

E-BUSINESS SUITE APPLICATIONS R12 (RUP 4) ORDER-TO-CASH (BATCH) BENCHMARK - USING ORACLE10g ON AN IBM SYSTEM x3550 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 an 8-core server.

Batch Workload				
50,000 Lines	Order/Inv.	Threads	Time (Min)	Hourly Order Line Throughput
HVOP		10	2.25	1,333,333
Pick Release		10	8.12	369,458
Interface Trip Stop		10	1.1	2,727,273
Inventory		10	1.98	1,515,152
Auto Invoice		2	3.5	857,143
Revenue Recognition		2	2	1,500,000
Accounting		4	2	1,500,000
Totals:			20.95	143,198
Wall Clock Duration*			30	100,000

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 September 2009, Oracle and IBM conducted a benchmark in San Mateo, 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 Oracle10g™ database (10.2.0.3) for the Linux® operating system on an IBM® System x3550 M2 server configured with two 4-core Intel® Xeon® 5500 processors (8-cores total), running Red Hat® Enterprise Linux® 5 (64-bit) OS. A single IBM Storage System™ DS4700 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 an IBM System x™ server running Linux.**

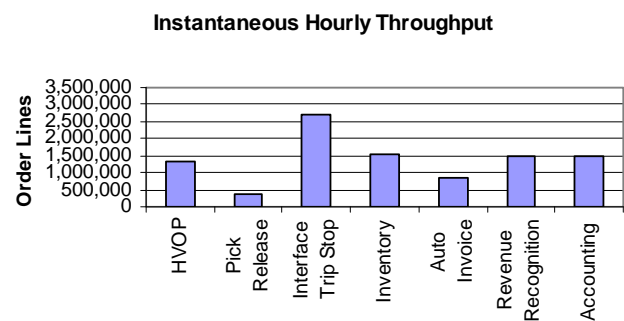


Figure 1: Oracle E-Business Order-to-Cash Batch Throughputs

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.



	<p>System x3550 M2 DB Server/App Server 8-core 64 GB</p> <p>34% Utilized</p>
	<p>DS4700 System Storage 5 Drawers – Data 1 Drawer – Logs</p> <p>1% Utilized</p>

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	8, 10
Pick Release	8, 10
Interface Trip Stop	8, 10
Inventory	8, 10
Auto Invoice	2
Revenue Recognition	2
Accounting Processes	4

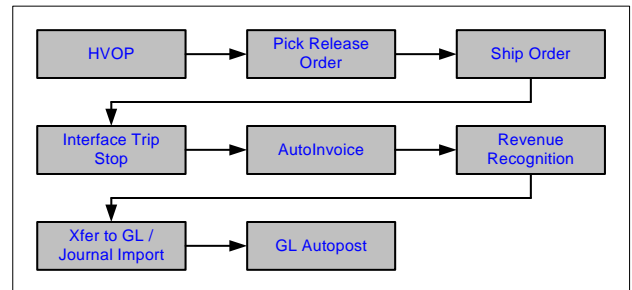


Figure 3: Order-to-Cash Process Flow

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.

BENCHMARK RESULTS

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.

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. Tables 2 and 3 show the processing time in minutes.

Batch Workload				
50,000 Lines	Order/Inv.	Threads	Time (Min)	Hourly Order Line Throughput
		10	2.25	1,333,333
	HVOP	10	8.12	369,458
	Pick Release	10	1.1	2,727,273
	Interface Trip Stop	10	1.98	1,515,152
	Inventory	2	3.5	857,143
	Auto Invoice	2	2	1,500,000
	Revenue Recognition	4	2	1,500,000
	Accounting			
	Totals:		20.95	143,198
	Wall Clock Duration*		30	100,000

Table 2: Order-to-Cash (10 Worker) Batch Performance

50,000 Lines	Order/Inv.	Threads	Time (Min)	Hourly Order Line Throughput
		8	2.17	1,382,488
	HVOP	8	7.97	376,412
	Pick Release	8	1	3,000,000
	Interface Trip Stop	8	2.58	1,162,791
	Inventory	2	3.47	864,553
	Auto Invoice	2	2	1,500,000
	Revenue Recognition	4	2	1,500,000
	Accounting			
	Totals:		21.19	141,576
	Wall Clock Duration*		31	96,774

Table 3: Order-to-Cash (8 Worker) 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 server. The value shown is the average across the processors (8 cores total).

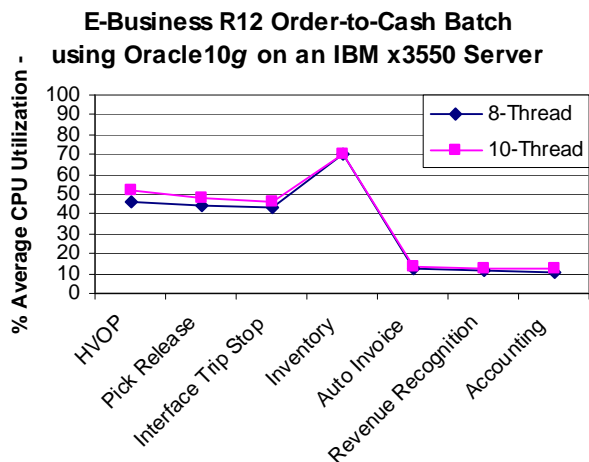


Figure 4: Average DB/App/Web CPU Utilization

Workload	% User	% System	% Idle	% I/O Wait
HVOP	34.64	5.86	53.64	6.14
Pick Release	36.49	3.88	55.40	4.12
Interface Trip Stop	33.67	5.17	56.83	4.50
Inventory	59.11	6.94	29.83	4.22
Auto Invoice	10.65	0.76	87.18	1.47
Revenue Recognition	8.86	1.00	88.57	1.57
Accounting	8.17	1.17	89.33	1.33
Wall Clock Avg.	27.63	2.97	66.39	2.98

Table 5: Average CPU Utilization Breakout (8 Workers)

Average GB Used	O-to-C	Caching
DB/App/Web Server	21 GB	~12 GB

Table 6: Average Memory Utilization

Workload	% User	% System	% Idle	% I/O Wait
HVOP	39.00	5.87	48.53	6.67
Pick Release	40.04	3.96	51.54	4.10
Interface Trip Stop	35.67	5.50	53.83	5.00
Inventory	59.05	6.68	29.47	4.63
Auto Invoice	11.63	0.68	86.11	1.53
Revenue Recognition	9.17	1.00	87.50	1.50
Accounting	9.17	1.00	87.50	1.50
Wall Clock Avg.	31.16	3.10	62.30	3.25

Table 4: Average CPU Utilization Breakout (10 Workers)

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 7: Data Composition

PATCHES

The following patches were applied to the benchmark environment on top of Oracle E-Business Applications R12 (RUP 4).

Patch for Bug 7142581.

Patch 8134220

Patch for Bug causing Oracle error 04030

1. Check category assignments

The category assignment for the items should be assigned to IC1. Example items EBSITEMS100001, EBSITEMS100002, etc.

To check:

- a) Navigate to Inventory: Setup: Items: Categories: Category Sets.
- b) Query the category set 'ICS'.
- c) Click on Assign.
- d) Query for the a few items, and make sure it's assigned to 'IC1', or query for category 'IC2', 'IC3', etc, these should not have items assigned.

2. Accounting Program error: xla_accounting_pkg.ValidateAAD. ORA-0000: normal, successful completion. Related bug 6849494

Make sure to validate the package. Validate the uncompiled AADs in Receivables.

There are two ways to compile the AADs in SLA:

Solution 1:

```
sqlplus apps/apps

alter session set plsql_optimize_level =
1
/
alter package APPS.XLA_XXXXXXXXX_PKG
compile
/
```

Solution 2:

- 1) Navigate to Receivables (or whichever product module) --> Accounting Setups ---> Application Accounting Definitions ---> Query for the AAD ---> Click on the Compile Button. This method, however, will only let you

validate one AAD at a time.

2) Submit the concurrent request 'Validate Application Accounting Definitions', this will allow to you compile all the AADs in AR at once.

3. Setup changes to ensure that Pick Release will spawn children processes and automatically apply the process deliveries, and submit Ship Confirm

The submit_O2C.sql program was modified to remove the code that calls Auto-create deliveries and Ship Confirm. Pick Release process will do following:

1. Spawn child process to pick release the eligible details.
2. Auto-create the deliveries for the Staged delivery details in the parent process.
3. Spawn child process to ship confirm the deliveries created above.

Setup changes:

For the new behavior for O2C for R12, apply the following setup changes:

- 1) Pick the Responsibility: Order Management Super User, Vision Operations (USA).
- 2) Navigate to Shipping > Setup > Shipping Parameters > select "V1"
- 3) go to the Pick Release tab and ensure "Print Pick Slip" is set to "At the End"
- 4) Navigate to Shipping > Setup > Picking > Define Pick Slip Grouping Rules
- 5) Ensure under Rule Name "Order Number", it shows "Group by" set to "Order Number".
- 6) Navigate to Shipping > Setup > Picking > Define Release Rules Form
- 7) Search for Rule "EBS_release_rule1"
 - a) go to the Inventory tab. Under Pick Slip Grouping Rule, select "Order Number".
 - b) go to the Shipping tab. set 'Autocreate Deliveries' to Yes.
 - c) go to the Shipping tab. set 'Auto Pick Confirm' to Yes.
 - d) go to the Shipping tab. set 'Ship Confirm Rule' to 'Auto Ship'.

4. Apply the package OEXPSHCB.pls

This was done for the ITS program failing with error ORA-04030 WHILE RUNNING INTERFACE TRIP STOP. Related bug 7142581.

5. ITS Program does not do multi-threading properly.

1. Change Stops per batch to 1 (done in submit_O2C.sql).
2. Set the 'Autocreate Delivery Criteria' value to 'Within Order' for release rule 'EBS_Release_Rule1'.

This will ensure that one delivery is created for each order. Hence, if you run a test case with 50 orders, then 50 deliveries/stops will be created and you can see the effect of running ITS program with multiple threads.

6. Most recent submit_O2C.sql does the following (you have the most recent).

- Removed calls to Auto deliveries and Ship Confirm as they are autoated now from Pick Release.
- Added code check_its which will submit Autoinvoice only after making sure all the records are processed successfully by ITS and have 'CLOSED' status.
- Interface Trip Stop program is submitted with Stops per batch = 1 which evenly distributes the processes lines among different threads.

APPLICATION TUNING

Database:

1. R12 tuning through RUP 4 and handover of benchmark kit.
2. The index wsh.wsh_delivery_details_n8 was dropped
3. The index wsh_pr_workers (NVL(INVENTORY_ITEM_ID,-99)) was created
4. The index GL_PERIOD_STATUSES (application_id , set_of_books_id , adjustment_period_flag) was created
5. The index wsh.wsh_pr_workers(delivery_id) was created

Application:

Prompt Disabling Gather Stats Job

```
conn / as sysdba
exec
DBMS_SCHEDULER.DISABLE('GATHER_STATS_JOB'
);
select enabled from dba_scheduler_jobs where job_name
like 'GATHER%';
```

Prompt Creating index wsh_pr_workers_n1 on

```
wsh_pr_workers
conn apps/apps@vis
create index wsh.wsh_pr_workers_n1 on
WSH_PR_WORKERS(NVL(INVENTORY_ITEM_ID,-99),
ORGANIZATION_ID,PROCESSED);
```

Prompt Creating index wsh_pr_workers_n2 on

```
wsh_pr_workers
create index wsh.wsh_pr_workers_n2 on
WSH_PR_WORKERS(DELIVERY_ID);
```

Prompt Creating Index

```
RA_CUSTOMER_TRX_LINES_C02 on
RA_CUSTOMER_TRX_LINES_ALL:
create index ar.RA_CUSTOMER_TRX_LINES_C02 on
ar.RA_CUSTOMER_TRX_LINES_ALL(INTERFACE_LI
NE_ATTRIBUTE6,INTERFACE_LINE_CONTEXT)
parallel nologging ;
```

Prompt Alter index RA_CUSTOMER_TRX_LINES_C02 to make it noparallel

```
alter index ar.RA_CUSTOMER_TRX_LINES_C02
noparallel ;
```

Prompt Creating index GL_PERIOD_STATUSES_CO1 on GL_PERIOD_STATUSES:

```
create index gl.GL_PERIOD_STATUSES_CO1 on
GL_PERIOD_STATUSES (APPLICATION_ID,
SET_OF_BOOKS_ID, END_DATE, START_DATE);
```

Prompt Now gathering stats on tables

```
exec fnd_stats.gather_table_stats(ownname => 'WSH',
tabname => 'WSH_DELIVERY_DETAILS', percent =>
100);
exec fnd_stats.gather_table_stats(ownname => 'WSH',
tabname => 'WSH_TRIP_STOPS', percent => 100);
exec fnd_stats.gather_table_stats(ownname => 'WSH',
tabname => 'WSH_PR_WORKERS', percent => 100);
exec fnd_stats.gather_table_stats(ownname => 'APPLSYS',
tabname => 'WF_ITEMS', percent => 100);
exec fnd_stats.gather_table_stats(ownname => 'APPLSYS',
tabname => 'WF_ITEM_ACTIVITY_STATUSES', percent
=> 100);
exec fnd_stats.gather_table_stats(ownname => 'APPLSYS',
tabname => 'WF_ITEM_ACTIVITY_STATUSES_H',
percent => 100);
exec fnd_stats.gather_table_stats(ownname => 'APPLSYS',
tabname => 'WF_ITEM_ATTRIBUTE_VALUES', percent
=> 100);
exec fnd_stats.gather_table_stats(ownname => 'APPLSYS',
tabname => 'WF_PROCESS_ACTIVITIES', percent =>
100);
```

Prompt Now gathering stats on indexes

```
exec fnd_stats.gather_index_stats(ownname=>'WSH',
indname=>'WSH_PR_WORKERS_N1', percent=>100);
exec fnd_stats.gather_index_stats(ownname=>'WSH',
indname=>'WSH_PR_WORKERS_N2', percent=>100);
exec fnd_stats.gather_index_stats(ownname=>'AR',
indname=>'RA_CUSTOMER_TRX_LINES_C02',
percent=>100);
exec fnd_stats.gather_index_stats(ownname=>'GL',
indname=>'GL_PERIOD_STATUSES_CO1',
percent=>100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEMS_N1', percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEMS_N2', percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEMS_N3', percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEMS_N4', percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEMS_N5', percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEMS_N6', percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEMS_PK', percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEM_ATTRIBUTE_VALUES_PK',
percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEM_ACTIVITY_STATUSES_H_N1',
percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEM_ACTIVITY_STATUSES_H_N2',
percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEM_ACTIVITY_STATUSES_H_N3',
percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEM_ACTIVITY_STATUSES_N1',
percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEM_ACTIVITY_STATUSES_N2',
percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEM_ACTIVITY_STATUSES_N3',
percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEM_ACTIVITY_STATUSES_N4',
percent => 100);
exec fnd_stats.gather_index_stats(ownname=>'APPLSYS',
indname=>'WF_ITEM_ACTIVITY_STATUSES_PK',
percent => 100);
```

Improvements made to Inventory Workers:

operations/welcome
Inventory, Vision Operations (USA)
Setup
Transactions
Interface Managers
V1
Material transaction
Worker Rows 500 (default 200)

Standard Manager: Work shifts, process=8,
sleep time=15 sec
Inventory Manager: Work shifts, process=8,
sleep time=10 sec

OPERATING SYSTEM TUNING DATABASE OPERATING SYSTEM TUNING

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

```
kernel.shmmni = 4096
kernel.sem = 250 32000 100 128
fs.file-max = 327679
net.ipv4.ip_local_port_range = 1024 65500
net.core.rmem_default = 419304
net.core.rmem_max = 4194304
net.core.wmem_default = 262144
net.core.wmem_max = 262144
net.ipv4.tcp_wmem = 262144 262144 262144
net.ipv4.tcp_rmem = 4194304 4194304 4194304
kernel.msgmax = 65536
kernel.msgmnb = 65535
kernel.msgmni = 2878
```

vm.nr_hugepages=8192

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

```
oracle soft nproc 131072
oracle hard nproc 131072
oracle soft nofile 131072
oracle hard nofile 131072
```

```
oracle hard memlock 16777216
oracle soft memlock 16777216
```

3. Hugepages was enabled for the database instance

BENCHMARK ENVIRONMENT

HARDWARE CONFIGURATION

An IBM System x3550 M2 (7946-AC1) server was used for the database/application/web server. It was equipped with the following:

- 2 × 2.93 GHz Intel Xeon Four-Core 5500 processors (8 cores total), each with 32 KB of Level 1 cache, 256 KB of Level 2 cache and 8 MB of Level 3 cache
- Total Memory: 64 GB (14.6 – 18.2 GB used)
- Network: Gigabit full duplex.
- Operating system: Red Hat Enterprise Linux 5
- For more details on the IBM System x3550 M2, please visit <http://www-03.ibm.com/systems/x/hardware/rack/x3550m2/index.html>
- Storage: DS4700 with a total of 48 × 145 GB 15K RPM fibre-channel disks.
- For more details on DS4700, please visit <http://www.ibm.com/servers/storage/disk/ds4000/ds4700/index.html>

SOFTWARE VERSIONS

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

Oracle10g™ 10.2.0.3 (64-bit)

Red Hat Enterprise Linux 5 update 3 (64-bit) (on the database/application/web server)

Note: Oracle accepts that the results obtained using Red Hat Enterprise Linux 5 very closely approximates results that may be obtained if Oracle Enterprise Linux 5 were used in this test.

Glossary and Acronyms:

ATP	Available to Promise
BEE	Batch Element Entries
HVOP	High Volume Order Processing
OASB	Oracle Applications Standard Benchmark
RAC	Real Applications Clusters

ORACLE

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

E-Business Suite, AppsNet, Collaboration Suite, Oracle Direct and RAC are trademarks of Oracle, Inc. Oracle, JD Edwards, PeopleSoft and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. The information contained herein is subject to change without notice. Copyright © 2009 Oracle, Inc. All rights reserved. C/N 0XX-1009 Results Registration Number: 0930

©2009 IBM, Inc. All rights reserved. IBM, the IBM logo, AIX, BladeCenter, POWER6 and IBM Systems Storage are trademarks or registered trademarks of IBM, Inc. in the United States and other countries.