

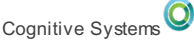


# IBM i Performance Tools for Application Developers

Dawn May – [dmmay@us.ibm.com](mailto:dmmay@us.ibm.com)  
@DawnMayiCan



© 2017 IBM Corporation



---

## Performance Disclaimer

- “it depends ...”
- Performance information and recommendations in this presentation are based on measurements, analysis, and projections in a controlled environment 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.


© 2017 IBM Corporation

2


## Agenda


- Brief Overview
- Introduction to IBM i Wait Accounting
  
- Performance Data Collectors
  - Collection Services
  - Job Watcher
  - Performance Explorer
  
- Performance Data Visualization and Diagnostics
  - Performance Data Investigator
  - iDoctor
  
- Examples

## Keep Current on PTFs

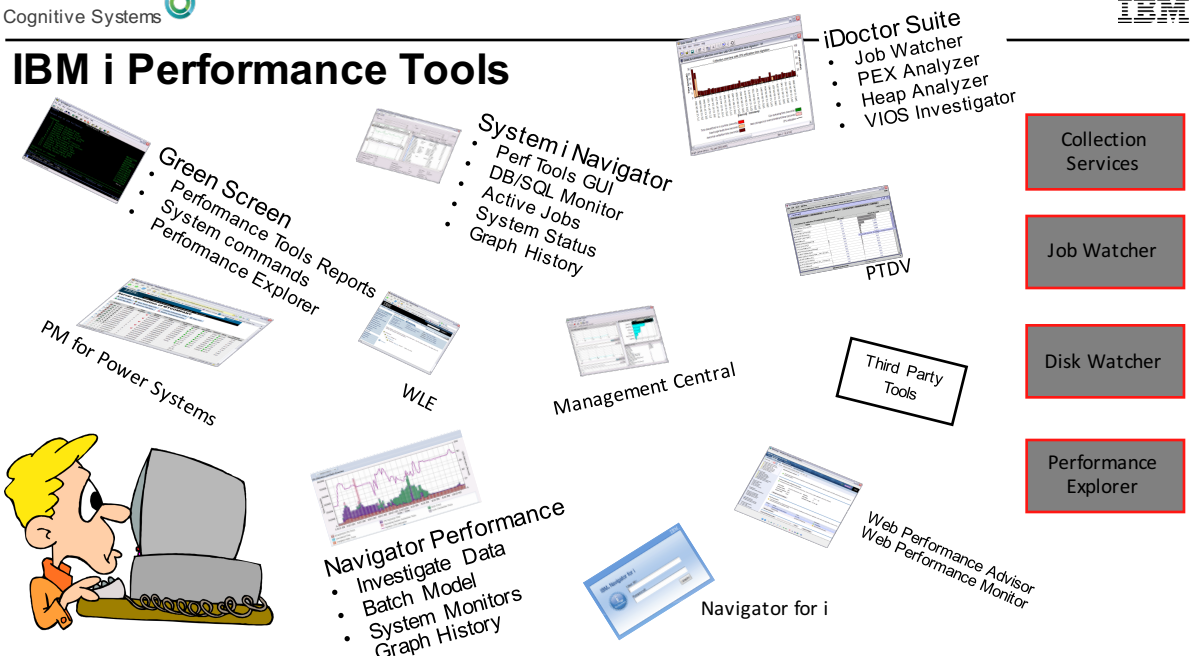
 It's always good practice to keep current on the latest fixes from IBM

- PTFs address defects
- PTFs introduce new capabilities
  - IBM i Technology Refresh Updates
  - IBM i Group PTFs
    - Database
    - Performance tools
    - Java
    - HTTP Server
      - HTTP Server Group PTF for latest Navigator for i functionality
  - PTFs for performance data collectors
    - Collection Services, Job Watcher, Disk Watcher, Performance Explorer

Cognitive Systems 





## IBM i Performance Tools




- Green Screen**
  - Performance Tools Reports
  - System commands
  - Performance Explorer
- System i Navigator**
  - Perf Tools GUI
  - DB/SQL Monitor
  - Active Jobs
  - System Status
  - Graph History
- iDoctor Suite**
  - Job Watcher
  - PEX Analyzer
  - Heap Analyzer
  - VIOS Investigator
- PTDV**
- Third Party Tools**
- Management Central**
- WLE**
- PM for Power Systems**
- Navigator Performance**
  - Investigate Data
  - Batch Model
  - System Monitors
  - Graph History
- Navigator for i**
- Web Performance Advisor**
- Web Performance Monitor**
- Collection Services**
- Job Watcher**
- Disk Watcher**
- Performance Explorer**

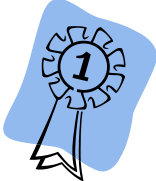
© 2017 IBM Corporation 5

Cognitive Systems 



## Performance Instrumentation and Data Collection

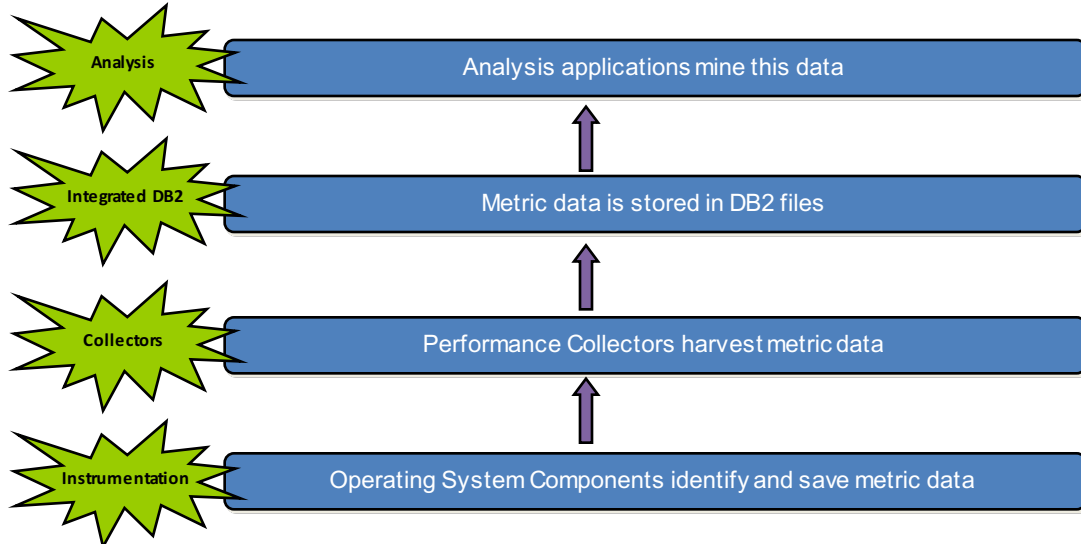
The  Advantage



- IBM **develops the software stack**, top to bottom
  - Instruments the software with component-specific performance metrics
- IBM develops the **performance data collectors** that harvest those performance metrics
- IBM i has an **integrated database** – DB2
  - This is a BIG DEAL
  - Performance data is stored in the database automatically
    - No “add on” application is necessary – it’s all in the Operating System
- IBM provides the **graphical analysis tools**
  - Analysis of the performance data in the DB2 files using SQL

© 2017 IBM Corporation **IBM i has the best performance instrumentation and data collection capabilities in the industry!** 6

## Performance Instrumentation and Data Collection



## Introduction to Wait Accounting

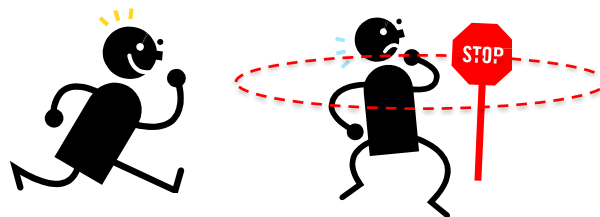
## Performance Fact

*“All computers wait at the same speed”*



## What is Wait Accounting?

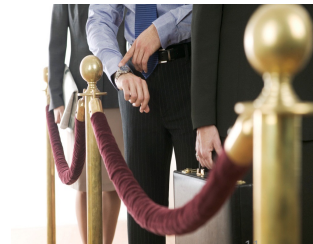
- Wait Accounting is the ability to determine what a job is doing when it is *not* running



i Exclusive! Patented technology built into IBM i.

## Wait Accounting Overview

- When a job is not running (i.e., not using CPU), it is waiting
  - *But what is it waiting for?*
- Waits may be normal, some waits are not normal
  - Wait Accounting helps to determine what the wait is and if it is a problem
- IBM i has instrumented most of the key wait conditions
  - Wait information is automatically collected by **Collection Services** and **Job Watcher**



## Wait States

- Wait information is tracked for each job, thread and task on system
- A job/thread/task is in one of three states:

### Using CPU

- “Dispatched CPU”  
Assigned to a virtual processor so it can begin execution of instructions

### Waiting for CPU

- “CPU Queuing”  
Ready to use processor, but waiting for it to become available

### Waiting for something else...

- Idle waits
- **Blocked waits**

These waits are typically the most interesting waits to focus on

## Types of Waits

- Idle waits
  - Typically a normal wait condition; for example:
    - Waiting for the “Enter” key to be pressed on a 5250 display session
    - Waiting for data from the network
- **Blocked waits**
  - Waits that occur while performing a work request
  - Blocked waits are what we want to look at more closely



## Wait Accounting - Buckets

- **Wait Buckets** = “Wait condition groups” instrumented in the operating system.
  - **Buckets** can then be **analyzed** to determine where a job is spending its time (running or waiting)
  - Categorized into **32** buckets
  - Buckets found in both **Collection Services** and **Job Watcher** data
  - Waits can be viewed at a **system-level** or at an **individual job/thread/task level**
    - Can also be grouped by generic job name, subsystem, user profile, pool ID, etc.



## 32 Wait Buckets

1. Time dispatched on a CPU
2. CPU queuing
3. Reserved
4. Other waits
5. Disk page faults
6. Disk non-fault reads
7. Disk space usage contention
8. Disk operation start contention
9. Disk writes
10. Disk other
11. Journaling
12. Semaphore contention
13. Mutex contention
14. Machine level gate serialization
15. Seize contention
16. Database record lock contention
17. Object lock contention
18. Ineligible waits
19. Main storage pool contention
20. Classic Java™ user including locks (to 6.1)
  - (7.2) Journal save while active
21. Classic Java JVM (up to 6.1), now reserved
22. Classic Java other (up to 6.1), now reserved
23. Reserved
24. Socket transmits
25. Socket receives
26. Socket other
27. IFS
28. PASE
29. Data queue receives
30. Idle/waiting for work
31. Synchronization Token contention
32. Abnormal contention

<http://www.ibm.com/developerworks/ibmi/library/i-ibmi-wait-accounting/>  
[http://public.dhe.ibm.com/services/us/gsc/idoctor/job\\_Waits\\_White\\_Paper\\_61\\_71.pdf](http://public.dhe.ibm.com/services/us/gsc/idoctor/job_Waits_White_Paper_61_71.pdf)

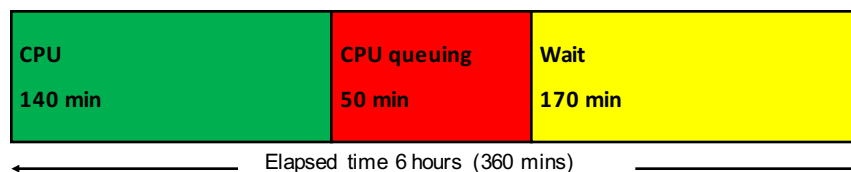
## Wait Accounting – “Run-wait” signature

Applying the concepts of wait accounting, we are now able to identify the amount of time the thread/task was running and the time the thread/task was waiting.

Consider the following:

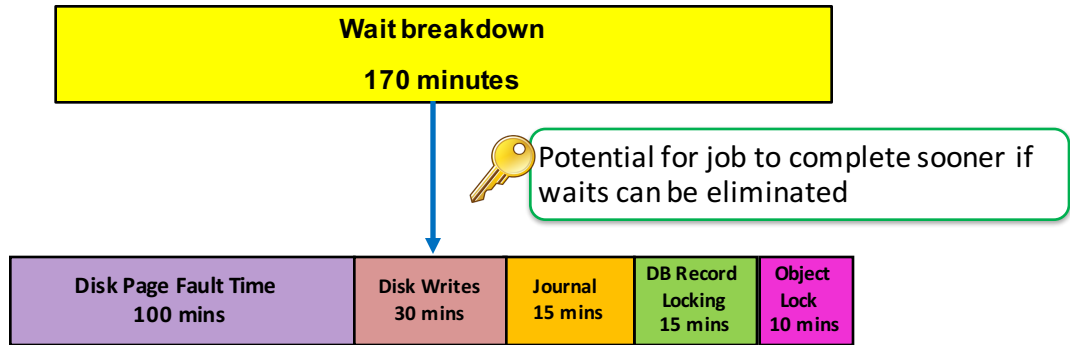
Batch job with total run time of 6 hours

Run-wait signature





## Wait Accounting – “Run-wait” signature



Now you can start asking questions such as:

- Are my pool sizes appropriate? What objects is the faulting occurring on?
- Is the write cache being overrun? Is the application forcing writes out synchronously?
- Are all the journals optimally configured? Are unnecessary objects being journaled?
- Am I locking records or objects unnecessarily?



## Metrics related to components of wait time

Total count	Disk reads 3,523	Disk writes 17,772	Record Locks 355	Journal 5,741
Total time	42 sec	73 sec	45 sec	44 sec
Avg time per wait	0.012 sec	0.004 sec	0.126 sec	0.007 sec

- Tools capture this level detail
- Useful to know both counts and time

Disk Page Faults Counts	Disk Page Faults Time (Seconds)
20498	26.37
7969	10.83
9080	15.86
1738	2.55
3124	1.22
1071	0.65
2013	0.94
3133	1.19
2939	1.36
1056	0.55

## Why developers should leverage wait accounting!!

- Helps you understand application characteristics
  - Is it CPU bound? I/O bound?
- Helps you to understand where to focus your effort and investment
  - Is there a bottleneck on CPU, Memory, I/O, Contention time?
  - Invest resources where greatest benefit will be
- Can offer insight into potential performance issues before end-users are affected
  - Can leverage aspects of wait accounting in test environment
    - Eliminate surprises
    - Identify bottlenecks that prevent scaling
- Provides valuable clues to help analyze performance issues as they arise
- Instrumentation part of base IBM i operating system, IBM tools available to help you analyze



## Common Waits that Applications use

- Disk Waits
- Semaphores, Mutexes, Synchronization Tokens
- Journaling
- Database record locks
- Object locks
- Sockets

## A few other things to know about waits...

- Some waits are “expected” and others “unexpected”
  - A record lock may be expected, but one that lasts for a very long duration is unexpected
- If waits can be reduced or eliminated, CPU can be used more efficiently
- One wait may be reduced/eliminated, only to have another wait surface
- Likely won’t be able to remove all wait times
- When is a wait “bad”?
  - Is there a business impact? Are users complaining?
  - “It depends” but **waits more than 25% of run time** may need additional analysis



## Tools for analyzing Wait Accounting information



## Wait Accounting - IBM i Performance Data Collectors

- **Collection Services**

- Collects data automatically 24X 7 at specified intervals (typically 5 or 15 minutes)
- System and job level data
- Starting point!



- **Job Watcher**

- Needs to be started/stopped (typically 5 or 10 second intervals)
- Additional detailed data such as call stacks, object waited on, holder
- Frequently needed to solve performance issues



## Performance Data

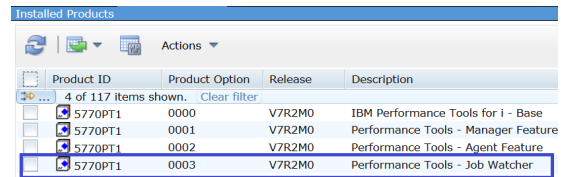
### *Collection vs Visualization*

- The performance data **collection** capability is built into the operating system
  - **Everyone** can collect all types of IBM i performance data
- The **visualization** of the data may require additional products
  - Visualization is the display of charts or tables of performance data
  - The Performance Data Investigator is part of the operating system
    - **Everyone** can visualize Collection Services data



## Two Graphical Analysis Tools

- Performance Data Investigator – Job Watcher
  - Requires 57xx-PT1 – Job Watcher feature
  - Geared to *average* user
  
- iDoctor – Job Watcher
  - IBM i Service offering, yearly license by serial number
  - Geared to *advanced* user



Product ID	Product Option	Release	Description
4 of 117 items shown. Clear filter			
5770PT1	0000	V7R2M0	IBM Performance Tools for i - Base
5770PT1	0001	V7R2M0	Performance Tools - Manager Feature
5770PT1	0002	V7R2M0	Performance Tools - Agent Feature
5770PT1	0003	V7R2M0	Performance Tools - Job Watcher

## Wait Accounting IBM i Graphical Analysis Tools

- Two powerful graphical tools to help make your analysis more efficient and productive:

### Performance Data Investigator (PDI)

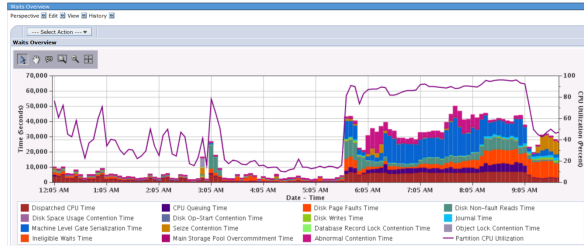
- Component in IBM Navigator for i (browser-based)
- Nothing to install, can view Collection Services for free
- <http://www.ibm.com/developerworks/ibmi/library/i-pdi/index.html>

### IBM iDoctor for IBM i

- Microsoft Windows based client
- Requires Job Watcher yearly license to see Collection Services data (IBM Service offering)
- [https://www-912.ibm.com/i\\_dir/idoctor.nsf](https://www-912.ibm.com/i_dir/idoctor.nsf)

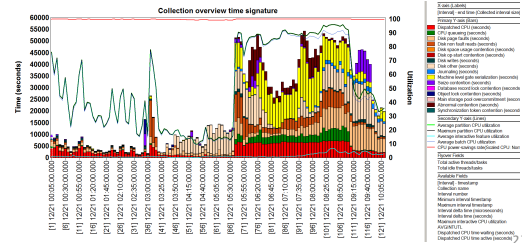
# Wait Accounting IBM i Graphical Analysis Tools

- Both graphical tools sit on top of same rich IBM i instrumentation, but not equivalent in presentation and function



← PDI

iDoctor ->

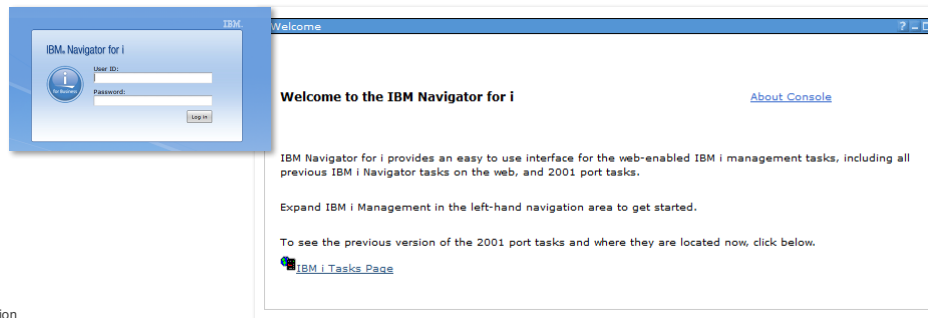


# Wait Accounting Analysis Strategy

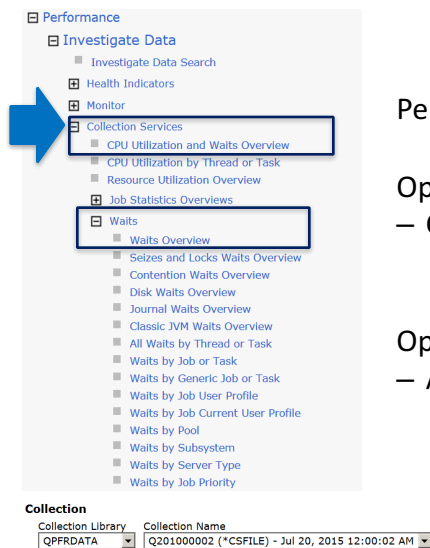
- Understand the “big picture” first
  - Understand overall partition characteristics first and where system bottlenecks may be that affecting your application
  - Typically done using **Collection Services** data
  - Drill down to job level
  - Waits can be analyzed in various useful ways:
    - Waits by Job or Task
    - Waits by Generic Job or Task
    - Waits by Job User Profile
    - Waits by Job Current User Profile
    - Waits by Pool
    - Waits by Subsystem
    - Waits by Server Type
    - Waits by Job Priority
- Continue detailed analysis at a Job Level using **Job Watcher**
  - Narrow focus to interesting timeframes / jobs
  - Many more job level details available

## Using Performance Data Investigator (PDI)

- IBM Navigator for i is the Web console for managing IBM i
  - Has much of the function as System i Navigator
    - but with a browser user interface
  - Simply point your browser to <http://systemname:2001>



## PDI Wait Accounting Perspectives - Where to start



Performance -> Investigate Data -> Collection Services:

Option 1: **CPU Utilization and Waits Overview**

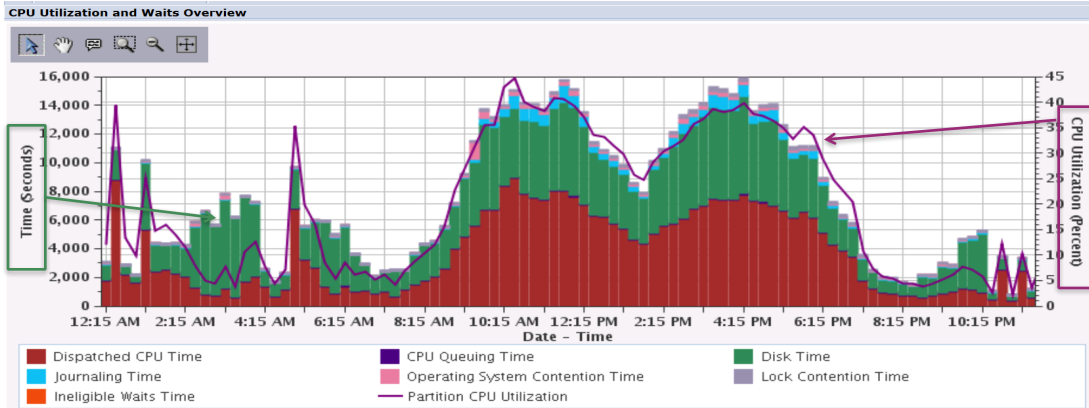
– Combines related waits into higher level buckets

Option 2: **Waits Overview**

– All individual “blocked” wait buckets shown

# System “run-wait” signature

## CPU Utilization and Waits Overview



- A chart that combines CPU utilization as well as the wait buckets can be very beneficial in assessing the health of your partition
- In this chart, we can see that the majority of the time, the jobs were spending time in CPU as well as in Disk. Minor amounts of Journal wait time and operating system contention time are also present.

31

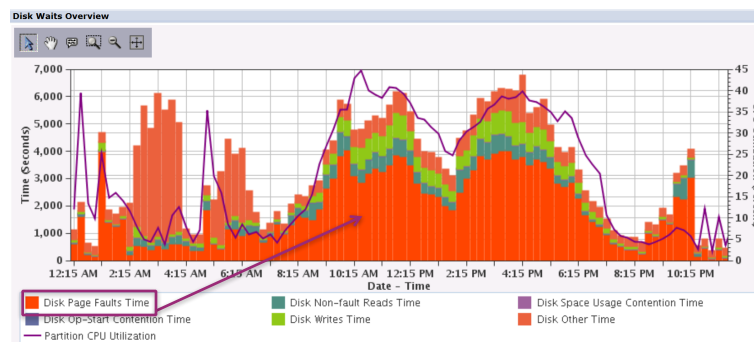
# CPU Utilization and Waits Overview

## Drilldown analysis

--- Select Action ---

- Waits Overview
- Seizes and Locks Waits Overview
- Contention Waits Overview
- Disk Waits Overview**

Because Disk wait time was fairly significant, drilldown to **Disk Waits Overview** to further examine the detailed waits contributing to this time:

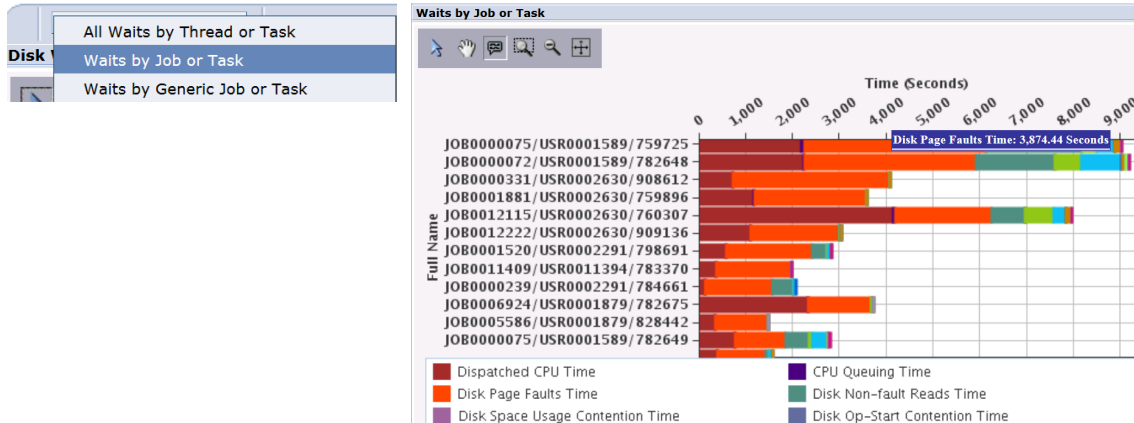


We can see that Disk Page Fault time is the biggest contributor to Disk Time.  
 (A job needed something in memory, it wasn't there, had to do an I/O to bring it into memory before job could continue running).



## Waits by Job or Task

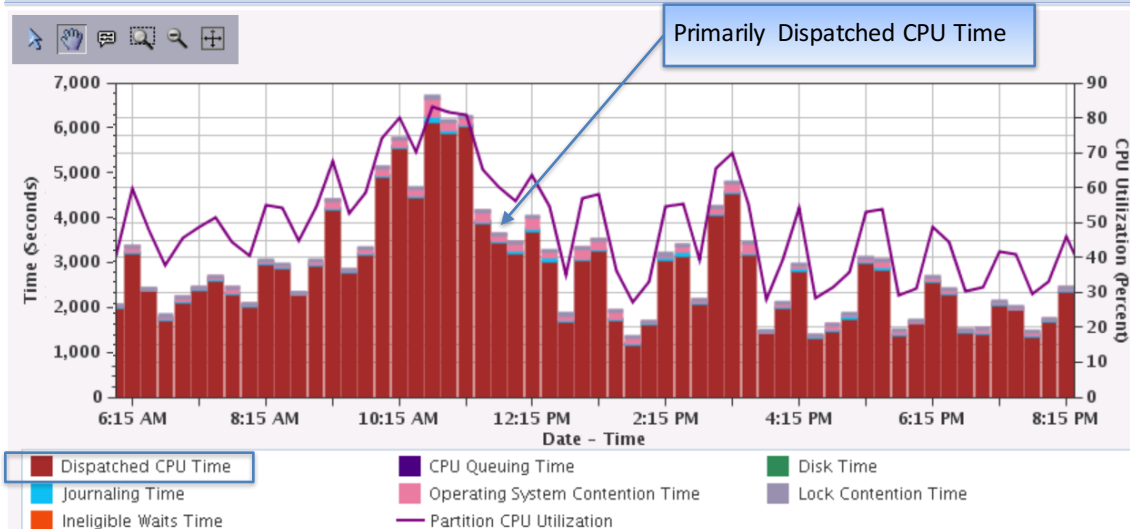
The next question likely would be which job(s) are incurring this wait time. Drilling down further, we can see the list of jobs incurring this wait time:



This type of chart can also be used to understand a job(s) "run-wait" signature.

## Efficient System with Little Waiting

CPU Utilization and Waits Overview

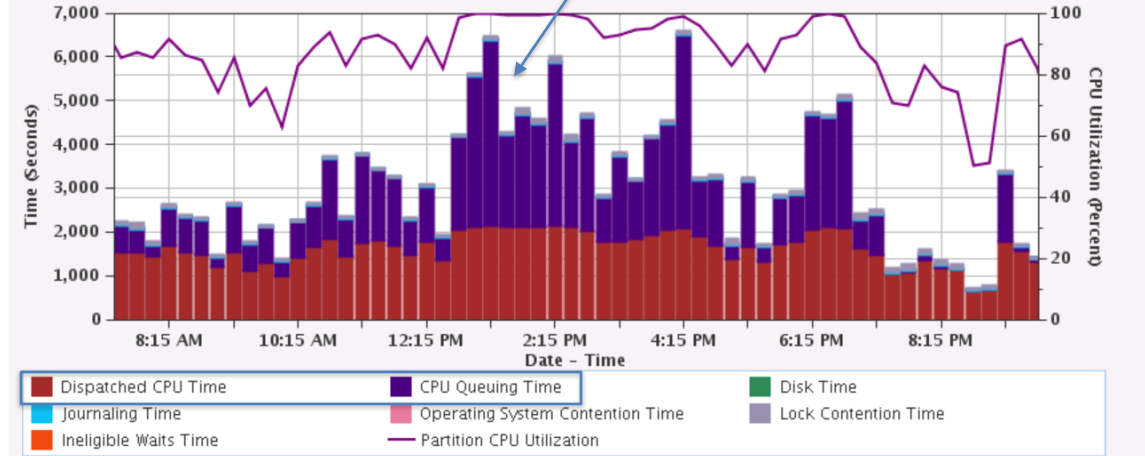


## Processor Bound System

CPU Utilization and Waits Overview



Dispatched CPU  
+  
CPU Queuing

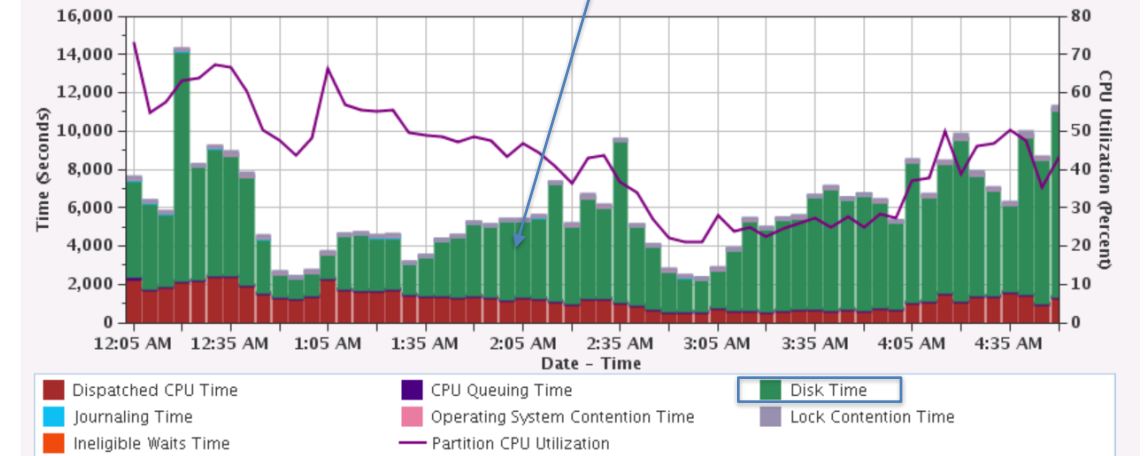


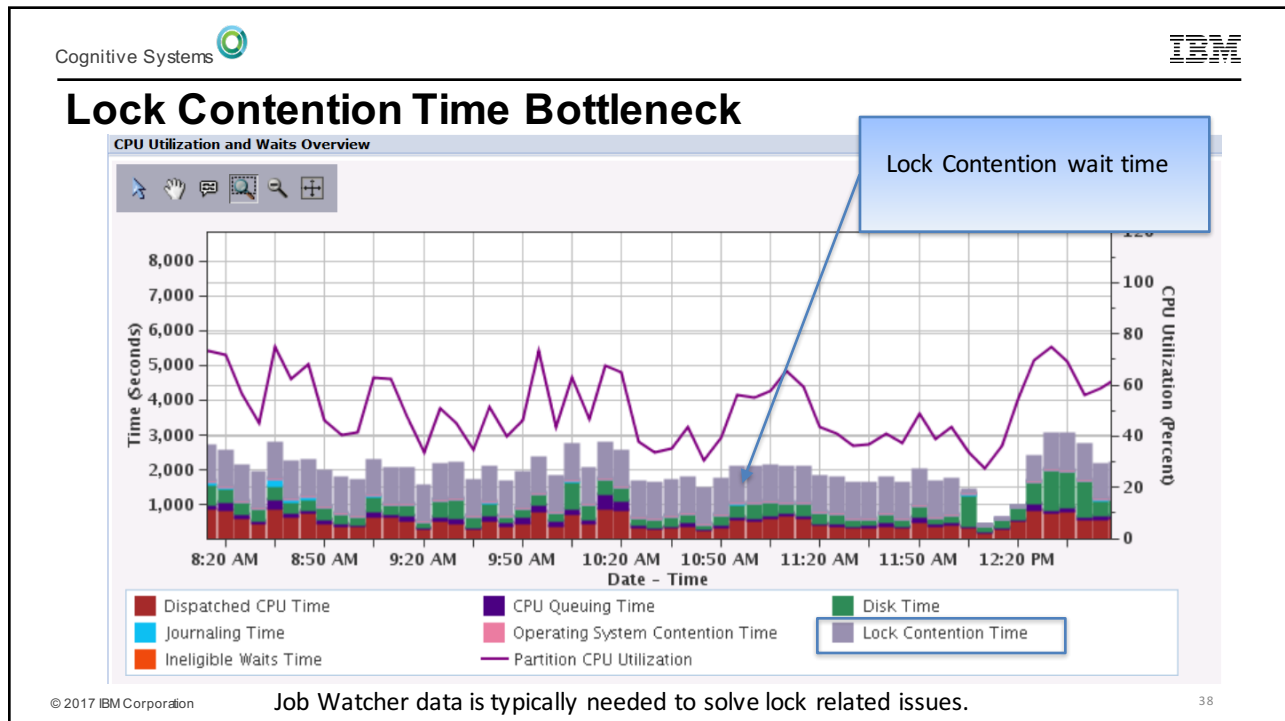
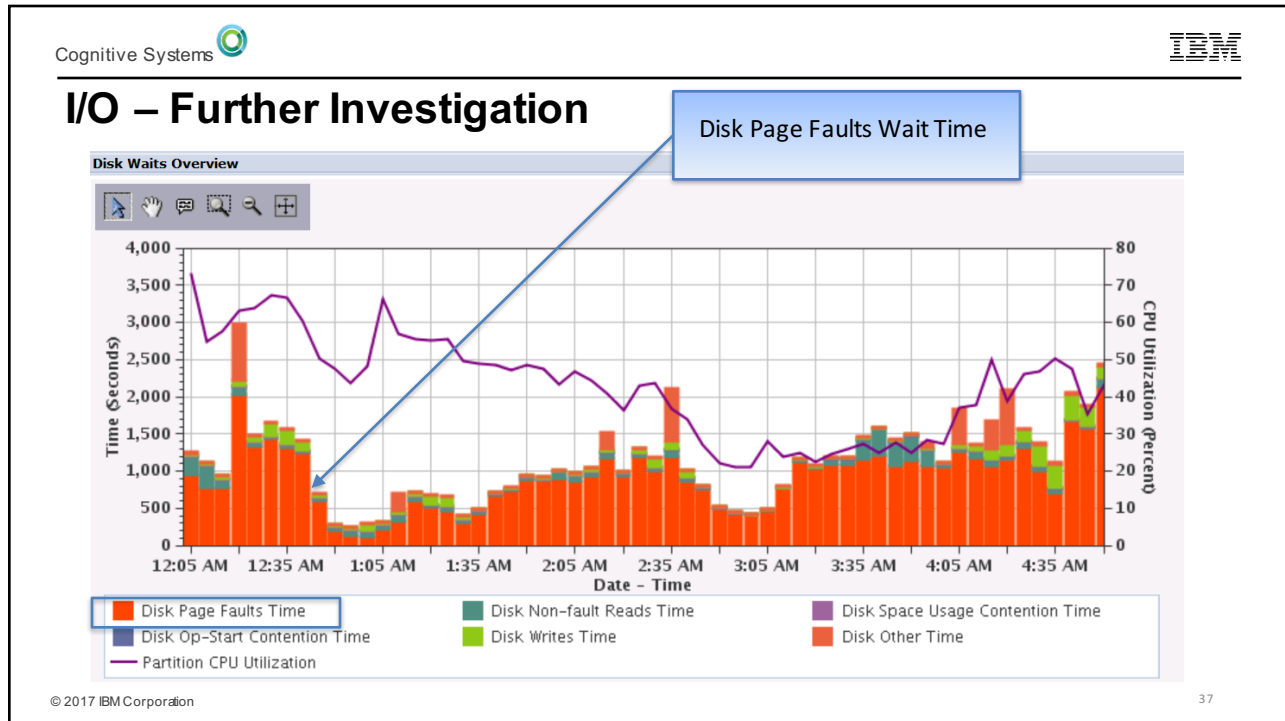
## I/O Bound System


CPU Utilization and Waits Overview



Disk Time





Cognitive Systems 

## Lock – Further Investigation

--- Select Action ---


CPU U

- Waits Overview
- Seizes and Locks Waits Overview
- Contention Waits Overview

The chart displays 'Seizes and Locks Waits Overview' with two y-axes: 'Time (Seconds)' on the left (0 to 1,400) and 'CPU Utilization (Percent)' on the right (0 to 120). The x-axis shows 'Date - Time' from 8:20 AM to 12:50 PM. A purple line represents 'Partition CPU Utilization', and a blue area represents 'Object Lock Contention Time'. A blue callout box points to the blue area with the text 'Object Lock Contention time'.

Job Watcher data will show object waited on, the holder, and call stacks for both the waiter and the holder (example shown later on...)

© 2017 IBM Corporation 39

Cognitive Systems 

## Wait Accounting at a Job Level

Would this job benefit from additional memory?

The bar chart shows 'Wait Accounting at a Job Level' for job 'PRICER/JSMITH/010628'. The x-axis is 'Time (Seconds)' from 0 to 25,000. The bar is composed of several colored segments representing different wait components. A legend below the chart lists the components:

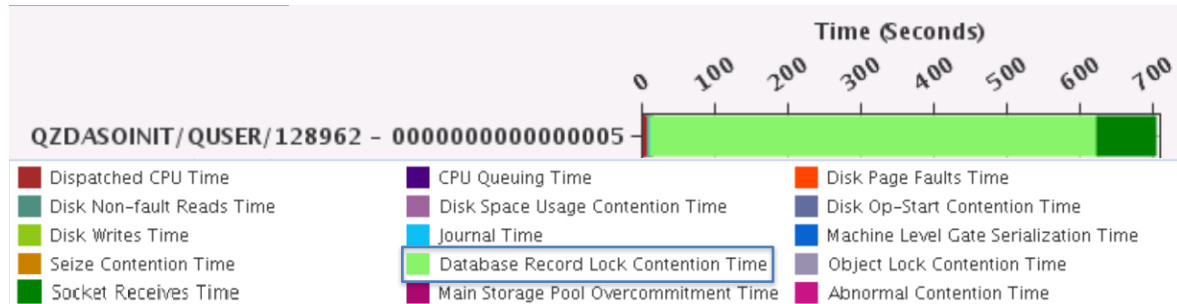
- Dispatched CPU Time
- CPU Queuing Time
- Disk Page Faults Time
- Disk Non-fault Reads Time
- Disk Space Usage Contention Time
- Disk Op-Start Contention Time
- Disk Writes Time
- Journal Time
- Machine Level Gate Serialization Time
- Seize Contention Time
- Database Record Lock Contention Time
- Object Lock Contention Time
- Socket Receives Time
- Main Storage Pool Overcommitment Time
- Abnormal Contention Time

© 2017 IBM Corporation 40

## Wait Accounting at a Job Level



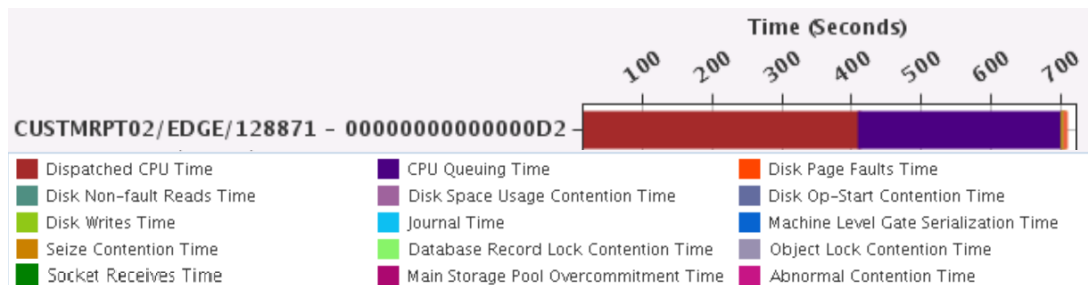
Would this job benefit from additional memory? CPU? Disk?



## Wait Accounting at a Job Level



Would this job benefit from an improved I/O subsystem?

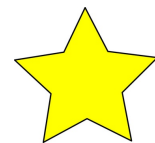


## Job Watcher - Additional Benefits

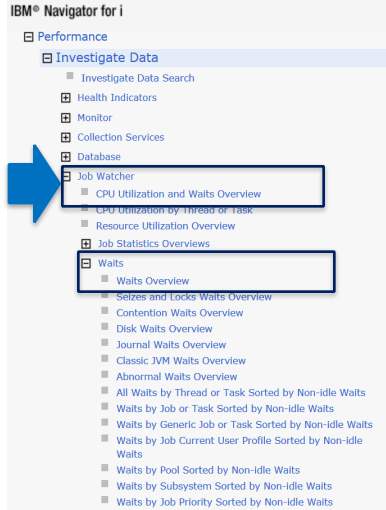
- Collects **more detailed** performance data than Collection Services
  - Call Stacks
  - SQL Statements
  - Additional wait accounting information:
    - Objects being waited on
    - Holder of object
- **More frequent** intervals (seconds)
- Need to start/stop Job Watcher
  - Navigator for i, iDoctor, green screen commands
- To see charts in PDI, need Performance Tools LPP Job Watcher option (chargeable) or iDoctor Job Watcher license for viewing in iDoctor

## Job Watcher - Holders versus Waiters

- IBM i keeps track of who is holding a resource, and if applicable, who is waiting to access that resource
  - A **Holder** is the job/thread/task that is holding the serialized resource
  - A **Waiter** is the job/thread/task that wants to access the serialized resource
- IBM i also maintains call stacks for every job/thread/task
- The combination of
  - **Who** - holders and waiters ... *who has it? who wants it?*
  - **What** – object being waited on
  - **How** - call stacks
 provides a very powerful solution for analyzing wait conditions



## Job Watcher – Where to Start



Performance -> Investigate Data -> Job Watcher:

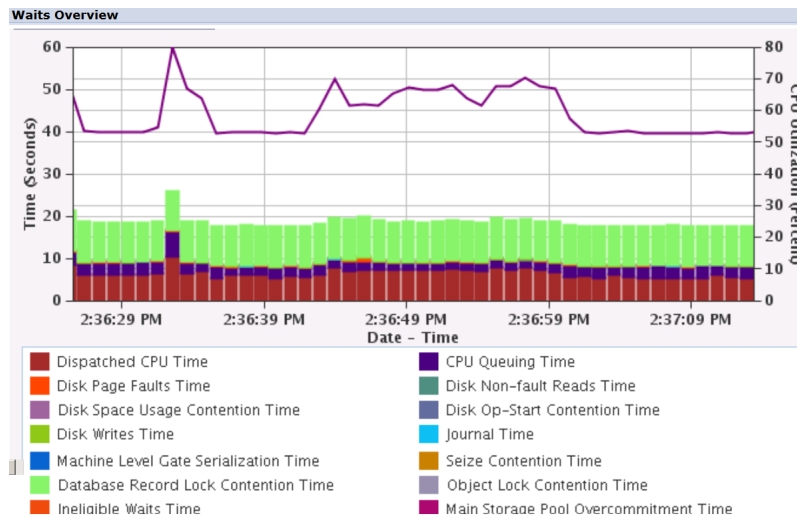
Option 1: **CPU Utilization and Waits Overview**  
 – Combines related waits into higher level buckets



Option 2: **Waits Overview**  
 – All individual “blocked” wait buckets shown

Notice similar perspectives available as Collection Services

## Job Watcher – Waits Overview

*Notice same wait buckets, but more granular intervals*



Cognitive Systems 


## Job Watcher – Additional Interval Details

**Thread or Task Details**

Job information: QZDASOINIT/QUSER/128962 - 0000000000000005

Current user profile: LISAW

Object waited on: INVENTORY INVENTORY

Wait duration: 581 milliseconds

Current or last wait: DB record lock: update

Holding job or task: QZDASOINIT/QUSER/128890

SQL client job: None detected this interval

[Show Holder](#)

Priority: 20

Pool: 2

Type description: PHYSICAL FILE MBR - DATA PART

Segment type description: DB PHYSICAL FILE MEMBER RECORDS

Wait object library: None detected this interval

Interval timestamp: Jan 3, 2014 2:36:28 PM

Interval (1 to 684):  [<](#) [>](#)

**Call Stack**



Call Level	Program	Module	Procedure
1			qutde_block_tra
2			longWaitReceive_9QuCounterFR12RmprReceiverPh
3			DBLockConflict_15RmslDBHashClassFR11RmslPlm
4			rmslDBHLock_FR11RmslPlmLad
5			getLockWithWait_18DbpmUpdateResourcecd
6			getLock_18DbpmUpdateResourcecd
7			getRowLock_18DbpmUpdateResourceFCUIRC9Dbp
8			execute_18DbpmUpdateLockNodeFR13DbpmQuer
9			vPositionNextAndExecute_18DbpmUpdateLockNod
10			positionNextEntryAndFetchOutline_17DbpmReadO

**SQL Statement**

Include Host Variables

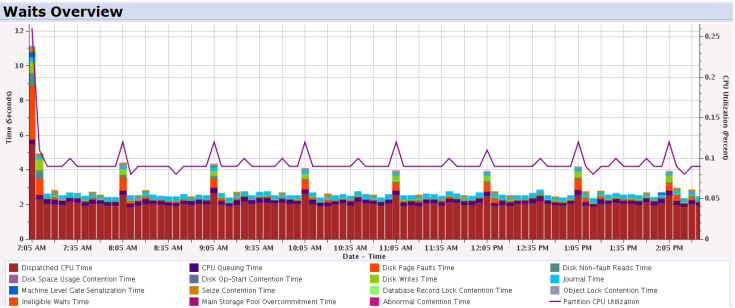
```
SELECT QUANTITY FROM WAREHSE42.INVENTORY WHERE ID=*DATA FORMAT ERRORTITY FROM WAREHSE42.INVENTORY WHERE ID=? FOR UPDATE
```

© 2017 IBM Corporation
47

Cognitive Systems 


## Wait Accounting: Be proactive!

- Use the rich IBM i wait accounting instrumentation found in
  - Collection Services & Job Watcher
  - Use PDI or iDoctor to view/analyze
- Understand your partition's "run-wait" signature and normal patterns
- Identify bottlenecks

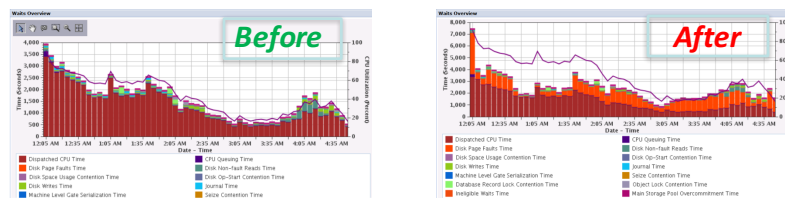


© 2017 IBM Corporation
48



## Recommendations: Be proactive!

- Keep a **baseline**
  - Collection Services (Job Watcher data is also nice to have)
    - Weekly, end-of-month, end-of-year
    - Prior to any hardware, software, configuration related change
- A baseline provides a **reference point**
  - It is the expected performance characteristics over a defined period of time
  - Having one makes it easier to recognize changes and its effect

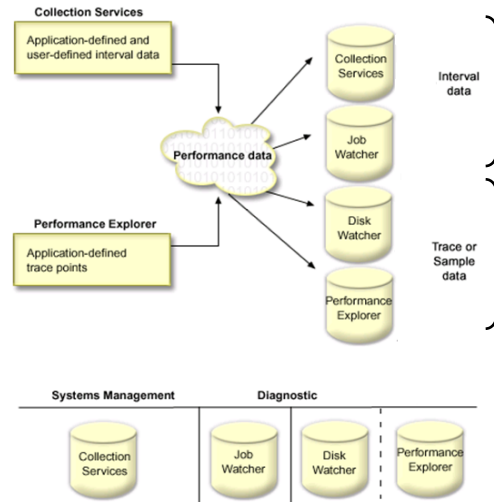


- **Wait bucket information can make it easier to determine what has changed! Both at a partition level as well as an individual job level**

## IBM i Performance Data Collectors

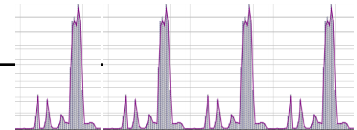
## Performance Data Collection Architecture

- Collection Services
- Job Watcher
- Disk Watcher
- Performance Explorer



## Collect System-wide Performance Data

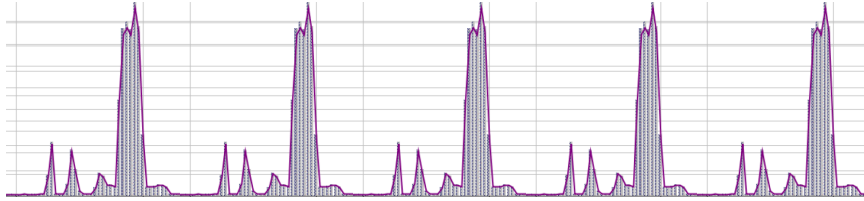
Collect Performance Data 24/7



If something goes wrong, you have data that will help **analyze** the problem, **fix** it, and **prevent** it from happening in the future

- If you can't solve the problem, you have information that makes it easier for IBM Support to **solve the problem faster**
- To provide a **reliable baseline** so you can **understand the impact** that a software, network, or environmental **change** had on the performance of your system
- To provide historical information that enables you to **plan for future growth** based on real trends, not guesses.

## Patterns in Performance Data



- Performance data typically has patterns
  - Daily, weekly, monthly, yearly



Understand your typical patterns

- Recognize change

## Job Watcher

## Job Watcher

- Job Watcher returns real-time information about a selected set of jobs, threads, or LIC tasks
- Job Watcher collects additional types of data that Collection Services does not, as well as more frequent intervals
  - Job Watcher has more overhead than Collection Services
- Data collected by Job Watcher includes
  - Wait times
  - CPU
  - I/O activity
  - Call Stacks
  - SQL statements
  - Communications statistics
  - Activation Group statistics



Run Job Watcher when you need detailed performance data for diagnostic purposes.

There are clients that run Job Watcher 24x7 to always have diagnostic data available.

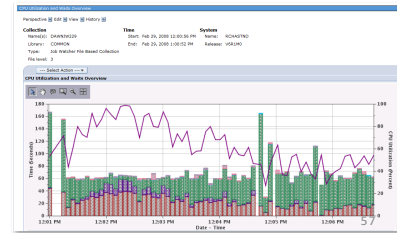
Need to manage the data carefully.

## Job Watcher

- Job Watcher collects **more detailed** performance data than Collection Services and at **more frequent intervals**
  - CPU and I/O (like Collection Services)
  - Call Stacks
  - SQL Statements
  - Detailed Wait information:
    - **Objects being waited on**, even records number of files
    - **Holder** of object
- Job Watcher does not collect everything that Collection Services collects.
  - It does not always collect information about every thread
    - Thread must use CPU during interval
    - Thread must exist for entire interval
  - It does not collect memory pool or detailed I/O statistics
- Data is written to DB2 files

## Job Watcher Usage Tips

- Use Job Watcher when you need detailed performance data to **resolve a problem**
  - Typically problem has been scoped first by Collection Services
  - Consider using a definition like the IBM-supplied Q10SECSQL (10 second intervals, gathers Call Stacks and SQL if it can) as a starting point
- For problem determination Job Watcher can be run on **specific jobs**
  - **Caution:** When using Job Watcher on specific job(s), you may not get detailed Holder information
- Multiple collections can be run at the same time
- Need to manage the amount of data collected
  - Specify Maximum Disk Space to consume on Start
  - Will stop collecting if specified ASP threshold is reached
  - iDoctor Monitors will clean up after themselves



© 2017 IBM Corporation

## Basic Job Watcher Data Collection Steps

- Select an an IBM-supplied **definition** to use
  - Or create your own custom definition
- **Start** the Job Watcher collection
- Let it **run** until the problem has occurred
- **Stop** the Job Watcher collection
- **Analyze** the data
- There may be times when you want to run Job Watcher for long periods of time
  - Consider 1 hour collection size
  - Manage size of data

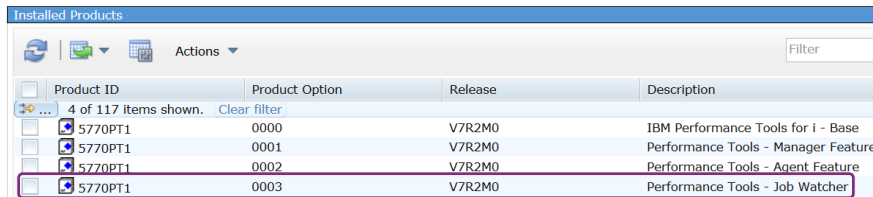


© 2017 IBM Corporation

58

## Job Watcher in Navigator for i

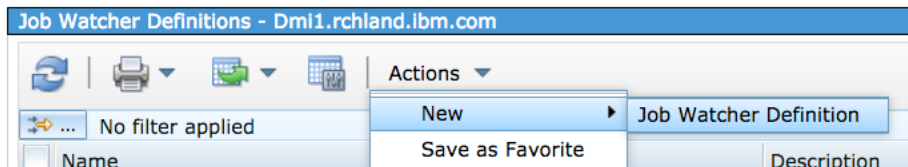
- Navigator for i Web console provides a GUI for Job Watcher
  - Requires the Performance Tools LPP, Job Watcher option to be installed



Product ID	Product Option	Release	Description
5770PT1	0000	V7R2M0	IBM Performance Tools for i - Base
5770PT1	0001	V7R2M0	Performance Tools - Manager Feature
5770PT1	0002	V7R2M0	Performance Tools - Agent Feature
5770PT1	0003	V7R2M0	Performance Tools - Job Watcher

## IBM-Supplied Job Watcher Definitions

- Many pre-defined Job Watcher definitions are available
  - The main difference is the sample interval
- Recommendations:
  - Collect with Call Stacks and SQL
  - Use 10 second intervals for general analysis (Q10secsql)
  - Use 5 second intervals for complex or intermittent issues, or for contention related problems (Q5secsql)



Name	Description
No filter applied	



Job Watcher Definitions - Dmi1.rchland.ibm.com

Actions

No filter applied

Name	Type	Description
<input type="checkbox"/> Q1sec	IBM-supplied	1 second intervals, Call stacks
<input type="checkbox"/> Q1secj	IBM-supplied	1 second intervals, Call stacks, J9
<input type="checkbox"/> Q1secsql	IBM-supplied	1 second intervals, Call stacks, Sql
<input type="checkbox"/> Q1secsqlj	IBM-supplied	1 second intervals, Call stacks, Sql, J9
<input type="checkbox"/> Q10sec	IBM-supplied	10 second intervals, Call stacks
<input type="checkbox"/> Q10secj	IBM-supplied	10 second intervals, Call stacks, J9
<input type="checkbox"/> Q10secsql	IBM-supplied	10 second intervals, Call stacks, Sql
<input type="checkbox"/> Q10secsqlj	IBM-supplied	10 second intervals, Call stacks, Sql, J9
<input type="checkbox"/> Q3minqzdas	IBM-supplied	QZDASOINIT jobs, 3 min intervals
<input type="checkbox"/> Q5minqzdas	IBM-supplied	QZDASOINIT jobs, triggers PEX stats
<input type="checkbox"/> Q5sec	IBM-supplied	5 second intervals, Call stacks
<input type="checkbox"/> Q5secj	IBM-supplied	5 second intervals, Call stacks, J9
<input type="checkbox"/> Q5secsql	IBM-supplied	5 second intervals, Call stacks, Sql
<input type="checkbox"/> Q5secsqlj	IBM-supplied	5 second intervals, Call stacks, Sql, J9

© 2017 IBM Corporation 61

Cognitive Systems  

## Job Watcher Definition

Add Job Watcher Definition

**Add Job Watcher Definition**

✓ Welcome

→ Basic Options

Advanced Options

Data Collection Options

Job Selection Options

Finish

**Basic Options**

\*Definition name:

Description:

**Data collection options**

Basic options

Include call stacks

Include SQL statements

In progress only

Last executed

Show advanced collection options

**Collection interval**

User-defined (0.1 - 3600.0)  seconds

No delay Data is collected as fast as possible. This is resource intensive.

Show advanced options

© 2017 IBM Corporation 62

**Add Job Watcher Definition**

Summary

Click Finish to create the definition defined below.

**Definition creation settings**

Definition name: Dawnmaydfn  
 Description: For my presentation  
 Collection interval: No delay  
 Call stack collection: CPU consumed, Conflict wait condition, Abnormal wait condition  
 SQL collection: Every interval, Last executed SQL statements, cursors, prepared statement arrays  
 Job selection: All  
 Task selection: All  
 Current user profile selection: None  
 Subsystem selection: None  
 Pool selection: None  
 Data availability: At end of collection  
 Collection file disk pool threshold: System  
 System disk pool threshold: System  
 Command string: Show

QSYS/ADDJWDFN DFN(DAWNMAJDFN) TEXT('For my presentation') COLITV(\*NODELAY) FRCRCD(\*CALC) TOASPTHLD(\*SYSTEM) SYSASPTHLD(\*SYSTEM) JOB(\*ALL) TASKNAME(\*ALL) INCALLFST(\*NO) ADDDTACGY((\*CALLSTACK 1) (\*SQLDETAIL 1)) WAITSTK((\*ABNWAIT 1) (\*CONFLICT 1))

OK

Cognitive Systems

**Start Job Watcher**

Basic Options

Specify the library and collection name for the data to be created. The collection name will be used to generate a matching member name in the Job Watcher files in the library specified.

Collection name:  Automatically generate using Julian date format (Qdddhhmms)  \*DawnDemo


Library name: DMMLIB

Description: Example of how to start Job Watcher

**Collection interval**

User-defined (0.1 - 3600.0) \*10 seconds  No delay Data is collected as fast as possible. This is resource intensive.



Cognitive Systems 

---

Start Job Watcher ? - □

**Start Job Watcher**


- ✓ Welcome
- ✓ Definition Selection
- ✓ Basic Options
- Termination
- Summary

**Termination**

Indicate below how the collection should end. At least one and up to all three of these options may be specified. The collection will end when one of the specified criteria has been met.


**Options (select one or more)**

- Maximum disk space to consume
- Maximum intervals to collect
- Maximum time to collect



Give some thought to these!

© 2017 IBM Corporation 65

Cognitive Systems 

---

Start Job Watcher ? - □

**Start Job Watcher**

- ✓ Welcome
- ✓ Definition Selection
- ✓ Basic Options
- ✓ Termination
- Summary

**Summary**

Click Finish to submit your request to start the Job Watcher collection.

**Collection creation settings**

Definition name: Dawnmaydfn

Definition properties:

Collection name: Dawndemo

Library name: Dmmlib

Description: Example of how to start Job Watcher

Collection interval: No delay

Maximum data to collect: 500 MB


Maximum intervals to collect: 1000

Maximum time to collect: 1 hours

Command string:





Start Job Watcher ? - □

**i** QSYS/SBMJOB CMD(CALL PGM(QSYS/QPYSTRCLTR) PARM(('QSYS/STRJW DFN(DAWNMAIDFN) COL(DAWNDEMO) LIB(DMMLIB) TEXT("Example of how to start Job Watcher") RPLDTA(\*YES) COLITV(\*NODELAY) ENDCOL((\*DASDMB 500) (\*NBRITV 1000) (\*NBRSEC 3600) ))' 179 'QIBM\_JW\_DMMLIB/DAWNDEMO')) JOB(QPYJWCOL) JOBD(QGPL/QDFTSVR) JOBQ(\*JOBQ) SYSLIB(\*SYSVAL) CURLIB(\*CRTDFT) INLLIB(\*JOBQ) DSPSBMJOB(\*NO) MSGQ(\*NONE) SRTSEQ(\*SYSVAL) LANGID(\*SYSVAL) CNTRYID(\*SYSVAL) CCSID(\*SYSVAL) INLSPGRP(\*JOBQ) ALWMLTTHD(\*NO)

Cognitive Systems  Job Watcher  
Active Job Watcher Collections

## Active Job Watcher Collections

Active Job Watcher Collections - Dmi1.rchland.ibm.com





 Actions


... No filter applied

Name	Library	Type	Status	Started
<input type="checkbox"/> DAWNDEMO	DMMLIB	Job Watcher File Based Collection	Active	4/28/17 6:10:27 PM

<input checked="" type="checkbox"/> DAWNDEMO	DMMLIB	Job Watcher File Based Collection
--	--------	-----------------------------------





- Stop
- Save
- Delete
- Move
- Copy
- Investigate Performance Data
- Properties

© 2017 IBM Corporation 67

Cognitive Systems  Job Watcher  
Active Job Watcher Collections

## Stop the Collection

Active Job Watcher Collections - Dmi1.rchland.ibm.com





 Actions

... No filter applied

Name	Library	Type	Status
<input checked="" type="checkbox"/> DAWNDEMO	DMMLIB	Job Watcher File Based Collection	Active





Stop Job Watcher

Indicate below the active Job Watcher collection which should be stopped. If the collection is not found or is not active then no action will be taken.

Collection name:

Library name:

Job Watcher Collections - Dmi1.rchland.ibm.com





 Actions

... No filter applied

Name	Library	Type	Status	Started
<input type="checkbox"/> DAWNDEMO	DMMLIB	Job Watcher File Based Collection	Complete	4/28/17
<input type="checkbox"/> DAWN JW2	COMMON	Job Watcher File Based Collection	Complete	3/12/08

## How Do I Run Job Watcher with the Commands?

- Add Job Watcher Definition (**ADDJWDFN**) to define the collection
  - Identifies the performance data that is to be collected
- Remove Job Watcher Definition (**RMVJWDFN**) to remove a definition  
(Note: Job Watcher Definitions can only be displayed through the GUI)
- Start Job Watcher (**STRJW**) to start the collection
- End Job Watcher (**ENDJW**) to end the collection (optionally)

## Running Job Watcher Continuously

- If you have a problem but Job Watcher is not running, you will need to recreate the problem to capture Job Watcher data
- Many clients prefer to run Job Watcher 24x7
  - Ensures you have diagnostic data the first time the problem occurs
- Options:
  - QMGTOOLS has the ability to run Job Watcher
  - iDoctor monitors
  - Your own CL program

## Running Job Watcher Continuously - example

- Create the objects and the job watcher data in a your own library – e.g., CRTLIB JOBWATCH
- Create a dataarea to control how many collections you want to have saved. In the STRJW command, we use the RPLDTA parameter to automatically replace the older collections with the same name. This way we don't have to worry about deleting older collections - they are simply overwritten when the new one with the same name is started.
- CRTDTAARA DTAARA(JOBWATCH/JOBWATCH) TYPE(\*DEC) LEN(15 5) VALUE(5)
- When the following program is called with the data area at 5 it will save 5 collections that have the names JWCOL1 .. to JWCOL5.
- \*Note: The sample program has a hardcoded "5" in it for the reset number when the counter goes to 1 -if you want to have more than 5 collections saved, you need to have that 5 changed in the little CLP. A simple improvement to this program would be to save the value retrieved from the data area and reset to that value rather than a hard-coded number
- In this example, we used the system supplied definition Q5SECSQL- you'll may want to set up your own definition and use it when you do the STRJW.
- Every hour a new JW collection will be started. The collection will run for 65 minutes (to have a 5 minute overlap for starting the next one and the previous one ending). This program will loop forever, keeping 5 job watcher collections, until the program is canceled.
- Submit the job to batch and let it run; end the batch job when you want to end your data collections. The last job watcher collection will remain active until it completes.

## Sample CL Program – Provided AS IS

```

PGM
      DCL          VAR(&DECNUMBER) TYPE(*DEC) LEN(15 5)
      DCL          VAR(&NUMBER) TYPE(*UINT) LEN(2)
      DCL          VAR(&COLNAME) TYPE(*CHAR) STG(*AUTO) LEN(10) +
                  VALUE(JWCOL)

      ADDLIBLE JOBWATCH
      MONMSG CPF9999

      BEGIN:

      RTVDTAARA   DTAARA (JOBWATCH/JOBWATCH *ALL) +
                  RTNVAR (&DECNUMBER)

      CHGVAR      VAR(&NUMBER) VALUE (&DECNUMBER)

      IF          COND (&DECNUMBER *EQ 1) THEN (CHGVAR +
                  VAR (&DECNUMBER) VALUE (5))
      ELSE       CMD (CHGVAR VAR (&DECNUMBER) VALUE (&DECNUMBER +
                  - 1))

      CHGDTAARA   DTAARA (JOBWATCH/JOBWATCH *ALL) +
                  VALUE (&DECNUMBER)

      CHGVAR      VAR(%SST (&COLNAME 6 2)) VALUE (&NUMBER)

      STRJW       DFN (Q5SECSQL) COL (&COLNAME) LIB (JOBWATCH) +
                  RPLDTA (*YES) ENDCOL ((*NBRSEC 3900))

      DLYJOB 3600

      GOTO BEGIN
      ENDPGM

```

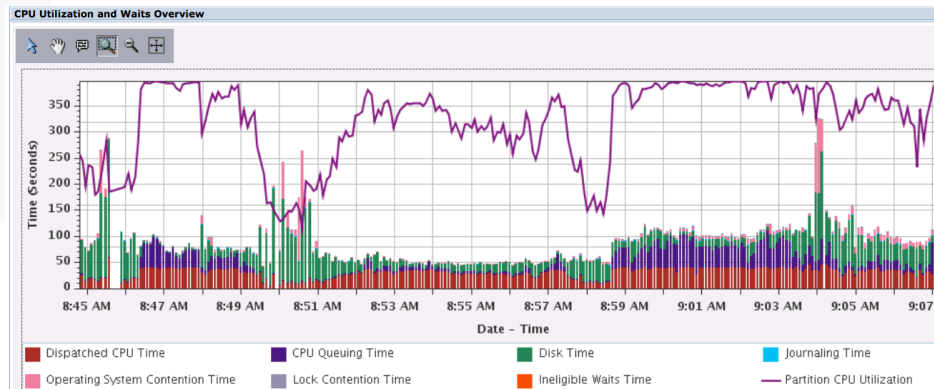
## Job Watcher Authority Requirements

- **Commands:**
  - You must have service (**\*SERVICE**) special authority
    - Change User Profile to add \*SERVICE authority to create Job Watcher Definitions or to Start Job Watcher
  - **-OR-** Be authorized to the Job Watcher function of the operating system
    - Change Function Usage (CHGFCNUSG) command, with a function ID of QIBM\_SERVICE\_JOB\_WATCHER can be used to change the list of users that are allowed to use this command.
    - CHGFCNUSG FCNID(QIBM\_SERVICE\_JOB\_WATCHER) USER(<usrprofile>) USAGE(\*ALLOWED)
- **Definitions:**
  - Additional authority is needed to see the definitions for each as they are shipped with public authority \*EXCLUDE. To see the definitions shipped in Job Watcher, users will need authority to the QAPYJWDFN file in QUSRSYS
- [www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20%20Technology%20Updates/page/Authority](http://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20%20Technology%20Updates/page/Authority)
- [https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20%20Technology%20Updates/page/Authorityadd\\_more\\_guidance\\_on\\_creating\\_the\\_JW\\_definition](https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20%20Technology%20Updates/page/Authorityadd_more_guidance_on_creating_the_JW_definition)

## Job Watcher – PDI interface

Job Watcher

- CPU Utilization and Waits Overview
- CPU Utilization by Thread or Task
- Resource Utilization Overview
- ▣ Job Statistics Overviews
- ▣ Waits
- ▣ CPU
- ▣ Physical Disk I/O
- ▣ Synchronous Disk I/O
- ▣ Storage Allocation
- ▣ Page Faults
- ▣ Logical Database I/O
- ▣ 5250 Display Transactions
- ▣ Job Watcher Database Files



# Job Watcher – Interval Details

### Thread or Task Details

Job information: QZDASOINIT/QUSER/128962 - 0000000000000005

Current user profile: LISAW

Object waited on: INVENTORY INVENTORY

Wait duration: 581 milliseconds

Current or last wait: DB record lock: update

Holding job or task: QZDASOINIT/QUSER/128890

SQL client job: None detected this interval

Priority: 20

Pool: 2

Type description: PHYSICAL FILE MBR - DATA PART

Segment type description: DB PHYSICAL FILE MEMBER RECORDS

Wait object library: None detected this interval

Interval timestamp: Jan 3, 2014 2:36:28 PM

Interval (1 to 684): < | 174 | >

### Call Stack

Call Level	Program	Module	Procedure
1			quitde_block_tra
2			longWaitReceive__9QuCounterFR12RmprReceiverP
3			DBLockConflict__15RmsIDBHashClassFR11RmsIPIm
4			rmsIDBHLock__FR11RmsIPImPLad
5			getLockWithWait__18DbpmUpdateResourcecd
6			getLock__18DbpmUpdateResourcecd
7			getRowLock__18DbpmUpdateResourceFCUIRC9Dbp
8			execute__18DbpmUpdateLockNodeFR13DbpmQuer
9			vPositionNextAndExecute__18DbpmUpdateLockNod
10			positionNextEntryAndFetchOutline__17DbpmReadO

### SQL Statement

Include Host Variables

```
SELECT QUANTITY FROM WAREHSE42.INVENTORY WHERE ID="DATA FORMAT ERRORITY FROM WAREHSE42.INVENTORY WHERE ID=? FOR UPDATE
```

# Performance Explorer

## Performance Explorer

- Performance Explorer helps identify the causes of performance problems that cannot be resolved using one of the other performance data collectors
  - Collects more detailed information about a specific application, program, or resource
- Performance Explorer is typically used for two main reasons:
  - Detailed performance trace data is needed to identify the performance problem
  - ★ – Analyzing the performance of applications
- Performance Data Investigator supports profile collections only
- iDoctor is required for advanced PEX Analysis

## Performance Explorer

- Performance Explorer is the most sophisticated IBM i performance tool
  - Can collect the details of every I/O operation, every task switch
  - Hundreds of events collected
  - Thus, most complex to use
  - More overhead
- Typically, problem has been scoped by other tools first
- Generally used by IBM performance analysis experts
  - Except.....

## Performance Explorer – “TPROF” usage

- “Trace-profile” is a fairly easy, and fairly “light-weight” PEX collection that can be useful to application developers, especially when trying to diagnose high CPU issues
  - Provides CPU usage at a program/module/procedure level
  - Make sure you have latest PEX PTFs applied  
<http://www-01.ibm.com/support/docviewwss?uid=nas8N1012020>
  - Can be run over subset of jobs

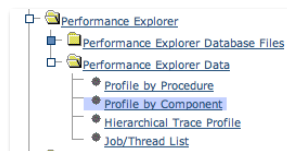
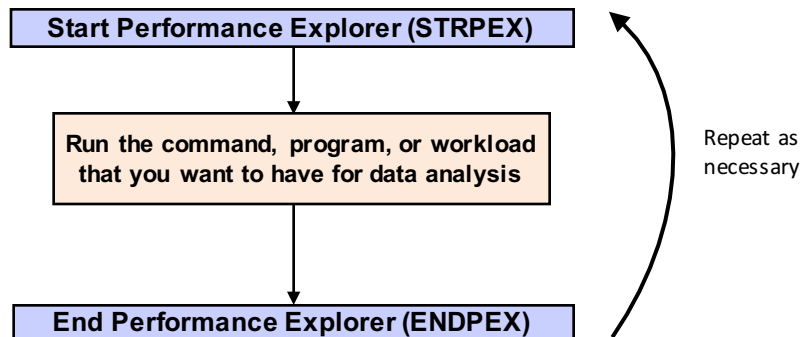
- Steps:

1. Add a PEX definition:

```
ADDPEXDFN DFN(TPROF)
TYPE(*TRACE)
JOB((*ALL *ALL))
TASK(*ALL)
MAXSTG(4000000)
INTERVAL(1)
TRCTYPE(*SLTEVT)
SLTEVT(*YES)
MCHINST(*NONE)
BASEVT((*PMCO *NONE *FORMAT2))
```

## Performance Explorer – “TPROF” usage

### 2. Collect data





# Performance Explorer TPROF reports – PDI

**Profile by Procedure**

Program Name	Module Name	Procedure Name	Component	Hit Count
CFTSMPI		#cftsmpl	SLIC Common Functions	332(48.61%)
STRHU		do_copyMemoryLarge	SLIC String Functions	94(13.76%)
DBRSQMN		#dbrsqmn	SLIC Database	85(12.45%)
CUSTOMER CUSTOMER		#DBXFMP2	MI Other	45(6.59%)
READER	READER	READER	MI Other	27(3.95%)
DBPM2010		sExecute_42VariableLen	SLIC Database	6(0.88%)
SHMUTLH		trimRangeForRead_145	SLIC Storage Management	6(0.88%)
HvString		HvString	SLIC Hypervisor	4(0.59%)
SMMSUBH		findStealablePage_205m	SLIC Storage Management	4(0.59%)
QDBGTEM	QDBGTEM	QDBGTEM	XPF Database Other	4(0.59%)

**Profile by Component**

Select	Total	Component	Procedure Name	Hit Count
<input type="checkbox"/>	683	Total		683(100%)
<input type="checkbox"/>	335	SLIC Common Functions		(48.05%)
<input type="checkbox"/>	118	SLIC Database		(17.28%)
<input type="checkbox"/>	85	#dbrsqmn	#dbrsqmn	(12.45%)
<input type="checkbox"/>	6	sExecute_42VariableLen	sExecute_42VariableLen	(0.88%)
<input type="checkbox"/>	4	sExecute_14HashOperati	sExecute_14HashOperati	(0.44%)
<input type="checkbox"/>	4	sExecute_17PackedDivide	sExecute_17PackedDivide	(0.29%)
<input type="checkbox"/>	2	sSld_19VariableLengthPr	sSld_19VariableLengthPr	(0.29%)
<input type="checkbox"/>	2	vPositionNextAndExecute	vPositionNextAndExecute	(0.29%)

**Job/Thread List**

Select	Name	Task/Thread Name	Cumulative CPU Time in Seconds	Active Time in Seconds	Hit Count
<input type="checkbox"/>	Total		143.50993	221051.77	509401459333(100%)
<input type="checkbox"/>	QDFTJOB		68.429	123.51712	443134332927(86.99%)
<input type="checkbox"/>	WEBADM		68.429	123.51712	
<input type="checkbox"/>	QDFTJOB/WEBADM/012		68.429	123.51712	
<input type="checkbox"/>	0000000000000004	MITHREAD	68.429	123.51712	
<input type="checkbox"/>	QZDASOINIT		8.869434	1354.2184	
<input type="checkbox"/>	QPADEV0005		0.462739	123.545746	
<input type="checkbox"/>	ADMIN1		0.10104	5105.88	
<input type="checkbox"/>	ADMIN2		0.101269	9519.301	
<input type="checkbox"/>	QLWISVR		0.101269	9519.301	

**Hierarchical Trace Profile**

Select	Name	Hit Count
<input type="checkbox"/>	Total	683(100%)
<input type="checkbox"/>	LIC	606(88.73%)
<input type="checkbox"/>	DB Code Burst	45(6.59%)
<input type="checkbox"/>	CUSTOMER CUSTOMER	45(6.59%)
<input type="checkbox"/>	#DBXFMP2	45(6.59%)
<input type="checkbox"/>	Amount of elements that did not match the filter: 1	0(0%)
<input type="checkbox"/>	MI ILE	27(3.95%)
<input type="checkbox"/>	MI OPM	4(0.59%)
<input type="checkbox"/>	Unknown	1(0.15%)

# Examples



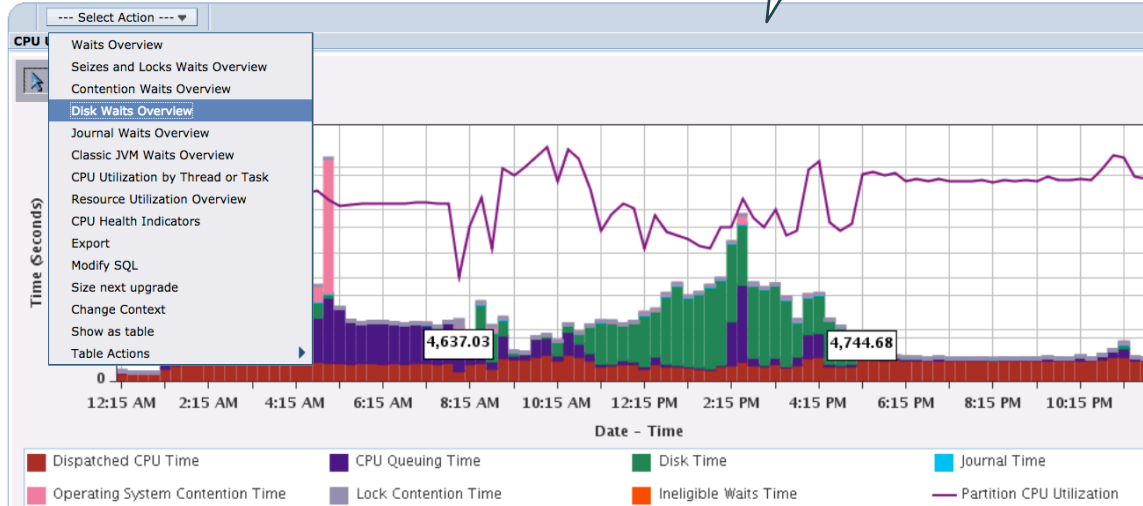
# What is causing disk wait time? Reviewing wait times with Collection Services



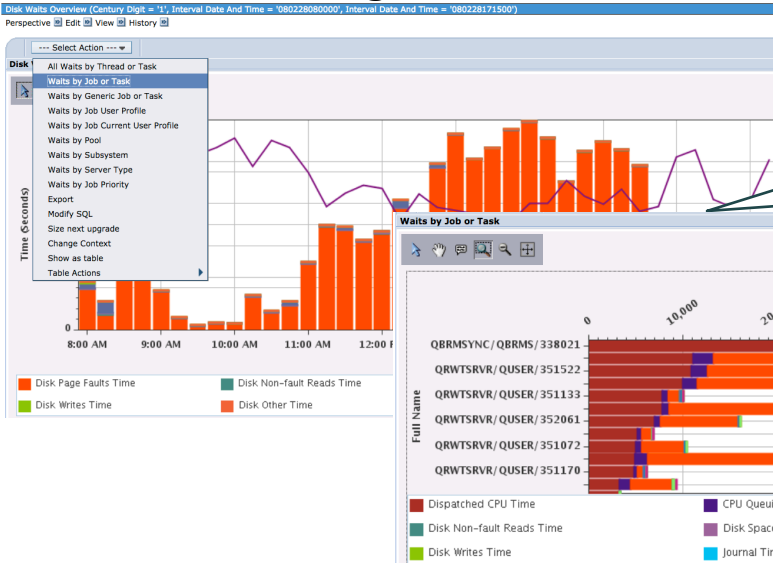
## Let's Look at the Disk Waits

That mountain of green is Disk Time.  
What caused it?

### CPU Utilization and Waits Overview



# We see it's faulting.... let's find out who did it



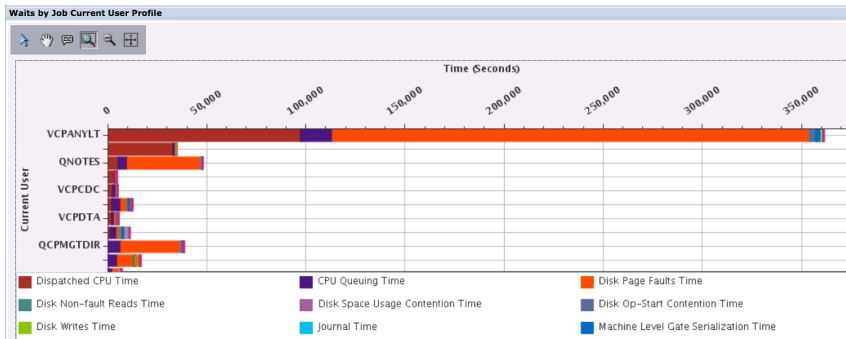
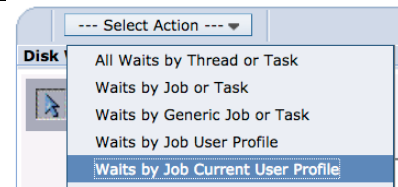
Drill down into "Waits by Job or Task".



We can see it is the QRWTSRVR DDM/DRDA server jobs.

# Let's find out who the user is

- We now have several clues:
  - We know the jobs - QRWTSRVR - DDM/DRDA server jobs
  - We know the time - early afternoon
  - We know the user profile - QUSER
    - But QUSER isn't helpful. We need the job's current user profile
  - Waits by Job Current User Profile shows us VCPANYLT is the guilty party

Disk Waits Overview (Century Digit = '1', Interval D... Perspective Edit View History




# Viewing Waits with Job Watcher

## Example of Machine Gate Serialization

© 2017 IBM Corporation 87



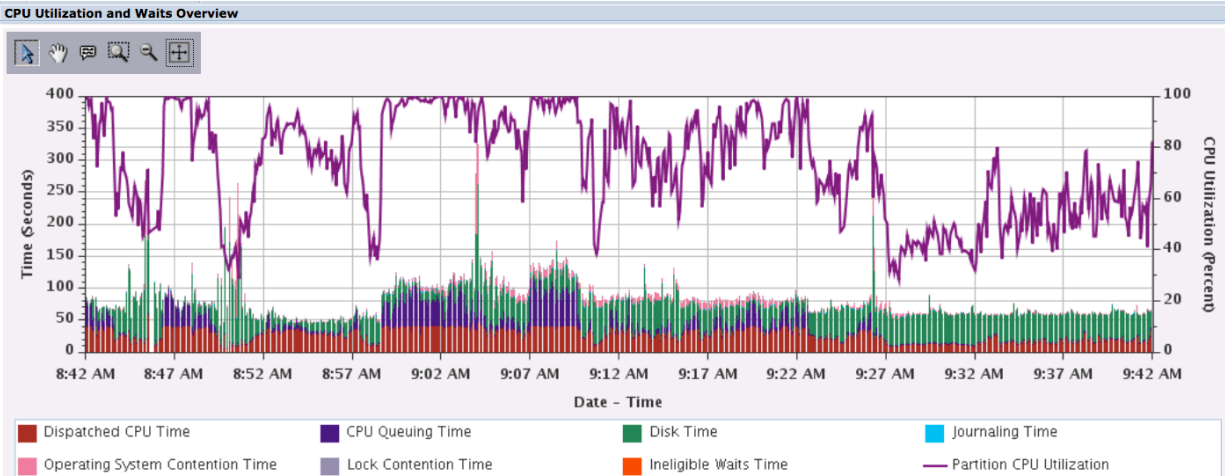
Job Watcher

- CPU Utilization and Waits Overview
- CPU Utilization by Thread or Task
- Resource Utilization Overview

## Let's look at the entire collection

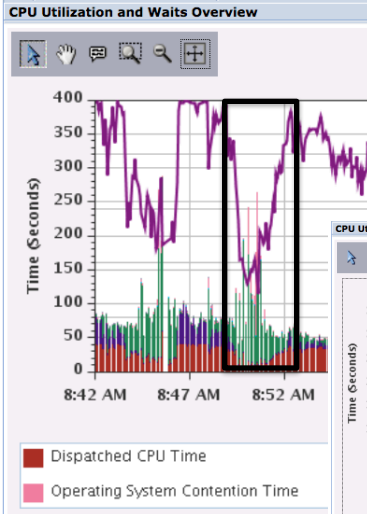
- CPU Utilization and waits gives an excellent overview of the entire partition
- There are a lot of interesting things to investigate.....

**CPU Utilization and Waits Overview**

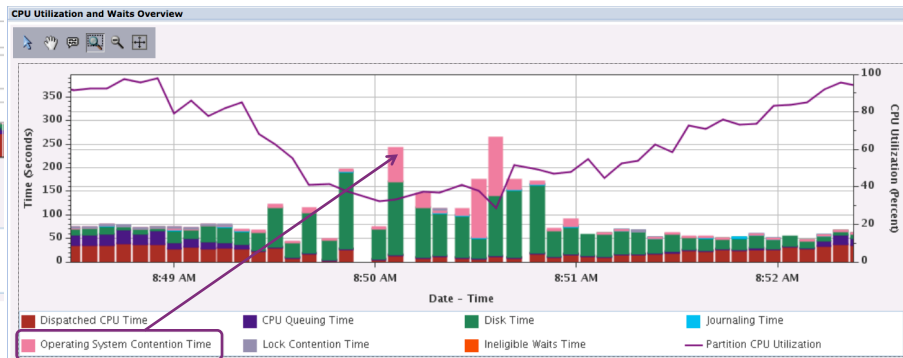


■ Dispatched CPU Time	■ CPU Queuing Time	■ Disk Time	■ Journaling Time
■ Operating System Contention Time	■ Lock Contention Time	■ Ineligible Waits Time	■ Partition CPU Utilization

## Zoom into the time where we see a large drop in CPU Utilization



We can see operating system contention occurred during the time when the CPU Utilization dropped.

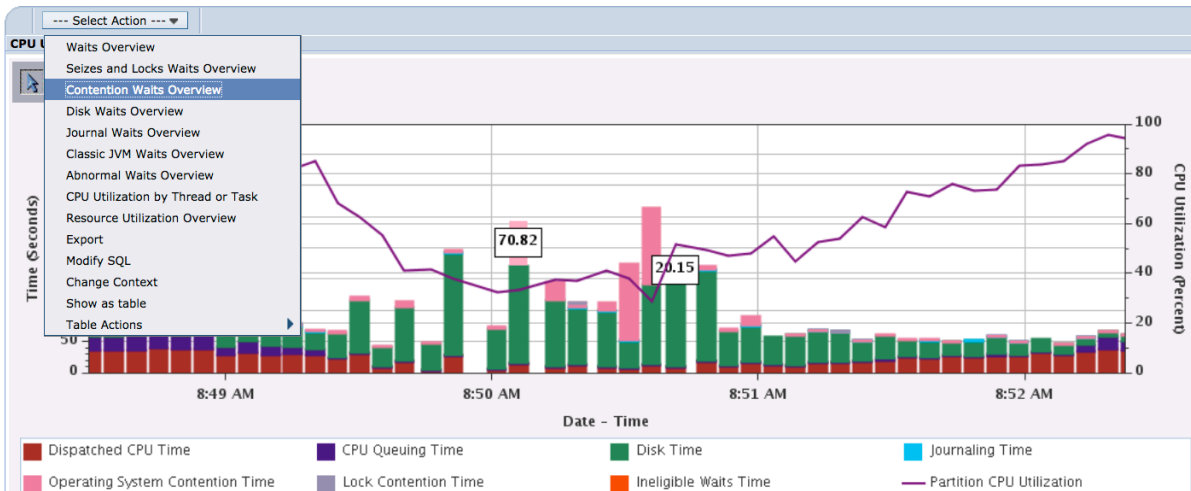


© 2017 IBM Corporation

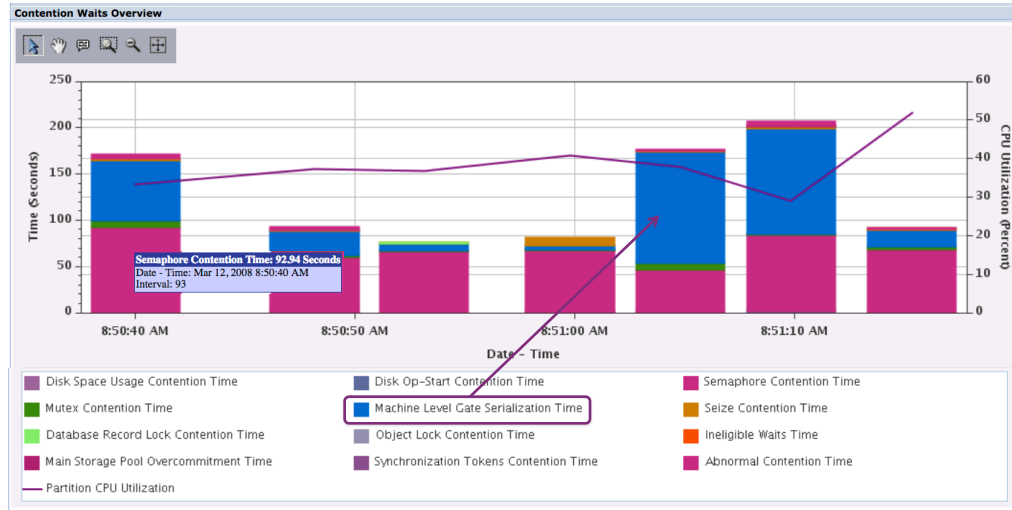
## Drill-down based upon what you see

Select the beginning and ending intervals to investigate and then drill into Contention Waits Overview  
Be sure to select the metric you are investigating.

CPU Utilization and Waits Overview

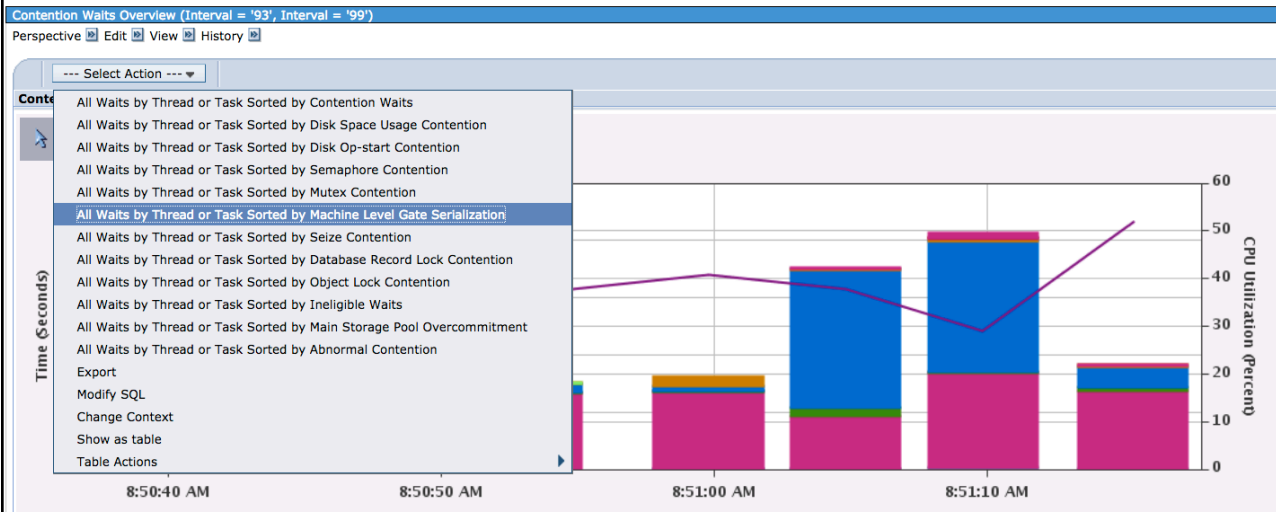


**Machine level gate serialization is a major reason for the contention waits**  
*The pink is semaphore contention (see the flyover). This is usually an uninteresting wait.*



We want to see if we can figure out who might be causing the contention.  
 Drill into **All Waits by Thread or Task Sorted by Machine Level Gate Serialization** so we can see the jobs/threads/tasks that are all waiting.  
 We selected **Machine Level Gate Serialization** because that's the predominate "bad" wait our graph shows us.

Note: Drilling into **waits by thread or task** can take some time.... be patient.

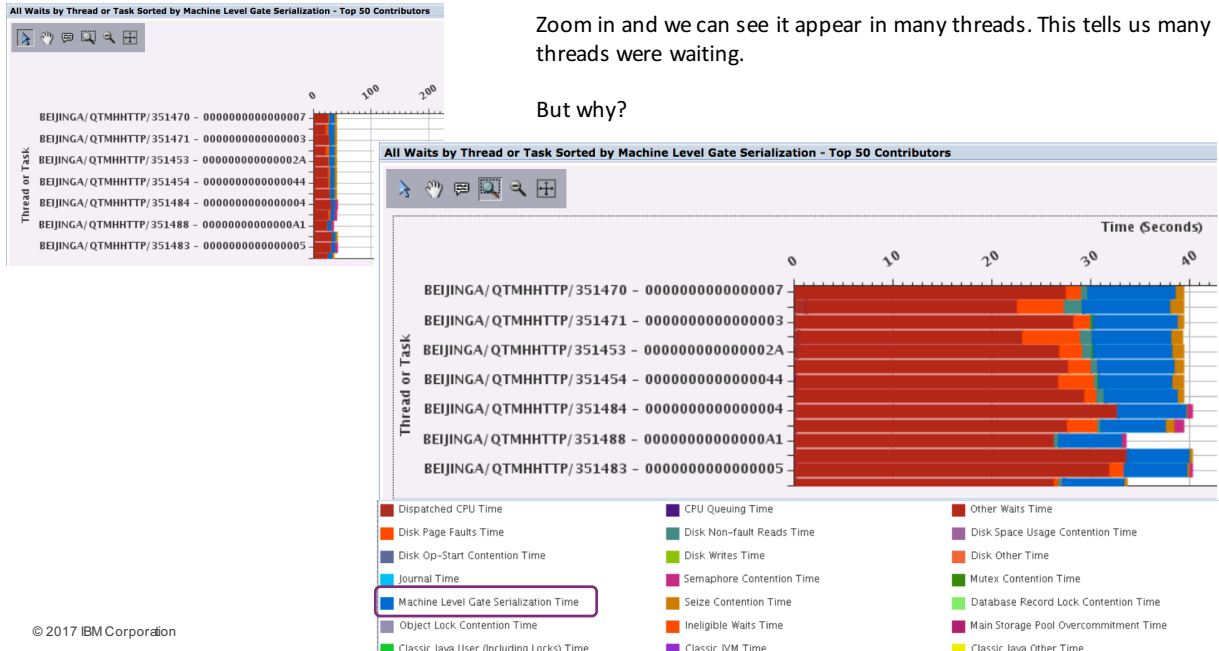


## Zoom into see more detail

We can't see the machine level gate serialization details at first;

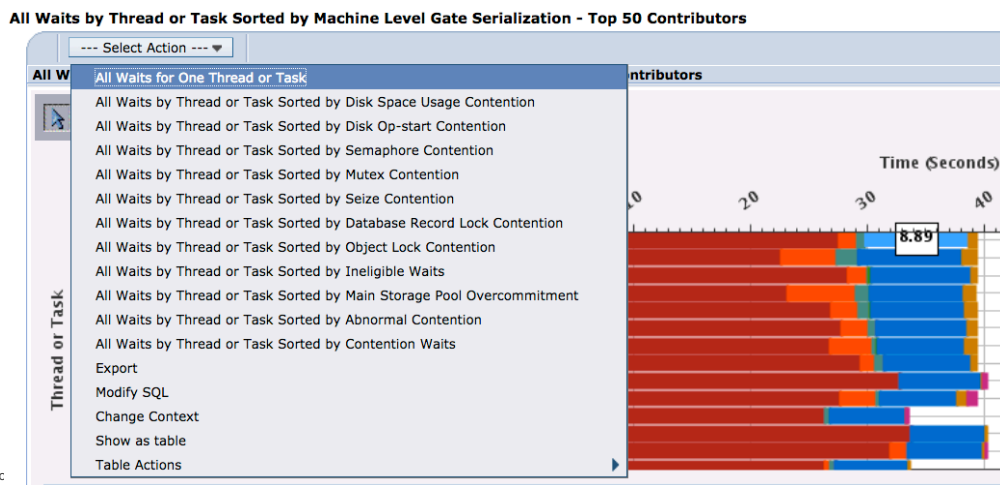
Zoom in and we can see it appear in many threads. This tells us many threads were waiting.



But why?



## Select a thread and look at the waits for that one thread

It may be necessary to drill down into interval details for several threads to find the one with the information we need...



Cognitive Systems  

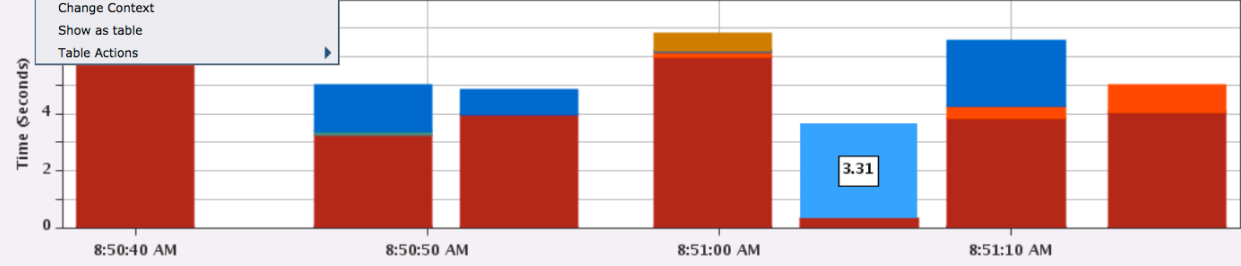
## Select an interval View Interval details for one thread or task

**All Waits for One Thread or Task**

--- Select Action ---

All W **Interval Details for One Thread or Task**



- Waits Overview
- Export
- Modify SQL
- Change Context
- Show as table
- Table Actions



Time (seconds)




8:50:40 AM 8:50:50 AM 8:51:00 AM 8:51:10 AM

© 2017 IBM Corporation 95



Cognitive Systems  

## And here is where we discover Job Watcher's power....

**Interval Details for One Thread or Task (Interval = '97', Initial Thread Task Count = '319696')**

Perspective  Edit  View 

**Thread or Task Details**

Job information:	BEIJINGA/QTMHHTTP/351470 - 0000000000000007	Priority:	25
Current user profile:	QTMHHTTP1	Pool:	2
Object waited on:	QAUDJRN	Type description:	JOURNAL
Wait duration:	3,307 milliseconds	Segment type description:	JOURNAL RESERVED
Current or last wait:	Qu gate - high performance	SQL client job:	None detected this interval
Holding job or task:	QDBSRV02/QSYS/345313	Interval timestamp:	Mar 12, 2008 8:51:05 AM
<a href="#">Show Holder</a>		Interval (1 to 710):	 97 

© 2017 IBM Corporation 96



We can review the call stack to see how we got to this wait point.

This thread is waiting for the QAUDJRN journal at 8:51:05.

In the call stack you will see an entry that shows the job is creating an audit journal entry.

Note that access to the audit journal is serialized by a “gate”. So why is this job blocked and waiting to create the audit record?

**Call Stack**




Call Level	Program	Module	Procedure	Offset
1			qutde_block_trace	00000E4
2			slowLock__10QuGateCodeFQ2_2Qu8Lock	000003F0
3			#journal	0000066C
4			auditIt__19CfCreateAuditRecordFv	00000444
5			validateDescQueue__11LoDescEntryFRFP	00000390
6			rcvDescriptors__12LoSocketUnixFR15Lo	000002B8
7			rcv__19LoReceiveStreamUnixFR15LoSo	000006A0
8			rcvmsg__8LoSocketFR15LoSocketManag	0000018C
9			rcvmsg__FtP6msghdrT1PtP7timeval15L	00000680
10			rcvMsgHandler__FP19LoSocketRecvMsg	00000448
11			LoSocketOp__FUTP13LoSocketOpHdr	00000254
12			socketop	000001DC
Total: 28				

## We can easily go look at the thread that is holding the resource

The call stack shows a “Change Journal” program (QJOCHJN)

Show Holder

Interval Details for One Thread or Task (Interval = '97', Initial Thread Task Count = '1476')

Perspective  Edit  View 

**Thread or Task Details**

Job information: QDBSRV02/QSYS/345313 - 0000000000000001	Priority: 16
Current user profile: QSYS	Pool: 2
Object waited on: QAUDJRN	Type description: JOURNAL
Wait duration: 104 milliseconds	Segment type description: JOURNAL SPACE IOCB
Current or last wait: MAINSTORE/LOGICAL-DASD-IO: PAGE FAULT	SQL client job: None detected this interval
Holding job or task: None detected this interval	Interval timestamp: Mar 12, 2008 8:51:06 AM
Show Holder	Interval (1 to 710): < 97 >

**Call Stack**

Call Level	Program	Module	Procedure
1			qutde_stackless_block
2			qu_dasd_fault_on_res_stack
3			#jomodjp
4			#cfmir
5			syscall_A_portal
6		QJOCHJN	
7			cblabbranch
8			aiuser_program_call_portal
9		QJORETRY	
10			cblabbranch
11			aiuser_program_call_portal
12		QDBSERVE	
Total: 15			

Since we know the object that was causing the contention was QAUDJRN, let's have a look....

Note the audit journal entries from the matching time period. Observe the job name...

NR is Next Receiver  
PR is Previous Receiver

```

Display Journal Entries

Journal . . . . . : QAUDJRN      Library . . . . . : QSYS
Largest sequence number on this screen . . . . . : 000000000008885894
Type options, press Enter.
  5=Display entire entry

Opt   Sequence Code  Type  Object      Library      Job           Time
-----
88885883 T    GS      BEIJINGA    8:51:02
88885884 T    SK      QSYSARB     8:51:02
88885885 J    NR      QDBSRV02   8:51:02
88885886 J    PR      QDBSRV02   8:51:06
88885887 T    GS      BEIJINGA    8:51:07
88885888 T    GS      BEIJINGA    8:51:07
88885889 T    GS      BEIJINGA    8:51:07
88885890 T    SK      QSYSARB     8:51:07
88885891 T    GS      BEIJINGA    8:51:07
88885892 T    GS      BEIJINGA    8:51:07
88885893 T    GS      BEIJINGA    8:51:07
88885894 T    GS      BEIJINGA    8:51:07

F3=Exit  F12=Cancel

More...

```

## What next?

- We observed significant system-wide machine gate serialization contention
- With Job Watcher, we determined that at that point in time, the QAUDJRN was changing journal receivers
- Perhaps we should review our security audit settings
- In this example, it was a stress-test System Test partition with a heavy workload and extreme settings



# Viewing Waits with Job Watcher

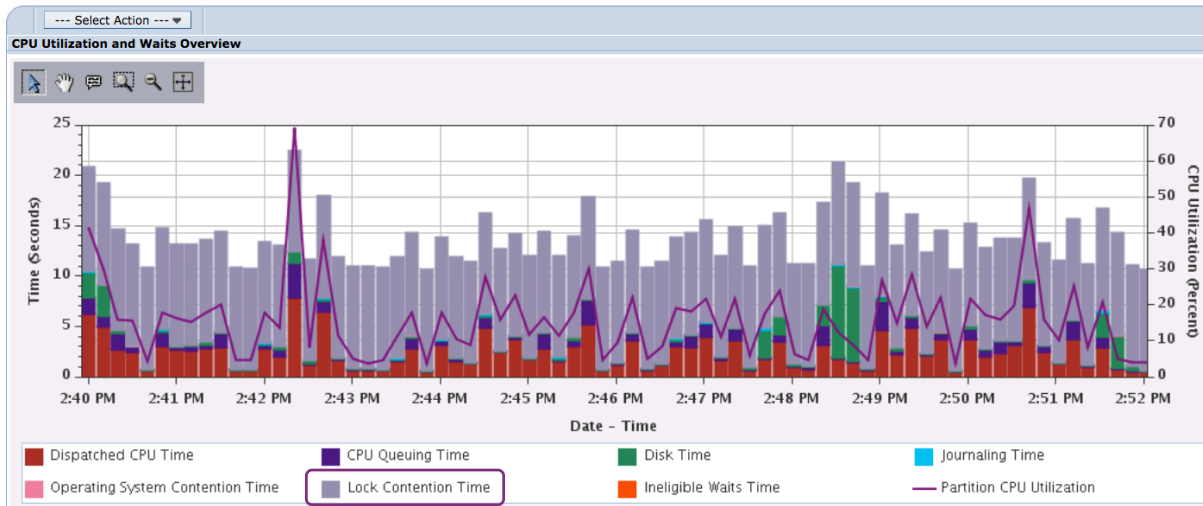
## Example of Object Lock Contention



### Once again, we start with CPU Utilization and Waits Overview

Observe the gray ... "Lock Contention Time"

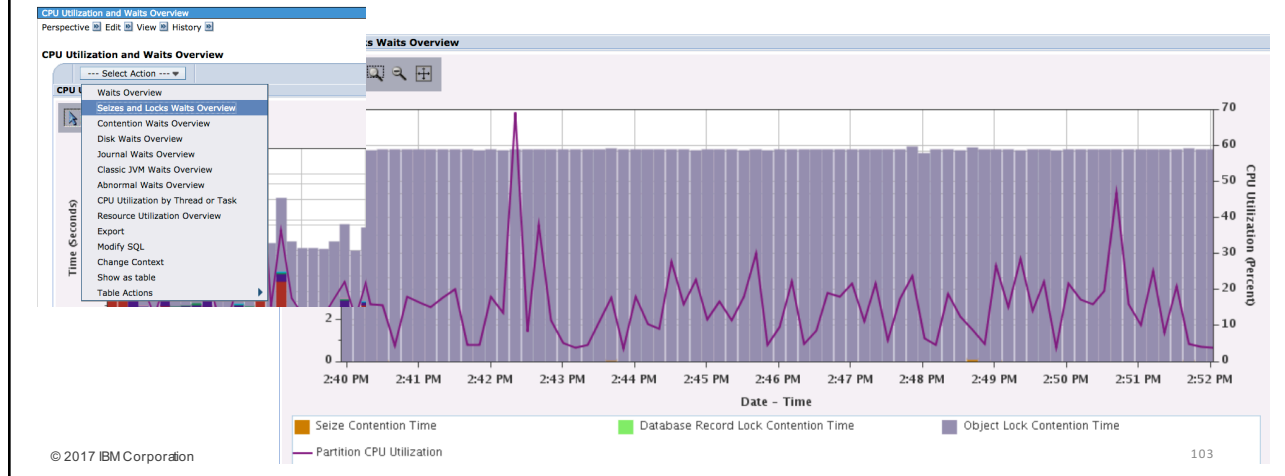
CPU Utilization and Waits Overview



## Drill into Seizes and Locks Waits Overview

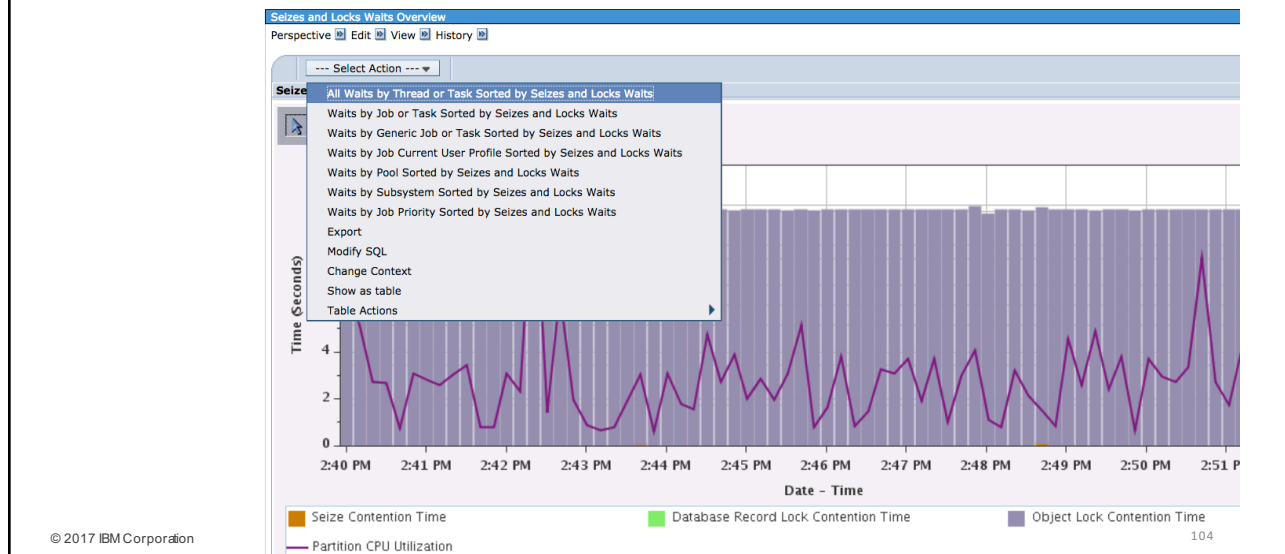
Tip: Drill-downs are named to drill into what you see

- This confirms we have some job holding a lock that another job is waiting on

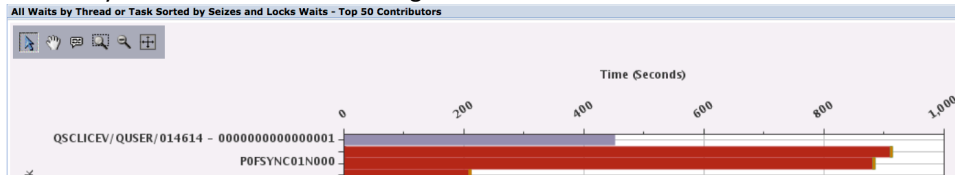


## Seizes and Lock Waits Overview → All Waits by Thread or Task...

Drill down into All Waits by Thread or Task Sorted by Seizes and Lock Waits

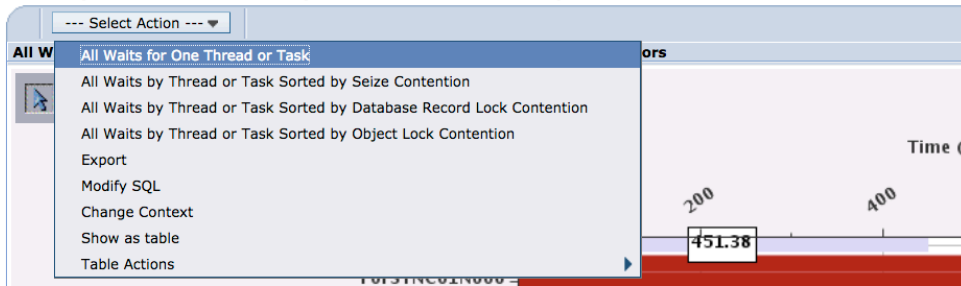


We can clearly see the thread that is waiting for the lock.



Select that thread and drill down to **All Waits for One Thread or Task**

**All Waits by Thread or Task Sorted by Seizes and Locks Waits - Top 50 Contributors**

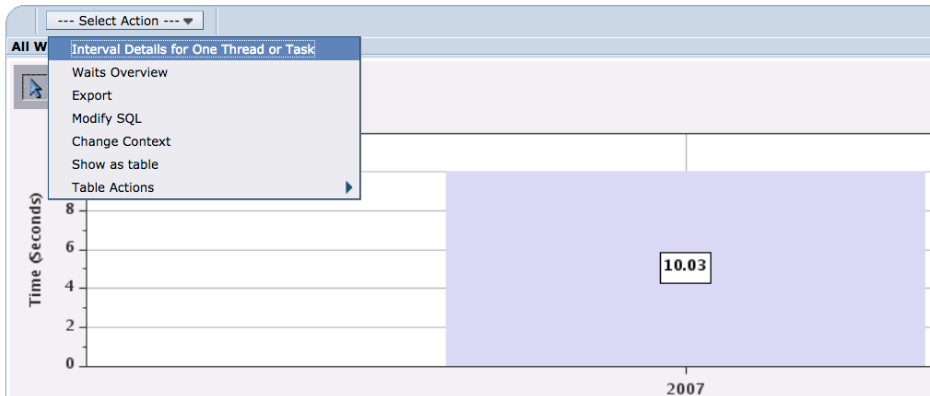


## We are now to the magic drill-down...

Select the interval for that job and drill down to

**Interval Details for One Thread or Task**

**All Waits for One Thread or Task**





# Interval Details

Interval Details for One Thread or Task (Interval = '47', Initial Thread Task Count = '2545')

Perspective  Edit  View 

### Thread or Task Details

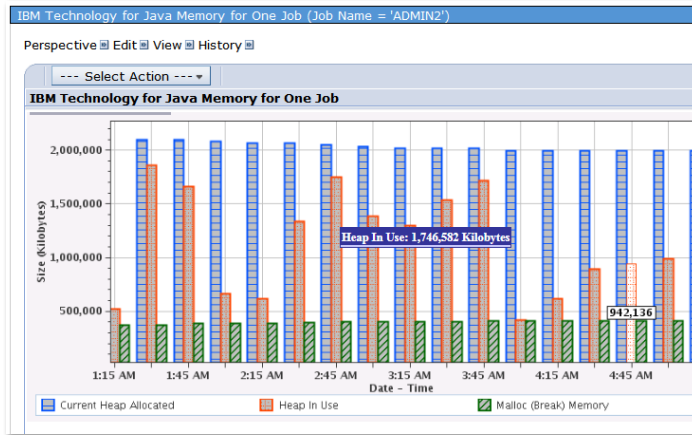
Job information:	QSCLICEV/QUSER/014614 - 0000000000000001	Priority	25
Current user profile:	QSYS	Pool:	2
Object waited on:	WATCHEVENTSPACE	Type description:	TEMPORARY - SPACE
Wait duration:	4,482 milliseconds	Segment type description:	BASE MI SYSTEM OBJECT
Current or last wait:		SQL client job:	None detected this interval
Holding job or task:	QZRCSRVS/QUSER/014097	Interval timestamp:	Dec 13, 2007 2:48:00 PM
<a href="#">Show Holder</a>		Interval (1 to 91):	<input type="text" value="47"/>  

### Call Stack

# More PDI Examples

## Java Perspectives

Drilldown for one job -  
Look at the heap and memory usage over time for one selected job.

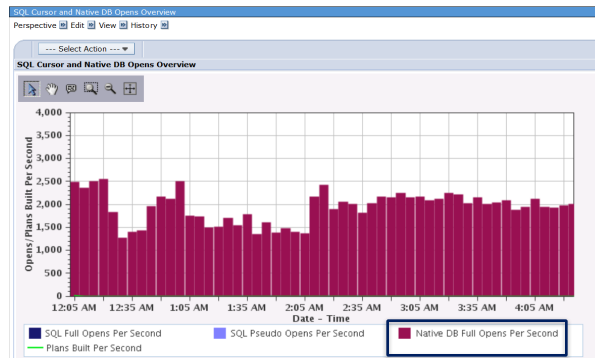


IBM Technology for Java Memory for One Job

## Database Full Opens

Full Opens are expensive resource-wise

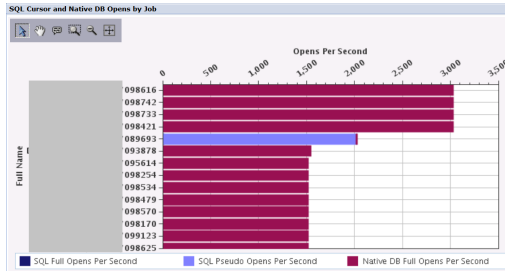
- Collection Services Database Files
  - Database
    - I/O Reads and Writes
    - SQL CPU Utilization by Job or Task
    - Database Locks Overview
  - Database I/O
    - SQL Cursor and Native DB Opens
      - SQL Cursor and Native DB Opens Overview
      - SQL Cursor and Native DB Opens by Job
      - SQL Cursor and Native DB Opens by Generic Job
      - SQL Cursor and Native DB Opens by Job Current User Profile
  - SQL Performance Data



General recommendation is to keep Native Full Opens per second < 1000

Next, find jobs doing full opens...

## Database Full Opens

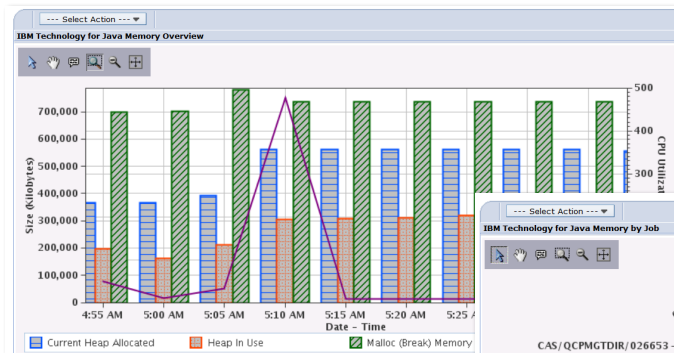


In an RPG program, full opens are caused by the use of SETON \*LR instead of RETRNR. Avoid LR if possible. Not setting on LR (in ORM programs) will keep the program in memory, keeps file open and pointers set, retains variable values, etc. Also, avoid \*NEW for the ACTGRP

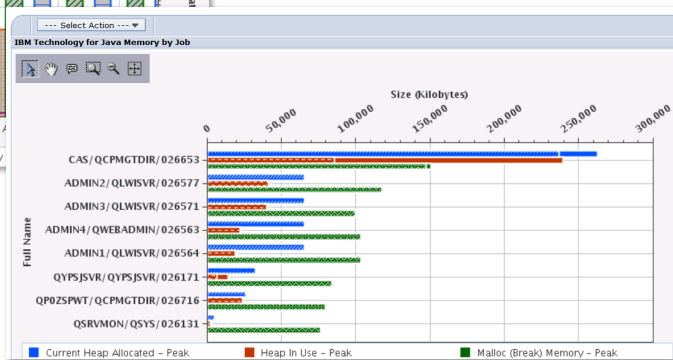
Shared file opens are far less expensive than full file opens. They consume less CPU, less storage and are faster than full opens. There are some implications of changing full file opens to shared file opens, but implementation of shared opens is typically easier to manage and implement than changing SETON LR to RETRNR in RPG programs. The following links provide explanation, usage and considerations of using shared ODBs:

- Sharing database files in the same job or activation group  
<http://pic.dhe.ibm.com/infocenter/series/v7r1m0/topic/dbp/rbaofosfile.htm?resultof=%22%44%50%22%20%22%6f%64%70%22%20>
- Open considerations for files shared in a job or an activation group  
<http://pic.dhe.ibm.com/infocenter/series/v7r1m0/topic/dbp/rbaofopenchtm>
- Input/output considerations for files shared in a job or an activation group  
<http://pic.dhe.ibm.com/infocenter/series/v7r1m0/topic/dbp/rbaofioc.htm>
- Close considerations for files shared in a job or an activation group  
<http://pic.dhe.ibm.com/infocenter/series/v7r1m0/topic/dbp/rbaofclose.htm>

## Java Perspectives in Collection Services



Find that job using a lot of heap...





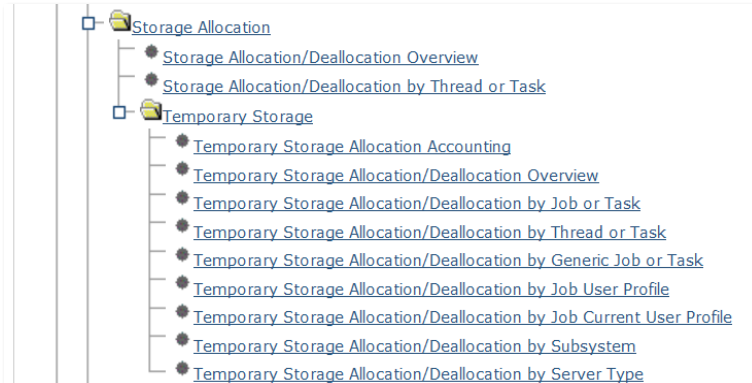
# Temporary Storage Allocation/Deallocation perspectives

7.2

## Storage Allocation Perspectives

Where is my temporary storage going?

Expand Collection Services



**Selection**

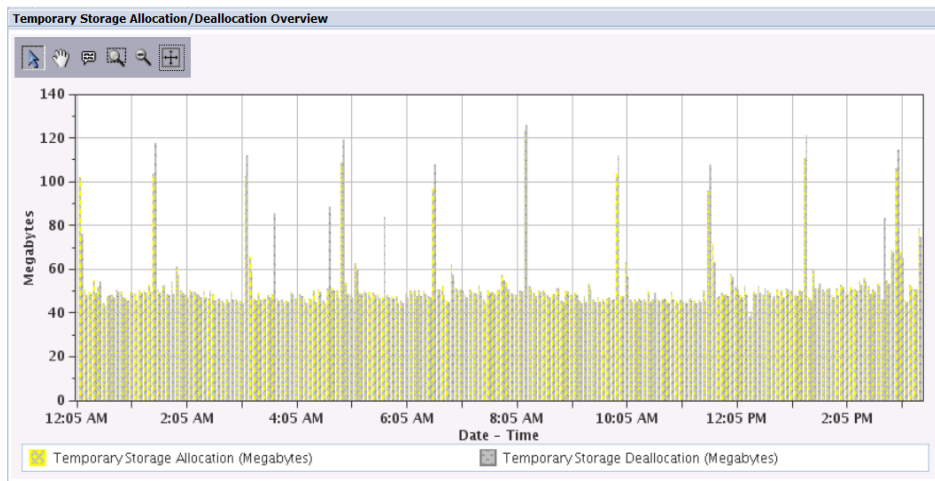
**Name**  
Temporary Storage Allocation Accounting

**Description**  
This chart shows the amount of temporary storage charged to active and ended jobs, the amount of user temporary storage, and the amount of temporary storage used for database and non database operations by the IBM i operating system across the system over time for the selected collections. Use this chart to select a time frame for further detailed investigation.

**View List**  
Temporary Storage Allocation Accounting and SQL Statements  
Temporary Storage Allocation Accounting and Disk Average Response

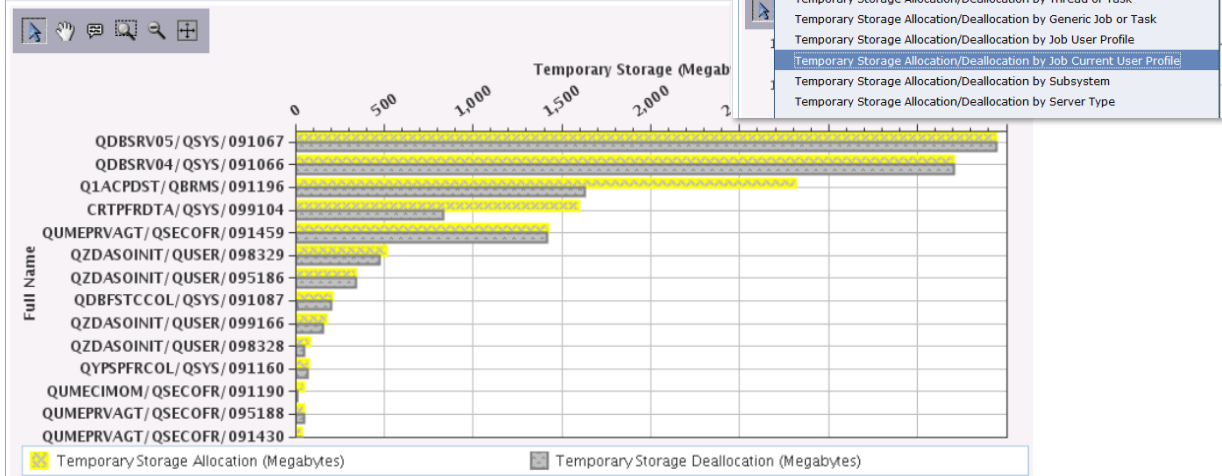
# Temporary Storage Allocation / Deallocation Overview

Generally, allocations and deallocations following a similar pattern



## From an overview perspective, drill down to more detail

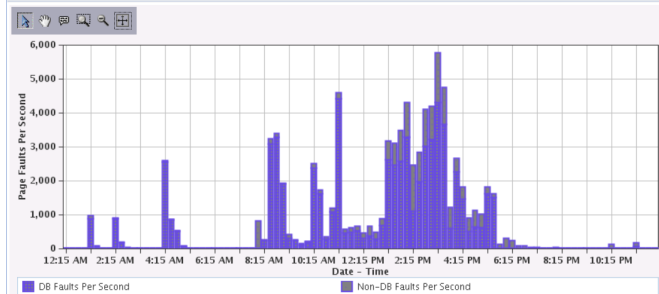
Temporary Storage Allocation by Job or Task



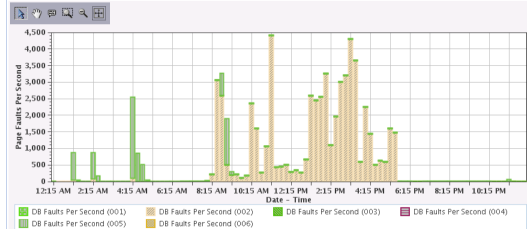
## What does the faulting look like when I was testing?

- Memory
- Memory Pool Sizes and Fault Rates
- Memory Pool Activity Levels
- DB and Non-DB Page Faults

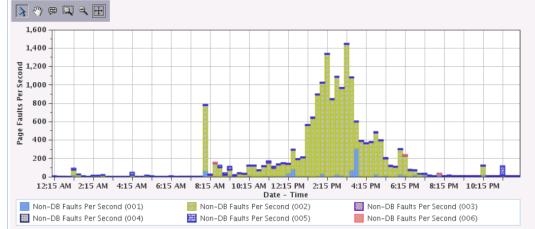
DB and Non-DB Page Faults Overview (All Pools)

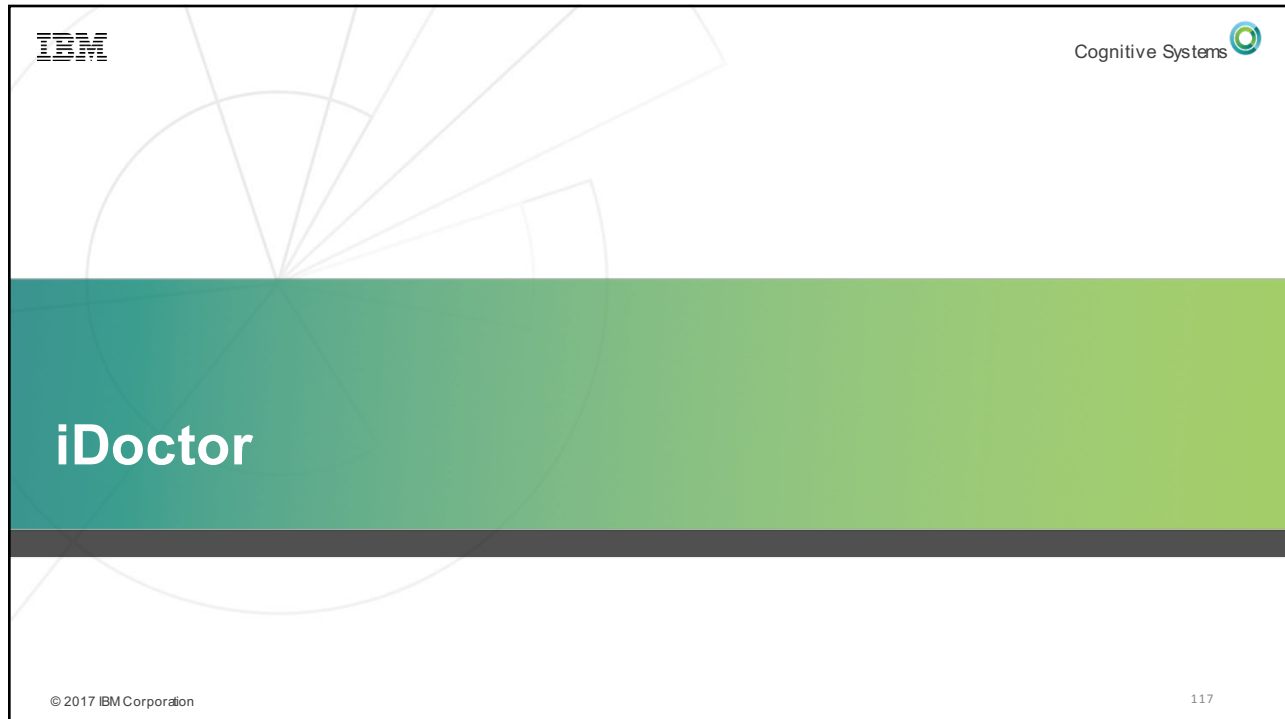


DB Page Faults (All Pools)



Non-DB Page Faults (All Pools)





The slide features the IBM logo in the top left and the Cognitive Systems logo in the top right. A large green horizontal bar spans the middle of the slide, with the word "iDoctor" written in white text on the left side. The background has a faint, light-colored circular graphic.

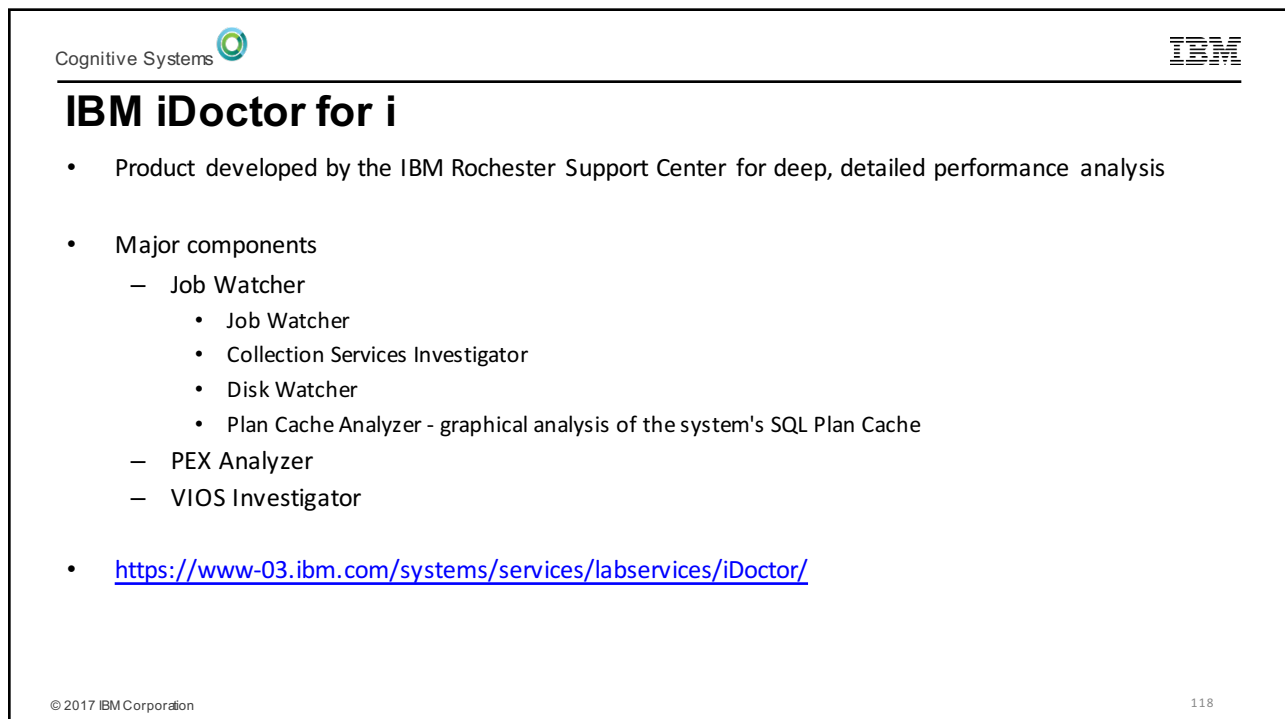
IBM

Cognitive Systems

# iDoctor

© 2017 IBM Corporation

117



The slide features the Cognitive Systems logo in the top left and the IBM logo in the top right. The title "IBM iDoctor for i" is prominently displayed. Below the title is a bulleted list of product details and components. A URL is provided at the bottom of the list.

Cognitive Systems

IBM

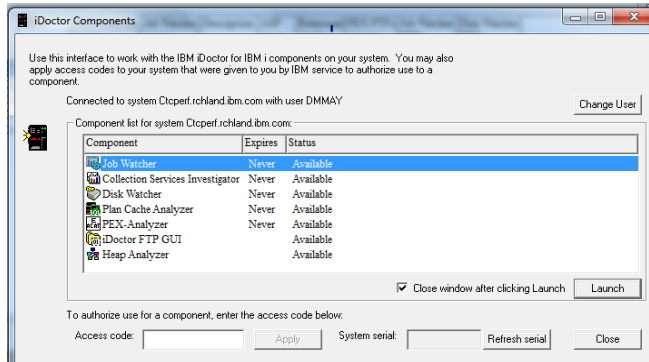
## IBM iDoctor for i

- Product developed by the IBM Rochester Support Center for deep, detailed performance analysis
- Major components
  - Job Watcher
    - Job Watcher
    - Collection Services Investigator
    - Disk Watcher
    - Plan Cache Analyzer - graphical analysis of the system's SQL Plan Cache
  - PEX Analyzer
  - VIOS Investigator
- <https://www-03.ibm.com/systems/services/labservices/iDoctor/>

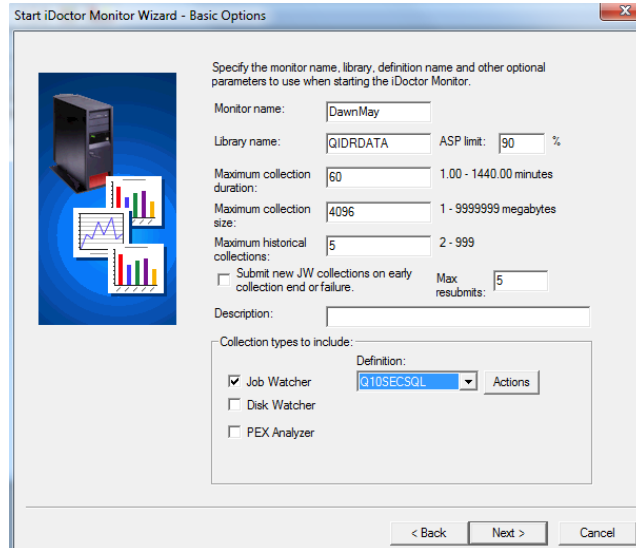
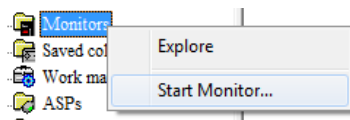
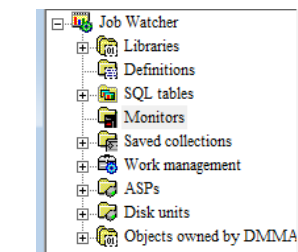
© 2017 IBM Corporation

118

# IBM iDoctor for i



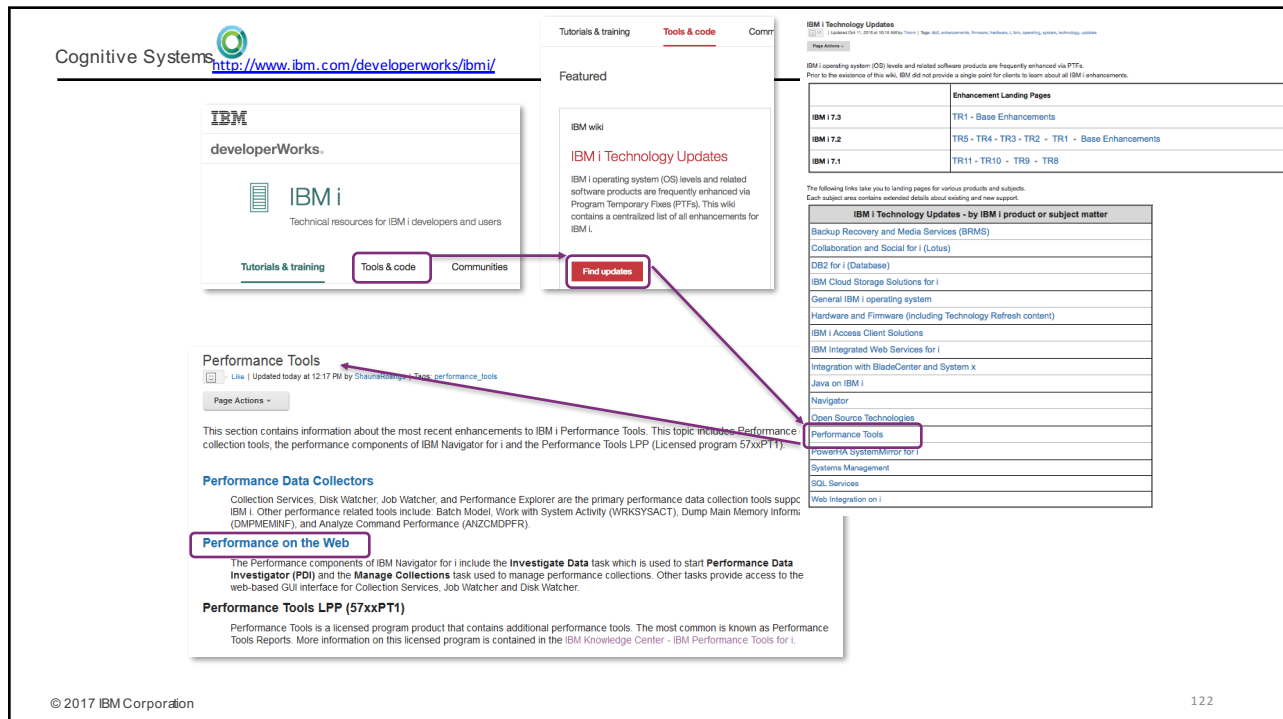
# iDoctor Monitors



## iDoctor versus Performance Data Investigator

- You have two graphical interfaces for performance data analysis... Which should you use? It depends...

Feature	iDoctor	PDI
Interface	Windows client	Browser
Wait Analysis	Yes	Yes
Collection Services	Yes	Yes
Job Watcher	Yes	Yes
Disk Watcher	Yes	Yes
Performance Explorer	Yes	Profile collections only
Database	Yes	Yes
Job Watcher Monitors	Yes	No
Customizable	Yes	Yes
User Defined graphs and queries	Yes	Yes
Update Frequency	Monthly	Twice Yearly
Support	Defect only	Standard SWMA
Chargeable	Yearly license	<ul style="list-style-type: none"> <li>Collection Services at no additional charge with i</li> <li>Disk Watcher, Database, and Performance Explorer included with base PT1 product</li> <li>Job Watcher is an additional option of PT1 and has an additional charge</li> </ul>
Experimental Features	Yes (e.g., VIOS Investigator)	No
Multinational language support	No	Yes



IBM | Technology Updates

IBM | operating system (OS) levels and related software products are frequently enhanced via Program Temporary Fixes (PTFs). This wiki contains a centralized list of all enhancements for IBM i.

**Performance Tools**

This section contains information about the most recent enhancements to IBM i Performance Tools. This topic includes Performance collection tools, the performance components of IBM Navigator for i and the Performance Tools LPP (Licensed program 57xxPT1).

**Performance Data Collectors**

Collection Services, Disk Watcher, Job Watcher, and Performance Explorer are the primary performance data collection tools supported by IBM i. Other performance-related tools include Batch Model, Work with System Activity (WRKSYSACT), Dump Main Memory Information (DMPMEMINF), and Analyze Command Performance (ANZCMDPPR).

**Performance on the Web**

The Performance components of IBM Navigator for i include the Investigate Data task which is used to start Performance Data Investigator (PDI) and the Manage Collections task used to manage performance collections. Other tasks provide access to the web-based GUI interface for Collection Services, Job Watcher and Disk Watcher.

**Performance Tools LPP (57xxPT1)**

Performance Tools is a licensed program product that contains additional performance tools. The most common is known as Performance Tools Reports. More information on this licensed program is contained in the IBM Knowledge Center - IBM Performance Tools for i.

IBM i Technology Updates - by IBM i product or subject matter

IBM i 7.x	Enhancement Landing Page
IBM i 7.3	TR1 - Base Enhancements
IBM i 7.2	TR5 - TR4 - TR3 - TR2 - TR1 - Base Enhancements
IBM i 7.1	TR11 - TR10 - TR9 - TR8



# ithankyou

[www.ibm.com/power/i](http://www.ibm.com/power/i)

## References



## IBM i Performance FAQ - a MUST read!

[http://www.ibm.com/common/ssi/cg-bin/ssialias?subtype=WH&infotype=SA&apname=STGE\\_PO\\_PO\\_USEN&htmlfid=POW031QUSEN&attachment=POW031QUSEN.PDF](http://www.ibm.com/common/ssi/cg-bin/ssialias?subtype=WH&infotype=SA&apname=STGE_PO_PO_USEN&htmlfid=POW031QUSEN&attachment=POW031QUSEN.PDF)



## IBM i on Power - Performance FAQ

April 3, 2017

## IBM i Web Sites with Performance Information



### IBM i Knowledge Center

[http://www.ibm.com/support/knowledgecenter/ssw\\_ibm\\_i/welcome](http://www.ibm.com/support/knowledgecenter/ssw_ibm_i/welcome)

- [7.1 - https://www.ibm.com/support/knowledgecenter/ssw\\_ibm\\_i\\_71/welcome.html](https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_71/welcome.html)
- [7.2 - http://www-01.ibm.com/support/knowledgecenter/ssw\\_ibm\\_i\\_72/rzahg/ic-homepage.htm](http://www-01.ibm.com/support/knowledgecenter/ssw_ibm_i_72/rzahg/ic-homepage.htm)
- [7.3 - https://www.ibm.com/support/knowledgecenter/ssw\\_ibm\\_i\\_73/rzahg/welcome.htm](https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_73/rzahg/welcome.htm)



### IBM i Performance Management

<http://www-03.ibm.com/systems/power/software/i/management/#tab2>

- Performance Management for Power Systems  
<http://www-03.ibm.com/systems/power/support/pmy/index.html>
- IBM Workload Estimator  
<http://www.ibm.com/systems/support/tools/estimator>
- iDoctor  
[http://www-912.ibm.com/i\\_dir/idoctor.nsf](http://www-912.ibm.com/i_dir/idoctor.nsf)
- Job Waits Documentation
  - [http://www.ibm.com/support/knowledgecenter/en/ssw\\_ibm\\_i\\_73/rzahg/rzahbasicwaitaccounting.htm](http://www.ibm.com/support/knowledgecenter/en/ssw_ibm_i_73/rzahg/rzahbasicwaitaccounting.htm)
  - <http://www.ibm.com/developerworks/ibmi/library/i-ibmi-wait-accounting/>
  - [http://public.dhe.ibm.com/services/us/igsc/idoctor/Job\\_Waits\\_White\\_Paper.pdf](http://public.dhe.ibm.com/services/us/igsc/idoctor/Job_Waits_White_Paper.pdf)

## Understanding “Time Dispatched on a CPU”

- **Time dispatched on a CPU (Bucket 1)**
  - Thread or task has been assigned to a processor and is NOT waiting
  - Complicated by certain features
    - Hardware Multi Threading (HMT)
      - Allows multiple threads/tasks to be assigned to a single physical processor
      - Causes bucket 1 time to be greater than actual CPU time
    - Background assisting tasks
      - Promote their CPU usage back into the client job/thread
      - Causes client thread’s bucket 1 time to be smaller than measured CPU time
    - LPAR shared/partial processors
      - Bucket 1 records time dispatched to the virtual processor
      - Bucket 1 time may be greater than CPU time because it may include time the thread/task is waiting for the physical processor behind the virtual processor

Bucket 1 does NOT equal CPU Time

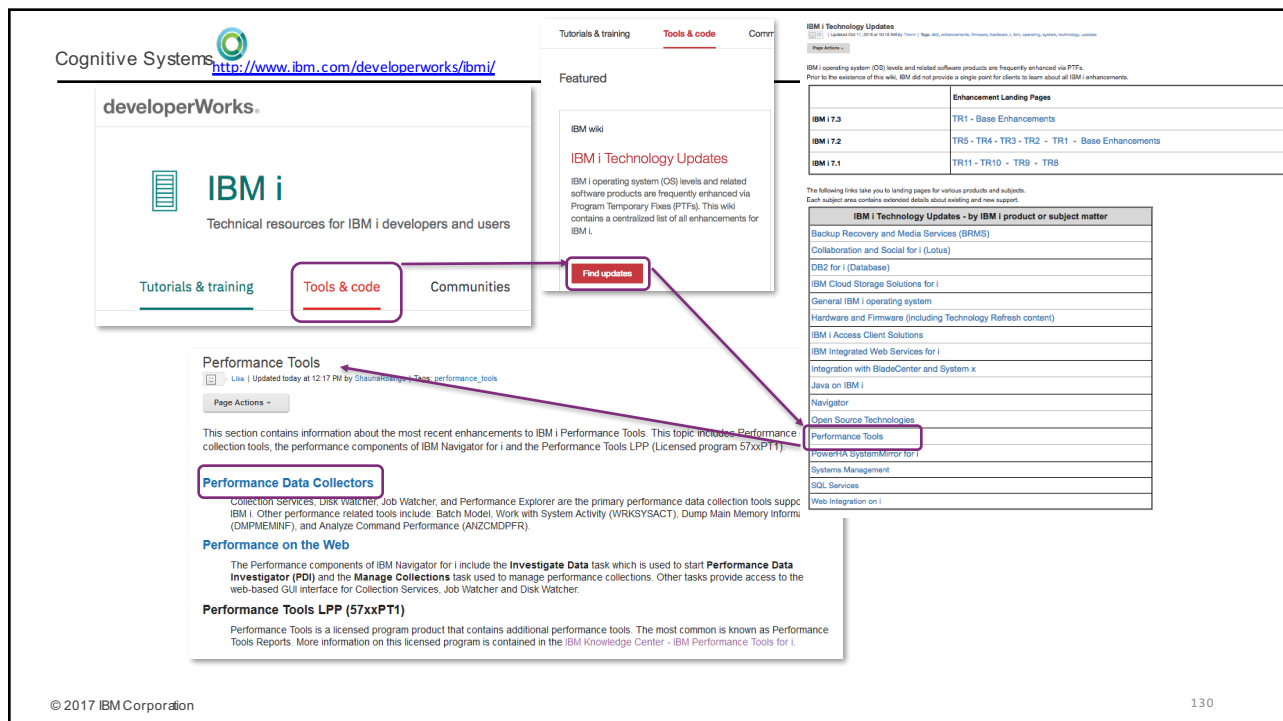
## Understanding “CPU Queuing”

- **CPU Queuing (Bucket 2)**
    - Thread or task has been assigned to a processor and is waiting for the CPU to become available
      - Too much work on the partition causing threads to need to wait for the processors
      - Spiky workloads
      - Workload Groups
      - Shared processors
- Latency due to hypervisor sharing the physical processors among multiple partitions



# CPU Queueing ... technical details

- A spikey workload.  
I/O completing in batches can cause this but so can software design. For instance, there are some seize-lock scenarios that unblock herds of tasks. A task switch trace can be used to instrument how the workload is behaving. This is a common reason.
- Shared processors.  
When there are idle processors, tasks are enqueued directly to a special per-processor queue until they overflow to a shared queue. On a dedicated processor system, the time it takes for the wait-state task to dequeue the task from this special queue and dispatch it is very low, but non-zero. The latency can be slightly higher when donation is used. In a SPLPAR, the latency can vary depending on outside factors, namely shared pool utilization, because the hypervisor is sharing the physical processors among multiple partitions. The separate components of VCPU latency is instrumented in Collection Services as of 7.1.  
  
Nodal dispatching.  
In a multi-node system, load imbalance could cause tasks to be queued in some nodes while other nodes are idle. This is generally not a problem with the default nodal affinity scheduling policy. However, a QTHDRSCAFN value set to \*HIGH instead of the default \*NORMAL can cause this problem. Node imbalance can be instrumented in 6.1 and later.
- Workload capping.  
The workload group can be over-committed even though the system is under-committed. There is separate instrumentation of workload group dispatch latency in Collection Services.



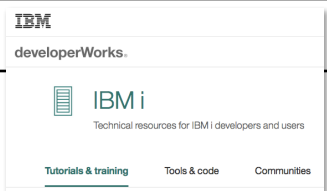
The screenshot shows the IBM developerWorks website. The main navigation bar includes 'Tutorials & training', 'Tools & code', and 'Communities'. The 'Tools & code' section is highlighted with a red box. Below it, the 'Performance Tools' article is visible, with a red box around the 'Performance Data Collectors' sub-section. To the right, the 'IBM i Technology Updates' section is shown, featuring a table of updates and a list of links to landing pages for various products.

IBM i	Enhancement Landing Page
IBM i 7.3	TR1 - Base Enhancements
IBM i 7.2	TR5 - TR4 - TR3 - TR2 - TR1 - Base Enhancements
IBM i 7.1	TR11 - TR10 - TR9 - TR8

The following links take you to landing pages for various products and subjects. Each subject area contains extended details about existing and new support.

IBM i Technology Updates - by IBM i product or subject matter
Backup Recovery and Media Services (BRMS)
Collaboration and Social for i (Lotus)
DB2 for i (Database)
IBM Cloud Storage Solutions for i
General IBM i operating system
Hardware and Firmware (including Technology Refresh content)
IBM i Access Client Solutions
IBM Integrated Web Services for i
Integration with BladeCenter and System x
Java on IBM i
Navigator
Open Source Technologies
Performance Tools
PowerHA SystemMirror for i
Systems Management
SQL Services
Web Integration on i

## IBM i Performance on developerWorks



- developerWorks  
<http://www.ibm.com/developerworks/ibmi/>
- Performance Tools  
[https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/IBM\\_i\\_Technology\\_Updates/page/Performance\\_Tools](https://www.ibm.com/developerworks/mydeveloperworks/wikis/home?lang=en#/wiki/IBM_i_Technology_Updates/page/Performance_Tools)
- IBM i Performance Data Investigator  
<http://www.ibm.com/developerworks/ibmi/library/i-pdi/index.html>
- IBM i Performance Data Investigator – Edit Perspectives  
<http://www.ibm.com/developerworks/ibmi/library/i-pdiedit/index.html>
- IBM i Wait Accounting  
<http://www.ibm.com/developerworks/ibmi/library/i-ibmi-wait-accounting/>
- How to use the Batch Model performance tool  
<https://www.ibm.com/developerworks/ibmi/library/i-how-to-use-the-batch-model-performance-tool/>
- A new way to analyze historical performance data on IBM i  
<https://www.ibm.com/developerworks/ibmi/library/i-new-way-to-analyze-historical-performance-data-trs/>

© 2017 IBM Corporation

131



Cognitive Systems

### i Can Blog of Blogs

Updated Sep 14, 2017 by DawnMay | Tags: [ican](#)

Page Actions ▾



**i Can**  
<http://www.ibmsystemsmag.com/Blogs/i-Can/>

For a simple list of all blogs on one page:

**"i Can" Blog of Blogs**  
[https://www.ibm.com/developerworks/community/wikis/home?lang=en - /wiki/Power\\_Systems/page/i\\_Can\\_Blog\\_of\\_Blogs](https://www.ibm.com/developerworks/community/wikis/home?lang=en - /wiki/Power_Systems/page/i_Can_Blog_of_Blogs)

This page is a listing of all the "i Can" blogs in chronological order (oldest at the bottom).

- End Subsystem Options
- Manage Trigger Programs in Production
- Trace Output Formatted for PCAP
- Route FTP and SMTP Workloads to Their Own Subsystem
- Integration Between Navigator and Access Client Solutions
- Client Advisory Council Meetings in June 2017
- Graph History -- Stacked Charts
- COMMON Americas Advisory Council (CAAC) and RFEs
- Subsystem Configuration Documentation
- IBM i Monitors: Automatic Notification and Replacement Variables
- PowerVM LinkedIn Group
- Copy an Open Spooled File
- Subsystem Management with Navigator for i
- Determining the Size of QTEMP
- SNMPv3 Auditing and Other Enhancements
- Here's to the Women in IT!
- February 2017 IBM i Announcements
- Application Administration with Access Client Solutions
- Application Administration with Navigator for i
- Optical Containers

© 2017 IBM Corporation

132

## IBM i Performance Analysis Workshop

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

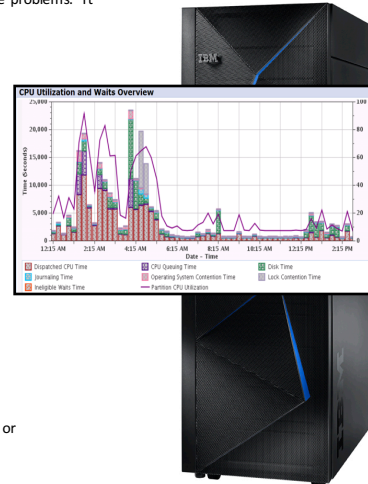
Managing and analyzing the data can be quite complex. During this workshop, the IBM Systems Lab Services IBM i team will share useful techniques for analyzing performance data on key IBM i resources, and will cover strategies for solving performance problems. It will aid in building a future foundation 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 public workshop availability and enrollment: [IBM i Performance Analysis Workshop](#)
  - For additional information, including private workshops, please contact Eric Barsness at [ericbar@us.ibm.com](mailto:ericbar@us.ibm.com) or Stacy Benfield at [stacylb@us.ibm.com](mailto:stacylb@us.ibm.com), members of Systems Lab Services Performance Optimization team.



IBM Systems Lab Services Power Systems Delivery Practice - [ibm.com/systems/services/labservices](http://ibm.com/systems/services/labservices) - [ibmsls@us.ibm.com](mailto: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](mailto:ericbar@us.ibm.com) for more details.

## Performance and Scalability Services

- The IBM i Performance and Scalability Services Center can provide facilities and hardware IN ROCHESTER to assist you in testing hardware or software changes
  - “Traditional” benchmarks
  - Release-to-release upgrades
  - Assess and tune application and database performance
  - Stress test your system
  - Determine impact of application changes
  - Proofs of Concept (e.g. HA alternatives; SSD analysis, external storage, etc.)
  - Evaluate application scalability
  - Capacity planning
- ... all with the availability of Lab Services IBM i experts and development personnel
- To learn more about Power offerings:
  - <https://www.ibm.com/it-infrastructure/services/lab-services/power>

## 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 inquiries, 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 5L, 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, Active Memory, Balanced Warehouse, CacheFlow, Cool Blue, IBM Systems Director VMControl, pureScale, TurboCore, ChipHopper, Cloudscape, DB2 Universal Database, DS4000, DS6000, DS8000, EnergyScale, Enterprise Workload Manager, General Parallel File System, GPFS, HACMP, HACMP/6000, HSM, 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+, POWER7, System i, System p, System p5, System Storage, System z, 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 full list of U.S. trademarks owned by IBM may be found at: <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries. AltVec is a trademark of Freescale Semiconductor, Inc.

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

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

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

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.

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

SPECint, SPECfp, SPECjbb, SPECweb, SPECAppServer, SPEC OMP, SPECviewperf, SPECcapc, SPECcpc, SPECjvm, SPECmail, SPECmap and SPECcfs are trademarks of the Standard Performance Evaluation Corp (SPEC).

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.

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

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

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

Revised December 2, 2010