

ORACLE LINUX TEST (OLT)

Oracle Linux Tests are designed to verify Linux kernel functionality and stability essential for the Oracle Database. The Oracle Linux Test (OLT) kit, which is distributed as an archive file , provides an automated framework, a set of tools and tests for virtualized and non-virtualized systems. This document is for QA teams who conduct the OLT either within the framework of the Oracle Validated Configurations program; or as additional Oracle Database-specific tests for deployment configurations; or as additional Oracle VM specific tests for deployment configurations; or as part of internal tests for newly developed software or hardware solutions. Readers should be familiar with Oracle Database, basic Linux and Oracle VM administration and hardware administration targeted for the validation.

OLT Overview

This document provides a description of the tests used in OLT, Version 2.3.0, which uses open source tools. The tests are designed based on an analysis of Oracle interaction with OS/Oracle VM and issues identified by customers. The OLT kit is comprised of a range of tests including installation, functional, stress and destructive tests. The OLT kit is executed on a system that is to be validated, and OLT tests—which take a week to run—execute in sequence with result logs generated during test execution; these logs are then used to determine the success or failure of the tests.

Note: OLT is a testing tool and not meant for performance comparisons.

Results Verification Methodology

Tests results are verified by the automatic analysis of:

- **OS errors** (/var/log/messages, dmesg, memory information, etc)
- **Oracle errors** (based on Oracle alert logs, trace files, etc)
- **Tests output** (based on test logs)
- **Oracle VM related errors** (/var/log/messages, dmesg, ./var/log/xen, etc)

OLT Tools

The OLT kit is comprised of four tools:

- Silentinstall tool
- OLTP workload tool
- DSS workload tool
- fio (open source tool)
- Custom tools
- Regression test tools
- virtualization test tools (xm-test and virt-config)

Silentinstall Tool

The silentinstall tool installs the Oracle product/version specified on the machine. The tool will install, re-link and do a basic configuration to validate the installation.

OLTP (Online Transaction Processing) Workload Tool

The OLTP workload tool simulates an OLTP environment/workload for one or more single instances on a node as well as for an Oracle Real Application Cluster (RAC) instance. The tool can be used to simulate stress on the following OS components - memory, I/O and interconnect. The tool provides the following capabilities:

- Creating databases of various sizes (9GB, 18GB, 36GB or 72GB) on various storage types
- For load variation on the system, it allows:
 - Varying SGA sizes and using different memory options
 - Varying Oracle settings for asynchronous I/O, direct I/O
 - Varying the workload type (I/O intensive, CPU intensive)

DSS (Decision Support System) Workload Tool

The DSS workload tool simulates a DSS environment. The tool is used to simulate stress on the system. On the operating system, the load is on memory, I/O and CPU. The tests provide the following capabilities:

- Creating databases of various sizes (1GB, 10GB, 30GB) on various storage types
- For load variation on the system, it allows:
 - Varying SGA sizes and using different memory options
 - Varying Oracle settings for asynchronous IO, direct IO

Fio Simulator

fio is an open source IO simulator. It can be used to simulate different IO load by varying parameters such as – IO drivers (libaio, sync), IO patterns (ran, seq), size, no of streams, type of operation, etc.

Custom Tools

Custom tools include test tools that can be run standalone or in parallel with other tests. These tools include Oracle utilities as well as specialized workload simulators. Currently, this includes kernel test modules such as mempressure, etc. to provide additional stress to the OS.

Regression Test tools

This contains tests that simulate bugs present in earlier kernels.

Xm-test

xm-test is a tool taken from **xen tools**. It runs standalone as part of dom0 tests. It can also be run in conjunction with io,memory type workloads on guests .

Virt-config

This tool provides automated setup and management of guests via a command line interface. This tool provides support for

- Guest creation (self, seed, Image, virt-install) for paravirtualized and hardware virtualized guests
- Guest configuration and reconfiguration (storage, memory, network, filesystems)
- Guest cleanup

Tests

Tests in the OLT kit include a comprehensive set of tests ranging from Oracle installation tests, functional tests that verify OS and virtualization functionality that Oracle requires, stress tests that stress the system through different

workloads, and tests developed based on customer scenarios. For baremetal systems, the tests are organized into the following broad categories: Installation, Functional, Stress, and RAC Destructive. For virtualization, the tests are organized into Dom0 and guest related tests.

Baremetal Systems

Installation Tests

-

Oracle is installed silently through the tests listed below either on a single node or for RAC on a cluster.

Test Suite Name	Test Description	Time
<u>Install-silent-oracle-<version></u> <i>Oracle Database Server installation on the OS</i>	Verify the Oracle installation on the system by: <ul style="list-style-type: none"> • Installing Oracle Database 11g Release 2 on the system • Applying Oracle patches (if required) • Checking for re-linking errors or related errors for Oracle during this process 	30 mins to 1.5 hours
<u>Install-silent-rac</u> <i>Oracle RAC installation on the OS</i>	Verify the Oracle installation on the system by: <ul style="list-style-type: none"> • Installing Oracle Database 11g Release 2 on the system • Applying Oracle patches (if required) • Checking for re-linking errors or related errors for Oracle during this process 	1.5 hours

System Verification Tests

Test Suite Name	Test Description	Time
sv-oltverify <i>Capture configuration of machine</i>	Capture the following configuration details and verify that they comply with Oracle recommendations: <ul style="list-style-type: none"> • kernel and ulimit setting • oracle user configuration • verify storage parameters if any 	2 mins
sv-asm	Verify the oracleasm module is included for the current kernel	1 min
sv-hangcheck	Verify if the hangcheck timer module is included for the current kernel	1 min
sv-ocfs2	Verify if the ocfs2 kernel module is included for the current kernel	1 min
sv-ethtool <i>Verify if ethtool works with the network driver</i>	Verify if ethtool works with the network driver for getting n/w related settings	2 mins
sv-miitool <i>Verify if miitool works with the network driver</i>	Verify if mii-tool works with the network driver for getting n/w related settings	2 mins

Test Suite Name	Test Description	Time
Sv-hyperthreading <i>Verify if hyperthreading is enabled on the system</i>	Verify if hyperthreading is enabled	2 mins
Sv-numa <i>Verify if numa is enabled or disabled on the system</i>	Verify if numa is enabled/disabled on the system by using libnuma	2 mins

Functional Tests

Test Suite Name	Test Description	Time
<u>ft-aio</u> <i>Asynchronous i/o with Oracle</i>	Set <i>filesystemio_options=asynch</i> in init.ora. Create a database (if not present) and execute an OLTP workload. Verify that Oracle IO layer uses aio system calls – <i>io_submit, io_getevents</i> - via <i>strace</i>	30 mins
<u>ft-dio</u> <i>Direct i/o with Oracle</i>	Set <i>filesystemio_options=directio</i> in init.ora. Create a database (if not present) and execute an OLTP workload. Verify that Oracle opens the database files with the <i>O_DIRECT</i> flag via <i>strace</i>	30 mins
<u>ft-hugetlb</u> <i>Hugepages with Oracle</i>	Reserve hugepages at bootup either via <i>grub</i> or at runtime through <i>/proc/sys/vm/nr_hugepages</i> (For some distributions configuring <i>memlock</i> is also required) Create a database (if not present) and execute an OLTP workload Verify that hugepages are being used from <i>/proc/meminfo</i> corresponding to the <i>sga</i> size and are released after the instance is shutdown	30 mins
<u>ft-aio-dio</u> <i>Asynchronous and direct i/o with Oracle</i>	Set <i>filesystemio_options=setall</i> in init.ora. Create a database (if not present) and execute an OLTP workload. Verify that Oracle IO layer uses aio system calls - <i>io_submit, io_getevents</i> - via <i>strace</i> and opens the database files with the <i>O_DIRECT</i> flag via <i>strace</i>	30 mins
<u>ft-shmfs-tmpfs</u> <i>x86: VLM with tmpfs</i>	Verify that Oracle uses <i>tmpfs</i> filesystem with <i>VLM</i> when <i>tmpfs</i> is mounted Create a database (if not present) and execute an OLTP workload. Verify that the <i>tmpfs</i> filesystem is used and there are no errors in Oracle	30 min

Test Suite Name	Test Description	Time
<u>ft-shmfs-ramfs</u> <i>x86: VLM with ramfs</i>	Verify that Oracle uses ramfs filesystem with VLM when ramfs is mounted Create a database (if not present) and execute an OLTP workload. Verify that the ramfs filesystem is used and there are no errors in Oracle	30 min
<u>ft-remap-file-pages</u> <i>x86: remap-file pages with Oracle</i>	Verify that remap-file-pages gets used by Oracle in VLM mode Create a database (if not present) and execute an OLTP workload Verify that Oracle uses the remap-file-pages system calls.	30 min

Stress Tests: Single Instance

Test result verification is done using the standard approach; please refer to the Results Verification Methodology section in the Overview.

Test Suite Name	Test Description	Time
<u>st-mem2</u> <i>Verify system behavior by providing high memory stress</i>	Execute 3 olt workloads in parallel across 3 database instances with a database hot backup consuming about 80% of the total memory and <ul style="list-style-type: none"> The 3 databases are of varying sizes - 9G (8k block size), 18G (4k block size), 72GB (8k block size) to provide a variable mix of load and block sizes. The 3 instances provide a mix of varying io (aio and direct io) and memory loads (sga and no of users) depending on the amount of system memory. The workloads used are io and cpu loads. 	5 hours
<u>st-mem-spike</u> <i>Verify system behavior by providing spiked memory loads to the system</i>	Start a database instance on a database of 18GB (4k block size) and execute transactions. Simulate sporadic batch (100) connections->query exe->disconnections to the database in periodic intervals.	2 hours
<u>St-mem1</u> <i>Verify system behavior by providing high memory stress</i>	Execute 3 olt workloads in parallel across 3 database instances consuming about 70% of the total memory. <ul style="list-style-type: none"> The 3 databases are of varying sizes - 9G (8k block size), 18G (4k block size), 72GB (8k block size) to provide a variable mix of load and block sizes. The 3 instances provide a mix of varying io (aio and direct io) and memory loads (sga and no of users) depending on the amount of system memory. The workloads used are io and cpu loads. 	4 hours

Test Suite Name	Test Description	Time
<u>St-mem3</u> <i>Verify system behavior by providing high memory stress</i>	Execute 3 olt workloads in parallel across 3 database instances consuming about 70% of the total memory. <ul style="list-style-type: none"> The 3 databases are of varying sizes - 9G (8k block size), 18G (4k block size), 72GB (8k block size) to provide a variable mix of load and block sizes. The 3 instances provide a mix of varying io (aio and direct io) and memory loads (sga and no of users) depending on the amount of system memory. The workloads used are cpu and erp based loads 	6 hours
<u>St-memleak1</u> <i>Verify if there is any potential memory leak in the system</i>	Startup 3 databases in sequence and execute an oltp workload on each instance for a period of 4 hours each. Verify that the system does not sufficiently deplete the memory and that the tests execute.	12 hours
<u>St-maxusers</u> <i>Find a range of users the system is capable of handling</i>	Determine a nominal range of users that the system can scale to for a particular configuration <ul style="list-style-type: none"> Execute oltp workload with “n” users Re-execute oltp workload with n+500 users based on available memory during previous run. If n+500 consumes more than the estimated amount of memory then the test is terminated with “n” as the max number of users. 	1-4 hours
<u>St-mem-io1</u> <i>Verify system behavior with spiked loads</i>	Execute an io intensive oltp workload. In parallel, execute mempressure to allocate memory in chunks and de-allocate them periodically. Execute an io intensive oltp workload. In parallel, simulate a spiked pattern of buffer head consumption on the system. Execute an io intensive oltp workload. In parallel, simulate a constant pressure-based pattern of buffer heads on the system.	12 hours
<u>St-fs2</u> <i>Verify the locking behaviour of files on a fs_</i>	Verify the locking behaviour with a file on a filesystem by - locking and unlocking the file when it is already created/moved/copied	10 min
<u>St-power-dss</u> <i>Verify system behavior when running a dss load</i>	Execute a cpu and io intensive dss workload.	1 hour
<u>St-tput-dss</u> <i>Verify system behavior when running a dss load</i>	Execute multiple streams of an cpu and io intensive dss workload.	1 hour

Test Suite Name	Test Description	Time
<u>St-mempressure1</u> <i>Verify impact on the system due to memory fragmentation</i>	Execute an oltp database workload. Execute a memory pressure test that allocates and de-allocates chunks of memory to cause fragmentation.	1-4 hours
<u>St-mempressure2</u> <i>Verify impact on the system due to spiked memory usage</i>	Simulate a spiked pattern of buffer head consumption on the system.	1 hour
<u>St-mempressure3</u> <i>Verify impact on the system on applying memory pressure.</i>	Simulate a constant pressure-based pattern of buffer heads on the system.	1 hour
<u>St-mempressure-high</u> <i>x86: Verify impact on the system on applying highmem pressure.</i>	Execute an oltp database workload. Generate pressure on highmem that allocates and de-allocates chunks of memory to cause fragmentation.	30 min
<u>St-io1</u> <i>Stress the io layer using aio</i>	Stress the io subsystem by writing to a files with async io using fio	30 min
<u>Bug-lowmempressure</u> <i>Verify bug with incorrect oom killer invocation</i>	Simulate bug where oom killer was being invoked with memory still available. Verify that the oom killer is not invoked during this test.	1 hour
<u>Bug-io1</u>	Simulate bug with cfq scheduler versus deadline scheduler. This affects the time taken for io when using the deadline scheduler. This is done using fio. Verification is manual. There should not be a large time difference in the completion latency time between the two schedulers for the io load.	30 min
<u>Bug-io2</u>	Simulate system deadlock seen with using aio+dio using fio as a simulator	30 min
<u>Bug-io3</u>	Simulate system panic seen with using aio_dio using fio as a simulator	30 min
<u>Bug-io4</u>	Simulate system panic using fio when using aio or splice system calls	~4 hours
<u>Bug-nfsmount</u>	Regression test for failure to mount seen with mount points with different mount option and created from same volume within a sub directory	2 mins
<u>Bug-fracturedblock</u>	Simulate a fractured block bug on NFS file system	6-7 hours

Test Suite Name	Test Description	Time
<u>bug-hugepages</u>	Simulate bugs seen with hugepage allocation/deallocations	45mins

Stress Tests: Real Application Clusters (RAC)

Test result verification is done using the standard approach; please refer to the Results Verification Methodology section in the Overview.

Test Suite Name	Test Description	Time
rac-st- <storagetype>-1 <i>Verify System behavior with RAC and a "medium" stressed system</i>	Execute an io intensive oltp workload across 'n' nodes with direct io enabled with about 50% of memory (sga=RAM/6) consumed.	2 hours
rac-st- <storagetype>-2 <i>Verify System behavior with RAC and a "medium" stressed system with async io</i>	Execute an io iand cpu intensive oltp workload across 'n' nodes with async io and direct io enabled with about 60% of memory (sga=RAM/3) consumed.	1 hour
rac-st- <storagetype>-3 <i>Verify System behavior with RAC using large SGA</i>	Execute a cpu intensive oltp workload across 'n' nodes with direct io enabled with about 70% of memory (sga=RAM/2) consumed.	1 hour
rac-st- <storagetype>- long1 <i>Verify System behavior/memory leaks with RAC and a "medium" stressed system</i>	Execute an oltp based workload load across an 'n' node RAC for 12 hours	12 hours

Test Suite Name	Test Description	Time
rac-st- <storagetype>- memleak1 <i>Verify System behavior/memory leaks with RAC</i>	Startup 3 databases in sequence and execute an oltp workload with a mix of cpu, io and cpu_io stress on each instance for a period of 4, 2 and 1 hours respectively.	8 hours
rac-st- <storagetype>- mempresure1/2 <i>Verify System behavior with RAC when having spiked loads</i>	Execute an oltp workload on an 'n' node RAC which consumes about 80% of memory. In parallel, simulate a spiked pattern of buffer head consumption on the system.	2 hours
rac-st- <storagetype>- mempresure1/3 <i>Verify System behavior with RAC when having additional memory pressure</i>	Execute an oltp workload on an 'n' node RAC which consumes about 80% of memory. In parallel, simulate a constant pressure-based pattern of buffer heads on the system.	2 hours
Rac-st- <storagetype>- memleak2 <i>Verify System behavior/memory leaks with RAC</i>	Startup 3 databases in sequence and execute an oltp workload on each instance for a period of 4 hours each <ul style="list-style-type: none"> • Verify that the system does not sufficiently deplete the memory and that the tests can run. 	12 hours
Rac-st- <storagetype>- gzipscp1 <i>Verify System behavior with RAC when having additional memory pressure</i>	Execute an oltp workload on an 'n' node RAC which consumes about 60% of memory for 8 hours. In parallel, provide additional memory pressure via gzip/scp.	8 hours

RAC Destructive Tests

Test result verification is done using the standard approach; please refer to the Results Verification Methodology section in the Overview.

Test Suite Name	Test Description	Time
Dest-1 (rac-dest-<>-asm/rac-st-<>-1) <i>Verify system, database sanity; failover when asm instance crashes</i>	Execute an oltp workload on an 'n' node RAC Simulate an asm instance crash on one node when the “n” nodes are fully loaded Verification: <ul style="list-style-type: none"> • Check for vip and process failover to the surviving node • Check for database and/or crs errors on both nodes • Check for storage related errors 	30 mins
Dest-2 (rac-dest-<>-crs/rac-st-<>-1) <i>Verify system, database sanity; failover when crs crashes</i>	Execute an oltp workload on an 'n' node RAC Simulate a crs crash on one node when the ‘n’ nodes are fully loaded Verification: <ul style="list-style-type: none"> • Check for vip and process failover to the surviving node • Check for database and/or crs errors on both nodes 	30 mins
Dest-3 (rac-dest-<>-crs/rac-st-<>-2) <i>Verify system, database sanity; failover when instance crashes</i>	Execute an oltp workload on an 'n' node RAC Simulate an instance crash on one node when the ‘n’ nodes are fully loaded Verification: <ul style="list-style-type: none"> • Check for vip and process failover to the surviving node • Check for database and/or crs errors on both nodes 	30 mins
Dest-4 (Manual) <i>Verify system, database sanity; failover when the network fails</i>	Execute an oltp workload on an 'n' node RAC Perform a network failure on one node when the ‘n’ nodes are fully loaded Verification: <ul style="list-style-type: none"> • Check for vip and process failover to the surviving node • Check for database and/or crs errors on both nodes 	30 mins
Dest-5 (Manual) <i>Verify system, database sanity; failover when the storage fails on a cluster where no. of nodes > 2.</i>	Execute an oltp workload on an 'n' node RAC Perform multiple failures on one or more nodes when the ‘n’ nodes (ONLY where n > 2) are fully loaded Verification: <ul style="list-style-type: none"> • Check for vip and process failover to the surviving node • Check for database and/or crs errors on both nodes • Check for storage related errors 	1 hour

Test Suite Name	Test Description	Time
Dest-6 (rac-dest- <storage>-HA1) (Manual) <i>Verify system, database sanity; failover when there is a failure and failback when the failure is no longer present.</i>	Execute an oltpworkload on 'n' node RAC. Perform a storage failure on one node. Failover should take place to one of the other nodes in the cluster. Restore the storage and the node (in order) that failed - failback should occur on the concerned node. Verification : <ul style="list-style-type: none"> • Check for vip and process failover to the surviving node • Check for invalid database and/or crs errors on both nodes • Check for invalid storage related errors • Check crs and database again comes back on failed node after failure and shares workload from the available nodes. • Check for crs and database errors on both the nodes 	1 hour

For Virtualization

Dom0

This covers the tests run on the OVM Server

Test Suite Name	Test Description	Time
Sv-ovs <i>Verify system configuration is detected correctly.</i>	Check for memory,cpus,HVM capability, Hypervisor architecture,network cards, on the Server. Verification: <ul style="list-style-type: none"> ● Check for any error messages in xm demsg and dmesg of dom0. ● Verify configuration detected with configuration present. 	5 Min
Xm-dom-test <i>Verify guest creation by varying vm.cfg parameters</i>	Create guests with different cpu,pae,apic.acpi combinations: when 1) pae=1 2) apic=1 3) acpi=1 for all guests Verification: <ul style="list-style-type: none"> ● Check for any error messages in xm demsg and dmesg of dom0. ● Check if guests boot up 	20 min per domain
Xm-dom-parallel <i>Verify creation of guests in parallel</i>	Create multiple doms in parallel. Verification: <ul style="list-style-type: none"> ● Check for any error messages in xm demsg and dmesg of dom0. ● Check if guests boot up 	20 min

Test Suite Name	Test Description	Time
Scp-test <i>Verify OVM Server and guest behaviour with large file copy</i>	Create and bring up domains scp large file across domains Verification: <ul style="list-style-type: none"> ● Check for any error messages in xm demsg and dmesg of dom0. ● Check if guests boot up ● Verify that scp completes 	10 min per domain
Xen-vbd-config <i>Verify guest creation with different storage options</i>	Create guests with different storage formats (hd, sd, xvd) based on HVM/PVM configurations Verification: <ul style="list-style-type: none"> ● Check for any error messages in xm demsg and dmesg of dom0. ● Check if guests boot up and storage is accessible without errors 	30 min per domain
Xen-vif-config <i>Verify bridge configuration on guests</i>	Create domains with various no of bridges, start from 1 to max available bridges in the dom0. Verification: <ul style="list-style-type: none"> ● Check for any error messages in xm demsg and dmesg of dom0. ● Check if guests boot up and network is configured correctly. 	20 min per domain.
Xm-stress	Creates, shuts down, destroys multiple domains randomly. Verification: <ul style="list-style-type: none"> ● Check that guests boot up/shutdown cleanly ● Verify that the system does not crash 	10 min per domain
xm-standalone-medium	Executes xm-test “medium” load tests	10 mins

Dom0+ DomU

Test Suite Name	Test Description	Time
Xm-test-create-destroy, xm-mem1-db2_1 <i>Verify xm create/destroy with workload running on guests</i>	Create and destroy multiple domains when running oracle workload on the guests Verification: <ul style="list-style-type: none"> ● Check for any error messages in xen logs and system logs ● Verify that the server functions correctly before and after the xm test. 	Max 2 hour (with db creation) per domain

Test Suite Name	Test Description	Time
Xm-test-pause-unpause,xm-mem1-dbt2_1 <i>Verify xm pause/unpause with workload running on guests</i>	Pause and unpause multiple domains when running oracle workload on the guests Verification: <ul style="list-style-type: none"> ● Check for any error messages in xen logs and system logs ● Verify that the guests functions correctly before and after the xm test. ● Verify that the dbt2 test completes successfully. 	Max 2 hour (with db creation) per domain
Xm-test-vcupin, xm-mem1-dbt2_3 <i>Verify cpu pinning with workload running on guests</i>	Verify cpu pinning when running oracle workload on the guests Verification: <ul style="list-style-type: none"> ● Check for any error messages in xen logs and system logs ● Verify that the guests functions correctly before and after the xm test. ● Verify that the dbt2 test completes successfully 	Max 2 hour (with db creation)per domain
Xm-test-save-restore,xm-mem1-dbt2_3 <i>Verify xm save/restore with workload running on guests</i>	Save and restore multiple domains when running oracle workload on the guests Verification: <ul style="list-style-type: none"> ● Check for any error messages in xen logs and system logs ● Verify that the guests functions correctly before and after the xm test. ● Verify that the dbt2 test completes successfully 	Max 2 hour (with db creation)per domain

DomU

Test Suite Name	Test Description	Time
ft-aio	Refer to Functional Tests	Refer to Functional Tests
ft-dio	Refer to Functional Tests	Refer to Functional Tests
ft-aio-dio	Refer to Functional Tests	Refer to Functional Tests

Test Suite Name	Test Description	Time
st-mem2-dom <i>Verify system behavior by providing high memory stress</i>	Execute 2 olt workloads in parallel across 2 database instances with a database hot backup consuming about 80% of the total memory and <ul style="list-style-type: none"> • The 2 databases are of varying sizes - 9G (8k block size), 72GB (8k block size) to provide a variable mix of load. • The 2 instances provide a mix of varying io (aio and direct io) and memory loads (sga and no of users) depending on the amount of system memory. The workloads used are io and cpu loads. 	4 hours

For migration tests for Validated configurations please refer to OLT-OVM-VC-Supplement.txt.

Oracle Validated Configurations

For more information about the Oracle Validated Configuration program, visit oracle.com/technology/tech/linux/validated-configurations