ORACLE E-BUSINESS BENCHMARK REV. 1.0

E-BUSINESS SUITE APPLICATIONS R12 (RUP 4) ORDER-TO-CASH (BATCH) BENCHMARK - USING ORACLE10*g* ON A CISCO UCS B200 M2 SERVER

As a global leader in e-business applications, Oracle is committed to delivering high performance solutions that meet our customers' expectations. Business software must deliver rich functionality with robust performance. This performance must be maintained at volumes that are representative of customer environments.

Oracle benchmarks demonstrate our software's performance characteristics for a range of processing volumes in a specific configuration. Customers and prospects can use this information to determine the software, hardware, and network configurations necessary to support their processing volumes.

The primary objective of our benchmarking effort is to provide as many data points as possible to support this important decision.

SUMMARY OF RESULTS

This batch benchmark test was run on a 12-core server.

Batch Workload			
50,000 Order/Inv. Lines	Threads	Time (Min)	Hourly Order Line Throughput
HVOP	12	1.75	1,714,285
Pick Release	12	5.90	508,474
Interface Trip Stop	12	0.62	4,838,709
Inventory	12	1.88	1,595,744
Auto Invoice	2	3.43	874,635
Revenue Recognition	2	0.48	6,250,000
Accounting	4	2.10	1,428,571
Totals:		16.16	185,643
Wall Clock Duration*		~18.77	159,829

Note that the hourly throughput numbers mentioned above are linear extrapolations. Many factors can influence performance and your results may differ.

* The "Wall Clock Duration" includes all of the job scheduling and management activity (parent process) as well as some idle intervals due to polling or waiting for all workers in a particular process to complete prior to kicking off the subsequent process. These intervals would not increase substantially, if at all, as the workload size is increased. Consequently, the throughput for larger workloads would converge towards the "Totals:" value.

BENCHMARK PROFILE

In August 2010, Oracle and Cisco conducted a benchmark in San Jose, CA to measure the batch performance of the Oracle E-Business Standard Benchmark processes in an environment running Oracle E-Business Suite R12 (RUP 4) with Oracle10*g*™ database (10.2.0.3) for the Linux operating system on a Cisco® UCS™ B200 M2 server configured with two six-core processors (12-cores total), running Red Hat® Enterprise Linux® 5.5 (64-bit) OS. A single EMC® CLARiiON® CX4 Model 240 disk array was used for storage.

The benchmark measured the Order Management batch business process hourly throughputs for a medium database model. Testing was conducted in a controlled environment with no other applications running. **The goal of this Benchmark was to obtain reference batch throughputs for Oracle E-Business Suite R12 Benchmark on a Cisco UCS server running Oracle Enterprise Linux or Red Hat Enterprise Linux 5.5.**

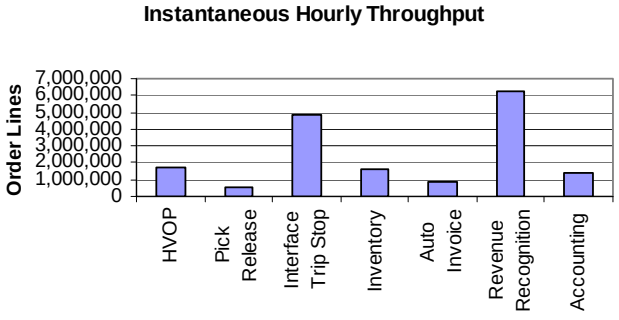

Figure 1: Oracle E-Business Order-to-Cash Batch Throughputs

BENCHMARK METHODOLOGY

E-Business Suite R12 Benchmark batch processes are initiated from a benchmark-provided SQL script.

The batch workloads were run as standard concurrent processes via the concurrent manager.

Figure 2 shows the configuration used for this benchmark run.

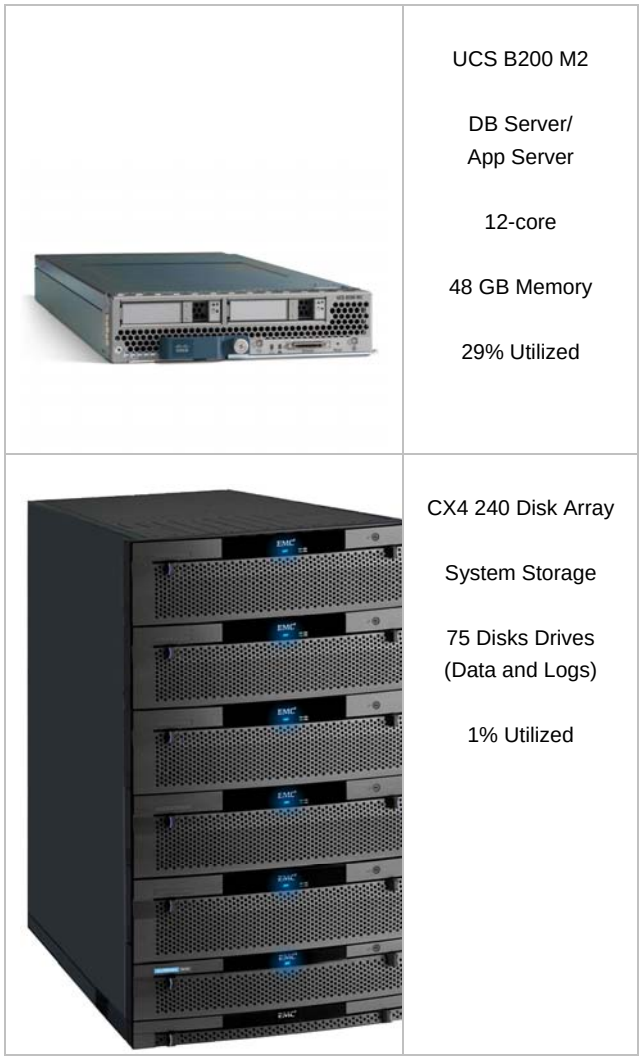

Figure 2: 2-Tier Configuration

This benchmark was run as a "Physical" 2-Tier configuration with a single machine hosting both the Database and Application server instances on a single OS image.

BENCHMARK BUSINESS PROCESSES

This E-Business Suite benchmark consists of a batch flow with seven metered processes.

Batch Order-to-Cash Processes

Business Process	Number of Threads Used
HVOP	12
Pick Release	12
Interface Trip Stop	12
Inventory	12
Auto Invoice	2
Revenue Recognition	2
Accounting Processes	4

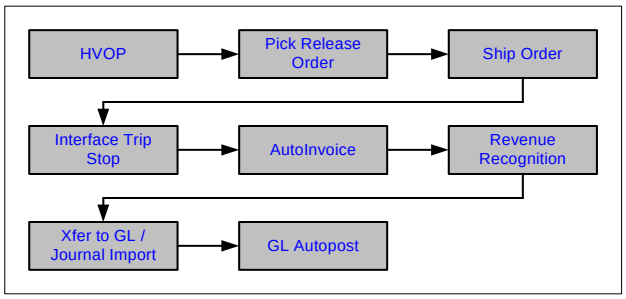

Figure 3: Order-to-Cash Process Flow

High Volume Order Processing (HVOP): The HVOP program processes orders by reading the rows from the Order Management Interface tables and converting the interface records into permanent order headers and their respective order lines. The orders are then booked and advanced to the shipping state.

Pick Release (and Ship Confirm): Pick Release finds and releases the eligible delivery lines that meet the release criteria, and creates move orders. The process of transacting move orders creates a reservation and determines the inventory source sub-inventory.
Ship Confirm is the process of confirming that items have shipped. When a delivery is ship-confirmed, Shipping Execution confirms that the delivery lines associated with the delivery have shipped.

Batch Order-to-Cash Processes Continued

Interface Trip Stop: The deliveries created in the previous step are then assigned to trips, which may involve multiple stops depending upon the shipping addresses of the deliveries. SRS has been modified to accept Organization code as a parameter and process the trip stops for the specified organization. Interface Trip Stop - SRS has also been enhanced to spawn multiple child processes to process trip stops in parallel. The parameter Stops per Batch is used to specify the number of stops to be processed by each thread of the Interface Trip Stop - SRS. Interface Trip Stop - SRS has also been enhanced to defer the Inventory Interface processes. In the E-Business Suite kit, this profile is set to Yes so that the Inventory Interface transactions are processed in the background by the Inventory transaction manager.

INV Material: The material transaction manager is configured to execute material transaction by periodic concurrent request submissions and by direct submission of multiple transaction managers via the benchmark SQL script. The execution interval is set to 5 minutes.

Auto-Invoice: The Auto-Invoice process is used to import invoices, credit memos, debit memos, and on–account credits. 'Receivables' ensures that the data imported is accurate and valid.

Revenue Recognition: Revenue Recognition program generates the revenue distribution records for the invoices and credit memos that use Invoicing and Accounting Rules. Accounting rules were assigned to recognize revenue over a 12-month accounting period. The Revenue Recognition program will create distribution records for the invoices and credit memos that are created in Receivables and imported using Auto-Invoice.

Transfer to General Ledger & Journal Import: The General Ledger Interface program transfers Receivables transaction accounting distributions to the general ledger interface table (GL_INTERFACE) and creates either detailed or summarized journal batches. "Receivables" creates un-posted journal entries in general ledger and executes Journal Import from Oracle General Ledger. It posts journal batches in Oracle General Ledger to update account balances.

General Ledger Auto-post: This posts journal batches to update the account balances of the detail and summary accounts. It can post actual budget or encumbrance journal batches.

BENCHMARK RESULTS

Batch Business Metrics		Achieved Output
Order to Cash		
Number of Order Lines Created/Booked		50,000
Number of Order Lines Picked		50,000
Number of Order Lines Ship Confirmed		50,000
Number of Order lines Interface Trip Stopped		50,000
Number of Invoice Headers Created		50,000
Number of Invoice Lines Created		100,000

Table 1: Batch Transactions Completed

50,000 order lines were processed in this test. Table 2 shows the processing time in minutes.

Batch Workload			
50,000 Order/Inv. Lines	**Threads**	**Time (Min)**	**Hourly Order Line Throughput**
HVOP	12	1.75	1,714,285
Pick Release	12	5.90	508,474
Interface Trip Stop	12	0.62	4,838,709
Inventory	12	1.88	1,595,744
Auto Invoice	2	3.43	874,635
Revenue Recognition	2	0.48	6,250,000
Accounting	4	2.10	1,428,571
Totals:		16.16	185,643
Wall Clock Duration*		~18.77	159,829

Table 2: Order-to-Cash Batch Performance

R12 Application changes, data model additions and test methodology improvements render direct comparison to previous Oracle E-Business release 11.5.10 and 11.5.9 results invalid.

SERVER PERFORMANCE

Figure 4 shows the average CPU utilization on the combined Database, App/Web and CM server. The value shown is the average across the processors (12 cores total).

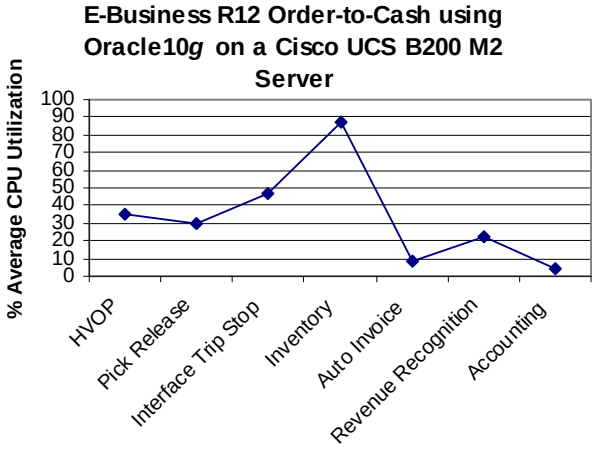

Figure 4: Average CPU Utilization

Online Workload		% User	% System	% I/O Wait	% Idle
HVOP		32.64	1.37	1.16	64.83
Pick Release		27.60	0.77	1.51	70.12
Interface Trip Stop		43.23	1.32	2.19	53.26
Inventory		84.26	1.93	1.51	12.31
Auto Invoice		7.22	0.20	0.87	91.71
Revenue Recognition		20.15	0.75	1.22	77.87
Accounting		3.33	0.16	0.57	95.94
Wall Clock Avg.		27.78	0.78	1.18	70.25

Table 3: Average CPU Utilization Breakout

Average GB Used	O-to-C
DB Server	26.5 GB

Table 4: Average Memory Utilization

I/O PERFORMANCE

An EMC CLARiiON CX4 Model 240 storage system equipped with 5 Disk Array Enclosures (75 disk drives total) was used for storage. The batch workload requires optimal I/O performance.

I/O Performance		Order-to-Cash
Transfers/Sec	Avg	37.61
	Peak	3096.60
Write KB/Sec	Avg	36.04
	Peak	3096.60
Read KB/Sec	Avg	1.56
	Peak	362.00
Avg Service Time (ms)	Avg	1.79
	Peak	40.00

Table 5: Average I/O Utilization Breakout

DATA COMPOSITION DESCRIPTION

Major data components for the model under test are summarized in the following table.

Application	Business Objects	Medium Model
TCA	Organizations	616,207
	Contacts	2,630,672
	Contact Points	2,073,332
	Accounts	609,422
	Account Sites	610,152
	Account Site Uses	1,065,726
Contracts	Contracts	0
Install Base	Instances	278,494
	Trackable Items	5
HR	Managers	400
	Employees	10,000
	Payroll Users	10,000
	Users	10,000
	Credit Card Entries	2,500,055
	Supplier(s)	5,000
Assets	Asset Categories	984
General Ledger	GL Code Combinations	93,417
Sales & Marketing	Resources	9,021
	Resource Groups	820
	Sales Leads	1,217,062
	Campaigns	1
	Sales Territories	8,200

Table 6: Data Composition

APPLICATION TUNING

1. Patches applied to Oracle E-Business Applications R12 (RUP 4) as reported by Oracle Applications Manager:

6645909

2. Increased Standard Manager
Work Shifts, Process=24, Sleep Time=10 sec.

3. Increased Inventory Manager
Work Shifts, Process=12, Sleep Time=10 sec.

4. Increased Inventory Workers
operations/welcome
Inventory, Visions Operations (USA)
Setup : Transactions
Interface Managers
V1 Visions Operations
Material transaction
Worker Rows 500 (default 200)

5. Decreased the poll interval and worker wait time for the Revenue Recognition and Accounting job steps.

DATABASE TUNING

1. Patches applied to Oracle 10g (10.2.0.3) as reported by opatch lsinventory:

4054238
4247037
4592596
4733582
4882839
4898580
4932527
4966417
4967236
5066528
5103126
5206570
5240469
5392772
5455880
5477912
5556081
5557962
5668025
5746153
5889331
5959099
6079591
6121183
6121242
6121243
6121244
6121245
6121246
6121247
6121248
6121249
6121250
6121257
6121258
6121260
6121261
6121263
6121264
6121266
6121267
6121268
6317866

2. The following index was modified.

```
  alter index
APPLSYS.WF_ITEM_ACTIVITY_STATUSES_PK
rebuild partition WF_ITEM48 compute statistics ;
  alter index
APPLSYS.WF_ITEM_ACTIVITY_STATUSES_PK shrink
space cascade ;
```

3. The following indexes were dropped.

```
  drop index APPLSYS.WF_PROCESS_ACTIVITIES_N2 ;
  drop index AR.RA_INTERFACE_LINES_C01 ;
  drop index INV.MTL_MATERIAL_TRANS_TEMP_N1 ;
  drop index
INV.MTL_MATERIAL_TRANS_TEMP_N14 ;
  drop index INV.MTL_MATERIAL_TRANS_TEMP_U1;
  drop index
INV.MTL_MATERIAL_TRANSACTIONS_U1;
  drop index INV.MTL_RESERVATIONS_U1;
  drop index
WSH.WSH_DELIVERY_ASSIGNMENTS_N3 ;
  drop index WSH.WSH_DELIVERY_DETAILS_N8 ;
```

4. The following indexes were created.

APPLSYS.FND_PROFILE_OPTION_VALUES (APPLICATION_ID, PROFILE_OPTION_ID, LEVEL_ID, LEVEL_VALUE, LEVEL_VALUE_APPLICATION_ID, PROFILE_OPTION_VALUE) COMPRESS 3

APPLSYS.WF_PROCESS_ACTIVITIES (PROCESS_ITEM_TYPE,PROCESS_NAME,PROCESS_VERSION,START_END, INSTANCE_ID)

AR.RA_CUST_TRX_LINE_GL_DIST_ALL (CUSTOMER_TRX_ID, ACCOUNT_CLASS, LATEST_REC_FLAG, CODE_COMBINATION_ID, ACCOUNT_SET_FLAG)

AR.RA_CUSTOMER_TRX_LINES_ALL (CUSTOMER_TRX_ID, LINE_TYPE)

AR.RA_CUSTOMER_TRX_LINES_ALL (INTERFACE_LINE_ATTRIBUTE6, INTERFACE_LINE_CONTEXT)

AR.RA_INTERFACE_LINES_ALL(INTERFACE_LINE_ATTRIBUTE6, INTERFACE_LINE_CONTEXT, LINE_TYPE)

GL.GL_LEDGERS (LEDGER_ID, CURRENCY_CODE)

GL.GL_LEDGERS (CURRENCY_CODE, OBJECT_TYPE_CODE, LE_LEDGER_TYPE_CODE)

GL.GL_PERIOD_STATUSES (APPLICATION_ID, SET_OF_BOOKS_ID, ADJUSTMENT_PERIOD_FLAG)

GL.GL_PERIOD_STATUSES (APPLICATION_ID, SET_OF_BOOKS_ID, END_DATE, START_DATE)

INV.MTL_MATERIAL_TRANSACTIONS (TRANSACTION_ID) GLOBAL PARTITION BY HASH (TRANSACTION_ID) PARTITIONS 128

INV.MTL_MATERIAL_TRANSACTIONS_TEMP (MOVE_ORDER_LINE_ID) GLOBAL PARTITION BY HASH (MOVE_ORDER_LINE_ID) PARTITIONS 128

INV.MTL_MATERIAL_TRANSACTIONS_TEMP (TRANSACTION_HEADER_ID, PROCESS_FLAG, TRANSACTION_BATCH_ID, ERROR_CODE)

INV.MTL_MATERIAL_TRANSACTIONS_TEMP (TRANSACTION_TEMP_ID) GLOBAL PARTITION BY HASH (TRANSACTION_TEMP_ID) PARTITIONS 128

INV.MTL_RESERVATIONS(RESERVATION_ID) GLOBAL PARTITION BY HASH (RESERVATION_ID) PARTITIONS 128

INV.MTL_SYSTEM_ITEMS_B (ORGANIZATION_ID, UPPER(SEGMENT1), CUSTOMER_ORDER_ENABLED_FLAG)

INV.MTL_UOM_CONVERSIONS (UOM_CODE)

INV.MTL_UOM_CONVERSIONS (UOM_CODE, DISABLE_DATE)

MRP.MRP_ATP_SCHEDULE_TEMP (SESSION_ID, STATUS_FLAG, NVL(ATO_MODEL_LINE_ID, ORDER_LINE_ID))

WSH.WSH_DELIVERY_ASSIGNMENTS (DELIVERY_DETAIL_ID, DELIVERY_ID, TYPE) PCTFREE 10 INITRANS 11 MAXTRANS 255
COMPUTE STATISTICS STORAGE(INITIAL 131072 NEXT 131072 MINEXTENTS 1 MAXEXTENTS 2147483645 PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT)

WSH.WSH_DELIVERY_DETAILS (MOVE_ORDER_LINE_ID) GLOBAL PARTITION BY HASH (MOVE_ORDER_LINE_ID) PARTITIONS 128

WSH.WSH_DELIVERY_LEGS (DELIVERY_ID, PICK_UP_STOP_ID, DROP_OFF_STOP_ID, DELIVERY_LEG_ID) GLOBAL PARTITION BY HASH (DELIVERY_ID) PARTITIONS 128

WSH.WSH_PR_WORKERS (NVL(INVENTORY_ITEM_ID,-99))

WSH.WSH_PR_WORKERS (DELIVERY_ID)

XLA.XLA_TRANSACTION_ENTITIES (LEDGER_ID, ENTITY_CODE, NVL(SOURCE_ID_INT_1, (-99)), NVL(SOURCE_ID_INT_2, (-99)), NVL(SOURCE_ID_INT_3, (-99)), NVL(SOURCE_ID_INT_3, (-99)),
NVL(SOURCE_ID_INT_4, (-99)), NVL(SOURCE_ID_CHAR_1,''), NVL(SOURCE_ID_CHAR_2,''), NVL(SOURCE_ID_CHAR_3,''), NVL(SOURCE_ID_CHAR_4,'')) local

XLA.XLA_EVENTS_GT (SOURCE_ID_INT_1, APPLICATION_ID, PROCESS_STATUS_CODE, EVENT_TYPE_CODE)

XLA.XLA_AE_LINES_GT (LEDGER_ID, AE_HEADER_ID, HEADER_NUM, BALANCE_TYPE_CODE)

5. Gather stats as follows:

```
  exec fnd_stats.gather_table_stats ('APPLSYS',
'FND_CONCURRENT_REQUESTS', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('APPLSYS',
'FND_PROFILE_OPTION_VALUES', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('APPLSYS',
'WF_ACTIVITIES', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('APPLSYS',
'WF_DEFERRED_TABLE_M', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('APPLSYS',
'WF_ITEM_ACTIVITY_STATUSES', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('APPLSYS',
'WF_ITEM_ATTRIBUTE_VALUES', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('APPLSYS',
'WF_ITEMS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('APPLSYS',
'WF_PROCESS_ACTIVITIES', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('AR',
'RA_CUST_TRX_LINE_GL_DIST_ALL', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('AR',
'RA_CUSTOMER_TRX_LINES_ALL', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('AR',
'RA_INTERFACE_LINES_ALL', 100, cascade=>TRUE );
  exec fnd_stats.gather_table_stats ('AR',
'AR_XLA_LINES_EXTRACT', 100, cascade=>TRUE );
  exec fnd_stats.gather_table_stats ('ASO',
'ASO_ORDER_FEEDBACK_T', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('GL',
'GL_BC_PACKETS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('GL',
'GL_CODE_COMBINATIONS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('GL',
'GL_CONSOLIDATION_HISTORY', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('GL',
'GL_JE_BATCHES', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('GL', 'GL_LEDGERS',
100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('GL',
'GL_PERIOD_STATUSES', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'MLOG$_MTL_MATERIAL_TRANSAC', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'MTL_MATERIAL_TRANSACTIONS', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'MTL_MATERIAL_TRANSACTIONS_TEMP', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'MTL_ONHAND_QUANTITIES_DETAIL', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'MTL_RESERVATIONS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'MTL_SERIAL_NUMBERS_TEMP', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'MTL_SYSTEM_ITEMS_B', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'MTL_TRANSACTIONS_INTERFACE', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'MTL_TXN_REQUEST_LINES', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'MTL_UOM_CONVERSIONS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('INV',
'ORG_ACCT_PERIODS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('MRP',
'MRP_ATP_SCHEDULE_TEMP', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('ONT',
'OE_HEADERS_IFACE_ALL', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('ONT',
'OE_ORDER_HEADERS_ALL', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('ONT',
'OE_ORDER_LINES_ALL', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('WMS',
'WMS_TRANSACTIONS_TEMP', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('WSH',
'MLOG$_WSH_DELIVERY_DETAILS', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('WSH',
'WSH_DELIVERY_ASSIGNMENTS', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('WSH',
'WSH_DELIVERY_DETAILS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('WSH',
'WSH_DELIVERY_LEGS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('WSH',
'WSH_PR_WORKERS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('WSH',
'WSH_WMS_SYNC_TMP', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('XLA',
'XLA_AE_LINES', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('XLA',
'XLA_AE_LINES_GT', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('XLA',
'XLA_DISTRIBUTION_LINKS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('XLA',
'XLA_EVENTS_GT', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('ZX',
'ZX_LINES_DET_FACTORS', 100, cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('ZX',
'ZX_TRANSACTION_LINES_GT', 100,
cascade=>TRUE ) ;
  exec fnd_stats.gather_table_stats ('ZX',
'ZX_VALIDATION_ERRORS_GT', 100,
cascade=>TRUE ) ;
```

OPERATING SYSTEM TUNING

DATABASE OPERATING SYSTEM TUNING

1. The following additional Kernel parameters were automatically setup during boot
via the /etc/sysctl.conf file:

```
fs.file-max = 6553600
kernel.core_uses_pid = 1
kernel.sem = 256 32000 100 142
kernel.shmall = 4294967296
kernel.shmmax = 68719476736
kernel.shmmni = 4096
kernel.msgmax = 65536
kernel.msgmnb = 65536
kernel.msgmni = 2878
kernel.sysrq = 0
net.ipv4.conf.default.rp_filter = 1
net.ipv4.conf.default.accept_source_route = 0
net.ipv4.ip_forward = 0
net.ipv4.ip_local_port_range = 1024 65000
net.ipv4.tcp_syncookies = 0
net.ipv4.tcp_wmem = 262144 262144 262144
net.ipv4.tcp_rmem = 4194304 4194304 4194304
net.core.rmem_default = 4194304
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 262144
vm.nr_hugepages=12300
```

2. The following limits were modified via the /etc/security/limits.conf file:

```
* hard nofile 131072
* soft nofile 131072
* hard nproc 131072
* soft nproc 131072
oracle - memlock 100000000
```

3. Hugepages were enabled for the database instance.

BENCHMARK ENVIRONMENT

HARDWARE CONFIGURATION

A Cisco® UCS™ B200 M2 server was used for the database server. It was equipped with the following:

- 2 × 3.33 GHz Intel® Xeon™ Six-Core X5680 processors with Hyper-Threading enabled (2-processors, 12-cores, 24-threads total), each with 12 MB of Level 3 cache
- 48 Gigabytes of Memory (~28.2 GB peak)
- 2 × 73 GB internal disk drives attached to an embedded LSI SAS1064E PCI Express 3-Gbps SAS Controller
- 1 × EMC CLARiiON CX4 Model 240 disk array attached to a Cisco UCS M71KR-Q CAN 4-Gbps Fibre Channel Controller for data and logs
- ~33 TB raw disk space available for allocation (75 × 450 GB)
- Approximately 393 GB of RAID 0 storage configured for this benchmark (data and logs)

SOFTWARE VERSIONS

Oracle's E-Business Suite (E-Business Suite Kit) R12 (RUP 4)

Oracle10*g*™ 10.2.0.3 (64-bit)

Red Hat® Enterprise Linux® 5 update 5 (64-bit) on the database/application/web server.

Glossary and Acronyms:

ATP	Available to Promise
BEE	Batch Element Entries
HVOP	High Volume Order Processing
OASB	Oracle Applications Standard Benchmark
RAC	**R**eal **A**pplications **C**lusters

Oracle
Applications Performance & Benchmarks
500 Oracle Parkway
Redwood Shores, California 94065
Tel 650/506-7000
Fax 650/506-7000
Email eBSkit_us@oracle.com
World Wide Web http://www.oracle.com

The results published in this report have been independently reviewed and audited by:

INFO SIZING