Oracle Multitenant Scalability Advantages
An Audit of Performance in the Consolidation Context

Executive Summary

Due to a dramatic increase in the amount and variety of data that organizations must manage in today's ever-changing business and technology environment, chief information officers (CIOs) and database administrators (DBAs) are looking for techniques to help control 'fully-loaded' database management costs. Fully-loaded database management costs include the related costs of software, hardware, and network, plus the related costs for information technology (IT), business, and sometimes 3rd party IT services personnel needed to attend to database testing, deployment, and administration.

One growing-in-popularity technique to control fully-loaded database management costs is known as "database consolidation." Database consolidation helps CIOs and DBAs make more effective use of their database infrastructure, reducing infrastructure underutilization, and enabling IT to deploy more databases onto a shared infrastructure in a cloud-like fashion. Naturally CIOs and DBAs want to implement database consolidation without causing user-visible application latency, without increasing the costs and complexity of database administration, and without endangering the availability of the databases involved.

Being the world's largest supplier of enterprise databases, Oracle has crafted database consolidation options to help its customers control database costs. For example, Oracle has engineered specific infrastructure appliances designed for database consolidation, such as Oracle SuperCluster. In addition, as the result of a multi-year effort to re-architect its market-leading database, in mid-2013 Oracle released Oracle Database 12c which included what Oracle refers to as its "Multitenant" database option. Multitenant is a candidate as a cloud database, but also a candidate for database consolidation.

Oracle created its Multitenant database option by splitting the database architecture into a container database ("CDB") that handles systems and background processing, and pluggable databases ("PDBs") that actually house the data. The PDBs operate in the context of the CDB container, enabling the CDB to optimally take advantage of recent advancements in processors and storage, and letting the CDB efficiently dole out system resources to the PDBs. In theory, by deploying the Multitenant database on a purpose-designed infrastructure like SuperCluster enterprises should be able to achieve database consolidation driven lower costs, without paying penalties in terms of performance, availability, or administrative costs.

Given that the Multitenant option used for database consolidation represents a new generation of database management, there are no standard benchmarks that test, for example, the performance of the PDBs operating in the CDB container versus how they operate in standalone mode. Oracle therefore, built a test bed to measure the performance and elasticity of consolidation using the Multitenant approach versus standalone databases. A detailed audit of that set of tests conducted by Oracle engineered is herein, but there are several key takeaways that are clearly discernible from the test results:

1. The Multitenant approach yields considerably higher – nearly double – throughput versus the standalone database approach in a consolidation scenario given the same number of databases.
2. When throughput was fixed, the Multitenant approach enabled 50% more PDBs to be consolidated on the database infrastructure when compared to standalone databases (aka non-CDB), clearly proving the consolidation advantages of Multitenant

3. Even when Oracle compared the performance of a single PDB running in a CDB versus a single standalone non-CDB Oracle Database 12c, there was no appreciable performance difference.
4. In a consolidation environment requiring elasticity, such as a private cloud, the Multitenant database consolidation option outperformed the non-CDB consolidation approach in terms of responsiveness to fluctuating demand and compute resource allocation.

Database Consolidation at the Core of IT Economics

Organizations of all kinds have pursued improving the economics of their IT investments during the 2000s. "Do more with less" describes the mantra of most CIOs and IT departments, and many would append "and do it faster" to the incantation. Given how virtually all applications require a database or data repository, database management is considered one of the lynchpins of IT economics. CIOs, DBAs, and other IT executives and administrators that do an effective job of controlling database management costs score well in terms of "Do more with less."

Oracle's new Oracle Database 12c with the Multitenant option aims to help CIOs and DBAs do more with less; to operate more databases on the same infrastructure without sacrificing performance or increasing administrative complexity. Oracle's engineers built and ran a test suite to determine how the Multitenant approach would perform and scale for database consolidation. This paper is an independent audit of Oracle's consolidation tests, and validates findings regarding the experiences customers might expect if they deploy Oracle Database 12c with the Multitenant option for the purpose of database consolidation.

To set the context for the tests, let's first look at the general concept of database consolidation, then delve briefly into the history and purpose of Oracle Database 12c with the Multitenant option. The test metrics, goals, and design are then outlined, followed by summaries of the test results, and key takeaways.

Database Consolidation to the Rescue of the Costs of Data Explosion

Given the stunning amount of data growth most organizations are experiencing, with expansion happening in terms of both volume and types of data, CIOs and DBAs cannot reasonably expect to reduce the number of databases their organization needs any time soon. In addition, enterprises are inexorably deploying new apps to private and/or hybrid clouds, which also increases the number of databases, and not just for production but also for development and testing.

In larger organizations "database consolidation" … has emerged as a favored method to reduce the fully-loaded costs of database management

CIOs and DBAs have looked to their database vendors to help find means of dealing with database growth economically. Some database vendors have responded with a variety of techniques to help control fully-loaded database management costs. For example, many database vendors now offer flexible licensing models giving customers the option to shift costs from CAPEX to OPEX. Vendors have also enhanced how their databases take cost-reducing advantage of advancements in processors and storage, and have innovated new approaches to reduce infrastructure underutilization. In larger organizations "database consolidation," which encompasses a variety of cost controlling techniques, has emerged as a favored method to reduce the fully-loaded costs of database management.

The primary aim of database consolidation is to run more databases on shared infrastructures, thereby lowering hardware costs per database, increasing utilization, and thus "doing more with the same hardware." The tricky part of achieving effective database consolidation comes in the "and do it faster" challenge: CIOs and DBAs cannot afford to pack more databases on the same infrastructure if the related manageability costs of consolidation increase, and the availability and performance of the databases deteriorate.

In database consolidation the goal thus concerns increasing the 'density,' or number, of databases running on a fixed infrastructure without sacrificing manageability, availability or performance. It is also easy to see how database consolidation fits into the philosophy of cloud computing where a shared, typically virtualized, infrastructure and application platform serves multiple application and user constituencies.

Considering Oracle Database 12c as a Consolidation Candidate

Around six years ago, Oracle commenced research and development to re-architect its multi-decade market leading enterprise database precisely to make the database more relevant to cloud computing, and to help customers improve database management economics. The result of Oracle's effort is a multitenant variant of the of Oracle Database 12c, referred to as Oracle Multitenant. Herein we will refer to Oracle Database 12c in general as "DB12c", and DB12c's new multitenant option as "Multitenant."

DB12c, and with it Multitenant, became generally available commercially in June of 2013. Some of the primary design objectives for Multitenant included:

- **Cloud:** Increase the database's relevancy for cloud computing, whether deployed in public, private, or hybrid clouds, suggesting cloud-style multi-tenancy.
- **Density:** Support customers wishing to consolidate databases, i.e. to increase database density
- **Manageability:** Make it simpler for DBAs, developers, and related personnel to administer databases.
- **Availability and Performance:** Do all of the above without sacrificing availability and performance, and preferably improving performance for the majority of use cases.

Why did the DB12c development project take so long? In addition to the considerable engineering effort required to re-architect the Oracle database for the cloud era, the other reason why the project took five years was that Oracle refused to let any of the Oracle database's enterprise-grade characteristics degrade. That is, the Oracle database has long been about stability, availability, manageability, updateability, performance, and scalability. Re-architecting the Oracle database while not sacrificing any of those enterprise traits required extremely careful engineering. Customers wanted the same enterprise-grade experience with DB12c as they did with previous Oracle database versions.

> Oracle refused to let any of the Oracle database's enterprise-grade characteristics degrade.

How has Oracle done in terms of meeting its primary design objectives for DB12c?

- **Cloud:** Clearly DB12c meets the requirements for cloud-style computing, for customers may buy and deploy DB12c in a variety of cloud-like approaches. Expect additional cloud deployment options to be made available by Oracle over the coming 12 months.
- **Density:** While every customer may have a different experience depending on their unique requirements and environments, based on the testing audited and documented herein, Multitenant stands out as an excellent candidate for database consolidation.
- **Manageability:** Oracle has proven several times over that the revamped administrative and maintenance approach for DB12c is not only easier for DBAs, but faster. The ease and time required to provision new databases, to clone databases, and to apply patches to databases, has

improved considerably, in some cases by orders of magnitude, particularly with Multitenant. Most of the efficiency gains Multitenant exhibits in terms of manageability and administration trace directly to the re-architecture that split the database architecture into "container databases" (CDBs) and "pluggable databases" (PDBs).

- **Availability and Performance:** Determining how successful Oracle was at maintaining, or even improving availability in DB12c over previous database versions was impossible in the short run. Today, however, after roughly a year of production experience, it seems safe to say that DB12c appears stable for the vast majority of customers to consider for production use.

  Measuring individual database performance via standard benchmarks, particularly for mission critical databases, offers customers a useful indication of likely performance. Oracle Real Application Testing is a tool customers often use to benchmark their important databases. Measuring, however, the performance of databases in the context of database consolidation is not as straightforward. Given Oracle's belief that Multitenant qualifies as an excellent candidate for database consolidation, it made sense to test DB12c's performance specifically in the context of database consolidation.

Oracle Database 12c Consolidation Density Test Approach

Before delving into the specifics of Oracle's tests of its DB12c database for consolidation density, it is imperative to understand some of the unique attributes of DB12c's architecture, particularly the Multitenant architecture, because its architecture directly influences the test design.

Oracle Database 12c Deployment Choices

In Oracle Database 12c, the new Multitenant option is just one deployment choice. The more classic standalone/non-CDB database deployment option is also fully supported. The Multitenant option itself can be used in single-tenant configuration (a single PDB per CDB – which does not require or trigger the licensed option) or the multitenant configuration (two or more PDBs per CDB, which does require the licensed option.)

Oracle Multitenant Fundamentals: Container and Pluggable Databases

DB12c introduced fresh lingo into the enterprise database idiom, particularly for those in the Oracle community, including the two key terms of CDB and PDB. Oracle accomplished the Multitenant option for DB12c by splitting the database architecture into two parts, one part managed by each PDB, and another part managed by the CDB. The CDB processes background tasks and handles system resources such as system metadata and direct interaction with the infrastructure. A PDB supports the actual customer schema and data, but is managed cloud-like as a service within a containing CDB. Thus, many PDBs may run in a single CDB. From a PDB's perspective, the CDB acts as a multi-tenant container. How theoretically might the PDB and CDB approach impact a customer's experience?

- **Consolidation Density:** By splitting system-oriented processing from application-oriented processing, Multitenant can achieve higher consolidation density versus consolidating non-CDBs on the same infrastructure. For the customer the value proposition is clear: You should be able to run more production databases using the same hardware with the Multitenant option than with standalone databases, or be able to run the same number of databases with higher performance enjoyed by the PDBs operating in the Multitenant option.
- **Maintenance:** When database patches and upgrades are required, they typically only need to be applied to the CDB, meaning that the maintenance work takes place more rapidly. Imagine a scenario where 10 PDBs are running associated with 1 CDB: the patches applied to the CDB are immediately applicable to the PDBs. 10 non-CDB standalone databases would each require their

own equivalent patching process. The math is easy. DBAs would take 1/10th the time to apply the patches when using the Multitenant option. A similar set of savings could potentially apply to backup, restore, disaster recovery and other data administration tasks.

These are compelling benefits: Lower hardware costs, quicker provisioning and administration in far shorter periods of time, saving both time and money. But for enterprises, what good would all of this do if customers pay for these improvements with poorer performance? The Oracle test suite evaluates the performance and consolidation densities of the two consolidation strategies, the non-CDB approach and the CDB/PDB approach, on the same infrastructure, to measure the performance and consolidation density of both solutions.

Testing Measurement and Metrics

The Oracle performance team had a long list of measurement tools at its disposal, in terms of both DB12c and Solaris operating system commands, monitors, and reports. Ultimately, however, the testing metrics were reduced to the two most well-known performance factors associated with database performance – 'throughput' and 'response time:'

- **Throughput:** The amount of data processed in a given timeframe, typically measured in "transactions per second" (TPS).
- **Response Time:** Measures the speed at which external client requests are handled by the database, typically measured in milliseconds (MS). This metric is also often referred to as "latency" since it reflects the wait time a client experiences between request and response.

Given both the rich set of capture and reporting/analysis tools the Oracle testing team had at its disposal to measure throughput and response, Oracle made appropriate choices in terms of measurement tools and metrics for its consolidation density testing.

Test Infrastructure, Workload, and Utilization Design

Test Infrastructure

Oracle's approach for choosing an infrastructure for its DB12c density testing was to pick a pre-configured solution that Oracle would typically recommend for organizations interested in database consolidation. Oracle picked its own Oracle SuperCluster T5-8 ("SuperCluster"). Oracle designed SuperCluster specifically to optimize performance for database consolidation. The SuperCluster includes the following components:

- **Server:**
  - Processors: 8 sockets, 16-cores/socket, 3.6 GHz SPARC T5 processor (128 cores total).
  - Memory/Cache: 2TB (total). L1 16KB data + 16KB instruction cache per core, L2 128K shared data and instruction cache per core, L3 8M cache per socket shared among cores.
  - Operating System: Solaris 11 Update 1 SRU 16.
- **Storage Servers and Network:** 8 X3-2 Exadata Storage Servers, Software version OSS 11.2.3.3.0 connected with the compute node via InfiniBand.

Oracle picked a pre-release database software candidate, Oracle 12.1.0.2.0, for the test. This candidate is close to being production ready, and Oracle felt that its performance characteristics would not vary from currently available database versions. Given that the hardware, operating system, and database software choices were typical of what a customer might purchase for consolidation purposes, particularly going forward, Oracle made appropriate choices for the test bed.

Note: Select screenshots illustrating some of the infrastructure and database configuration, and test results are available in Appendix I.

Workload

Oracle decided to use mainly transactional workloads, thus the test results herein reflect a primarily write-oriented application environment. Oracle databases are often used for OLTP applications, so the decision to test with workloads that generated nearly 3/4ths, 72% to be exact, writes makes sense Read-oriented database workloads, such as data warehouses, are also good candidates for consolidation, but, as noted, the focus of this audit is on OLTP – a more mission critical choice. Note that write transactions, which include insert, update, and delete primitives, also include the writing of a redo log for data protection and recovery purposes. Thus write-oriented tests typically carry a natively heavier I/O load than more read oriented tests. The exception for this assertion is when the read-oriented test involves distributed processing for highly complex, typically join-oriented queries, which was not part of this study's test bed.

CPU Utilization and Test Length

Oracle intentionally designed tests to operate at CPU utilization in the realistic 50-to-70% range. Oracle also intentionally ran the tests for longish periods of time, such as a half hour, to ensure the database performance test was not unduly influenced by the test start-up which carries different performance dynamics than a long-running production database. Oracle captured metrics separately for the warm-up period versus production steady state.

In a consolidation scenario it is highly unlikely that all consolidated databases would operate at 70% utilization simultaneously. Therefore there is good reason to infer that customers, as reflected in test 4, will typically experience even better results in an environment consolidated with Multitenant than is reflected by these test results.

The Four Tests

Oracle created four tests to simulate several consolidation scenarios, including:

1. **Baseline:** Oracle refers to this as its "cost" test, and the approach is to measure performance of a CDB with a single PDB versus a single non-CDB. Why bother with this non-consolidation example? The test establishes the baseline performance "cost" of the Multitenant approach compared to the traditional non-CDB approach. With the test Oracle hoped to prove that there is minimal overhead associated with the 1 PDB/CDB option versus 1 non-CDB. By inference, if the single PDB/CDB versus non-CDB test came out roughly even, it suggests that the Multitenant option does not carry an appreciable processing penalty even when compared to a single standalone database. To ensure that this holds true also for large databases, Oracle ran this test against a database utilizing almost a full socket, much larger than the databases used for evaluation of consolidation density.
2. **Efficiency:** Oracle deployed 252 PDBs using a single CDB, versus 252 non-CDBs. Oracle measured the performance of each scenario in terms of background processing to determine which scenario made optimal use of, for example, memory. The "winner" between the PDBs versus non-CDBs would be the test that exhibited (a) higher aggregate throughput – the throughput across all databases, and (b) faster response time.
3. **Density:** Oracle set minimum throughput requirements for each of the PDBs or non-CDBs. Oracle then loaded as many user databases on SuperCluster as possible, stopping when the CPU utilization targets were reached, and when no more databases could be added without violating the minimum throughput requirements of the other databases. The determining metric was the number of user databases, PDBs or standalone/non-CDBs, which would operate at the target throughput levels and target CPU utilization on SuperCluster.

4. **Elasticity:** One of the key notions of cloud computing or virtualization is elasticity – that a cloud service is able to contract and expand to meet usage peaks and valleys. In the first three tests Oracle maintained a constant load, but in this final test Oracle attempted to reflect cloud reality by varying the loads over the course of the test. 8 PDBs in a single CDB were matched against 8 non-CDBs. Oracle accomplished this test by applying sine curves of load for each of the 8 databases and ensuring that the curves were out of phase with one-another. The expectation in this test is for Multitenant to flex effectively enough to equal the throughput of individual non-CDBs that manage their own dedicated set of background and system resources.

Database Size Consideration

Oracle did not use exactly the same database sizes for all four tests. For example, in tests 2 and 3, Oracle split the number of PDB and non-CDB databases proportionally across 3 sizes, small (1 GB), medium (5 GB), and large (15 GB). While certainly in a production consolidation situation database sizes will be all over the place, and the acceptable user visible response requirement will vary by application associated with each PDB or non-CDB, Oracle's use of different database sizes is indicative of true customer environments. Given that every organization will possess its own unique mix of database sizes, it is recommended that, for example, in a proof of concept for DB12c, that the DBA or project manager choose a representative sample of databases in terms of size, as well as read versus write tendencies.

Additional Technical Detail

For those readers with deep Oracle database technical acumen, Oracle has published a white paper describing the specifics of these tests. In that white paper Oracle goes into deeper explanations as to why the results came out the way they did, and also exposes a wider array of related metrics. The purpose of the paper you are reading now is to validate in general the Oracle testing approach, to underscore the legitimacy of the results, and to offer general guidance for CIOs and DBAs. If, however, you want to understand the detailed effects of, for example, buffer cache, the allocation of memory pools, and the number of storage IOPS associated with these tests, please consult the Oracle white paper.

Oracle DB12c Consolidation Density Test Results

Test 1: Baseline (Cost) Test

Database(s)	Throughput	Response Time (Average)	CPU Utilization
1 non-CDB	16,000 TPS	4.8 ms	52%
1 CDB, 1 PDB	16,000 TPS	5..0 ms	54%
Key Takeaway: There is effectively no performance difference between a non-CDB standalone database architecture compared to a Multitenant architecture in the case of a single PDB (single tenant) running in a CDB. The test exhibited a slight advantage to non-CDB in terms of response times, however, it is likely that lighter workloads would result in a smaller response time difference. It is also highly unlikely that users would notice the minor response time difference.			

Test 2: Efficiency Test

Database(s)	Aggregate Throughput	Response Time (Average)	CPU Utilization
252 non-CDBs	72,600 TPS	6.7 ms	68%
1 CDB, 252 PDBs	130,300 TPS	9.9 ms	68%

Key Takeaway: The Multitenant consolidation approach yields considerably higher – nearly double – throughput versus the non-CDB/standalone database consolidation approach.

The throughput results are underscored by far fewer storage IOPs for Multitenant, by a factor of less than half, and a much smaller aggregate memory footprint for Multitenant by about an 8x factor. The response time difference for non-CDBs of less than 3 milliseconds is a tiny price to pay for such dramatic throughput improvements. The response time difference was determined to be directly associated with log file synchronization requirements associated with log writing processes. A consolidation mix with a lower OLTP concentration (i.e. more read, less write) should effectively eliminate the response time difference. Note that database sizes were varied into 3 different sizes (small, medium, large) but CPU utilization was kept consistent.

Test 3: Density Test

Database(s)	Aggregate Throughput	Response Time (Average)	CPU Utilization
168 non-CDBs	Fixed	7.3 ms	68%
1 CDB, 252 PDBs	Fixed	9.92 ms	68%

Key Takeaway: Multitenant was the winner by a wide margin over the non-CDB approach, with 50% more database consolidation density operating in a shared infrastructure where throughput (TPS) for each individual database was constant for both environments.

This test best exemplifies a true consolidation scenario, with different database sizes operating at a pre-defined target throughput requirement per database. Non-CDB response time was, on average, less than 3 milliseconds better than Multitenant which is not user visible, but (a) Multitenant was able to handle 50% more user databases (PDBs), and (b) again the response time differential may be traced directly to the high write concentration of the test scenario.

Test 4: Elasticity Test

Database(s)	Aggregate Throughput	Response Time (Average)	CPU Utilization
8 non-CDBs	80,340 TPS	13.8 ms	Non-linear*
1 CDB, 8 PDBs	80,440 TPS	11.4 ms	Non-linear

Key Takeaway: In a consolidated environment requiring elasticity, the Multitenant database consolidation outperforms the non-CDB/standalone technique in terms of response time by applying a predictable, steady allocation of compute resources to deal with fluctuating demand.

The CDB's buffer cache was able to distribute resources smoothly across the shifting demands of each database effectively enough that it matched the standalone databases in terms of throughput, and pulled ahead in terms of response time. PDBs required about 20% fewer storage IOPs than the non-CDBs. Non-CDBs took longer to reach their configured peak load than the PDBs, so while the aggregate throughput appeared equal because non-CDBs were able to catch up, apparent throughput favored PDBs.

*Non-linear represents the sinusoidal curve of demand where the workload fluctuates over time, versus a constant demand workload.

EMA Perspective

The new Multitenant option of DB12c offers Oracle customers a long list of benefits, from easier and faster administration, to cloud deployment options that were not available in previous Oracle database versions. In particular, given the high level of interest in database consolidation, Multitenant offers customers an enticing approach to squeeze more databases onto the same infrastructure, without sacrificing throughput or response time. In fact, based on the tests conducted by Oracle engineers, it appears that in consolidation scenarios, throughput is significantly improved with Oracle Multitenant as compared to consolidated non-CDBs, and when throughput is fixed the consolidation density of Oracle Multitenant option consolidation far outstrips that of the non-CDB approach

Every customer will have a different experience depending on their particular infrastructure choice and database mix, but regardless, based on the tests reviewed herein, DB12c with the Multitenant option stands out as a strong candidate for database consolidation. Though the Oracle SuperCluster engineered system also appears to be an excellent candidate to support DB12c for consolidation purposes, the advantages of Multitenant consolidation would likely be realized on other infrastructure solutions as well.

Though not tested, from these tests, it is implicit that the Multitenant option will also achieve better density and performance results versus a consolidation scenario using virtual machines (VMs). The VM consolidation approach carries the overhead of multiple OSes, hypervisors, and individual Oracle database binaries – all in addition to the individual background processes that each standalone database must handle. Based on the audited Oracle tests, DB12c with the Multitenant option should do well in consolidation scenarios concentrating on I/O intensive OLTP workloads, but there is also good reason to believe that consolidated workloads using the Multitenant option that involve more of a mix of intensive read/write and read-oriented applications will perform even better and/or enjoy higher consolidation density with Multitenant.

Oracle Database 12c with the Multitenant option will enable CIOs and DBAs across a broad array of organizations to control infrastructure costs while effectively addressing the ever-growing data demands of their business.

The Multitenant database option for Oracle Database 12c gives CIOs and DBAs a highly promising approach to control or even lower fully-loaded database management costs through database consolidation. Oracle Multitenant doesn't merely do an excellent job improving consolidation density, it also potentially will improve throughput and effectively handle elasticity, while simultaneously decreasing the complexity of administration. The net effect of the CDB is better utilization of processors and memory, a startling decrease in storage IOPS, and effective allocation of buffer cache to support consolidation scenarios demanding elasticity. Oracle Database 12c with the Multitenant option will enable CIOs and DBAs across a broad array of organizations to control infrastructure costs while effectively addressing the ever-growing data demands of their business.

Appendix I. Select Screenshots

NOTE: Proprietary hostnames and IPs have been deliberately obfuscated at the request of Oracle.

Database Version

```
Oracle Interim Patch Installer version 12.1.0.1.2
Copyright (c) 2013, Oracle Corporation.  All rights reserved.


Oracle Home       : [redacted]
Central Inventory : [redacted]
   from           : [redacted]
OPatch version    : 12.1.0.1.2
OUI version       : 12.1.0.2.0
Log file location : [redacted]

Lsinventory Output file location : [redacted]

--------------------------------------------------------------------------------
Installed Top-level Products (1):

Oracle Database 12c                                                  12.1.0.2.0
There are 1 products installed in this Oracle Home.


There are no Interim patches installed in this Oracle Home.


--------------------------------------------------------------------------------
```

Operating System Version

```
Oracle Corporation      SunOS 5.11      on11update-bugfixes     February 2014
You have new mail.
[redacted]
SunOS [redacted] 5.11 on11update-bugfixes sun4v sparc sun4v
root@[redacted]:~# pkg info kernel
          Name: system/kernel
       Summary: Core Kernel
   Description: Core operating system kernel, device drivers and other modules.
      Category: System/Core
         State: Installed
     Publisher: on-nightly
       Version: 0.5.11
 Build Release: 5.11
        Branch: [redacted]
Packaging Date: February 26, 2014 09:53:44 PM
          Size: 13.40 MB
          FMRI: [redacted]
```

Test 3 Density – CDB/PDB Results

Transaction Summary

Level 2	Time	Warmup1 22:24:25 - 22:34:25 (600.018 s)	load - stop 22:34:25 - 22:54:25 (1200.057 s)			
	TX Throughput [/s]	79661.34	130259.06			
	Avg. TX Resp. Time [ms]	7.40	9.92			
	Successful TX	47816661.00	156387028.00			
	Failed TX	0.00	0.00			
	Failure Rate [%]	0.00	0.00			
Level 3	Time	Warmup1 22:24:25 - 22:34:25 (600.018 s)	load 22:34:25 - 22:39:25 (300.015 s)	snap1 22:39:25 - 22:44:25 (300.012 s)	snap2 22:44:25 - 22:49:25 (300.015 s)	snap3 22:49:25 - 22:54:25 (300.015 s)
	TX Throughput [/s]	79661.34	130302.92	130188.72	130276.43	130268.17
	Avg. TX Resp. Time [ms]	7.40	10.15	9.74	10.13	9.81
	Successful TX	47816661.00	39110362.00	39075628.00	39101809.00	39099229.00
	Failed TX	0.00	0.00	0.00	0.00	0.00
	Failure Rate [%]	0.00	0.00	0.00	0.00	0.00

Test 3 Density – Non-CDB Results

Transaction Summary

Level 2	Time	Warmup1 00:13:25 - 00:23:25 (600.019 s)	load - stop 00:23:25 - 00:43:25 (1200.058 s)			
	TX Throughput [/s]	53136.72	86910.90			
	Avg. TX Resp. Time [ms]	5.43	7.33			
	Successful TX	31890342.00	104321288.00			
	Failed TX	1.00	10.00			
	Failure Rate [%]	0.00	0.00			
Level 3	Time	Warmup1 00:13:25 - 00:23:25 (600.019 s)	load 00:23:25 - 00:28:25 (300.016 s)	snap1 00:28:25 - 00:33:25 (300.015 s)	snap2 00:33:25 - 00:38:25 (300.013 s)	snap3 00:38:25 - 00:43:25 (300.014 s)
	TX Throughput [/s]	53136.72	86910.69	86910.89	86910.96	86911.05
	Avg. TX Resp. Time [ms]	5.43	7.32	7.30	7.34	7.34
	Successful TX	31890342.00	26080296.00	26080353.00	26080324.00	26080315.00
	Failed TX	1.00	2.00	2.00	3.00	3.00
	Failure Rate [%]	0.00	0.00	0.00	0.00	0.00