Two Billion Entry Directory Benchmark

Oracle Internet Directory v10.1.4.0.1

Performance Engineering Group
Persistent Systems Ltd.

March 2008

Performance Engineering Group
Persistent Systems Ltd.
Bhageerath, 402 Senapati Bapat Road
Pune 411 016, India
Web: www.persistentsys.com
Tel: +91 20 2567 8900 Fax: +91 20 3023 4001

PERSISTENT

Table of Contents

Table of Figures

Table of Tables

1 Executive Summary

Directory Servers are core components of Identity Management (IdM) infrastructures, responsible for the real-time delivery of identity data to every modern enterprise application. Inadequate performance or scalability shifts directories from being an *enabling* component of enterprise business to being a *limiting* component.

Oracle Internet Directory (OID) is Oracle's LDAP V3 compliant directory server. OID is designed for high-performance, high-availability applications using Oracle Database for data storage. OID deployments can be scaled across a range of hardware architectures, including SMP, NUMA, and Clusters.

This benchmark evaluates OID performance under scales and throughput rates representative of those encountered in production use in both the communications industry and as part of large scale service delivery architectures (e.g., those supporting social networking or e-commerce sites).

The benchmark was conducted on a single OID directory instance with **two billion** entries, each entry containing 19 attributes. Benchmarks were performed using SGI Altrix 4700 server, with 32 dual core Itanium2 CPUs (1.6GHz) and 256 GB RAM, running Linux.

The benchmark demonstrated:

- Over 100,000 operations per second with 2.5 ms average latency
- Scalability to over 16,000 concurrent clients
- High speed data load – five hours online data load and 19 hours post-load index creation

This benchmark is currently the only published benchmark that demonstrates directory server scalability to two billion entries. The results suggest that OID is currently the only viable choice for very large-scale directory deployments.

This benchmarking exercise was carried out by Persistent Systems' **Performance Engineering Group** (PEG). Persistent Systems (http://www.persistentsys.com/) is a leading offshore software product development services company. Focusing on the outsourced product development space, Persistent is the software development partner for many Fortune 500 Independent Software Vendors.

Persistent PEG provides independent vendor-neutral software and system performance evaluation, benchmarking, and certification services as part of its Performance Engineering Practice. Hardware for the benchmark was made available at Silicon Graphics' Global Benchmark Centre.

Oracle engineering personnel assisted with the tuning and configuration of the Oracle Internet Directory installation used as the benchmark target. Benchmark execution, data collection, analysis, and reporting were the responsibility of Persistent PEG.

2 Business Drivers and Requirements

2.1 Business Drivers

Recent business and technology trends are converging with respect to the demands they place on the technology that supports and enables modern day businesses. In an increasingly globalized marketplace where companies and businesses are no longer restricted to single countries or continents, competitive advantage can derive from the ability to seamlessly cross classification boundaries in the delivery of services and products to customers.

Identity Management (IdM) solutions tie together the various vendors, service providers, and end customers into global supply chains, transnational commerce platforms and worldwide customer bases. These solutions need to scale up to qualitatively higher levels of complexity, scale, and performance.

Directory servers are critical components of the IdM solution set. Scalability and performance requirements for IdM solutions naturally impose corresponding requirements on directory server infrastructure. While a chain is only as strong as its weakest link, it is increasingly critical that directory services do not become the bottleneck for identity management systems.

2.2 Deployment Scale Landscape

It is instructive to consider some illuminating data points drawn from multiple industries and application areas that point to the scale of IdM solutions required today and that will be required tomorrow:

- Research firm Wireless Intelligence (A joint venture of Ovum and the GSM Association) predicts that global cellular phone connections will touch 3 billion by the end of 2007. This represents an increase of 500 million worldwide since September 2006. China will cross 500 million connections in Q3 2007, with China Mobile alone accounting for 70% of market share, or 350 million subscribers.

- At the end of October 2007, cellular phone subscriber numbers for India had crossed 213 million. The top five Indian cellular operators count between 20 million to 50 million+ subscribers. The Telecom Regulatory Authority of India has directed the implementation of cross-operator cellular number portability starting Q1 2008. This is expected to lead to a qualitatively higher scale of cell number and subscriber tracking infrastructure as operators are effectively required to see a unified subscriber base.

- During their 2006 Annual User Conference, online auction giant eBay announced it had 200 million registered users. This does not include internet properties like rent.com or shopping.com which are owned by eBay, or the free internet telephony company Skype which was acquired by eBay and claims a 100 million registered users of its own.

- Security concerns and regulatory and law enforcement requirements are leading airlines to track and maintain greater levels of passenger information. In addition, attempts to reduce costs owing to delayed and lost luggage are driving the adoption of smart tagging based luggage tracking solutions. The International Air Travel Association (IATA) projects the number of air passengers (domestic and international) to touch 2.75 billion by 2011. International passengers are projected to rise from 760 million in 2006 to 980 million by 2011.

Note that the numbers cited above are for *numbers of users.* Individual pieces of user-relevant information that needs to be maintained in individual business contexts (such as SMS/MMS service specifications for cell phone users, multiple avatars or screen names for social networking sites, group memberships or frequent flier program information etc) can translate into many times that number of entries in directory terms. Thus, requirements for DIT size in the billions are already on the horizon.

Individually, these numbers would give pause for thought; taken together, they point to a consistent trend, across widely different business domains and geographies, of a scale of operation orders of magnitude greater than what was typical only a few years ago. As worldwide growth continues to be fuelled by emerging markets where technology penetration has been low, the demands for scalability only increase.

2.3 Business Induced Performance and Scalability Requirements

Business requirements drive demands on directory server performance in multiple ways; as the directory server is typically on the critical path in multiple business process or user scenario driven workflows, directory server performance in terms of response times for LDAP operations and the ability to support high levels of concurrent client access are as important as the ability to support extremely large DIT sizes. A crucial requirement is that directory performance in response time/concurrency terms hold steady as DIT sizes increase.

Scalability with respect to very large DIT sizes is qualitatively different as typical quoted performance numbers for directory benchmarks depend to a great degree on caching directory entry working sets to optimize search or lookup performance; for moderate sized DITs this represents a good operations/CPU scalability metric. However, for very large DITs (such as were used in this benchmark, which are an order of magnitude larger than in most benchmarks that are currently available) the efficiency of caching rapidly falls off as typical entry working sets, even with locality of reference, exceed what can reasonably be cached. In production this effect can manifest as a sudden, sharp, drop in operation throughput (and a corresponding increase in operation latency) when the DIT crosses a size threshold, with adverse consequences for business process availability.

An orthogonal requirement is the ability to exploit available hardware resources (in particular CPU) and scale as additional capacity is brought online. With the maturing of models of computing capacity provisioning on demand, the shift towards multi-core CPU architectures is at the initial stages.

The continual pressure to optimally provision and operate business information systems infrastructure, the ability to seamlessly scale up, without unreasonable effort to re-configure, tune, or optimize is particularly important.

2.4 Directory Servers: Current Performance Spreads

There are a variety of LDAP directory server products available in the market, ranging from free Open Source servers to high-end server products from leading software vendors. In essence, the basic functionality and performance in the directory server space appears to be commoditized.

However, with the increasing importance of identity management infrastructures as well as the increasing scale of required directory deployments, a sharper differentiation in product capabilities is becoming apparent, particularly at extreme scales of directory size or demanded performance.

At the scale of DIT sizes described in this benchmark, most directory products cannot claim to even support DIT sizes in this range, leave alone with high-sustained LDAP operation rates. In this sense, Oracle's Internet Directory can be said to have "de-commoditized" the LDAP directory space. This is particularly apparent at "pushing the envelope" or billion entry sized directory. While authoritative comparisons are not easily available, currently available information suggests *at least* an order of magnitude difference in supported DIT scales.

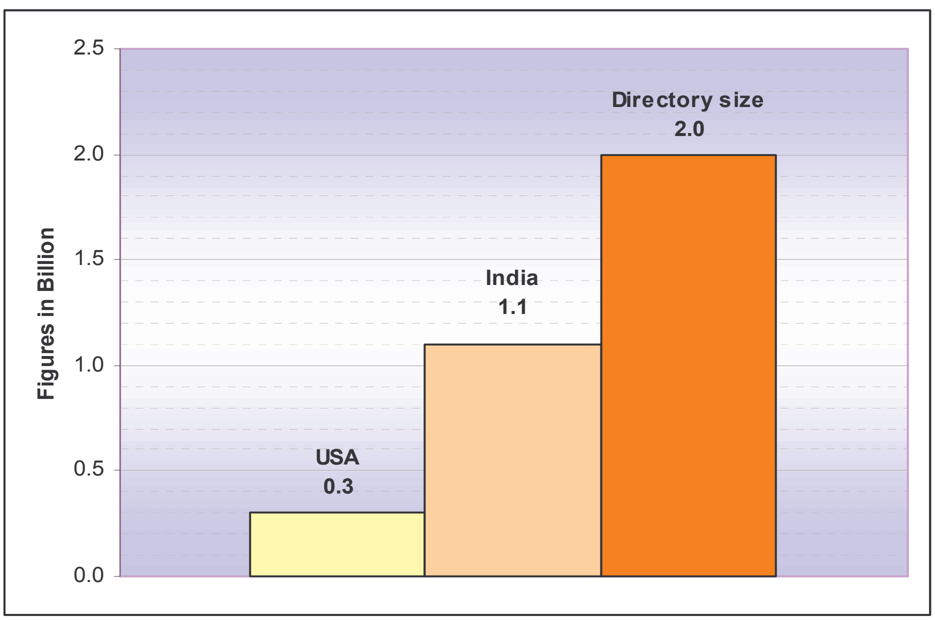

Figure 1 Comparison of Directory size with approximate populations of USA and India as of January 1, 2008

2.5 Directory Servers: Future Performance and Scalability Expectations

Current business drivers are pushing requirements for DIT size into ever-larger regions. In addition, current processor trends are for an increasing number of cores per processor chip becoming available, while processor clock rates remain essentially flat, or increase less slowly than has historically been the case. These independent trends converge in a requirement that directory server software scale to very large scale DIT sizes exploiting all available processing resources, implying linear scalability with the number of available cores.

3 Benchmark Findings

3.1 Benchmark Design

The benchmark was designed to evaluate OID performance under DIT scales and throughput rates representative of what would be demanded of a large-scale directory server deployment in production. In addition to core benchmark metrics of operation response times and throughputs, the benchmark also assessed OID performance on metrics important in the context of operation, specifically setup and provisioning.

The benchmark was conducted on a single OID directory instance with **two billion** entries, each entry containing 19 attributes. Benchmarks were performed using SGI Altrix 4700 server, with 32 dual core Itanium2 CPUs (1.6GHz) and 256 GB RAM, running Linux.

3.2 Benchmark Results

Five LDAP load operations were tested including search, bind, authenticate (search and bind), update, and compare. Some operations were tested under different level of workload. Key results from the benchmark are presented in the table below:

Test Scenario	Concurrent Clients	Throughput (per sec)	Latency (ms)	CPU Utilization (%)	Duration (Min)
Search (low latency)	240	99,669	2.4	98	30
Search (peak throughput)	800	101,262	7.9	99	30
Search (high client concurrency)*	16,000	99,425	160.8	100	15
Authenticate (search + bind)	240	36,812	6.5	76	30
Authenticate (search + bind)	800	36,648	21.8	83	30
Bind	800	80,378	9.0	98	15
Update	400	13,900	16.0	38	30
Compare	800	83,761	9.5	99	30

* Due to the limitation of client software scalability, we were only able to test up to 16K concurrent clients.

Table 1 Benchmark Results Summary

The results clearly demonstrated:

- High throughput of over 100,000 operations per second
- Low latency of under 2.5 ms, and
- Scalability to over 16,000 concurrent clients

3.3 Ease of Installation/Configuration

At install time, default installation options were chosen; no specific tuning or optimization driven parameter settings were specified. Subsequent changes to OID configuration parameters, as suggested by initial performance experiments and in consultation with Oracle personnel are documented in the report. OID installation was straightforward and did not raise any issues or run into problems.

3.4 High Speed of Bulk Data Provisioning

Oracle Internet Directory is one of the few LDAP servers to use a full-fledged database engine as the data store. As a consequence, the directory server benefits directly from all the performance related design and optimization effort that has gone into the Oracle database engine.

Provisioning billion-entry sized DIT in OID was straightforward and efficient. OID employs a two-stage bulk loading process. In the first stage LDIF files are converted to a proprietary format suitable to fast loading into the Oracle database backend; the second stage constitutes the actual bulk loading of data to the database. The two-stage approach allows available processing capacity to be exploited, as stage one can be parallelized and performed offline. End to end provisioning times with stage wise time breakdowns are documented in Section 4.2.3.

4 Technical Details

4.1 Benchmark Design

This section focuses on the design details of the benchmark, primarily on hardware details, load generation framework and various configuration parameters for Oracle Internet Directory 10.1.4.0.1, the database and of operating system (Suse SLES9). Details about the usage of SLAMD for load generation follow the "Load Generation" section.

4.1.1 Workload Model

4.1.1.1 Workload Rationale

The benchmark workload was defined based on the desire to independently characterize each LDAP operation's performance characteristics under a two billion entry DIT size. The operations selected as individual benchmark workload components capture those LDAP operations that dominate directory usage in typical directory deployments; this observation is essentially application domain neutral and has been validated across a range of actual directory deployments spanning multiple size ranges and domains.

While a specific application domain may see a varied mix of LDAP operations in the workload it is exposed to, this is not an issue here as each individual operation is characterized independently. A workload with a variable mix of LDAP operation components was deferred for later investigation as result of time and hardware availability constraints.

The specific LDAP operation scenarios that comprised the workload are documented below.

The DIT consisted of two billion entries, with each entry containing 19 attributes including operational and user attributes and two objectclasses. Five LDAP load operation tests were identified. The six test scenarios are explained below.

4.1.1.2 Workload Scenarios

- **LDAP Search Operations Test**
  This test scenario involved concurrent clients binding (i.e. authenticating) once to OID and then performing repeated LDAP Search operations. The salient characteristics of this test scenario are as follows –
  - *Bind DN*: cn=orcladmin (root of the DIT)
  - *Search Scope*: SUBTREE
  - *Search Base*: ou=people,dc=com
  - *Search filter*: uid =t[1:2B]
  - *Attributes to Return* uid
  - Each LDAP search operation matches a single entry

  - All SLAMD (see 4.1.2.1) clients use the same range, specified in the filter above. Entries are accessed in sequential order

- **LDAP Authentication Operations Test**
  This test scenario involved concurrent clients binding once to OID and then performing repeated LDAP authentication operations. Each LDAP authentication operation involved an LDAP search operation followed by an LDAP bind operation. The salient characteristics of this test scenario are as follows –
  - *BIND DN*: uid=t[1:2B], ou=people,dc=com
  - *User Search Base*: uid=t[1:2B],ou=people,dc=com
  - *Login ID Attribute*: userpassword
  - Each LDAP search operation matches a unique entry; no two clients look up the same LDAP entry.
  - All SLAMD threads access the entries within its unique range in sequential order

- **LDAP Bind Operations Test**
  This test scenario involved concurrent clients performing LDAP bind operations as different directory users. The salient characteristics of this test scenario are as follows –
  - A custom script was used to simulate a bind operation against the directory server as a random user.
  - *BIND DN*: uid=t[1:2B], ou=people,dc=com
  - 80 Clients used each client with 10 threads (total 800 threads), binding as random user in its range.
  - No two clients bind as the same user.

- **LDAP Modify Operations Test**
  This test scenario consisted of concurrent clients binding once to OID and then performing repeated LDAP Modify operations. The LDAP Modify operation is a 'replace' operation on an attribute with a single value. The salient characteristics of this test scenario are as follows –
  - Modify-replace of telephonenumber attribute with random value
  - 40 Clients with 10 threads each thread modifying entry in its range.
  - No two clients modify the same entry

- **LDAP Compare Operations Test**
  This test scenario involved concurrent clients binding once to OID and then performing repeated LDAP Compare operations. The salient characteristics of this test scenario are as follows –
  - *Bind DN*: cn=orcladmin (root of the DIT)
  - *Entry DN*: uid=t[1:2B],ou=people,dc=com
  - *Attribute to Compare*: userpassword
  - *Value of Attribute*: welcome1

- **LDAP High Client Concurrency Test**
  This test scenario involved concurrent clients binding once to OID and then performing repeated LDAP Search operations. The salient characteristics of this test scenario are as follows –
  - *Bind DN*: cn=orcladmin (root of the DIT)
  - *Search Scope*: SUBTREE
  - *Search Base*: ou=people,dc=com
  - *Search filter*: uid =t[1:2B]
  - *Attributes to Return* uid
  - Each LDAP search operation matches a single entry
  - All SLAMD clients use the same range, specified in the filter above. Entries are accessed in sequential order
  - Due to the limitation of client software scalability, we were only able to test up to 16K concurrent clients.

4.1.2 Load Generation

4.1.2.1 Software Framework

Load generation for the OID two Billion Benchmark was performed using the SLAMD Distributed Load Generation Engine (http://www.slamd.com/). SLAMD is an open source load generation framework which was originally designed and implemented (by Sun Microsystems) with the specific intention of providing a load generation mechanism for LDAP servers. While it has subsequently been expanded to support a variety of load generation requirements, it retains its LDAP origins through a set of LDAP-specific load generation jobs which make it particularly easy to construct load scenarios for LDAP directory servers.

SLAMD version 2.0.0-alpha1 was used for the load generation. The following jobs available as part of SLAMD's set of standard jobs were used as starting points:

- LDAP Auth Rate
- LDAP SearchRate
- LDAP CompRate

SLAMD standard job code (in Java) was modified where necessary to match required workload characteristics (for example, the LDAP CompRate job was modified to accept the value of the compared attribute, the value being the same for all entries).

4.1.2.2 Load Generation Hardware

The load generation substrate consisted of eight client machines with identical hardware specifications as follows:

Server Model	**SGI XE 310**	
Processor	2 x 2.66GHz QuadCore Xeon Processors (5300 Series)	
Memory	16GB (DDR2 667 FSB-DIMMs)*	
Network Interface	2 x 1000Base-TX Ethernet	
Operating System	SuSE Linux Enterprise Server 10 (SLES10 SP1)	

Table 2 Load Generation Hardware

*One Load Generation Client came fitted with 32 GB Memory.

4.1.2.3 Network Configuration/Topology

An isolated private gigabit network was established to provide connectivity between the eight client load generators and the OID server. Each client system was connected into the 24-port bandwidth switch using a single 1000Base-TX link. The OID server was configured with eight 1000Base-TX interfaces into the same switch. The benchmark network was dedicated to benchmark purposes; no network traffic other than load generation requests and OID responses was allowed during benchmark experiment execution.

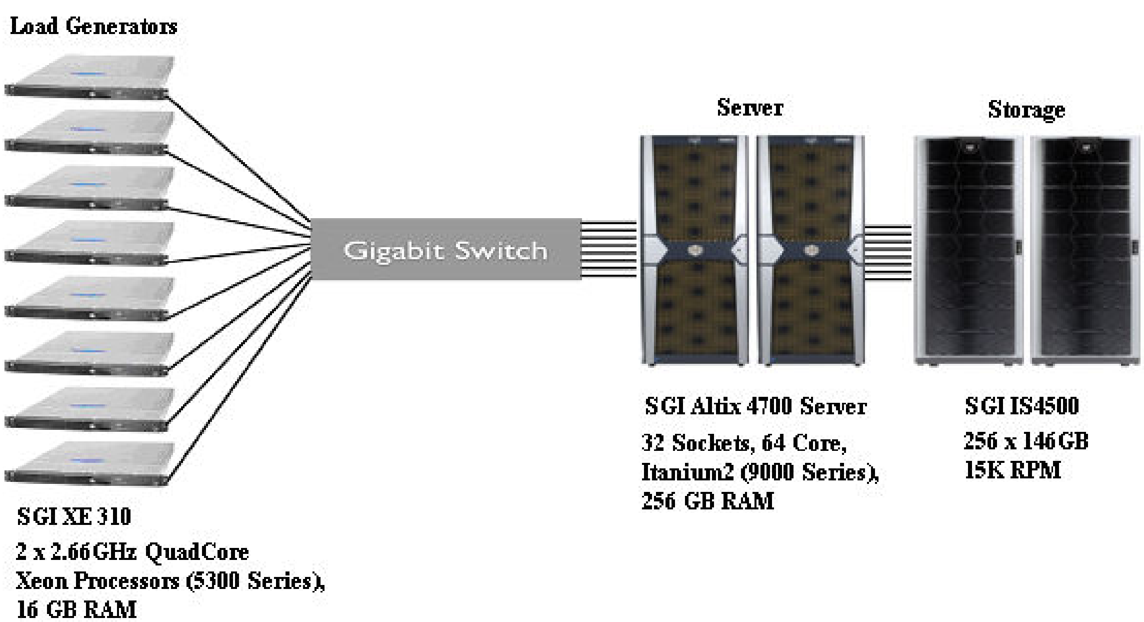

Figure 2 Network Configuration/Topology

4.1.2.4 Oracle Internet Directory 10.1.4.0.1 Server Testbed

The Oracle Internet Directory Server, the target of the benchmark, was installed on a SGI Altix server with specifications as described below. The hardware that was made available to deploy OID software consisted of a single node with access to a shared storage.

Server Hardware Node Specifications:

Server Model	**SGI Altix 4700 Server**	
Processors	32 Sockets, 64 Core, Itanium2 (9000 Series), 1.6 GHz, 9 MB L3 cache, CPU front side bus speed – 667 MHz	
Memory	256 GB RAM (DDR2-400 DIMMs)	
Network Interface	Dual-port Gigabit NIC Cards (PCI-X)	
Bus Adapters	Dual-port 4Gbit LSI FC HBAs (PCI-X)	
I/O Blades	4 x PCI-X IO Blades 1 BaseIO Blade	
Operating System	OS SuSE LINUX Enterprise Server 9 (SLES9 SP3)	

Table 3 Server Hardware Node Specifications

Storage Specifications: Two RAID Arrays

Model		**SGI IS4500**	
Disk Enclosures		8	
Fibre Channel Drives		37 TB (256 x 146GB) 15K RPM	
Channel	**Host**	8 x 4Gigabit (2 active)	
	Drive	4 x 4Gigabit (2 active)	
Links		Each host has 4 Fibre Channel 4 Gbs Links	

Table 4 Storage Specifications

4.1.2.4.1 Testbed System Configurations

4.1.2.4.1.1 File System Configuration

File system was not mounted as a UNIX file system; raw partitions were used instead. However, for purposes of installing software, creating LDIF files and intermediate files for bulk loading, an xfs file system was used with 4.7 TB which was mounted as /bench.

Raw partition configuration details are documented in Appendix A.

4.1.2.4.1.2 OS Kernel Configuration

Kernel parameters for the underlying OS kernels were carefully chosen from the point of view of maximizing underlying OS performance. A total of 11 parameters were changed, the details of which are given in Appendix B

4.2 Benchmark Execution

4.2.1 Performance Testbed

OID was installed on a target server (details given in section 4.1.2.4)

4.2.2 System Install

OID v10.1.4.0.1 and RDBMS v10.2.0.3 was used. OID/Database was installed in standard way using the Oracle universal installer.

4.2.3 Provisioning

4.2.3.1 Bulk Load of Data into OID

In this section the data characteristic, data load and attendant information is enumerated.

4.2.3.1.1 Data Characteristics

Some of the salient characteristics of the LDIF data that was loaded into OID are as follows:

- The LDIF data was generated using SLAMD LDIF generation tool
- Each entry consisted of 11 attributes including 2 objectclasses
- LDIF files were generated in two parts, first for 0-1B and then from 1B-2B. For each phase 3 LDIF files were generated to speed the bulkload process:

- **Part 1**
    - 0-1bBenchmark_1.ldif (1 - 300000000 entries)
    - 0-1bBenchmark_2.ldif (300000001 - 600000000 entries)
    - 0-1bBenchmark_3.ldif (600000001 - 1000000000 entries)
    - The total size of the all 3 files was 225.5 GB.
    - This step took 12 hours

- **Part 2**
    - 1-2bBenchmark_1.ldif (1000000001 - 1300000000 entries)
    - 1-2bBenchmark_2.ldif (1300000001 - 1600000000 entries)
    - 1-2bBenchmark_3.ldif (1600000001 - 2000000000 entries)
    - The total size of the all 3 files was 225.5 GB.
    - This step took 12 hours

4.2.3.1.2 Preparing for Bulk Load

The pre-bulkload steps that are imperative are documented here.

- The out-of-box OID LDAP schema has several attributes 'indexed' (cataloged) by default. For the purposes of saving disk space and time to load data, those attributes that were not planned to be used in LDAP search filters were deleted using the OID 'catalog' tool.
- The disk space requirement for the "intermediate files" generated by OID 'bulkload' tool was estimated to be 1250GB and this was ensured to be available in /bench/IMFILES/ directory.
- The disk space requirement for the database data files was estimated to be 4.5TB and sufficient DB data files (with 'autoextend on') were created for the relevant OID tablespaces. See Table 1 for details.
- Large TEMP BIGFILE tablespace was created in the database to help during index creation. See Table 6 for details.
- Database tuning for bulkload

Parameter Name	Value
SGA_MAX	240G
SGA_TARGET	140G
PGA_AGGREGATE	96G
PROCESSES	3500
PARALLEL_MAX_SERVER	32

Table 5 Parameter values for Database Tuning

4.2.3.1.3 OID Bulk Load Execution

Data was loaded in bulk into OID in three steps. Three separate OID instances were installed to enable initial concurrent data loading. The bulkload was concurrently executed from these three OID nodes. This approach required some customization of the tools and procedure. The bulkload "generate" and "load" phases enumerated below was executed in parallel from the three OID instances for the six LDIF files generated above, three at a time.

- Step 1: The "generate" phase
  The first step is where the 'bulkload' tool will read the LDIF file, perform some basic validation and generate intermediate files in /bench/IMFILES/ directory. The intermediate files are used in the "load" phase. The invocation of the 'bulkload' tool for this phase is given in Appendix C.

  Only after the above invocation completes successfully and reports no errors whatsoever, we proceed to the next step. If there are any errors reported, we can fix those errors and re-execute the 'bulkload' tool as above.

- Step 2: The "load" phase
  Here the intermediate files generated in the prior phase will be used as input to Oracle SQL*LDR which loads the data directly into the Oracle Database. The database indexes are dropped prior to loading the data so that the data can be loaded in direct_path mode that essentially ensures the fastest load performance. The invocation of the 'bulkload' tool for this phase is as given in Appendix D.

- Step 3: The “index” phase
  The out-of-box behavior of ‘bulkload’ tool is to create the necessary database indexes in the ‘load’ phase itself. However, as noted earlier we customized the tool and procedure to enable concurrent bulkload from the OID node by having 3 LDIF files each for the 0-1B and 1B-2B entries. The customization of the ‘bulkload’ tool involved removing the database indexes creation during the “load” phase. Hence, all the required OID database indexes were created in a separate “index” phase using custom DML scripts. 2_create_indexes.sql & 3_create_indexes.sql were started at the same time.

Summary of Bulk Load Timings for the first Billion entries

- Offline Activity = 35 hours 47 mins
- Bulk Load “load” phase = 2 hours 28 mins
- Database Index creation = 16 hours

Summary of Bulk Load Timings for the second Billion entries

- Offline Activity = 33 hours 01 mins
- Bulk Load “load” phase = 2 hours 32 mins
- Database Index creation = 19 hours 41 mins

4.2.3.1.4 OID Database Size after Bulkload

The size of the various OID database tablespaces is documented in the table below. The size estimates were done prior to Bulk Load and datafiles were added where necessary. The actual size reflects the usage after data was loaded into OID.

OID Tablespace Name	*Description*	*Actual Size After Bulk Data Load*
OLTS_BATTRSTORE	Holds binary attributes data. In this benchmark, there were no binary attributes.	1 MB
OLTS_CT_STORE	Typically holds all the catalog tables and indexes. In this benchmark, we added the contents of olts_attrstore tablespace also into olts_ct_store so that we could create on bigfile for this tablespace and use one large raw partition for the bigfile.	4,300 GB
OLTS_SRVMGSTORE	Holds the OID server statistics information	1 MB
OLTS_DEFAULT	Used during bulkload operation and also to hold change log and other transient data.	1.6 MB
TEMP	Used during index creation, can be resized to a much lower value (to less than 5 GB) after the bulkload operation is complete.	520 GB
UNDO	Used during updates to the directory	16 GB

Table 6 OID Tablespace after 2B enteries were loaded in to OID

4.2.3.1.5 An important Post-Bulk-Load Step

Since the data is loaded into OID Database using Oracle SQL*LDR in Direct Path mode (which by the way is the most efficient way to load large data sets into an Oracle database), the first time that a data block is accessed after dataload is complete, Oracle database does a block clean out procedure which involves writing some meta data to the data block and flushing the data block back to the disk.

This means that after a fresh bulk load of data into OID, the first LDAP operation to access a data block will also initiate a write. That is not significant in a typical production roll out since this is a one-time occurrence and the complete DIT will incur this cost over a period of time. However, for a performance benchmark, this will have a noticeable impact in the initial tests.

The preferred approach is to bring up OID after a bulk load and to perform a test run of LDAP search operations that will touch all the LDAP entries in the DIT. And, if there is a requirement to setup replications, it is imperative that this step is done prior to the DB Copy or any other data migration step is performed as a part of replication setup. Note that this discussion is relevant to performance benchmarks only.

4.2.4 Experimental Strategy

Experience from actual OID deployments across a range of customers suggests that the most important operational parameter is operation throughput, for each operation type. Consequently, the benchmark philosophy was to evaluate the tradeoff between the supplementary metrics of concurrency (i.e. number of simultaneous clients supported) and operation latency, subject to a minimum threshold of operation throughput.

It was observed that beyond 240 clients, with increasing concurrency, the throughput for search operation was almost constant. This was verified by doubling the number of clients till 16,000 beyond which the load generation system becomes the bottleneck. From within this range, with an exception of Modify operation, a representative sample point of 800 clients was taken for measuring the throughput/latency values for other operation.

4.2.5 Tuning and Optimization

- **The Test Clients**
  SLAMD clients were used to generate LDAP operations load in most of the tests. A total of 8 nodes were available for the tests.

- **OID LDAP Instance Configuration**
  - ‘orclserverprocs’ was set to 72, this is the number of OID LDAP Server processes used in the OID LDAP instance.
  - ‘orclmaxcc’ was set to 1, this influences the number of threads and database connections used by OID LDAP Server processes.

- **OID Root DSE**
  - In the root DSE entry, the attribute ‘orclmatchdnenabled’ was set to 0 (zero) to disable detailed matching DN information

- **OID DSA Configuration**
  - In the entry “cn=dsaconfig,cn=configsets,cn=oracle internet directory”, the following attributes were updated –
    - ‘orclskiprefinsql’ was set to 1, this enables optimized SQL generation and can be used when LDAP referral entries are not present in the DIT as was the case in this benchmark. Presence of LDAP referral entries is not common.
    - ‘Orclmaxconnincache’ was set to 300000000.

- **Password Policies and Verifier Profiles**
  - The Password Policies on OID were disabled**.**

- **OID Change Log Generation**
  - OID Change log was disabled**.**

4.2.5.1 Database Redo Logs

Eight redo log files of size 1.4GB each were configured in two groups.

4.2.5.2 Database Tuning for OID TEST:

Parameter Name	Value
SGA_MAX	240G
SGA_TARGET	200G
PGA_AGGREGATE	2G
PROCESSES	3500
PARALLEL_MAX_SERVER	32
OPEN_CURSORS	1800
DB_CACHE_ADVICE	OFF
CURSOR_SPACE_FOR_TIME	TRUE

Table 7 Database Tuning Parameters for OID Test

4.3 Results and Analysis

#	Test Name	Duration (Minutes)	Directory size	Concurrent Clients	Throughput (per sec)	Latency (ms)	CPU Utilization (%)
1	Search	30	2B	240	99,669	2.406	98
2	Search	30	2B	800	101,262	7.896	99
3	Search	15	2B	16,000	99,425	160.761	100
4	Authentication (search + bind)	30	2B	240	36,812	6.517	76
5	Authentication (search + bind)	30	2B	800	36,648	21.826	83
6	Bind	15	2B	800	80,378	9.0	98
7	Update	30	2B	400	13,900	16.0	38
8	Compare	30	2B	800	83,761	9.546	99

Table 8 Summary of Results

4.3.1.1 Graphs

4.3.1.1.1 Search Operation

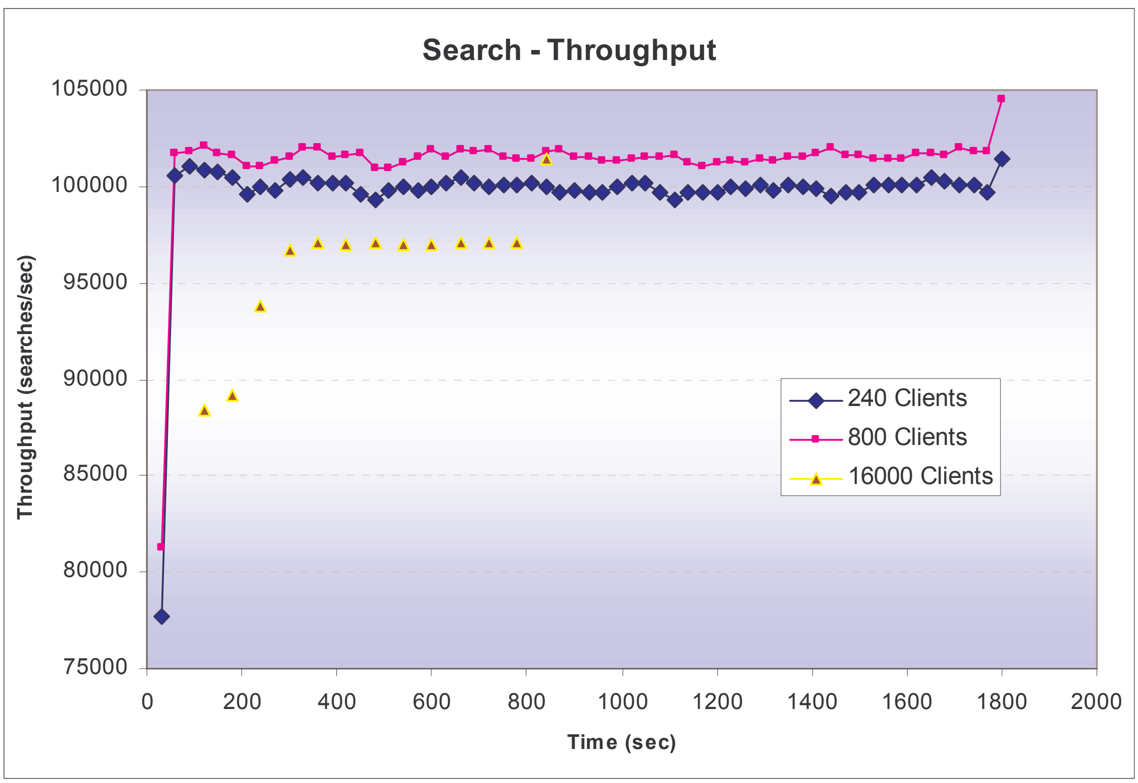

Figure 3 Throughput for Search Operation

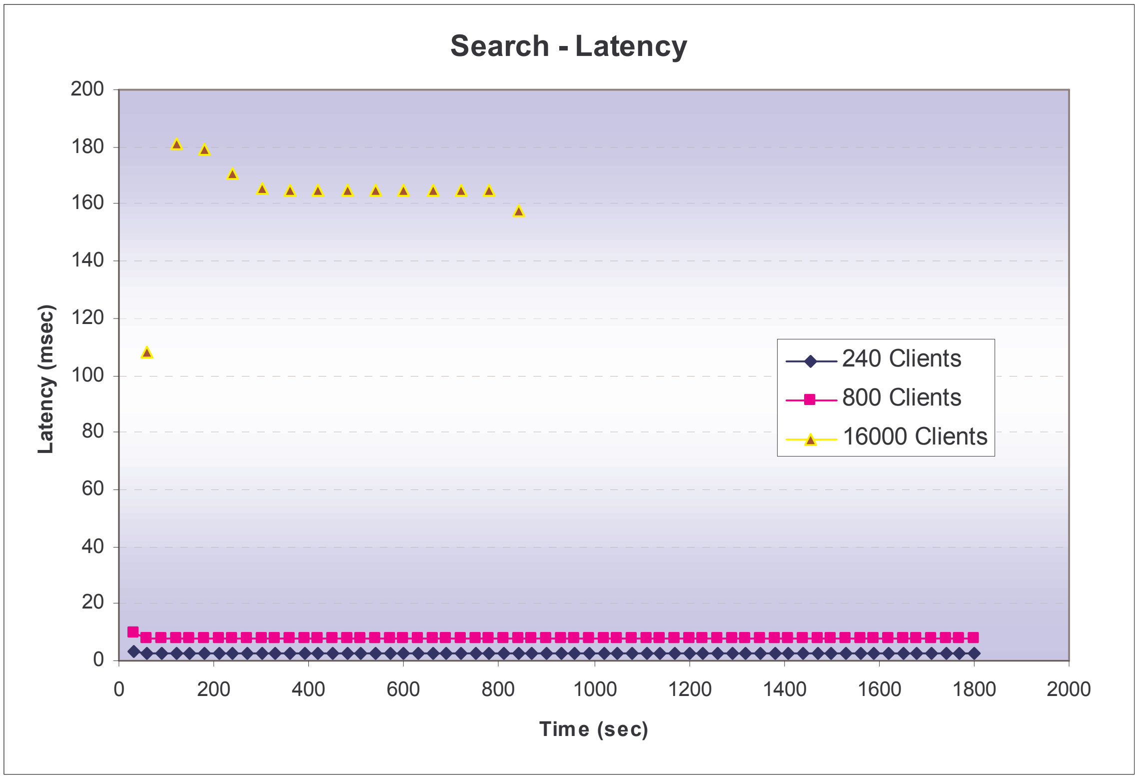

Figure 4 Latency for Search Operation

4.3.1.1.2 Authentication Operation

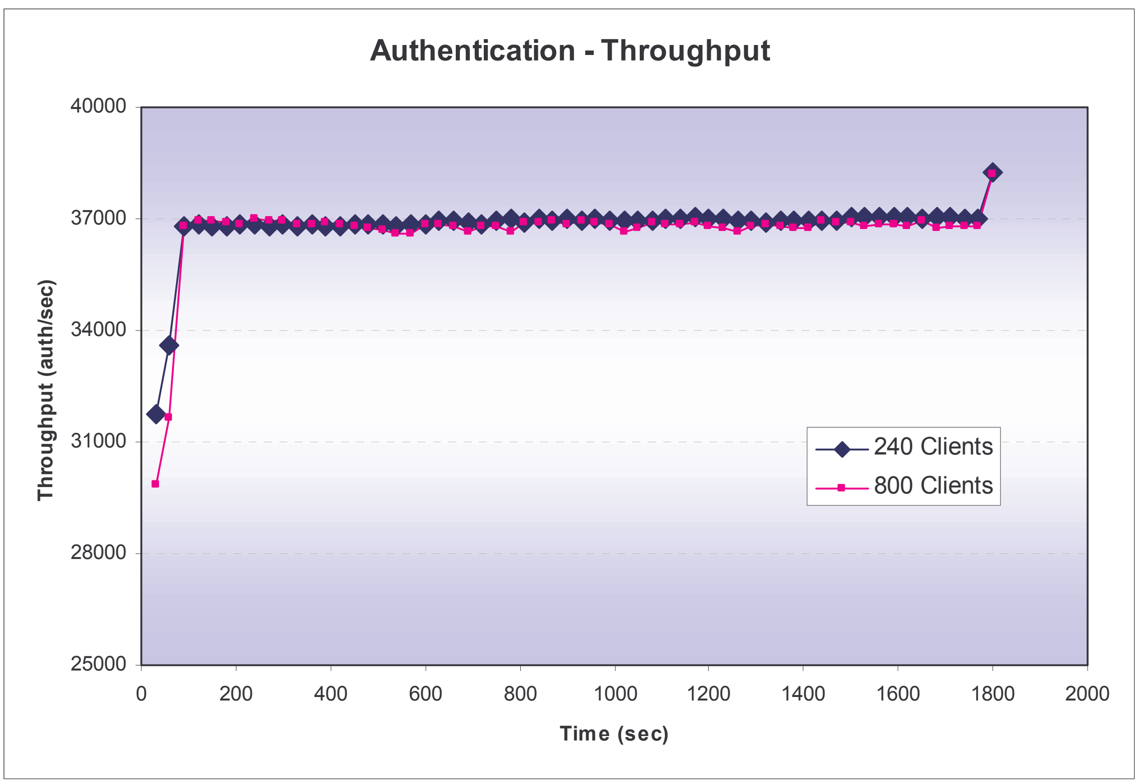

Figure 5 Throughput for Authentication Operation

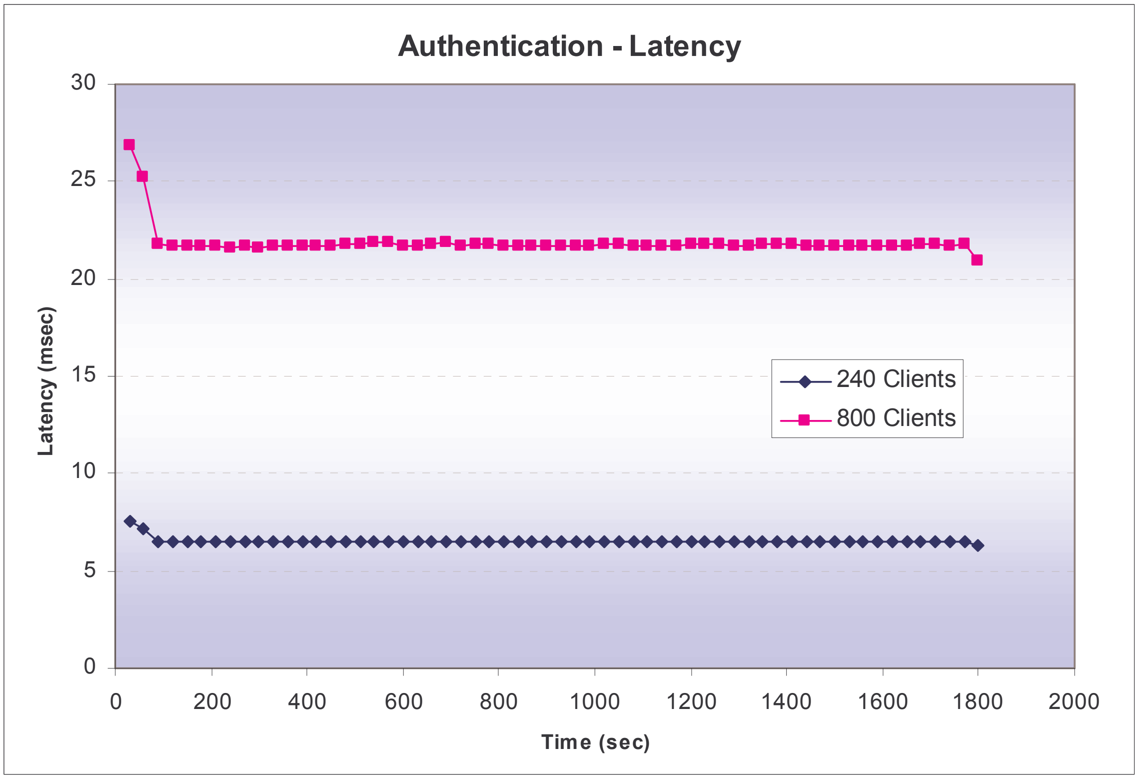

Figure 6 Latency for Authentication Operation

4.3.1.1.3 Compare Operation

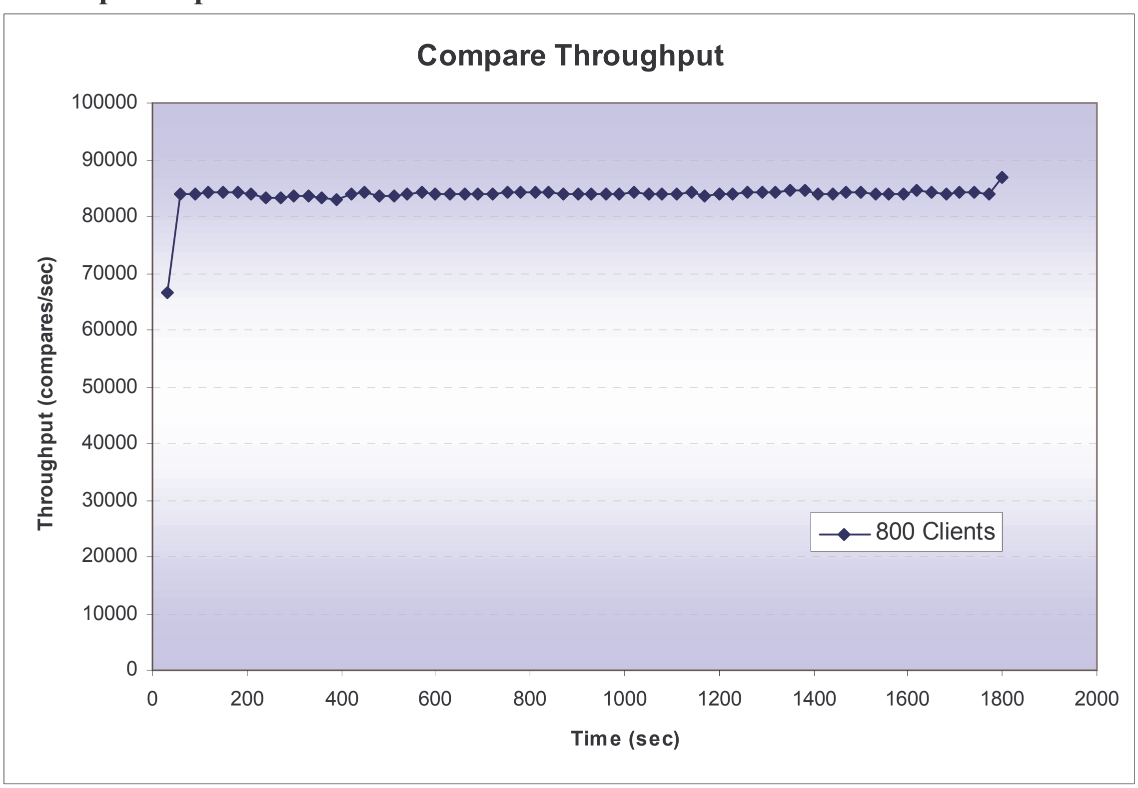

Figure 7 Throughput for Compare Operation

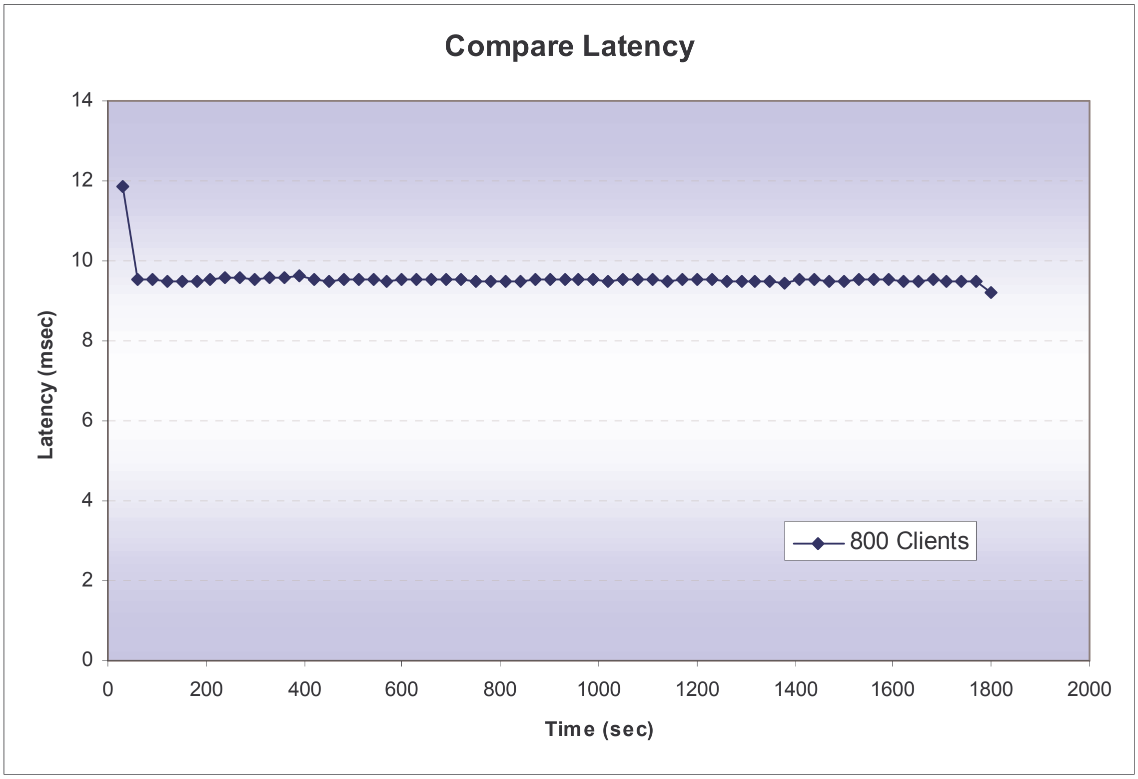

Figure 8 Latency for Compare Operation

4.4 Conclusion

The following conclusions were drawn from the benchmark –

- OID installation and configuration is straightforward and does not require undue time or effort.
- The Bulk Load is a very fast operation in OID and two Billion users LDIF data was loaded in under 94 hours.
- It was observed that, for the chosen test scenarios, OID could easily scale up to 101,000 searches/sec, 37,000 authentications/sec, 80,500 binds/sec, 14,000 modifications/sec and 84,000 comparisons/sec. OID sustained 99,500 searches/sec while serving 16,000 concurrent clients, beyond which, even though the system resources were not fully utilized, the load generation mechanism becomes the bottleneck.
- Given our experience with OID, we feel it is suitable for high-end mission critical applications requiring LDAP services, owing to its performance in the set of experiments conducted.

Appendix A: Raw Partition configuration

- **OD1: RAID 0+1**

#	Size (GB)	Device	Purpose	XVM Volume Name	IS4500-A Luns	IS4500-B Luns
1	4	/dev/lxvm/vol-od1-1	SYSTEM tablespace	vol-od1-1	101	
2	4	/dev/lxvm/vol-od1-2	SYSAUX tablespace	vol-od1-2	102	
3	40	/dev/lxvm/vol-od1-3	Default UNDOTBS1 tablespace	vol-od1-3	103	
4	1	/dev/lxvm/vol-od1-4	USERS tablespace	vol-od1-4	104	
5	1	/dev/lxvm/vol-od1-5	EXAMPLE tablespace	vol-od1-5	105	
6	1	/dev/lxvm/vol-od1-6	Control file 1	vol-od1-6	106	
7	1	/dev/lxvm/vol-od1-7	Control file 2	vol-od1-7	107	
8	1	/dev/lxvm/vol-od1-8	Control file 3	vol-od1-8	108	
9	1	/dev/lxvm/vol-od1-9	Server parameter file (SPFILE)	vol-od1-9	109	
10	1	/dev/lxvm/vol-od1-10	Passwd file	vol-od1-10	110	
11	1	/dev/lxvm/vol-od1-11	Olts_battrstore	vol-od1-11	111	
12	60	/dev/lxvm/vol-od1-12	Default Temp tablespace	vol-od1-12	112	
13	20	/dev/lxvm/vol-od1-13	Olts_default table space	vol-od1-13	113	
Total	136					

- **OD2: RAID 0**

#	Size (GB)	Device	Purpose	XVM Volume Name	IS4500-A Luns	IS4500 -B Luns
1	16	/dev/lxvm/vol-od2-1	Redo log file 1	vol-od2-1		201
2	16	/dev/lxvm/vol-od2-2	Redo log file 2	vol-od2-2		202
3	16	/dev/lxvm/vol-od2-3	Redo log file 3	vol-od2-3		203
4	16	/dev/lxvm/vol-od2-4	Redo log file 4	vol-od2-4		204
5	16	/dev/lxvm/vol-od2-5	Redo log file 5	vol-od2-5		205
6	16	/dev/lxvm/vol-od2-6	Redo log file 6	vol-od2-6		206
7	16	/dev/lxvm/vol-od2-7	Redo log file 7	vol-od2-7		207

8	16	/dev/lxvm/vol-od2-8	Redo log file 8	vol-od2-8		208
9	1400	/dev/lxvm/vol-od2-9	OLTS Temp tablespace	vol-od2-9		209
Total	1528					

- **OD3: RAID 0+1**

#	Size (GB)	Device	Purpose	XVM Volume Name	IS4500-A Luns	IS4500-B Luns
1	17000	/dev/lxvm/vol-od3-1	Olts_ct_store tablespace	vol-od3-1	122,123,124,125,126,127,128	222,223,224,225,226,227,228
Total	17000					

Appendix B: OS Kernel Configuration

Name	Value
kernel.shmall	208857600
kernel.shmmax	493183820800
kernel.shmmni	4096
kernel.sem	260 32000 128 256
kernel.msgmnb	65536
kernel.msgmni	3000
kernel.msgmax	8192
net.core.rmem_default	1048576
net.core.wmem_default	2097152
net.core.rmem_max	2097152
net.core.wmem_max	262144

Appendix C: Bulkload generate phase

The invocation of the ‘bulkload’ tool for *generate* phase is as given below –

a. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 generate=true file=/bench/LDIF/0-1bBenchmark_1.ldif
b. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 generate=true file=/bench/LDIF/0-1bBenchmark_2.ldif
c. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 generate=true file=/bench/LDIF/0-1bBenchmark_3.ldif

d. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 generate=true file=/bench/LDIF/1-2bBenchmark_1.ldif
e. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 generate=true file=/bench/LDIF/1-2bBenchmark_2.ldif
f. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 generate=true file=/bench/LDIF/1-2bBenchmark_3.ldif

Appendix D: Bulkload load phase

The invocation of the 'bulkload' tool for *load* phase is as given below –

a. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 load =true file=/bench/LDIF/0-1bBenchmark_1.ldif
b. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 load =true file=/bench/LDIF/0-1bBenchmark_2.ldif
c. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 load =true file=/bench/LDIF/0-1bBenchmark_3.ldif
d. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 load =true file=/bench/LDIF/1-2bBenchmark_1.ldif
e. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 load =true file=/bench/LDIF/1-2bBenchmark_2.ldif
f. $ORACLE_HOME/ldap/bin/bulkload connect=<db-conn-str> threads=16 load =true file=/bench/LDIF/1-2bBenchmark_3.ldif