IBM Power Systems with IBM i using Solid State Drives to boost your Oracle's JD Edwards EnterpriseOne performance

Diane Webster
IBM Oracle International Competency Center

January 2012

Table of contents

Change history

Version	Date	Author	Editing Description
1.0	January 2012	Diane Webster	Original

Abstract

Solid state drive (SSD) technology can provide superior I/O performance compared to traditional hard disk drives (HDDs). SSDs use flash memory to store data instead of rotating disks, resulting in no seek time or rotational delays when accessing data. Access speeds for flash memory can be up to ten times faster than access speeds for HDDs, allowing SSDs to deliver thousands of I/O Operations Per Second (IOPS) instead of the hundreds of IOPS typically seen for spinning hard disk drives.

Solid state drives can help optimize high I/O access applications to improve performance, throughput, response time and overall system responsiveness. This means that all JD Edwards EnterpriseOne activity, but particularly the Universal Batch Engine (UBE) jobs have the potential to see greatly improved performance with SSDs by breaking through the I/O constraints typically seen in these batch jobs.

This paper shows how solid state disk improved the performance of the JD Edwards EnterpriseOne Pre-Payroll process by 57 percent. It will also show a step by step process to identify areas in a JD Edwards EnterpriseOne environment that may benefit from solid state drives. Then using a process available in IBM i 7.1, it will show how data can be easily be moved to SSDs to achieve the best possible performance.

To learn more about SSDs visit the IBM Techdocs website at www.ibm.com/support/techdocs and search for SSD.

Introduction

On the IBM Power® Systems POWER7® processor-based servers using the IBM i operating system, there are a number of methods and tools available to help ensure that solid state disk drives can be leveraged for the best I/O performance. In environments with a mixture of HDD and SSD drives, the goal is to move the majority of random read requests to the SSDs while keeping a majority of the write requests spread across the HDDs.

This paper will take you through a process to help achieve optimal disk performance and identify the potential improvements that could be realized when using solid state drives. This process includes:

- Identifying subsystems and jobs where performance may be improved with solid state drives
- Moving data to the solid state drives by leveraging the TRCASPBAL and STRASPBAL commands
- Analyzing the impact of moving the data to solid state drives
- Determining what tables and indexes were moved to solid state drives

It is important to note that the performance improvement of solid state drives can be impacted by a wide variety of factors. These include but are not limited to:

- Current disk performance
- CPU and memory utilization
- Data placement on disk drives
- Database optimization and index usage

As a result, individual results may be different from those seen in the scenario described below. However, the process that follows can be used to identify areas in your environment to leverage solid state disk drives.

Scenario definition

The scenario used in this paper ran on an IBM i Solution Edition for JD Edwards server (Power® 740 version) with IBM i 7.1 and IBM WebSphere® Application Server 7.0. The partition used for the workload had six cores with 60 gigabytes of memory. The disk configuration included 24 disk arms (139.5 GB feature code 3677) and four mirrored solid state drives (177 GB feature code 1996) in the same auxiliary storage pool (ASP). These solid state drives were mounted on a PCIe adapter (feature code 2055). IBM also offers SSDs that can be placed in the system unit or in a disk drawer. SSDs which can be placed in a disk drawer (177 GB feature codes 1787 or 1794) can offer an advantage over 1996's because they leverage an adapter with write cache (380 MB of cache with feature code 5805 or 1.8 GB of cache with feature code 5913).

The application workload in the test scenario used JD Edwards EnterpriseOne 9.02 with tools release 8.98.4.4. The primary workload consisted of 800 interactive users running concurrently with two UBEs, R07200 (Pre-payroll) and R3483 (Master Planning Schedule - Multiple Plant). The interactive workload was generated with load generation software LoadRunner using the JD Edwards EnterpriseOne Day in the Life (DIL) test kit. The kit covers five major JD Edwards EnterpriseOne modules and uses 17 different applications. The applications used are listed in Table 1 below:

	Application	Description
1	P03B102	Apply Receipts
2	P0411	Supplier Ledger Inquiry
3	P051191	Daily Time Entry
4	P17500	Case Management Add
5	P31114	W.O. Completion
6	P3411	MRP Messages (WO Orders)
7	P3411	MRP Messages (OP Orders)
8	P3411	MRP Message (OT Orders)
9	P4113	Inventory Transfer
10	P42101	S.O. Entry – 10 line items
11	P42101	S.O. Update
12	P4310	P.O. Entry – 25 line items
13	P4312	P.O. Receipts
14	P4314	Voucher Match
15	P4915	Ship Confirm – Approval only
16	P4915	Ship Confirm – Confirm/Ship only
17	P4915	Ship Confirm – Confirm and Change entry

Table 1 - Applications included in the DIL test kit

The pre-payroll UBE processed 20,000 employees with daily timecard data. R3483 processed 40,000 inventory items.

The steps to execute the test scenario were:

- Run workload to establish baseline on the configuration
- Analyze system performance to determine the need for SSD drives
- Add solid state drives to configuration
- Run the workload a second time to collect disk usage statistics with the TRCASPBAL command

- Execute the STRASPBAL command to move “hot” data to solid state drives
- Run workload a third time to measure the impact of solid state drives

Analyze current system performance

A first step to understanding how solid state disk drives can improve performance is to analyze the current system performance. The primary areas to identify are the subsystems and jobs that experience high disk page fault wait times. High disk page faults are most commonly caused by random read requests from the application. On IBM i, the biggest performance benefits may be achieved by moving data with frequent random reads to solid state drives. To easily identify this, we use performance data collected by IBM i Collection Services. Before beginning, ensure that you have performance data collected over the time period that you want to analyze.

To identify areas with significant disk page fault wait times, use the IBM i Systems Director Navigator for i. This is accessed with the URL:

- http://<system_name>:2001/ibm/console, where <system_name> is the name of your IBM i system.

After you are logged in to Systems Director Navigator, use the following path: Open the IBM i Management section in the left hand navigator bar > Click on Performance > Click on Investigate Data in the right hand tab > Under Perspectives, click Collection Services > Click Waits >Click Waits by Subsystem (see Figure 1).

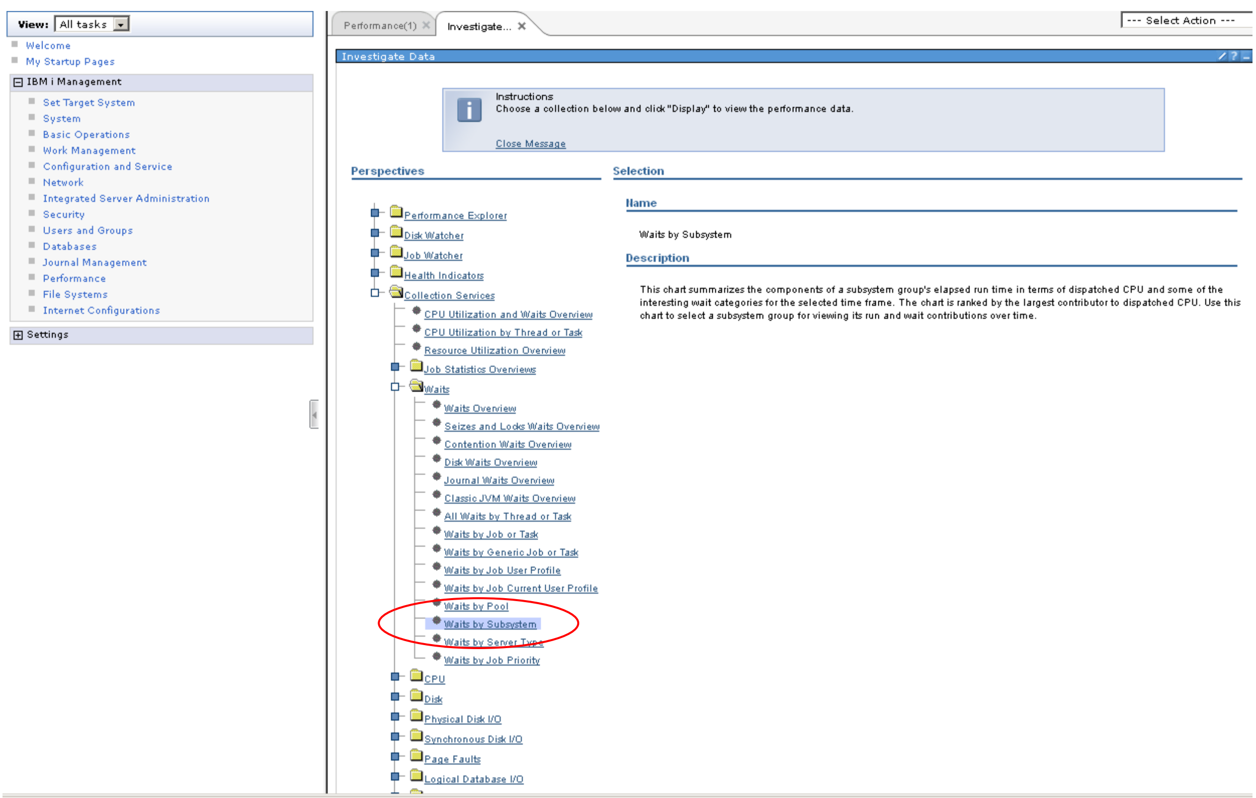


Figure 1 - Investigate Performance Data

Then scroll down to the Collection Section (see Figure 2).

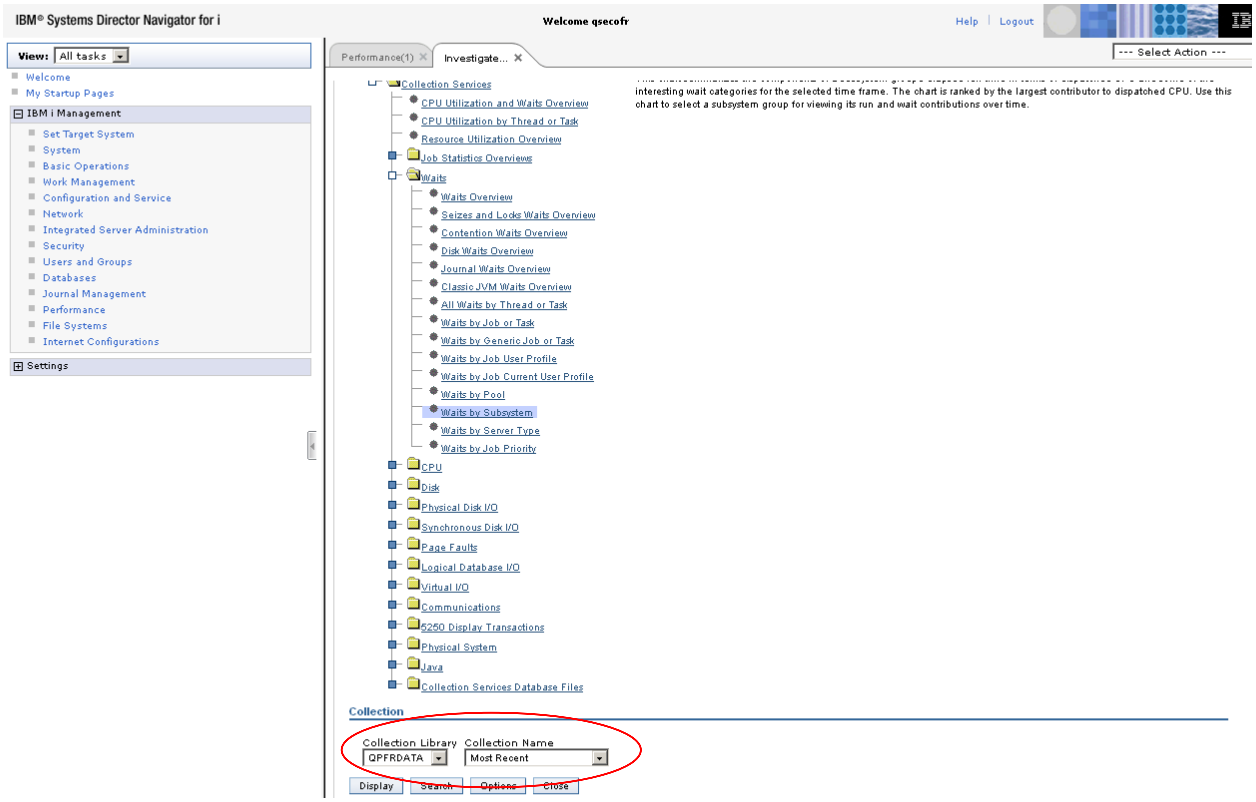


Figure 2 - Collection Services

Under the Collection Library and Collection Name, select the library and collection name to analyze. The default is the most recent performance data collection on the system, including the current active data collection.

Then click Display.

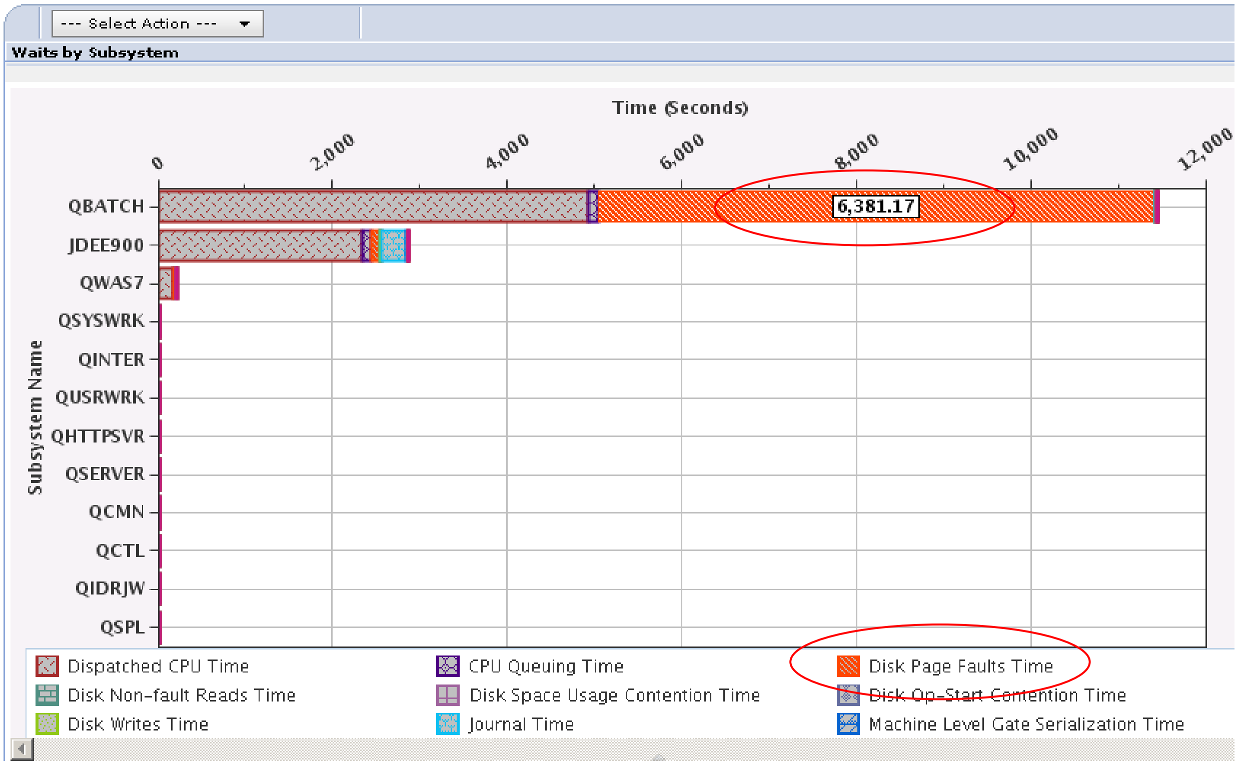


Figure 3 - Waits by Subsystem

In the graph that is displayed (see Figure 3), we see QBATCH subsystem has the highest amount of wait which consists of Dispatched CPU Time, and Disk Page Fault time. To display the value for Disk Page Fault time, click on the Disk Page Fault element of the graph. In this case, we see that the jobs in the QBATCH subsystem wait a total of 6381.17 seconds for disk page fault activity. We want to focus on this, because Solid State drives can specifically help us reduce this type of wait activity.

Click on the Select Action drop box and select Waits by Job or Task (see Figure 4).

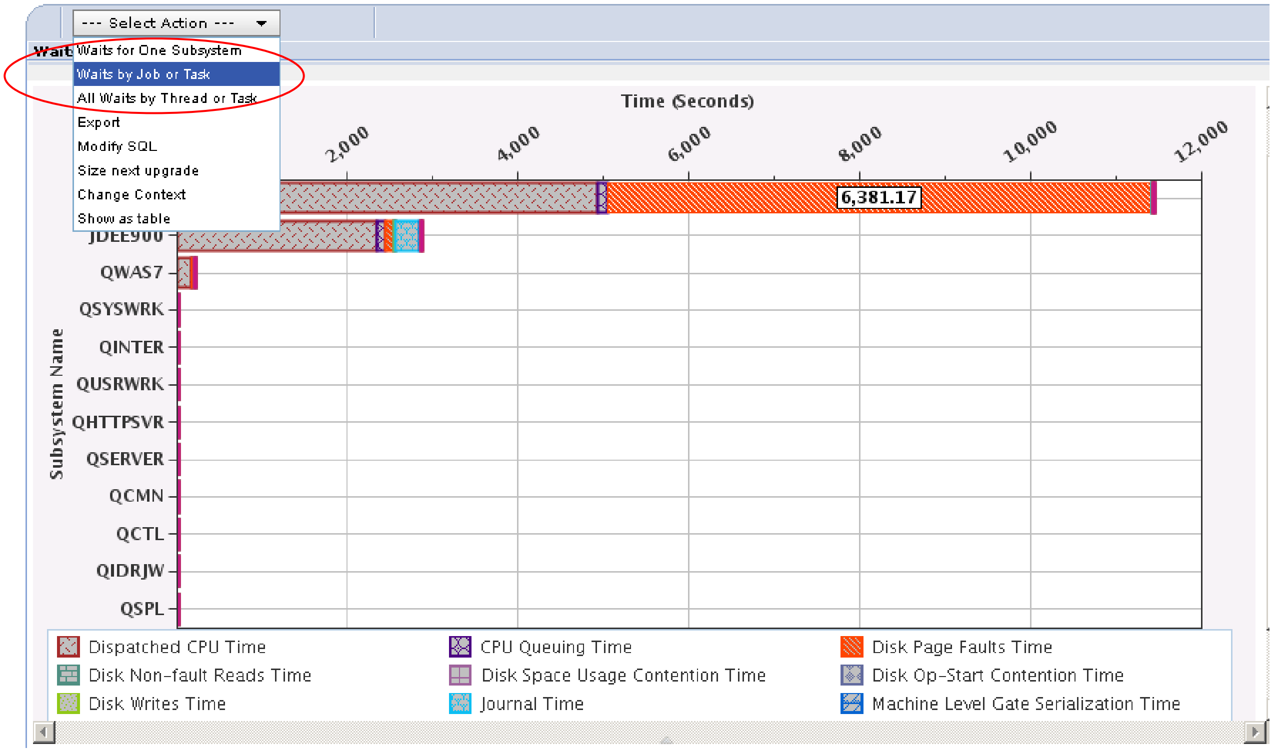


Figure 4 - Select Action

HINT: Make sure that you click on the bar for the subsystem before you select “Wait by Job or Task” from the drop down box. Otherwise you will see all jobs on the system.

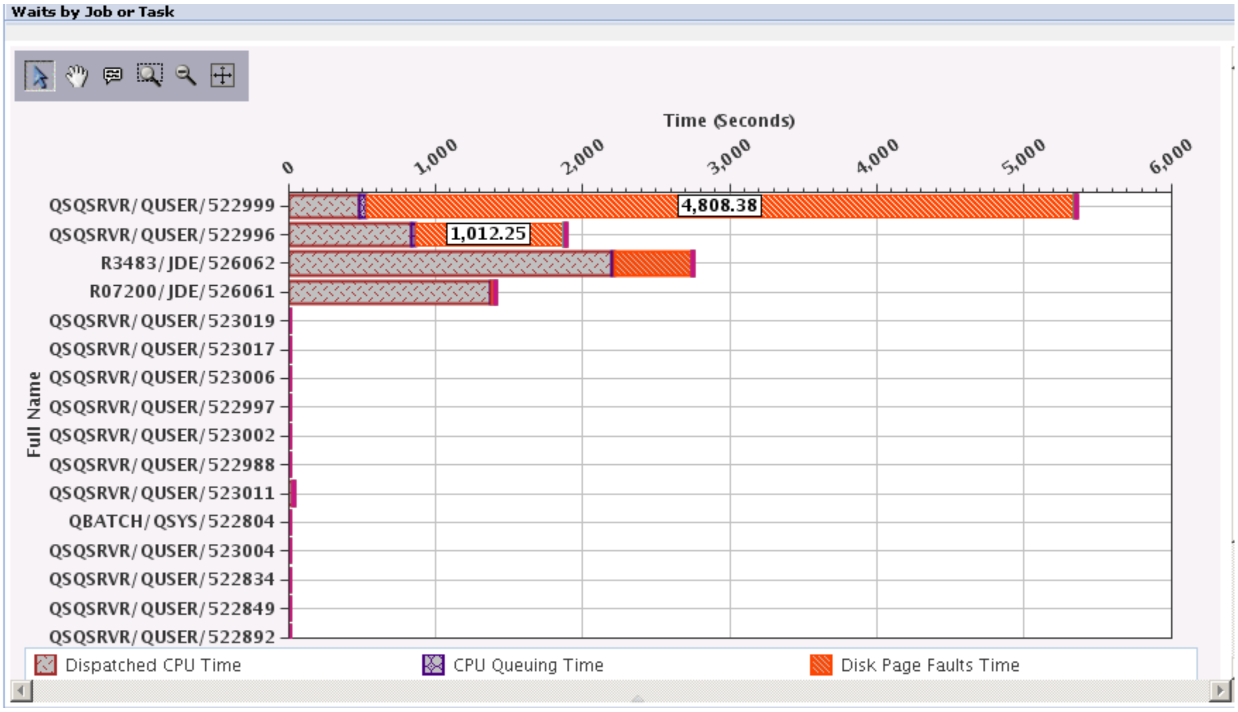


Figure 5 - Waits by Job or Task

We see that in the QBATCH subsystem a number of database connection jobs (QSQSRVR) show high wait times as well as two UBEs, R3483 and R07200. Job number 522999 has over 4800 seconds of disk page fault wait time (see Figure 5).

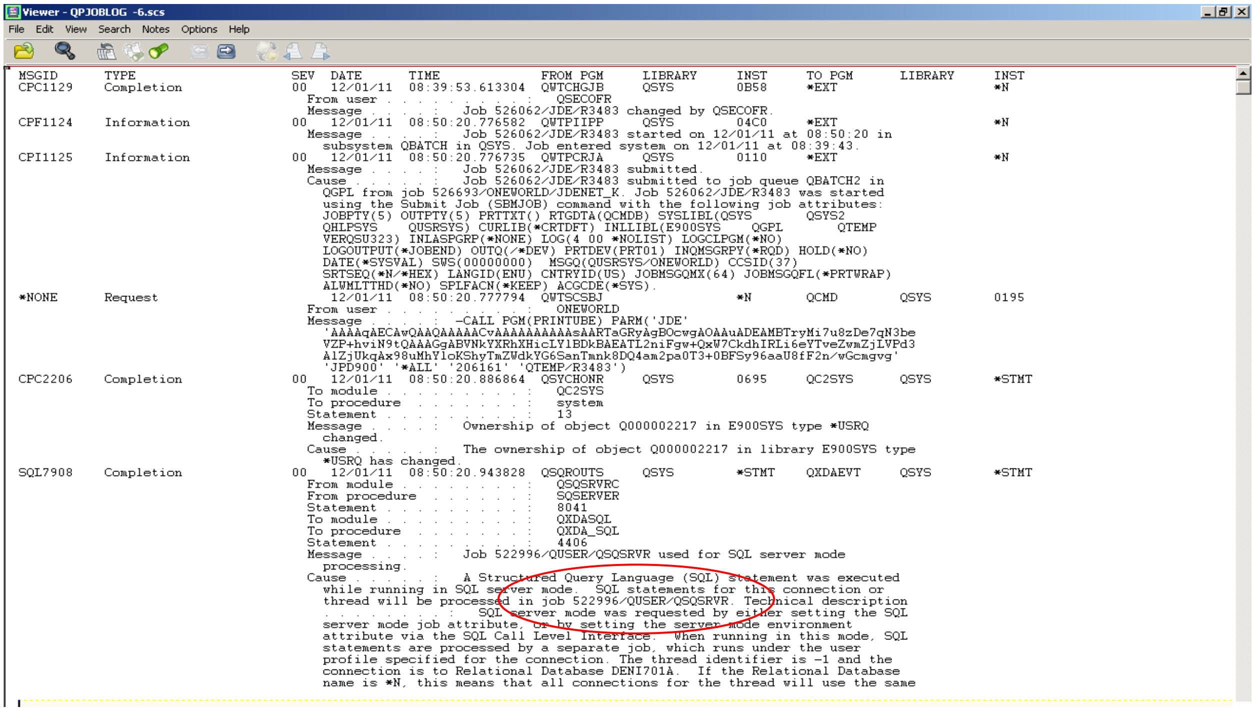
JD Edwards EnterpriseOne UBE and kernel jobs (JDENET_K) do not process their own database requests. When a job executes SQL, the job will connect to a QSQSRVR job which performs all database requests for a specific schema. UBEs and Kernel jobs will typically have connections to more than one QSQSRVR job. To identify the QSQSRVR jobs used by each UBE,or kernel job look at the job's job log. This can be done from either System Director Navigator, or from IBM i command line.

The command to use from the command line is DSPSPLF FILE(QPJOBLOG) JOB(<jobnumber/job user/jobname>).. For example:

- DSPSPLF FILE(QPJOBLOG) JOB(526061/JDE/R3483)
- DSPSPLF FILE(QPJOBLOG) JOB(526062/JDE/R07200

In this example, we will view the job logs from System Director Navigator.

Hint: Job logs can be displayed from System Director Navigator using the following path: Expand IBM i Management > Select Work Management > In the right hand tab, Click Output Queues > View the pop-up menu for QEZJOBLOG > Click Printer Output > Find the job log > View the pop-up menu and click open.

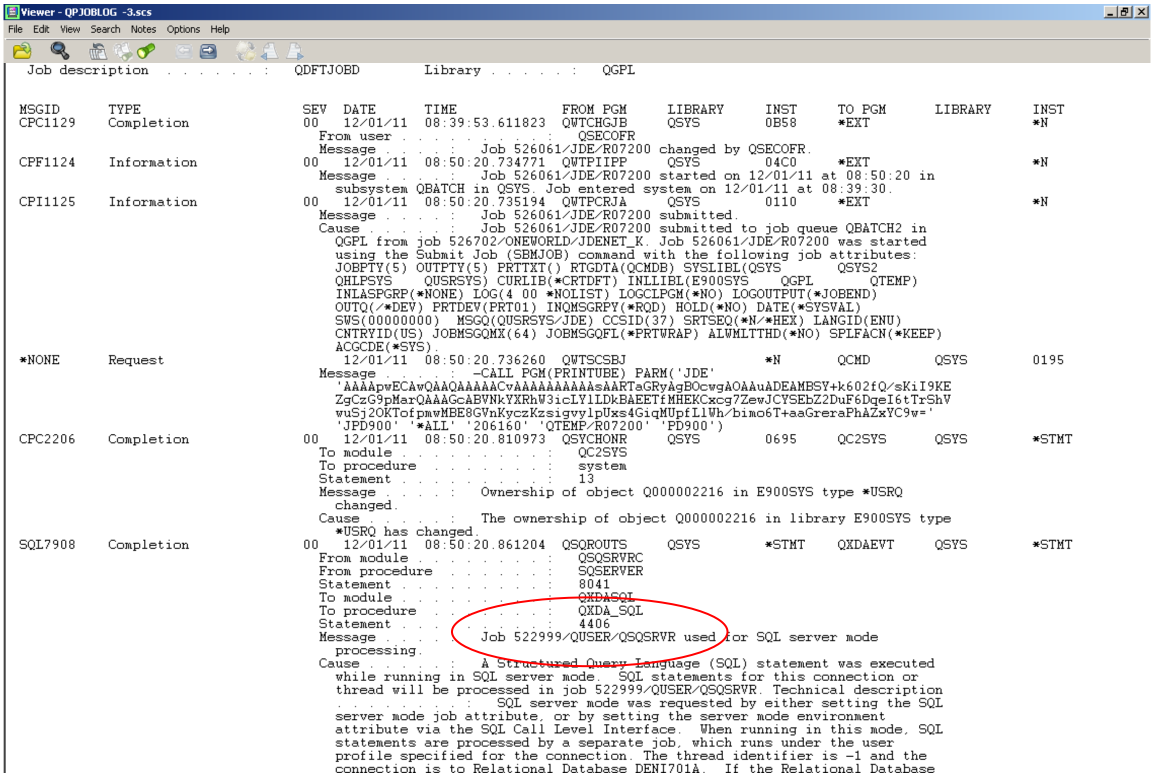


```
Viewer - QPJOBLOG -6.scs
File  Edit  View  Search  Notes  Options  Help

MSGID     TYPE                    SEV  DATE      TIME             FROM PGM    LIBRARY    INST    TO PGM     LIBRARY     INST
CPC1129   Completion              00   12/01/11  08:39:53.613304  QWTCHGJB    QSYS       0B58    *EXT                   *N
                                   From user . . . . . . . . . :   QSECOFR
                                   Message . . . . :   Job 526062/JDE/R3483 changed by QSECOFR.
CPF1124   Information             00   12/01/11  08:50:20.776582  QWTPIIPP    QSYS       04C0    *EXT                   *N
                                   Message . . . . :   Job 526062/JDE/R3483 started on 12/01/11 at 08:50:20 in
                                     subsystem QBATCH in QSYS. Job entered system on 12/01/11 at 08:39:43.
CPI1125   Information             00   12/01/11  08:50:20.776735  QWTPCRJA    QSYS       0110    *EXT                   *N
                                   Message . . . . :   Job 526062/JDE/R3483 submitted.
                                   Cause . . . . . :   Job 526062/JDE/R3483 submitted to job queue QBATCH2 in
                                     QGPL from job 526693/ONEWORLD/JDENET_K. Job 526062/JDE/R3483 was started
                                     using the Submit Job (SBMJOB) command with the following job attributes:
                                     JOBPTY(5) OUTPTY(5) PRTTXT() RTGDTA(QCMDB) SYSLIBL(QSYS      QSYS2
                                     QHLPSYS    QUSRSYS) CURLIB(*CRTDFT) INLLIBL(E900SYS    QGPL       QTEMP
                                     VERQSU323) INLASPGRP(*NONE) LOG(4 00 *NOLIST) LOGCLPGM(*NO)
                                     LOGOUTPUT(*JOBEND) OUTQ(/*DEV) PRTDEV(PRT01) INQMSGRPY(*RQD) HOLD(*NO)
                                     DATE(*SYSVAL) SWS(00000000)  MSGQ(QUSRSYS/ONEWORLD) CCSID(37)
                                     SRTSEQ(*N/*HEX) LANGID(ENU) CNTRYID(US) JOBMSGQMX(64) JOBMSGQFL(*PRTWRAP)
                                     ALWMLTTHD(*NO) SPLFACN(*KEEP) ACGCDE(*SYS).
*NONE     Request                      12/01/11  08:50:20.777794  QWTSCSBJ               *N      QCMD       QSYS        0195
                                   From user . . . . . . . . . :   ONEWORLD
                                   Message . . . . :  -CALL PGM(PRINTUBE) PARM('JDE'
                                     'AAAAqAECAwQAAQAAAAACvAAAAAAAAAAsAARTaGRyAgBOcwgAOAAuADEAMBTryMi7u8zDe7qN3be
                                     VZP+hviN9tQAAAGgABVNkYXRhXHicLYlBDkBAEATL2niFgw+QxW7CkdhIRLi6eYTveZwmZjLVPd3
                                     AlZjUkqAx98uMhYloKShyTmZWdkYG6SanTmnk8DQ4am2pa0T3+0BFSy96aaU8fF2n/wGcmgvg'
                                     'JPD900' '*ALL' '206161' 'QTEMP/R3483')
CPC2206   Completion              00   12/01/11  08:50:20.886864  QSYCHONR    QSYS       0695    QC2SYS     QSYS        *STMT
                                   To module . . . . . . . . . :   QC2SYS
                                   To procedure  . . . . . . . :   system
                                   Statement . . . . . . . . . :   13
                                   Message . . . . :   Ownership of object Q000002217 in E900SYS type *USRQ
                                     changed.
                                   Cause . . . . . :   The ownership of object Q000002217 in library E900SYS type
                                     *USRQ has changed.
SQL7908   Completion              00   12/01/11  08:50:20.943828  QSQROUTS    QSYS       *STMT   QXDAEVT    QSYS        *STMT
                                   From module . . . . . . . . :   QSQSRVRC
                                   From procedure  . . . . . . :   SQSERVER
                                   Statement . . . . . . . . . :   8041
                                   To module . . . . . . . . . :   QXDASQL
                                   To procedure  . . . . . . . :   QXDA_SQL
                                   Statement . . . . . . . . . :   4406
                                   Message . . . . :   Job 522996/QUSER/QSQSRVR used for SQL server mode
                                     processing.
                                   Cause . . . . . :   A Structured Query Language (SQL) statement was executed
                                     while running in SQL server mode.  SQL statements for this connection or
                                     thread will be processed in job 522996/QUSER/QSQSRVR. Technical description
                                     . . . . . . . . :   SQL server mode was requested by either setting the SQL
                                     server mode job attribute, or by setting the server mode environment
                                     attribute via the SQL Call Level Interface.  When running in this mode, SQL
                                     statements are processed by a separate job, which runs under the user
                                     profile specified for the connection. The thread identifier is -1 and the
                                     connection is to Relational Database DENI701A.  If the Relational Database
                                     name is *N, this means that all connections for the thread will use the same
```

Figure 6 - Job Log for R3483

In Figure 6, we see the job log for R3483. Each database connection made by this UBE has a corresponding SQL7908 message. This message shows the job number for the database connection. In this job log, the job number 522996 is connected to R3483. With this information, we know that for R3483, approximately 1000 seconds of its runtime was waiting for disk reads (see Figure 5).



```
Viewer - QPJOBLOG -3.scs
File  Edit  View  Search  Notes  Options  Help
 Job description  . . . . . . . :   QDFTJOBD        Library . . . . . :   QGPL

MSGID      TYPE                     SEV  DATE      TIME             FROM PGM     LIBRARY     INST     TO PGM      LIBRARY     INST
CPC1129    Completion               00   12/01/11  08:39:53.611823  QWTCHGJB     QSYS        0B58     *EXT                    *N
                                      From user . . . . . . . . . :   QSECOFR
                                      Message . . . . :   Job 526061/JDE/R07200 changed by QSECOFR.
CPF1124    Information              00   12/01/11  08:50:20.734771  QWTPIIPP     QSYS        04C0     *EXT                    *N
                                      Message . . . . :   Job 526061/JDE/R07200 started on 12/01/11 at 08:50:20 in
                                        subsystem QBATCH in QSYS. Job entered system on 12/01/11 at 08:39:30.
CPI1125    Information              00   12/01/11  08:50:20.735194  QWTPCRJA     QSYS        0110     *EXT                    *N
                                      Message . . . . :   Job 526061/JDE/R07200 submitted.
                                      Cause . . . . . :   Job 526061/JDE/R07200 submitted to job queue QBATCH2 in
                                        QGPL from job 526702/ONEWORLD/JDENET_K. Job 526061/JDE/R07200 was started
                                        using the Submit Job (SBMJOB) command with the following job attributes:
                                        JOBPTY(5) OUTPTY(5) PRTTXT() RTGDTA(QCMDB) SYSLIBL(QSYS       QSYS2
                                        QHLPSYS    QUSRSYS) CURLIB(*CRTDFT) INLLIBL(E900SYS    QGPL       QTEMP)
                                        INLASPGRP(*NONE) LOG(4 00 *NOLIST) LOGCLPGM(*NO) LOGOUTPUT(*JOBEND)
                                        OUTQ(/*DEV) PRTDEV(PRT01) INQMSGRPY(*RQD) HOLD(*NO) DATE(*SYSVAL)
                                        SWS(00000000)  MSGQ(QUSRSYS/JDE) CCSID(37) SRTSEQ(*N/*HEX) LANGID(ENU)
                                        CNTRYID(US) JOBMSGQMX(64) JOBMSGQFL(*PRTWRAP) ALWMLTTHD(*NO) SPLFACN(*KEEP)
                                        ACGCDE(*SYS).
*NONE      Request                       12/01/11  08:50:20.736260  QWTSCSBJ                 *N       QCMD        QSYS        0195
                                      Message . . . . :  -CALL PGM(PRINTUBE) PARM('JDE'
                                        'AAAApwECAwQAAQAAAAACvAAAAAAAAAAsAARTaGRyAgBOcwgAOAAuADEAMBSY+k602fQ/sKiI9KE
                                        ZgCzG9pMarQAAAGcABVNkYXRhW3icLYlLDkBAEETfMHEKCxcg7ZewJCYSEbZ2DuF6DqeI6tTrShV
                                        wuSj2OKTofpmwMBE8GVnKyczKzsigvylpUxs4GiqMUpfLlWh/bimo6T+aaGreraPhAZxYC9w='
                                        'JPD900' '*ALL' '206160' 'QTEMP/R07200' 'PD900')
CPC2206    Completion               00   12/01/11  08:50:20.810973  QSYCHONR     QSYS        0695     QC2SYS      QSYS        *STMT
                                      To module . . . . . . . . . :   QC2SYS
                                      To procedure  . . . . . . . :   system
                                      Statement . . . . . . . . . :   13
                                      Message . . . . :   Ownership of object Q000002216 in E900SYS type *USRQ
                                        changed.
                                      Cause . . . . . :   The ownership of object Q000002216 in library E900SYS type
                                        *USRQ has changed.
SQL7908    Completion               00   12/01/11  08:50:20.861204  QSQROUTS     QSYS        *STMT    QXDAEVT     QSYS        *STMT
                                      From module . . . . . . . . :   QSQSRVRC
                                      From procedure  . . . . . . :   SQSERVER
                                      Statement . . . . . . . . . :   8041
                                      To module . . . . . . . . . :   QXDASQL
                                      To procedure  . . . . . . . :   QXDA_SQL
                                      Statement . . . . . . . . . :   4406
                                      Message . . . . :   Job 522999/QUSER/QSQSRVR used for SQL server mode
                                        processing.
                                      Cause . . . . . :   A Structured Query Language (SQL) statement was executed
                                        while running in SQL server mode.  SQL statements for this connection or
                                        thread will be processed in job 522999/QUSER/QSQSRVR. Technical description
                                        . . . . . . . . :   SQL server mode was requested by either setting the SQL
                                        server mode job attribute, or by setting the server mode environment
                                        attribute via the SQL Call Level Interface.  When running in this mode, SQL
                                        statements are processed by a separate job, which runs under the user
                                        profile specified for the connection. The thread identifier is -1 and the
                                        connection is to Relational Database DENI701A.  If the Relational Database
```

Figure 7 - Job Log for R07200

In Figure 7, we see the job log for R07200. In this job log we see, that job number 522299 is connected to R07200. This is the database connection that had over 4800 seconds of Disk Page Fault wait time (see Figure 5).

We have identified that high disk page fault waits are occurring on this system. We also know which UBEs the wait is associated with. The next step in the process is to trace the activity with the TRCASPBAL command when those UBEs are running. Then we will use STRASPBAL to have the system move data to solid state drives.

TRCASPBAL

The Trace ASP Balance (TRCASPBAL) command gathers ASP (auxiliary storage pool) disk usage statistics. The trace monitors the frequency that data is accessed on disk units within a specified ASP. As a result, the 'high' use data and the 'low' use data is identified on each disk.

To start the trace, use the TRCASPBAL command:

TRCASPBAL SET(*ON) ASP(<Aspnum>) TIMLMT(*NOMAX)

<ASPNUM> is the ASP where your data resides and should also be the ASP where the solid state drives are located. You can either specify time limit (TIMLMT) or set TIMLMT to *NOMAX. The time limit parameter allows you to control how long the trace will run. If you specify *NOMAX for the time limit, you need to manually stop the trace using TRCASPBAL SET(*OFF) ASP(<aspnum>).

Note: It is important to remember that you want to run the trace during the time period when any jobs are running that have been identified with high disk page fault waits.

After the desired time, turn the trace off with the command:

TRCASPBAL SET(*OFF) ASP(<aspnum>)

Where <aspnum> is the ASP number specified when the trace was started.

Note: If you specified a time limit, it is not required to run the TRCASPBAL (*OFF), if the time limit has already passed.

After the trace has stopped, verify that the trace is no longer running, by using the CHKASPBAL command (see Figure 8).

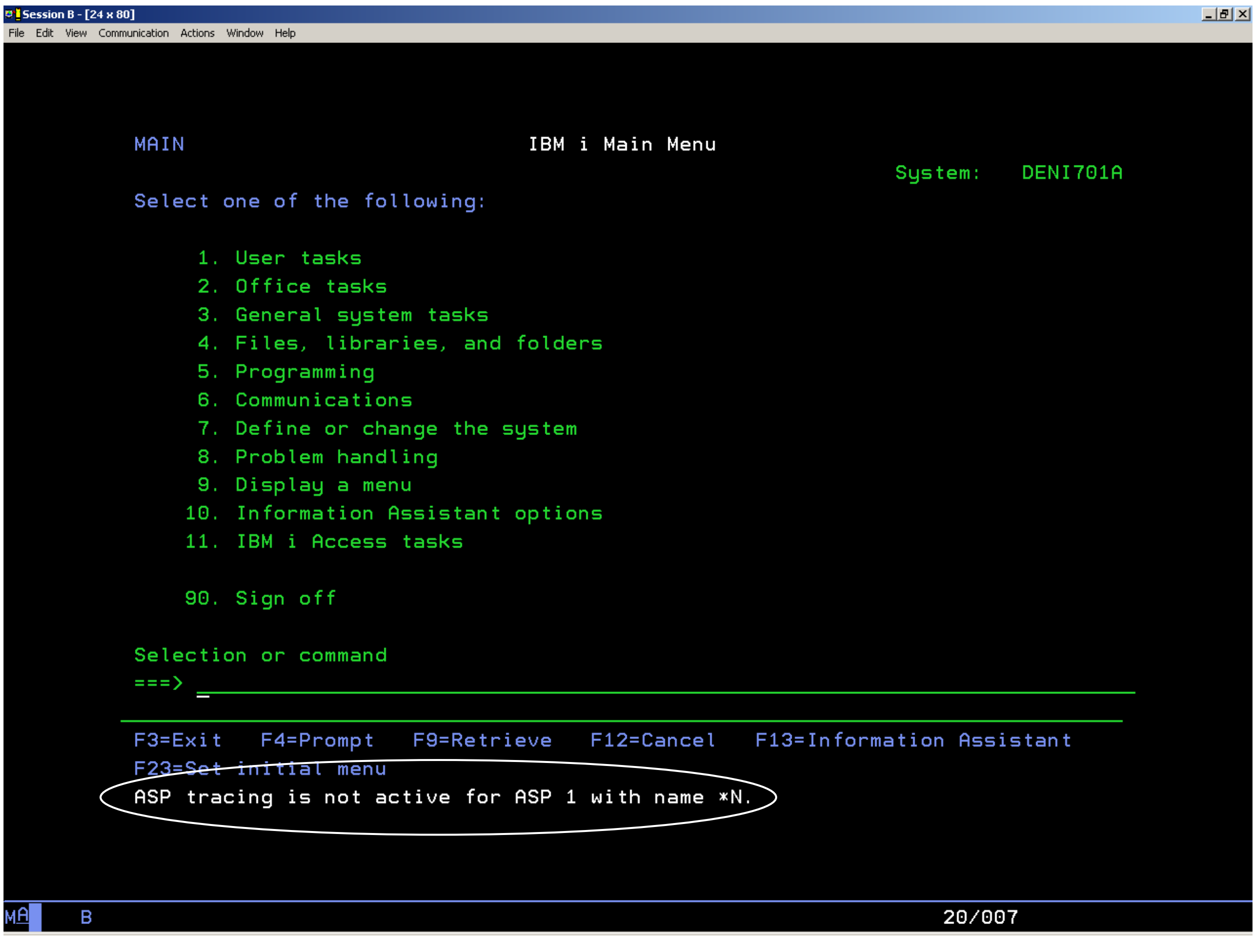
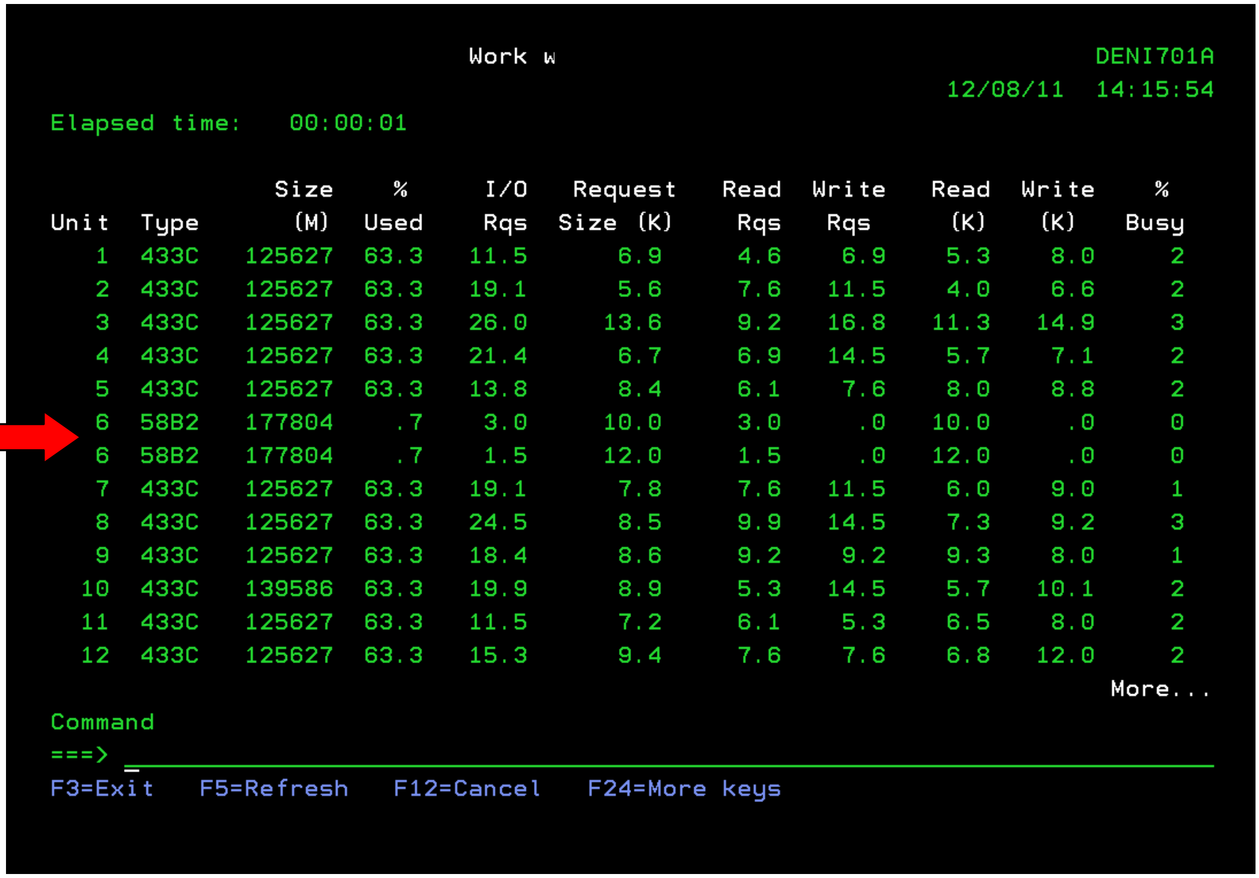


Figure 8 - ASP tracing not active

STRASPBAL

To move data to the solid state drives, use the STRASPBAL command with TYPE(*HSM). Using the trace data that was collected, this command automatically determines the data that is best suited to move to solid state drives. It also removes any data from the solid state drives that has not been identified as ”hot” data.

Prior to running STRASPBAL, use the WRKDSKSTS command to see the distribution of data on the SSD and HDD drives. In Figure 9, we see the solid state drives (Type 58B2) are less than 1% full.


Work w
DENI701A
12/08/11 14:15:54
Elapsed time: 00:00:01

Unit	Type	Size (M)	% Used	I/O Rqs	Request Size (K)	Read Rqs	Write Rqs	Read (K)	Write (K)	% Busy
1	433C	125627	63.3	11.5	6.9	4.6	6.9	5.3	8.0	2
2	433C	125627	63.3	19.1	5.6	7.6	11.5	4.0	6.6	2
3	433C	125627	63.3	26.0	13.6	9.2	16.8	11.3	14.9	3
4	433C	125627	63.3	21.4	6.7	6.9	14.5	5.7	7.1	2
5	433C	125627	63.3	13.8	8.4	6.1	7.6	8.0	8.8	2
6	58B2	177804	.7	3.0	10.0	3.0	.0	10.0	.0	0
6	58B2	177804	.7	1.5	12.0	1.5	.0	12.0	.0	0
7	433C	125627	63.3	19.1	7.8	7.6	11.5	6.0	9.0	1
8	433C	125627	63.3	24.5	8.5	9.9	14.5	7.3	9.2	3
9	433C	125627	63.3	18.4	8.6	9.2	9.2	9.3	8.0	1
10	433C	139586	63.3	19.9	8.9	5.3	14.5	5.7	10.1	2
11	433C	125627	63.3	11.5	7.2	6.1	5.3	6.5	8.0	2
12	433C	125627	63.3	15.3	9.4	7.6	7.6	6.8	12.0	2

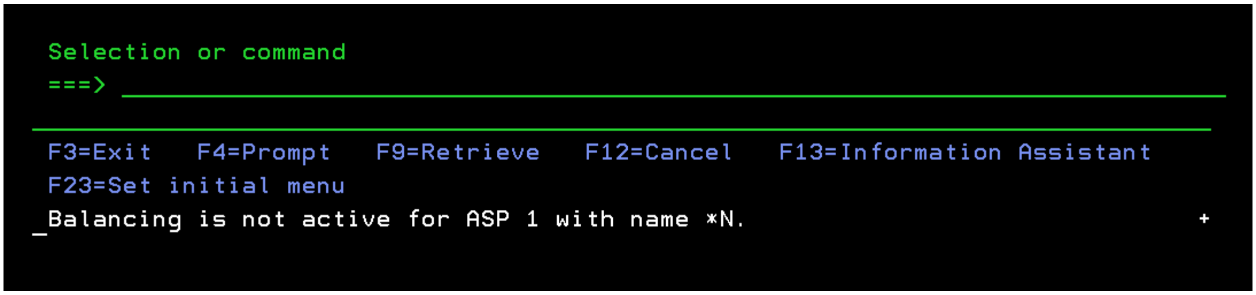
More...
Command
===>
F3=Exit F5=Refresh F12=Cancel F24=More keys

Figure 9 - WRKDSKSTS prior to running TRCASPBAL

After the trace ends there is asynchronous activity which must complete before data can be moved. Wait a few minutes, and then use the STRASPBAL command with TYPE (*HSM):

STRASPBAL TYPE(*HSM) ASP(<ASPNUM>) TIMLMT(*NOMAX) SUBTYPE(*CALC) where ASPNUM is the auxiliary storage pool used in the TRCASPBAL command.

The STRASPBAL command can run for several minutes. When the command completes, asynchronous tasks move the data. Data is migrated in two phases. Cold data is removed from the SSDs first, and then hot data is moved to the SSDs. You can use the CHKASPBAL command to check the status of the balancing. When the balancing completes, CHKASPBAL shows CPI18A6 indicating that balancing is not active (see Figure 10).



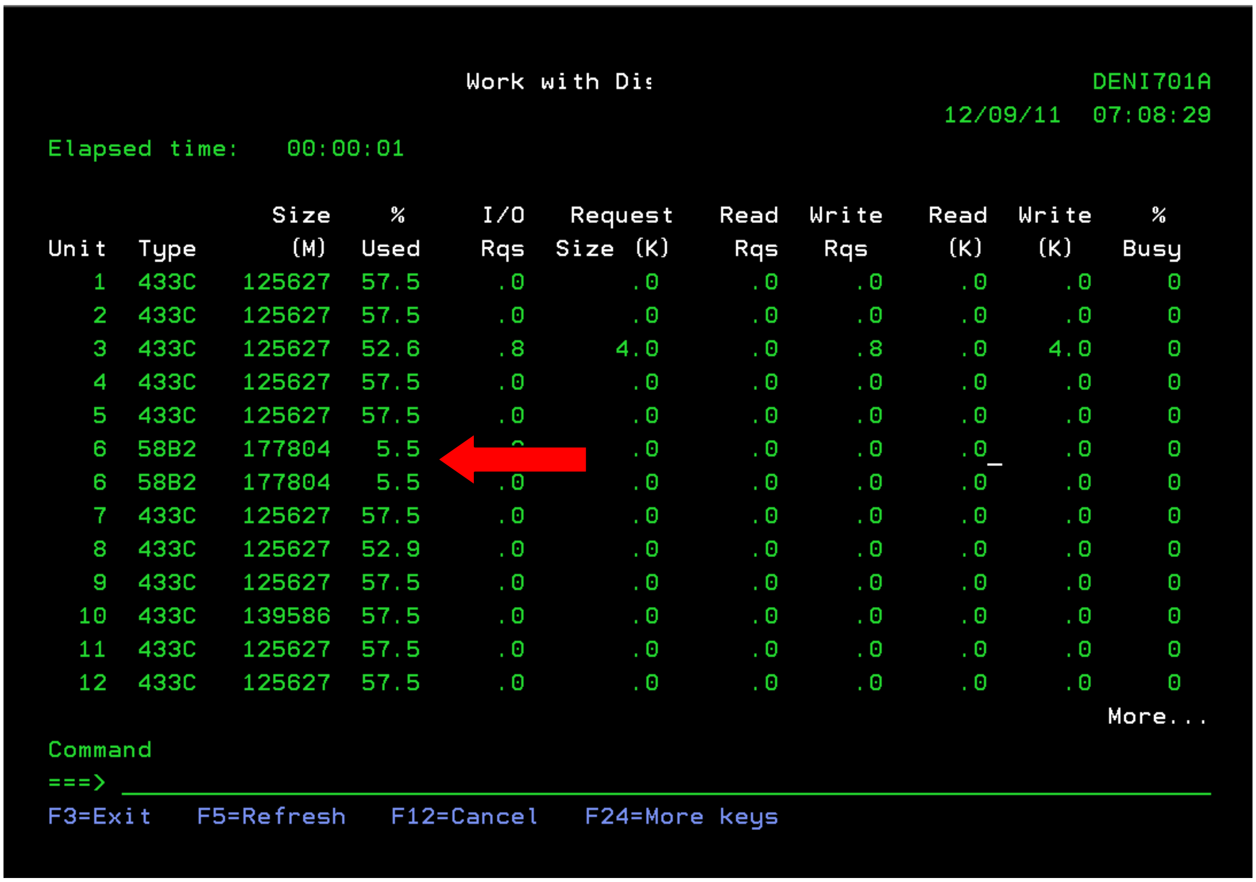
Selection or command
===> ____

F3=Exit F4=Prompt F9=Retrieve F12=Cancel F13=Information Assistant
F23=Set initial menu
_Balancing is not active for ASP 1 with name *N. +

Figure 10 - CPI18A6 Balancing is not active

Note: When balancing has completed the trace data will be deleted.

The WRKDSKSTS command shows an increase in the amount of data that is on the solid state drives (see Figure 11).



Work with Di: DENI701A
12/09/11 07:08:29
Elapsed time: 00:00:01

Unit	Type	Size (M)	% Used	I/O Rqs	Request Size (K)	Read Rqs	Write Rqs	Read (K)	Write (K)	% Busy
1	433C	125627	57.5	.0	.0	.0	.0	.0	.0	0
2	433C	125627	57.5	.0	.0	.0	.0	.0	.0	0
3	433C	125627	52.6	.8	4.0	.0	.8	.0	4.0	0
4	433C	125627	57.5	.0	.0	.0	.0	.0	.0	0
5	433C	125627	57.5	.0	.0	.0	.0	.0	.0	0
6	58B2	177804	5.5	[illegible]	.0	.0	.0	.0	.0	0
6	58B2	177804	5.5	.0	.0	.0	.0	.0	.0	0
7	433C	125627	57.5	.0	.0	.0	.0	.0	.0	0
8	433C	125627	52.9	.0	.0	.0	.0	.0	.0	0
9	433C	125627	57.5	.0	.0	.0	.0	.0	.0	0
10	433C	139586	57.5	.0	.0	.0	.0	.0	.0	0
11	433C	125627	57.5	.0	.0	.0	.0	.0	.0	0
12	433C	125627	57.5	.0	.0	.0	.0	.0	.0	0

More...
Command
===> ____
F3=Exit F5=Refresh F12=Cancel F24=More keys

Figure 11 - WRKDSKSTS after STRASPBAL

Reviewing the Results

The first cycle of tracing the workload and balancing the disk data is finished. To analyze the results, collect performance data during the time period when the jobs identified run again. In the test scenario the runtime for R07200 dropped from 156 minutes to 66 minutes, and the runtime for R3482 was reduced from 76 minutes to 70 minutes.

To review the changes in the disk page fault wait times for R07200 and R3483, we compare the original graphs with graphs generated after data was moved to solid state drives.

Comparing the two Wait by Subsystem graphs we see the disk page fault wait time is reduced to 1859 seconds (see Figure 12). The original page fault wait time as seen in Figure 13 is 6381 seconds.

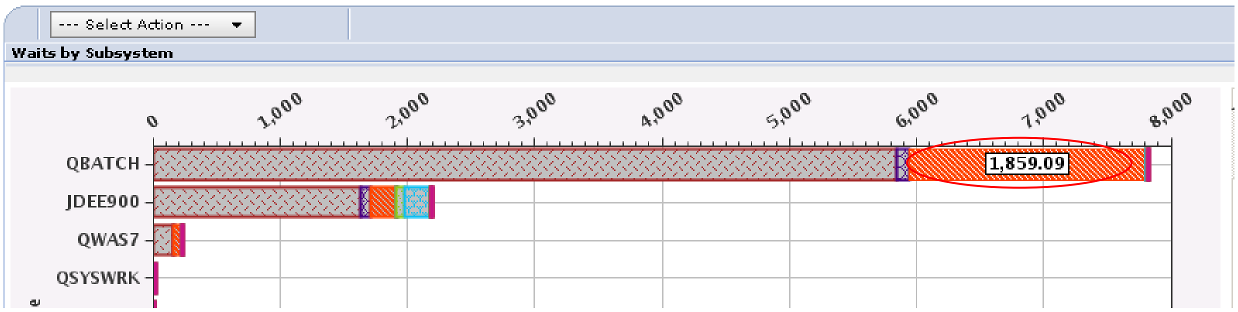


Figure 12 - Waits by subsystem after TRCASPBAL

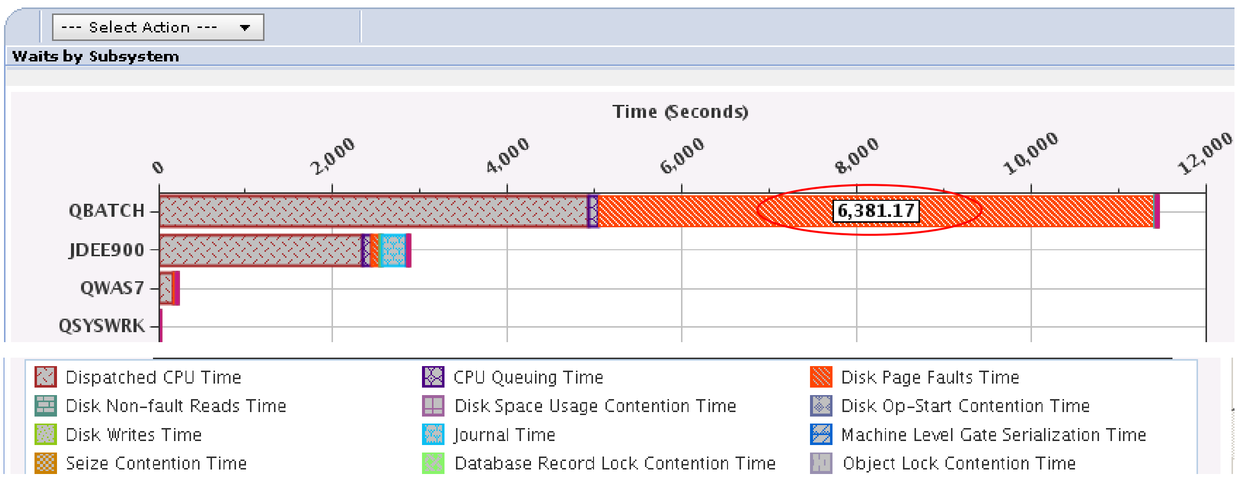


Figure 13 - Original wait times by subsystem

If we once again drill into the waits by job details, we see that QSQSRVR job for R07200 (job number 523017) had 1045 seconds of disk read wait time (see Figure 14) as compared to the original 4808 seconds (see Figure 15). The database job for R3483 had a reduction of disk page fault wait time from 1012 seconds down to 432 seconds.

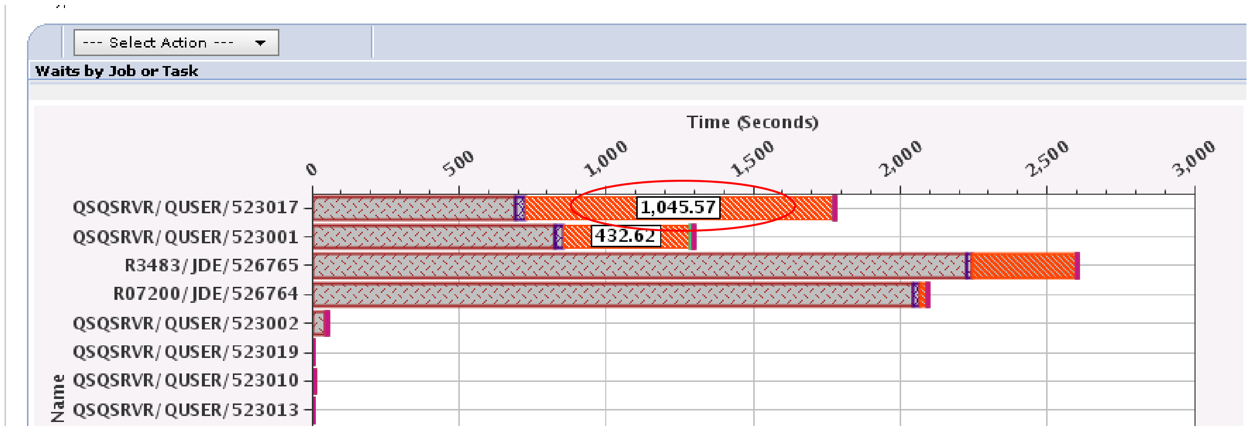


Figure 14 - Wait by job after TRACASPBAL

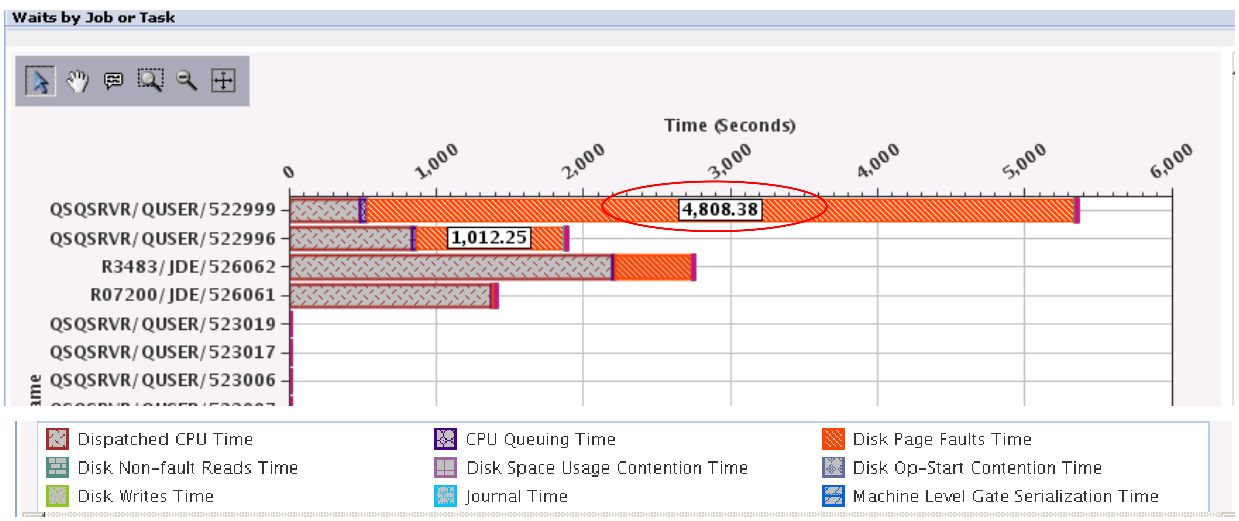


Figure 15 - Original waits by job

Also notice the reduction in the Dispatched CPU time for R07200 in these graphs. This is likely due to a reduction in the time the UBE waited for the database requests to return from the QSQSRVR job.

The test results show a substantial reduction in both the Disk Page Fault Wait time and in the job runtimes, particularly R07200.

In general, when disk performance improves, a number of other system-wide performance metrics can be impacted. CPU utilization typically increases. Jobs wait less time to read data from disk, and they are able to do more work in a shorter amount of time which results in higher CPU utilization. Additionally, when jobs process faster, they read data more quickly which can result in increased page fault rates as data is moved in and out of memory faster.

Let's review some additional graphs from System i Navigator to see if this occurred in the test scenario.

To compare CPU utilization, look at the CPU utilization by subsystem graphs.

Hint: This graph can be displayed from System Director Navigator using the following path: Expand IBM i Performance > In the right hand tab, Click Collection Services > Click CPU > Click CPU Utilization by Subsystem > In the Collection section, select the collection library and collection name > Click Display.

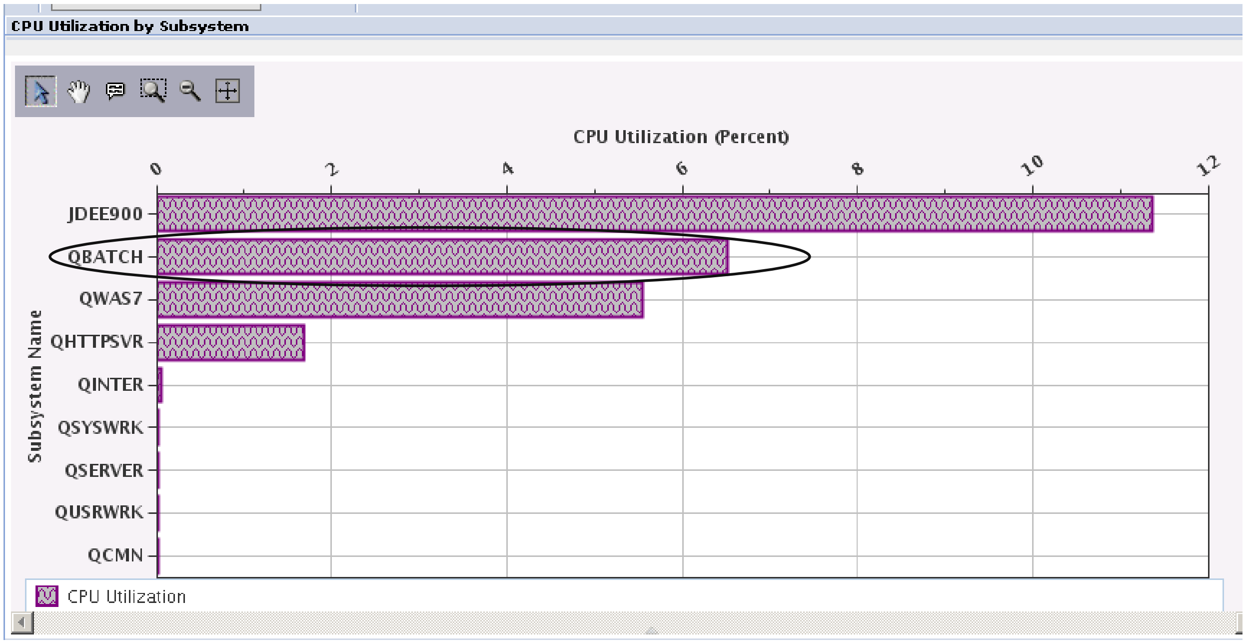


Figure 16 - CPU utilization by subsystem before TRCASPBAL

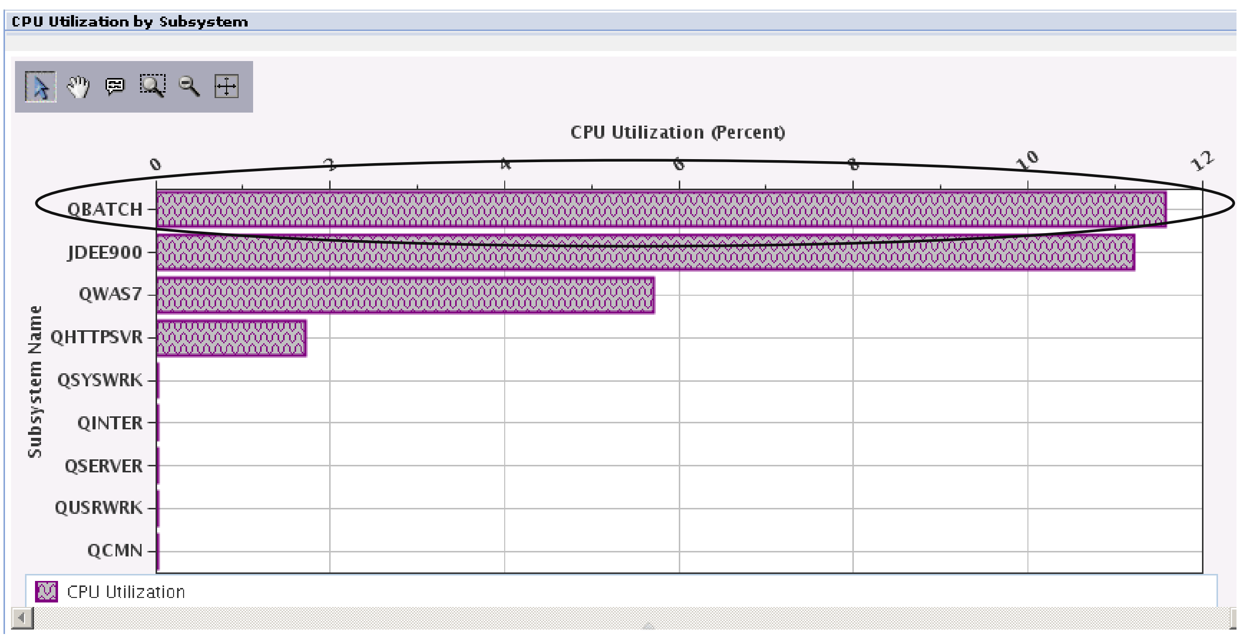


Figure 17 - CPU utilization by subsystem after TRCASPBAL

The QBATCH subsystem CPU utilization increased from approximately six percent (see Figure 16) to over eleven percent (see Figure 17).

Now compare the Page Faults by Pool for the two scenarios.

Hint: This graph can be displayed from System Director Navigator using the following path: Expand IBM i Performance > In the right hand tab, Click Collection Services > Click Page Faults > Click Page Faults by Subsystem > In the Collection section, select the collection library and collection name > Click Display.

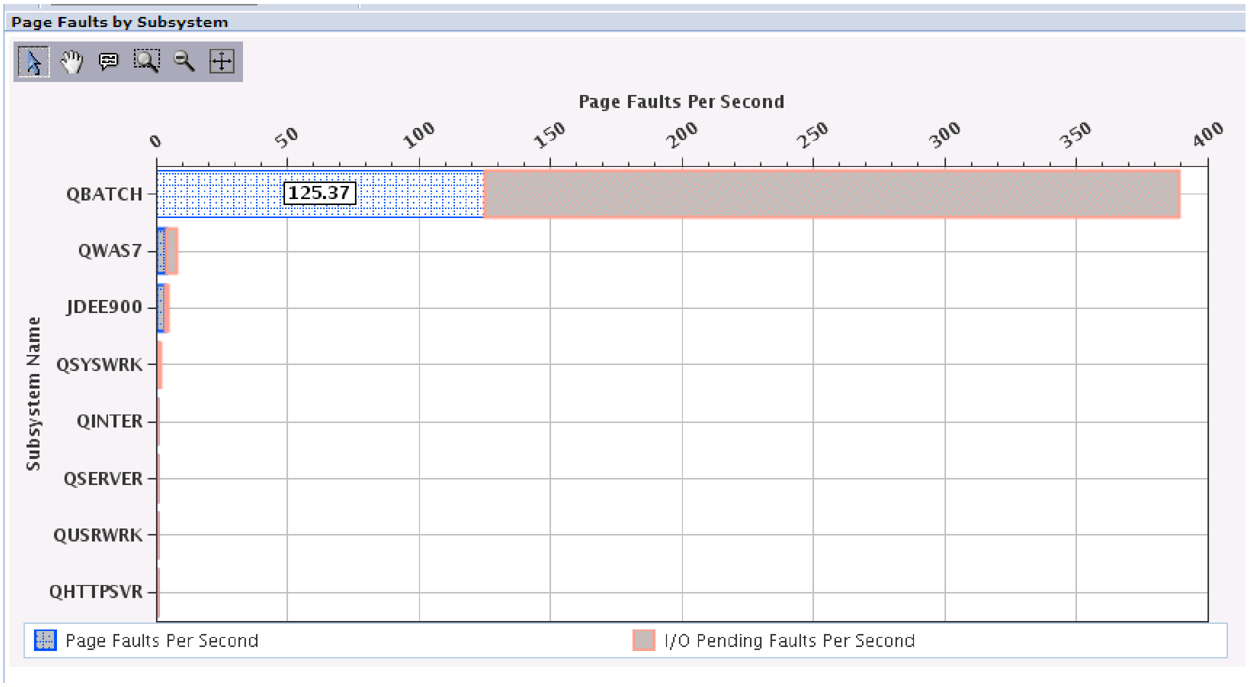


Figure 18 - Page faults by subsystem before TRCASPBAL

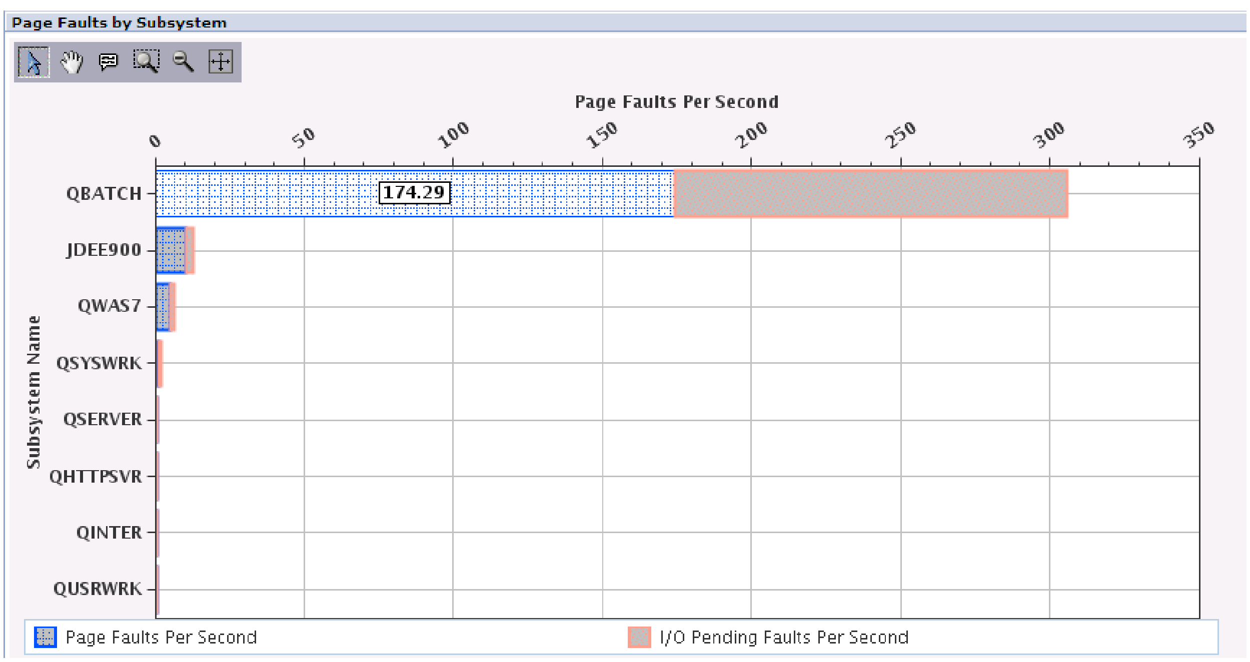


Figure 19 - Page faults by subsystem after TRCASPBAL

When comparing these results, we notice two different things occurred. First, page faults per second increased from roughly 125 (see Figure 18) to 175 faults per second (see Figure 19). We also see a reduction in I/O Pending Faults. I/O Pending Faults occur when a page is requested from disk, and the page is already in the process of being read by another thread. When handling this page fault, it is detected and the faulting thread simply waits for the read to complete. The reduction in I/O Pending Faults seen in this scenario is an outcome of the faster reads from the solid state drives.

Next examine the Resource Utilization Overview graphs to illustrate how moving data to the SSDs increased the system utilization for this scenario. These graphs are found directly under the Collection Services folder in the Investigate tab (see Figure 20).

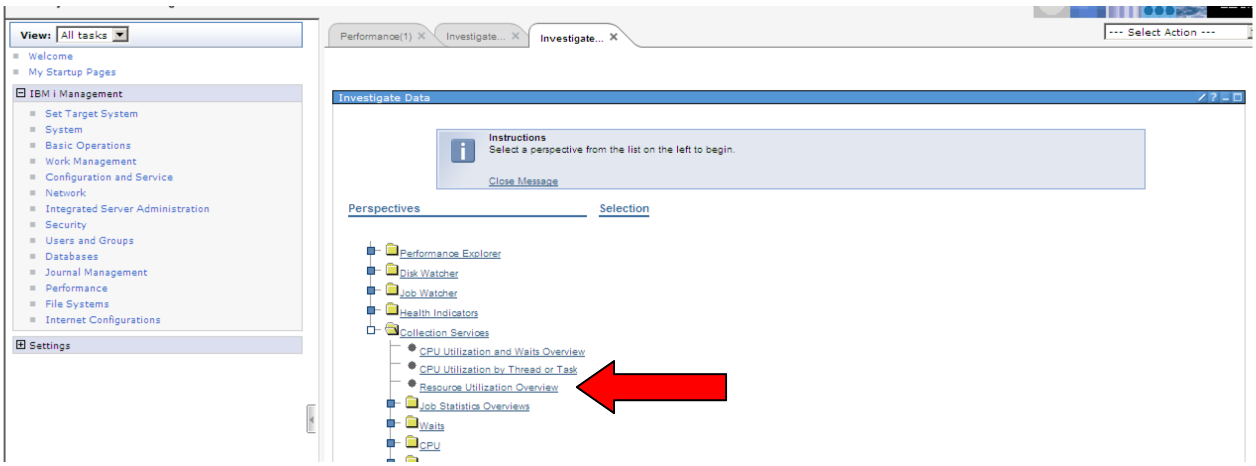


Figure 20 - Accessing the resource utilization graphs

Two graphs are displayed in the overview. The first one shows disk utilization by percent busy and percent full. The second graphs titled “Resource Utilization Rates” shows I/Os per second for several different data points. Compare the Resource Utilization Rates before and after data is moved to the solid state drives.

In this scenario, Figure 21 and Figure 22 show an increase in the logical I/Os per second (top line) after data was moved to solid state drives. Logical I/O is a unit of measure commonly used to determine the amount of work done on the system.

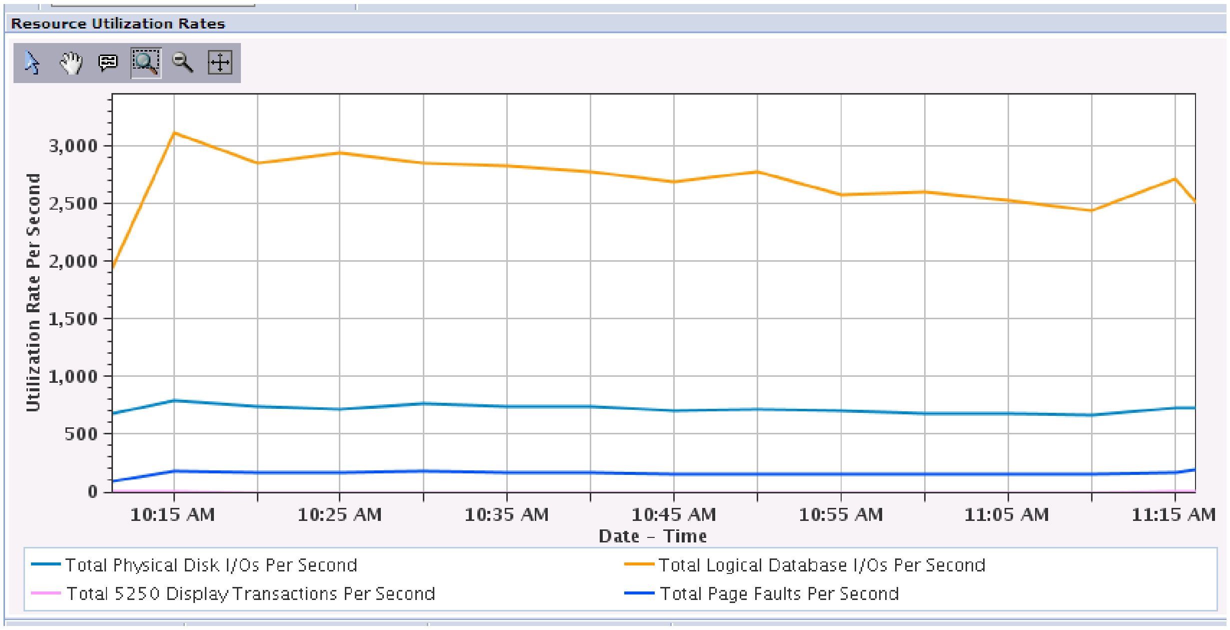


Figure 21 - Utilization rates before TRCASPBAL

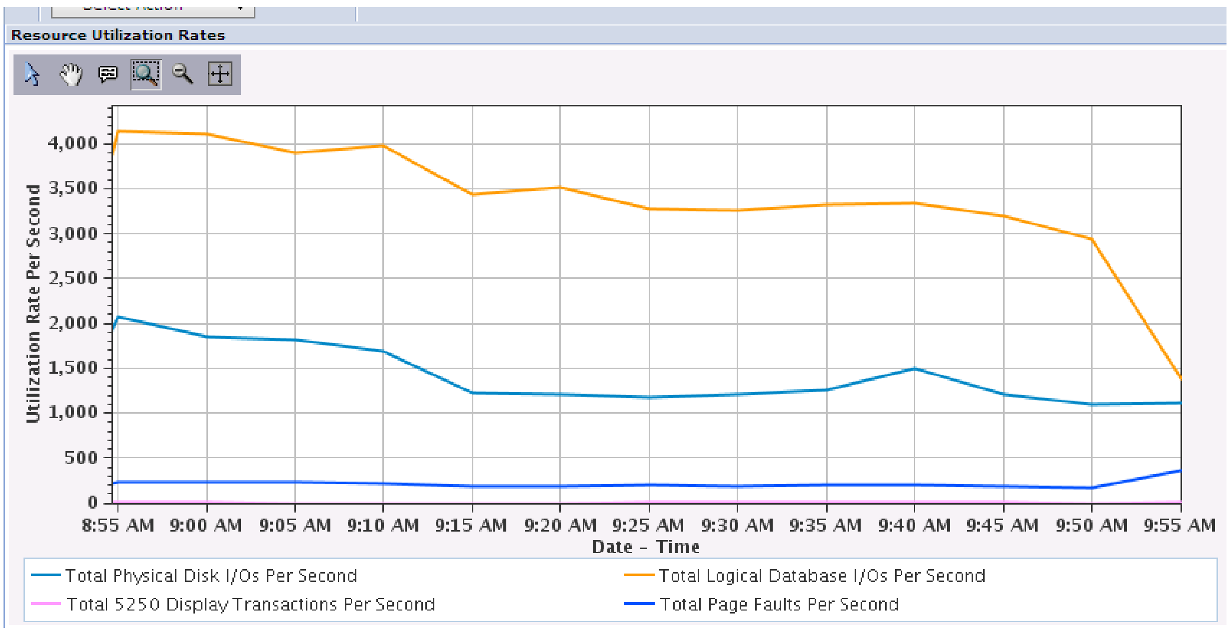


Figure 22 - Utilization rates after TRCASPBAL

Total Physical Disk I/Os per second (middle line) also increased after data was moved to the SSDs. This is due to the fact that data is read much faster from SDD than from HDD, and as a result, more I/Os can be processed.

After reviewing the results, we see all of the factors that we expect to see after moving data to solid state drives:

- Improved batch run time

- Increased work being done by the system, represented by Logical I/O per second
- Increased physical disk I/O represented by Total Physical Disk I/O
- Increased CPU utilization due to less time spent waiting for disk I/O
- Increased page fault rates due to increased work being done on the system

As the final step in the process, you can review the data that STRASPBAL moved to the solid state drives. One way to do this is to use a set of queries over the SYSPARTITIONDISK and SYSPARTITIONINDEXDISK views; these views are in the QSYS2 library. These two views contain tracking information about each table and index stored on each disk within the system. Tables and indexes may be spread across multiple disks. Querying this information shows each disk where a table or index is located. Example SQL statements are provided in Appendix A.

Next Steps

The "hot" data for an application changes over time as workloads increase and as the data volumes grow. Therefore, it is important to develop a methodology to keep the most active data on the solid state drives. There are several approaches which can be used separately or in combination to accomplish this.

The first approach is to use regularly scheduled cycles of the TRCASPBAL and STRASPBAL commands. One automated approach for this is discussed in section 8.3.2 of the "IBM i 7.1 Technical Overview". Remember that TRCASPBAL only collects statistics over jobs that are running on the system during the trace period, so the time period when the trace is run is important. Also keep in mind that STRASPBAL will remove any data from SSD that was not "hot" during the most recent trace.

TRCASPBAL alone may not always provide the most optimal performance in all situations. Consider the example where jobs execute only once over a specific set of data, payroll, or month-end or year-end processing. As a result, in i 7.1, the *MP balance type was added to the STRASPBAL command to simplify pinning data to the solid state drives. The *MP balance type has three options:

- *CALC - moves objects marked with a media reference attribute to SSDs and objects without a media preference
- *SSD - moves data marked without *SSD unit preference from SDD units to HDD units
- *HDD - moves data marked with *SSD unit preference from HDD units to SSD units

Once data is moved to SSD, using the *MP balance type appropriately will cause the STRASPBAL command to honor the unit preference flag to leave the data on the SSDs.

Note: The unit preference can be set with the CHGPF, CHGLF command. An example is CHGPF(PRODDTA/F0010) UNIT(*SSD).

A final consideration is use of an advanced macro to keep "unmarked" data off of the SSDs. Without this option, the system spreads data across all drives in an ASP regardless of whether they are SSD or HDD drives. To set this option, do the following:

- Issue the STRSST command
- Select Option 1. Start a service tool
- Select Option 4. Display/Alter/Dump
- Select Option 1. Display/Alter storage
- Select Option 2. Licensed Internal Code (LIC) data
- Select Option 14. Advanced analysis

- On the blank line, type smsetstayoffssd

For more information on the advanced macros, see the IBM technical document Customer Use of SSD (Solid State Drives).

Summary

We have gone through the process to:

- Identify jobs where performance may be improved with solid state drives
- Use TRCASPBAL and STRASPBAL commands to move active data to solid state drives
- Analyze the impact to performance after data is placed on solid state drives
- Keep data on SSDs current

Although both R07200 and R3483 saw performance improvements with the use of solid state drives, the performance improvement seen in each environment may be smaller or larger than those seen in the scenario described. Reasons for the different results include:

- Overall system CPU and memory utilization
- Data placement on disk drives
- Database optimization and index usage
- Current disk performance
- Customer specific configuration in the JD Edwards EnterpriseOne modules

Appendix A - Identifying the data that was moved to SSD

The following query will show the tables that have data on the SSDs and the amount data that was moved. This query looks at table from the PRODDTA schema and is ordered by the tables with largest amount of data on SSD.

```
SELECT MAX(table_schema) AS table_schema, MAX(table_name) AS table_name,

MAX(table_partition) AS table_partition,

SUM(CASE WHEN unit_type = 1 THEN unit_space_used ELSE null END)/1048576  AS
ssd_space_MB,

SUM(CASE WHEN unit_type = 0 THEN unit_space_used ELSE null END)/1048576 AS
non_ssd_space_MB

FROM qsys2.syspartitiondisk a

WHERE system_table_schema = 'PRODDTA'

GROUP BY a.table_schema, a.table_name, table_partition

having SUM(CASE WHEN unit_type = 1 THEN unit_space_used ELSE null END) is not
null

ORDER BY 4 desc;
```

Below are the results from this query for our test scenario. We see that large portions of the F06116, F3411, F3412 and F06136 tables were moved to SSD.

TABLE_SCHEMA	TABLE_NAME	TABLE_PARTITION	SSD_SPACE_MB	NON_SSD_SPACE_MB
PRODDTA	F06116	F061100004	1452	2
PRODDTA	F3411	F3411	1172	0
PRODDTA	F3412	F3412	325	1
PRODDTA	F06136	F06136	138	0
PRODDTA	F3460	F3460	133	155
PRODDTA	F4102	F410200001	113	1001
PRODDTA	F0709	F070900001	75	106
PRODDTA	F06106	F06106	73	1
PRODDTA	F0618	F061800001	56	0
PRODDTA	F06146	F06146	42	0
PRODDTA	F07352	F07352	35	10
PRODDTA	F060116	F060100001	28	0
PRODDTA	F06145	F06145	26	0
PRODDTA	F3111	F3111	14	4467
PRODDTA	F3413	F3413	10	90
PRODDTA	F060118	F06011	9	0
PRODDTA	F07353	F07353	8	56
PRODDTA	F060120	F06012	6	0
PRODDTA	F42199	F421900001	5	2681
PRODDTA	F07350	F073500001	4	5
PRODDTA	F0115	F0115	3	1
PRODDTA	F4211	F421100003	3	49603

Figure 23 - Results showing tables moved to SSD

To see the list of all the table where any portion of the data was moved to solid state. Use the following query.

```
SELECT DISTINCT table_schema, table_name, table_partition

FROM qsys2.syspartitiondisk a

WHERE system_table_schema = 'PRODCTL' and UNIT_TYPE = 1;
```

The queries for the SYSPARTITIONINDEXDISK view are shown below:

```
SELECT index_schema, index_name, index_member, index_type,

   SUM(CASE WHEN unit_type = 1 THEN unit_space_used ELSE null END)/1048576  AS
ssd_space_MB,

   SUM(CASE WHEN unit_type = 0 THEN unit_space_used ELSE null END)/1048576 AS
non_ssd_space_MB

FROM qsys2.syspartitionindexdisk  b

WHERE system_table_schema = 'PRODDTA'

GROUP BY index_schema, index_name, index_member, index_type

having SUM(CASE WHEN unit_type = 1 THEN unit_space_used ELSE null END) is not
null

order by 5 desc;
```

```
SELECT DISTINCT index_schema, index_name, index_member, index_type

FROM qsys2.syspartitionindexdisk  b

WHERE system_table_schema = 'PRODCTL' and UNIT_TYPE = 1;
```

Appendix B - Resources

SSD Resources

- IBM i 7.1 Technical Overview
  http://www.redbooks.ibm.com/redbooks/pdfs/sg247858.pdf
- Transform Your Batch Run Performance with SSDs and IBM i 7.1
  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/5cb5ed706d254a8186256c71006d2e0a/5ce40eb9ec61ebff862578a90067fc21/$FILE/TransformWithSSD.pdf
- Customer Use of SSD (Solid State Drives)
  http://www-912.ibm.com/s_dir/slkbase.NSF/0/d42db94d78598d038625785c006ddc31?OpenDocument&ExpandSection=3

General System Tuning

- End to End Performance Management on IBM i (November 2009)
  http://www.redbooks.ibm.com/redbooks/pdfs/sg247808.pdf
- IBM Power Systems Performance Capabilities Reference IBM i operating system 7.1
  http://www-03.ibm.com/systems/resources/systems_power_software_i_perfmgmt_pcrm_oct2011.pdf
- Performance Management for IBM System i Home Page
  http://www-03.ibm.com/systems/i/advantages/perfmgmt/
- IBM System i and IBM i Information Center
  http://publib.boulder.ibm.com/infocenter/iseries/v7r1m0/index.jsp
- IBM / Oracle Informational APARs
  http://www-03.ibm.com/systems/i/advantages/oracle/

General EnterpriseOne tuning

- IBM Power Systems with IBM i performance and tuning tips for Oracle's JD Edwards EnterpriseOne WebSphere-based HTML servers
  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101777
- IBM Power Systems with IBM i Performance and Tuning Tips for Oracle's JD Edwards EnterpriseOne 9.0
  http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP101504

Oracle Minimum Technical Requirements

- Oracle Minimum Technical Requirements (MTRs) (userid and password required)
  https://support.oracle.com/CSP/ui/flash.html#tab=KBHome(page=KBHome&id=()),(page=KBNavigator&id=(bmDocType=REFERENCE&bmDocTitle=JD%20Edwards%20EnterpriseOne%20Current%20MTR%20Index&viewingMode=1143&from=BOOKMARK&bmDocDsrc=KB&bmDocID=747323.1))

Trademarks and special notices

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

IBM, the IBM logo, Power, POWER7, Power Systems,and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

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

The information provided in this document is distributed “AS IS” without any warranty, either express or implied.

The information in this document may include technical inaccuracies or typographical errors.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Some information addresses anticipated future capabilities. Such information is not intended as a definitive statement of a commitment to specific levels of performance, function or delivery schedules with respect to any future products. Such commitments are only made in IBM product announcements. The information is presented here to communicate IBM's current investment and development activities as a good faith effort to help with our customers' future planning.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Photographs shown are of engineering prototypes. Changes may be incorporated in production models.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.