Université IBM i 2018

16 et 17 mai

IBM Client Center Paris

S48 – Best Practices for IBM i Memory Tuning for Performance

Stacy L. Benfield
IBM i Performance Consultant - Lab Services Power Systems Delivery Practice
IBM i Large User Group (LUG) Program Manager
IBM, Rochester MN
stacylb@us.ibm.com

Agenda

- Memory Concepts and page faulting
- IBM i memory tuning features
- Tools used to identify memory issues
- Best practices for memory tuning

Importance of Memory

- The CPU(s) can’t do anything if the required data is not in memory
- Having to go to disk is slow, in relative terms

How to improve performance:

- Tuning can help your system use its memory better
- IBM i has technology built in to help manage memory automatically
- IBM i has excellent performance tools for analyzing memory performance
- Adding hardware such as more memory or SSDs can reduce I/O wait times
- Application changes can greatly reduce how often your jobs wait on disk I/O

Bringing data into memory

- Implicit memory transfers – ***"page fault reads"***
  - Page faults – Synchronous memory transfer where jobs wait until I/O completes
  - I/O Pending faults – waiting on your/or someone else's I/O request to complete
  - Collection Services & Job Watcher have a "wait bucket" to track this time

- Explicit memory transfers– ***"non-fault reads"***
  - Asynchronous memory transfer where OS pre-fetches data before application needs it (jobs don't have to wait)
    - DB2 "read-ahead" or "asynchronous brings"
  - SETOBJACC command puts an object into a memory pool
  - Read part of CRTDUPOBJ and CPYF operations
  - Collection Services & Job Watcher have a "wait bucket" to track this time

Types of I/O Requests for Reads and Writes

- Synchronous I/Os
  - Processing waits until I/O completes
  - Contributes to response/runtime
  - Examples (reads):

    SETOBJACC, CRTDUPOBJ, CPYF, Faults, …
- Asynchronous I/Os
  - Processing concurrent with I/O
  - Can turn synchronous
  - Examples (reads):

    DB2 asynchronous brings

Page Fault basics

The primary focus for memory analysis is to reduce the amount of time waiting on disk faults.

- Page Faults are normal and expected
  - It is the mechanism used to bring most things into memory
  - Some things need to be brought into memory to be cleaned up / deleted
- A single page fault can bring multiple pages into memory
  - Faults per second < pages per second
  - We don’t typically care about pages per second
- Requested pages are placed into the memory pool the job is running in
  - Pages can be shared between jobs
  - A job can access a page in a different pool

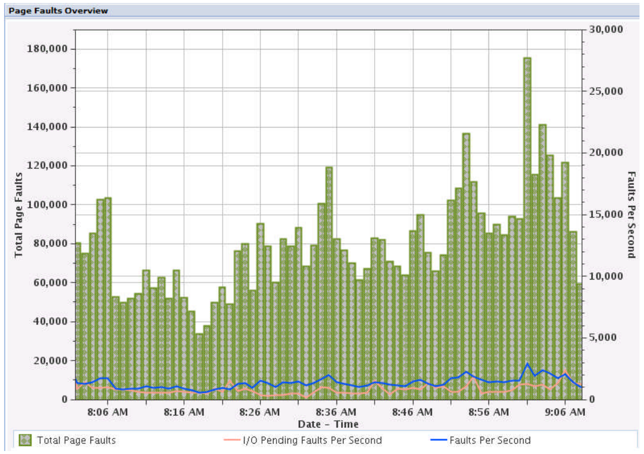
Why do we care if page fault wait times are high

- High page fault wait times indicates pool tuning issues, unnecessary memory usage, and/or lack of memory that will affect performance
  - Longer interactive response times for I/O intensive transactions
  - Longer batch run times
  - Poor disk response times
  - Less efficient query implementation methods

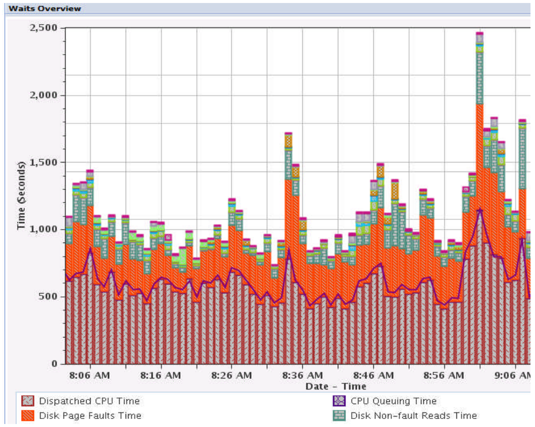
Page Faults – Faults Per Second vs. Wait Time - PDI

- Focus on time being spent waiting on faults, **not** the rate of faults
  - Large memory pools can have a high number of faults, but little impact to wait times
- However, rates can be useful for monitoring and when you can't get wait time easily
  - Dashboard, System Monitors, WRKSYSSTS, etc.
  - Also can indicate new workload has started up
- Which chart below is more helpful in determining if faulting might be a problem?

faults/sec

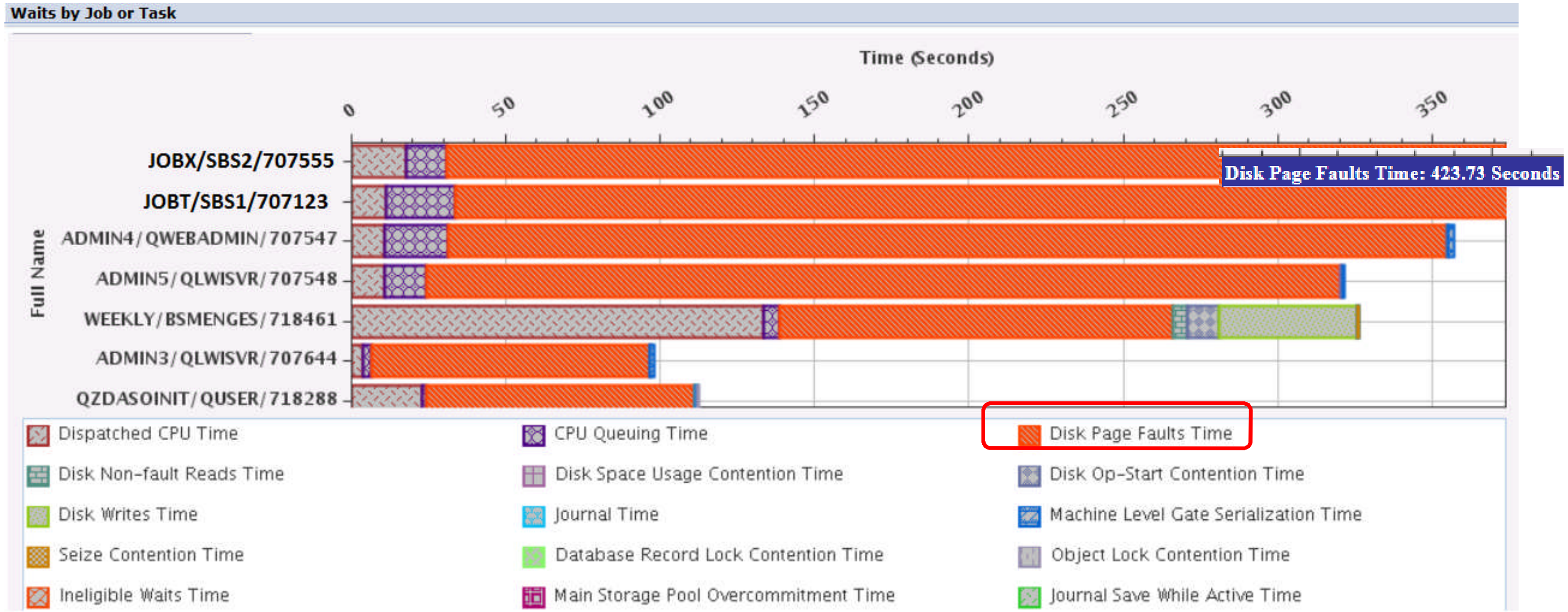


wait time



Page Faults Wait Time at a Job Level

- Collection Services (and Job Watcher) collect disk page fault wait time both at a **system** level, as well as an **individual job/thread/task level**
- ??? Is it more interesting to know that JOBX did 120 faults per second, or that it waited 370 seconds (out of 5 minutes) on disk page faults



IBM tasks associated with removing data from memory

System Controlled

- There are two tasks that are responsible for writing changed pages of memory out to disk to make room for data being paged into memory.
  - SMPOL001: low priority page out task
  - SMPO0001: high priority page out task
    - becomes active if the low priority task cannot keep up with demand
- High activity in these tasks, especially the high priority task, can indicate a need for additional memory.

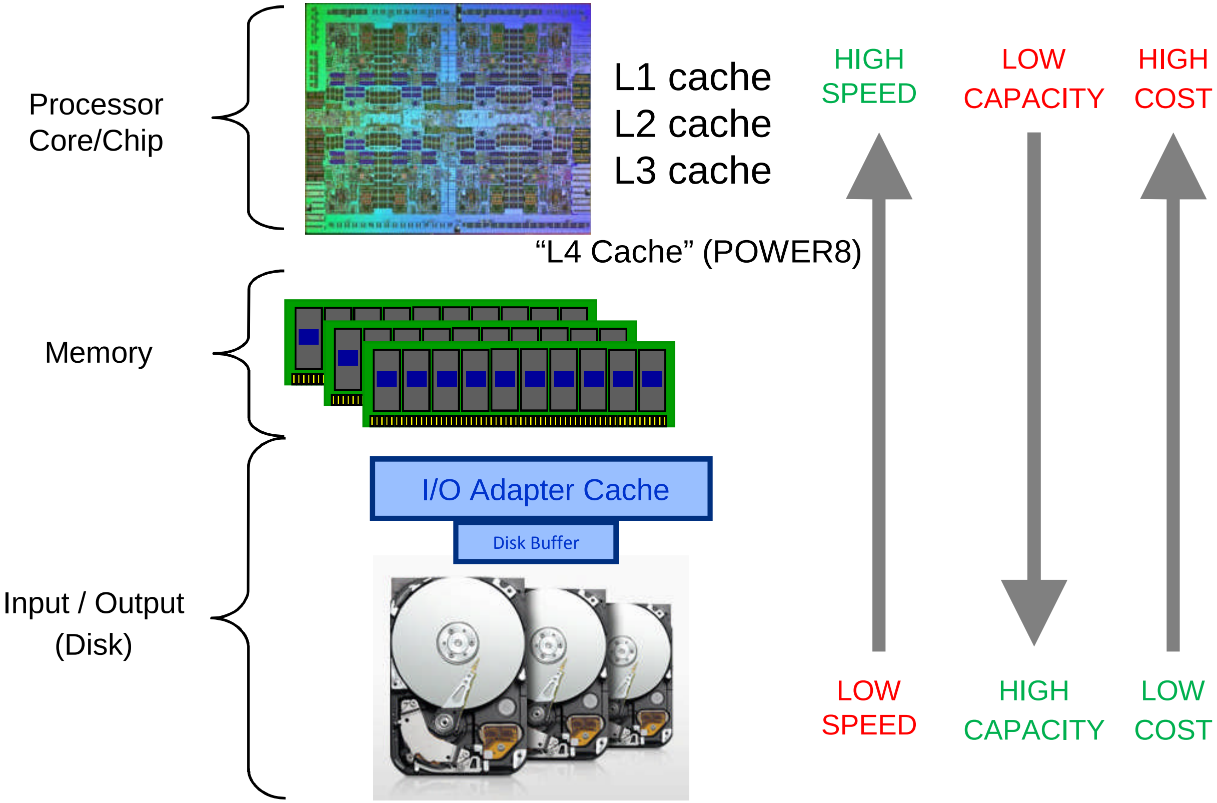
User Program Controlled

- Changed pages are written out when a job ends

Single-level Storage

- Unique to IBM i
- Main memory, traditional spinning disks, and solid-state drives all appear as **one** address space
- Objects automatically managed by the OS, moving between memory and disk
- One copy of file/data can be shared by many users, jobs, programs
- Objects spread across drives for parallelism
- No need to create “table spaces”, “buffer pools”, etc. required by other platforms
- **Think of memory as a giant cache for disk**

IBM i Memory / Storage Hierarchy



Processor Core/Chip

L1 cache
L2 cache
L3 cache

"L4 Cache" (POWER8)

Memory

I/O Adapter Cache

Disk Buffer

Input / Output (Disk)

HIGH SPEED

LOW CAPACITY

HIGH COST

LOW SPEED

HIGH CAPACITY

LOW COST

POWER Memory / Storage Hierarchy

	Size	Speed / cycles
POWER8 L1 cache	32K instr + 64K data per core	~3 cycles
POWER8 L2 cache	512 KB per core	~10 cycles
POWER8 L3 cache	96 MB shared per chip	~30 cycles
POWER8 “L4 cache”	16 MB per memory chip*	~200 cycles
Internal Memory	10s of GB per core	~300-800 cycles (~100ns)
Solid State Drives	100s of GB per drive	<1 ms
Hard Disk Drives	Can be TBs per drive	1-5 ms

100x

10,000x

Range due to “distance” of access.

1 ms = 1000 µs = 1,000,000 ns
400 cycles is about 0.1 µs if 4.0 GHz

Memory Configuration – Verify there's enough memory first

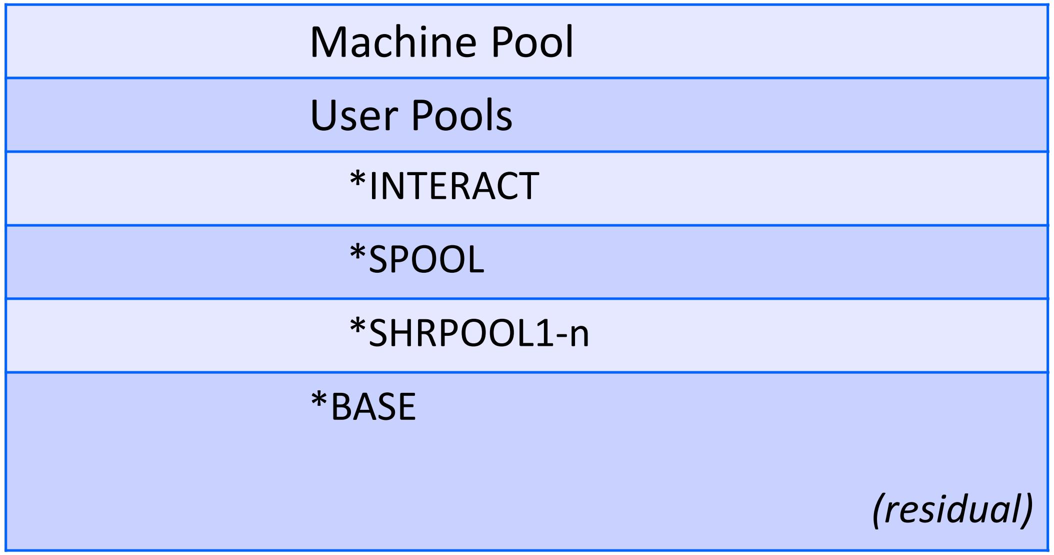
- A general rule of thumb for memory (minimum requirements) based off internal benchmarks
  - 32 GB/core for POWER8
  - 24 GB/core for POWER7/7+
  - 16 GB/core for POWER6
- Partitioning considerations
  - If dynamically adding cores, also typically want to add memory (DLPAR)
  - Uncapping can lead to an imbalance between CPU and memory
- Can utilize SSDs/Flash technology to improve page fault wait times

Memory Pools

- Memory pools are logical subdivisions of physical memory
- Used with subsystems to isolate memory usage by different applications
- Two types – shared and private

	Shared	Private
Subsystems	single or multiple	single
QPFRADJ	yes	no
Expert Cache	yes	no

Memory Pools



Machine Pool

User Pools

*INTERACT

*SPOOL

*SHRPOOL1-n

*BASE

(residual)

Pool Maximum Activity Level

- The maximum number of threads in the pool that can use the CPU(s) concurrently
  - Threads without an activity level are ineligible to run
- Does not apply to the machine pool
  - No jobs run in the machine pool
- Can be adjusted + or - by the Performance Adjustor
  - Adjuster conservative on decreasing
- Considerations
  - Generally want high enough to avoid transitions to ineligible
    - There is an “Ineligible Waits time” bucket in CS and JW
  - Setting too low can lead to severe performance problems
  - Setting too high can lead to more faulting

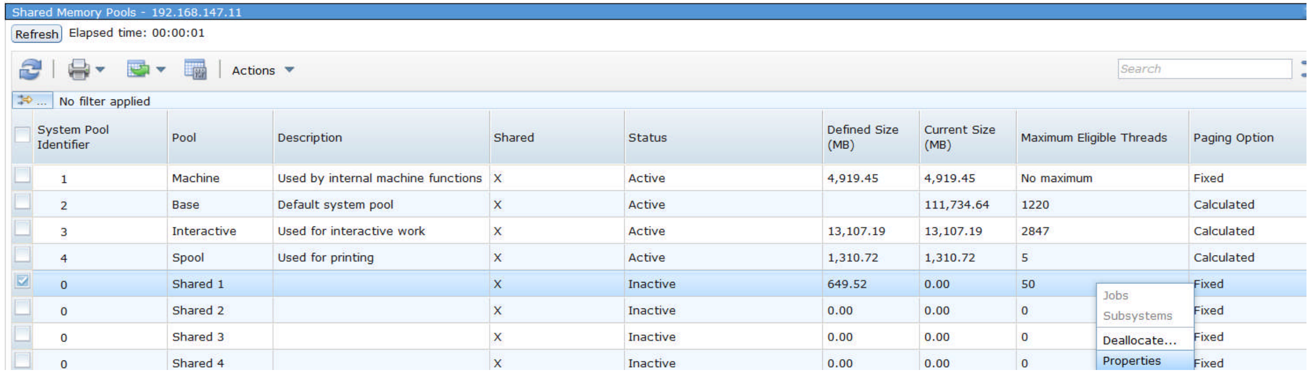
IBM i memory tuning features

- Expert Cache
- Automatic performance adjust system value (QPFRADJ)
- SETOBJACC command
- DB2 keep in memory

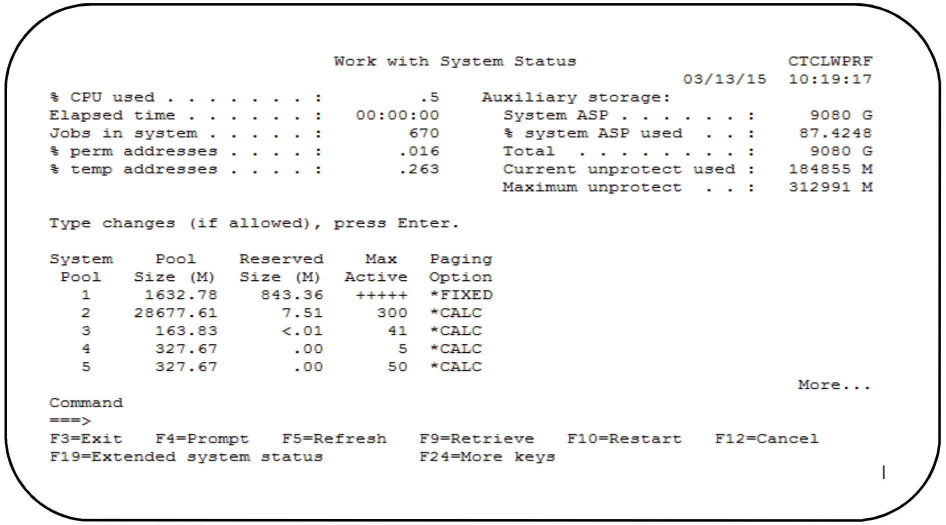
Expert Cache (*CALC)

- Paging parameter for shared storage pools
  - *FIXED
  - ***CALC** – enables expert cache
- What it does:
  - Monitors the I/O reference pattern for database files
  - Reduces I/O operations by adjusting the size and type of I/Os
- Cannot be used by:
  - Private pools (can be used by running an API – QWCCHGTN)
  - Machine pool
- Activity you may see:
  - SMXCSPRVSR (Expert Cache supervisor) task
  - SMXCAGERnn tasks (1 per *CALC pool, 01 => pool 2)
- Expert Cache almost always provides benefit and should be enabled ★

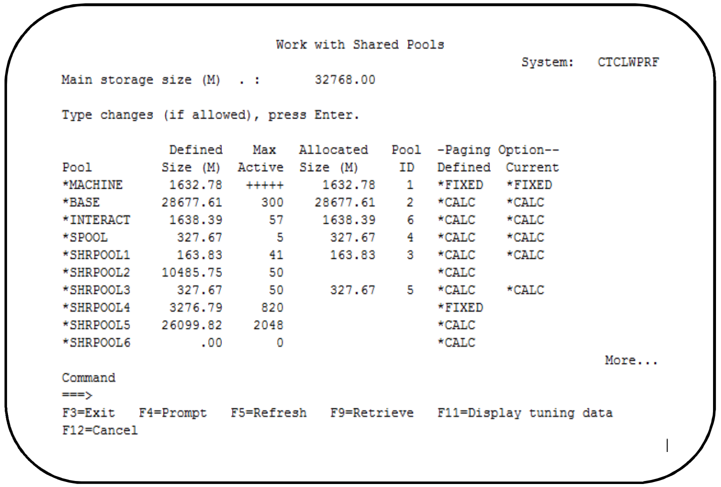
Expert Cache Enablement



□	System Pool Identifier	Pool	Description	Shared	Status	Defined Size (MB)	Current Size (MB)	Maximum Eligible Threads	Paging Option
□	1	Machine	Used by internal machine functions	X	Active	4,919.45	4,919.45	No maximum	Fixed
□	2	Base	Default system pool	X	Active		111,734.64	1220	Calculated
□	3	Interactive	Used for interactive work	X	Active	13,107.19	13,107.19	2847	Calculated
□	4	Spool	Used for printing	X	Active	1,310.72	1,310.72	5	Calculated
☑	0	Shared 1		X	Inactive	649.52	0.00	50	Fixed
□	0	Shared 2		X	Inactive	0.00	0.00	0	Fixed
□	0	Shared 3		X	Inactive	0.00	0.00	0	Fixed
□	0	Shared 4		X	Inactive	0.00	0.00	0	Fixed



```
                    Work with System Status                      CTCLWPRF
                                                       03/13/15  10:19:17
% CPU used . . . . . . . :          .5    Auxiliary storage:
Elapsed time . . . . . . :    00:00:00      System ASP . . . . . . :     9080 G
Jobs in system . . . . . :         670      % system ASP used  . . :    87.4248
% perm addresses . . . . :        .016      Total  . . . . . . . . :     9080 G
% temp addresses . . . . :        .263      Current unprotect used :   184855 M
                                            Maximum unprotect  . . :   312991 M

Type changes (if allowed), press Enter.

System      Pool    Reserved    Max   Paging
 Pool    Size (M)   Size (M)  Active  Option
   1      1632.78     843.36   +++++  *FIXED
   2     28677.61       7.51     300  *CALC
   3       163.83       <.01      41  *CALC
   4       327.67        .00       5  *CALC
   5       327.67        .00      50  *CALC
                                                                          More...
Command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Restart   F12=Cancel
F19=Extended system status         F24=More keys
```

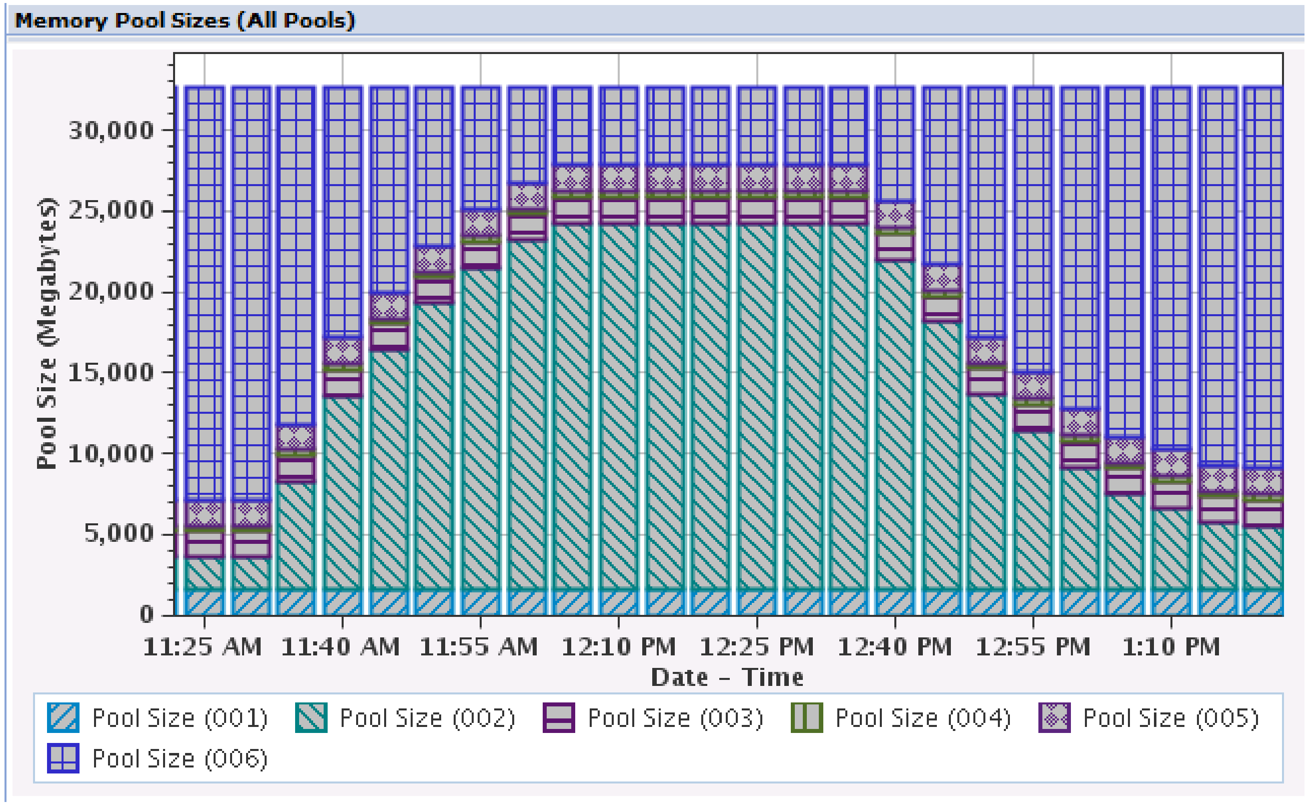


```
                           Work with Shared Pools
                                                             System:   CTCLWPRF
Main storage size (M)  . :       32768.00

Type changes (if allowed), press Enter.

               Defined    Max   Allocated   Pool  -Paging Option--
Pool           Size (M)  Active  Size (M)    ID   Defined   Current
*MACHINE        1632.78  +++++     1632.78    1   *FIXED    *FIXED
*BASE          28677.61    300    28677.61    2   *CALC     *CALC
*INTERACT       1638.39     57     1638.39    6   *CALC     *CALC
*SPOOL           327.67      5      327.67    4   *CALC     *CALC
*SHRPOOL1        163.83     41      163.83    3   *CALC     *CALC
*SHRPOOL2      10485.75     50                    *CALC
*SHRPOOL3        327.67     50      327.67    5   *CALC     *CALC
*SHRPOOL4       3276.79    820                    *FIXED
*SHRPOOL5      26099.82   2048                    *CALC
*SHRPOOL6           .00      0                    *CALC
                                                                       More...
Command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F11=Display tuning data
F12=Cancel
```

The Performance Adjuster

- Enabled using the QPFRADJ system value.
- Will manage the size of the shared memory pools for you.
  - Will also adjust the maximum activity level.
- Uses complex algorithms to ensure your pools are operating at peak efficiency.
- Refer to *The Performance Adjuster (QPFRADJ)* experience report on the IBM i Information Center.
  http://publib.boulder.ibm.com/infocenter/iseries/v7r1m0/index.jsp?topic=%2Fexperience%2Fwork3abstract.htm

Automatic Changes in Pool Sizes - PDI



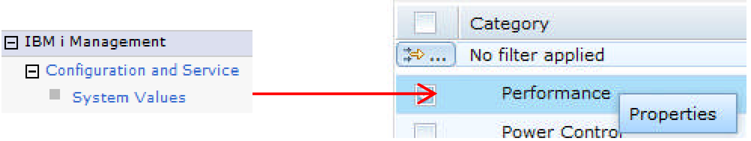
Turning QPFRADJ On or Off

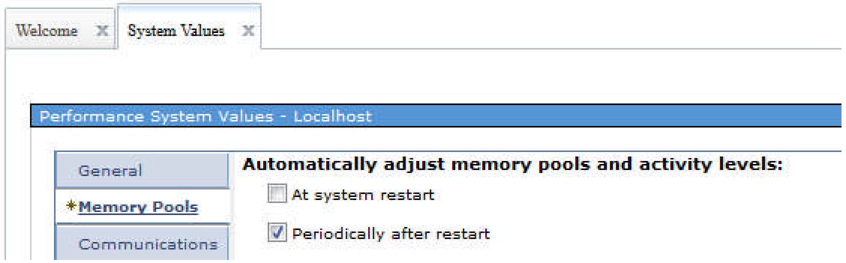
The Performance Adjuster is controlled by the System Value QPFRADJ

- 0=No adjustment
- 1=At IPL only based on static information
- 2=At IPL and automatic (default)
- 3=Automatic only

User managed!

Recommended

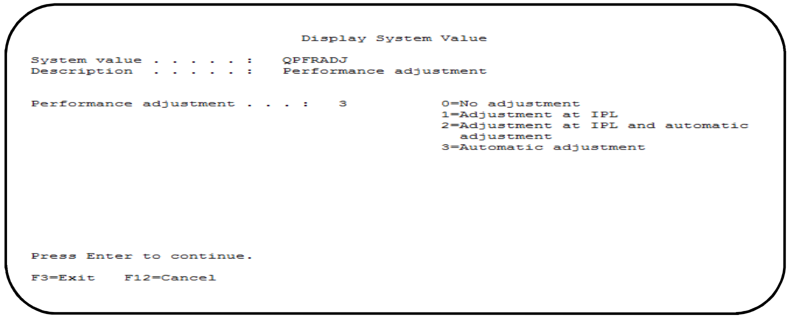




Automatically adjust memory pools and activity levels:

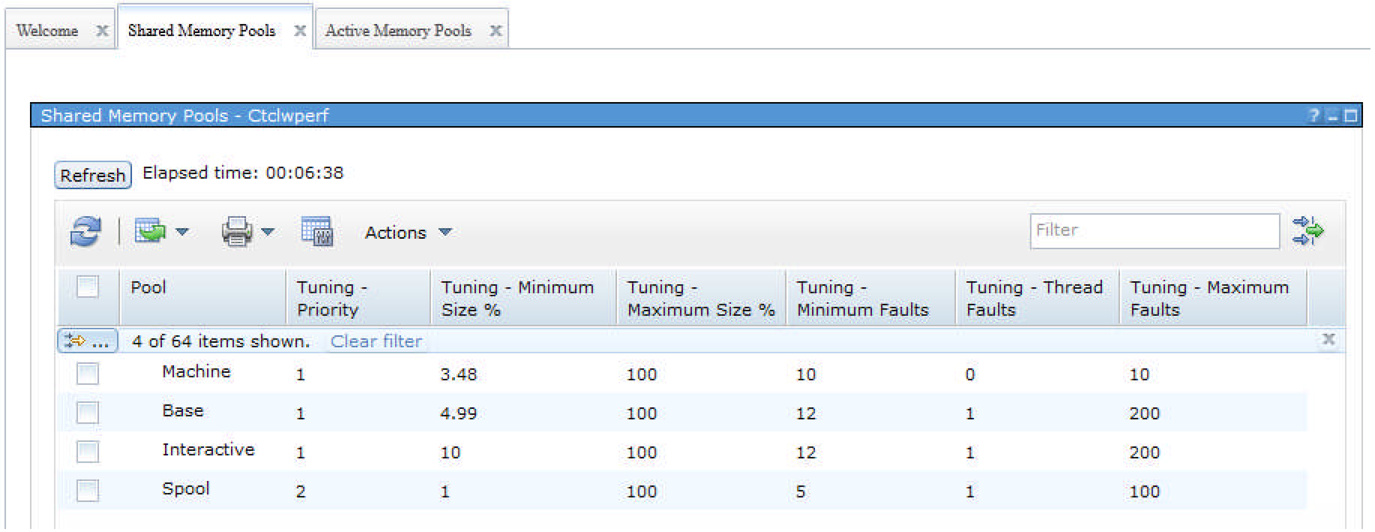
☐ At system restart

☑ Periodically after restart



```
                         Display System Value

System value . . . . . . :   QPFRADJ
Description  . . . . . . :   Performance adjustment

Performance adjustment . . . :   3            0=No adjustment
                                              1=Adjustment at IPL
                                              2=Adjustment at IPL and automatic
                                                adjustment
                                              3=Automatic adjustment







Press Enter to continue.

F3=Exit   F12=Cancel
```

Tuning QPFRADJ (WRKSHRPOOL)

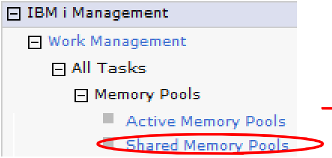


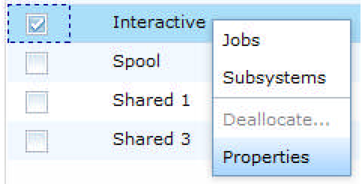
Shared Memory Pools - Ctclwperf

Elapsed time: 00:06:38

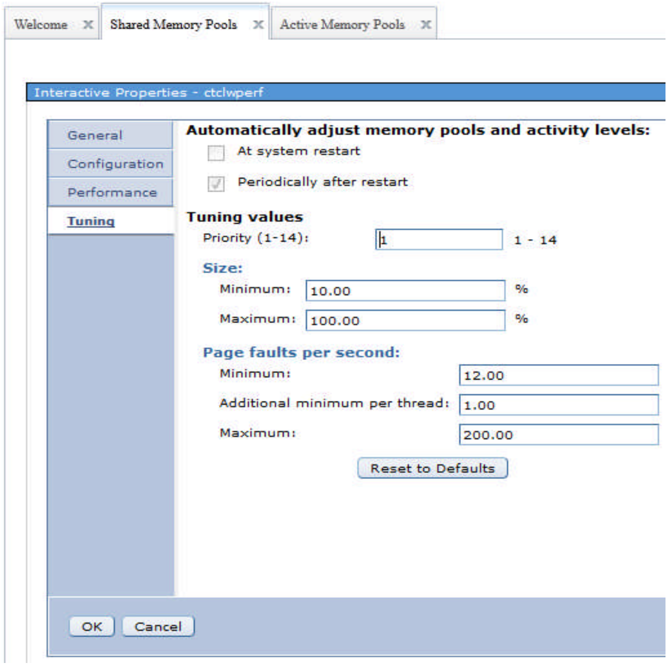
4 of 64 items shown.

□	Pool	Tuning - Priority	Tuning - Minimum Size %	Tuning - Maximum Size %	Tuning - Minimum Faults	Tuning - Thread Faults	Tuning - Maximum Faults
□	Machine	1	3.48	100	10	0	10
□	Base	1	4.99	100	12	1	200
□	Interactive	1	10	100	12	1	200
□	Spool	2	1	100	5	1	100





Tuning QPFRADJ



Automatically adjust memory pools and activity levels:

□ At system restart

☑ Periodically after restart

Tuning values

Priority (1-14): 1   1 - 14

Size:

Minimum: 10.00 %

Maximum: 100.00 %

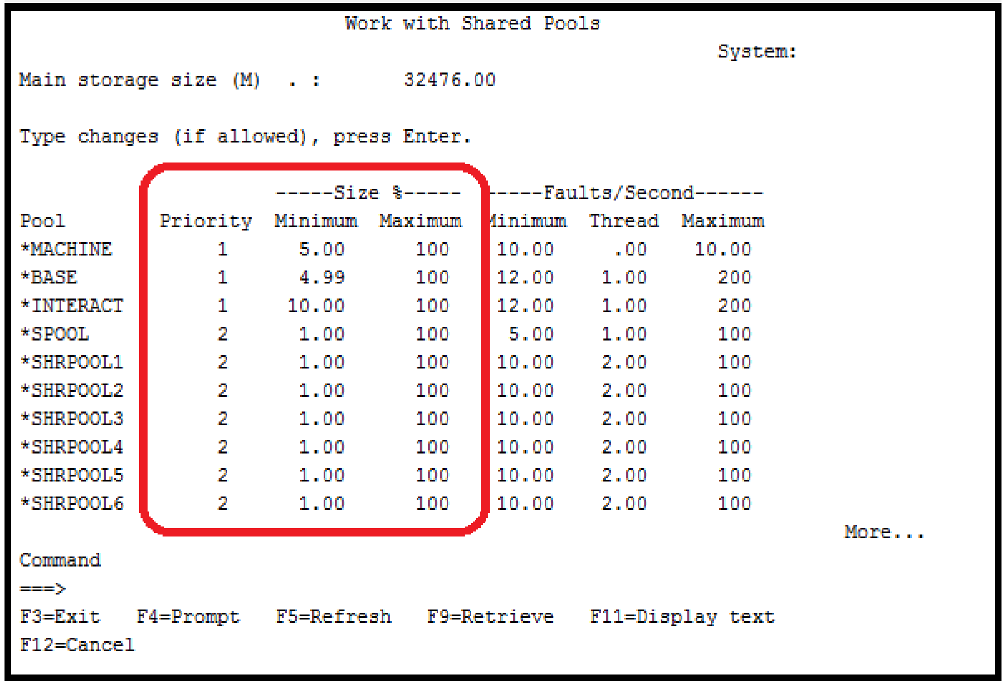
Page faults per second:

Minimum: 12.00

Additional minimum per thread: 1.00

Maximum: 200.00

WRKSHRPOOL:



```
                            Work with Shared Pools
                                                            System:
Main storage size (M)  . :        32476.00

Type changes (if allowed), press Enter.

                      -----Size %----------Faults/Second------
Pool        Priority  Minimum  Maximum  Minimum  Thread  Maximum
*MACHINE         1       5.00      100   10.00      .00   10.00
*BASE            1       4.99      100   12.00     1.00     200
*INTERACT        1      10.00      100   12.00     1.00     200
*SPOOL           2       1.00      100    5.00     1.00     100
*SHRPOOL1        2       1.00      100   10.00     2.00     100
*SHRPOOL2        2       1.00      100   10.00     2.00     100
*SHRPOOL3        2       1.00      100   10.00     2.00     100
*SHRPOOL4        2       1.00      100   10.00     2.00     100
*SHRPOOL5        2       1.00      100   10.00     2.00     100
*SHRPOOL6        2       1.00      100   10.00     2.00     100
                                                                       More...
Command
===>
F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F11=Display text
F12=Cancel
```

Tuning shared pools when QPFRADJ is ON

- Use WRKSHRPOOL command to tune shared memory pools
- Determine min/max size for critical pools
  - Monitor max active settings
- Set a range of pool priorities.
- **Java/WebSphere pools minimum size should always be enough to contain all JVMs in the pool**
- Set max on memory intensive pools to limit impact to other jobs on the system
- Large changes to the size of the memory pool can cause the query optimizer to rebuild access plans which can contribute to poor performance
- **If see pools sitting at min size a lot, consider decreasing its min size**
- **If see pools hitting max size, consider increasing max size.**
- **Keep total of minimum sizes < 70% of memory to allow QPFRADJ some flexibility**
- When adding memory, may need to adjust min/max
  - When memory is for a particular workload, rather than a general upgrade
  - Values are percentages of total

Refer to *The Performance Adjuster (QPFRADJ)* experience report on the IBM i Information Center.
http://publib.boulder.ibm.com/infocenter/iseries/v7r1m0/index.jsp?topic=%2Fexperience%2Fwork3abstract.htm

SETOBJACC to ‘pin’ objects in memory

- Allows selected objects (database file, index or program) to be "pinned" in memory to reduce I/O
    - Typically, define a private pool where no jobs run
    - Can use a shared pool. Set Min/Max to keep QPFRADJ from shrinking
- May increase memory requirements
- Protects objects from “demand paging” activity
- Objects loaded into memory very quickly by a single thread
- Typically used to improve performance of batch jobs
- Run a CLRPOOL command before loading objects in pool
- Re-run SETOBJACC periodically to pin changed pages (updates/inserts)
- Can load and purge objects programmatically during batch processing

SETOBJACC/CLRPOOL Command Technote:

http://www.ibm.com/support/docview.wss?uid=nas1dc0a2297bdaefddb86256d6c0069907f

DB2 KEEPINMEM to ‘pin’ objects in memory

- New in IBM i 7.1
- Can be used **with tables and indexes**
- Objects are brought into memory when first accessed by SQE queries
- Brought in asynchronously and using parallel I/O
  (vs. SETOBJACC which uses 1 thread and synch IO)
- Usage:
  - CHGPF FILE(library/table) KEEPINMEM(*YES)
  - CHGLF FILE(library/index) KEEPINMEM(*YES)
- The MEMORY_POOL_PREFERENCE parameter in the QAQQINI file determines which pool the objects will be held in unless the object is already in memory
- In IBM i 7.2. the KEEP IN MEMORY memory-preference support has been extended to the DDL statements: ALTER TABLE, CREATE INDEX, CREATE TABLE and DECLARE GLOBAL TEMPORARY TABLE

Memory Pool and Subsystem Creation Recommendations

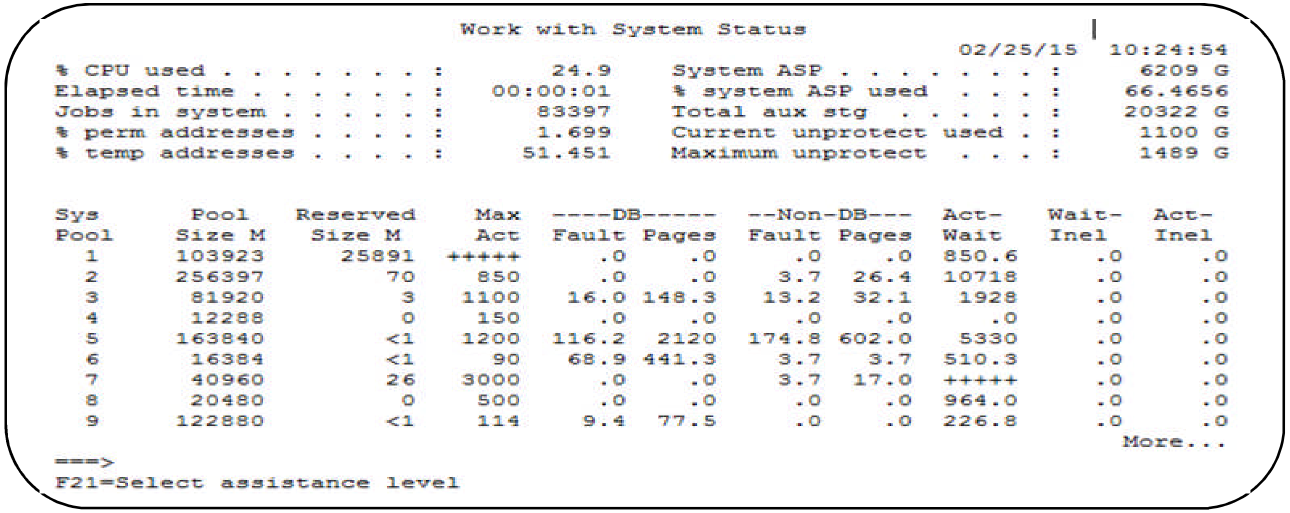
- Minimize user work in *BASE (in general)
- Put disparate workloads into separate pools
- Often want to run SQL in its own pool
- Heavy memory use when not critical
  - Often see software replication jobs in own pool with lots of faulting, but still keeping up
- Java / WebSphere should run in its own memory pool
  - These environments cannot tolerate faulting
- Controlling potentially inefficient work
  - QZDA jobs in own pool by IP for controlling ad-hoc queries
- Max active settings
  - If not using QPFRADJ set max active high enough to avoid transitions to ineligible
  - If running 6.1 don’t set too high for pools running SQL
- Settings related to QPFRADJ (min/max size, etc.)
- If see high priority page out task (SMPO0001), pool needs more memory.

Tools Used for Memory Analysis

- Real Time Tools
  - Green screen commands / IBM Navigator for i
    - WRKSYSSTS, WRKACTJOB
  - IBM Navigator for i Dashboard
  - IBM Navigator for i Monitors (New in IBM i 7.2)
- Collector based
  - Collection Services – System and Job level memory related wait times, fault rates, and pool sizes
  - Job Watcher – Object waited on, Call Stacks, and SQL statement
- GUI Tools
  - **Performance Data Investigator** ("PDI" Web based, part of IBM Navigator for i) – Collection Services, Job Watcher, Disk Watcher, limited PEX
  - **IBM iDoctor for i** (Windows based) – Collection Services, Job Watcher, Disk Watcher, PEX Analyzer
  - System i Navigator

WRKSYSSTS – Faulting by Memory pool

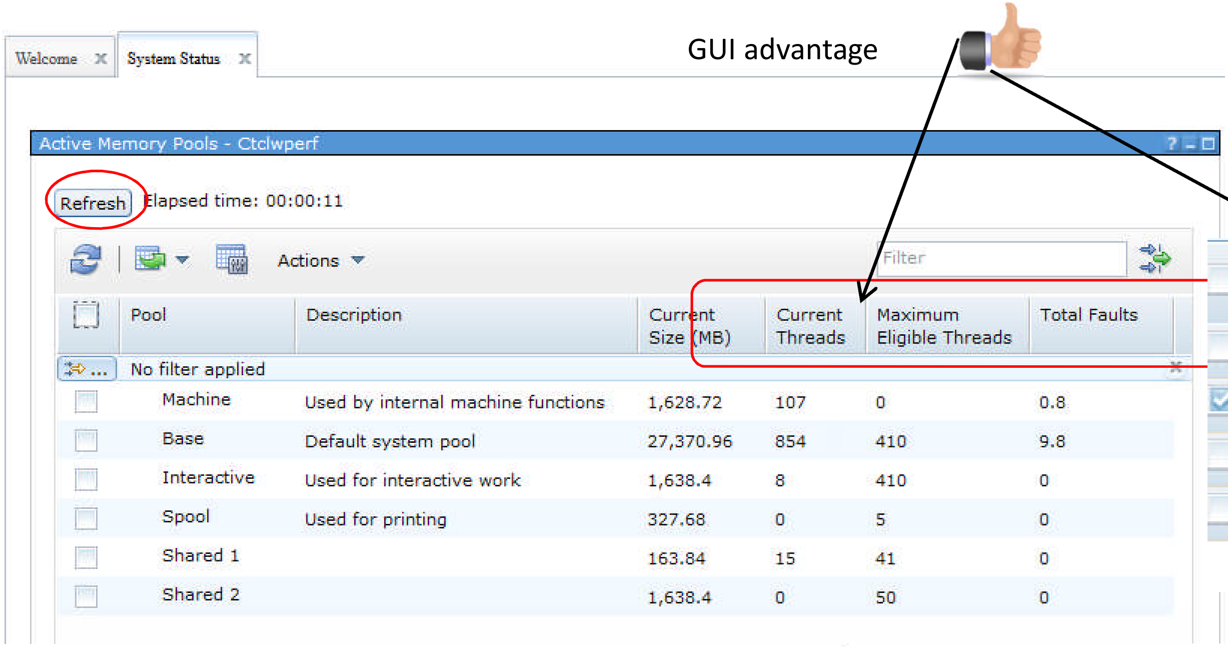
- Can be invoked from green screen or web based IBM navigator for i
- Faults per second column doesn’t show real fault wait times
- Monitor the max active and ineligibles



```
                           Work with System Status
                                                            02/25/15  10:24:54
% CPU used . . . . . . . :       24.9    System ASP . . . . . . . :     6209 G
Elapsed time . . . . . . :   00:00:01    % system ASP used  . . . :    66.4656
Jobs in system . . . . . :      83397    Total aux stg  . . . . . :    20322 G
% perm addresses . . . . :      1.699    Current unprotect used . :     1100 G
% temp addresses . . . . :     51.451    Maximum unprotect  . . . :     1489 G


Sys      Pool   Reserved    Max  ----DB-----  --Non-DB---  Act-   Wait-  Act-
Pool    Size M   Size M     Act  Fault Pages  Fault Pages  Wait   Inel   Inel
  1     103923     25891  +++++     .0    .0     .0    .0  850.6     .0     .0
  2     256397        70    850     .0    .0    3.7  26.4  10718     .0     .0
  3      81920         3   1100   16.0 148.3   13.2  32.1   1928     .0     .0
  4      12288         0    150     .0    .0     .0    .0     .0     .0     .0
  5     163840        <1   1200  116.2  2120  174.8 602.0   5330     .0     .0
  6      16384        <1     90   68.9 441.3    3.7   3.7  510.3     .0     .0
  7      40960        26   3000     .0    .0    3.7  17.0  +++++     .0     .0
  8      20480         0    500     .0    .0     .0    .0  964.0     .0     .0
  9     122880        <1    114    9.4  77.5     .0    .0  226.8     .0     .0
                                                                       More...
===>
F21=Select assistance level
```

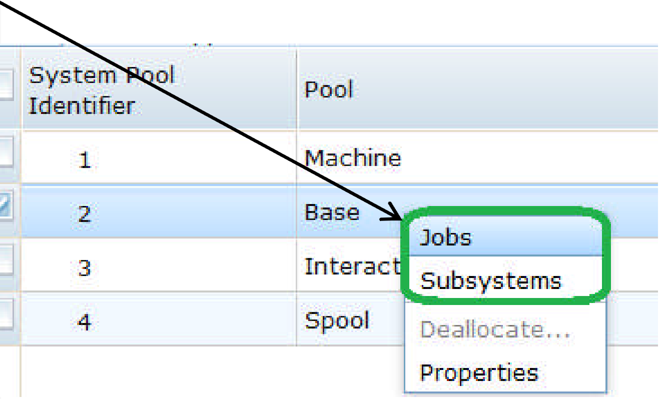
Memory Pools (WRKSYSSTS)

GUI advantage

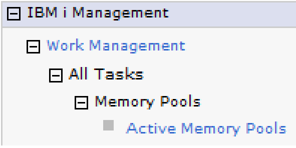


Elapsed time: 00:00:11

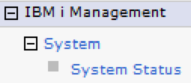
□	Pool	Description	Current Size (MB)	Current Threads	Maximum Eligible Threads	Total Faults
□	Machine	Used by internal machine functions	1,628.72	107	0	0.8
□	Base	Default system pool	27,370.96	854	410	9.8
□	Interactive	Used for interactive work	1,638.4	8	410	0
□	Spool	Used for printing	327.68	0	5	0
□	Shared 1		163.84	15	41	0
□	Shared 2		1,638.4	0	50	0

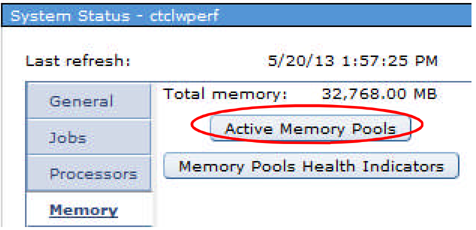


	System Pool Identifier	Pool
□	1	Machine
☑	2	Base
□	3	Interact
□	4	Spool



or



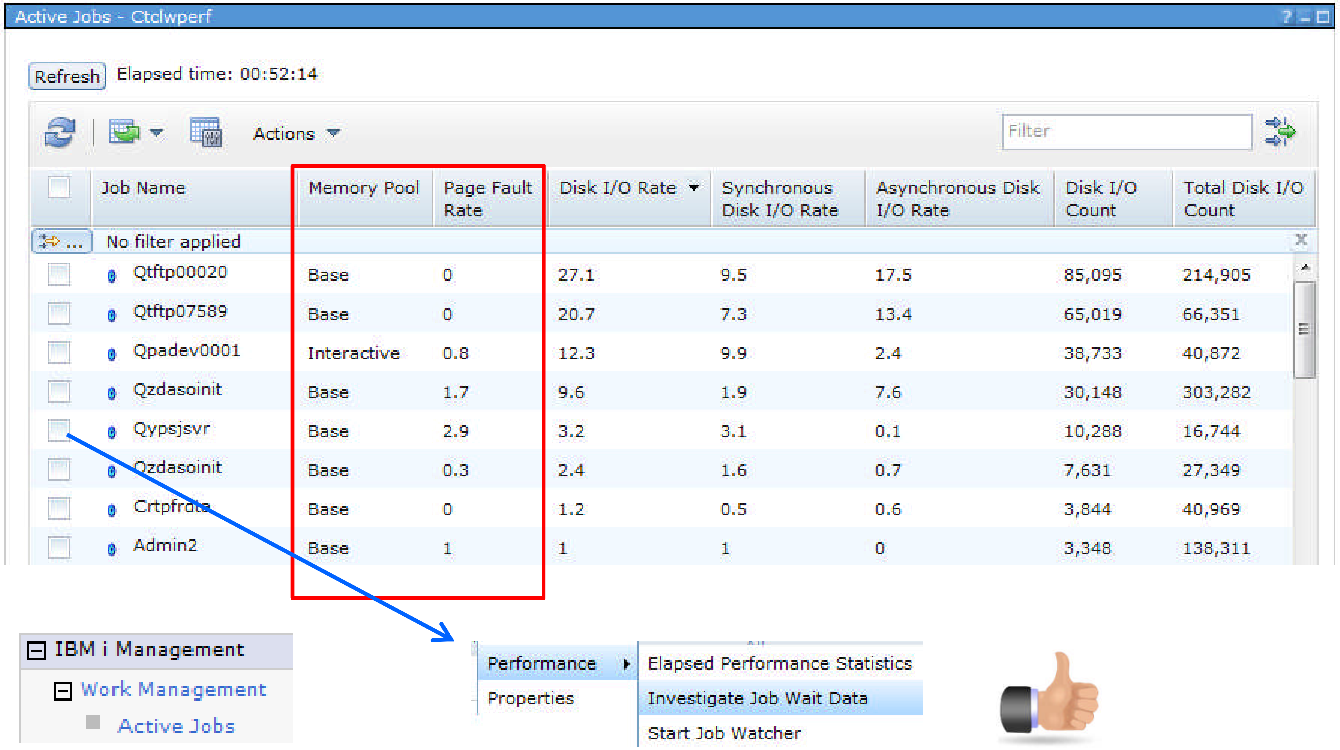


Last refresh: 5/20/13 1:57:25 PM

Total memory: 32,768.00 MB

WRKACTJOB – Page faulting by job

The Page fault rate column is available of the IBM Navigator for i interface



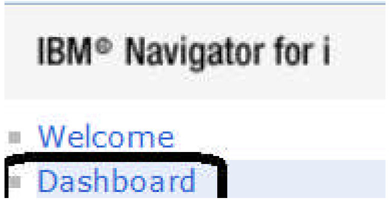
Active Jobs - Ctclwperf

Refresh Elapsed time: 00:52:14

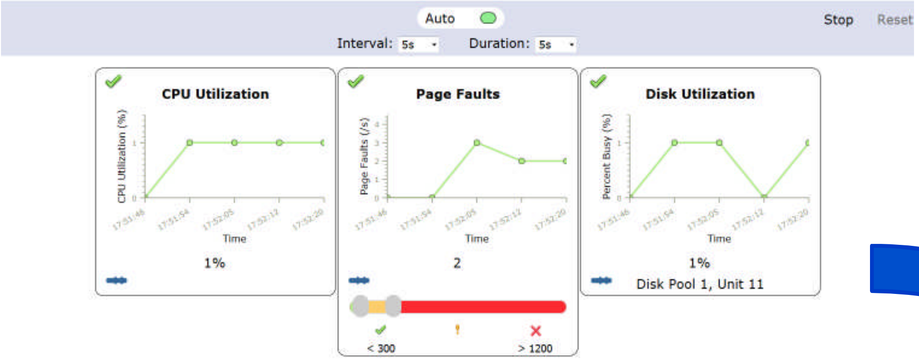
No filter applied

Job Name	Memory Pool	Page Fault Rate	Disk I/O Rate	Synchronous Disk I/O Rate	Asynchronous Disk I/O Rate	Disk I/O Count	Total Disk I/O Count
Qtftp00020	Base	0	27.1	9.5	17.5	85,095	214,905
Qtftp07589	Base	0	20.7	7.3	13.4	65,019	66,351
Qpadev0001	Interactive	0.8	12.3	9.9	2.4	38,733	40,872
Qzdasoinit	Base	1.7	9.6	1.9	7.6	30,148	303,282
Qypsjsvr	Base	2.9	3.2	3.1	0.1	10,288	16,744
Qzdasoinit	Base	0.3	2.4	1.6	0.7	7,631	27,349
Crtpfrdta	Base	0	1.2	0.5	0.6	3,844	40,969
Admin2	Base	1	1	1	0	3,348	138,311

Navigator Dashboard



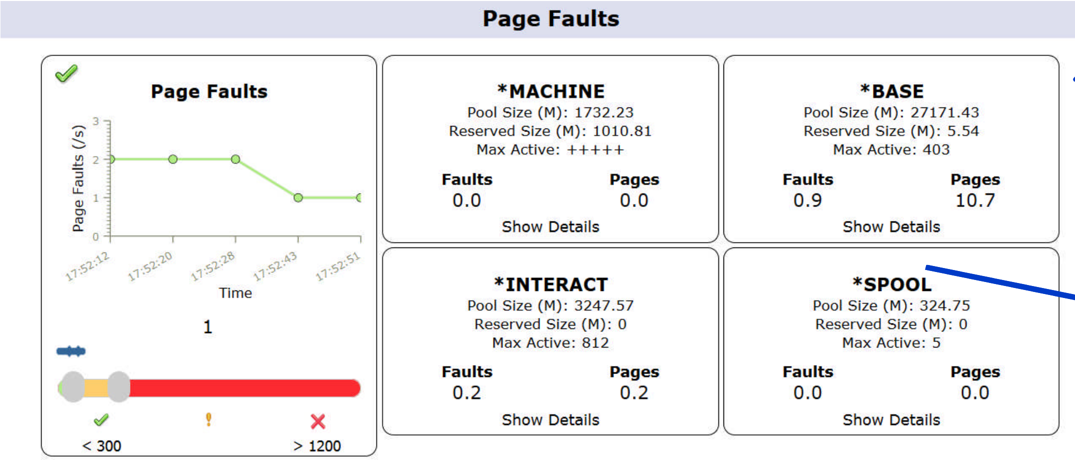
IBM® Navigator for i



Auto

Interval: 5s

Duration: 5s



Page Faults

***MACHINE**
Pool Size (M): 1732.23
Reserved Size (M): 1010.81
Max Active: +++++

Faults	Pages
0.0	0.0

Show Details

***BASE**
Pool Size (M): 27171.43
Reserved Size (M): 5.54
Max Active: 403

Faults	Pages
0.9	10.7

Show Details

***INTERACT**
Pool Size (M): 3247.57
Reserved Size (M): 0
Max Active: 812

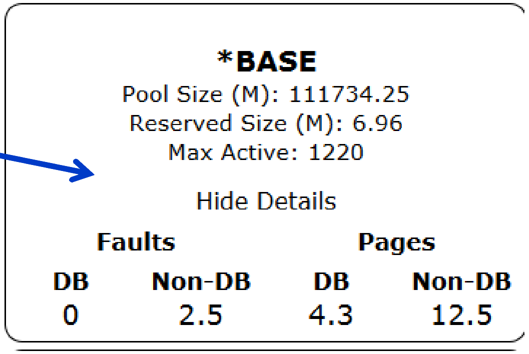
Faults	Pages
0.2	0.2

Show Details

***SPOOL**
Pool Size (M): 324.75
Reserved Size (M): 0
Max Active: 5

Faults	Pages
0.0	0.0

Show Details



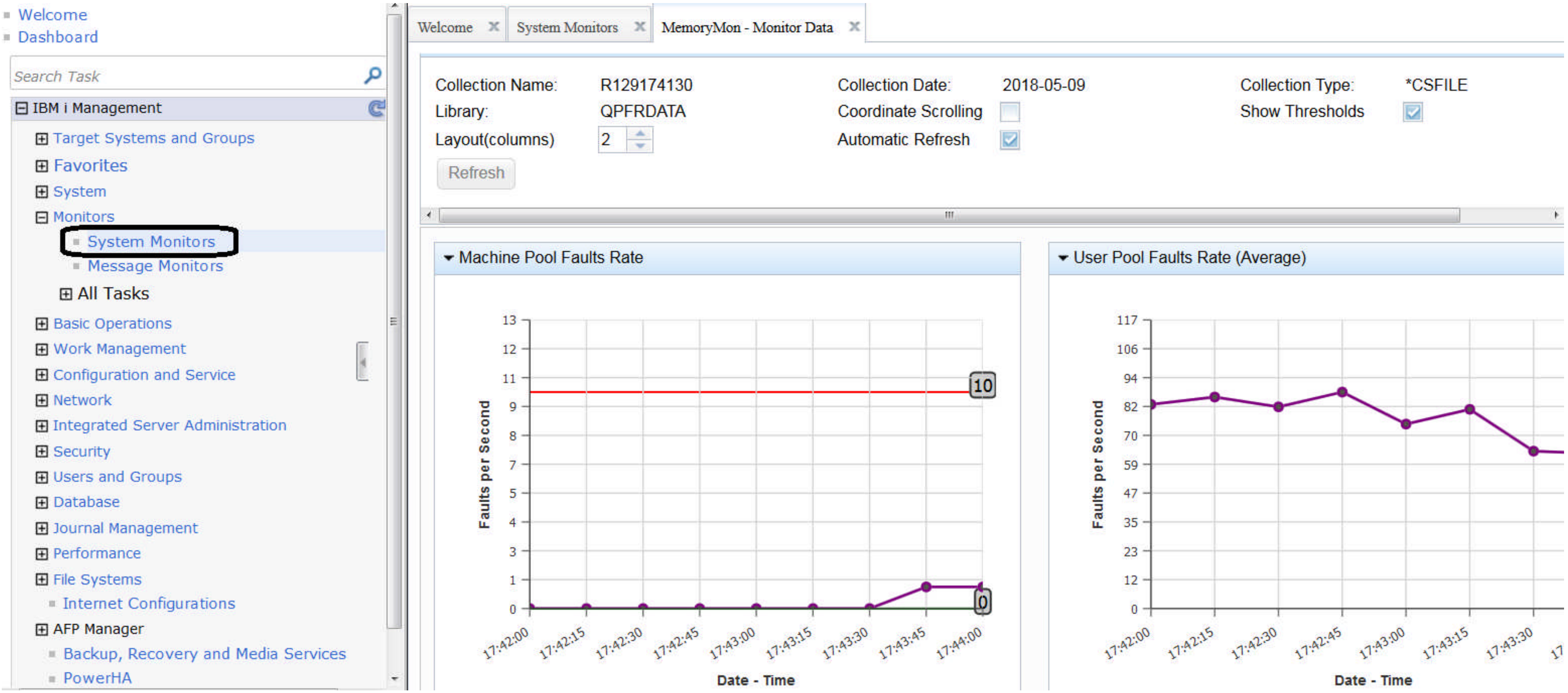
***BASE**
Pool Size (M): 111734.25
Reserved Size (M): 6.96
Max Active: 1220

Hide Details

Faults		Pages	
DB	Non-DB	DB	Non-DB
0	2.5	4.3	12.5

Also available in iAccess Mobile: http://system.name:2001/iamobile

Navigator for i System Monitors (7.2)



Collection Name: R129174130

Library: QPFRDATA

Layout(columns) 2

Collection Date: 2018-05-09

Coordinate Scrolling □

Automatic Refresh ☑

Collection Type: *CSFILE

Show Thresholds ☑

IBM Graphical Analysis Tools

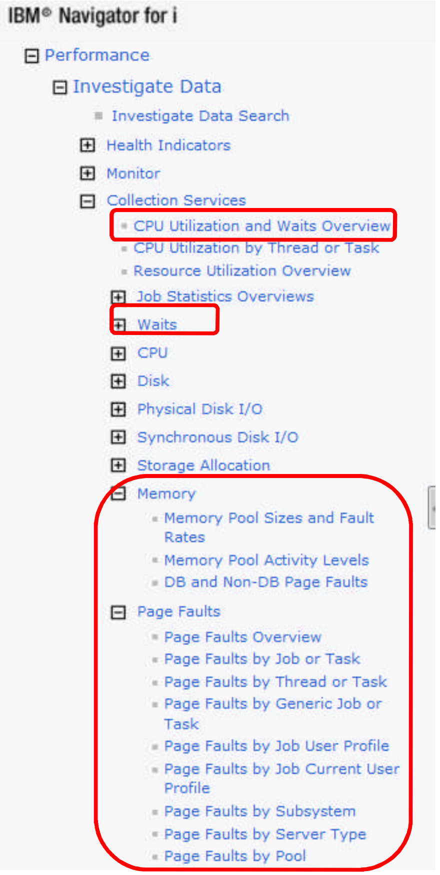
IBM provides two powerful tools to aid in making your analysis **more efficient and productive:**

- **Performance Data Investigator**
- **IBM iDoctor for IBM i**

Both solutions support data analysis (varying degrees) for the 4 collectors:

- Collection Services
- Job Watcher
- Disk Watcher
- Performance Explorer (PEX)

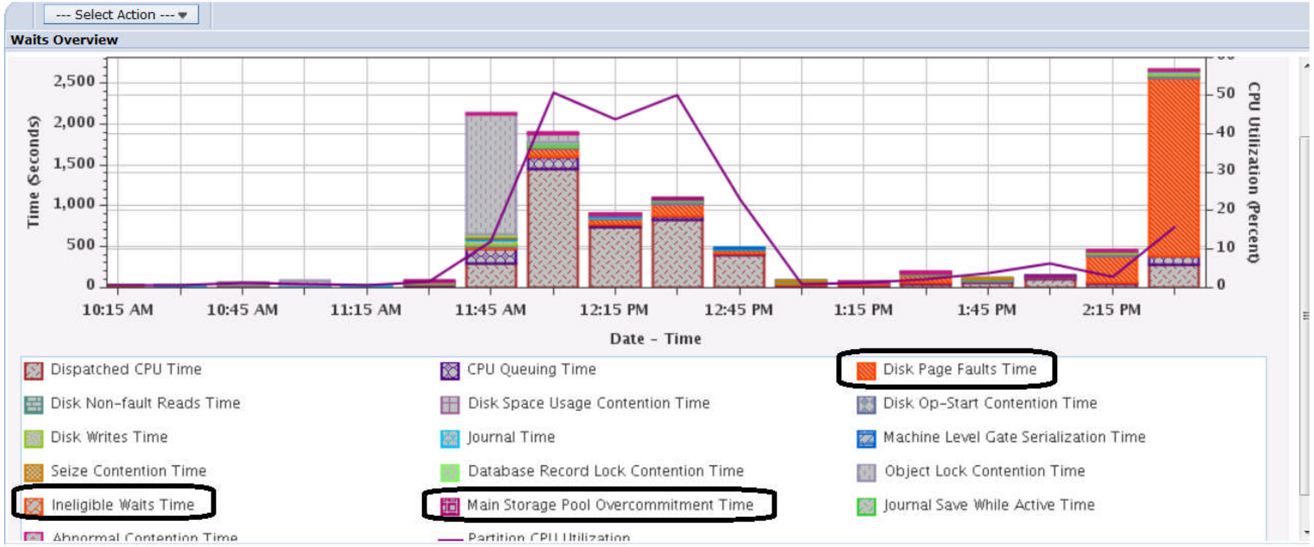
Performance Data Investigator (PDI)



IBM® Navigator for i

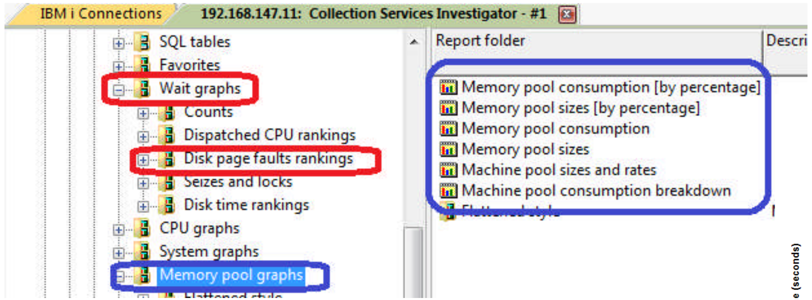
- ⊟ Performance
  - ⊟ Investigate Data
    - Investigate Data Search
    - ⊞ Health Indicators
    - ⊞ Monitor
    - ⊟ Collection Services
      - CPU Utilization and Waits Overview
      - CPU Utilization by Thread or Task
      - Resource Utilization Overview
      - ⊞ Job Statistics Overviews
      - ⊞ Waits
      - ⊞ CPU
      - ⊞ Disk
      - ⊞ Physical Disk I/O
      - ⊞ Synchronous Disk I/O
      - ⊞ Storage Allocation
      - ⊟ Memory
        - Memory Pool Sizes and Fault Rates
        - Memory Pool Activity Levels
        - DB and Non-DB Page Faults
      - ⊟ Page Faults
        - Page Faults Overview
        - Page Faults by Job or Task
        - Page Faults by Thread or Task
        - Page Faults by Generic Job or Task
        - Page Faults by Job User Profile
        - Page Faults by Job Current User Profile
        - Page Faults by Subsystem
        - Page Faults by Server Type
        - Page Faults by Pool

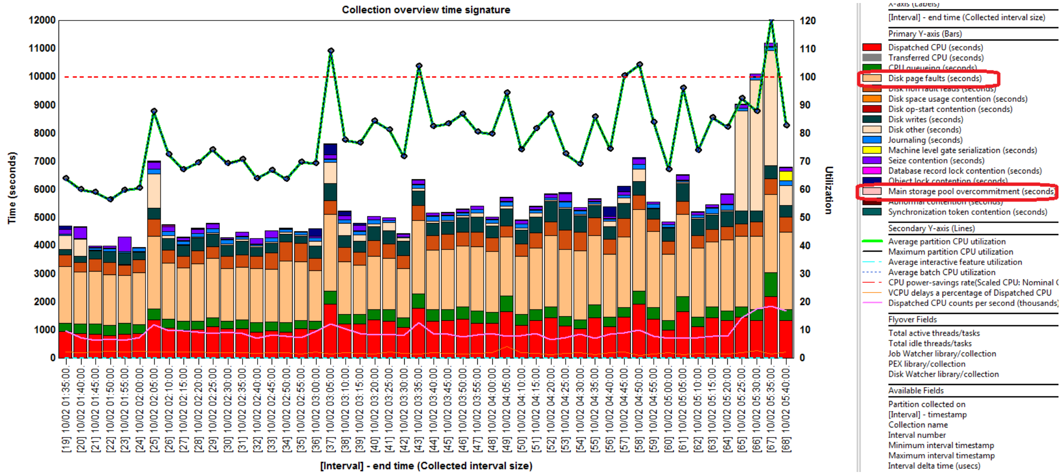
- Browser (web) based solution
- Integrated as part of IBM i OS
- Included in IBM Navigator for i



IBM iDoctor for IBM i

- Microsoft Windows based client
- Service/Support offering
- Deep Job Watcher and PEX analysis capability





Graphical Analysis Tools

- You have two graphical interfaces for performance data analysis…
  - Which is right for you?

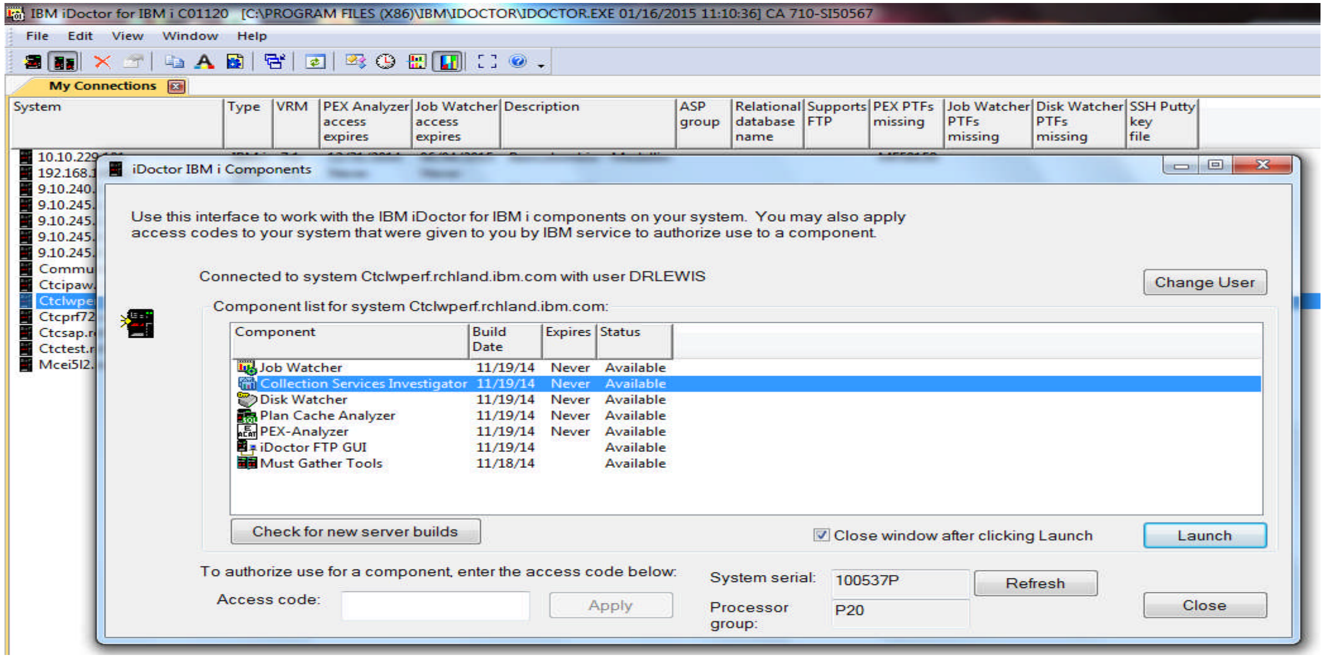
Feature	iDoctor	PDI
Interface	**Windows client**	**Browser**
Wait Analysis	Yes	Yes
Collection Services	Yes	Yes
Job Watcher	Yes (In-depth)	Yes
Disk Watcher	Yes	Yes
Performance Explorer	Yes (In-depth)	Profile collections only
Level of analysis provided	**Deep**	**Basic to Medium**
Job Watcher Monitors (Built –in)	Yes	No
User Defined graphs and queries	Yes	Yes
Update Frequency	Quarterly	Twice Yearly
Support	Email idoctor@us.ibm.com	Standard SWMA
Chargeable	**Yearly license for each component (by serial number)** **▪Job Watcher** **–Includes Job Watcher, Collection Services Investigator, and Disk Watcher** **▪PEX Analyzer**	**▪Collection Services & Health Indicators at no additional charge with i** **▪Disk Watcher, Database, and Performance Explorer included with base PT1 (Performance Tools LPP) product – Option 1 Manager feature** **▪Job Watcher is an additional option of PT1 and has an additional charge - Option 3 Job Watcher**
DS8K graphs & VIOS Investigator	Yes	No
Multinational language support	No	Yes

Collection Services - Memory Related Data

- What will be analyzed using CS?
  - Page fault wait time for system, job, thread, memory pool etc.
    - At a job level, can be useful to know if the wait time occurred primarily during start-up time, or consistently while active
  - Rate of faulting in each memory pool
  - Memory pool size/config of the jobs with high page faults

Memory Analysis - CSI

- The first step is to identify page fault wait times in a collection services member



Use this interface to work with the IBM iDoctor for IBM i components on your system. You may also apply access codes to your system that were given to you by IBM service to authorize use to a component.

Connected to system Ctclwperf.rchland.ibm.com with user DRLEWIS

Component list for system Ctclwperf.rchland.ibm.com:

Component	Build Date	Expires	Status
Job Watcher	11/19/14	Never	Available
Collection Services Investigator	11/19/14	Never	Available
Disk Watcher	11/19/14	Never	Available
Plan Cache Analyzer	11/19/14	Never	Available
PEX-Analyzer	11/19/14	Never	Available
iDoctor FTP GUI	11/19/14		Available
Must Gather Tools	11/18/14		Available

☑ Close window after clicking Launch

To authorize use for a component, enter the access code below:

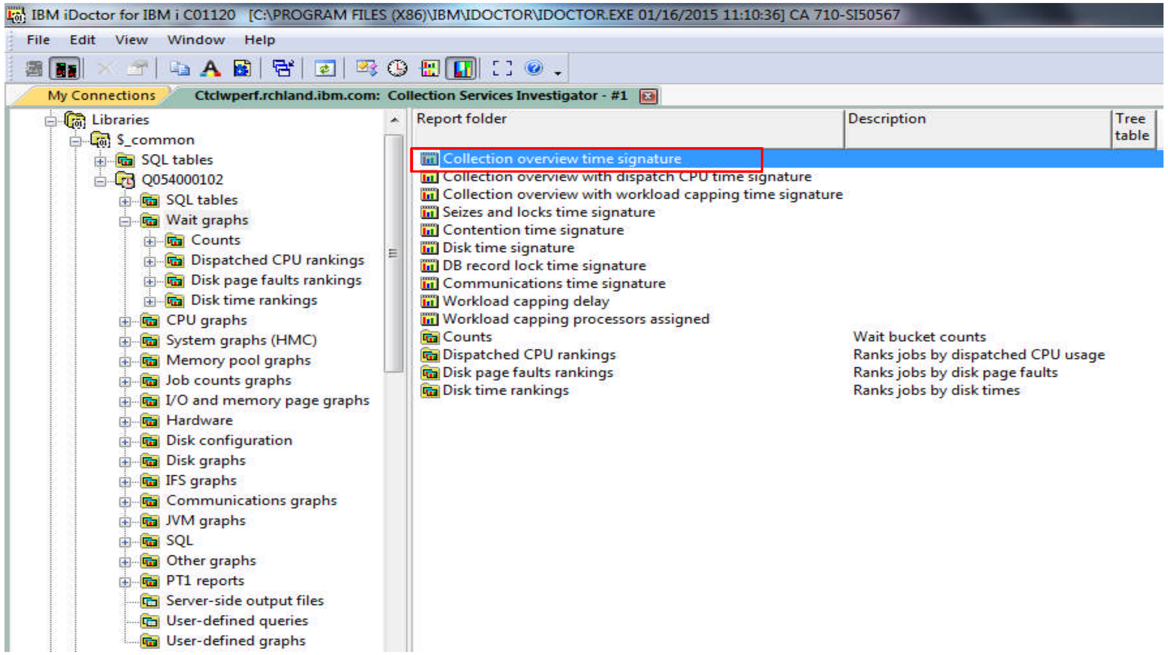
Access code:

System serial: 100537P

Processor group: P20

Memory Analysis - CSI

- A good place to start is the Collection overview time signature.
- We want to see the relative amount of time being spent waiting on disk faults.

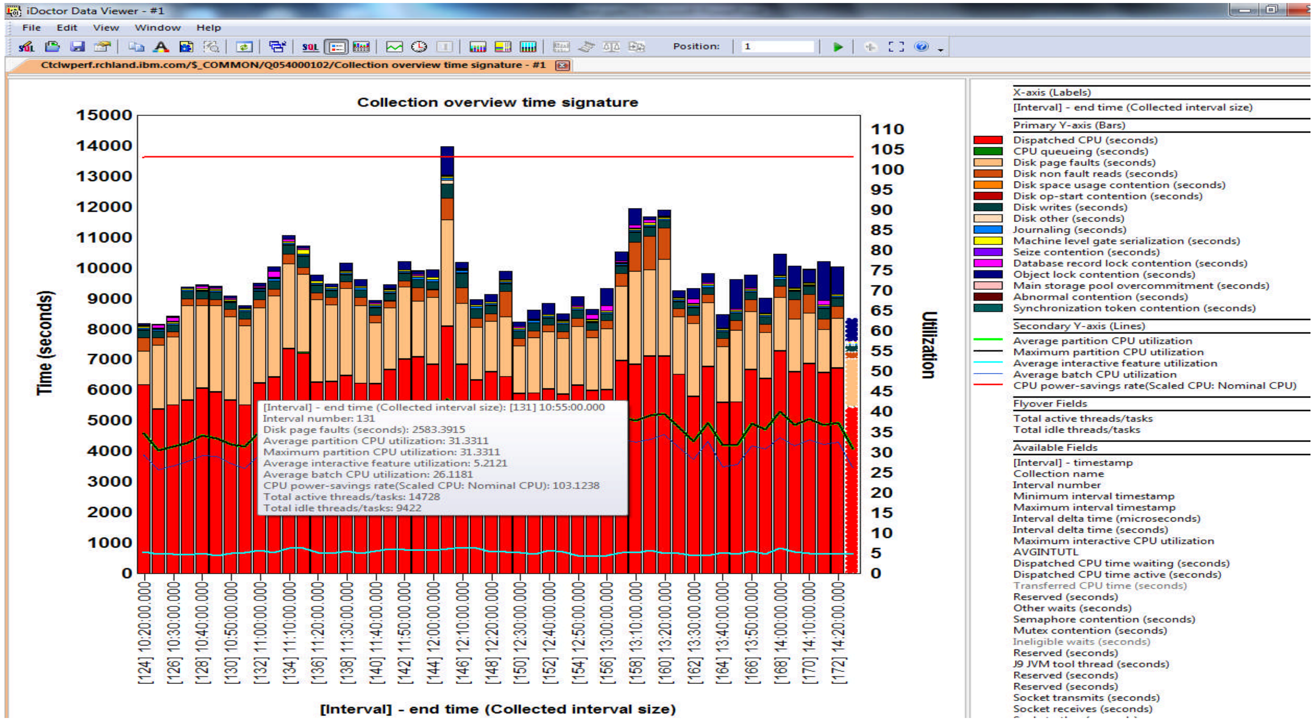

IBM iDoctor for IBM i C01120 [C:\PROGRAM FILES (X86)\IBM\IDOCTOR\IDOCTOR.EXE 01/16/2015 11:10:36] CA 710-SI50567
File
Edit
View
Window
Help
My Connections
Ctclwperf.rchland.ibm.com: Collection Services Investigator - #1

- Libraries
  - $_common
    - SQL tables
    - Q054000102
      - SQL tables
      - Wait graphs
        - Counts
        - Dispatched CPU rankings
        - Disk page faults rankings
        - Disk time rankings
      - CPU graphs
      - System graphs (HMC)
      - Memory pool graphs
      - Job counts graphs
      - I/O and memory page graphs
      - Hardware
      - Disk configuration
      - Disk graphs
      - IFS graphs
      - Communications graphs
      - JVM graphs
      - SQL
      - Other graphs
      - PT1 reports
      - Server-side output files
      - User-defined queries
      - User-defined graphs

Report folder	Description	Tree table
Collection overview time signature		
Collection overview with dispatch CPU time signature		
Collection overview with workload capping time signature		
Seizes and locks time signature		
Contention time signature		
Disk time signature		
DB record lock time signature		
Communications time signature		
Workload capping delay		
Workload capping processors assigned		
Counts	Wait bucket counts	
Dispatched CPU rankings	Ranks jobs by dispatched CPU usage	
Disk page faults rankings	Ranks jobs by disk page faults	
Disk time rankings	Ranks jobs by disk times	

Memory Analysis - CSI

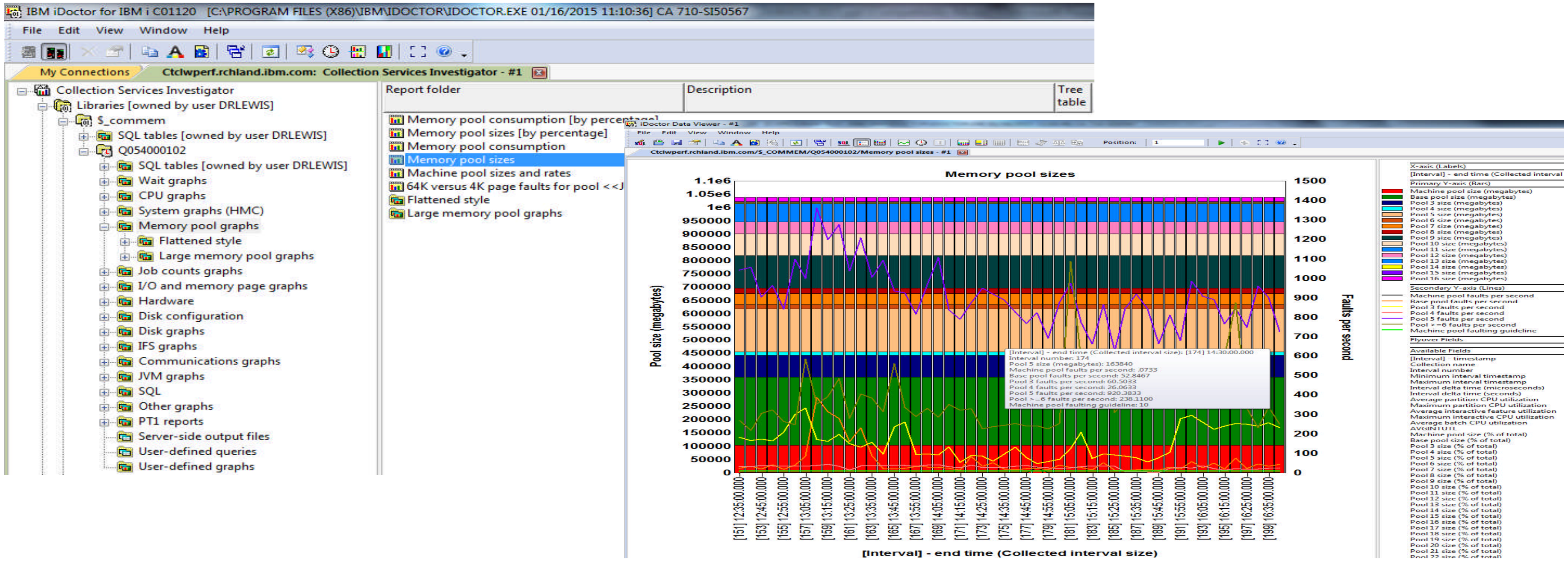
- Identify intervals with high page fault wait times (tan)
- Wait times could be skewed some by large collection services intervals


iDoctor Data Viewer - #1
File
Edit
View
Window
Help
Position:
1
Ctclwperf.rchland.ibm.com/$_COMMON/Q054000102/Collection overview time signature - #1
Collection overview time signature
Time (seconds)
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
Utilization
110
105
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
[124] 10:20:00.000
[126] 10:30:00.000
[128] 10:40:00.000
[130] 10:50:00.000
[132] 11:00:00.000
[134] 11:10:00.000
[136] 11:20:00.000
[138] 11:30:00.000
[140] 11:40:00.000
[142] 11:50:00.000
[144] 12:00:00.000
[146] 12:10:00.000
[148] 12:20:00.000
[150] 12:30:00.000
[152] 12:40:00.000
[154] 12:50:00.000
[156] 13:00:00.000
[158] 13:10:00.000
[160] 13:20:00.000
[162] 13:30:00.000
[164] 13:40:00.000
[166] 13:50:00.000
[168] 14:00:00.000
[170] 14:10:00.000
[172] 14:20:00.000
[Interval] - end time (Collected interval size)
X-axis (Labels)
[Interval] - end time (Collected interval size)
Primary Y-axis (Bars)
🟥 Dispatched CPU (seconds)
🟩 CPU queueing (seconds)
🟧 Disk page faults (seconds)
🟫 Disk non fault reads (seconds)
🟧 Disk space usage contention (seconds)
🟥 Disk op-start contention (seconds)
■ Disk writes (seconds)
□ Disk other (seconds)
🟦 Journaling (seconds)
🟨 Machine level gate serialization (seconds)
🟪 Seize contention (seconds)
🟪 Database record lock contention (seconds)
🟦 Object lock contention (seconds)
□ Main storage pool overcommitment (seconds)
🟫 Abnormal contention (seconds)
■ Synchronization token contention (seconds)
Secondary Y-axis (Lines)
— Average partition CPU utilization
— Maximum partition CPU utilization
— Average interactive feature utilization
— Average batch CPU utilization
— CPU power-savings rate(Scaled CPU: Nominal CPU)
Flyover Fields
Total active threads/tasks
Total idle threads/tasks
Available Fields
[Interval] - timestamp
Collection name
Interval number
Minimum interval timestamp
Maximum interval timestamp
Interval delta time (microseconds)
Interval delta time (seconds)
Maximum interactive CPU utilization
AVGINTUTL
Dispatched CPU time waiting (seconds)
Dispatched CPU time active (seconds)
Transferred CPU time (seconds)
Reserved (seconds)
Other waits (seconds)
Semaphore contention (seconds)
Mutex contention (seconds)
Ineligible waits (seconds)
Reserved (seconds)
J9 JVM tool thread (seconds)
Reserved (seconds)
Reserved (seconds)
Socket transmits (seconds)
Socket receives (seconds)

[Interval] - end time (Collected interval size): [131] 10:55:00.000
Interval number: 131
Disk page faults (seconds): 2583.3915
Average partition CPU utilization: 31.3311
Maximum partition CPU utilization: 31.3311
Average interactive feature utilization: 5.2121
Average batch CPU utilization: 26.1181
CPU power-savings rate(Scaled CPU: Nominal CPU): 103.1238
Total active threads/tasks: 14728
Total idle threads/tasks: 9422

Memory Analysis – Memory Pool Graphs in CSI

- No change in pool size indicates that performance adjuster system value (QPFRADJ) was not on during this collection



IBM iDoctor for IBM i C01120 [C:\PROGRAM FILES (X86)\IBM\IDOCTOR\IDOCTOR.EXE 01/16/2015 11:10:36] CA 710-SI50567
File
Edit
View
Window
Help
My Connections
Ctclwperf.rchland.ibm.com: Collection Services Investigator - #1

- Collection Services Investigator
  - Libraries [owned by user DRLEWIS]
    - $_commem
      - SQL tables [owned by user DRLEWIS]
      - Q054000102
        - SQL tables [owned by user DRLEWIS]
        - Wait graphs
        - CPU graphs
        - System graphs (HMC)
        - Memory pool graphs
          - Flattened style
          - Large memory pool graphs
        - Job counts graphs
        - I/O and memory page graphs
        - Hardware
        - Disk configuration
        - Disk graphs
        - IFS graphs
        - Communications graphs
        - JVM graphs
        - SQL
        - Other graphs
        - PT1 reports
        - Server-side output files
        - User-defined queries
        - User-defined graphs

Report folder	Description	Tree table
Memory pool consumption [by percentage]		
Memory pool sizes [by percentage]		
Memory pool consumption		
Memory pool sizes		
Machine pool sizes and rates		
64K versus 4K page faults for pool <<J		
Flattened style		
Large memory pool graphs		

[Interval] - end time (Collected interval size): [174] 14:30:00.000
Interval number: 174
Pool 5 size (megabytes): 163840
Machine pool faults per second: .0733
Base pool faults per second: 52.8467
Pool 3 faults per second: 60.5033
Pool 4 faults per second: 26.0633
Pool 5 faults per second: 920.3833
Pool >=6 faults per second: 238.1100
Machine pool faulting guideline: 10

X-axis (Labels)
[Interval] - end time (Collected interval

Primary Y-axis (Bars)
🟥 Machine pool size (megabytes)
🟩 Base pool size (megabytes)
■ Pool 3 size (megabytes)
■ Pool 4 size (megabytes)
■ Pool 5 size (megabytes)
■ Pool 6 size (megabytes)
🟧 Pool 7 size (megabytes)
■ Pool 8 size (megabytes)
■ Pool 9 size (megabytes)
■ Pool 10 size (megabytes)
🟦 Pool 11 size (megabytes)
■ Pool 12 size (megabytes)
🟦 Pool 13 size (megabytes)
🟨 Pool 14 size (megabytes)
🟪 Pool 15 size (megabytes)
■ Pool 16 size (megabytes)

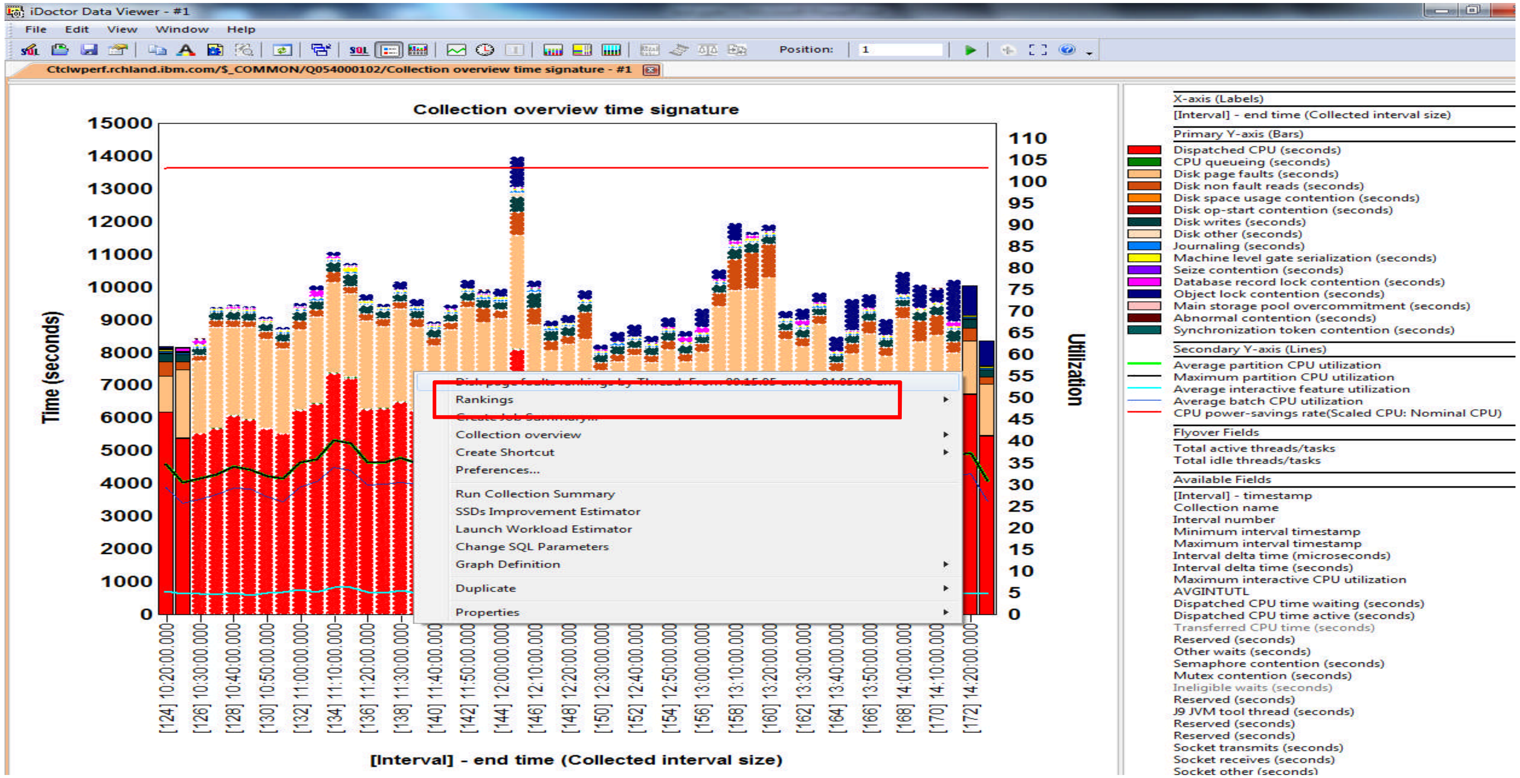
Secondary Y-axis (Lines)
— Machine pool faults per second
— Base pool faults per second
— Pool 3 faults per second
— Pool 4 faults per second
— Pool 5 faults per second
— Pool >=6 faults per second
— Machine pool faulting guideline

Flyover Fields

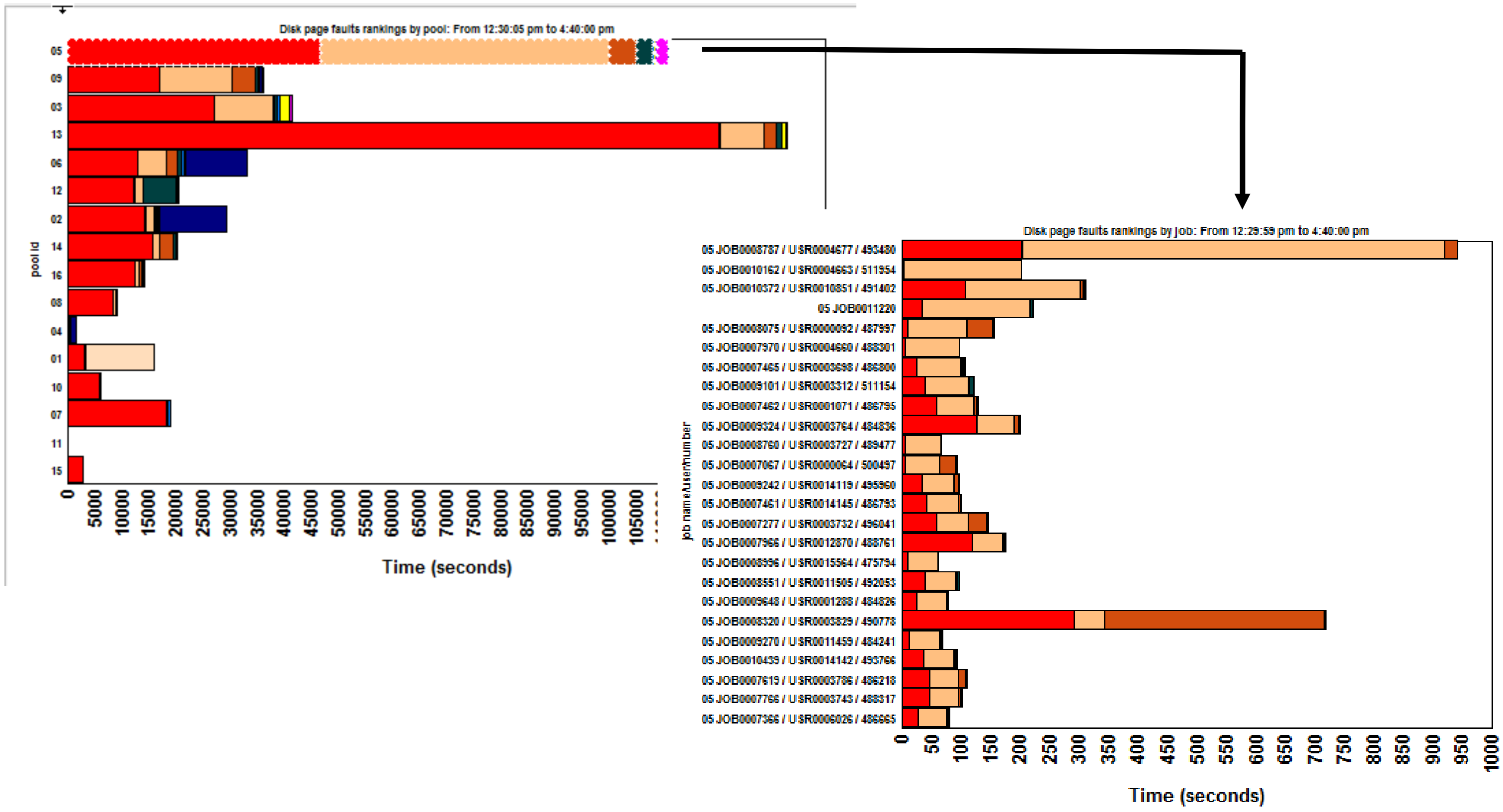
Available Fields
[Interval] - timestamp
Collection name
Interval number
Minimum interval timestamp
Maximum interval timestamp
Interval delta time (microseconds)
Interval delta time (seconds)
Average partition CPU utilization
Maximum partition CPU utilization
Average interactive feature utilization
Maximum interactive CPU utilization
Average batch CPU utilization
AVGINTUTL
Machine pool size (% of total)
Base pool size (% of total)
Pool 3 size (% of total)
Pool 4 size (% of total)
Pool 5 size (% of total)
Pool 6 size (% of total)
Pool 7 size (% of total)
Pool 8 size (% of total)
Pool 9 size (% of total)
Pool 10 size (% of total)
Pool 11 size (% of total)
Pool 12 size (% of total)
Pool 13 size (% of total)
Pool 14 size (% of total)
Pool 15 size (% of total)
Pool 16 size (% of total)
Pool 17 size (% of total)
Pool 18 size (% of total)
Pool 19 size (% of total)
Pool 20 size (% of total)
Pool 21 size (% of total)
Pool 22 size (% of total)

Memory Analysis - CSI

- Highlight intervals with high page fault waits and right click to rank by memory pool

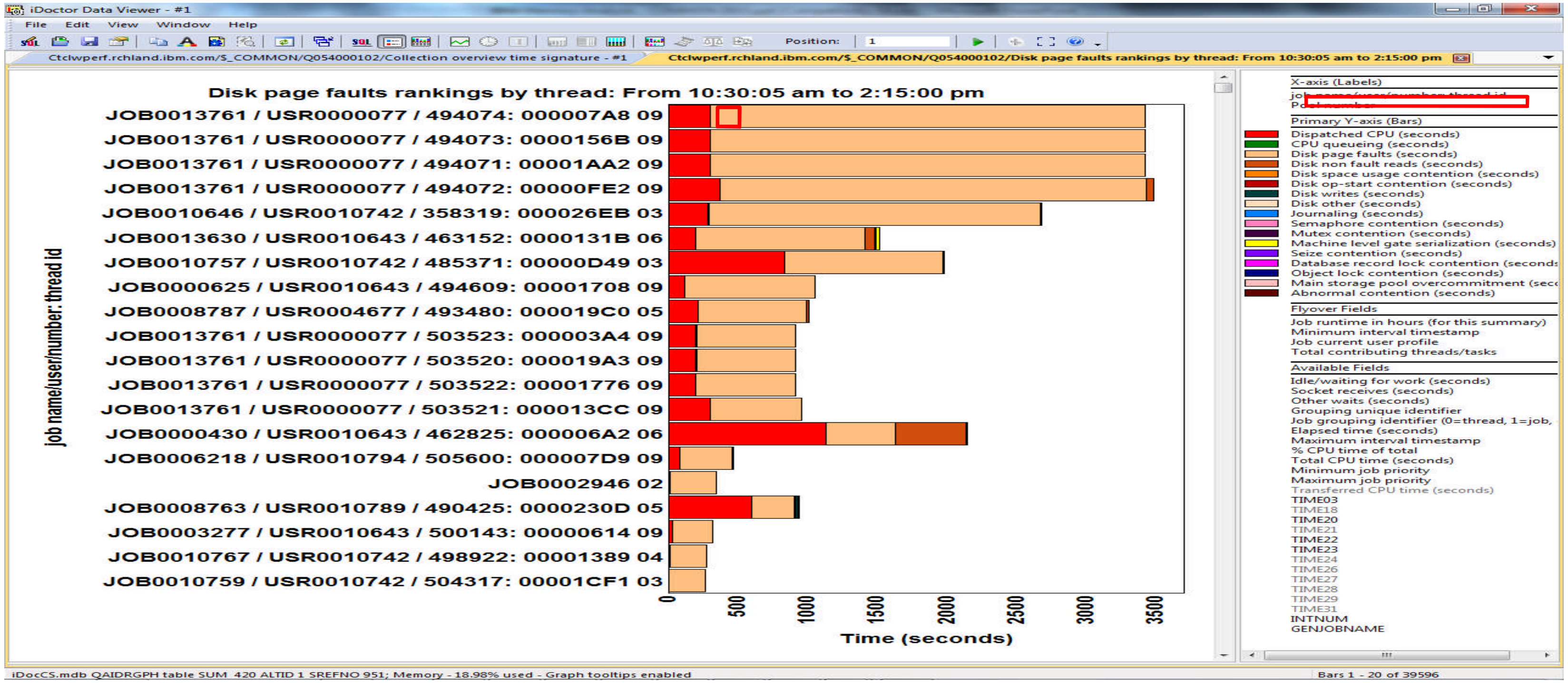

iDoctor Data Viewer - #1
File
Edit
View
Window
Help
Position:
1
Ctclwperf.rchland.ibm.com/$_COMMON/Q054000102/Collection overview time signature - #1
Collection overview time signature
Time (seconds)
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
110
105
100
95
90
85
80
75
70
65
60
55
50
45
40
35
30
25
20
15
10
5
0
Utilization
[124] 10:20:00.000
[126] 10:30:00.000
[128] 10:40:00.000
[130] 10:50:00.000
[132] 11:00:00.000
[134] 11:10:00.000
[136] 11:20:00.000
[138] 11:30:00.000
[140] 11:40:00.000
[142] 11:50:00.000
[144] 12:00:00.000
[146] 12:10:00.000
[148] 12:20:00.000
[150] 12:30:00.000
[152] 12:40:00.000
[154] 12:50:00.000
[156] 13:00:00.000
[158] 13:10:00.000
[160] 13:20:00.000
[162] 13:30:00.000
[164] 13:40:00.000
[166] 13:50:00.000
[168] 14:00:00.000
[170] 14:10:00.000
[172] 14:20:00.000
[Interval] - end time (Collected interval size)
Disk page faults rankings by Thread: From 09:15:05 am to 04:05:00 am
Rankings
Create Job Summary...
Collection overview
Create Shortcut
Preferences...
Run Collection Summary
SSDs Improvement Estimator
Launch Workload Estimator
Change SQL Parameters
Graph Definition
Duplicate
Properties
X-axis (Labels)
[Interval] - end time (Collected interval size)
Primary Y-axis (Bars)
Dispatched CPU (seconds)
CPU queueing (seconds)
Disk page faults (seconds)
Disk non fault reads (seconds)
Disk space usage contention (seconds)
Disk op-start contention (seconds)
Disk writes (seconds)
Disk other (seconds)
Journaling (seconds)
Machine level gate serialization (seconds)
Seize contention (seconds)
Database record lock contention (seconds)
Object lock contention (seconds)
Main storage pool overcommitment (seconds)
Abnormal contention (seconds)
Synchronization token contention (seconds)
Secondary Y-axis (Lines)
Average partition CPU utilization
Maximum partition CPU utilization
Average interactive feature utilization
Average batch CPU utilization
CPU power-savings rate(Scaled CPU: Nominal CPU)
Flyover Fields
Total active threads/tasks
Total idle threads/tasks
Available Fields
[Interval] - timestamp
Collection name
Interval number
Minimum interval timestamp
Maximum interval timestamp
Interval delta time (microseconds)
Interval delta time (seconds)
Maximum interactive CPU utilization
AVGINTUTL
Dispatched CPU time waiting (seconds)
Dispatched CPU time active (seconds)
Transferred CPU time (seconds)
Reserved (seconds)
Other waits (seconds)
Semaphore contention (seconds)
Mutex contention (seconds)
Ineligible waits (seconds)
Reserved (seconds)
J9 JVM tool thread (seconds)
Reserved (seconds)
Reserved (seconds)
Socket transmits (seconds)
Socket receives (seconds)
Socket other (seconds)

Identify top jobs in the pool with most page faults



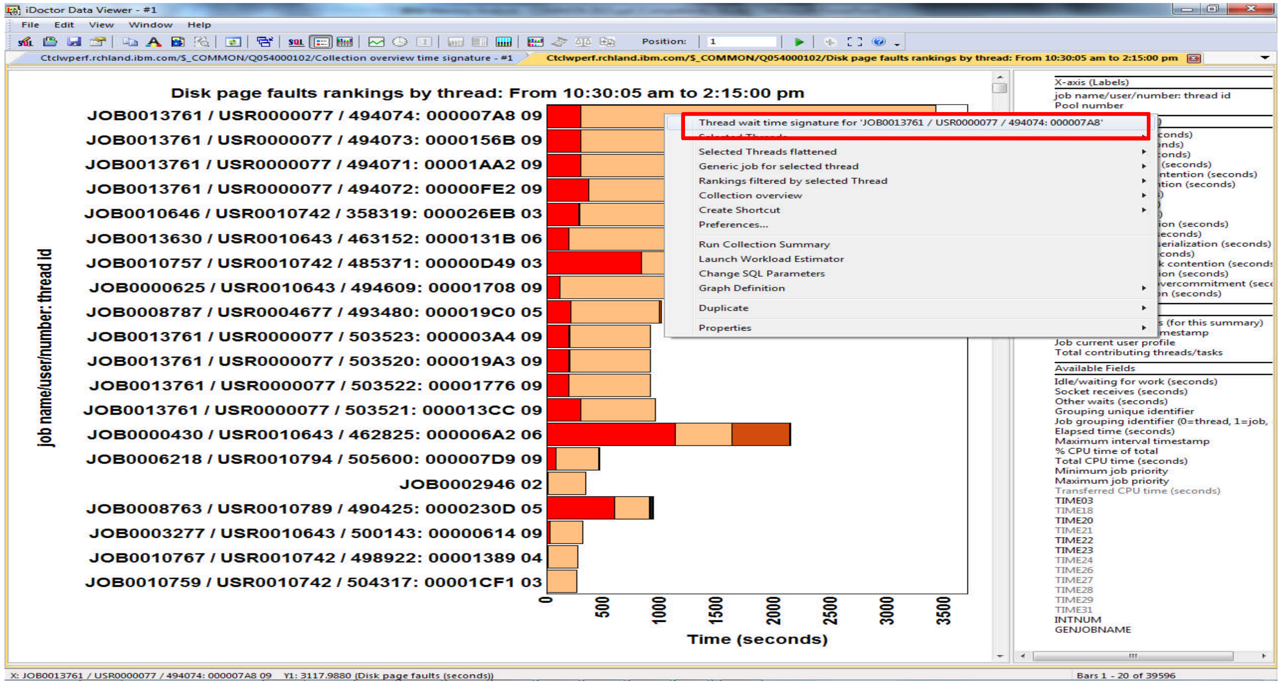
Worst page faulting jobs for all pools

- The following chart shows worst offending jobs and the pools they run in
- We added the Pool No. to the X-Axis label.


iDoctor Data Viewer - #1
File
Edit
View
Window
Help
Position:
1
Ctclwperf.rchland.ibm.com/$_COMMON/Q054000102/Collection overview time signature - #1
Ctclwperf.rchland.ibm.com/$_COMMON/Q054000102/Disk page faults rankings by thread: From 10:30:05 am to 2:15:00 pm
Disk page faults rankings by thread: From 10:30:05 am to 2:15:00 pm
job name/user/number: thread id
JOB0013761 / USR0000077 / 494074: 000007A8 09
JOB0013761 / USR0000077 / 494073: 0000156B 09
JOB0013761 / USR0000077 / 494071: 00001AA2 09
JOB0013761 / USR0000077 / 494072: 00000FE2 09
JOB0010646 / USR0010742 / 358319: 000026EB 03
JOB0013630 / USR0010643 / 463152: 0000131B 06
JOB0010757 / USR0010742 / 485371: 00000D49 03
JOB0000625 / USR0010643 / 494609: 00001708 09
JOB0008787 / USR0004677 / 493480: 000019C0 05
JOB0013761 / USR0000077 / 503523: 000003A4 09
JOB0013761 / USR0000077 / 503520: 000019A3 09
JOB0013761 / USR0000077 / 503522: 00001776 09
JOB0013761 / USR0000077 / 503521: 000013CC 09
JOB0000430 / USR0010643 / 462825: 000006A2 06
JOB0006218 / USR0010794 / 505600: 000007D9 09
JOB0002946 02
JOB0008763 / USR0010789 / 490425: 0000230D 05
JOB0003277 / USR0010643 / 500143: 00000614 09
JOB0010767 / USR0010742 / 498922: 00001389 04
JOB0010759 / USR0010742 / 504317: 00001CF1 03
0
500
1000
1500
2000
2500
3000
3500
Time (seconds)
X-axis (Labels)
job name/user/number: thread id
Pool number
Primary Y-axis (Bars)
Dispatched CPU (seconds)
CPU queueing (seconds)
Disk page faults (seconds)
Disk non fault reads (seconds)
Disk space usage contention (seconds)
Disk op-start contention (seconds)
Disk writes (seconds)
Disk other (seconds)
Journaling (seconds)
Semaphore contention (seconds)
Mutex contention (seconds)
Machine level gate serialization (seconds)
Seize contention (seconds)
Database record lock contention (seconds
Object lock contention (seconds)
Main storage pool overcommitment (seco
Abnormal contention (seconds)
Flyover Fields
Job runtime in hours (for this summary)
Minimum interval timestamp
Job current user profile
Total contributing threads/tasks
Available Fields
Idle/waiting for work (seconds)
Socket receives (seconds)
Other waits (seconds)
Grouping unique identifier
Job grouping identifier (0=thread, 1=job,
Elapsed time (seconds)
Maximum interval timestamp
% CPU time of total
Total CPU time (seconds)
Minimum job priority
Maximum job priority
Transferred CPU time (seconds)
TIME03
TIME18
TIME20
TIME21
TIME22
TIME23
TIME24
TIME26
TIME27
TIME28
TIME29
TIME31
INTNUM
GENJOBNAME
iDocCS.mdb QAIDRGPH table SUM 420 ALTID 1 SREFNO 951; Memory - 18.98% used - Graph tooltips enabled
Bars 1 - 20 of 39596

Memory Analysis - CSI

- Right click on jobs of interest to get wait time signature
- It is often useful to group threads in different ways such as by generic name, current user id etc.

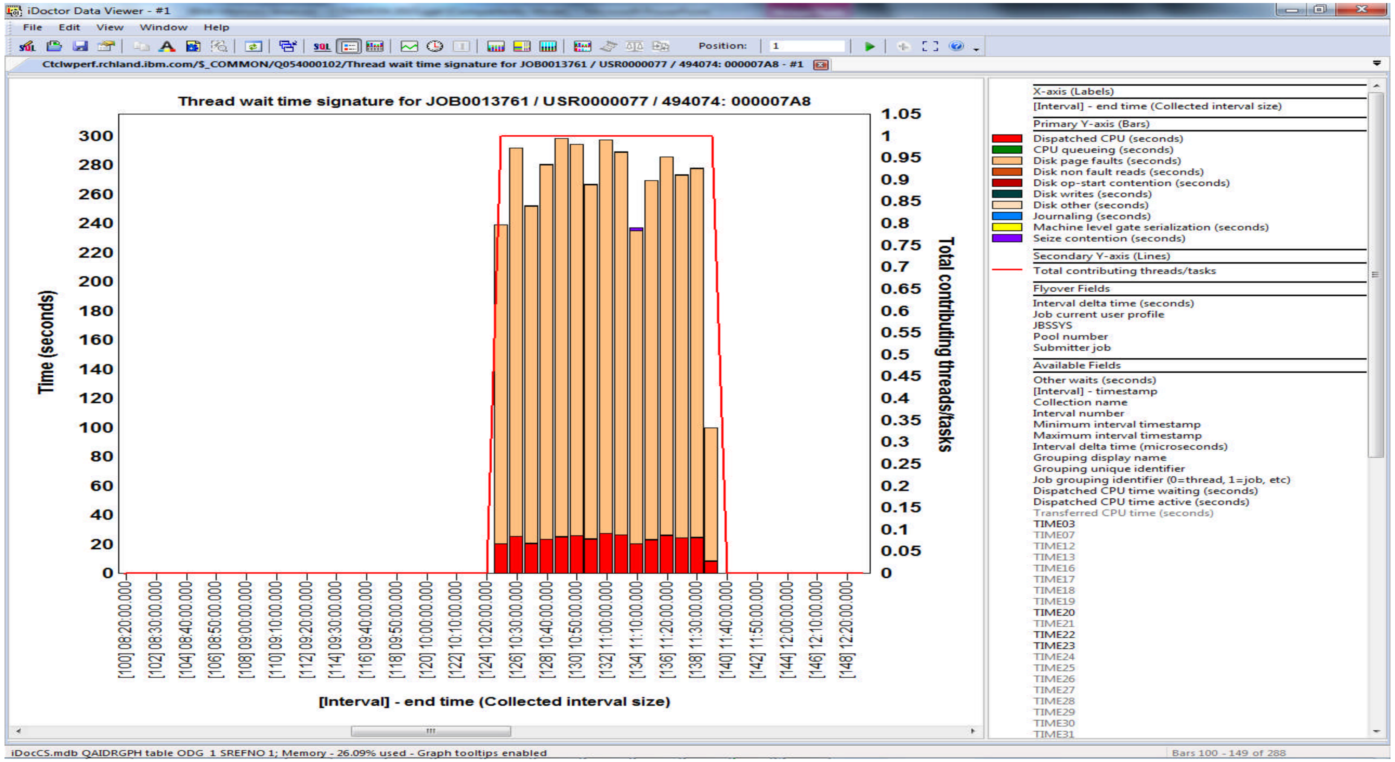

iDoctor Data Viewer - #1
File
Edit
View
Window
Help
Position:
1
Ctclwperf.rchland.ibm.com/$_COMMON/Q054000102/Collection overview time signature - #1
Ctclwperf.rchland.ibm.com/$_COMMON/Q054000102/Disk page faults rankings by thread: From 10:30:05 am to 2:15:00 pm
Disk page faults rankings by thread: From 10:30:05 am to 2:15:00 pm
job name/user/number: thread id
JOB0013761 / USR0000077 / 494074: 000007A8 09
JOB0013761 / USR0000077 / 494073: 0000156B 09
JOB0013761 / USR0000077 / 494071: 00001AA2 09
JOB0013761 / USR0000077 / 494072: 00000FE2 09
JOB0010646 / USR0010742 / 358319: 000026EB 03
JOB0013630 / USR0010643 / 463152: 0000131B 06
JOB0010757 / USR0010742 / 485371: 00000D49 03
JOB0000625 / USR0010643 / 494609: 00001708 09
JOB0008787 / USR0004677 / 493480: 000019C0 05
JOB0013761 / USR0000077 / 503523: 000003A4 09
JOB0013761 / USR0000077 / 503520: 000019A3 09
JOB0013761 / USR0000077 / 503522: 00001776 09
JOB0013761 / USR0000077 / 503521: 000013CC 09
JOB0000430 / USR0010643 / 462825: 000006A2 06
JOB0006218 / USR0010794 / 505600: 000007D9 09
JOB0002946 02
JOB0008763 / USR0010789 / 490425: 0000230D 05
JOB0003277 / USR0010643 / 500143: 00000614 09
JOB0010767 / USR0010742 / 498922: 00001389 04
JOB0010759 / USR0010742 / 504317: 00001CF1 03
0
500
1000
1500
2000
2500
3000
3500
Time (seconds)
Thread wait time signature for 'JOB0013761 / USR0000077 / 494074: 000007A8'
Selected Threads
Selected Threads flattened
Generic job for selected thread
Rankings filtered by selected Thread
Collection overview
Create Shortcut
Preferences...
Run Collection Summary
Launch Workload Estimator
Change SQL Parameters
Graph Definition
Duplicate
Properties
X-axis (Labels)
job name/user/number: thread id
Pool number
conds)
onds)
conds)
(seconds)
ntention (seconds)
tion (seconds)
ion (seconds)
econds)
erialization (seconds)
conds)
k contention (seconds
ion (seconds)
vercommitment (sec
n (seconds)
s (for this summary)
mestamp
Job current user profile
Total contributing threads/tasks
Available Fields
Idle/waiting for work (seconds)
Socket receives (seconds)
Other waits (seconds)
Grouping unique identifier
Job grouping identifier (0=thread, 1=job,
Elapsed time (seconds)
Maximum interval timestamp
% CPU time of total
Total CPU time (seconds)
Minimum job priority
Maximum job priority
Transferred CPU time (seconds)
TIME03
TIME18
TIME20
TIME21
TIME22
TIME23
TIME24
TIME26
TIME27
TIME28
TIME29
TIME31
INTNUM
GENJOBNAME

X: JOB0013761 / USR0000077 / 494074: 000007A8 09 Y1: 3117.9880 (Disk page faults (seconds))

Bars 1 - 20 of 39596

Memory Analysis - CSI

- Click on any interval to get the specific wait statistics



Ctclwperf.rchland.ibm.com/$_COMMON/Q054000102/Thread wait time signature for JOB0013761 / USR0000077 / 494074: 000007A8 - #1

X-axis (Labels)

[Interval] - end time (Collected interval size)

Primary Y-axis (Bars)

- 🟥 Dispatched CPU (seconds)
- 🟩 CPU queueing (seconds)
- 🟧 Disk page faults (seconds)
- 🟧 Disk non fault reads (seconds)
- 🟥 Disk op-start contention (seconds)
- ■ Disk writes (seconds)
- □ Disk other (seconds)
- 🟦 Journaling (seconds)
- 🟨 Machine level gate serialization (seconds)
- 🟪 Seize contention (seconds)

Secondary Y-axis (Lines)

- — Total contributing threads/tasks

Flyover Fields

Interval delta time (seconds)
Job current user profile
JBSSYS
Pool number
Submitter job

Available Fields

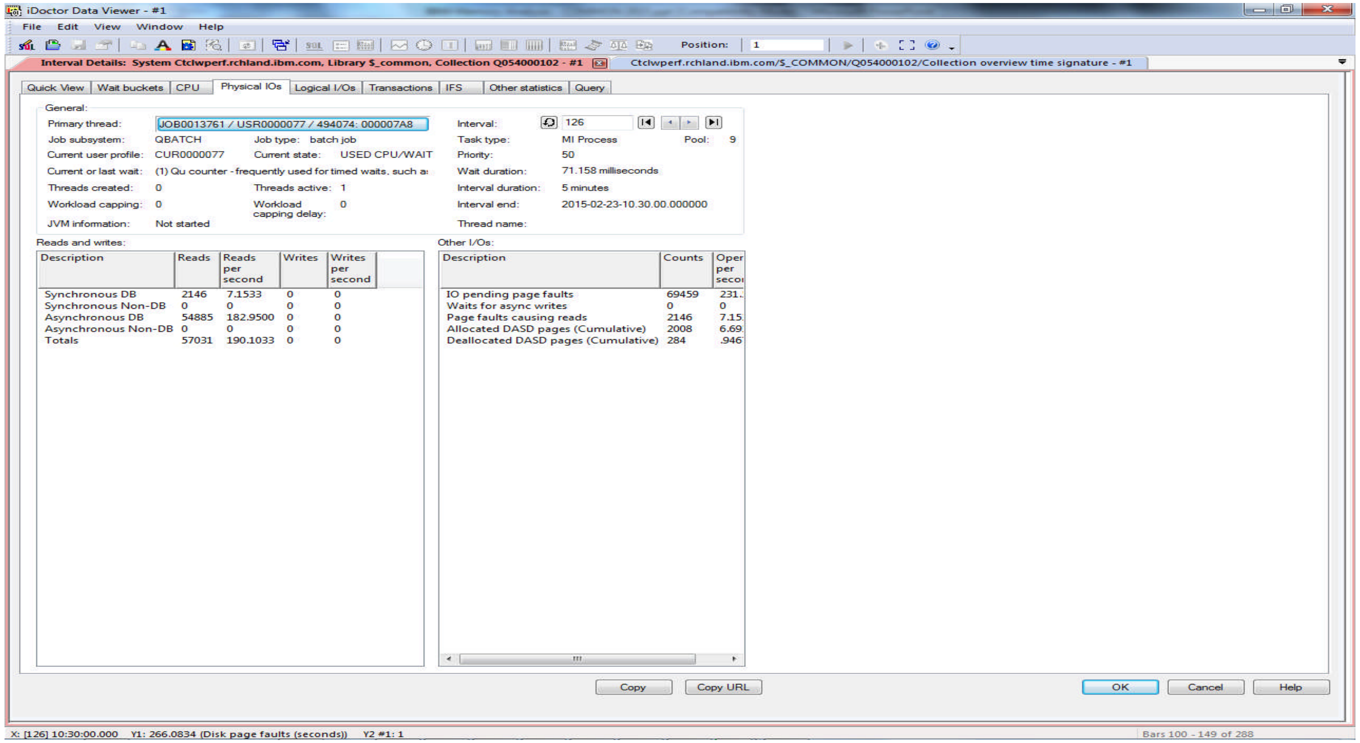
Other waits (seconds)
[Interval] - timestamp
Collection name
Interval number
Minimum interval timestamp
Maximum interval timestamp
Interval delta time (microseconds)
Grouping display name
Grouping unique identifier
Job grouping identifier (0=thread, 1=job, etc)
Dispatched CPU time waiting (seconds)
Dispatched CPU time active (seconds)
Transferred CPU time (seconds)
TIME03
TIME07
TIME12
TIME13
TIME16
TIME17
TIME18
TIME19
TIME20
TIME21
TIME22
TIME23
TIME24
TIME25
TIME26
TIME27
TIME28
TIME29
TIME30
TIME31

iDocCS.mdb QAIDRGPH table ODG 1 SREFNO 1; Memory - 26.09% used - Graph tooltips enabled

Bars 100 - 149 of 288

Memory Analysis - CSI

- The wait time details are shown in the tabs below



Interval Details: System Ctclwperf.rchland.ibm.com, Library $_common, Collection Q054000102 - #1

Ctclwperf.rchland.ibm.com/$_COMMON/Q054000102/Collection overview time signature - #1

General:

Primary thread: JOB0013761 / USR0000077 / 494074: 000007A8

Job subsystem: QBATCH   Job type: batch job

Current user profile: CUR0000077   Current state: USED CPU/WAIT

Current or last wait: (1) Qu counter - frequently used for timed waits, such a:

Threads created: 0   Threads active: 1

Workload capping: 0   Workload capping delay: 0

JVM information: Not started

Interval: 126

Task type: MI Process   Pool: 9

Priority: 50

Wait duration: 71.158 milliseconds

Interval duration: 5 minutes

Interval end: 2015-02-23-10.30.00.000000

Thread name:

Reads and writes:

Description	Reads	Reads per second	Writes	Writes per second
Synchronous DB	2146	7.1533	0	0
Synchronous Non-DB	0	0	0	0
Asynchronous DB	54885	182.9500	0	0
Asynchronous Non-DB	0	0	0	0
Totals	57031	190.1033	0	0

Other I/Os:

Description	Counts	Oper per secoı
IO pending page faults	69459	231.
Waits for async writes	0	0
Page faults causing reads	2146	7.15
Allocated DASD pages (Cumulative)	2008	6.69
Deallocated DASD pages (Cumulative)	284	.946

X: [126] 10:30:00.000   Y1: 266.0834 (Disk page faults (seconds))   Y2 #1: 1

Bars 100 - 149 of 288

Collection Services Memory analysis summary

- We can tell how much faulting is happening at a system level
  - And at a job or thread or task level
- We can tell what pools have the most faulting
- We can see if memory pools are changing in size
- We can't tell what the jobs are doing that is causing faulting
  - Or what they are faulting on

- For additional details….
  - Job Watcher data can provide answers to many questions.

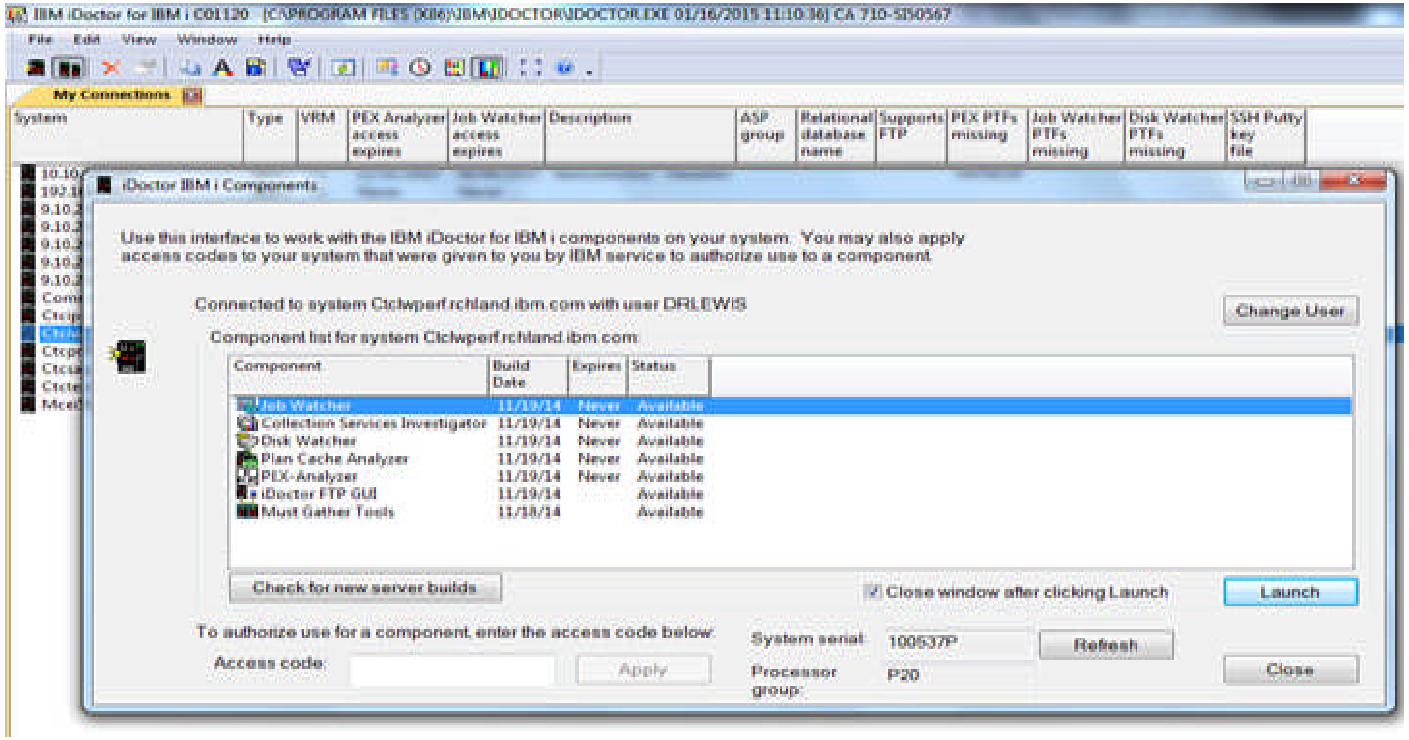
Job Watcher Memory Related Data

- Job Watcher has similar information as Collections Services. Key differences are:
  - Intervals typically much shorter (5-10 seconds vs. 5-15 minutes)
  - Additional data is collected
    - Call stacks are collected
    - Objects being waited on are collected
    - SQL being run is collected

- Things are different once we start looking at interval details

Launching Job Watcher

- Select Job Watcher and Click Launch



IBM iDoctor for IBM i C01120 [C:\PROGRAM FILES (X86)\IBM\IDOCTOR\IDOCTOR.EXE 01/16/2015 11:10:36] CA 710-SI50567
File Edit View Window Help
My Connections
System
Type
VRM
PEX Analyzer access expires
Job Watcher access expires
Description
ASP group
Relational database name
Supports FTP
PEX PTFs missing
Job Watcher PTFs missing
Disk Watcher PTFs missing
SSH Putty key file
iDoctor IBM i Components

Use this interface to work with the IBM iDoctor for IBM i components on your system. You may also apply access codes to your system that were given to you by IBM service to authorize use to a component.

Connected to system Ctclwperf.rchland.ibm.com with user DRLEWIS

Component list for system Ctclwperf.rchland.ibm.com

Component	Build Date	Expires	Status
Job Watcher	11/19/14	Never	Available
Collection Services Investigator	11/19/14	Never	Available
Disk Watcher	11/19/14	Never	Available
Plan Cache Analyzer	11/19/14	Never	Available
PEX-Analyzer	11/19/14	Never	Available
iDoctor FTP GUI	11/19/14		Available
Must Gather Tools	11/18/14		Available

☑ Close window after clicking Launch

To authorize use for a component, enter the access code below:

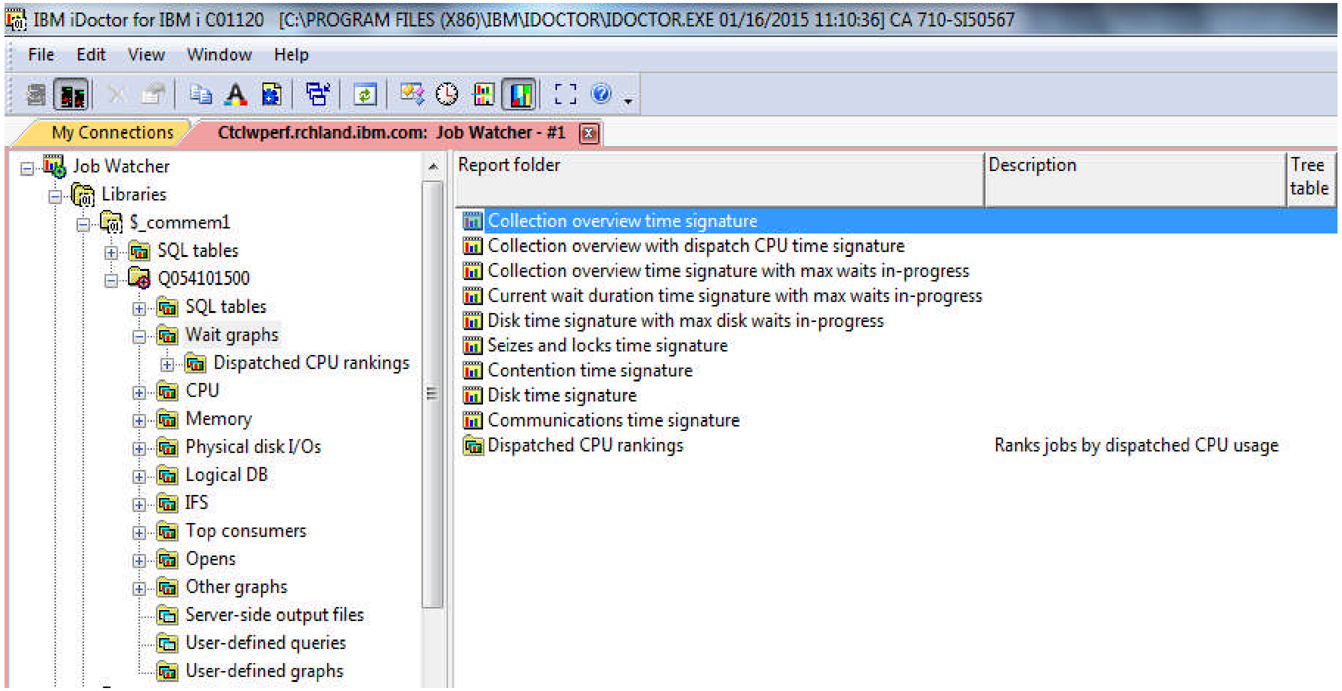
Access code:

System serial: 100537P

Processor group: P20

Memory Analysis - Job Watcher

- Entry Point is the same as CSI – Collection Overview time Signature, but the chart will show shorter intervals for a more obvious impact of all wait times

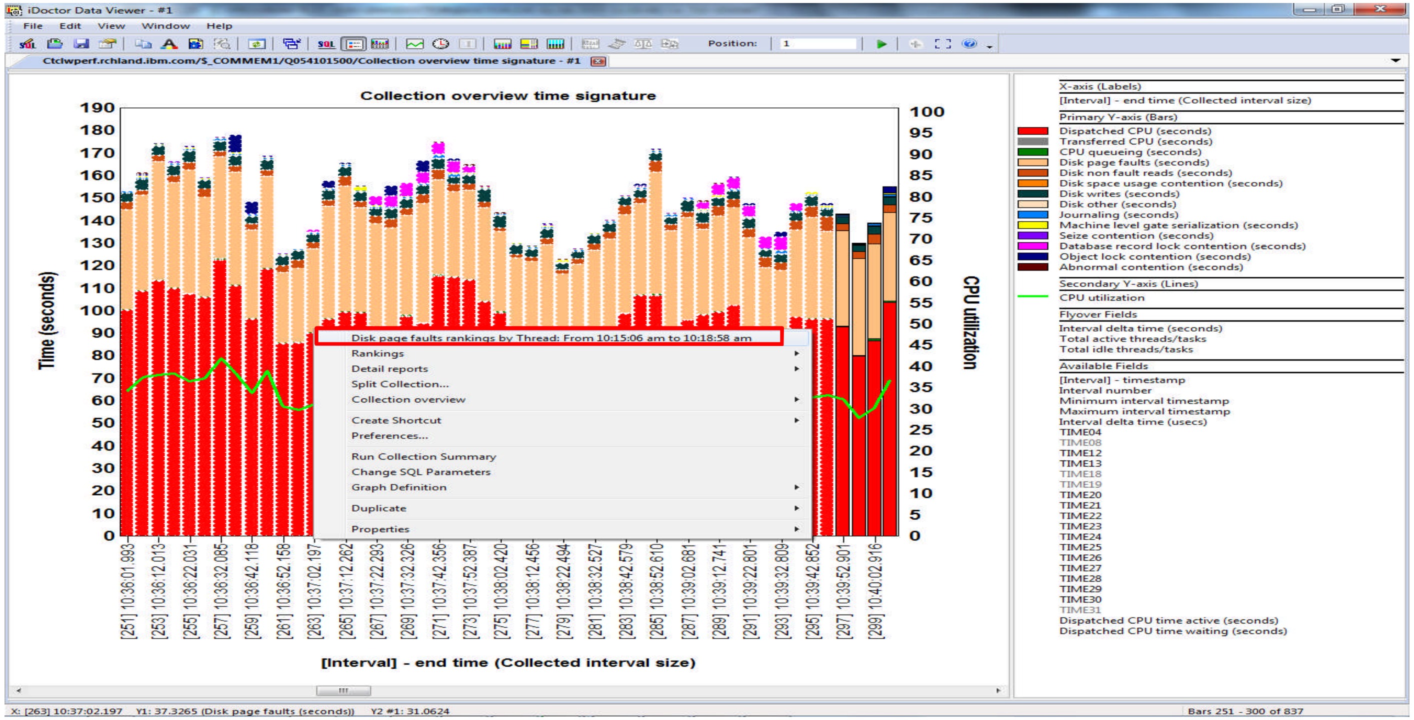


IBM iDoctor for IBM i C01120 [C:\PROGRAM FILES (X86)\IBM\IDOCTOR\IDOCTOR.EXE 01/16/2015 11:10:36] CA 710-SI50567

Report folder	Description	Tree table
Collection overview time signature		
Collection overview with dispatch CPU time signature		
Collection overview time signature with max waits in-progress		
Current wait duration time signature with max waits in-progress		
Disk time signature with max disk waits in-progress		
Seizes and locks time signature		
Contention time signature		
Disk time signature		
Communications time signature		
Dispatched CPU rankings	Ranks jobs by dispatched CPU usage	

Job Watcher drill down for memory analysis

- The same drill down process can be used here as it was in CSI

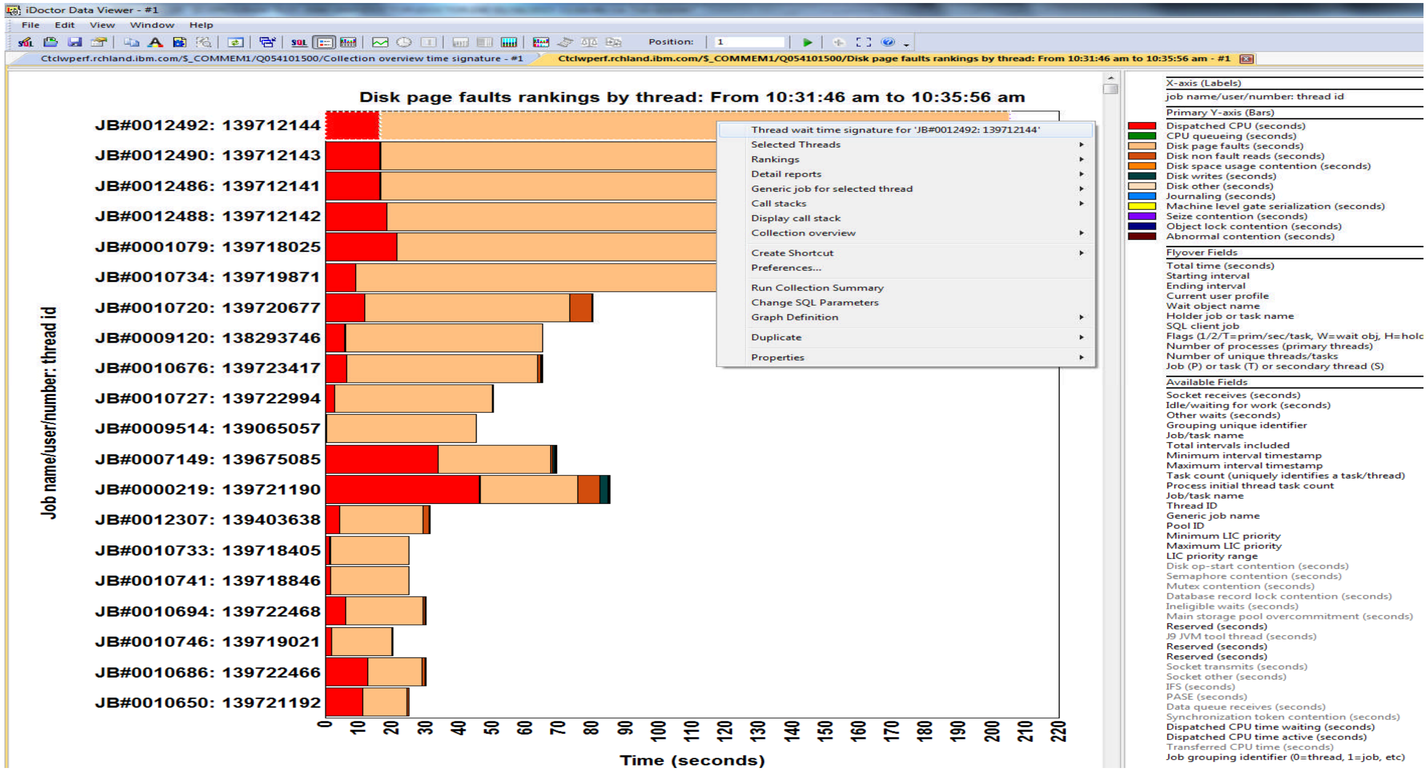


Ctclwperf.rchland.ibm.com/$_COMMEM1/Q054101500/Collection overview time signature - #1

X: [263] 10:37:02.197 Y1: 37.3265 (Disk page faults (seconds)) Y2 #1: 31.0624 Bars 251 - 300 of 837

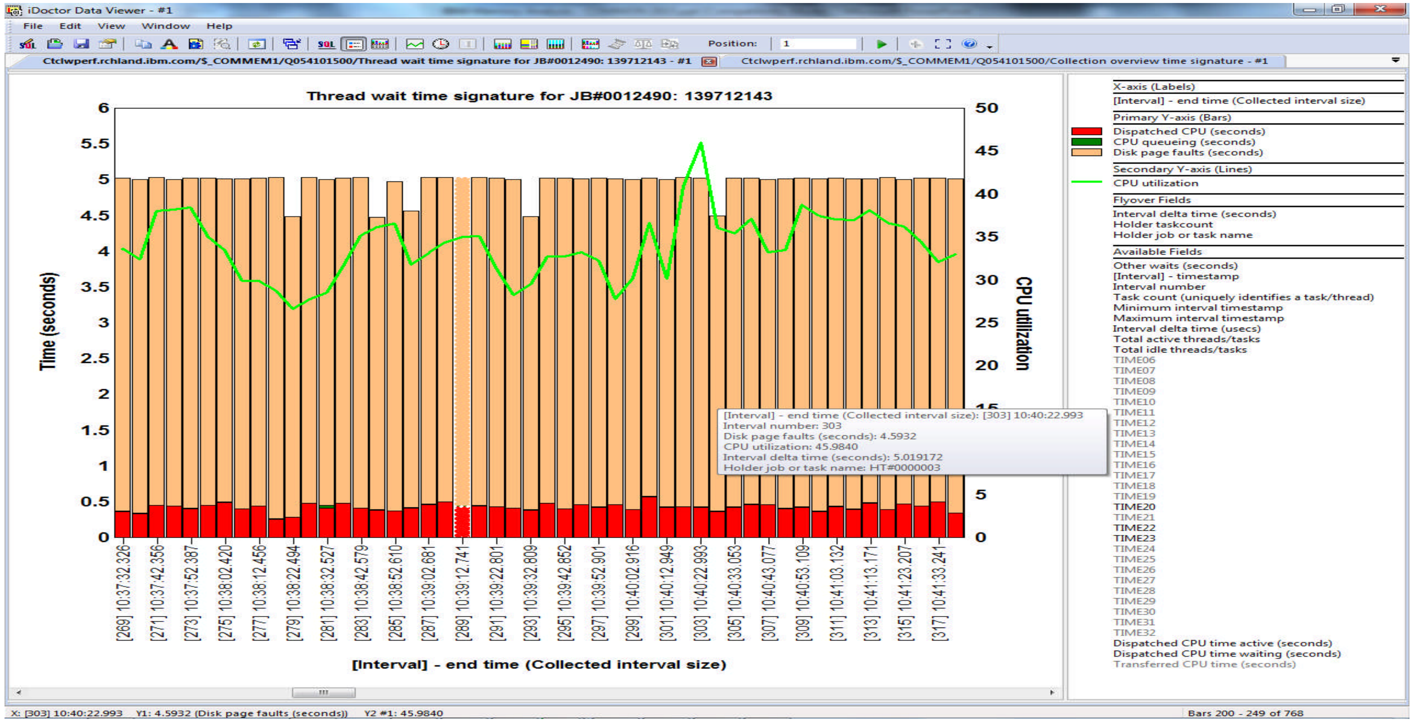
Analyzing job-threads with high page faulting

- Threads ranked by longest page fault wait times


iDoctor Data Viewer - #1
File
Edit
View
Window
Help
SQL
Position:
1
Ctclwperf.rchland.ibm.com/$_COMMEM1/Q054101500/Collection overview time signature - #1
Ctclwperf.rchland.ibm.com/$_COMMEM1/Q054101500/Disk page faults rankings by thread: From 10:31:46 am to 10:35:56 am - #1
Disk page faults rankings by thread: From 10:31:46 am to 10:35:56 am
Job name/user/number: thread id
JB#0012492: 139712144
JB#0012490: 139712143
JB#0012486: 139712141
JB#0012488: 139712142
JB#0001079: 139718025
JB#0010734: 139719871
JB#0010720: 139720677
JB#0009120: 138293746
JB#0010676: 139723417
JB#0010727: 139722994
JB#0009514: 139065057
JB#0007149: 139675085
JB#0000219: 139721190
JB#0012307: 139403638
JB#0010733: 139718405
JB#0010741: 139718846
JB#0010694: 139722468
JB#0010746: 139719021
JB#0010686: 139722466
JB#0010650: 139721192
0
10
20
30
40
50
60
70
80
90
100
110
120
130
140
150
160
170
180
190
200
210
220
Time (seconds)
Thread wait time signature for 'JB#0012492: 139712144'
Selected Threads
Rankings
Detail reports
Generic job for selected thread
Call stacks
Display call stack
Collection overview
Create Shortcut
Preferences...
Run Collection Summary
Change SQL Parameters
Graph Definition
Duplicate
Properties
X-axis (Labels)
job name/user/number: thread id
Primary Y-axis (Bars)
🟥 Dispatched CPU (seconds)
🟩 CPU queueing (seconds)
Disk page faults (seconds)
Disk non fault reads (seconds)
🟧 Disk space usage contention (seconds)
Disk writes (seconds)
Disk other (seconds)
🟦 Journaling (seconds)
🟨 Machine level gate serialization (seconds)
🟪 Seize contention (seconds)
Object lock contention (seconds)
Abnormal contention (seconds)
Flyover Fields
Total time (seconds)
Starting interval
Ending interval
Current user profile
Wait object name
Holder job or task name
SQL client job
Flags (1/2/T=prim/sec/task, W=wait obj, H=hold
Number of processes (primary threads)
Number of unique threads/tasks
Job (P) or task (T) or secondary thread (S)
Available Fields
Socket receives (seconds)
Idle/waiting for work (seconds)
Other waits (seconds)
Grouping unique identifier
Job/task name
Total intervals included
Minimum interval timestamp
Maximum interval timestamp
Task count (uniquely identifies a task/thread)
Process initial thread task count
Job/task name
Thread ID
Generic job name
Pool ID
Minimum LIC priority
Maximum LIC priority
LIC priority range
Disk op-start contention (seconds)
Semaphore contention (seconds)
Mutex contention (seconds)
Database record lock contention (seconds)
Ineligible waits (seconds)
Main storage pool overcommitment (seconds)
Reserved (seconds)
J9 JVM tool thread (seconds)
Reserved (seconds)
Reserved (seconds)
Socket transmits (seconds)
Socket other (seconds)
IFS (seconds)
PASE (seconds)
Data queue receives (seconds)
Synchronization token contention (seconds)
Dispatched CPU time waiting (seconds)
Dispatched CPU time active (seconds)
Transferred CPU time (seconds)
Job grouping identifier (0=thread, 1=job, etc)

Investigating wait signature over time

- Use your mouse to fly over the intervals to see the CPU Utilization and page fault wait times in seconds



Ctclwperf.rchland.ibm.com/$_COMMEM1/Q054101500/Thread wait time signature for JB#0012490: 139712143 - #1

Ctclwperf.rchland.ibm.com/$_COMMEM1/Q054101500/Collection overview time signature - #1

[Interval] - end time (Collected interval size): [303] 10:40:22.993
Interval number: 303
Disk page faults (seconds): 4.5932
CPU utilization: 45.9840
Interval delta time (seconds): 5.019172
Holder job or task name: HT#0000003

X-axis (Labels)
[Interval] - end time (Collected interval size)

Primary Y-axis (Bars)
🟥 Dispatched CPU (seconds)
🟩 CPU queueing (seconds)
🟧 Disk page faults (seconds)

Secondary Y-axis (Lines)
— CPU utilization

Flyover Fields
Interval delta time (seconds)
Holder taskcount
Holder job or task name

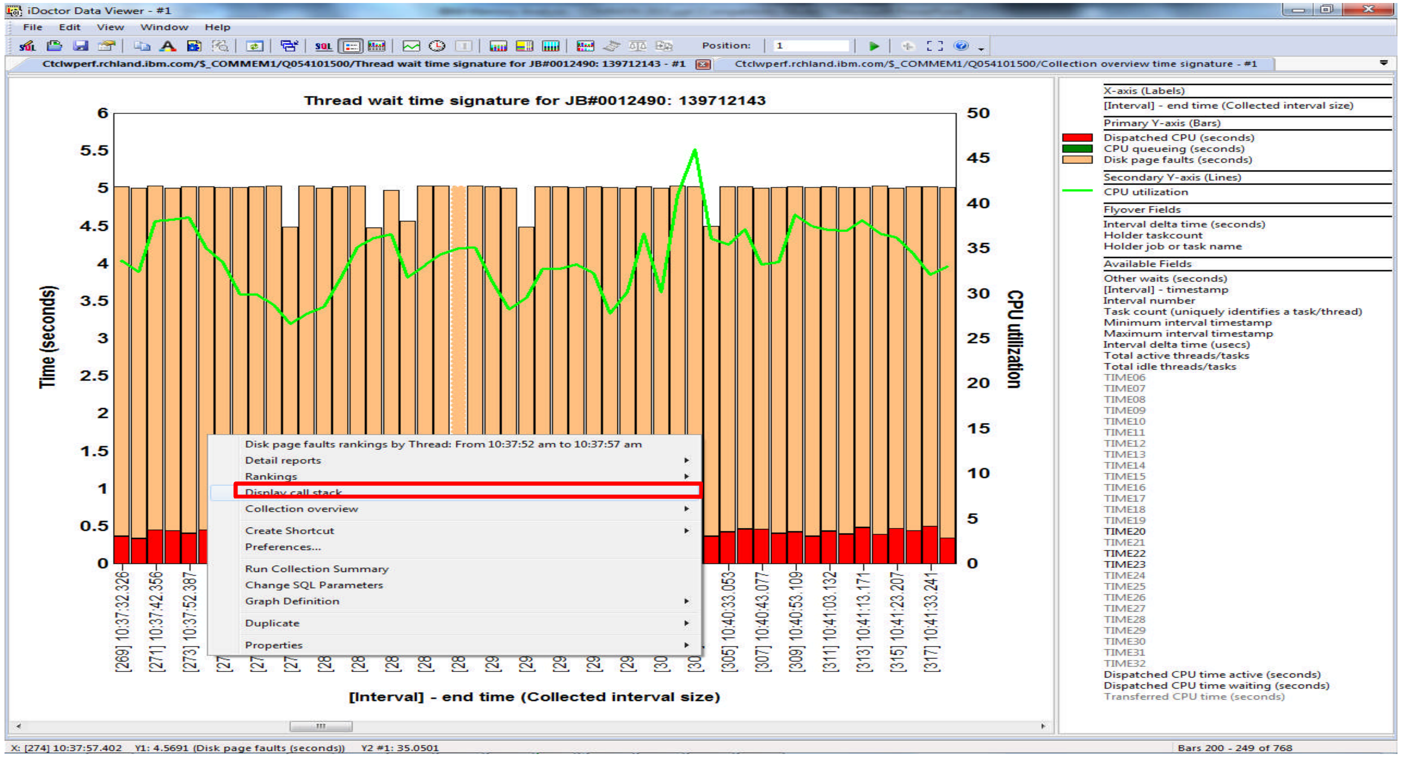
Available Fields
Other waits (seconds)
[Interval] - timestamp
Interval number
Task count (uniquely identifies a task/thread)
Minimum interval timestamp
Maximum interval timestamp
Interval delta time (usecs)
Total active threads/tasks
Total idle threads/tasks
TIME06
TIME07
TIME08
TIME09
TIME10
TIME11
TIME12
TIME13
TIME14
TIME15
TIME16
TIME17
TIME18
TIME19
TIME20
TIME21
TIME22
TIME23
TIME24
TIME25
TIME26
TIME27
TIME28
TIME29
TIME30
TIME31
TIME32
Dispatched CPU time active (seconds)
Dispatched CPU time waiting (seconds)
Transferred CPU time (seconds)

X: [303] 10:40:22.993 Y1: 4.5932 (Disk page faults (seconds)) Y2 #1: 45.9840

Bars 200 - 249 of 768

Reviewing Call stacks

- Right click on the intervals of interest and click on “Display call stack”



iDoctor Data Viewer - #1

Ctclwperf.rchland.ibm.com/$_COMMEM1/Q054101500/Thread wait time signature for JB#0012490: 139712143 - #1

Ctclwperf.rchland.ibm.com/$_COMMEM1/Q054101500/Collection overview time signature - #1

Position: 1

Thread wait time signature for JB#0012490: 139712143

X-axis (Labels)
[Interval] - end time (Collected interval size)

Primary Y-axis (Bars)
🟥 Dispatched CPU (seconds)
🟩 CPU queueing (seconds)
🟧 Disk page faults (seconds)

Secondary Y-axis (Lines)
— CPU utilization

Flyover Fields
Interval delta time (seconds)
Holder taskcount
Holder job or task name

Available Fields
Other waits (seconds)
[Interval] - timestamp
Interval number
Task count (uniquely identifies a task/thread)
Minimum interval timestamp
Maximum interval timestamp
Interval delta time (usecs)
Total active threads/tasks
Total idle threads/tasks
TIME06
TIME07
TIME08
TIME09
TIME10
TIME11
TIME12
TIME13
TIME14
TIME15
TIME16
TIME17
TIME18
TIME19
TIME20
TIME21
TIME22
TIME23
TIME24
TIME25
TIME26
TIME27
TIME28
TIME29
TIME30
TIME31
TIME32
Dispatched CPU time active (seconds)
Dispatched CPU time waiting (seconds)
Transferred CPU time (seconds)

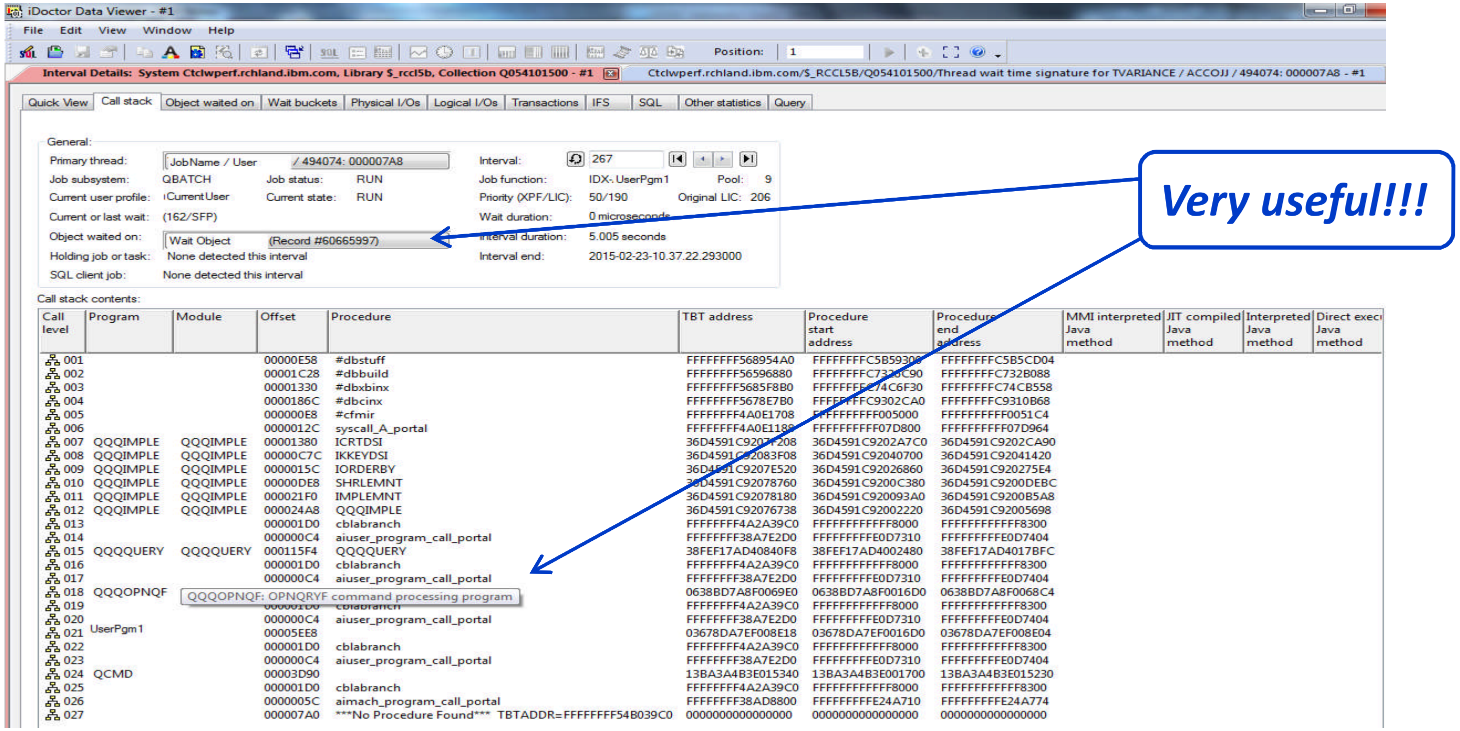
Disk page faults rankings by Thread: From 10:37:52 am to 10:37:57 am

X: [274] 10:37:57.402 Y1: 4.5691 (Disk page faults (seconds)) Y2 #1: 35.0501

Bars 200 - 249 of 768

Call Stack information

- The call stack will show the object faulted on, pool, and program driving it


iDoctor Data Viewer - #1
File
Edit
View
Window
Help
Position:
1
Interval Details: System Ctclwperf.rchland.ibm.com, Library $_rccl5b, Collection Q054101500 - #1
Ctclwperf.rchland.ibm.com/$_RCCL5B/Q054101500/Thread wait time signature for TVARIANCE / ACCOJJ / 494074: 000007A8 - #1
Quick View
Call stack
Object waited on
Wait buckets
Physical I/Os
Logical I/Os
Transactions
IFS
SQL
Other statistics
Query

Very useful!!!

General:

Primary thread: JobName / User / 494074: 000007A8

Interval: 267

Job subsystem: QBATCH

Job status: RUN

Job function: IDX-.UserPgm1

Pool: 9

Current user profile: CurrentUser

Current state: RUN

Priority (XPF/LIC): 50/190

Original LIC: 206

Current or last wait: (162/SFP)

Wait duration: 0 microseconds

Object waited on: Wait Object (Record #60665997)

Interval duration: 5.005 seconds

Holding job or task: None detected this interval

Interval end: 2015-02-23-10.37.22.293000

SQL client job: None detected this interval

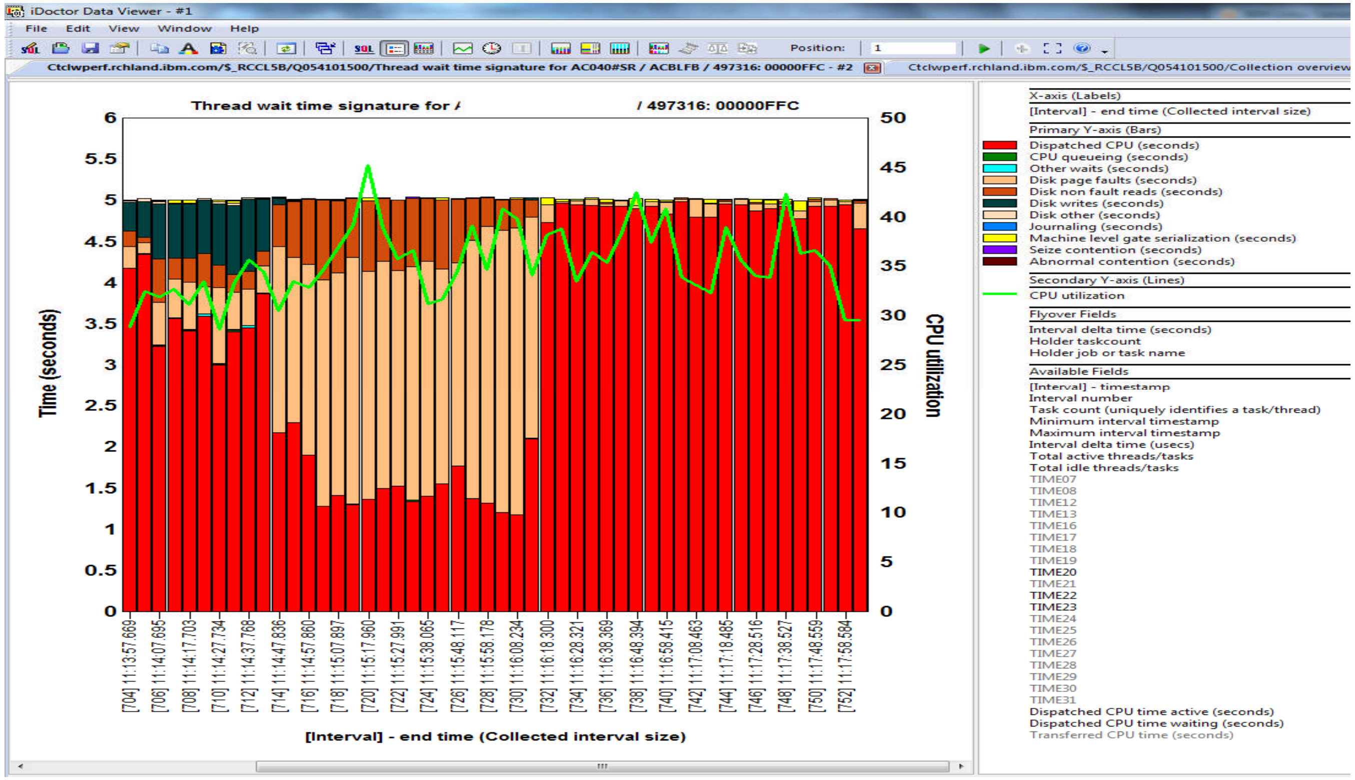
Call stack contents:

Call level	Program	Module	Offset	Procedure	TBT address	Procedure start address	Procedure end address	MMI interpreted Java method	JIT compiled Java method	Interpreted Java method	Direct exec Java method
001			00000E58	#dbstuff	FFFFFFFF568954A0	FFFFFFFFC5B59300	FFFFFFFFC5B5CD04				
002			00001C28	#dbbuild	FFFFFFFF56596880	FFFFFFFFC7328C90	FFFFFFFFC732B088				
003			00001330	#dbxbinx	FFFFFFFF5685F8B0	FFFFFFFFC74C6F30	FFFFFFFFC74CB558				
004			0000186C	#dbcinx	FFFFFFFF5678E7B0	FFFFFFFFC9302CA0	FFFFFFFFC9310B68				
005			000000E8	#cfmir	FFFFFFFF4A0E1708	FFFFFFFFFF005000	FFFFFFFFFF0051C4				
006			0000012C	syscall_A_portal	FFFFFFFF4A0E1188	FFFFFFFFFF07D800	FFFFFFFFFF07D964				
007	QQQIMPLE	QQQIMPLE	00001380	ICRTDSI	36D4591C9207F208	36D4591C9202A7C0	36D4591C9202CA90				
008	QQQIMPLE	QQQIMPLE	00000C7C	IKKEYDSI	36D4591C92083F08	36D4591C92040700	36D4591C92041420				
009	QQQIMPLE	QQQIMPLE	0000015C	IORDERBY	36D4591C9207E520	36D4591C92026860	36D4591C920275E4				
010	QQQIMPLE	QQQIMPLE	00000DE8	SHRLEMNT	36D4591C92078760	36D4591C9200C380	36D4591C9200DEBC				
011	QQQIMPLE	QQQIMPLE	000021F0	IMPLEMNT	36D4591C92078180	36D4591C920093A0	36D4591C9200B5A8				
012	QQQIMPLE	QQQIMPLE	000024A8	QQQIMPLE	36D4591C92076738	36D4591C92002220	36D4591C92005698				
013			000001D0	cblabranch	FFFFFFFF4A2A39C0	FFFFFFFFFFFF8000	FFFFFFFFFFFF8300				
014			000000C4	aiuser_program_call_portal	FFFFFFFF38A7E2D0	FFFFFFFFFE0D7310	FFFFFFFFFE0D7404				
015	QQQQUERY	QQQQUERY	000115F4	QQQQUERY	38FEF17AD40840F8	38FEF17AD4002480	38FEF17AD4017BFC				
016			000001D0	cblabranch	FFFFFFFF4A2A39C0	FFFFFFFFFFFF8000	FFFFFFFFFFFF8300				
017			000000C4	aiuser_program_call_portal	FFFFFFFF38A7E2D0	FFFFFFFFFE0D7310	FFFFFFFFFE0D7404				
018	QQQOPNQF		[illegible]	[illegible]	0638BD7A8F0069E0	0638BD7A8F0016D0	0638BD7A8F0068C4				
019			000001D0	cblabranch	FFFFFFFF4A2A39C0	FFFFFFFFFFFF8000	FFFFFFFFFFFF8300				
020			000000C4	aiuser_program_call_portal	FFFFFFFF38A7E2D0	FFFFFFFFFE0D7310	FFFFFFFFFE0D7404				
021	UserPgm1		00005EE8		03678DA7EF008E18	03678DA7EF0016D0	03678DA7EF008E04				
022			000001D0	cblabranch	FFFFFFFF4A2A39C0	FFFFFFFFFFFF8000	FFFFFFFFFFFF8300				
023			000000C4	aiuser_program_call_portal	FFFFFFFF38A7E2D0	FFFFFFFFFE0D7310	FFFFFFFFFE0D7404				
024	QCMD		00003D90		13BA3A4B3E015340	13BA3A4B3E001700	13BA3A4B3E015230				
025			000001D0	cblabranch	FFFFFFFF4A2A39C0	FFFFFFFFFFFF8000	FFFFFFFFFFFF8300				
026			0000005C	aimach_program_call_portal	FFFFFFFF38AD8800	FFFFFFFFFE24A710	FFFFFFFFFE24A774				
027			000007A0	***No Procedure Found*** TBTADDR=FFFFFFFF54B039C0	0000000000000000	0000000000000000	0000000000000000				

QQQOPNQF: OPNQRYF command processing program

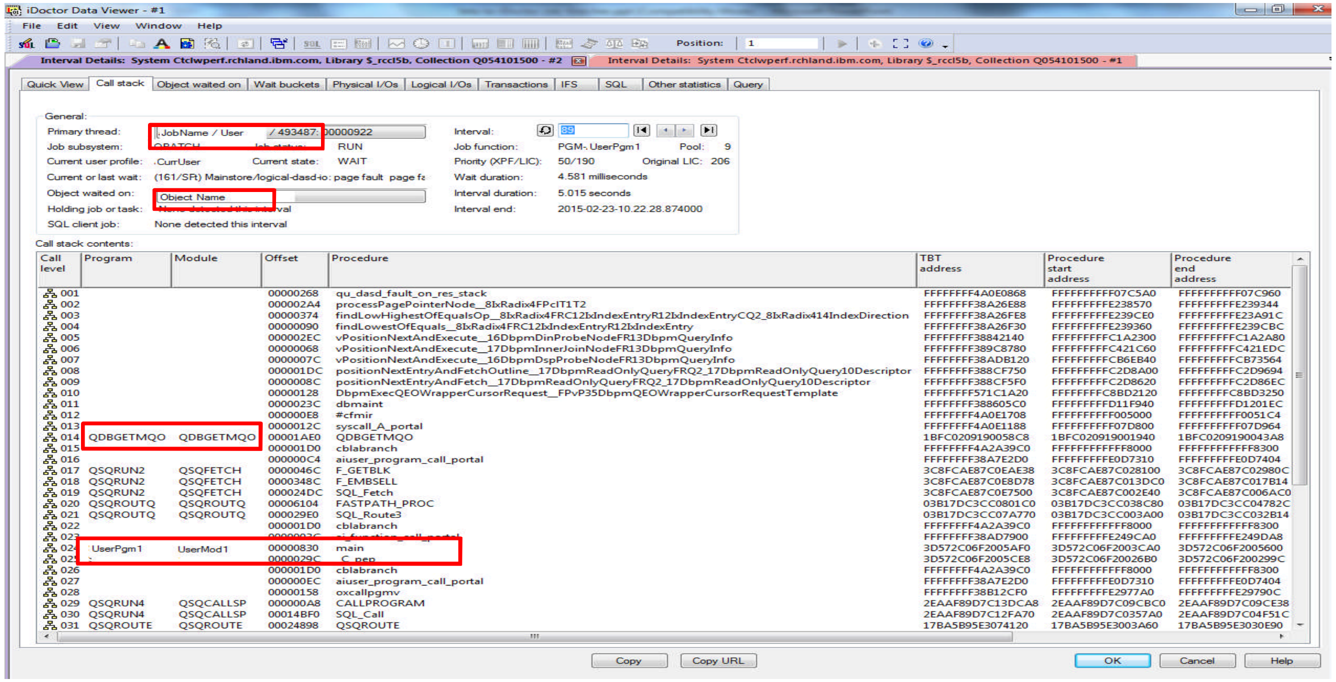
Example: Finding cause for intermittent page faults in a job

- In this example, a job shows intermittent faults



Identify program causing page faults

- The user program here is calling an SQL statement that then calls the QDBGETMQO (table or index scan) MI program


iDoctor Data Viewer - #1
File Edit View Window Help
Position: 1
Interval Details: System Ctclwperf.rchland.ibm.com, Library $_rccl5b, Collection Q054101500 - #2
Interval Details: System Ctclwperf.rchland.ibm.com, Library $_rccl5b, Collection Q054101500 - #1
Quick View
Call stack
Object waited on
Wait buckets
Physical I/Os
Logical I/Os
Transactions
IFS
SQL
Other statistics
Query

General:

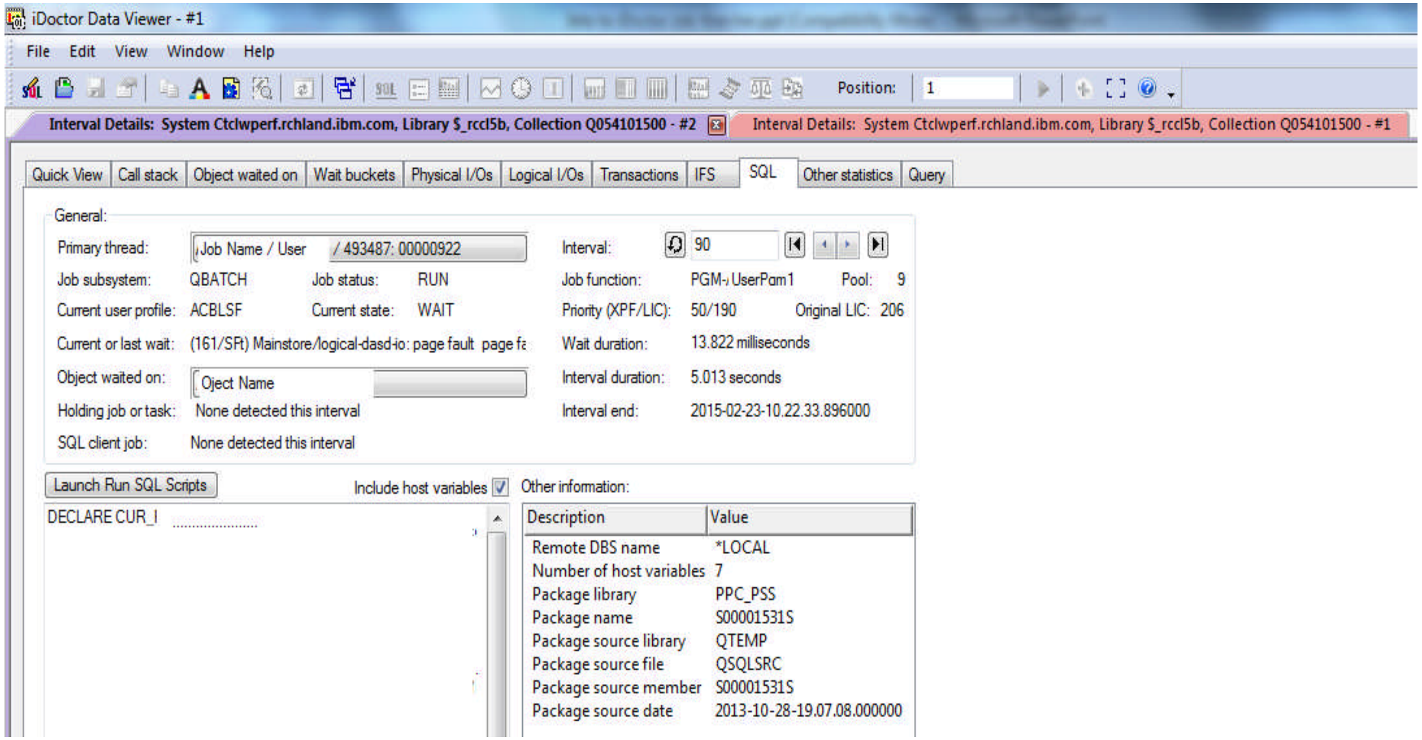
Primary thread:	JobName / User / 493487: 00000922		Interval:	89	
Job subsystem:	QBATCH	Job status: RUN	Job function:	PGM-UserPgm1	Pool: 9
Current user profile:	CurrUser	Current state: WAIT	Priority (XPF/LIC):	50/190	Original LIC: 206
Current or last wait:	(161/SFt) Mainstore/logical-dasd-io: page fault page fa		Wait duration:	4.581 milliseconds	
Object waited on:	Object Name		Interval duration:	5.015 seconds	
Holding job or task:	None detected this interval		Interval end:	2015-02-23-10.22.28.874000	
SQL client job:	None detected this interval				

Call stack contents:

Call level	Program	Module	Offset	Procedure	TBT address	Procedure start address	Procedure end address
001			00000268	qu_dasd_fault_on_res_stack	FFFFFFFF4A0E0868	FFFFFFFFFF07C5A0	FFFFFFFFFF07C960
002			000002A4	processPagePointerNode__8IxRadix4FPcIT1T2	FFFFFFFF38A26E88	FFFFFFFFFE238570	FFFFFFFFFE239344
003			00000374	findLowHighestOfEqualsOp__8IxRadix4FRC12IxIndexEntryR12IxIndexEntryCQ2_8IxRadix414IndexDirection	FFFFFFFF38A26FE8	FFFFFFFFFE239CE0	FFFFFFFFFE23A91C
004			00000090	findLowestOfEquals__8IxRadix4FRC12IxIndexEntryR12IxIndexEntry	FFFFFFFF38A26F30	FFFFFFFFFE239360	FFFFFFFFFE239CBC
005			000002EC	vPositionNextAndExecute__16DbpmDinProbeNodeFR13DbpmQueryInfo	FFFFFFFF38842140	FFFFFFFFFC1A2300	FFFFFFFFFC1A2A80
006			00000068	vPositionNextAndExecute__17DbpmInnerJoinNodeFR13DbpmQueryInfo	FFFFFFFF389C8780	FFFFFFFFFC421C60	FFFFFFFFFC421EDC
007			0000007C	vPositionNextAndExecute__16DbpmDspProbeNodeFR13DbpmQueryInfo	FFFFFFFF38ADB120	FFFFFFFFFCB6EB40	FFFFFFFFFCB73564
008			000001DC	positionNextEntryAndFetchOutline__17DbpmReadOnlyQueryFRQ2_17DbpmReadOnlyQuery10Descriptor	FFFFFFFF388CF750	FFFFFFFFFC2D8A00	FFFFFFFFFC2D9694
009			0000008C	positionNextEntryAndFetch__17DbpmReadOnlyQueryFRQ2_17DbpmReadOnlyQuery10Descriptor	FFFFFFFF388CF5F0	FFFFFFFFFC2D8620	FFFFFFFFFC2D86EC
010			00000128	DbpmExecQEOWrapperCursorRequest__FPvP35DbpmQEOWrapperCursorRequestTemplate	FFFFFFFF571C1A20	FFFFFFFFFC8BD2120	FFFFFFFFFC8BD3250
011			0000023C	dbmaint	FFFFFFFF388605C0	FFFFFFFFFD11F940	FFFFFFFFFD1201EC
012			000000E8	#cfmir	FFFFFFFF4A0E1708	FFFFFFFFFF005000	FFFFFFFFFF0051C4
013			0000012C	syscall_A_portal	FFFFFFFF4A0E1188	FFFFFFFFFF07D800	FFFFFFFFFF07D964
014	QDBGETMQO	QDBGETMQO	00001AE0	QDBGETMQO	1BFC0209190058C8	1BFC020919001940	1BFC0209190043A8
015			000001D0	cblabranch	FFFFFFFF4A2A39C0	FFFFFFFFFFFF8000	FFFFFFFFFFFF8300
016			000000C4	aiuser_program_call_portal	FFFFFFFF38A7E2D0	FFFFFFFFFE0D7310	FFFFFFFFFE0D7404
017	QSQRUN2	QSQFETCH	0000046C	F_GETBLK	3C8FCAE87C0EAE38	3C8FCAE87C028100	3C8FCAE87C02980C
018	QSQRUN2	QSQFETCH	0000348C	F_EMBSELL	3C8FCAE87C0E8D78	3C8FCAE87C013DC0	3C8FCAE87C017B14
019	QSQRUN2	QSQFETCH	000024DC	SQL_Fetch	3C8FCAE87C0E7500	3C8FCAE87C002E40	3C8FCAE87C006AC0
020	QSQROUTQ	QSQROUTQ	00006104	FASTPATH_PROC	03B17DC3CC0801C0	03B17DC3CC038C80	03B17DC3CC04782C
021	QSQROUTQ	QSQROUTQ	000029E0	SQL_Route3	03B17DC3CC07A770	03B17DC3CC003A00	03B17DC3CC032B14
022			000001D0	cblabranch	FFFFFFFF4A2A39C0	FFFFFFFFFFFF8000	FFFFFFFFFFFF8300
023			[illegible]	ai_function_call_portal	FFFFFFFF38AD7900	FFFFFFFFFE249CA0	FFFFFFFFFE249DA8
024	UserPgm1	UserMod1	00000830	main	3D572C06F2005AF0	3D572C06F2003CA0	3D572C06F2005600
025			0000029C	_C_pep	3D572C06F2005CE8	3D572C06F20026B0	3D572C06F200299C
026			000001D0	cblabranch	FFFFFFFF4A2A39C0	FFFFFFFFFFFF8000	FFFFFFFFFFFF8300
027			000000EC	aiuser_program_call_portal	FFFFFFFF38A7E2D0	FFFFFFFFFE0D7310	FFFFFFFFFE0D7404
028			00000158	oxcallpgmv	FFFFFFFF38B12CF0	FFFFFFFFFE2977A0	FFFFFFFFFE29790C
029	QSQRUN4	QSQCALLSP	000000A8	CALLPROGRAM	2EAAF89D7C13DCA8	2EAAF89D7C09CBC0	2EAAF89D7C09CE38
030	QSQRUN4	QSQCALLSP	00014BF0	SQL_Call	2EAAF89D7C12FA70	2EAAF89D7C0357A0	2EAAF89D7C04F51C
031	QSQROUTE	QSQROUTE	00024898	QSQROUTE	17BA5B95E3074120	17BA5B95E3003A60	17BA5B95E3030E90

Identify the SQL statement causing high page faults

- The SQL tab will often show which statement was run to cause the faulting



iDoctor Data Viewer - #1
File Edit View Window Help
Position: 1
Interval Details: System Ctclwperf.rchland.ibm.com, Library $_rccl5b, Collection Q054101500 - #2
Interval Details: System Ctclwperf.rchland.ibm.com, Library $_rccl5b, Collection Q054101500 - #1
Quick View
Call stack
Object waited on
Wait buckets
Physical I/Os
Logical I/Os
Transactions
IFS
SQL
Other statistics
Query
General:
Launch Run SQL Scripts

Primary thread: Job Name / User / 493487: 00000922

Job subsystem: QBATCH Job status: RUN

Current user profile: ACBLSF Current state: WAIT

Current or last wait: (161/SFt) Mainstore/logical-dasd-io: page fault page fa

Object waited on: Oject Name

Holding job or task: None detected this interval

SQL client job: None detected this interval

Interval: 90

Job function: PGM-UserPgm1 Pool: 9

Priority (XPF/LIC): 50/190 Original LIC: 206

Wait duration: 13.822 milliseconds

Interval duration: 5.013 seconds

Interval end: 2015-02-23-10.22.33.896000

Include host variables ☑

```
DECLARE CUR_I .....................
```

Other information:

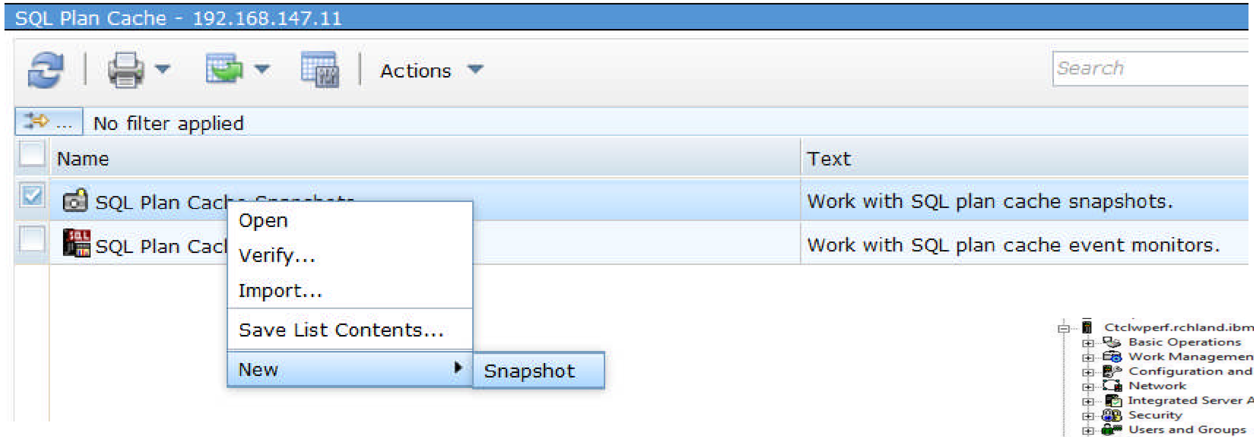
Description	Value
Remote DBS name	*LOCAL
Number of host variables	7
Package library	PPC_PSS
Package name	S00001531S
Package source library	QTEMP
Package source file	QSQLSRC
Package source member	S00001531S
Package source date	2013-10-28-19.07.08.000000

Job Watcher Memory Analysis Summary

- We can tell with more granularity what CS data showed us:
  - The job waiting the most on disk faults
  - The type of faults (DB vs. non-DB)
  - The average wait time on faults
  - The pool the job is running in
  - Etc.
- Additionally, we now know:
  - The object being faulted on
  - The program running
  - The OS operation causing the faulting
  - May also get SQL statement and host variables if running SQL

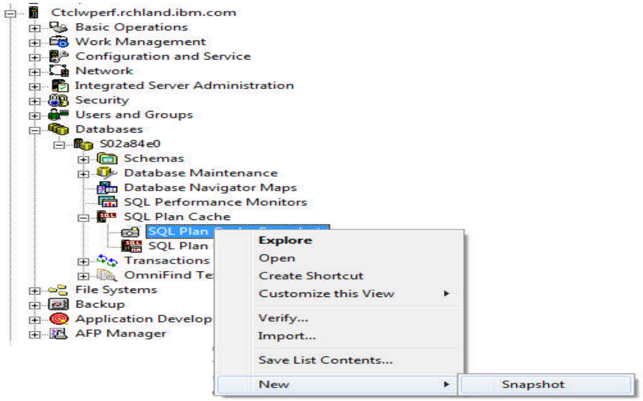
Navigator SQL Plan Cache Snapshot Analysis

- Run a plan cache snapshot against the tables being faulted in the most



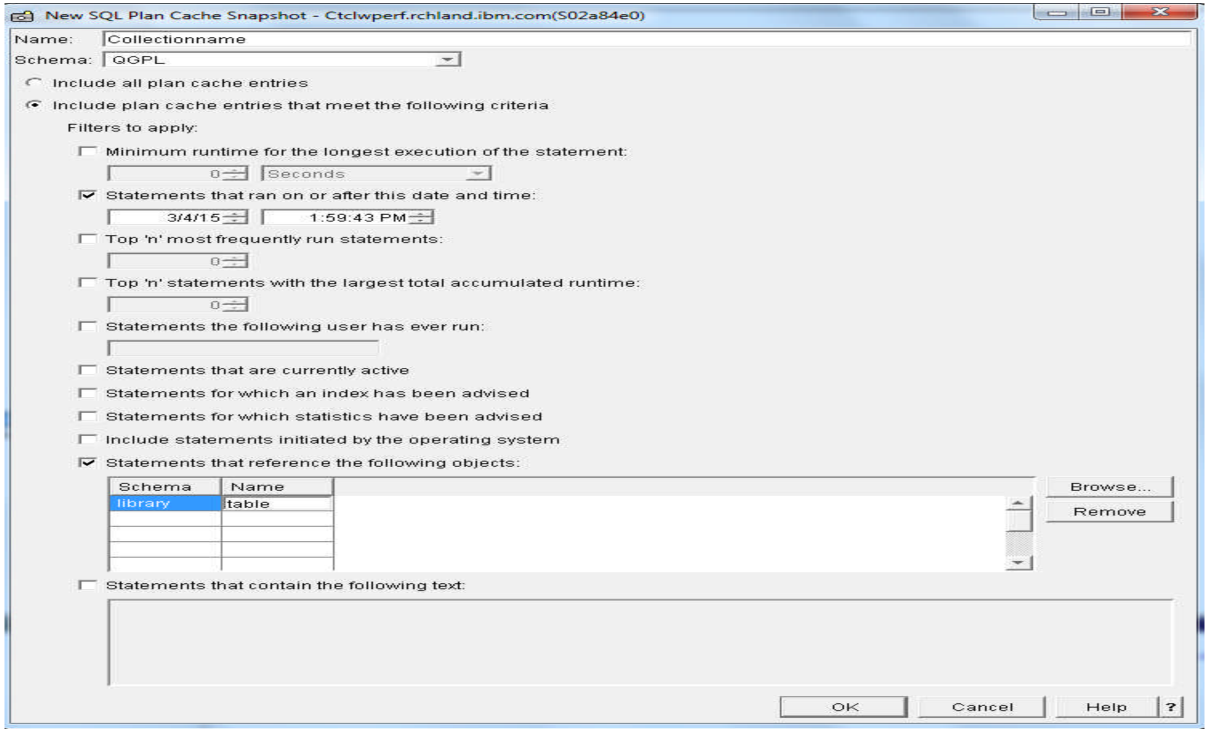
SQL Plan Cache - 192.168.147.11

Name	Text
☑ SQL Plan Cache Snapshots	Work with SQL plan cache snapshots.
□ SQL Plan Cache	Work with SQL plan cache event monitors.



Plan Cache Snapshot Analysis

- Filter the plan cache snapshot by objects



New SQL Plan Cache Snapshot - Ctclwperf.rchland.ibm.com(S02a84e0)

Name: Collectionname

Schema: QGPL

○ Include all plan cache entries

● Include plan cache entries that meet the following criteria

Filters to apply:

□ Minimum runtime for the longest execution of the statement:
0 Seconds

☑ Statements that ran on or after this date and time:
3/4/15 1:59:43 PM

□ Top 'n' most frequently run statements:
0

□ Top 'n' statements with the largest total accumulated runtime:
0

□ Statements the following user has ever run:

□ Statements that are currently active

□ Statements for which an index has been advised

□ Statements for which statistics have been advised

□ Include statements initiated by the operating system

☑ Statements that reference the following objects:

Schema	Name
library	table

□ Statements that contain the following text:

Data Management Best Practices for Memory

- Reduce file sizes
- Implement an index strategy
- Review index page size
- Remove deleted records (RGZPFM command)
- Purge historical data
- Consider adding SSD arms for those objects that have to be faulted in
- Avoid making any pool with activity too small – the faulting caused additional I/O, which in turn can affect response times for other jobs

IBM i Memory Tuning Best Practices - Summary

- Verify the memory configuration
  - Memory per processor
  - Processor affinity score
  - Reduce IO time by removing I/O bottlenecks
- Verify the pools are tuned correctly
  - Min/max pool size set up
  - Jobs running in the right pool
  - IBM i performance features are being used
- Tune application data access
- Optimize SQL queries
- Verify files size are right for an OLTP environment
  - Remove deleted records
  - Purge historical data

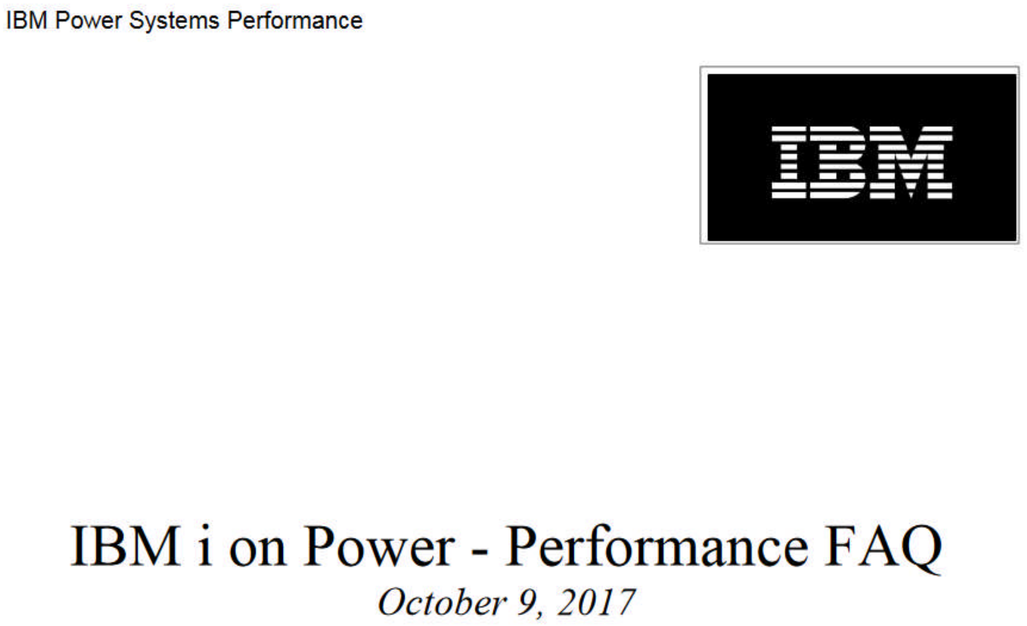
Questions?

References

IBM i Performance FAQ a MUST read!

October 2017 update *(watch for a Spring 2018 soon!):*

https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=POW03102USEN



IBM Power Systems Performance

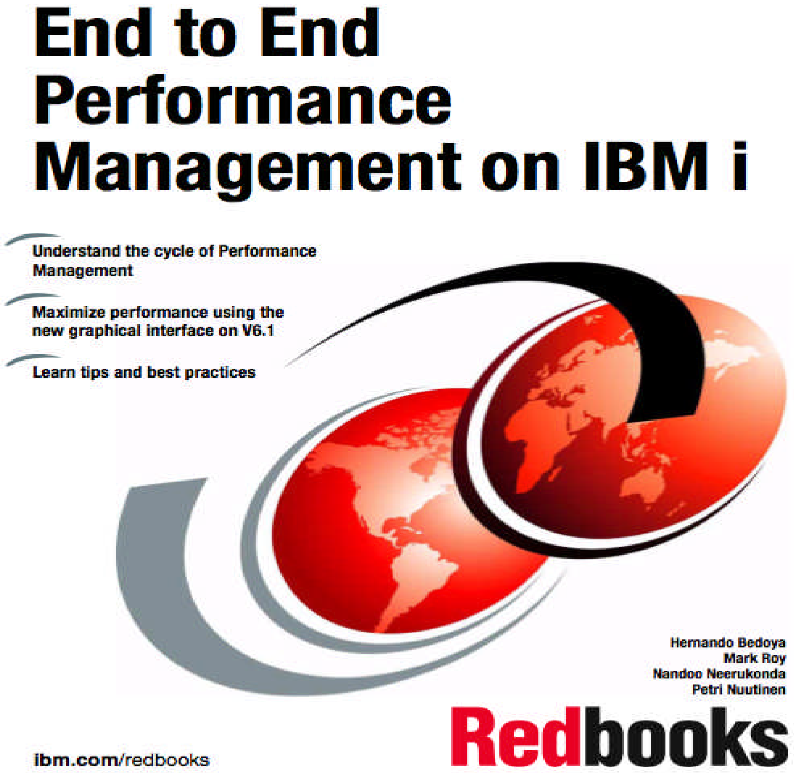
IBM i on Power - Performance FAQ

October 9, 2017

IBM i Web Sites with Performance Information

- IBM Knowledge Center:
  - 7.2 Performance
  - 7.3 Performance
- IBM i Performance Management:
  i Performance Management
- developerWorks:
  - IBM i Performance Tools: developerWorks Performance Tools
  - IBM i Performance Data Investigator: developerWorks PDI
- IBM iDoctor for IBM i: iDoctor
- IBM i Wait Accounting information:
  - Job Waits Whitepaper
  - KnowledgeCenter: The basics of Wait Accounting
  - developerWorks: IBM i Wait Accounting

A Redbooks publication!



End to End Performance Management on IBM i

- Understand the cycle of Performance Management
- Maximize performance using the new graphical interface on V6.1
- Learn tips and best practices

Hernando Bedoya
Mark Roy
Nandoo Neerukonda
Petri Nuutinen

Redbooks

ibm.com/redbooks

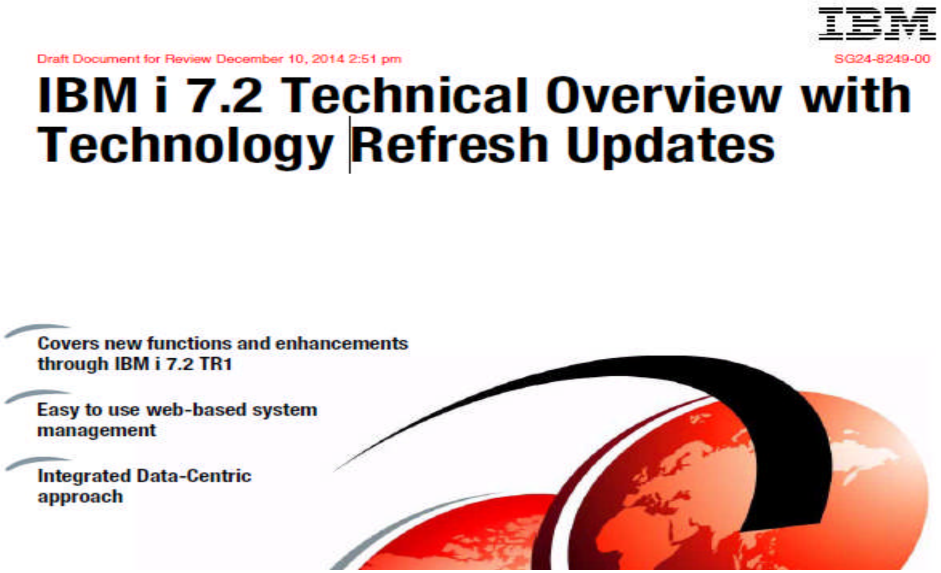
http://www.redbooks.ibm.com/redbooks/pdfs/sg247808.pdf

IBM i 7.2 Technology Refresh Updates

Covers the 7.2 content through Technology Refresh 1

Section 2.8 – Performance

Section 8.6.7 – Job level SQL stats in Collection Services



IBM

Draft Document for Review December 10, 2014 2:51 pm

SG24-8249-00

IBM i 7.2 Technical Overview with Technology Refresh Updates

Covers new functions and enhancements through IBM i 7.2 TR1

Easy to use web-based system management

Integrated Data-Centric approach

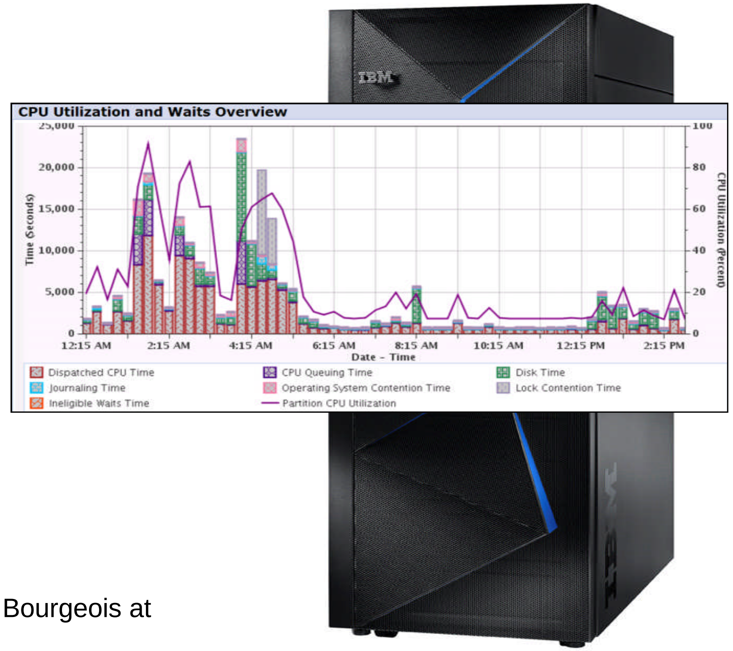
IBM i Performance Analysis Workshop

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
- 3-4 day workshop, public or private (on-site)
  - For general public workshop availability and enrollment: IBM i Performance Analysis Workshop
  - ★ For public workshop availability and enrollment in France, please contact Philippe Bourgeois at pbourgeois@fr.ibm.com or Françoise Laurens at f_laurens@fr.ibm.com
  - For additional information, including private workshops, please contact Eric Barsness at ericbar@us.ibm.com or Stacy Benfield at stacylb@us.ibm.com, members of Systems Lab Services

IBM Systems Lab Services Power Systems Delivery Practice - **ibm**.com/systems/services/labservices - ibmsls@us.ibm.com

IBM i Performance and Optimization Services

The IBM i Performance and Optimization team specializes in resolving a wide variety of performance problems. Our team of experts can help you tune your partition and applications, including:

- Reducing batch processing times
- Resolving SQL query and native IO performance problems
- Tuning RPG, COBOL, C, and Java (including WebSphere Application Server) programs
- Removing bottlenecks, resolving intermittent issues
- Resolving memory leaks, temporary storage growth problems, etc.
- Tuning memory pools, disk subsystems, system values, and LPAR settings for best performance
- Optimizing Solid State Drive (SSD) performance
- Tuning client interfaces such as ODBC, JDBC, .Net and more

Skills transfer and training for performance tools and analysis also available!

Contact Eric Barsness at ericbar@us.ibm.com for more details.

www.ibm.com/systems/services/labservices

And finally…..

Thank you

Don't forget to fill-in the feedback form!

thirty
years

i
for Business

ithankyou

www.ibm.com/power/i

Special notices

This document was developed for IBM offerings in the United States as of the date of publication. IBM may not make these offerings available in other countries, and the information is subject to change without notice. Consult your local IBM business contact for information on the IBM offerings available in your area.

Information in this document concerning non-IBM products was obtained from the suppliers of these products or other public sources. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. Send license inquires, in writing, to IBM Director of Licensing, IBM Corporation, New Castle Drive, Armonk, NY 10504-1785 USA.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

The information contained in this document has not been submitted to any formal IBM test and is provided "AS IS" with no warranties or guarantees either expressed or implied.

All examples cited or described in this document are presented as illustrations of the manner in which some IBM products can be used and the results that may be achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions.

IBM Global Financing offerings are provided through IBM Credit Corporation in the United States and other IBM subsidiaries and divisions worldwide to qualified commercial and government clients. Rates are based on a client's credit rating, financing terms, offering type, equipment type and options, and may vary by country. Other restrictions may apply. Rates and offerings are subject to change, extension or withdrawal without notice.

IBM is not responsible for printing errors in this document that result in pricing or information inaccuracies.

All prices shown are IBM's United States suggested list prices and are subject to change without notice; reseller prices may vary.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

Any performance data contained in this document was determined in a controlled environment. Actual results may vary significantly and are dependent on many factors including system hardware configuration and software design and configuration. Some measurements quoted in this document may have been made on development-level systems. There is no guarantee these measurements will be the same on generally-available systems. Some measurements quoted in this document may have been estimated through extrapolation. Users of this document should verify the applicable data for their specific environment.

Special notices (cont.)

IBM, the IBM logo, ibm.com AIX, AIX (logo), AIX 6 (logo), AS/400, BladeCenter, Blue Gene, ClusterProven, DB2, ESCON, i5/OS, i5/OS (logo), IBM Business Partner (logo), IntelliStation, LoadLeveler, Lotus, Lotus Notes, Notes, Operating System/400, OS/400, PartnerLink, PartnerWorld, PowerPC, pSeries, Rational, RISC System/6000, RS/6000, THINK, Tivoli, Tivoli (logo), Tivoli Management Environment, WebSphere, xSeries, z/OS, zSeries, AIX 5L, Chiphopper, Chipkill, Cloudscape, DB2 Universal Database, DS4000, DS6000, DS8000, EnergyScale, Enterprise Workload Manager, General Purpose File System, , GPFS, HACMP, HACMP/6000, HASM, IBM Systems Director Active Energy Manager, iSeries, Micro-Partitioning, POWER, PowerExecutive, PowerVM, PowerVM (logo), PowerHA, Power Architecture, Power Everywhere, Power Family, POWER Hypervisor, Power Systems, Power Systems (logo), Power Systems Software, Power Systems Software (logo), POWER2, POWER3, POWER4, POWER4+, POWER5, POWER5+, POWER6, POWER6+, System i, System p, System p5, System Storage, System z, Tivoli Enterprise, TME 10, Workload Partitions Manager and X-Architecture are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml

The Power Architecture and Power.org wordmarks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

UNIX is a registered trademark of The Open Group in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries or both.

Microsoft, Windows and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries or both.

Intel, Itanium, Pentium are registered trademarks and Xeon is a trademark of Intel Corporation or its subsidiaries in the United States, other countries or both.

AMD Opteron is a trademark of Advanced Micro Devices, Inc.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

TPC-C and TPC-H are trademarks of the Transaction Performance Processing Council (TPPC).

SPECint, SPECfp, SPECjbb, SPECweb, SPECjAppServer, SPEC OMP, SPECviewperf, SPECapc, SPEChpc, SPECjvm, SPECmail, SPECimap and SPECsfs are trademarks of the Standard Performance Evaluation Corp (SPEC).

NetBench is a registered trademark of Ziff Davis Media in the United States, other countries or both.

AltiVec is a trademark of Freescale Semiconductor, Inc.

Cell Broadband Engine is a trademark of Sony Computer Entertainment Inc.

InfiniBand, InfiniBand Trade Association and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.

Other company, product and service names may be trademarks or service marks of others.

End of Presentation material…..