



**John Byrd
Rick Bardine
Vince Russo**

DWL Customer and IBM Performance and Scalability Results

A report of tests conducted by DWL (USA) Inc. and IBM Business Consulting Services.

Executive summary and findings

This IBM Redpaper describes the results of testing done by IBM® and DWL® (USA), Inc. to demonstrate the performance and scalability of the DWL product, DWL Customer™, on IBM @server pSeries hardware on an AIX platform, running IBM WebSphere® Application Server Advanced Edition.

The goal of the performance tests was to exercise a mix of transactions representative of those used in customer data hubs, using a peak-hour load. The tests were designed to simulate the demanding performance environments of large international financial institutions.

In terms of both volume of customer data and volume of transactions, the retail financial services environment is extremely performance-intensive. The results of testing within that environment indicate that this solution may be successfully implemented in other, less performance-intensive industries.

DWL Customer v.3.5 was tested with 85 million account records containing 105 million customer records in the database, at a rate of approximately 600,000 transactions per hour. *The results demonstrated that DWL Customer, using the IBM hardware and software configuration identified herein, delivered an average throughput of approximately 165 transactions per second with an average latency ranging from 124 milliseconds to 288 milliseconds;* the transaction mix included add, update, and inquiry transactions. During the test, more than 10,000 new parties were added per one-half hour. The performance results within this report suggest that DWL Customer can meet the demanding performance requirements of many organizations.

The goal of the scalability tests was to determine how DWL Customer scales when application servers are added to the base configuration. For the configuration tested, DWL Customer v.3.4 demonstrated near-linear scalability, as it achieved at least 80% of the

transaction throughput (volume/time period) on an additional WebSphere application server in comparison to the transaction throughput that was supported on the first application server.

See “Test results” on page 10 for more information.

DWL Customer background

DWL (USA), Inc. (<http://www.dwl.com/>) is a leader in Enterprise Business Services software, an emerging class of services-oriented applications. DWL works with organizations worldwide to address enterprise-wide customer data integration (CDI) strategies using DWL Customer, a leading enterprise customer data hub. DWL Customer unifies customer data and transactions from product silos and delivers this knowledge to all channels. It is based on a services-oriented architecture and built on J2EE standards, helping to make it a scalable and market-proven application for customer data integration. Additionally, DWL Customer can help organizations derive additional ROI from current Customer Relationship Management (CRM) implementations and even failed CRM projects that lacked sufficient customer data management foundations or a customer hub during the time of implementation.

DWL and IBM have a Global Strategic Alliance to deliver leading CDI solutions that transform the way the world's largest companies manage and use valuable customer information. Together, this global partnership delivers the outstanding software, robust hardware, middleware, and systems integration expertise that leading organizations need to solve their CDI challenges.

The DWL and IBM joint offering is designed to deliver a rapid ROI, unifying companies' existing IT investments through an enterprise customer hub. DWL Customer, a service-oriented application, provides all channels with a unified customer view and update environment, and aligns multiple front office systems with multiple back office systems in real-time. Rather than replace or make costly renovations to existing front or back office systems, DWL and IBM's services-based XML and J2EE platform allows companies to quickly repurpose their existing technology. This on demand solution also allows for a phased implementation that can easily be rolled out across additional products and lines of business. The result? Scalable, state-of-the-art technology that can help dramatically reduce customer processing costs, provide more knowledgeable and personalized service, and increase sales by responding in real time to customer life events. These are just a few of the reasons why some of the world's largest insurers, banks, and financial market companies are choosing DWL and IBM to make their customer-centric vision a reality.

Performance test and results

To validate the performance of the DWL application running on the IBM infrastructure, DWL and IBM measured the ability of DWL Customer v.3.5 to execute a mix of transactions typical of financial services firms (get, update, search, and add, for example), using expected transaction percentages based on call center traffic. The transaction volume was representative of peak-hour operations at a financial services company. DWL Customer was tested on IBM @server pSeries hardware running AIX. The application ran on top of WebSphere Application Server Advanced Edition V4.0.5, using IBM DB2® Enterprise Extended Edition (EEE) V8.1 and WebSphere MQ V5.3.1.

DWL Customer v.3.5 was tested with 85 million account records containing 105 million customer records in the database, at a rate of approximately 600,000 transactions per hour. *The results were that DWL Customer, running on this IBM hardware and software configuration, delivered an average throughput of approximately 165 transactions per second*

with an average latency ranging from 124 milliseconds to 288 milliseconds; the transaction mix included add, update, and inquiry transactions. During the test, more than 10,000 new parties were added per one-half hour. The tests indicate that DWL Customer is a highly robust application. Unlike front- and back-office application customer information files (CIFs) that may not be able to meet demanding performance requirements, DWL Customer was designed to perform in highly demanding performance environments, making it an optimal CDI solution for large organizations.

Performance test scope

The test was performed on a database with 85 million account records containing 105 million customer records. The transaction mix simulates target percentages for get (89%), search (1%), update (7%), and add (3%) operations. The transactions used DWL Customer services for data elements that are commonly accessed, such as name, address, city, state, zip, phone number, and e-mail contact information. The search transaction requests in the test included, at a minimum, the first digit of the last name and first name, as well as a further qualifying data element (address, zip, or e-mail address). The search result was limited to a maximum of 20 records and no in-text wildcards were allowed for any search data elements.

The targeted DWL transaction volumes were based on the peak-hour estimates for call center traffic across multiple channels: Channel-1 (90,000 contacts), Channel-2 (150,000 contacts), Channel-3 (30,000 contacts), Channel-4 (30,000 contacts), Channel-5 (6,000 contacts), and Channel-6 (150,000 contacts) (see Table 1). As noted earlier, the performance tests exceeded these initial target volumes, and were run at a rate of approximately 600,000 transactions per hour.

Table 1 Transaction volume assumptions

Channel	Peak Hour Contacts	DWL Transactions Per Contact	Totals DWL Transactions Per Peak Hour
Channel-1	30,000	3	90,000
Channel-2	150,000	1	150,000
Channel-3	10,000	3	30,000
Channel-4	10,000	3	30,000
Channel-5	2,000	3	6,000
Channel-6	30,000	5	150,000
Total			456,000

The targeted volume of transactions for the transaction mix by transaction type is shown in Table 2.

Table 2 Targeted transaction volumes

Transaction Name	Volume
SearchContract	275,130
AddContract	20,088
UpdatePartyContactMethod	11,340
UpdateContract	11,340

Transaction Name	Volume
GetCardholderByAdminSysKey	150,660
SearchPerson	3,780

Scalability test and results

The objective of the scalability test was to demonstrate that DWL Customer v.3.4 could achieve near-linear scalability across multiple WebSphere Application Servers. This would help demonstrate that there are no inherent bottlenecks in DWL Customer that prevent scaling the hardware and infrastructure to support customer-required transaction volumes for production (on-line and batch). Near-linear scalability was defined as being able to demonstrate at least 80% of the transaction throughput (volume/time period) on an additional WebSphere Application Server in comparison to the transaction throughput that is supported on the first application server. The application ran on top of WebSphere Application Server Advanced Edition V3.5, using IBM DB2® Enterprise Extended Edition (EEE) V7.2.6 for AIX.

For the configuration tested, DWL Customer demonstrated near-linear scalability.

Scalability test scope

The scalability test required highly scalable hardware to provide the ability to deal with capacity and tuning requirements. The test environment consisted of a 24 CPU IBM @server pSeries 690 Turbo server and an Enterprise Storage System consisting of 1.3 TB of disk. The pSeries server was partitioned for this test to support a database server and three application servers. Segue Silk Performer running on three load drivers was used to drive RMI messages against the base DWL Customer application running on three application servers. Each load driver was hardwired to one application server.

This test was primarily focused on:

- ▶ Demonstrating near linear scalability of query traffic on multiple application servers
- ▶ Demonstrating near linear scalability of insert traffic on multiple application servers

A set of scripts was developed that mirrored the load that needed to be simulated by the test. In particular, two types of scripts were developed:

- ▶ Scripts that request data based on an account number and return appropriate demographic data (name, address, and so on). The basic DWL Customer transaction used is SearchContract.
- ▶ Scripts that add new parties to the database. The basic DWL Customer transaction used is AddContract.

The database was initially populated with approximately 10,000 unique party records. During the many tests that were run, this number was gradually increased via the insert tests so that at the conclusion of the test there were 112,252 unique records in the customer database. The number of records allowed for the running of each test without the need for reusing a record. The size of the database was not large enough to eliminate reading of data from a buffered cache; accordingly, data was read from a buffered cache. The performance of the insert test was limited to establishing a baseline and validating that no unexpected bottlenecks exist for insert traffic.

The basic test combined a single application server and the database server. For queries, the goal was to achieve a throughput of 35 transactions per second. Virtual users were added to the test until the 35 transactions per second rate was achieved. The tests were maintained for a sufficient period of time to reach a steady-state condition. Early in the testing period tests were maintained for ten to 15 minutes. As it was determined that a steady-state was reached within one minute into the tests, many of the inquiry tests were subsequently run for only five minutes. Most insertions tests ran for 10 minutes or more with some reaching 30 minutes. Average response time statistics were somewhat influenced by the shortness of the tests, but graphs of all tests showed steady-state conditions. The CPU utilization and memory utilization for the application server never exceeded 80%. During the test, the database server resource utilization was always less than 40% and less than half of the maximum allowable connections were used. All formal tests were repeated at least three times in order to ensure that tests were consistent and repeatable. This proved to be the case for all tests.

Once the query test was successfully run with the single application server, the test was then run with a second application server. The goal of the second test was to meet 180% of the transactions per second rate of the first test, 63 transactions per second. Resource utilization of each of the application servers was always less than 75% and the database server resource utilization was well below the desired maximum of less than 80%.

Multiple iterations of all of the tests were run to ensure that the WebSphere Application Servers and database server were sufficiently tuned to meet the objectives (correct size of connection pool, for example). Tuning was not carried out to achieve optimum performance.

Once the query test was successfully completed, a similar set of tests was executed for account additions. The test throughput goal for account additions was 10,000 insertions in 30 minutes (six transactions/second).

Hardware environment

The main hardware for both test environments was a pSeries 690 Regatta Server (7040-681). The machine was configured with 24 1.3 GHz Power4 CPUs, 48 GB of main memory, and four I/O Drawers, as detailed in Table 3. The machine was partitioned for the test as four logical partitions (LPARs).

Table 3 Detailed system configuration for hardware

Component	Configuration
Central Electronics Unit	<ul style="list-style-type: none"> ▶ Configured for 4 LPARs ▶ 24-way 1.3 GHz POWER 4 Turbo Processors ▶ 48 GB RAM ▶ Redundant Power ▶ 32x max CD-ROM ▶ 20 GB/40 GB 4 mm tape drive
I/O Drawer 1	<ul style="list-style-type: none"> ▶ Two pairs of 18.2 GB disks (one LPAR) ▶ Fibre Channel Adapters ▶ SX-GB Ethernet Adapters ▶ Two 10/100 Ethernet Adapters ▶ Ultra SCSI Adapter
I/O Drawer 2	<ul style="list-style-type: none"> ▶ Two pairs of 18.2 GB disks (one LPAR) ▶ Fiber Channel Adapters ▶ SX-GB Ethernet Adapters ▶ Two 10/100 Ethernet Adapters ▶ Ultra SCSI Adapter

Component	Configuration
I/O Drawer 3	<ul style="list-style-type: none"> ▶ Two pairs of 18.2 GB Disks (one LPAR) ▶ Fiber Channel Adapters ▶ Two SX-GB Ethernet Adapters ▶ Two 10/100 Ethernet Adapters
I/O Drawer 4	<ul style="list-style-type: none"> ▶ Two pairs of 18.2 GB Disks (one LPAR) ▶ Fiber Channel Adapters ▶ SX-GB Ethernet Adapters ▶ Two 10/100 Ethernet Adapters

Database server hardware

One of the pSeries 690 LPARs was used for the database server. The LPAR was configured with 12 1.3GHz CPUs, 34 GB of main memory, two pairs of internally mirrored drives, a Gigabit Ethernet adapter, and dual Fiber Channel Interfaces to Disk Storage.

Disk storage hardware

Disk storage was provided by an IBM Enterprise Storage Server (ESS) Model 800 (2105-800). The ESS unit was configured with approximately 5 TB of disk storage (see Table 4).

Table 4 Detailed system configuration for storage hardware

Component	Configuration
Base Enclosure	<ul style="list-style-type: none"> ▶ 48 x 36.4 GB Disks ▶ 48 x 72.8 GB Disks ▶ 8 GB Cache ▶ Short Wave 2 GB Fibre Channel/FICON Adapters ▶ Redundant Power ▶ Battery Backup

Application server hardware

Each application server LPAR was configured with four 1.3 GHz CPUs, 4 GB of main memory, 2 pairs of internally mirrored drives, and a Gigabit Ethernet Adapter.

Test driver hardware

Figure 1 on page 7 shows the hardware configuration for the performance tests. Four Intel® Pentium® servers were used to drive the transaction load for the test. WebSphere MQ load balancing was used.

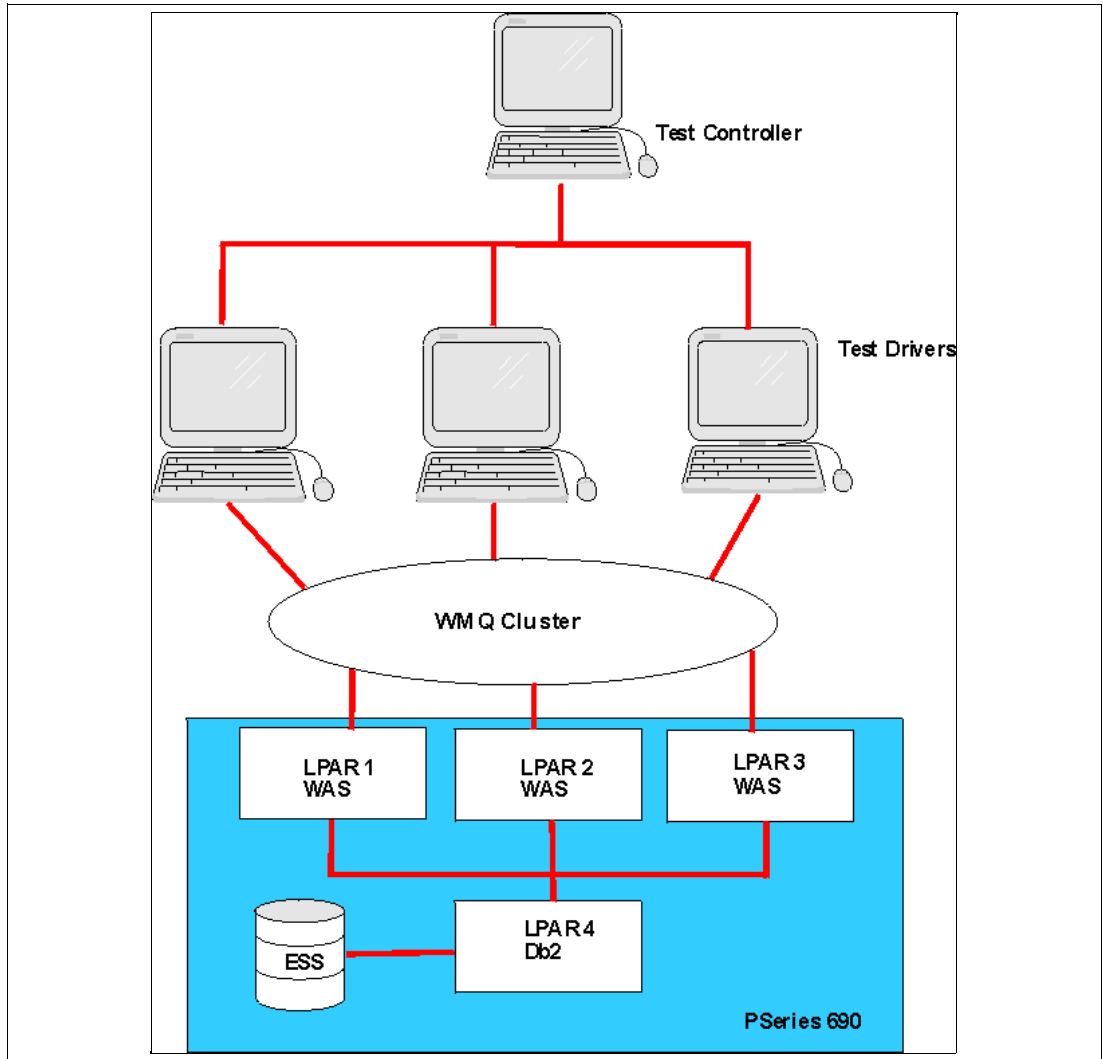


Figure 1 Hardware configuration for performance tests

Figure 2 on page 8 shows the hardware configuration for the scalability tests. Three Intel® Pentium® servers were used to drive the transaction load for the test. The test driver machines were dedicated to a given application server as required. This eliminated the need for a load balancer for the purposes of this test.

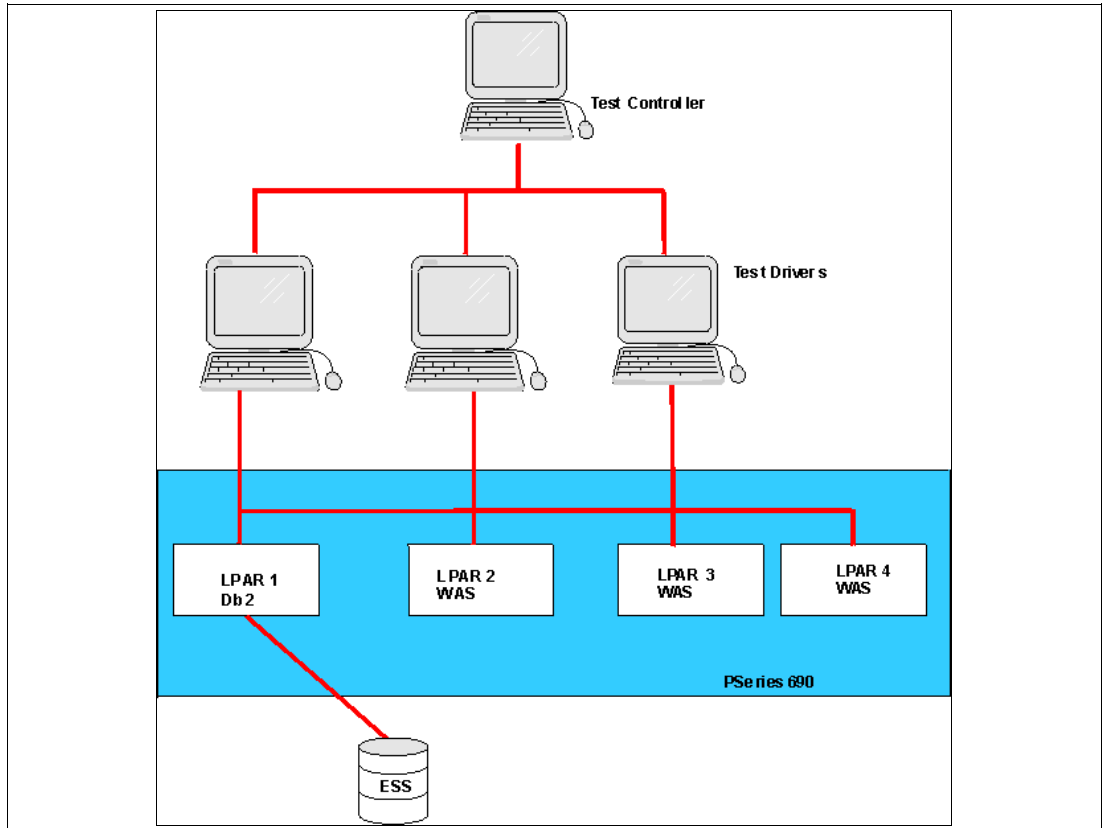


Figure 2 Hardware configuration for scalability tests

Figure 3 on page 9 shows a diagram of the network configuration used for the tests.

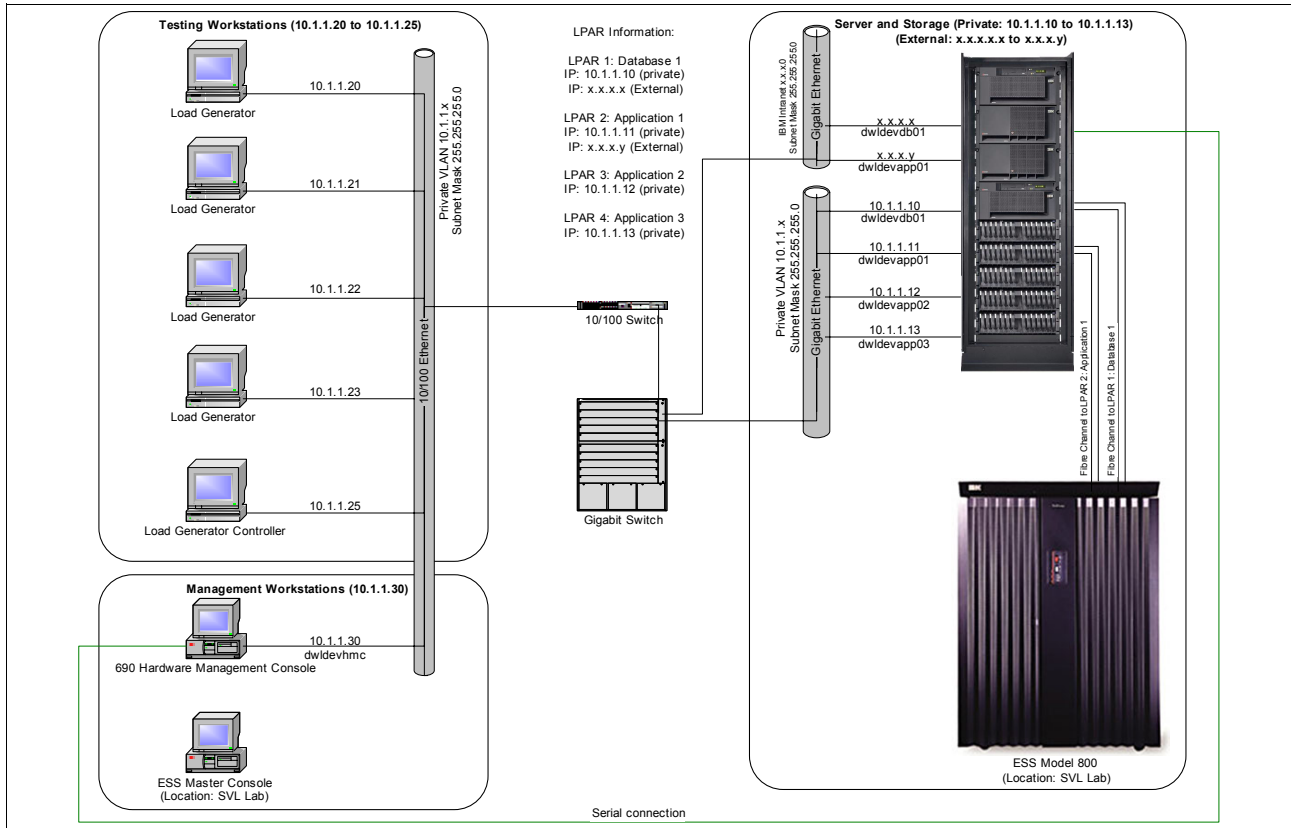


Figure 3 Environment network diagram

Software environment

The tests used the software indicated in Table 5.

Table 5 Software environment

	Performance tests	Scalability tests	Notes
Operating system for all LPARs	IBM AIX 5L Version 5.1 ML3	IBM AIX 5L Version 5.1 ML3	
Application server	IBM WebSphere Application Server V4.0.5	IBM WebSphere Application Server V3.5	A separate database instance for each application server was used to store the administration repository.
Database Server	IBM DB2 Enterprise Extended Edition (EEE) V8.1 for AIX	IBM DB2 Enterprise Extended Edition (EEE) V7.2.6 for AIX	

	Performance tests	Scalability tests	Notes
DWL Customer version installed on each application server	DWL Customer V3.5	DWL Customer V3.4	DWL Customer tables were installed on the database server using scripts provided with the DWL Customer product. The database was populated with unique party records, as indicated earlier in this document.
Test Driver software	Segue SilkPerformer V	Segue SilkPerformer V	
Message transport (used on both the test driver and application server machines as the mechanism to transport the XML messages)	WebSphere MQ V5.3.1	Not applicable	

Testing method

For the performance tests, a sample production system was set up to verify that DWL Customer is both a stable and robust solution. Two separate environments were used:

- ▶ An optimized environment that used a simulated database of customer records.
- ▶ An out-of-the box (“real-world”) environment that used sample customer records.

Both tests used a load generator tool to test simulated virtual users performing various transactions described earlier.

See “Appendix A: Test tools and methodology (optimized environment)” on page 13 and “Appendix B: Test prerequisites and methodology (out-of-the-box environment)” on page 16 for details.

The scalability tests were conducted using the same methodology described earlier.

Test results

Performance and scalability tests were run on unrelated data sets to invalidate the cache in the database server and the Enterprise Storage System (ESS). Multiple tests were executed to simulate the production transaction mix, as outlined earlier. A minimal number of errors (78) were encountered due to failed search transactions where the “search-person” was not in the database.

Multiple tests were executed and demonstrated either repetitive or stable results. All of the transactions were chosen randomly by the Segue test tool to avoid spurts of the same type of transaction. To ensure that the tests were run on unrelated data sets with non-cached data, the database and ESS storage caches were cleared. Test data files were generated to eliminate duplicate accounts both within and across files. A record was output to a file for each transaction processed. At the end of the test, the output files of all virtual users were concatenated and sorted by processing time; all server times are synchronized. During each

test, statistics were captured by a number of different tools, including **vmstat**, **iostat**, **filemon**, and **netstat**.

In the optimized environment, DWL Customer delivered an average throughput of approximately 165 transactions per second with an average latency of 124 milliseconds. The tests demonstrated that the results were consistent and repeatable. Changing data sets for each test ensured that the data was not in the DB2 or ESS cache and would result in I/O. The transaction mix percentages were rounded upwards to compensate for any errors. The same Search Person data file was used for both tests, accounting for the same number of errors.

The out-of-the-box performance test used a target transaction mix of 300,000 transactions per peak hour. The same transaction mix was included (get, add, update, search, and a custom transaction for retrieving cardholders). A total of 472,338 transactions were processed in under 47 minutes, 30 seconds (at a rate of approximately 600,000 transactions per hour). The demonstration yielded an average response time of 288 milliseconds.

In addition to these demanding load tests, the scalability of DWL Customer v.3.4 was tested by adding servers to the base configuration. The tests used a customized WebSphere MQ adapter. DWL Customer demonstrated near-linear scalability, as shown in Table 6. Performance did not degrade significantly with the addition of new servers to the cluster. DWL demonstrated at least 80% of the transaction throughput (volume/time period) on an additional WebSphere application server in comparison to the transaction throughput that is supported on the first application server.

Table 6 DWL Customer scalability results

Test	Number of Servers	Test Length (s)	Virtual Users	Think Time (ms)	Response Time (ms)	Throughput (trans/s) ^a	AppServer CPU	DBServer CPU
1	1	300	10	100	89	48.5	66	5
2	1	300	10	100	87	47.5	65	5
3	1	300	10	100	86	47.7	65	5
4	2	300	20	100	82	94.9	65	9
5	2	300	20	100	87	92.3	64.5	9
6	2	300	20	100	88	92.2	65	9
7	3	300	30	100	90	128	65	14
8	3	300	30	100	88	128.2	65	14
9	3	300	30	100	88	127.7	65	14

a. These numbers include ramp-up and ramp-down times.

The single application server loads (Tests 1 through 3) indicate a linear increase in CPU load with an increase in DWL Customer transaction load (DCTs/second).

Conclusion

The performance tests indicated that the DWL Customer application, running WebSphere Application Server Advanced Edition on IBM @server pSeries hardware on the AIX operating system, could meet the throughput and latency requirements needed for a transaction volume of 600,000 transactions per hour against a database of 105 million customer records. DWL Customer delivered impressive transaction response times that

should satisfy the requirements for most companies looking to deploy an enterprise customer data hub. In addition, DWL Customer demonstrated near-linear scalability for the stated configuration. These results suggest that DWL Customer, using IBM infrastructure, can handle the high levels of data load and transactions demanded by financial services organizations.

Appendix A: Test tools and methodology (optimized environment)

Various tools were used to drive the load and collection performance and resource metrics, as described below.

Test drivers and response metric collection

Segue Silk Performer V was used to drive a load using virtual users spread across multiple load driver machines. The virtual users used a Java Driver Script to send the XML requests to a custom built MQ interface. A program on the application server pulled these messages from the queue and made the request to the DWL application using RMI. The subsequent XML responses were put back into the queue and read by the Java Driver Script.

The Java Driver scripts used Segue Java APIs to return results to Silk Performer for collection and formatting.

Server resource metric collection

A Web-based server monitoring tool was used to collect server resource metrics. The tool allows performance metrics to be gathered over the network without installing agents on the servers. All data is stored on the server. From the data, the tool can generate reports or export the data to be used in other programs (such as Excel). It can monitor various components of servers. For example, it is configured to track generic server monitors (CPU utilization and disk space), application server monitors, and database server monitors.

Some real-time measurements were also taken using `vmstat`. The WebSphere Resource Analyzer was used to monitor various run-time performance parameters for the application servers.

Performance measurement approach

The effort consisted of the following activities:

- ▶ Update to WebSphere Application Server V4.0 Advanced Edition and DB2 Enterprise Extended Edition V8.1.
- ▶ Install DWL Customer base package Version 3.5 Gold.
- ▶ Validate installation by running the DWL standard installation test XML, which was used to validate DLW Customer function.
- ▶ Technical Planning - Develop a detailed technical plan for the tests, including:
 - Specific technical objectives.
 - Performance test scope.
 - Assumptions.
 - A prioritized set of tests with acceptance criteria.
 - Test environment.
 - Test approach.
 - Required technical skills, personnel, and responsibilities.

- Workload definition matrix.
- Systems validation testing, measurement, and evaluation.
- Test data requirements.
- ▶ Script Development and Test - Capture detailed script information in machine-readable and narrative form:
 - Establish the script test environment (a subset of the performance test environment).
 - Create a framework for script development (connection process, error recovery, and form generation).
 - Collect test input data (for example, user IDs and passwords, variable key fields, and names).
 - Code and test scripts and MQ interfaces.
- ▶ Test Execution and System Measurement - Verify test environment setup:
 - Collect an appropriate volume of test data input.
 - Populate the database with the required number of records.
 - Run a series of tests at increasing transaction rates.
 - Collect and analyze system measurement statistics.
 - Assist system integrators with tuning changes, as required.
 - Re-run tests until performance objectives are met or the system capacity is exceeded.
- ▶ Report Preparation - Collect test results and measurement data:
 - Create tables and graphs of system performance.
 - Write a test summary with conclusions and recommendations.

Database population

The database was populated with approximately 105 million unique party records, with 83 million accounts. The files were loaded into the database using the DB2 load utility, which enables you to specify an input file and target table with certain parameters, such as the delimiter, temp files, message file, dump file, and recoverable. DWL and IBM provided the key relationships and table load order required to load the DWL Customer database tables that must be populated or partially populated.

Validation

Prior to running any tests, the environment was validated by running the test cases shipped with DWL Customer. In addition, each of the scripts was validated by running tests with single users and with multiple unique users to ensure that the correct data was returned for each request. The results of the validation were used to determine the number of virtual users necessary to meet the desired load and transaction mix.

Methodology

The basic performance test consisted of simulating the specified load and transaction mix against an environment consistent with what is estimated to be required for a single production stream. The load tester's virtual users were split among the transaction scripts to achieve the required transaction mix. Tests were run for a minimum of fifteen minutes and upwards of one hour in order to ensure that the ramp-up of the virtual users did not affect the average response time statistics and to simulate peak hour transaction loads. Tests were run

multiple times to verify consistent results. A separate longer test was run in order to demonstrate the ability to meet the expected number of insertions in a half-hour.

Metrics

During test execution, data was collected for transactions per second and transaction response time. In addition, the resource utilization was monitored for the WebSphere application servers and database server. Resource measurements included CPU utilization, memory utilization, database connection pools, EJB Connections, and database I/O.

Appendix B: Test prerequisites and methodology (out-of-the-box environment)

The following tasks were performed prior to running the out-of-the-box tests:

- ▶ Populate the database with 83 million contracts and 105 million customer records. This represented sample customer data.
- ▶ Extract 3 million random contract IDs from the database to generate 20 discrete datasets of 150,000 records each (no duplicates).
- ▶ Randomly select names from the database to support the personSearch transaction.
- ▶ Load files into each of three load driver servers and configure each server for 13 virtual users. Virtual users were adjusted to ensure that the elapsed test time approximated one hour.
- ▶ Clear the ESS and DB2 caches.

The test methodology included the following:

- ▶ All tests were run a minimum of three times.
- ▶ Stringent change controls were used (changes were limited to one parameter at a time).
- ▶ Regression testing was mandatory.
- ▶ Random transaction datasets (with no duplicates) were generated.
- ▶ All transaction datasets were tracked and all results were archived.
- ▶ The ESS cache and DB2 cache were cleared between tests.
- ▶ The following were actively monitored: CPU, memory, disk, and network utilization performance.

Appendix C: Further information on hardware and software

This appendix provides additional information and Web site references for the hardware and software used in the testing.

About DWL Customer

See <http://www.dwl.com/na/solutions/index.php> for information on DWL Customer.

About IBM eServer pSeries products

See the following Web site for information on the IBM @server platforms:

http://www-132.ibm.com/content/home/store_IBMPublicUSA/en_US/eServer/pSeries/pSeries.html

About WebSphere

IBM WebSphere Application Server is the market-leading Java™ technology-based Web application server, integrating enterprise data and transactions with the e-business world. It provides a rich, e-business application deployment environment with a complete set of application services, including capabilities for transaction management, security, clustering, performance, availability, connectivity, and scalability. See <http://www.ibm.com/software/info1/websphere/index.jsp> for more information on WebSphere.

Appendix D: About IBM Business Consulting Services

See <http://www.ibm.com/services/bcs/> for information.

Notices and trademarks

© Copyright IBM Corporation 2004. All rights reserved.

Printed in the U.S.A.

06-04

All Rights Reserved

DWL is a registered trademark and DWL Customer is a trademark of DWL (USA) Inc.

Segue and SilkPerformer are registered trademarks of Segue Software, Inc.

Intel and Pentium are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

IBM, the IBM logo, AIX, DB2, DB2 Universal Database, IBM @server pSeries, Lotus, and WebSphere are trademarks or registered trademarks of IBM Corporation in the United States, other countries, or both.

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

Other company, product and service names used herein may be trademarks or service marks of other companies.

IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as transaction mix, hardware platform, network parameters, database content, and database size. Therefore, no assurance can be given that individual users will achieve throughput or performance improvements equivalent to the ratios stated here, even if they use the same or similar equipment and software applications.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM AND DWL EXPRESSLY DISCLAIM ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM and DWL shall have no responsibility to update this information, and this document could include technical inaccuracies or typographical errors. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, and so on) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements, or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility, or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding IBM patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing

IBM Corporation

North Castle Drive
Armonk, NY 10504-1785
U.S.A.



For additional information, please contact your local DWL or IBM sales representative.

DWL Relationship contact: Steve Leightell, DWL Alliances, (416) 682 3059, sleightell@dwl.com

IBM Relationship contact: Michele Brignull, Certified Client Executive – IBM, (802) 878 1990, mbrignul@us.ibm.com

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.