Université IBM i 2018

16 et 17 mai

IBM Client Center Paris

S21 – i Can Find your Performance Bottlenecks

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

Performance Disclaimer

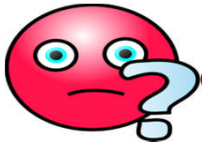
- “it depends …”

This presentation is for you if…..

- You added additional hardware resources to your partition - only to find out that your batch job did not run any faster

- CPU utilization is low, yet your job(s) seem to run slow

- Your boss asks you to find ways to speed up your end-of-month batch processing, and you don't know where to begin!



This presentation is for you if…..

- You want to learn more about the rich performance instrumentation that is available on IBM i

Agenda

- **Introduction to IBM i Wait Accounting**
  - Concepts
  - Wait Buckets
- **Tools**
  - Collection Services, Job Watcher
  - Performance Data Investigator, iDoctor
- **Recommendations**
- **References**

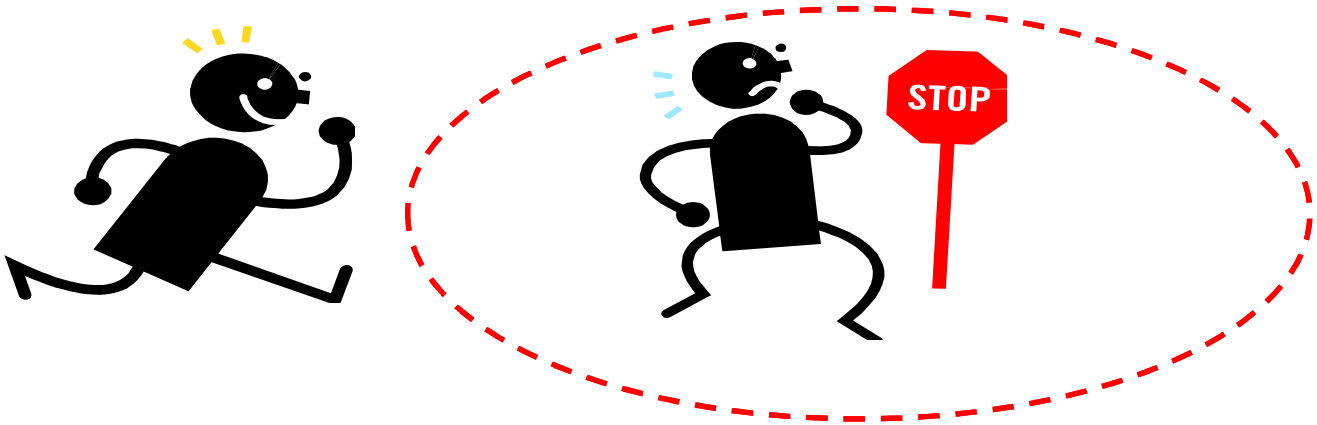
Introduction to Wait Accounting

Performance Fact:

“All computers wait at the same speed”

What is Wait Accounting?

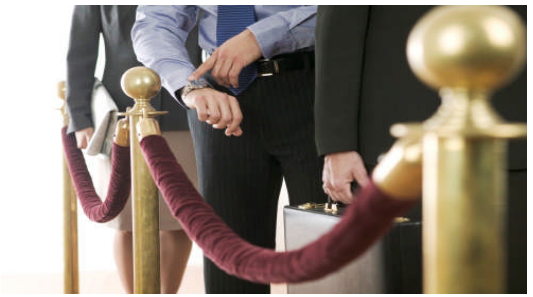
Wait Accounting = the ability to determine what a job is doing when it is not running



Exclusive!! Patented IBM i technology built into IBM i

Wait Accounting Overview

- When a job is not running (using CPU), it is waiting
  - *But why is it waiting? How long is it waiting? And 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 (automatically!) 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

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 it’s 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, current user profile, pool ID, etc.

32 Wait Buckets (6.1 and beyond)

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)
22. Classic Java other (up to 6.1)
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**

RED = Blocked Waits

http://www.ibm.com/developerworks/ibmi/library/i-ibmi-wait-accounting/
http://public.dhe.ibm.com/services/us/igsc/idoctor/Job_Waits_White_Paper_61_71.pdf

Common Waits that Applications use

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

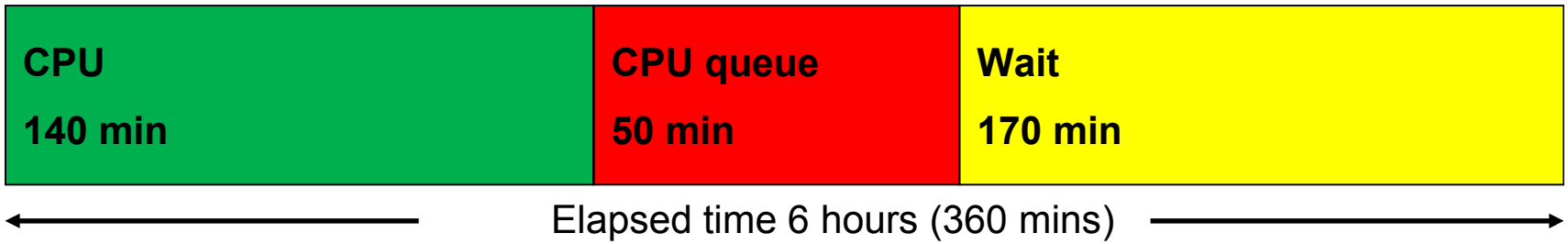
Wait Accounting – “Run-wait” signature

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.

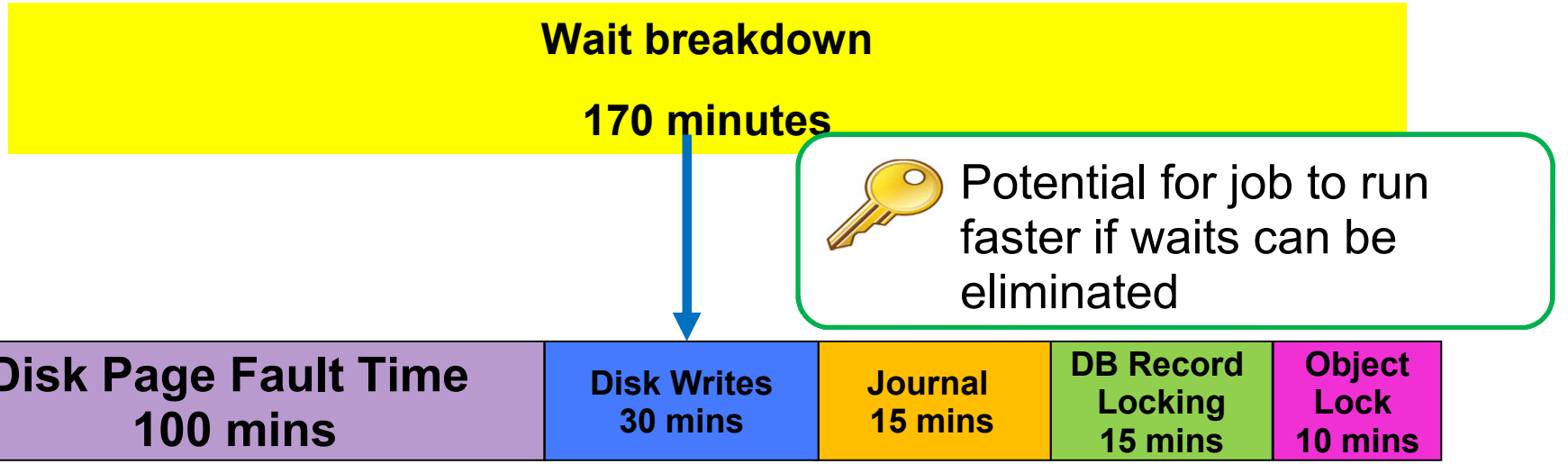
Consider the following:

Batch job with total run time of 6 hours

Run-wait signature

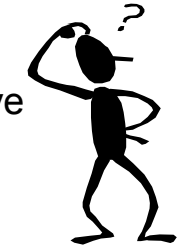


Wait Accounting – “Run-wait” signature



Now you can start asking questions such as:

- Are my pool sizes appropriate? What object(s) is the faulting occurring on?
- Is the write cache being overrun? Is the application forcing writes out synchronously? Excessive database file (opens)/closes?
- Are all the journals optimally configured? Are unnecessary objects being journaled?
- Am I locking records or objects unnecessarily?



Why you should leverage Wait Accounting!!

- Helps you understand both system and 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, or contention time?
  - Invest resources where greatest benefit will be
    - Fixing application vs. adding hardware

- 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

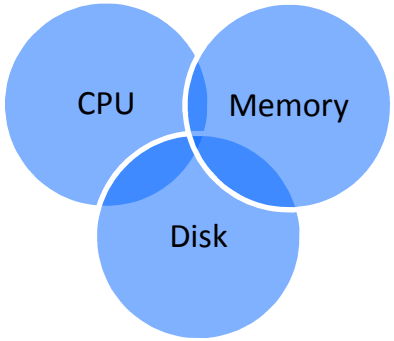
- Provides valuable clues to help analyze performance issues as they arise

- Instrumentation part of base IBM i operating system, IBM tools available to help you analyze

Goals…..

✓ Maintain a balanced system

- Adequate processor, memory, and I/O allocated for workloads(s)
- Jobs aren't waiting to use resource
- Don't focus on one and neglect others!



✓ Minimize contention times

- Reduce time jobs spend waiting on database record locks, object locks, etc.

Tools for analyzing Wait Accounting information

Wait Accounting - Data Collectors

- **Collection Services**

  - Collects data automatically 24 X 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 complex performance issues

Wait Accounting - Visualization Tools

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

Performance Data Investigator (PDI)

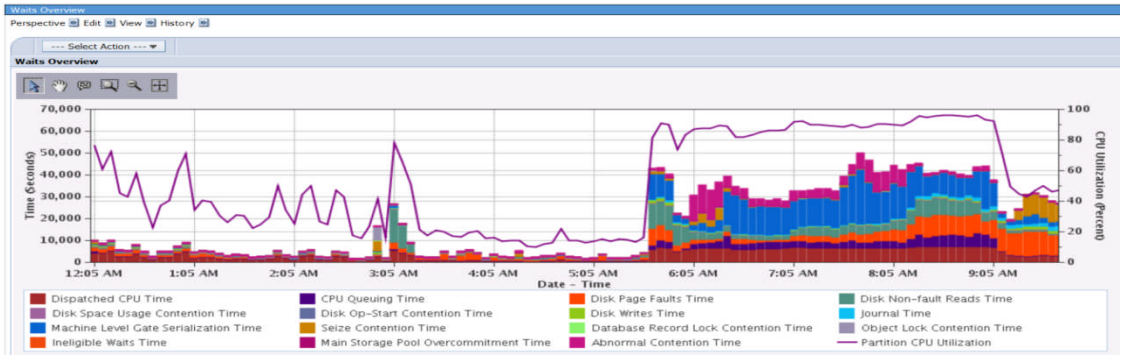
- Component in IBM Navigator for i (browser-based)
- Nothing to install, can view Collection Services for “free”
- http://www.ibm.com/developerworks/ibmi/library/i-pdi/index.html

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

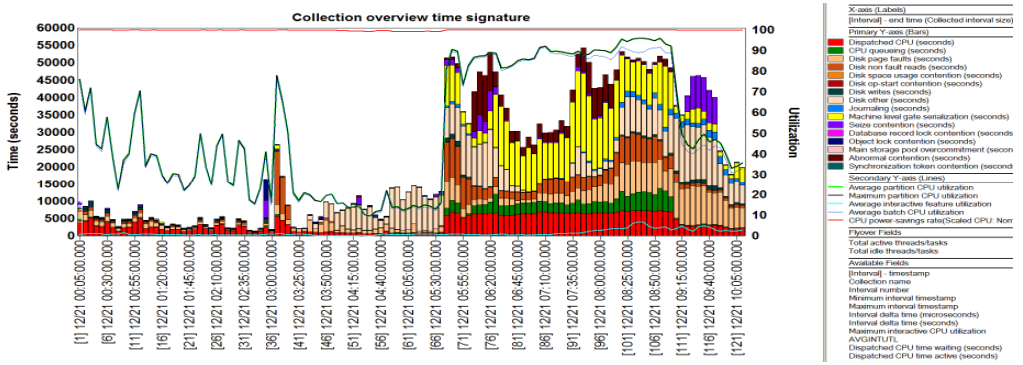
Wait Accounting IBM i Graphical Analysis Tools

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



← PDI

iDoctor ->

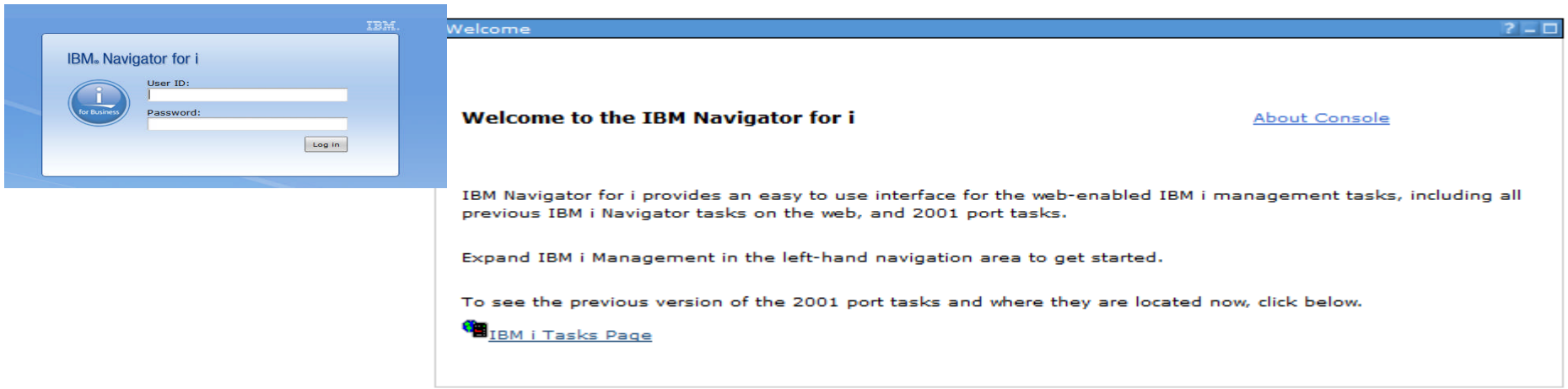


Using Performance Data Investigator (PDI)

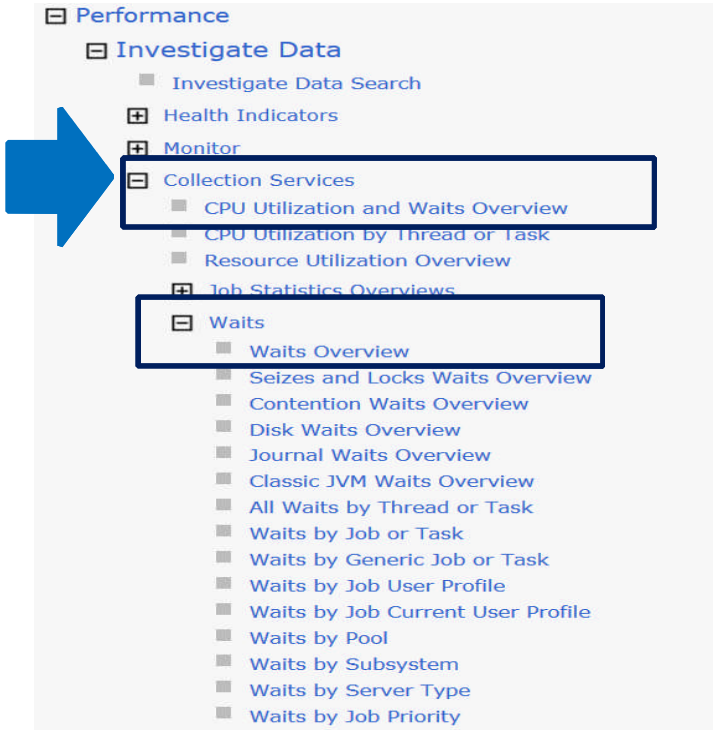
IBM Navigator for i is the Web console for managing IBM i

- 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:

Select Perspective

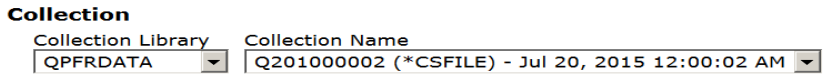
- Option 1: **CPU Utilization and Waits Overview**

– Combines related “blocked” waits into higher level Buckets

e.g. Disk time = Disk write + Disk page fault + Disk non-fault read + Disk space usage contention + Disk other

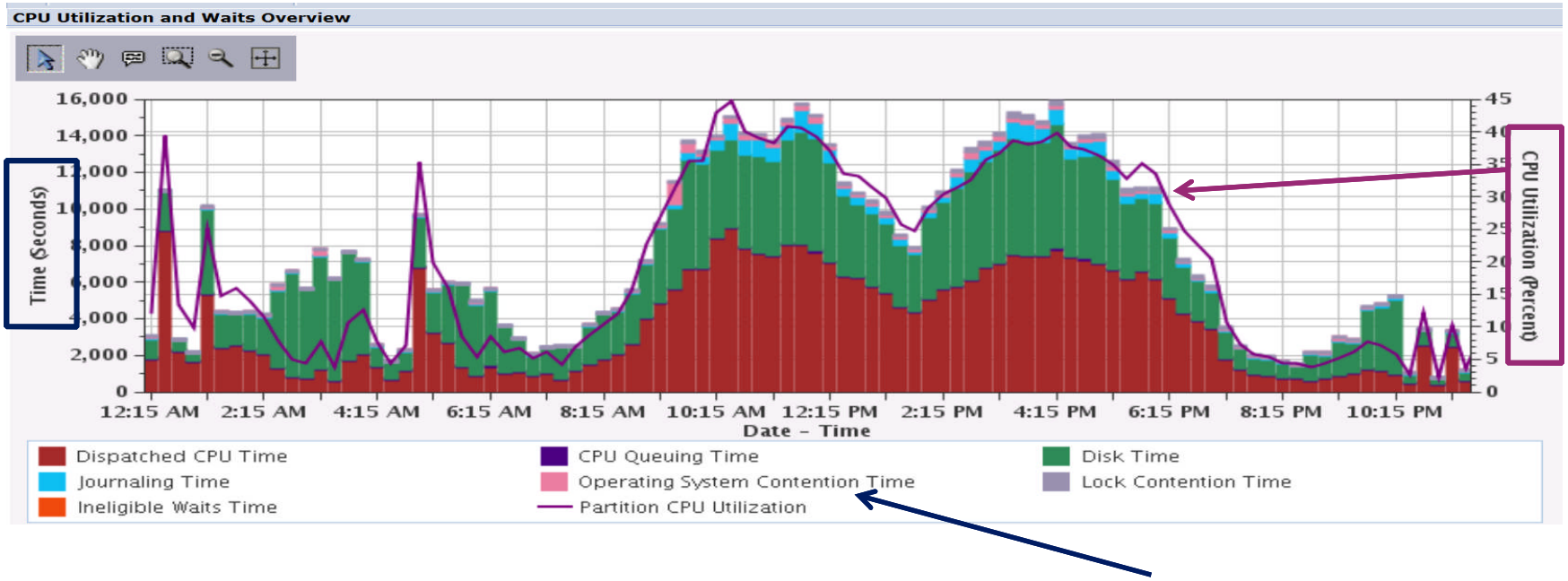
- Option 2: **Waits Overview**

– All individual “blocked” wait buckets shown



Select library/collection of interest:

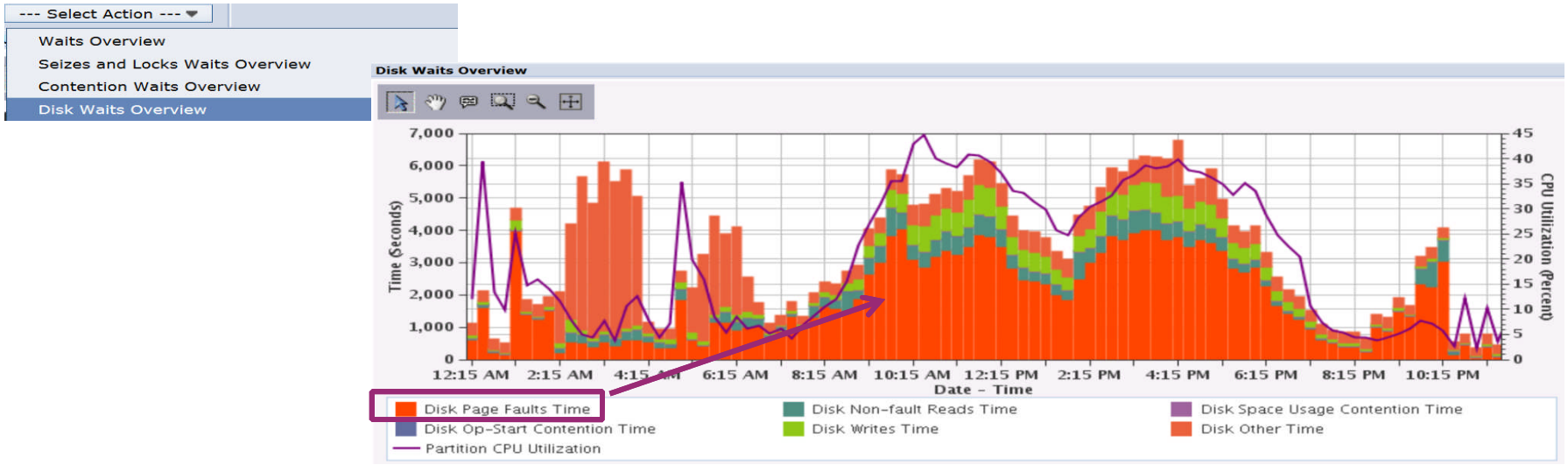
CPU Utilization and Waits Overview – “system run-wait signature”



- 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.

CPU Utilization and Waits Overview – drilldown analysis

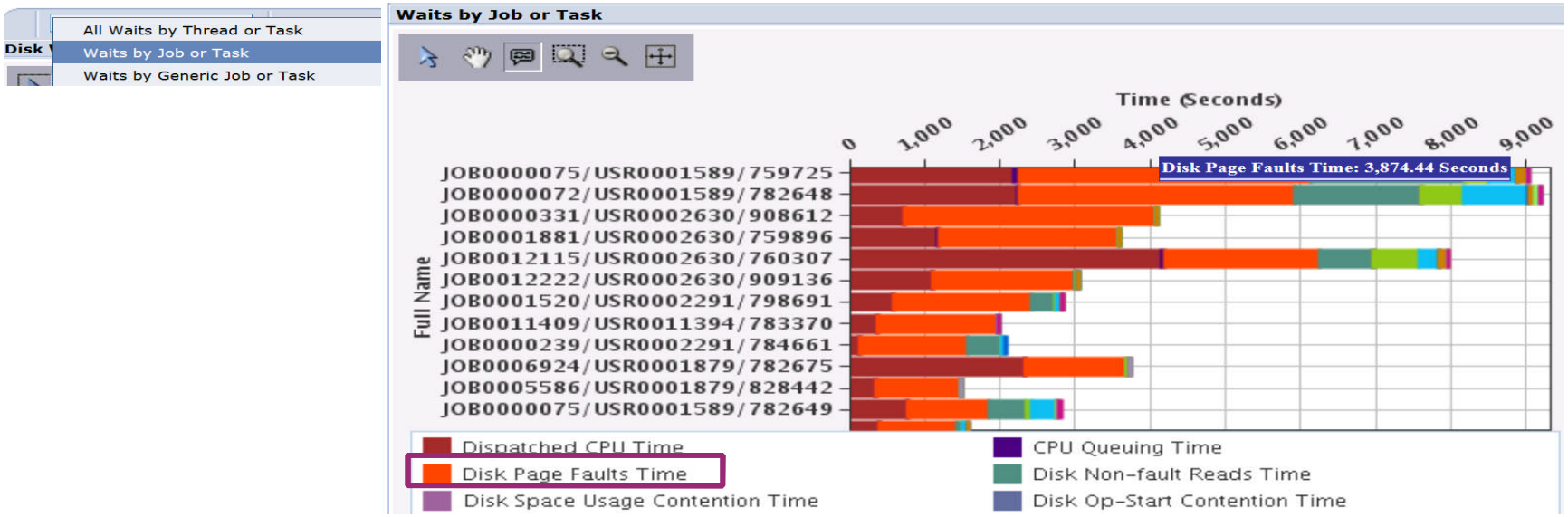
Because Disk wait time was fairly significant, drilldown to Disk Waits Overview to further examine the detailed waits contributing to this time:



- Can now 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

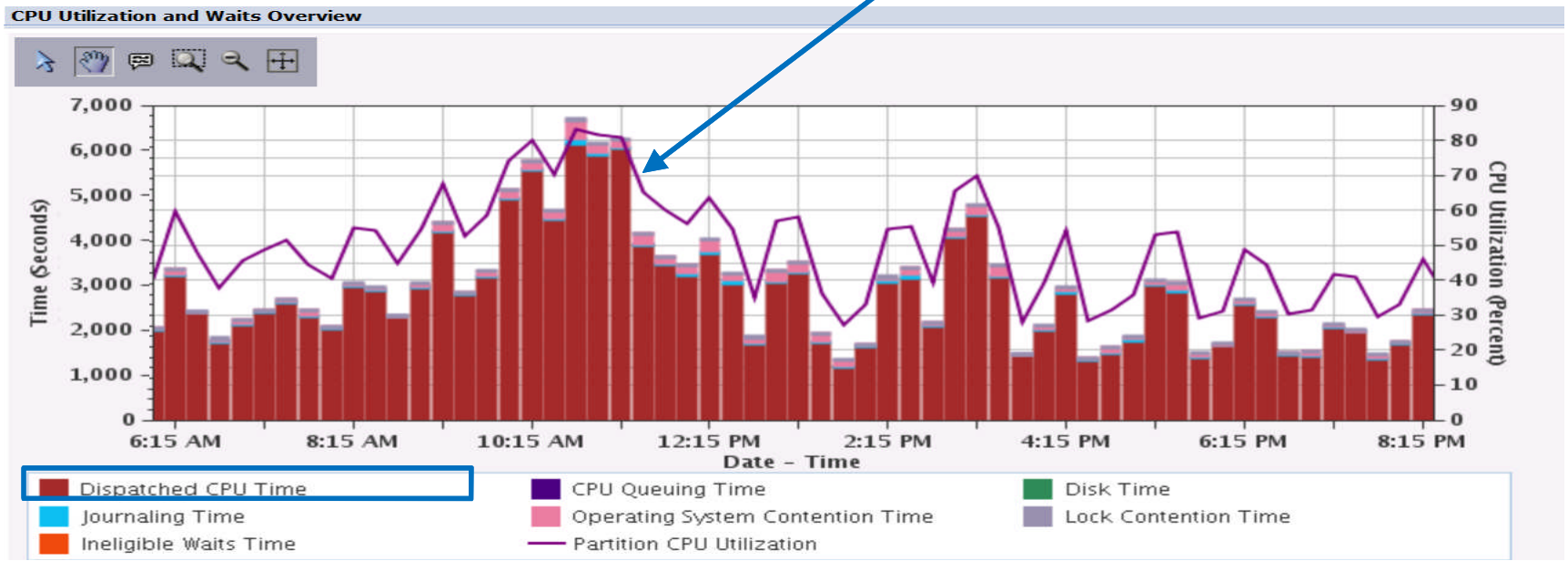
The next question likely would be which job(s) are incurring this wait time. Drilling down further, we can see the list of jobs incurring this wait time:



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

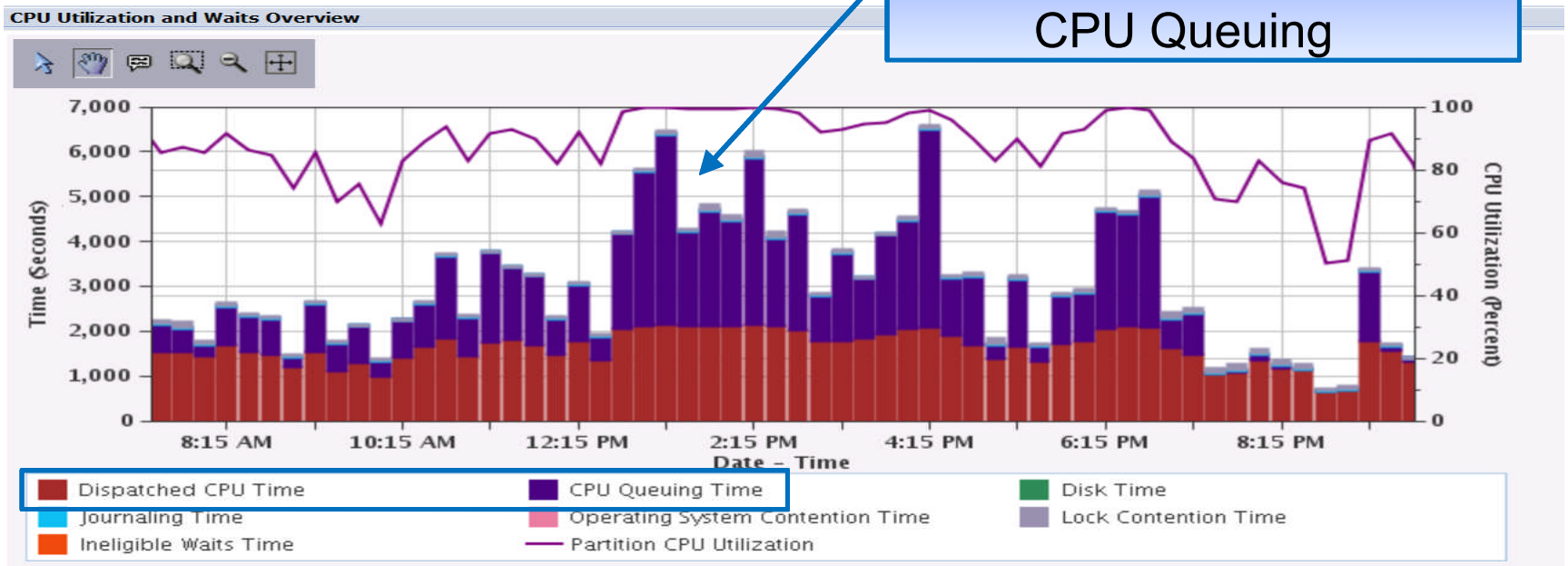
Efficient System with Little Waiting

Primarily Dispatched CPU Time

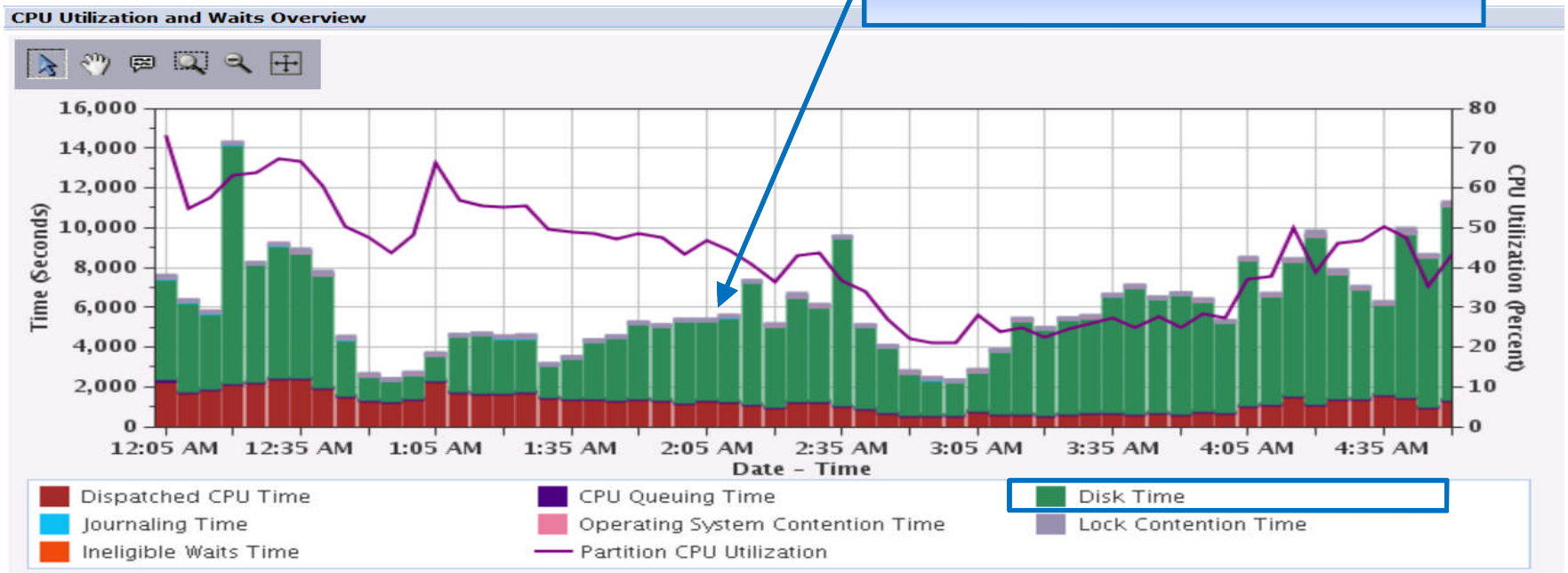


Processor Bound System

Dispatched CPU
+
CPU Queuing

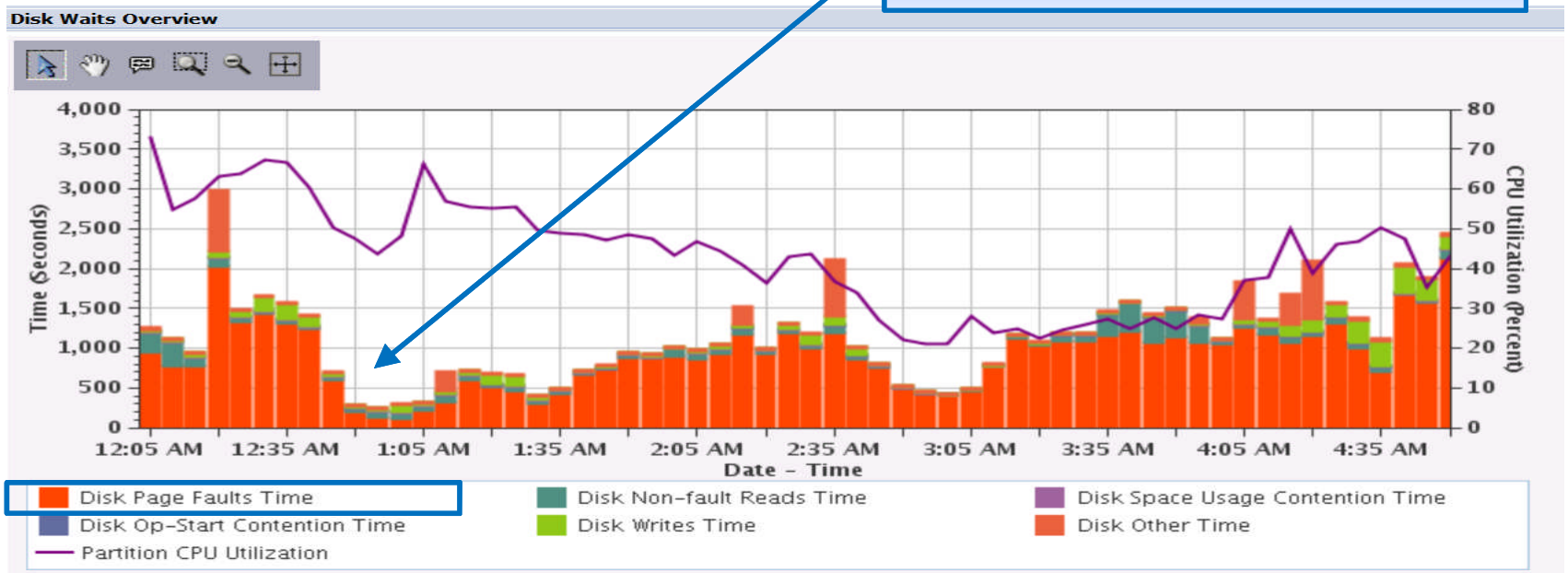


Disk Bound System

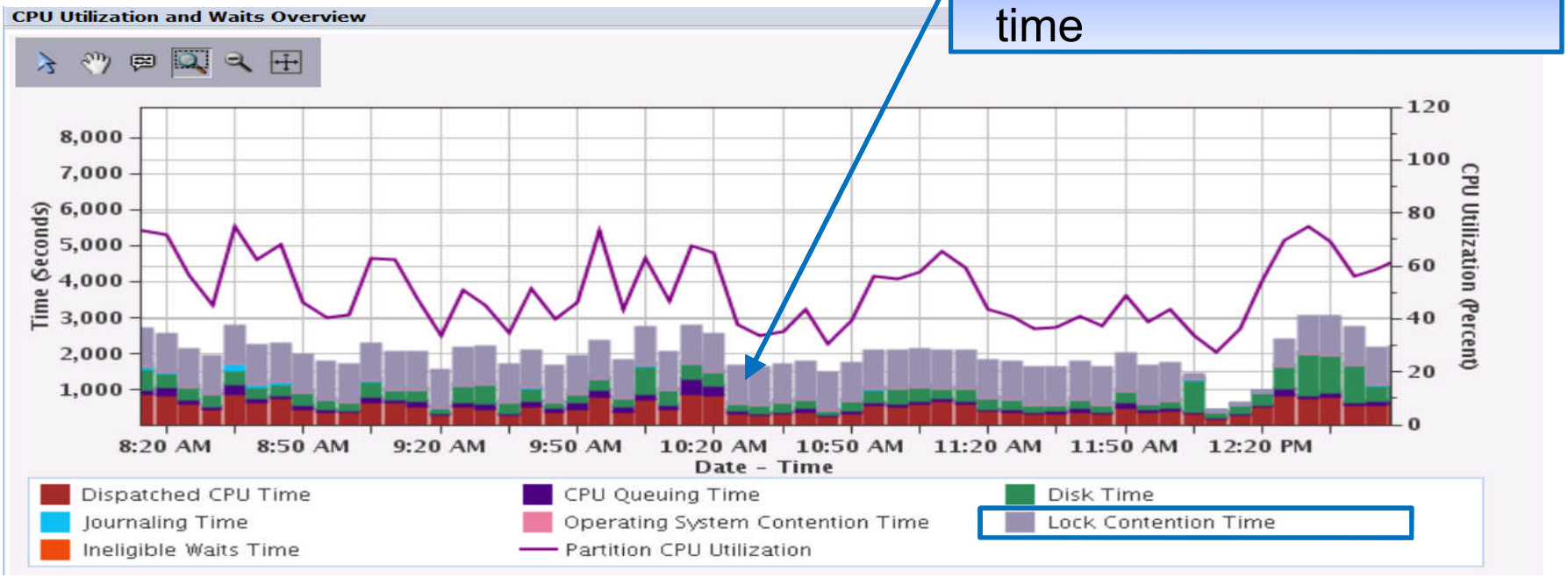


I/O – Further Investigation

Disk Page Faults Wait Time

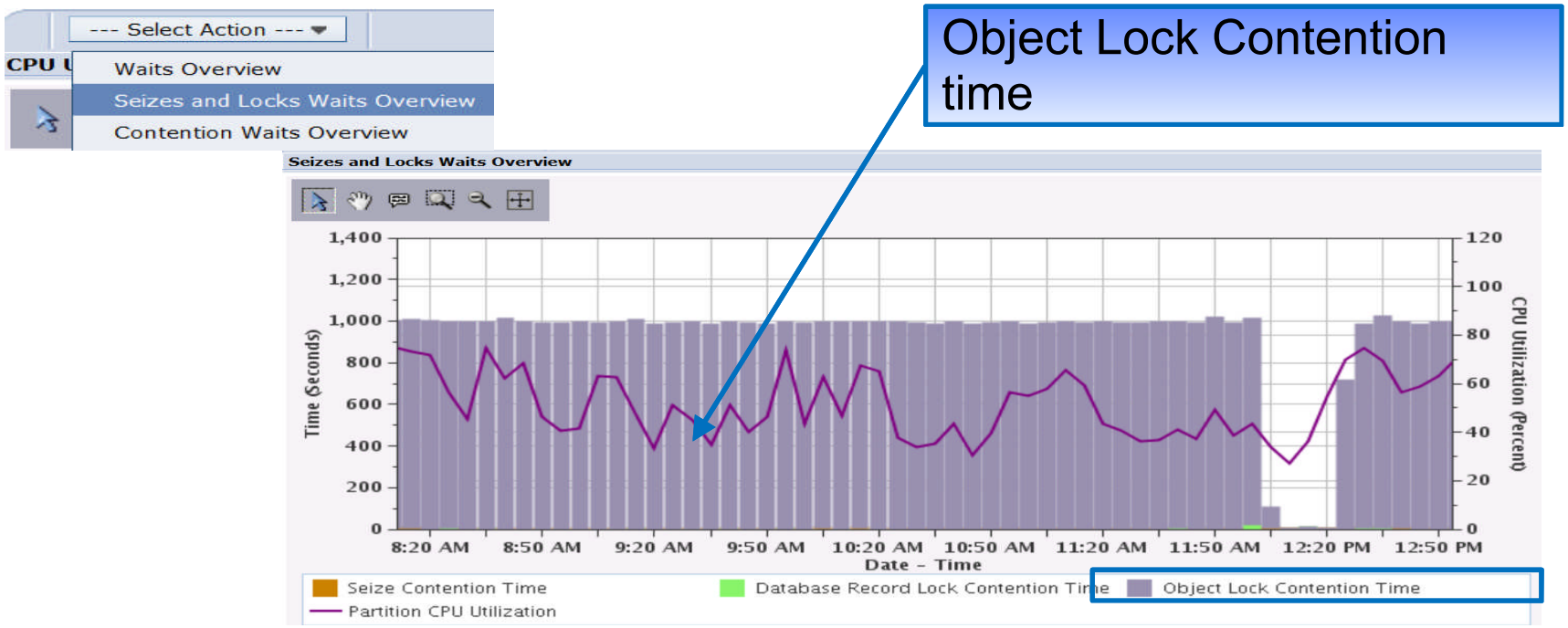


Lock Contention Time Bottleneck



Job Watcher data is typically needed to solve lock related issues.

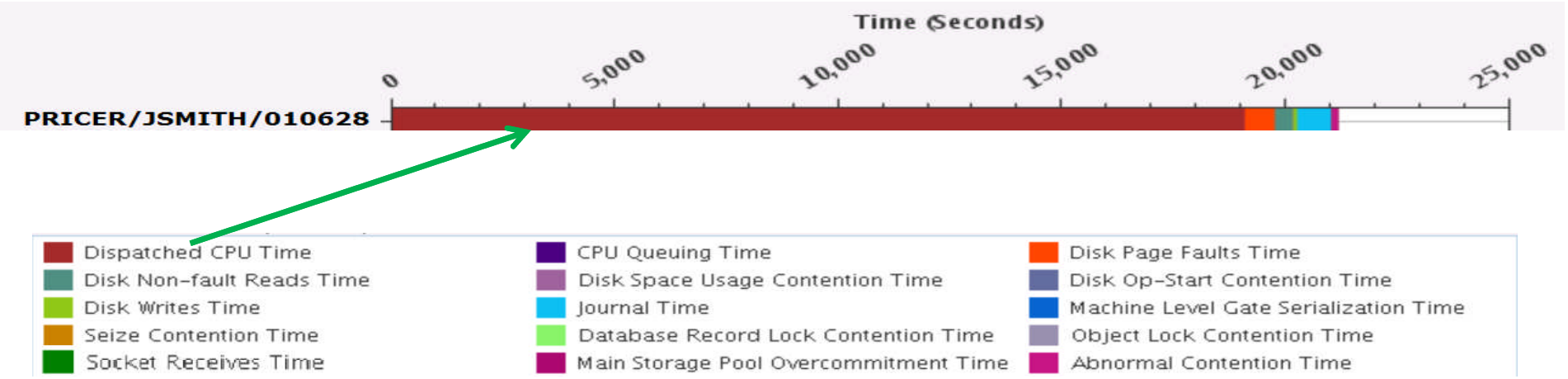
Lock – Further Investigation



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

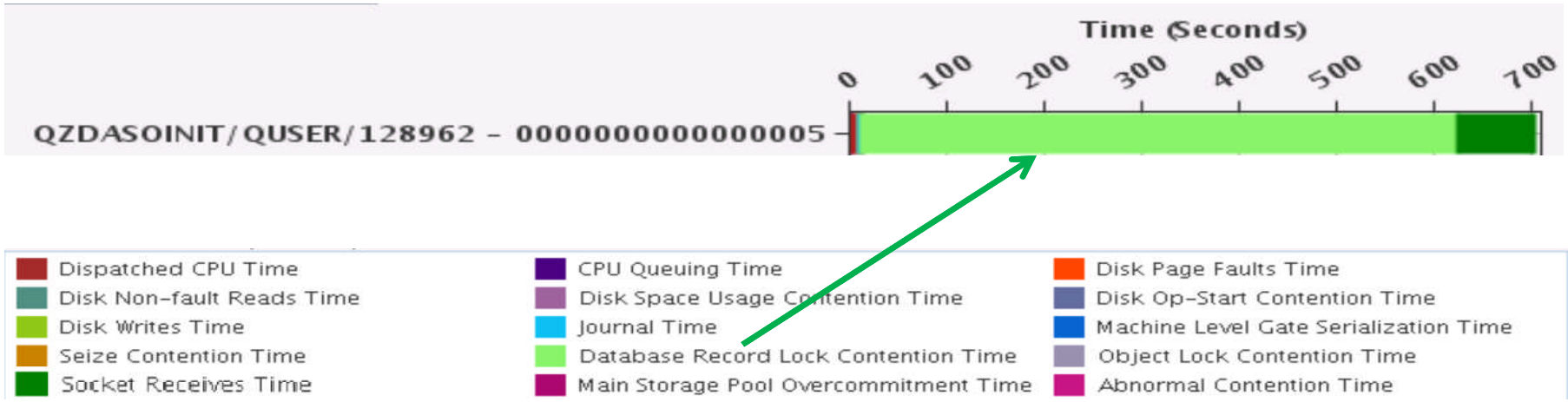
Wait Accounting at a Job Level

Would this job benefit from additional memory?



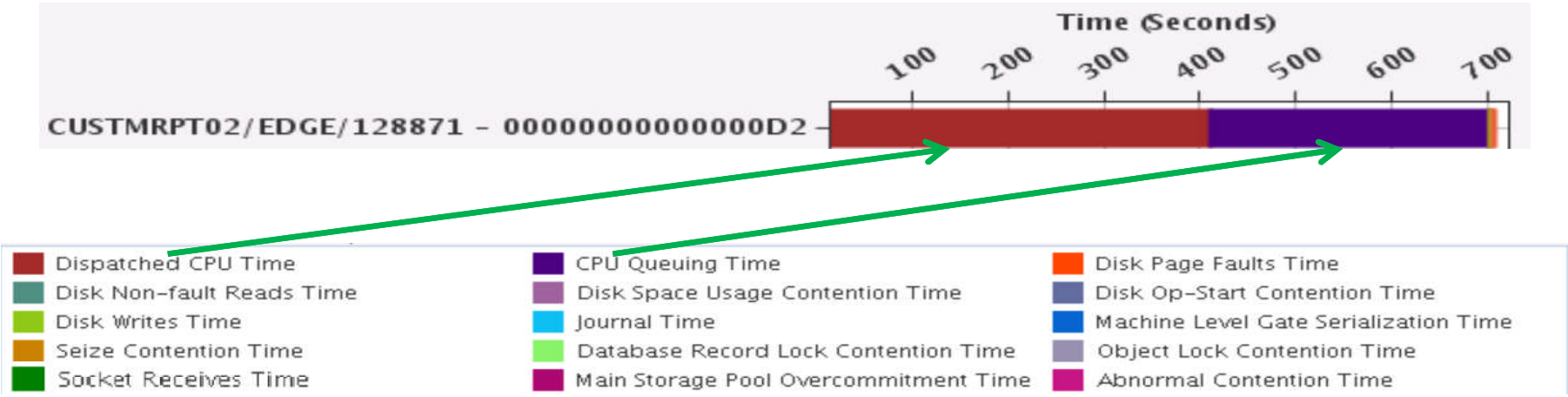
Wait Accounting at a Job Level

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?



Additional Wait Information - Counts

Database Record Lock Contention Time (Seconds)	Database Record Lock Contention Counts	Database Record Lock Contention Contributing Jobs	Object Lock Contention Time (Seconds)	Object Lock Contention Counts	Object Lock Contention Contributing Jobs	Disk Space Us Contention Counts
0	0	0	0.01	22	3	
0	0	0	0.01	18	1	
0	0	0	0.01	21	2	
0	0	0	0.01	21	2	
0	0	0	0.01	20	1	
0	0	0	224.31	35	3	
0	0	0	0.75	21	2	
0	0	0	0.01	21	2	
0	0	0	0.01	20	1	
0	0	0	0.01	20	1	
0	0	0	0.01	20	1	
0	0	0	0.01	20	1	
0	0	0	0.01	20	1	
0	0	0	0.01	20	1	
0	0	0	0.01	20	1	
0	0	0	0.01	20	1	
0	0	0	0.01	20	1	
0	0	0	0.01	20	1	
0	0	0	0.01	20	1	
0	0	0	0.01	21	2	

- Wait counts and number of contributing jobs are also tracked
- Information can be found in PDI tables (“Show as table”)
- Counts tracked at both partition and job level

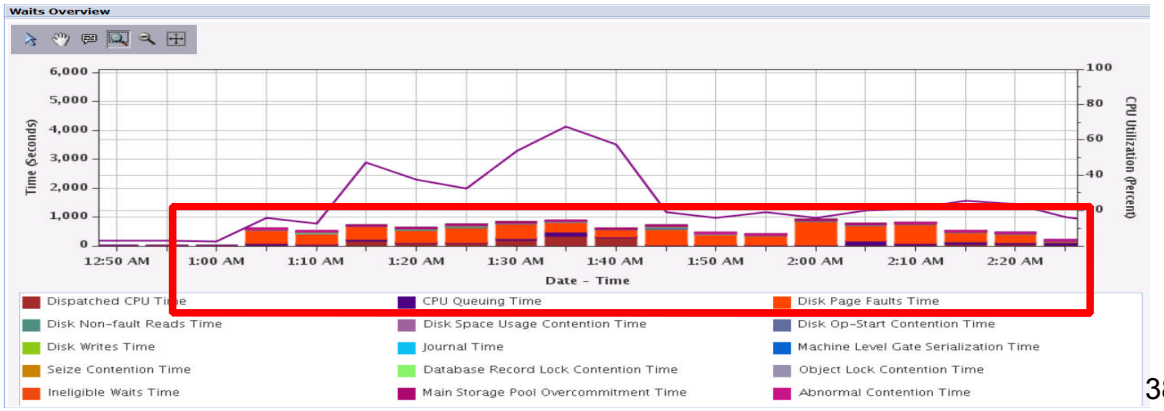
A few other things to know about waits…

- Some waits are “expected” and others “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

Wait Accounting Analysis Strategy

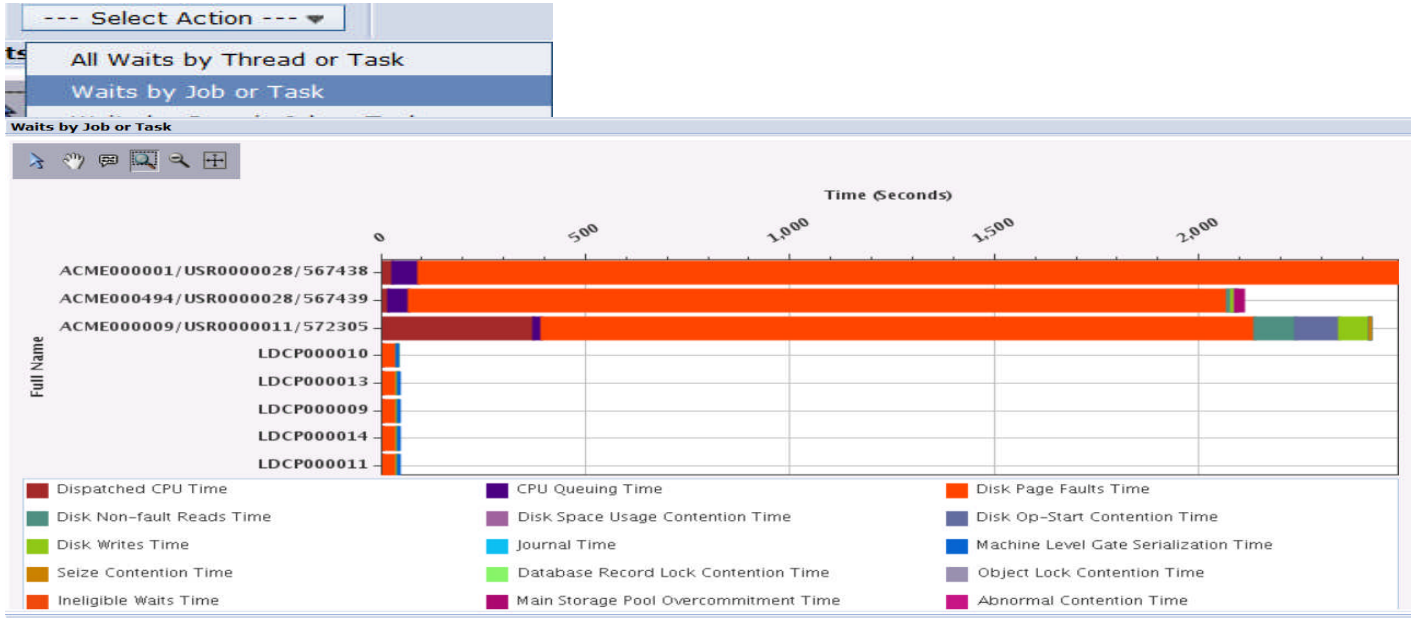
- Understand the “big picture” first
  - Understand overall partition characteristics first and where system bottlenecks may be that affecting your application
  - Use **Collection Services** data to scope problem

My night operator complains that jobs run slow between 1:00 a.m. and 2:30 a.m. every night….



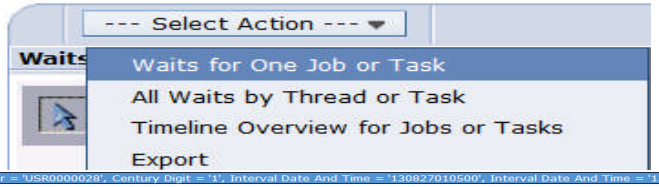
Wait Accounting Analysis Strategy

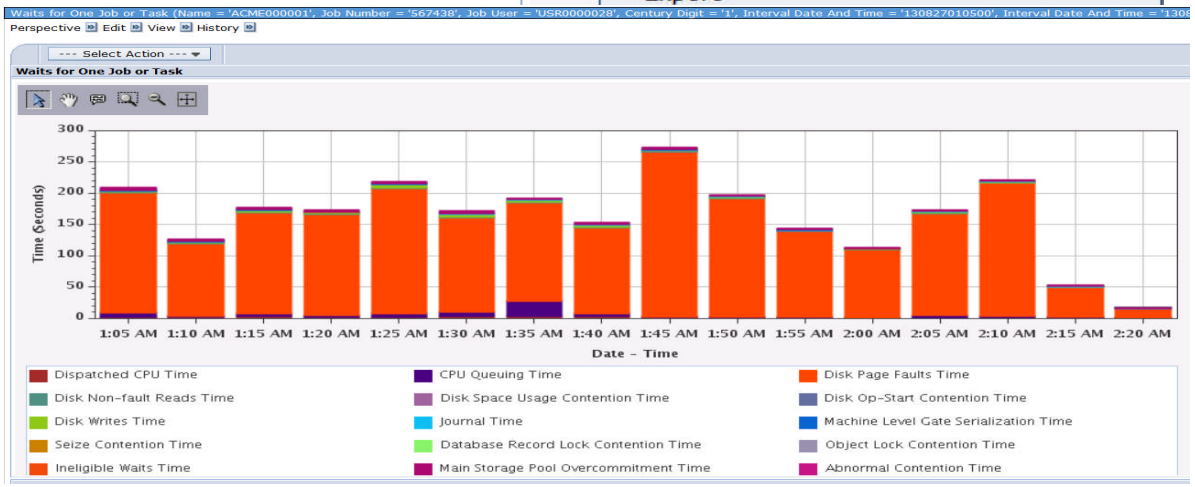
- Disk Page Fault Wait time likely is the cause. Next, you want to understand which jobs are impacted by this wait time (is it one or many?):



Wait Accounting Analysis Strategy

- Drill down to the job level to understand whether a particular job is incurring the majority of wait in just a few intervals, or does it incur wait time over many intervals?
  - Select a job (click on) then





Wait Accounting Analysis Strategy

- We now know the wait time likely responsible for slow down, and the jobs impacted.
  - We can start to take a look at memory/pool sizes, paging, etc
  - We may need to involve application development team to understand what these jobs are doing

If you can't solve the problem with Collection Services data…..

- Continue detailed analysis at a Job Level using **Job Watcher**
  - Narrow focus to interesting timeframes / jobs
  - Many more job level details available (in this case, we would know the object being waited on, and call stacks for clues as to what the job/programs are doing)

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

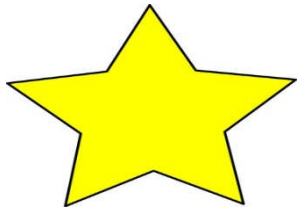
Job Watcher - Holders versus Waiters

- 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

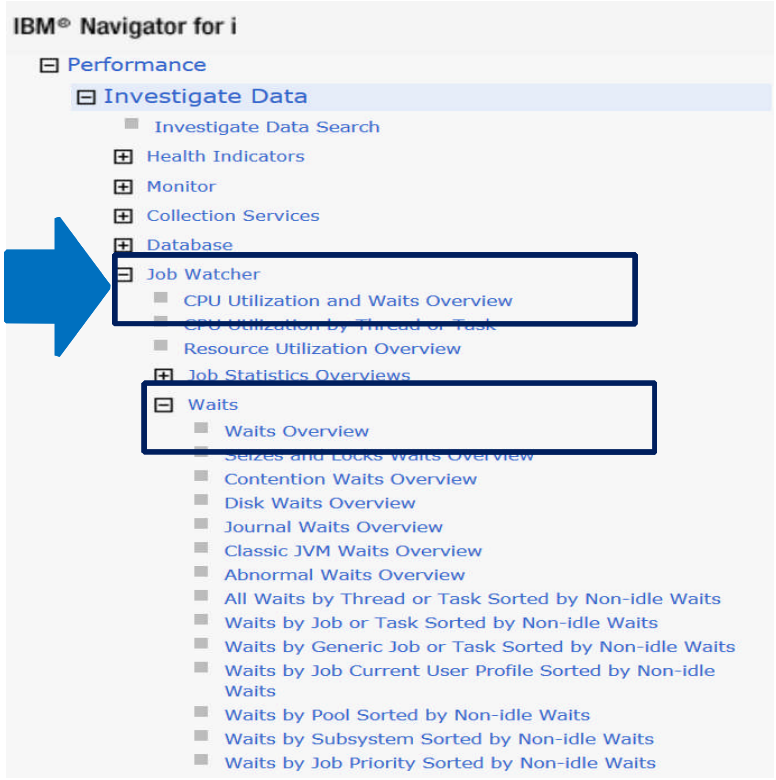
- IBM i also maintains call stacks for every job/thread/task

- 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



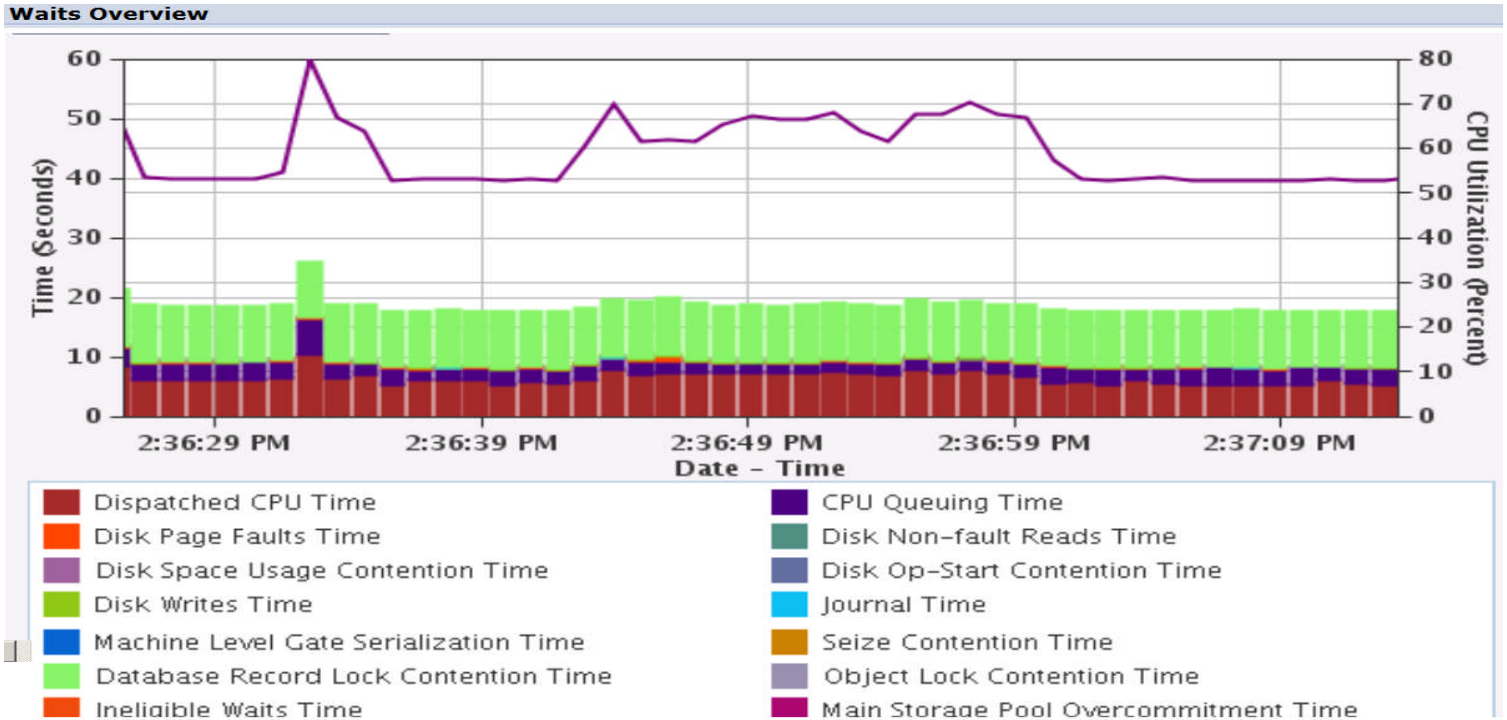
Performance -> Investigate Data -> Job Watcher:

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

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

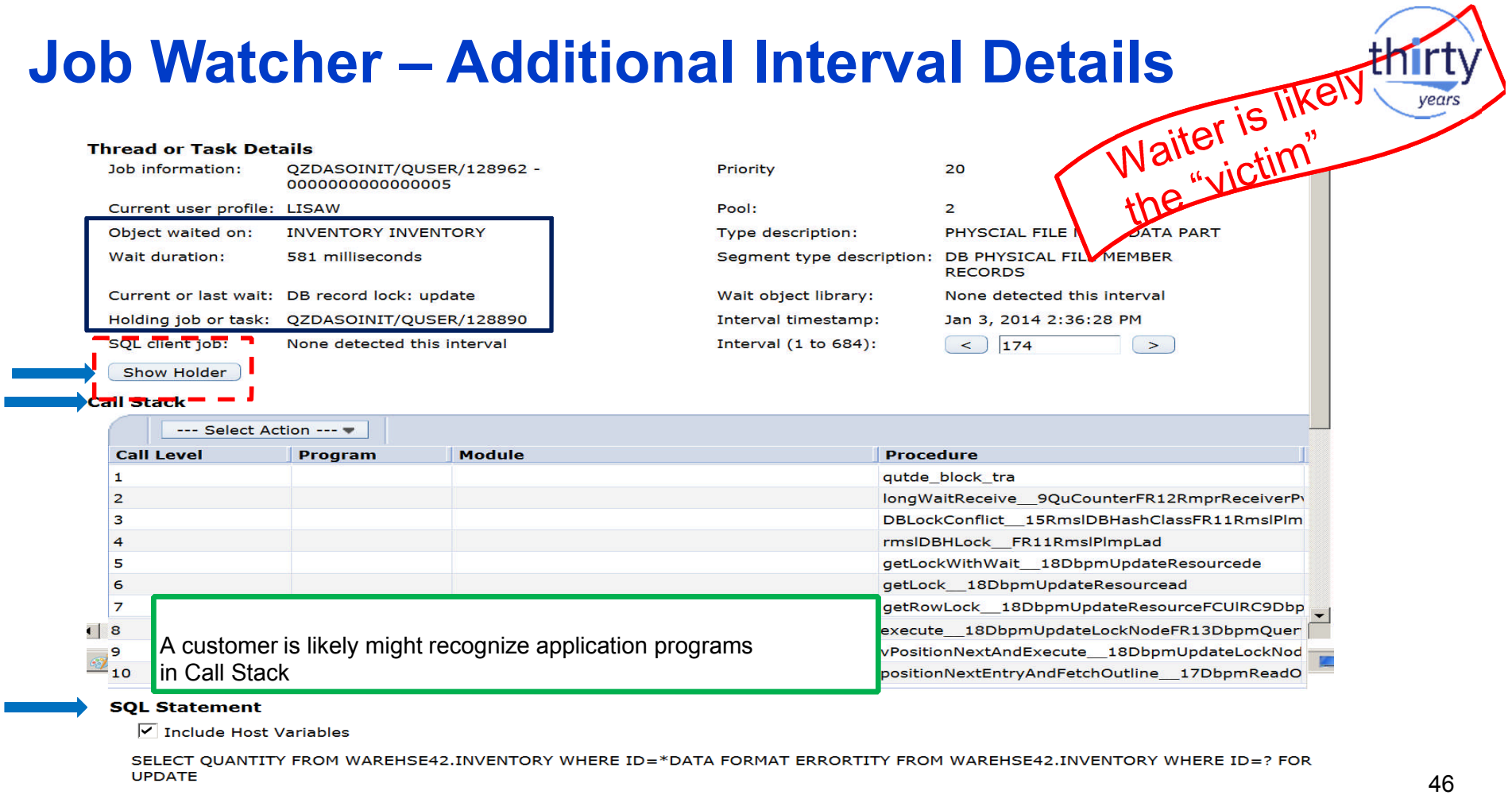
Notice similar perspectives available as Collection Services

Job Watcher – Waits Overview



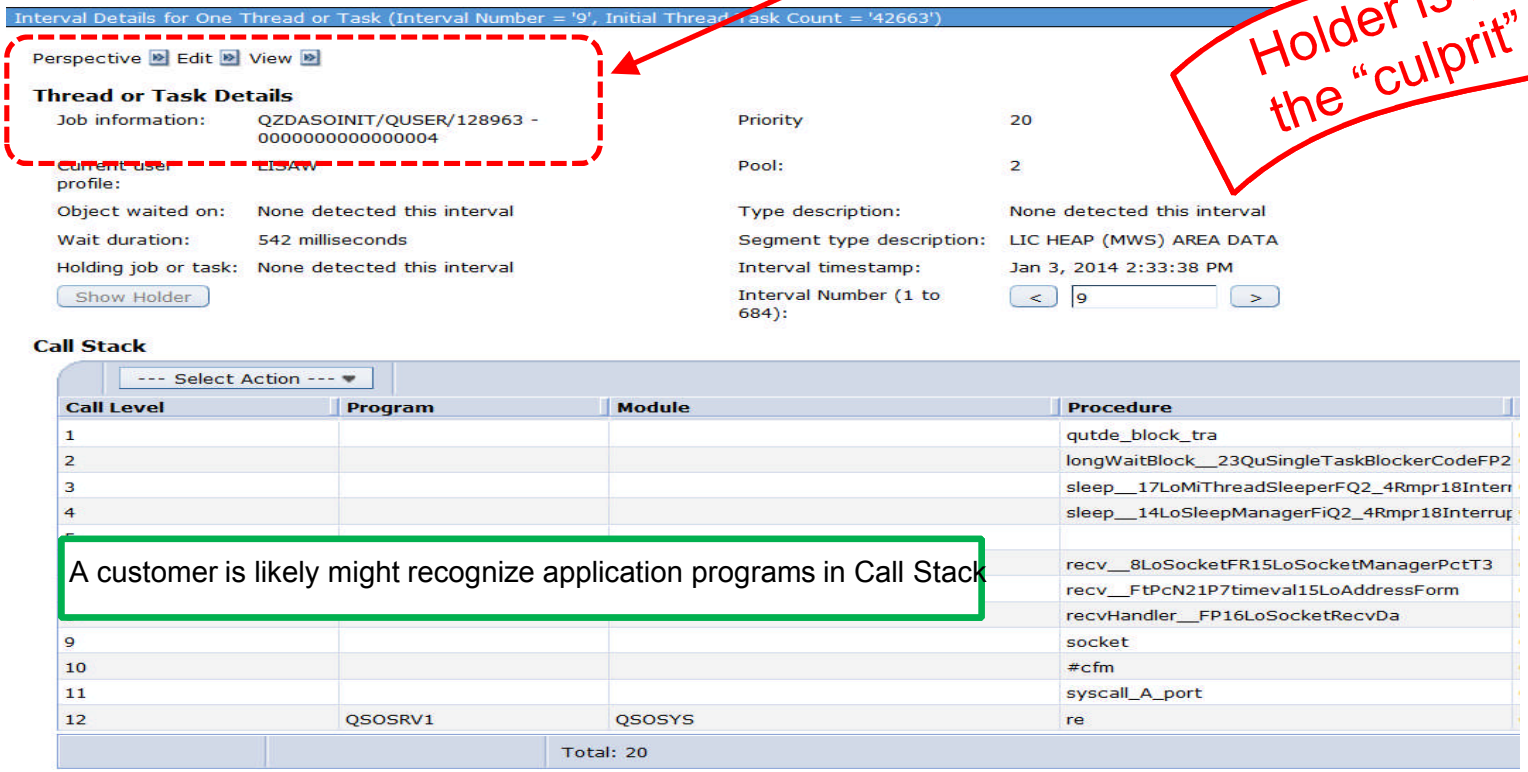
Notice same wait buckets, but more granular intervals and additional job drill down

Job Watcher – Additional Interval Details



Job Watcher – Show Holder

- When clicking the “Show Holder” button, the **holding** job or task info will be displayed:

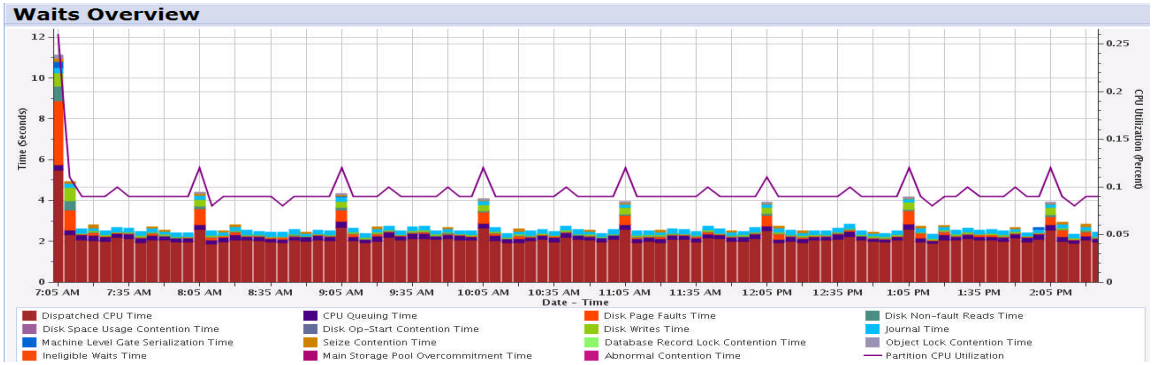


Recommendations

Wait Accounting - Recommendations: Be proactive!

- Use the rich IBM i wait accounting instrumentation found in
  - Collection Services & Job Watcher
  - Use PDI or iDoctor to view/analyze

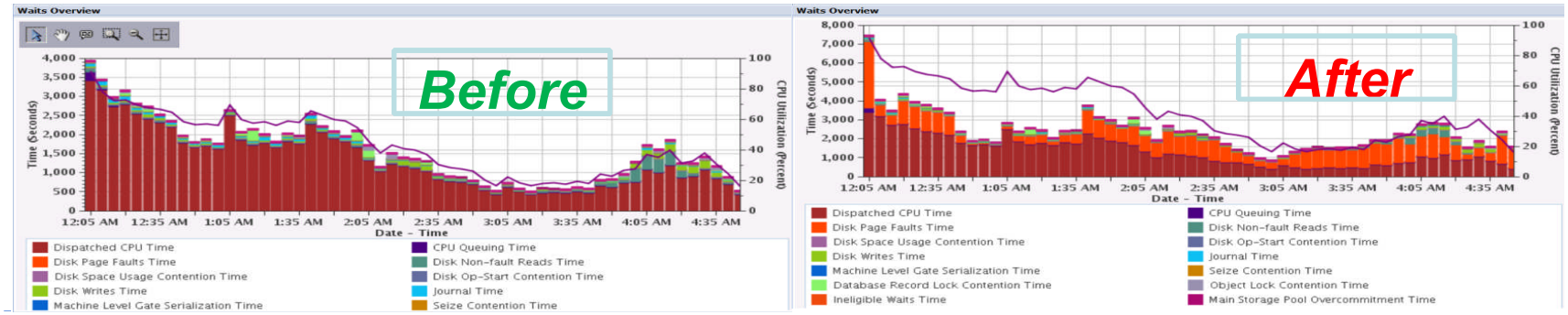
- ***Understand your partition's "run-wait" signature and normal patterns***



- Identify bottlenecks

Wait Accounting - Recommendations: Be proactive!

- Keep a **baseline**
  - Collection Services (Job Watcher data is also useful 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

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

References

IBM i Performance FAQ a MUST read!

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

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

IBM Power Systems Performance

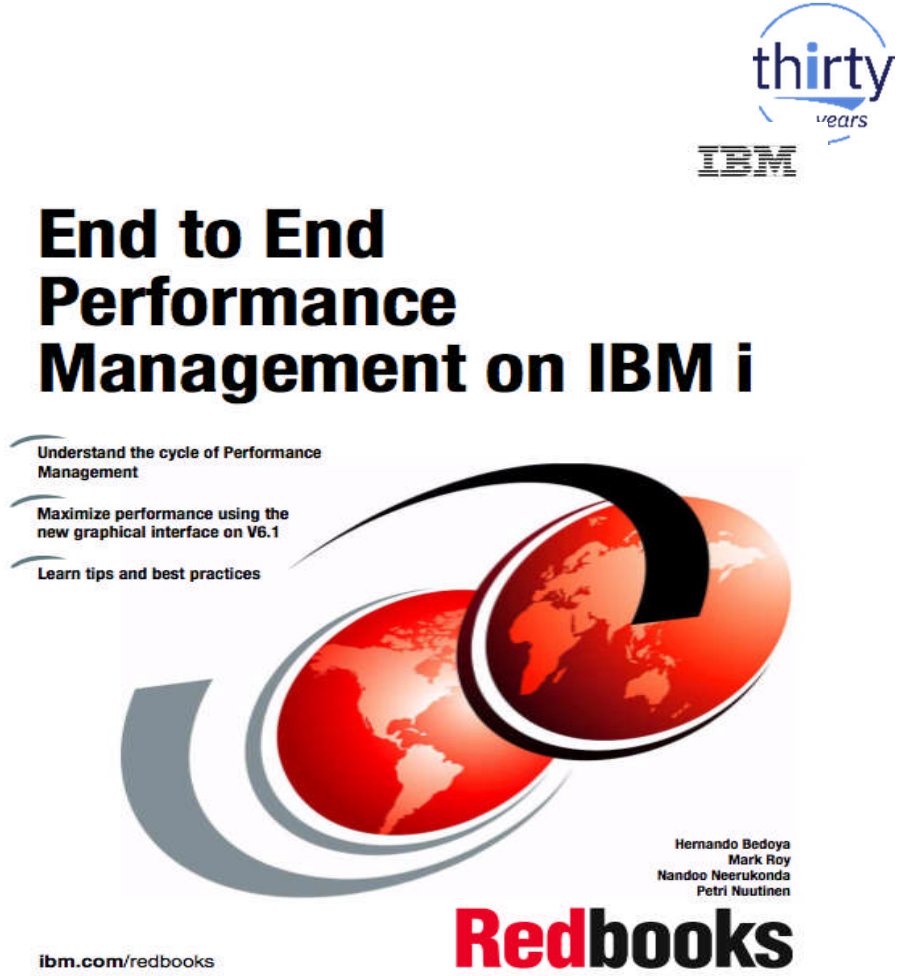
IBM i on Power - Performance FAQ

October 9, 2017

IBM i Web Sites with Performance Information

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

A Redbooks publication!



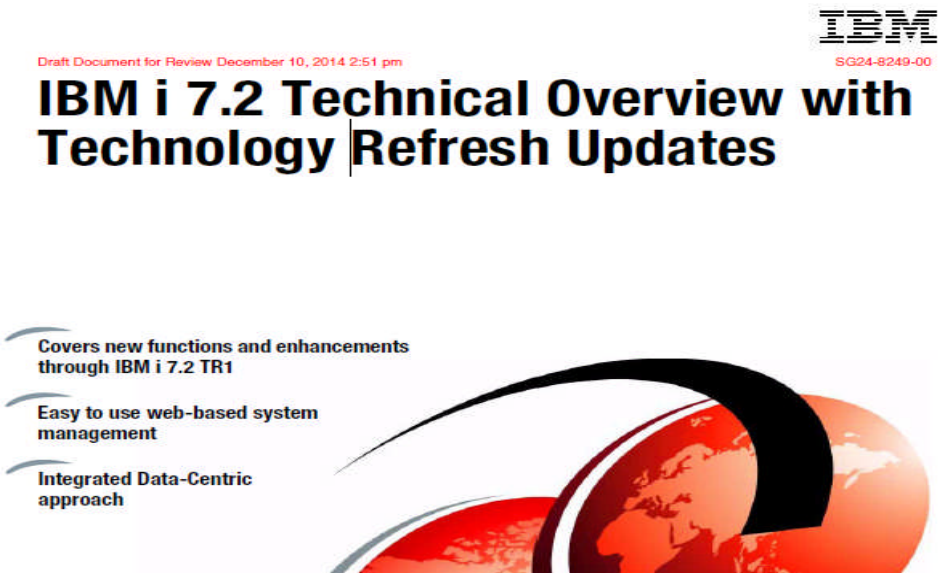
http://www.redbooks.ibm.com/redbooks/pdfs/sg247808.pdf

IBM i 7.2 Technology Refresh Updates

Covers the 7.2 content through Technology Refresh 1

Section 2.8 – Performance

Section 8.6.7 – Job level SQL stats in Collection Services



IBM i Performance and Optimization Services

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

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

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

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

www.ibm.com/systems/services/labservices

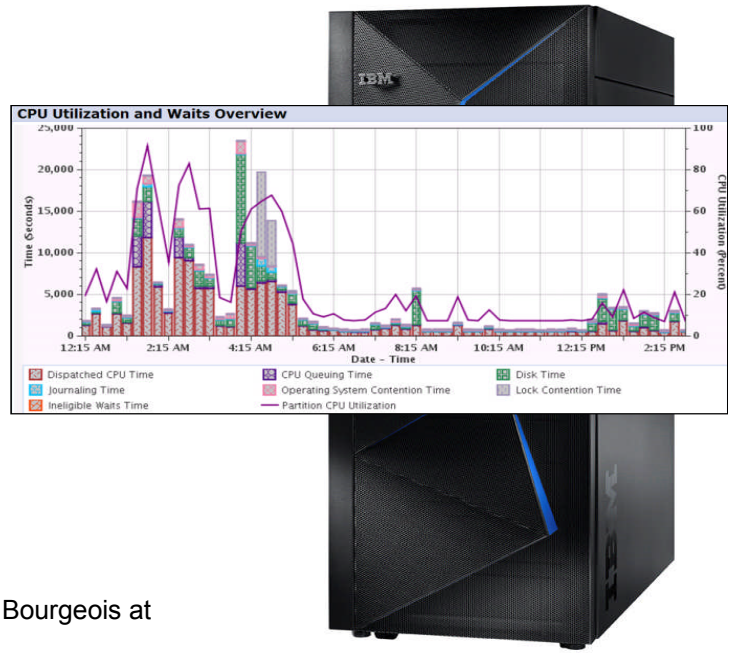
IBM i Performance Analysis Workshop

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

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

Overview:

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



Workshop details:

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

And finally…..

Thank you

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

ithankyou

www.ibm.com/power/i

End of Presentation material…..