Red Hat Reference Architecture Series

Performance & Scalability of the PROFILE® Core Banking Platform

(based on Fidelity National Information Services PROFILE®)

Profile 7
GT.M 5.2
Red Hat Enterprise Linux 5
HP DL580 G5

Version 1

November 2007

Performance & Scaling of the
PROFILE® Core Banking Platform
(based on Fidelity National Information Services PROFILE®)

1801 Varsity Drive
Raleigh NC 27606-2072 USA
Phone: +1 919 754 3700
Phone: 888 733 4281
Fax: +1 919 754 3701
PO Box 13588
Research Triangle Park NC 27709 USA

The GPG fingerprint of the security@redhat.com key is:
CA 20 86 86 2B D6 9D FC 65 F6 EC C4 21 91 80 CD DB 42 A6 0E

Table of Contents

1. Executive Summary

Fidelity National Information Services, Inc. (FIS), a worldwide leader in processing and technology solutions for financial institutions, recently announced new performance benchmarks for FIS' PROFILE® core banking application, operating on one of the market's newest technology platforms created through its go-to-market alliance with leading technology providers Red Hat, Intel and HP.

On this market-leading technology stack consisting of the latest HP ProLiant DL580 G5 server platform featuring Intel 7300 series quad-core Xeon processors, Red Hat Enterprise Linux 5 (RHEL 5) operating system and Cisco Systems networking infrastructure, PROFILE's performance was benchmarked at more than 3,200 online banking transactions per second on a 25 million account database. The real-time core banking solution also processed 25 million interest accruals and balance accumulation updates at a rate exceeding 25,800 transactions per second, or more than 93 million transactions per hour.

	Transactions Per Second (TPS) Rates		
	Online TPS	Online Response Time (seconds)	Accruals Processing Accounts/Sec
10 Million Accounts	3,682	0.031	31,056
25 Million Accounts	3,269	0.036	25,853

"As demonstrated in these benchmark results, PROFILE continues to surpass performance and scalability standards throughout the financial services market, exceeding the capacity requirements of 99 percent of the banking market," said Frank Sanchez, president of FIS' Enterprise Solutions division. "It also showcases one of the industry's lowest total-cost-of-ownership systems in the market today."

The tightly engineered market alliance among FIS, Red Hat, Intel and HP and its resulting technology platform is distinctive not only because it creates one of the best business- continuity models in the industry, but because it also creates a packaged, validated and certified configuration that is generally available across all components.

"By bringing together these powerful components of the overall solution, this alliance highlights the performance, the availability and the operational completeness of the solution," said Sanchez. "This unique alliance and the impressive total cost of ownership advantages it brings to the financial services markets directly support our ongoing thought leadership toward promoting new technology solutions to meet emerging market needs and our joint commitment to providing the industry with the lowest total cost of ownership of any banking platform system."

Robert Hunt, research director in the Retail Banking Practice at TowerGroup believes that a compelling business case exists for mid-tier banks to implement a single, integrated core banking system that utilizes a low-cost operating platform.

"Most mid-tier banks continue to process their core systems using best-of-breed mainframe-based systems developed in the 1980s," said Hunt. "The hardware and software savings achievable by modernizing the core processing environment can provide banks with a significant cost advantage."

PROFILE is FIS' premier real-time, multicurrency core banking system that supports hundreds of institutions around the world, ranging from de novo startups to top-tier global banks. Companies using PROFILE as their core banking system experience industry-leading total cost of ownership benefits based on lower infrastructure costs, ongoing operating costs and increased productivity.

PROFILE® running on Red Hat Enterprise Linux & Intel Xeon based Servers provides mainframe-class performance at a fraction of the cost with greater than 10x savings in cost per account, per month compared to legacy systems because of a combination of factors including:

- Savings on Hardware = Commodity Intel Xeon processor based servers compared to proprietary servers
- Savings on Open Source Operating System = Red Hat Enterprise Linux 5 (RHEL 5)
- Savings on Open Source Database = FIS GT.M 5.2

As testing and development continue, additional savings are expected in PROFILE and other components of the FIS banking application environment which surrounds PROFILE:

- Savings on Logical Volume Manager (LVM), Multi-path I/O Software (MPIO), Global File System (GFS) = included with RHEL 5
- Savings on Open Source Java Enterprise Edition, Enterprise Portal, SOA = JBoss

2. FIS PROFILE® - Architecture

2.1 Multi-Threaded, Message-Based Online Processing

The PROFILE Application Server is a long running process designed to receive a message from an external touchpoint, and format, interpret, process, and commit a transaction to the database and then return a response to the originating client. This server architecture is available to all types of clients, including front-end applications as well as middle tier solutions (e.g. FIS Xpress application server) by exposing API's and specific formats and standards (such as IFX). .

Examples of client applications include FIS Touchpoint solutions, FIS Profile Direct, 3rd party teller systems, online debit networks and ODBC/JDBC clients.

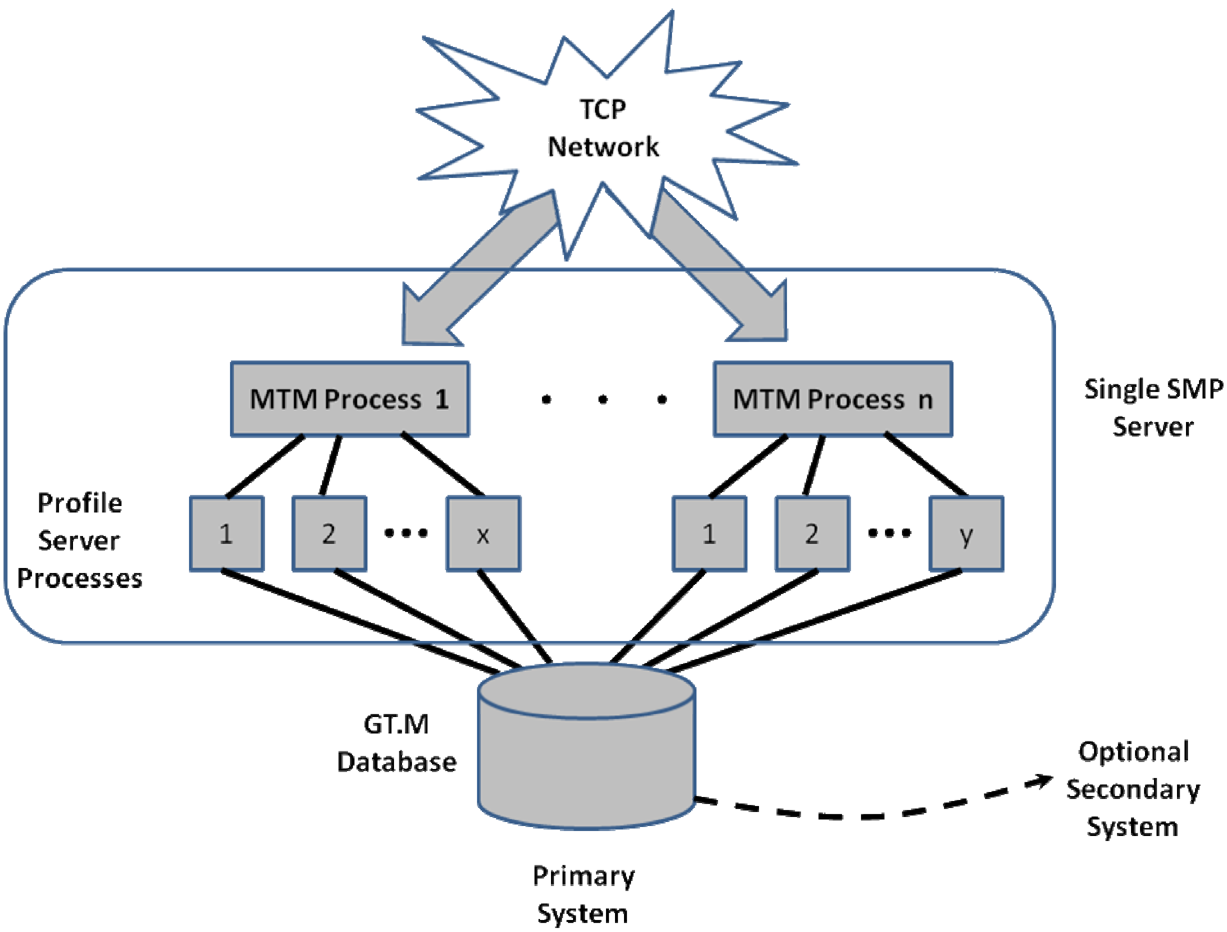


Figure 1

The Profile Application Server depends on a mechanism to deliver messages across LANs/WANs and in to the Profile system for processing and subsequent delivery of responses back to the originating client. The proprietary FIS mechanism used here is a communications monitor called the Message Transport Monitor (MTM). The MTM receives messages sent to the Profile Application Server over TCP/IP. The MTM routes client messages to an available Profile server from a pool of processes that it manages and also routes the corresponding reply message back to the originating client. The MTM uses queues to send and receive messages to/from the FIS Profile server processes.

FIS Profile server processes are multi-purpose, homogenous instances of the application, each fully capable of responding to all supported message types, including remote procedure calls (RPCs) to the PROFILE application, SQL transactions, industry-standard and proprietary financial transactions and customized message interfaces. Since any application server process can respond to any and all messages, separate function oriented servers are not required and server-to-server inter-process communication is eliminated. This architecture simplifies configuration and tuning requirements and optimizes server availability. Multiple server processes provide scalability in an SMP environment, improve resource overlap and utilization and also improve reliability and response time consistency. In the event that any one server process encounters a long running transaction, the effect is limited to that single transaction.

The FIS Profile banking server architecture supports multiple MTM processes, each listening on its own socket and with its own pool of application server processes. In addition to providing “horizontal” scalability, utilizing this capability can provide application granularity by establishing *classes* of service. For example, client service representatives providing private bank services to high net worth customers could be associated with separate MTM and banking server processes. As a practical matter, the level of performance achieved already means that few, if any, customer sites choose to run multiple MTM processes. Rates exceeding 3,200 online financial transactions per second have been achieved in benchmarking.

2.2 Multi-Threaded Scheduled Processing

The accounting and/or operational requirements of a financial institution may dictate that certain processes should be performed on a periodic, or scheduled basis (e.g., daily, monthly or annually). Many of these processes, such as account interest accrual and posting, involve a high percentage and sometimes all of an institution’s accounts. Traditional, batch core processing systems perform these activities as jobs that move data from one flat file to another and can operate rapidly because the flat files are optimized for sequential access. However, real-time processing demands have required transaction consistency across the entire database 100% of the time.

FIS Profile is a real-time system that processes client financial transactions on a 24x7 basis. Within its architecture, scheduled processes must interact with the same database as the online transactions and must serialize with other transactions occurring concurrently. Typically, scheduled processes must complete within a predetermined processing window (for example, there may be a deadline to provide a feed to a general ledger system), resulting in throughput requirements that are often significantly greater than those of online transaction processing.

For example, the peak online transaction rates of even the world's largest banks are typically below one thousand transactions per second, a number that FIS Profile can handle with capacity to spare. On the other hand, the overnight processing of a bank with tens of millions of accounts requires scheduled transaction rates on the order of many thousands of accounts per second.

FIS Profile real-time scheduled processing is designed to achieve maximum performance on today's multiprocessor/multi-core computer systems. All standard scheduled processes that involve a large number of records are multi-threaded. In multi-threaded scheduled processing, multiple server processes are started, each of which works separately and independently on a portion of the total workload.

A process called the Heavy Thread Manager (HTM) parcels out groups of records to the server processes. Since the work is distributed among many server processes, the overall throughput is multiplied and the elapsed time to complete a scheduled process is reduced. The actual time to complete a scheduled process is determined by a number of factors including the hardware configuration, logic complexity and I/O requirements of the scheduled process and overall system activity. Rates of over 25,000 scheduled transactions per second have been achieved during benchmarking.

FIS Profile processing and the DBMS have been designed to work with each other. During scheduled processing, the updating of each account is treated as a transaction and each account is therefore a checkpoint. If the system crashes during a scheduled process and then recovers, the restarted scheduled process determines the last account processed and then continues with the next account.

FIS Profile real-time scheduled processes use a relaxed algorithm to commit transactions to the database. In the event that the system crashes during a scheduled process the work is reprocessed from the point where the last transaction was committed, or checkpointed. This feature substantially improves throughput in time-critical scheduled processing at no cost during normal processing. If an online transaction is received and processed during a scheduled process, any preceding scheduled transactions are immediately check-pointed, thereby maintaining transaction serialization.

The combination of multi-threaded real-time scheduled processing and checkpointing yields scheduled-processing rates on the order of thousands of transactions per second. This throughput is required to meet the needs of financial institutions with millions to tens of millions of accounts. FIS Profile is unique in its ability to synchronize scheduled and online transactions, supporting the requirements of a true 24x7 real-time system.

3. FIS PROFILE® – Benchmarking Methodology

Depending on the specific platform solution being benchmarked, databases with several million accounts are routinely tested. These databases have a distribution of accounts based on a matrix of products typical to the target market – very large banks, within a specific global region.(in fact, the standard methodology evolved from a benchmark conducted in 1997 for a multi-national bank that required the Profile database to scale to 40 million accounts.).

The benchmark consists of two parts, real-time scheduled processing and online transaction processing. The objective of real-time scheduled processing is to measure how efficient a set of standard Profile scheduled processes that represent a typical set of tasks required in daily overnight processing can complete. The objective of the online transaction processing is to create a maximum load of transactions using workload generator(s) to determine how much transaction throughput the system can sustain.

The benchmark test suite represents key business functions as well as significant underlying processing against the Profile database. Online transactions are measured in terms of response times of complete and atomic messages processed by the Profile Application Server. The financial transaction messages process two sided transactions (debits and credits to the account as well as the general ledger account). The real-time scheduled process transactions selected for the tests serve as key indicators of overall processing performance due to the underlying impact on the Profile database as well as representing real world, daily processing requirements of any sized banking application.

3.1 Simulating Online Processing

For the Profile online transaction processing benchmark, a transaction generator is used to create complete network messages that simulate online transaction processing requests from clients. Since the Profile Application Server receives and responds to messages it receives from the message transport, it neither knows nor cares where the messages actually originate. The client transaction simulators generate messages, throttled on the client side and can continue to increase the load on the Profile Application Server side. The load generators scale horizontally, allowing for an unlimited load to be created and sent to the Profile Application Servers. Typically, two sub-networks are used to connect the workstations to separate network cards on the Profile server to ensure that the network does not limit transaction throughput.

The online test simulates online activity including monetary transactions (60%) and account inquiries (40%). This set of test utilized a Fidelity developed Java load tool to generate the online load. These transactions are passed to Profile Application Servers on the target machine through Profile's Message Transaction

Monitor (MTM). The MTM is connected to the clients using a TCP/IP connection. Transactions received by the server(s) are processed and responded with a reply message.

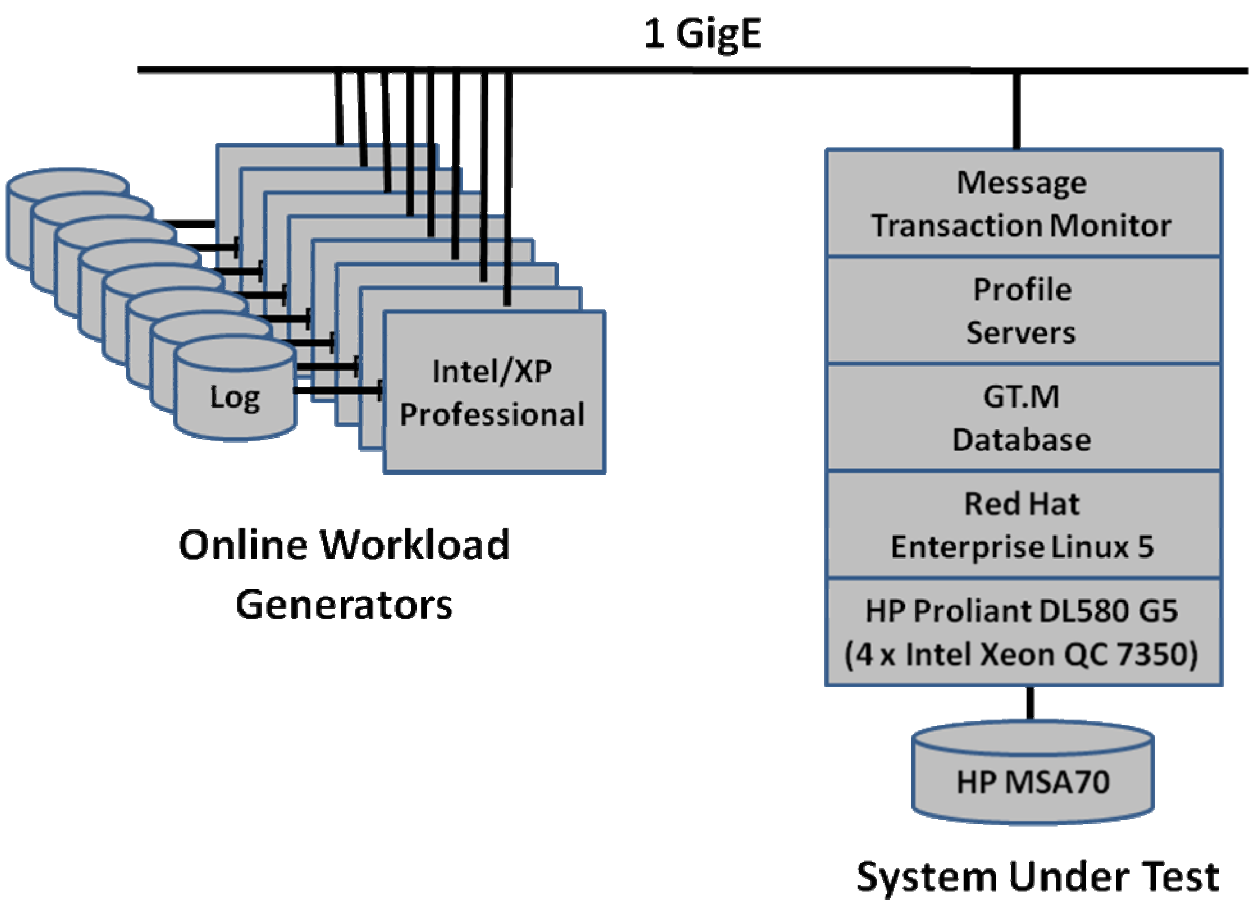


Figure 2

The client transaction generators create messages for random transactions against random accounts in the database according to a statistical distribution.

3.2 Simulating Scheduled Processing

The real-time scheduled processes are:

- Scheduled Transaction Posting,
- Interest Posting,
- Accrual and account update processing.

Scheduled transaction posting refers to a class of operations where a debit or credit transaction is received from a system file via an external source, such as a clearing house, or which may even be generated within Profile, such as a scheduled transfer of an amount from one account to another. The Profile benchmark includes three processes that perform real-time scheduled processing.

Scheduled Transaction Posting takes the transactions from a system file and processes them with multiple threads. An institution that receives and processes a daily feed with a large number of transactions from a clearinghouse would typically process these transactions using this functionality.

In the standard benchmark, Accrual and Account Update Processing accesses and updates <u>every</u> account in the database to calculate and update the accrued interest as well as other housekeeping functions for an account. Interest posting credits accrued interest to those accounts that are scheduled to receive interest on the posting date.

3.2.1 Scheduled Transaction Posting

General real-time Scheduled Transaction Posting is a multi-threaded function that processes financial transaction records delivered in a standard input/source table. For the benchmark, an input file containing random accounts was used to generate the transaction set. This test measures transaction processing from simulated sources, such as ACH or batch ATM, etc.

3.2.2 Interest Posting

Account Interest Posting is a multi-threaded real-time scheduled function that processes transactions to those accounts that are scheduled to post interest on the system date that the test is run. The account database was created with a significantly large number of accounts set to post interest transactions as part of this test.

3.2.3 Accruals and Account Housekeeping

Accrual and account update processing is a multi-threaded, real-time scheduled process that calculates one day's accrued interest and updates the applicable fields for every account record in the database. This information is summarized by currency code, product class, product group, product type, G/L set code and cost center. The summary information is one of the sources that feed the General Ledger. Because every account record in the database is processed, this function serves as a significant indicator of overall performance.

4. FIS PROFILE® - Benchmark Description and Results

The 25 million account benchmark uses:

- 1 load generator per Intel Xeon cpu-core with 10 client processes per load generator for a total of 16 load generators and 160 client processes.
- 8 MTMs with 8 PROFILE server processes each for a total of 64 PROFILE server processes.
- Each scheduled job spawns 16 heavy threaded processes dedicated to that job.

4.1 System Under Test (SUT)

4.1.1 Server

Vendor Model	1 x HP DL580 G5 (based on Intel Xeon Processor X7350) (Tigerton)
Processors	4 sockets, 4 cores/socket = 16 cores
Clock Speed	2.93 GHz
Memory	128 GB

4.1.2 Storage

Vendor Model	4 x HP StorageWorks MSA70 Direct Attached Storage with 25 HDDs each
Disk Controller	4 x HP Smart Array P800 Controllers
Disks	100 x 146GB 10K RPM Serial Attach SCSI (SAS) Drives (14.4 TB raw)
Fault Tolerance	Support for various RAID levels (0 / 1 / 5 / 6)

[RAID 6 with ADG: Allocates the equivalence of two parity drives across multiple drives and allows simultaneous write operations.

Distributed Data Guarding (RAID 5): Allocates parity data across multiple drives and allows simultaneous write operations.

Drive Mirroring (RAID 1 and 1+0 Striped Mirroring): Allocates half of the drive array to data and the other half to mirrored data, providing two copies of every file.]

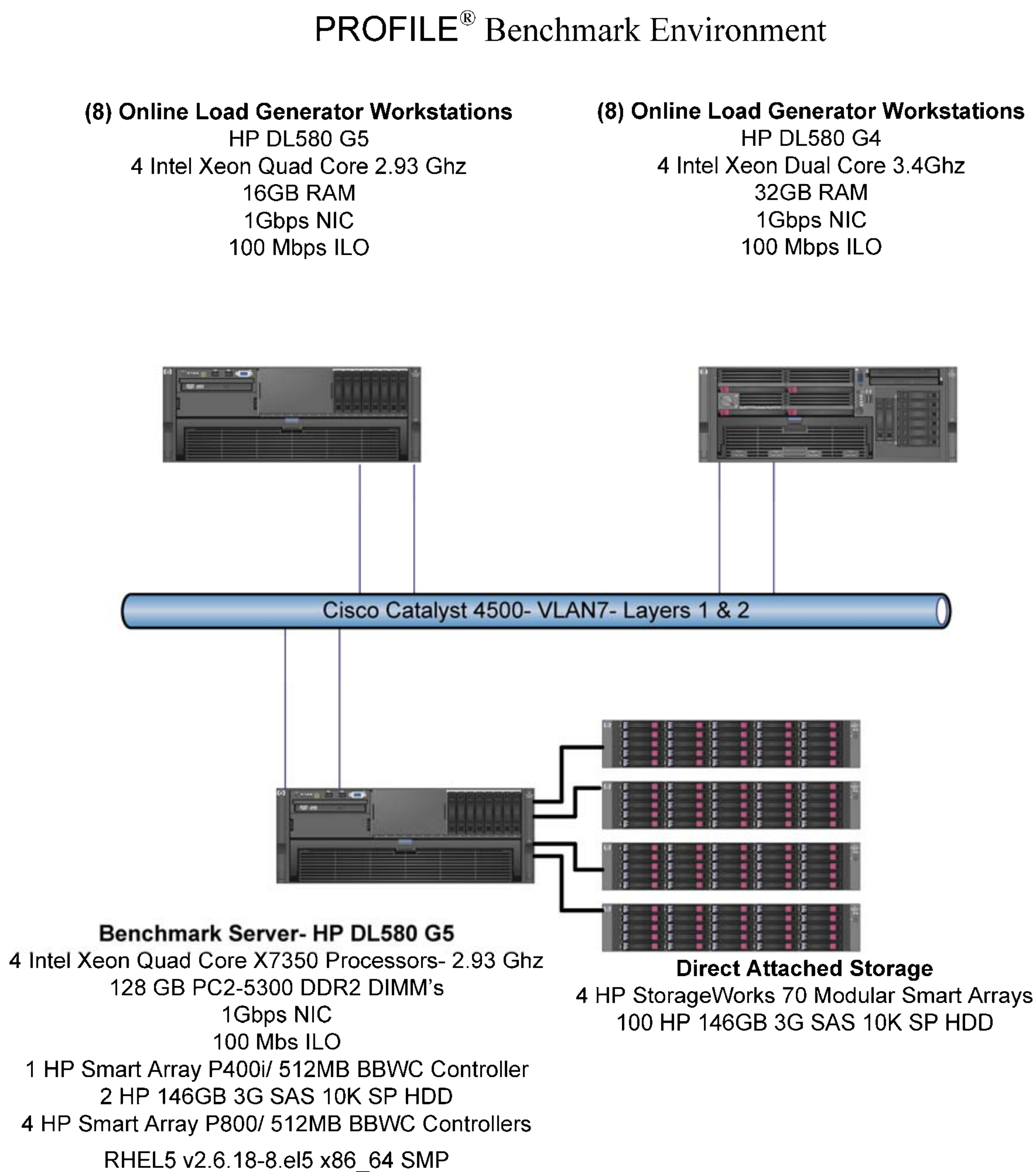

PROFILE® Benchmark Environment
(8) Online Load Generator Workstations
HP DL580 G5
4 Intel Xeon Quad Core 2.93 Ghz
16GB RAM
1Gbps NIC
100 Mbps ILO
(8) Online Load Generator Workstations
HP DL580 G4
4 Intel Xeon Dual Core 3.4Ghz
32GB RAM
1Gbps NIC
100 Mbps ILO
Cisco Catalyst 4500- VLAN7- Layers 1 & 2
Benchmark Server- HP DL580 G5
4 Intel Xeon Quad Core X7350 Processors- 2.93 Ghz
128 GB PC2-5300 DDR2 DIMM's
1Gbps NIC
100 Mbs ILO
1 HP Smart Array P400i/ 512MB BBWC Controller
2 HP 146GB 3G SAS 10K SP HDD
4 HP Smart Array P800/ 512MB BBWC Controllers
RHEL5 v2.6.18-8.el5 x86_64 SMP
Direct Attached Storage
4 HP StorageWorks 70 Modular Smart Arrays
100 HP 146GB 3G SAS 10K SP HDD

Figure 3

4.1.3 OS & System Software

Vendor	Red Hat
Version	Red Hat Enterprise Linux (RHEL) 5 - 64 bit
Kernel Version	Kernel v2.6.18-8.1.15.el5 SMP
File System	EXT3
Other	The RHEL High Availability software solution is not configured

4.1.4 Application, DB & Other Layered Software

Application	PROFILE v7
Database	GT.M v5.2-00B • Single database instance • Database Journaling enabled

4.2 Benchmark Database

4.2.1 Database Configuration

	Database Configuration		
	# of Customer Records	# of Account Records	# History Records
10 Million Accounts	10,000,000	10,000,000	50,000,000
25 Million Accounts	25,000,000	25,000,000	125,000,000

4.2.2 Database and Database Journal File Systems

Mount Point	RAID Level	Number of Spindles	Estimated Size	Enclosure Number
/fis14	1+0	12 x 146 GB	808 GB	4
/fis12	1+0	12 x 146 GB	808 GB	3

4.2.3 Database File Size (mounted in /fis14) – 10M Account DB

Database Name	Bytes
profile.acn	20,520,145,408
profile.cif	4,924,887,552
profile.dtx	4,104,081,920
profile.hist	12,312,113,664
profile.tmp	410,468,864
profile.ttx	820,871,680
profile.ubg	2,052,076,032
profile.xref	4,104,081,920
Total	49,248,727,040

4.2.4 Database File Size (mounted in /fis14) – 25M Account DB

Database Name	Bytes
profile.acn	51,300,262,400
profile.cif	12,722,516,480
profile.dtx	8,208,097,792
profile.hist	41,040,224,768
profile.tmp	1,026,073,088
profile.ttx	2,052,076,032
profile.ubg	5,130,088,960
profile.xref	10,260,107,776
Total	131,739,447,296

4.2.5 Database Product Distribution

				10M DB	25M DB
Product Class	Product Group	Product Type	% Distribution	Number of Accounts	Number of Accounts
Deposit	Certificate of Deposit (CD)	350	10%	1,000,000	2,500,000
Deposit	Regular Checking (DDA)	400	40%	4,000,000	10,000,000
Deposit	Regular Savings (SAV)	300	20%	2,000,000	5,000,000
Loan	Installment LN	500	10%	1,000,000	2,500,000
Loan	Fixed MTG LN	700	10%	1,000,000	2,500,000
Loan	Revolving Credit (RC)	600	10%	1,000,000	2,500,000
			Total DB Size	10,000,000	25,000,000

4.2.6 Database Transaction Volumes

	Database Transaction Volumes			
	Online Transactions	Transaction Inclearing	Interest Posting	Accrual Processing
10 Million Accounts	800,000	200,000	3,000,000	10,000,000
25 Million Accounts	800,000	300,000	7,500,000	25,000,000

4.3 Test Results

4.3.1 Online & Scheduled Process TPS

	TPS Rates				
	Online TPS	Online Response Time (seconds)	Transaction Posting TPS	Interest Posting TPS	Accruals Processing Accounts/Sec
10 Million Accounts	3,682	0.031	4,545	7,557	31,056
25 Million Accounts	3,269	0.036	1,068	7,591	25,853

4.3.2 Scheduled Process Runtimes

	Scheduled Process Runtimes		
	Avg. Transaction Posting Runtime (Seconds)	Avg. Interest Posting Runtime (Seconds)	Avg. Accruals Processing Runtime (Seconds)
10 Million Accounts	44	397	322
25 Million Accounts	281	988	967

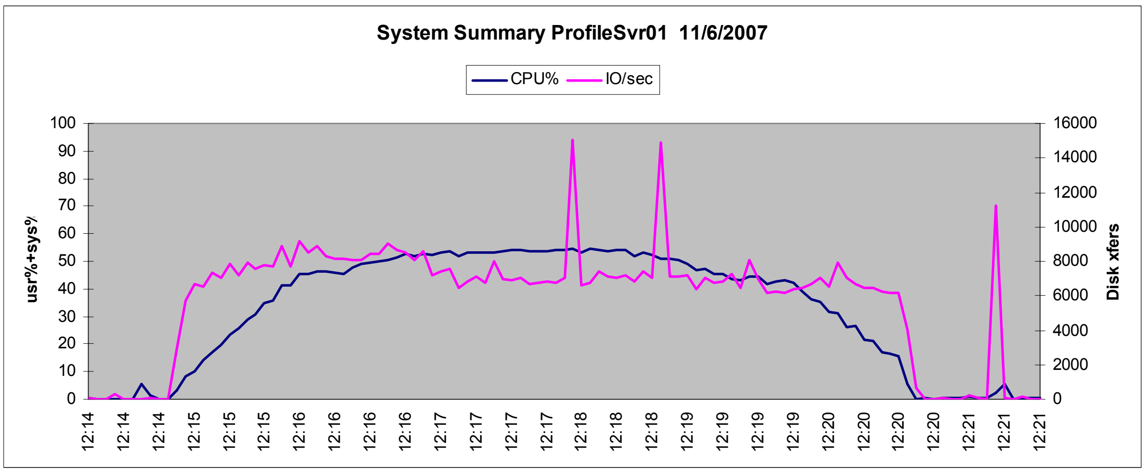
4.3.3 Average CPU Utilization (Steady State)

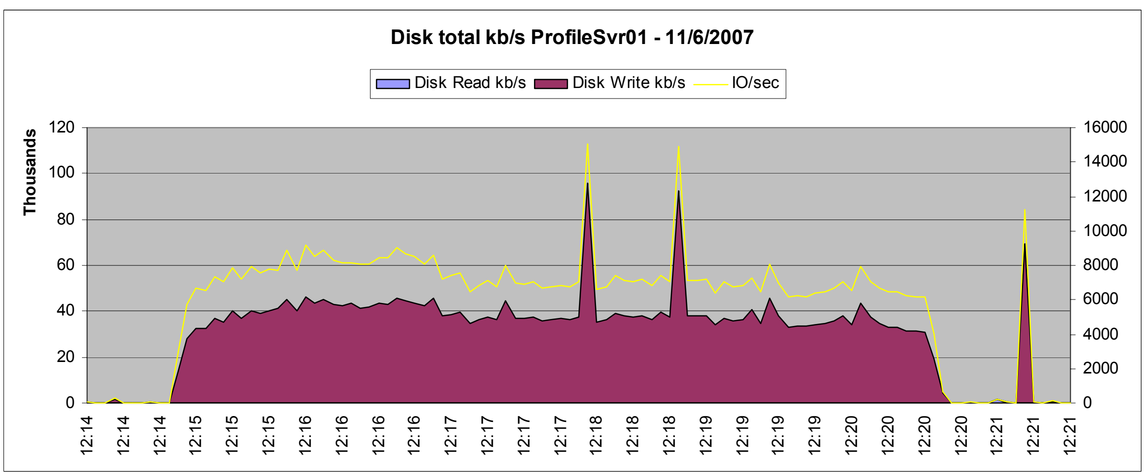
	CPU Utilization			
	On Line Simulation	Transaction Posting	Interest Posting	Accruals Processing
10 Million Accounts	54%	40%	75%	97%
25 Million Accounts	53%	19%	77%	70%

4.4 Resource Utilization Plots

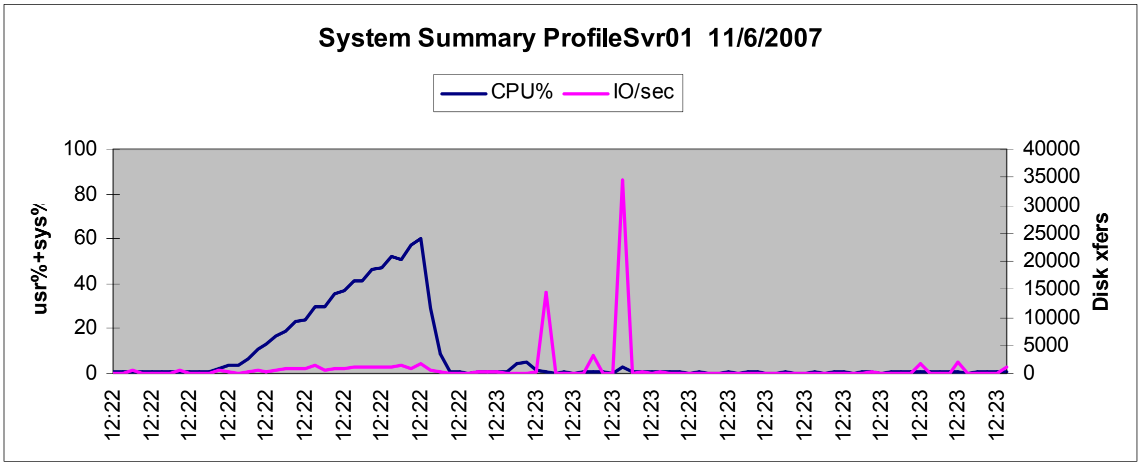
4.4.1 Resource Utilization (CPU and Disk) - 10 Million Account DB

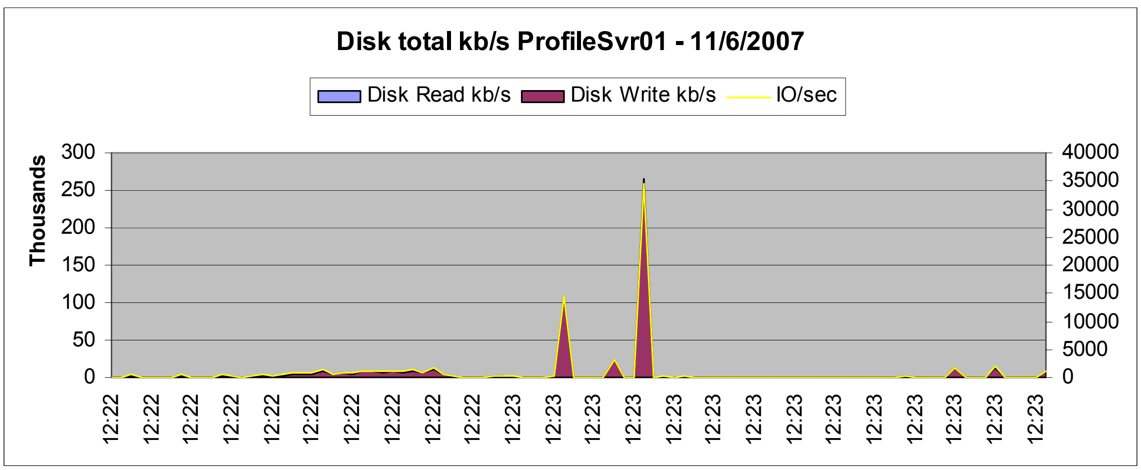
On Line Simulation



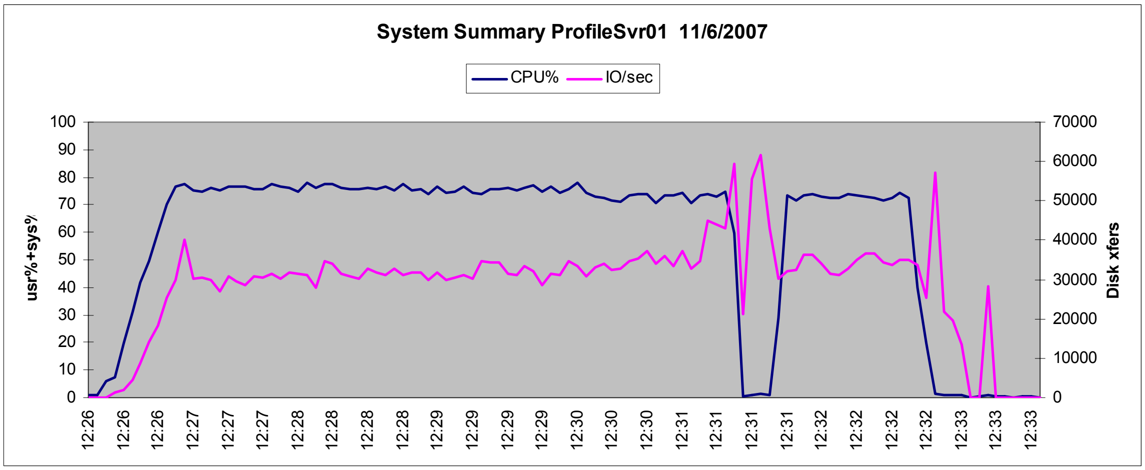


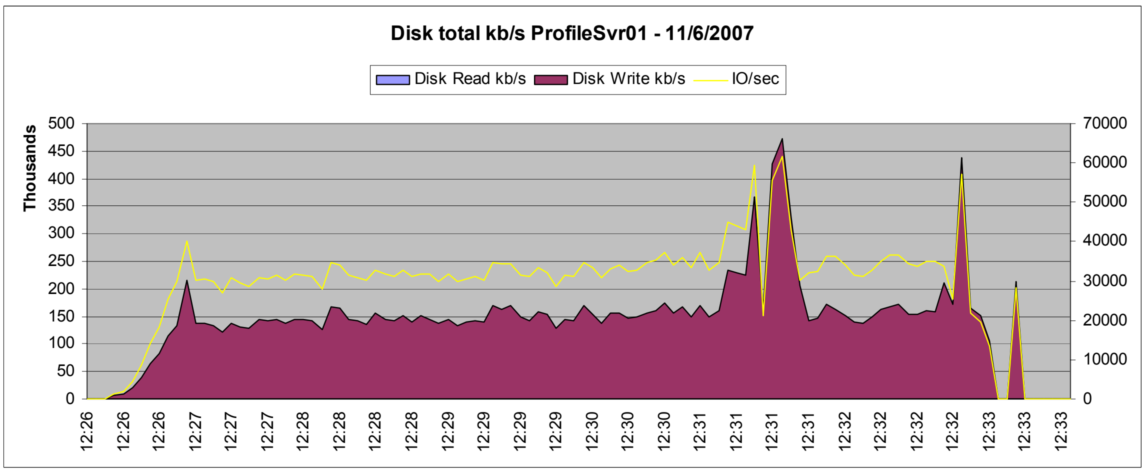
Scheduled Transaction Posting



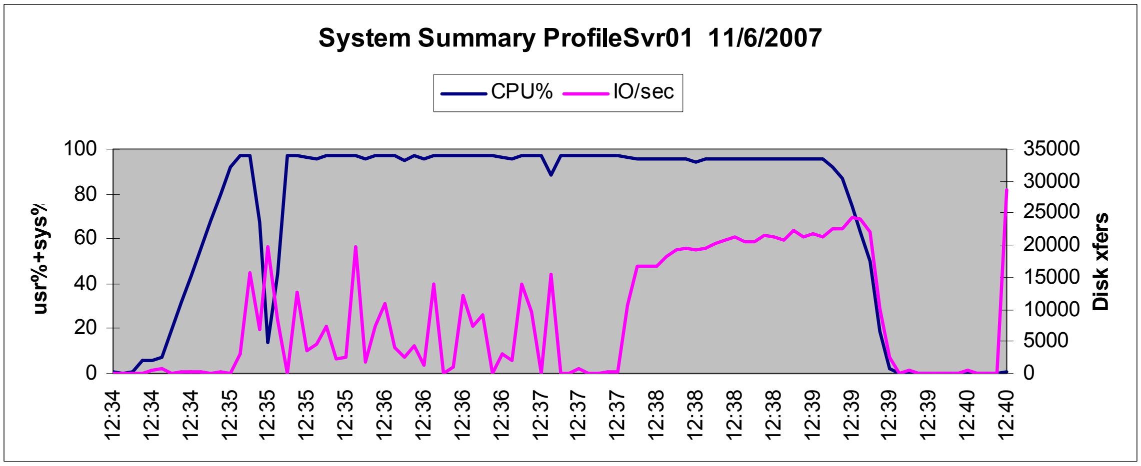


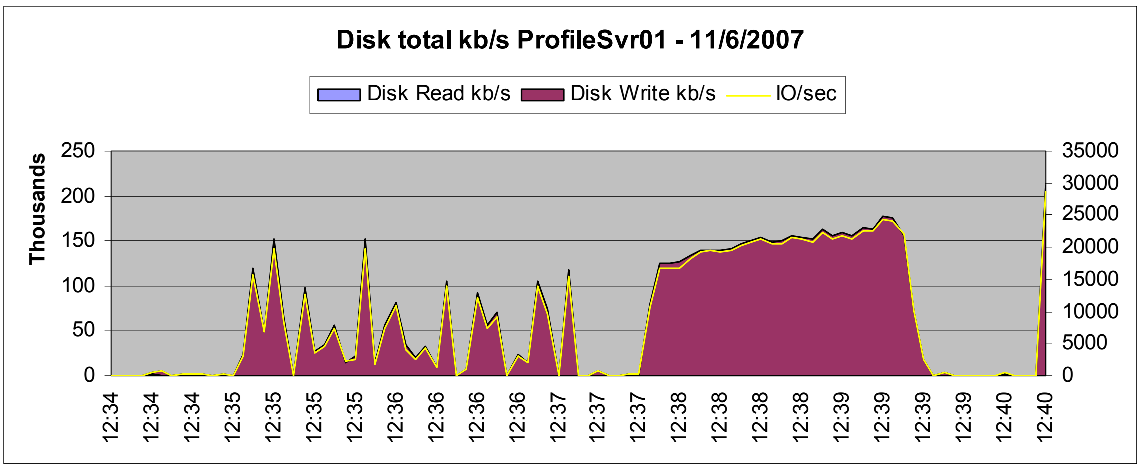
Scheduled Interest Posting





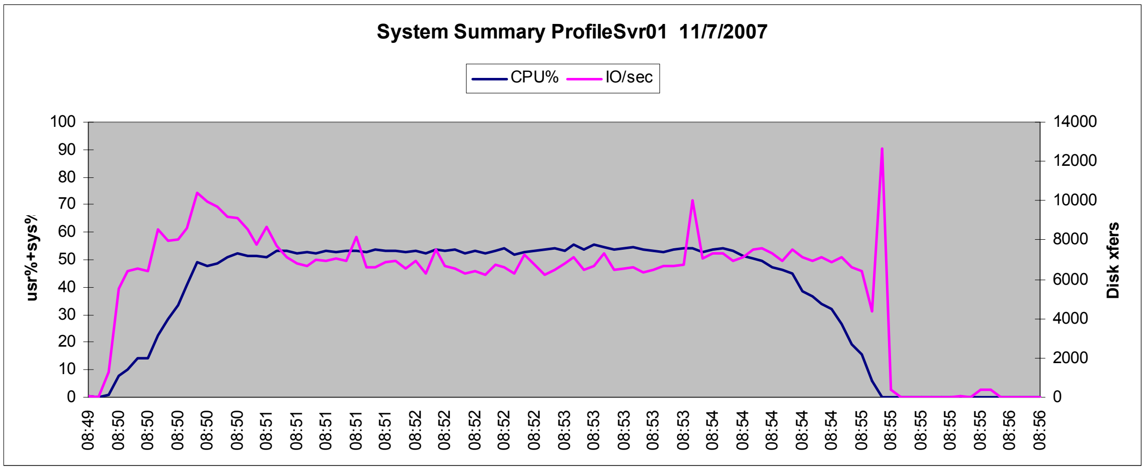
Scheduled Accruals Processing

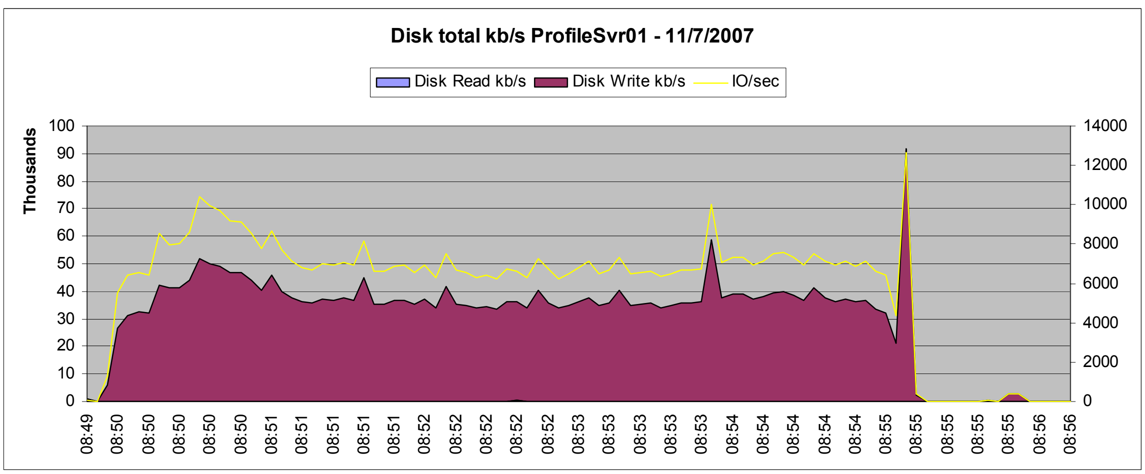




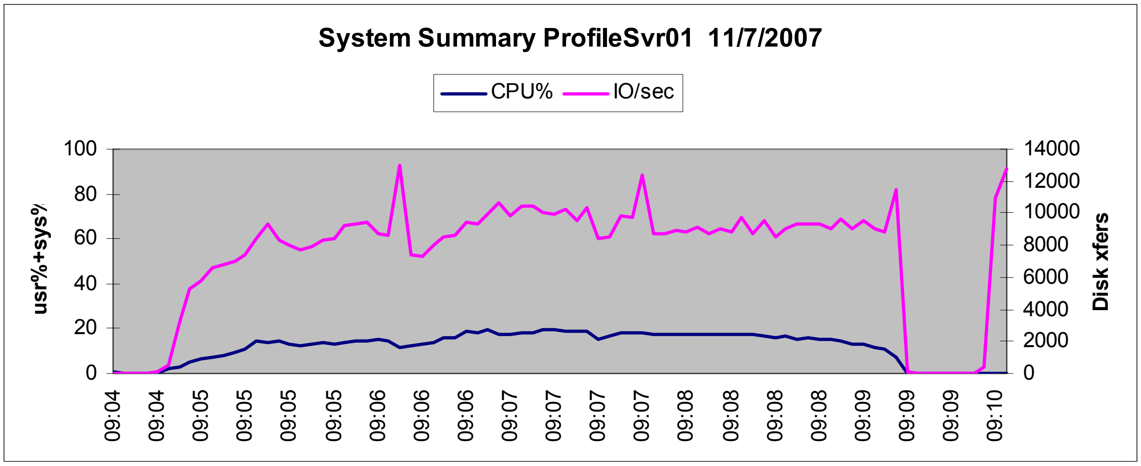
4.4.2 Resource Utilization (CPU and Disk) – 25 Million Account DB

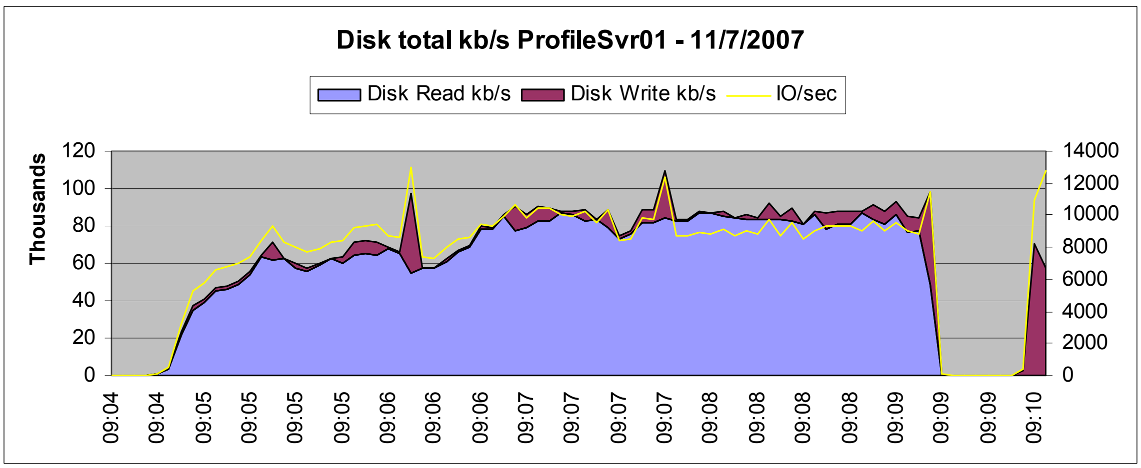
On Line Simulation



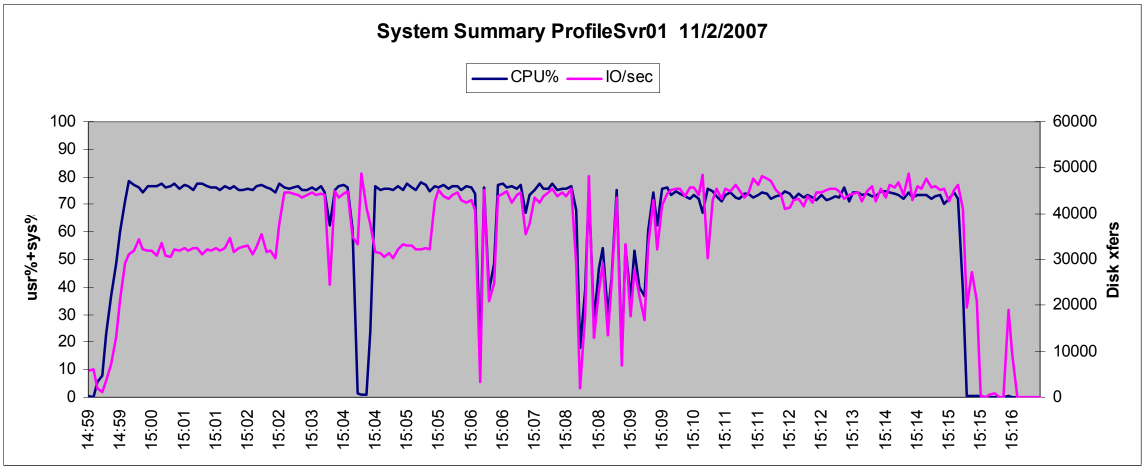


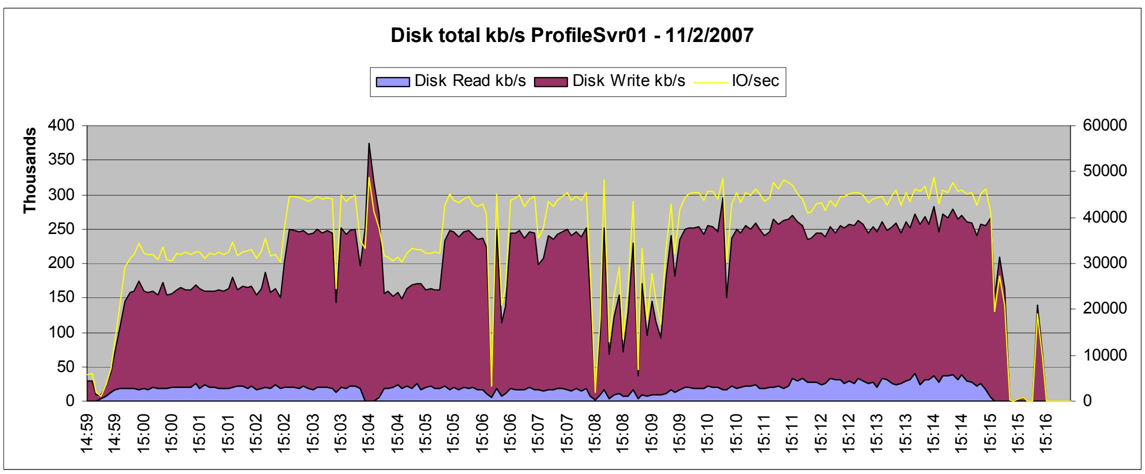
Scheduled Transaction Posting



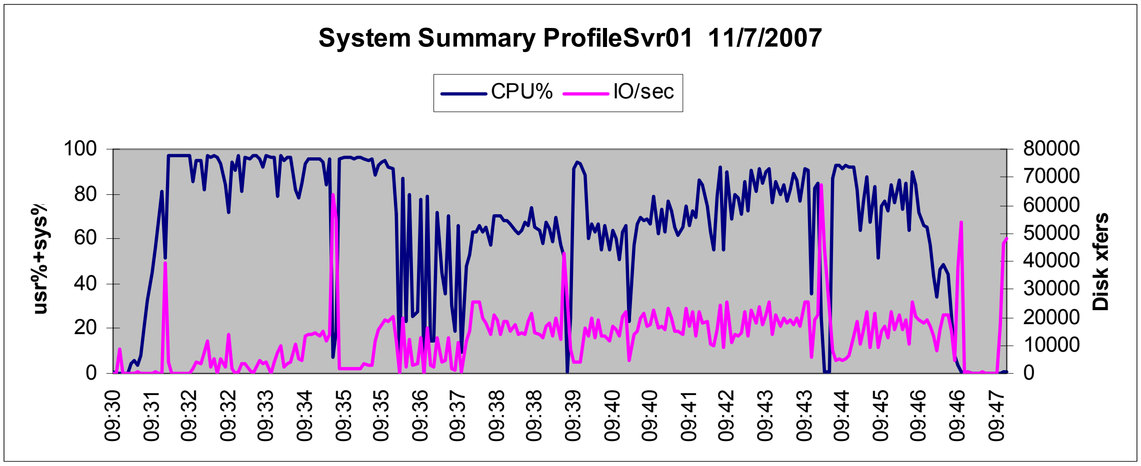


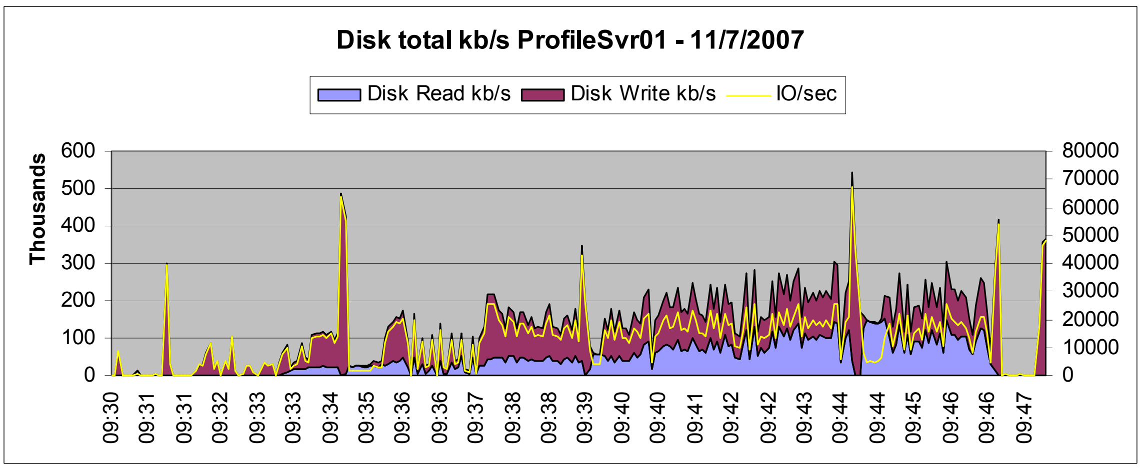
Scheduled Interest Posting





Scheduled Accruals Processing





5. Conclusions

The objectives of these benchmark tests were to evaluate the FIS Profile application configured on Red Hat Enterprise Linux (RHEL) running on commodity x86-64 processors; to utilize information gained through measurement and observation in order to apply tuning and modifications for optimizations; and to validate this technology stack as a viable, competitive solution.

The results indicate that despite very limited tuning and use of a less than optimal storage subsystem to deploy the database, FIS Profile was still able to achieve exceptional throughput for both on-line as well as scheduled transactions. Additionally, the processing tests from the scheduled transaction posting function produced spurious results, indicating that some anomalous activities may have been occurring concurrent with the test. Further analysis and testing is warranted and planned as part of the continued scaling up of the account database.

Following this test, our short-term and intermediate term objectives include further optimizing the existing components of the configuration as well as scaling up the account volume and real-time on-line and scheduled load to levels beyond the current tests and to utilize additional and/or different tuning parameters in order to maintain the exceptional price performance results from this current set of tests. Looking beyond the current 25 million accounts, we anticipate much larger database configurations, with an ultimate goal of attaining optimal price/performance for 100 million accounts. Additionally, we anticipate that the optimized configuration will also incorporate logical multi-site replication, with a highly available primary system replicating to a simulated disaster recovery system.

Finally, the out-of-the-box results attained in this set of tests, with 25 million accounts processing in a very modest system configuration provides a very promising outlook for much larger scale tests!