IBM Oracle International Competency Center
Noember 2006

IBM System i Performance Optimization with Oracle's JD Edwards EnterpriseOne

Improvements to Sales Order Update, Pick Slip Processing, Fixed Asset Depreciation, and Advanced Pricing

By Patrick Moore
Solution Sales Specialist
IBM Oracle International Competency Center

Table of contents

Executive summary

Oracle® and IBM have recently completed a number of joint projects to improve the performance of the fixed assets depreciation, sales order update, pick slip processing batch UBEs (Universal Batch Engines), and advanced pricing. This collaborative effort resulted in enhancements and tuning to the application software, configuration, database and operating system that in total delivered significant throughput improvements. These enhancements are now incorporated into the software so that all JD Edwards EnterpriseOne customers running in a System i™ environment will benefit.

Project background

The performance optimization project team was staffed with experts from the IBM Oracle International Competency Center (IOICC) located in Denver, Colorado, experts from the System i Benchmarking Center in Rochester, Minnesota, and Oracle software engineers from the Performance and Benchmarking and Software Engineering organizations. This team of experts worked together to deliver the tremendous performance improvements detailed below.

Test environment – UBE projects

A test environment was established on an IBM System i platform located at the IBM Benchmarking Center. This system was configured as shown in table 1.

System i – UBE projects	
OS level	V5R3
Model	870-2486
CPW Rating	20,000
Processors	16
Memory	96 GB
DASD – 63 x 70 GB drives	2 TB

Table 1. Hardware configuration

For each of the customer cases, a test environment was established by taking an image of the customer's entire JD Edwards EnterpriseOne production system and installing it on the target test system. This included duplicating the deployment server and using the customer's production database.

Customer case 1

A large manufacturer approached Oracle and requested assistance with their sales order and pick slip processing throughput, specifically the R42800 (Sales Order Update) and R42520 (Pick slips) UBEs.

This customer needed to process a high volume of sales orders and get them into the manufacturing cycle quickly to successfully meet the demands of their largest customers. Their sales order update process was processing 70,583 records per hour which was not sufficient to meet their growing business needs, and they were processing 78,246 records per hour for their pick slips. This was impacting their ability to fulfill orders in a timely manner for their largest customers.

The Performance Optimization Team loaded the customer's data on the System i platform and began running the batch jobs to simulate the customer's environment. The team of engineers then analyzed all areas that could impact performance - application code, technical foundation, database, operating system and configuration. They made several changes including:

- Changing the number of concurrent jobs to enhance throughput
- Adding several tables to cache to reduce I/O
- Applying cache to several tables to avoid redundant database I/O
- Changing the I/O to some tables to Primary Key to allow JDB caching
- Changing an algorithm in the application code to reduce repeated I/O
- Adding new UBEs to populate a table to avoid repeated hits to NULL records
- Applying system code fixes to improve the processing of numeric database values

A detailed list of remedies by application release that resulted from this optimization work with Sales Order Update is shown in Appendix A.

The end result was that the customer saw an 89% improvement in sales order processing – from 70,583 records per hour to 133,474 records per hour. With pick slips, the performance increased from 78,246 records per hour to 111, 220 records per hour, which was a 42% improvement.

Application	Before Optimization	After Optimization	Improvement
Sales order update	70,583 records per hour	133,474 records per hour	89%
Pick slip processing	78,246 records per hour	111,220 records per hour	42%

Table 2. Performance optimization results for Customer case 1

Customer case 2

A large retailer requested assistance from Oracle with their fixed asset depreciation processing, specifically the R12855 (Fixed Assets Depreciation) UBE. One of the biggest challenges in retail is the need to track and depreciate each fixed asset - and each rack in each store is considered a fixed asset, for example. This customer was having problems processing millions of records every month, depreciating each fixed asset, and meeting the monthly accounting close cycles. The fixed assets depreciation process was initially processing 33,670 records per hour. This throughput was not sufficient for them to complete their monthly financial close cycle and recognize all of the depreciation.

The Performance Optimization Team loaded the customer's data on the System i platform and began running the fixed assets depreciation jobs to simulate the customer's environment. The team of engineers then analyzed all areas that could impact performance - application code, technical foundation, database, operating system and configuration. They made several changes including:

- Placing depreciation formula tables into cache to avoid redundant database I/O

- Changing the application code to avoid excessive I/O to Fiscal Year / Date Patterns table for values which do not change to avoid redundant I/O
- Changing the application code to check decimal separator only once to avoid redundant processing and redundant database I/O
- Placing several tables in cache to reduce I/O
- Applying a system code fix to improve the processing of numeric database values
- Changing the configuration to disable outbound interoperability via processing options
- Changing the configuration to disable headers on output reports
- Increasing the number of concurrent jobs to enhance throughput
- Rebuilding indexes to 1 TB maximum size
- Rebuilding the F0911 table to reuse deleted records
- Applying the System i "holey writes" feature to improve the performance of INSERTs across concurrent jobs

A detailed list of remedies by application release that resulted from this optimization work with Fixed Asset Depreciation is shown in Appendix B.

These improvements delivered a whopping 4097% improvement in the fixed assets deprecation process – from 33,670 records per hour to 1,400,000 records per hour. As an added benefit of the optimization work that was done for this customer the R12825 (Year End Close) UBE throughput was improved by 25%.

Application	Before Optimization	After Optimization	Improvement
Fixed assets depreciation	33,760 records per hour	1,400,000 records per hour	4097%

Table 3. Performance optimization results for Customer case 2

Customer case 3

A large appliance and equipment manufacturer was experiencing performance issues with their complex advanced pricing algorithms. Oracle development identified a project team to address these performance issues with a goal to reduce CPU consumption and improve response times for sales order detail line entry when complex advanced pricing algorithms were used.

In this case the project team decided to test the efficacy of placing some of the most active system tables used in advanced pricing processing into a shared memory cache in order to reduce the repetitive accesses to these tables via the JDB APIs.

Test environment – advanced pricing project

The first step was to establish a test system based on the customer's actual operating environment in a logical partition (LPAR) of a System i 870 located in the IOICC labs at the JD Edwards Denver development facility. The configuration details for the LPAR and server are shown in table 4.

System i – advanced pricing project	
OS level	V5R2
Model	870-2486
CPW Rating	11,250 (of 20,000 total)
Processors	9 (of 16 total)
Memory	35 GB (of 96 GB total)
DASD – 43 x 70 GB drives	1.3 TB

Table 4. Hardware configuration

As with the previous UBE testing, the test environment attempted to duplicate the customer's operating environment as closely as possible and included using a copy of the customer's production database and deployment server.

Three use cases and the necessary test scripts were then developed, debugged and refined and the system tests were executed. The test results are shown in Table 5.

Case	Pricing Cache OFF	Pricing Cache ON	Improvement
5 lines / 139 adjustments			
Total COK CPU time (s)	3757	2483	34%
CPU consumption (%)	25.9	18.0	30%
EditLine average runtime (ms)	663	398	40%
Detail line average response time (s)	1.04	0.72	31%
5 lines / 4 adjustments			
Total COK CPU time (s)	2045	1963	4%
CPU consumption (%)	17.5	14.9	15%
EditLine average runtime (ms)	311	294	5.5%
Detail line average response time (s)	0.62	0.57	7.6%
18 lines / 139 adjustments			
Total COK CPU time (s)	15140	8299	45%
CPU consumption (%)	60.7	39	36%
EditLine average runtime (ms)	972	477	51%
Detail line average response time (s)	2.5	1.3	48%

Table 5. Performance optimization results for customer case 3

In addition, a detailed analysis of the debug logs and performance monitor data showed that for the 18 line / 139 adjustment use case overall select operations were reduced by 75%, fetch operations reduced by 36%, and open / close operations reduced by 44%.

These results demonstrate that customers using complex advanced pricing rules with large sales order volumes will see a reduction in CPU consumption and improved response times with the shared memory caching technique tested here. Results will vary and customers with simpler advanced pricing rules and lower sales order volumes will not see as much benefit.

The changes that resulted from this testing are generally available for customers in JD Edwards EnterpriseOne 8.12 and are known as the Accelerated Pricing Resolution Engine.

Conclusion

The performance results that the optimization team was able to achieve would could not have been delivered if either company had tried to solve the issues alone since the changes and recommendations spanned application software, hardware and middleware. This level of commitment and partnership between Oracle and IBM provides direct benefits to our customers – we focus our energies on tuning and optimizing our applications and platforms, which in turn help our customers run their businesses successfully. And that's the ultimate goal.

Appendix A – Sales Order Update list of remedies

Type of change	Description	SAR	ESU	
Application Code Change	Apply USER cache to F4095 AAI table – avoid redundant DB I/O	7096885	Xe	ESU JD23006.exe for Xe
			8.0	ESU JE9928.exe for ERP 8.0
			8.9	ESU PF3871.exe for 8.9
			8.10	ESU PG4664.exe for 8.10
			8.11	Base
Application Code Change	Apply USER cache to F4096 Flex accounting table – avoid redundant DB I/O	7096965	Xe	ESU JD23006.exe for Xe
			8.0	ESU JE9928.exe for ERP 8.0
			8.9	ESU PF3871.exe for 8.9
			8.10	ESU PG4664.exe for 8.10
			8.11	Base
Application Code Change	Change I/O to F1690 to Primary Key to allow JDB cache	7096973	Xe	ESU JD22022.exe for Xe
			8.0	ESU JE7324.exe for ERP 8.0
			8.9	ESU PF2357.exe for 8.9
			8.10	Base
			8.11	Base
Application Code Change	Change I/O to F0901 to Primary Key to allow JDB cache	7097010	Xe	ESU JD23006.exe for Xe
			8.0	ESU JE9928.exe for ERP 8.0
			8.9	ESU PF3871.exe for 8.9
			8.10	Base
			8.11	Base
Application Code Change	Change algorithm to reduce repeated I/O to F4201	7097141	Xe	ESU JD23919.exe for Xe
			8.0	ESU JE9561.exe for ERP 8.0
			8.9	ESU PF4496.exe for 8.9
			8.10	Base
			8.11	Base
Application Code Change	Add new UBE to populate F0901T table – avoid repeated hits to NULL records. There is a paper fix for 7097116.	7097116	Xe	ESU JD23699.exe for Xe
			8.0	ESU JE9136.exe for ERP 8.0
			8.9	N/A
			8.10	N/A
			8.11	N/A
Application Code Change	Apply existing ESU to avoid excessive Work Center error messages and jde.log error messages		Xe	ESU JD15204.exe for Xe
			8.0	ESU JE50.exe for ERP 8.0
			8.9	Base
			8.10	N/A
			8.11	N/A
JDB cache	Add F0901T to cache. There is a paper fix for 7097116.	7097116	Xe	ESU JD23699.exe for Xe
			8.0	ESU JE9136.exe for ERP 8.0
			8.9	N/A
			8.10	N/A
			8.11	N/A

JDB cache	Add several tables to cache via JDB_AddTabletoDBCache() API: F0101, F4101, F4102, F03012, F7307. There is a paper fix for 7097052.	7097052	Xe ESU JD24501.exe for Xe 8.0 ESU JE9928.exe for ERP 8.0 8.9 ESU PF2597.exe for 8.9 8.10 Base 8.11 Base
System code fix	Apply system code fix which improves the processing of numeric database values		SP22_T1
Concurrency	10 concurrent jobs on 18 CPU box		
Application Code Change	Apply a B9 load balancer to XE to implement concurrency (multiple parallel versions) more effectively	7097503	Xe Not planned
Third party product	Identified issue with the Vertex tax calculation software: locking on REGSEQ table in concurrent processing. Will be fixed in future Vertex code update.		

Appendix B – Fixed Asset Depreciation list of remedies

Type of change	Description	SAR	ESU
Application Code Change	Place Depreciation formulas tables (F12852, F12853) into USER cache – avoids redundant DB I/O	6949356 & 7006667	Xe ESU JD24449.exe for Xe 8.0 ESU JE9877.exe for ERP 8.0 8.9 ESU PF4338.exe for 8.9 8.10 Base 8.11 Base
Application Code Change	Avoid excessive I/O to Fiscal Year / Date Patterns table (F0008) table for values which do not change - avoids redundant DB I/O	6949356 & 7006667	Xe ESU JD24449.exe for Xe 8.0 ESU JE9877.exe for ERP 8.0 8.9 ESU PF4338.exe for 8.9 8.10 Base 8.11 Base
Application Code Change	Check decimal separator only once – avoids redundant processing and redundant DB I/O to the F00921	6949356 & 7006667	Xe ESU JD24449.exe for Xe 8.0 ESU JE9877.exe for ERP 8.0 8.9 ESU PF4338.exe for 8.9 8.10 Base 8.11 Base
JDB cache	Place a number of tables into JDB cache via JDB_AddTabletoDBCache() API: F0010, F0901, F0006, F0911T, F1201	6949356 & 7006667	Xe ESU JD24449.exe for Xe 8.0 ESU JE9877.exe for ERP 8.0 8.9 ESU PF4338.exe for 8.9 8.10 Base 8.11 Base
System code fix	Apply system code fix which improves the processing of numeric database values		SP22_S1
E1 configuration	Disable Outbound Interoperability via processing options		
E1 configuration	Disable headers on output report		
Concurrency	10 jobs / 20 jobs on 24 CPU box		
Database configuration	Re-build F0911 indexes to 1TB maximum size		
Database configuration	Re-build F0911 table to REUSE DELETED RECORDS		
Operating system	Apply the System i "Holey Writes" feature configuration to improve the performance of INSERTs across concurrent jobs		

Contact information

Please send questions or comments via email to IBM Oracle International Competency Center at ibmoracl@us.ibm.com

For more information

For additional information on integrated, collaborative enterprise solutions from IBM and Oracle, call 1 888 426-5505 or visit: ibm.com/solutions/oracle

IBM Corporation
Integrated Marketing Communications
Server Group
Route 100
Somers, NY 10589
U.S.A.
ibm.com

Visit *www.ibm.com/pc/safecomputing* periodically for the latest information on safe and effective computing. Warranty Information: For a copy of applicable product warranties, write to: Warranty Information, P.O. Box 12195, RTP, NC 27709, Attn: Dept. JDJA/B203. IBM makes no representation or warranty regarding third-party products or services including those designated as ServerProven or ClusterProven.

This publication was developed for products and services offered in the United States. IBM may not offer the products, services or features discussed in this document in other countries. Information is subject to change without notice. Consult your local IBM representative for information on offerings available in your area.

The examples given in this paper are hypothetical examples of how a customer can use the products described herein and examples of potential cost or efficiency savings are not based on any actual case study. There is no guarantee of comparable results. Many factors determine the sizing requirements and performance of a systems architecture. IBM assumes no liability for the methodology used for determining the configurations recommended in this document nor for the results it provides. Any performance data contained in this presentation was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements quoted in this presentation 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 presentation may have been estimated through extrapolation. Actual results may vary. Users of this presentation should verify the applicable data for their specific environment.

Information in this presentation concerning non-IBM products was obtained from the suppliers of these products, published announcement material or other publicly available sources. IBM has not tested these products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.