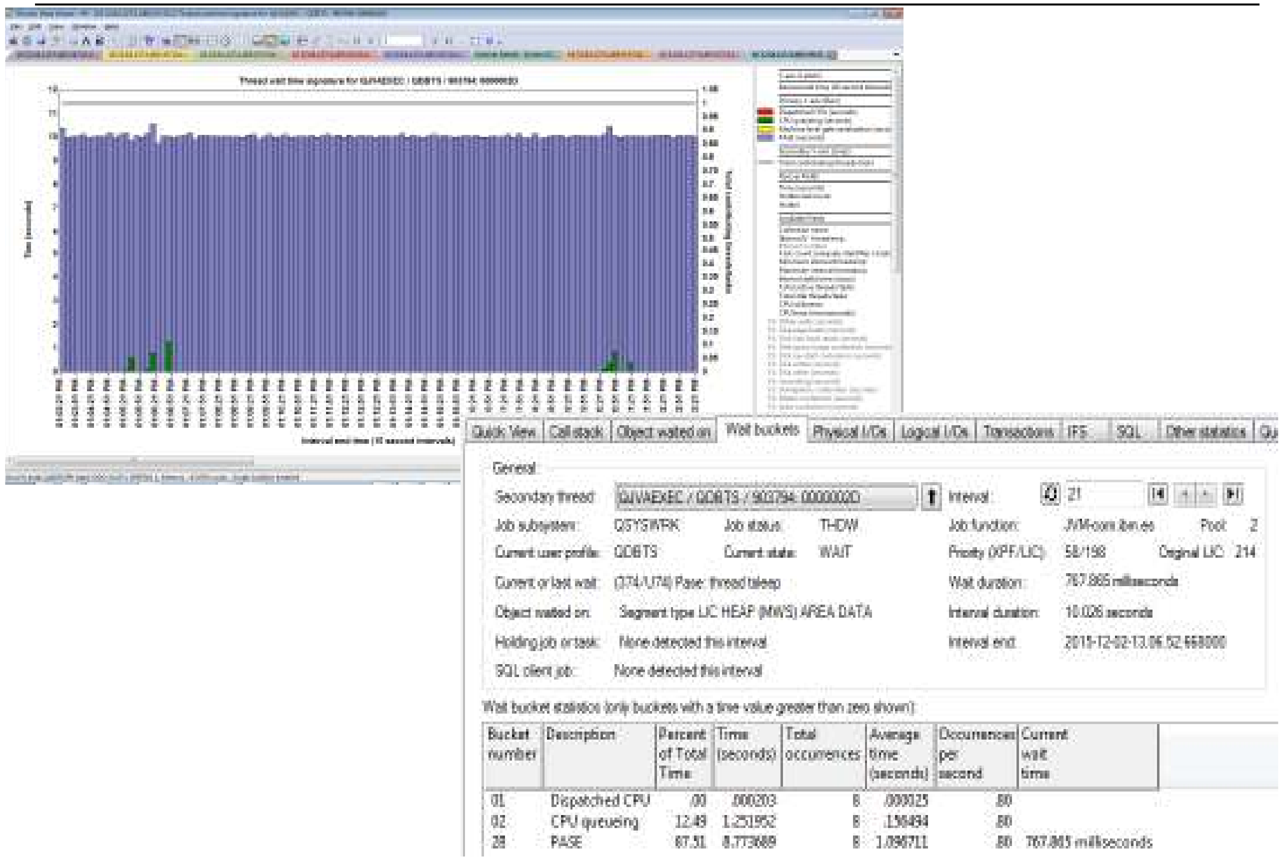
P7 example CPU queuing



P7 example CPU queuing



P7 example CPU queuing



Understanding the JVM (Java Virtual Machine) GC (Garbage Collection)

- Memory is cleaned up when no more addresses is available. This is called the GC (Garbage Collection).
- To be able to do the GC as fast as possible, then the GC has a number of help threads, they are called GC slave threads.
- The number of GC slave threads is per default depending on the hardware configuration.
  - Number of VP's in the partition multiplied by the SMT(1,2,4,8) minus 1.
  - If a P7 LPAR had VP of 20, then you would have the GC slave threads running here for ONE JVM:
    - 20 * 4 – 1 = 79 GC slave threads per JVM
- The –XgcthreadsXX setting could be set to control the number of threads running for GC.
- If you want to use 8 threads, for use of maximum 2 cores, which can be set using –Xgcthreads08
- By changing this for all JVM's in the partition, then reduction in active threads can be high.

JVM Systems properties

IBM i and the JVM determine the values for Java system properties by using the following order of precedence:

1. Command line or JNI invocation API
2. QIBM_JAVA_PROPERTIES_FILE environment variable
3. user.home SystemDefault.properties file
4. /QIBM/UserData/Java400/SystemDefault.properties
5. Default system property values

JVM default settings

- Environment variables are added like this:

EDTF STMF(‘/QIBM/UserData/Java400/mySystem.properties‘)

ADDENVVAR ENVVAR(QIBM_JAVA_PROPERTIES_FILE)
VALUE(/QIBM/userData/java400/mySystem.properties)

- The SystemDefault.properties can be changed here:

EDTF STMF(‘/QIBM/UserData/Java400/SystemDefault.properties‘)

- Add the following entries in both cases:

  - #AllowOptions
  - -Xgcthreads08

Java system properties

Knowledge center:

http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzaha/sysprop.htm?lang=en

Java™ system properties determine the environment in which you run your Java programs. They are similar to system values or environment variables in IBM® i.

Starting an instance of a Java virtual machine (JVM) sets the values for the system properties that affect that JVM.

You can choose to use the default values for Java system properties or you can specify values for them by using the following methods:

- Adding parameters to the command line (or the Java Native Interface (JNI) invocation API) when you start the Java program
- Using the QIBM_JAVA_PROPERTIES_FILE job-level environment variable to point to a specific properties file. For example:

```
ADDENVVAR ENVVAR(QIBM_JAVA_PROPERTIES_FILE)
  VALUE(/QIBM/userdata/java400/mySystem.properties)
```

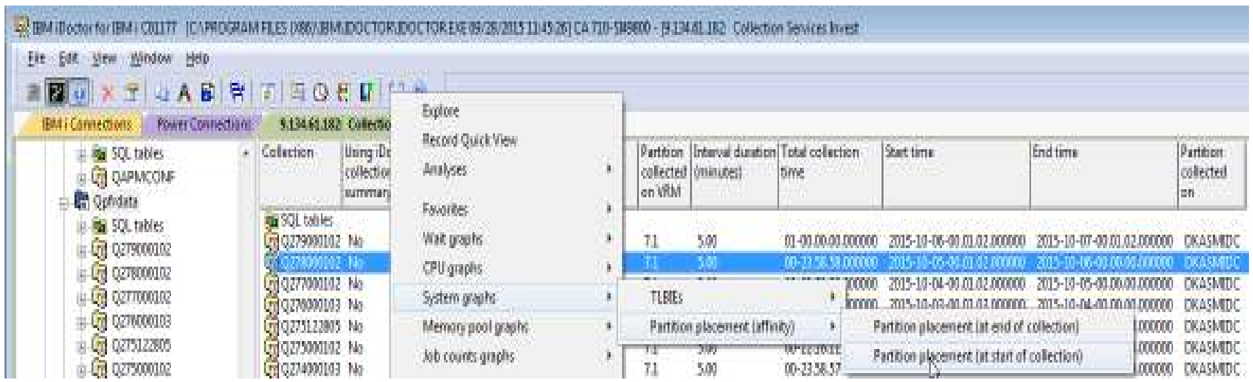
- Creating a SystemDefault.properties file that you create in your user.home directory
- Using the /QIBM/userdata/java400/SystemDefault.properties file

IBM i and the JVM determine the values for Java system properties by using the following order of precedence:

1. Command line or JNI invocation API
2. QIBM_JAVA_PROPERTIES_FILE environment variable
3. user.home SystemDefault.properties file
4. /QIBM/UserData/Java400/SystemDefault.properties
5. Default system property values

CPU/Memory placement

iDoctor/CSI shows the placement



CPU/Memory placement

Large partition with less good placement

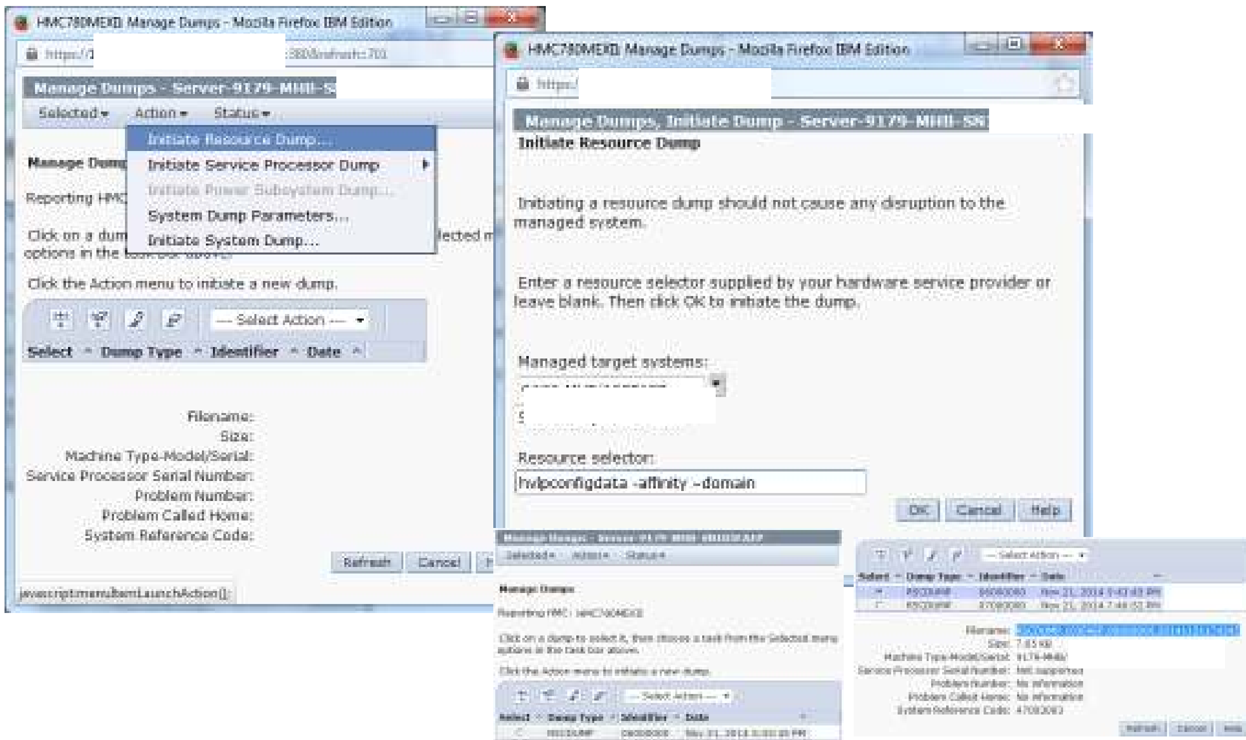
iDoctor Data Viewer - #1 - [Piraeus/QMPGDATA/Q211080003/Partition placement (at end of collection) - #1]

File Edit View Window Help

Interval end time	Shared LPAR	Affinity score	Book	Node	Partition processors	Partition memory (gigabytes)	Processors % of total	Memory % of total	Total processors	Total memory (gigabytes)
2014-07-31-08.00.00.000000	0	73	0000	0000	4	172.9121	8.33%	12.39%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0000	0001	4	123.4385	8.33%	8.85%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0000	0002	4	240.0762	8.33%	17.21%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0000	0003	4	123.4443	8.33%	8.85%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0001	0004	4	247.8301	8.33%	17.76%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0001	0005	4	123.4355	8.33%	8.85%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0001	0006	4	248.4609	8.33%	17.81%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0001	0007	4	115.0742	8.33%	8.25%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0002	000A	4	0	8.33%	.00%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0002	000B	4	0	8.33%	.00%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0002	0008	4	0	8.33%	.00%	48	1,394.6719
2014-07-31-08.00.00.000000	0	73	0002	0009	4	0	8.33%	.00%	48	1,394.6719

iDocCS.mdb QAIDRSQL table SUM 800 Rows 1 - 12 of 12

CPU/Memory placement via HMC dump



CPU/Memory placement via HMC dump

C:\Program Files (x86)\PuTTY>

pscp -scp hscroot@<ipadr>:/dump/RSCDUMP.103FAEP.06000000.20141121174345 c:\download\dmp1

- Some memory is not coming from a chip with cores
- 402 * 256MB = 100GB, so 1/5 of all memory access
- This may be solved by a frame reboot or if firmware/HMC/OS is on right level DPO (Dynamic Platform Optimizer)

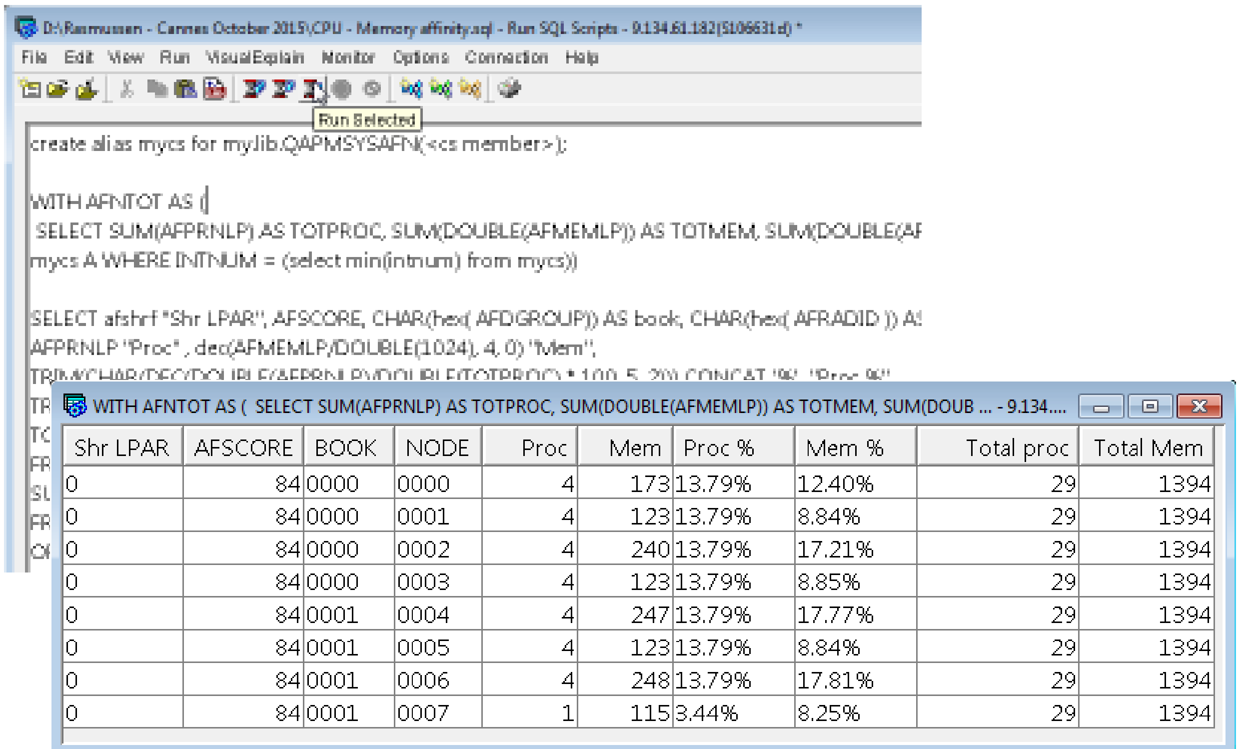
Shared Pool domains

Domain		Procs		Units	Memory			Proc Units		Memory		Ratio
SEC	PRI	Total	Free	Free	Total	Free	LP	Tgt	Aloc	Tgt	Aloc	
0		1600	0	0	1024	105						0
	0	800	0	0	512	0						0
							1	800	800	478	478	
	1	800	0	0	512	105						0
							1			402	402	
1		1600	0	0	768	105						0
	4	800	0	0	512	71						0
							1	400	400	440	440	
	5	800	0	0	256	34						0
							1	200	200	220	220	
2		1600	0	0	768	105						0
	8	800	0	0	512	71						0
							1	400	400	440	440	
	9	800	0	0	256	34						0
							1	200	200	220	220	

CPU/Memory placement from CS (original from iDoctor)

```
create alias mycs for mylib.QAPMSYSAFN(<cs member>);
WITH AFNTOT AS (
 SELECT SUM(AFPRNLP) AS TOTPROC, SUM(DOUBLE(AFMEMLP)) AS
TOTMEM, SUM(DOUBLE(AFMEMLP))/1024 AS TOTMEM_GB FROM
mycs A WHERE INTNUM = (select min(intnum) from mycs))
SELECT afshrf "Shr LPAR", AFSCORE, CHAR(hex( AFDGROUP)) AS book,
CHAR(hex( AFRADID )) AS node,
AFPRNLP "Proc" , dec(AFMEMLP/DOUBLE(1024), 4, 0) "Mem",
TRIM(CHAR(DEC(DOUBLE(AFPRNLP)/DOUBLE(TOTPROC) * 100, 5, 2))) CONCAT
'%'  "Proc %",
TRIM(CHAR(DEC(DOUBLE(AFMEMLP)/DOUBLE(TOTMEM) * 100, 5, 2))) CONCAT
'%'  "Mem %",
TOTPROC "Total proc", dec(TOTMEM_GB, 4, 0) "Total Mem"
FROM (SELECT A.*, '20' || SUBSTR(DTETIM, 1, 2) || '-' || SUBSTR(DTETIM, 3, 2) || '-
' || SUBSTR(DTETIM, 5, 2) || '-' || SUBSTR(DTETIM, 7, 2) || '.' ||
SUBSTR(DTETIM, 9, 2) || '.' || SUBSTR(DTETIM, 11, 2) || '.000000' AS INTENDSTR
FROM mycs A WHERE INTNUM = (select min(intnum) from mycs)) X, AFNTOT
ORDER BY 4, 5;
```

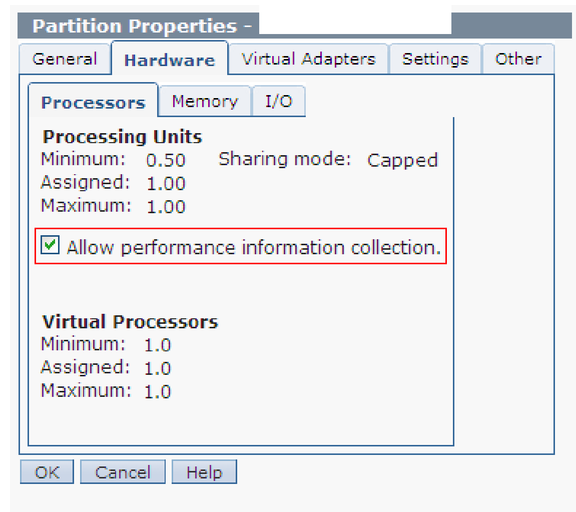
CPU/Memory placement from CS (original from iDoctor)



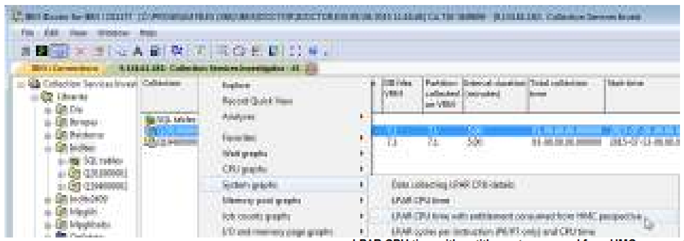
Shr LPAR	AFSCORE	BOOK	NODE	Proc	Mem	Proc %	Mem %	Total proc	Total Mem
0	84	0000	0000	4	173	13.79%	12.40%	29	1394
0	84	0000	0001	4	123	13.79%	8.84%	29	1394
0	84	0000	0002	4	240	13.79%	17.21%	29	1394
0	84	0000	0003	4	123	13.79%	8.85%	29	1394
0	84	0001	0004	4	247	13.79%	17.77%	29	1394
0	84	0001	0005	4	123	13.79%	8.84%	29	1394
0	84	0001	0006	4	248	13.79%	17.81%	29	1394
0	84	0001	0007	1	115	3.44%	8.25%	29	1394

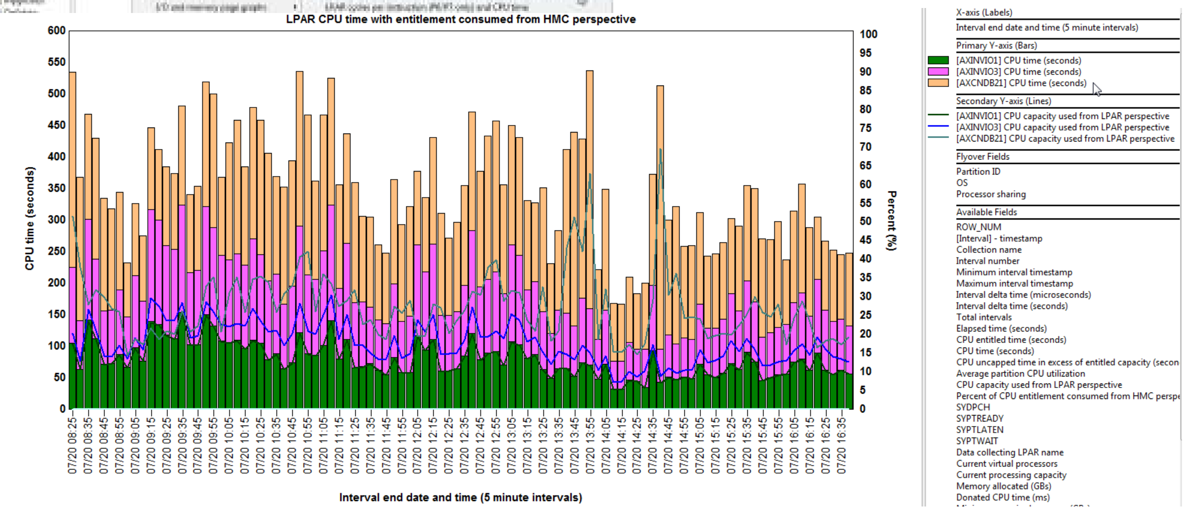
HMC performance data collection

- Enable performance collection in the LPAR profile to see shared processor pool utilization in Collection Service data and collection in NMON (VIOS, AIX and Linux)
- Performance information collection can be enable in the partition profile’s Hardware tab. This is a dynamic setting and does not require profile reactivation.
- This will also allow to get the affinity score from the HMC (similar to the DPO (Dynamic Platform Optimizer).



Other LPAR's using resources





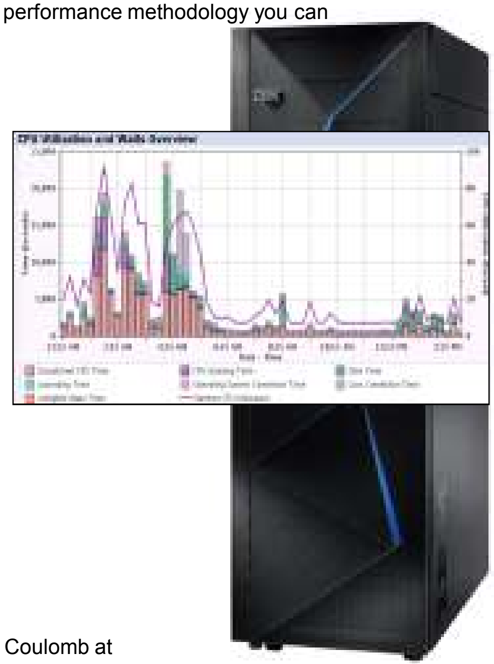
Session summary

- Introduction to Virtualization
- Why is virtualization performance important for IBM i?
- VIO Server
- CPU and memory affinity
  - How to check the CPU/Memory placement
- Does other LPAR's using the resources
  - CPU information from HMC
- Micro partitioning is fine but can cause more wait time as expected
- Virtualization tips on IBM i
  - Be aware on power saving mode
  - Default JVM setting should be reviewed

IBM i Performance Analysis Workshop

Learn the science and art of performance analysis, methodology and problem solving

Managing and analyzing the data can be quite complex. During this workshop, the IBM Systems Lab Services IBM i team will share useful techniques for analyzing performance data on key IBM i resources, and will cover strategies for solving performance problems. It will aid in building a future foundation of performance methodology you can apply in your environment.



Overview:

- Topics covered include:
  - Key performance analysis concepts
  - Performance tools
  - Performance data collectors (Collection Services, Job Watcher, Disk Watcher, and Performance Explorer)
  - Wait accounting
- Core methodology and analysis of:
  - Locks
  - Memory
  - I/O subsystem
  - CPU
- Concept reinforcement through case studies and lab exercises
- Discussions on theory, problem solving, prevention and best practices

Workshop details:

- Intermediate IBM i skill level
- 3 day workshop in Copenhagen (25-27 April), Prague (16-18 May)
  - For more information and other locations: IBM i Performance Analysis Workshop
  - For additional information and enrolment, please contact Beatrice Coulomb at BCOULOMB@fr.ibm.com. Remember that you can use your service voucher and education vouchers.

IBM Lab Services Vouchers for IBM i

- With the IBM i and selected Power Systems™ servers, valuable education and services vouchers are included at **no additional charge**
- Vouchers are designed to help you more fully understand and use the advanced features and capabilities of IBM i
- Vouchers are only available with selected new Power Systems servers, and vouchers are not available with system upgrades
- Vouchers are valid for 5 years beyond the ship date

- For more information, eligible systems and registration information see: http://www-03.ibm.com/systems/power/hardware/vouchers/index.html
- Or contact:
  - Beatrice Coulomb - BCOULOMB@fr.ibm.com
  - Claude Roustan - claude.roustan@fr.ibm.com

IBM i Vouchers – Available services

- **IBM i Performance**
  - **IBM i Performance**
  - **SQL Performance**
- **IBM i Database**
  - **DB2 for IBM i Best Practices**
  - **DB2 Web Query for IBM i**
- Security
  - Security Assessment
  - PowerSC
  - Single Sign On
- Availability
  - IBM i Availability Assessment
  - IBM i BRMS
  - PowerHA on IBM i
- System Solutions
  - Migration Assistance
  - PowerVM Virtual I/O Server and IBM i
  - External Storage for IBM i
- Middleware
  - WebSphere® with IBM i
  - PHP and Open Source on IBM i
- Applications
  - SAP on IBM