A Performance Evaluation of Storage and Retrieval of DICOM Image Content in Oracle Database 11g Using HP Blade Servers and Intel Processors

An Oracle White Paper
July 2008

·

A Performance Evaluation of Storage and Retrieval of DICOM Content in Oracle Database 11g Using HP Blade Servers and Intel Processors

•

A Performance Evaluation of Storage and Retrieval of DICOM Content in Oracle Database 11g Using HP Blade Servers and Intel Processors

EXECUTIVE SUMMARY

One of the world's leading PACS vendors challenged us to meet their image retrieval requirement of 300 images per second with DICOM images stored in Oracle Database. We set out to meet their challenge by defining a representative workload based on input from major PACS and modality vendors, Hospitals, Life Science researchers and Pharmaceutical companies, and by performing a DICOM benchmark using this workload on commodity HP and Intel hardware with Linux OS and Oracle Database.

How did we do? We were able to achieve nearly three times the number we were challenged to meet, achieving peak image retrieval speeds of 852 images per second.

How do other databases compare? No one else has DICOM support in the database, so no other database product can do what Oracle can do.

Retrieval Performance

Oracle Multimedia shows sustained read speeds of over 850 Cardiac CT images / second on one dual CPU HP computer with Intel chips on LINUX OS. (Cardiac CT images were each 512k bytes.) For a 2 node RAC cluster on the same boxes, the retrieval speed was over 1550 images / second. The read speed for Mammography images, which averaged 26 MB in size, was over 500 MB/sec on a single node and over 1 GB/sec on a 2-node RAC cluster.

Write Performance

Oracle Multimedia shows sustained write speeds of over 550 Cardiac CT images / second on one dual CPU HP computer with Intel chips on LINUX OS. This is an ingestion rate of over 950 Cardiac CT studies per hour. A terabyte of images can be loaded in less than 65 minutes. The write speed for Mammography images exceeded 440 MB/sec on a single node.

Hardware Used

The benchmark was performed on the following hardware:

Server: = 2 X HP BL480c Blades

2 x 3.2 GHz Intel Xeon X5640 Quad Core Processors with 48 GB memory

Storage: 2 x HP StorageWorks EVA 8100 ; 112 X 146 GB FC 15k HDD

BACKGROUND

Oracle Database 11g Release 1 introduces two new features that enable customers to build large, high performance repositories and / or archives of medical and life science content that are managed and secured using Oracle Database.

Oracle Multimedia DICOM Format Support

With Oracle Database 11g Release 1 (11.1), Oracle Multimedia provides full support for DICOM (Digital Imaging and Communications in Medicine), the format universally recognized as the standard for medical imaging. Applications can now use Oracle Multimedia DICOM Java and PL/SQL APIs to store, manage, and manipulate DICOM content.

Oracle Multimedia DICOM support includes

- An ORDDicom object type
- DICOM metadata extraction
- DICOM conformance validation
- DICOM image processing and image compression
- Making DICOM content anonymous
- Creating content from images and metadata
- A run-time updatable DICOM data model
- User defined extensions to the standard DICOM model

For more information on Oracle Multimedia DICOM, see http://www.oracle.com/technology/products/intermedia/index.html

Oracle SecureFiles

Oracle SecureFiles is a new feature in Oracle Database 11g that is specifically engineered to deliver high performance and scalability for storing file data compared to that of traditional file systems while retaining the advantages of Oracle database storage.

With SecureFiles, Oracle has introduced numerous architectural enhancements for greatly improved performance and scalability, efficient storage and easier manageability. SecureFiles features include.

- Encryption
- Intelligent Pre-fetching
- New Network Layer
- Filesystem-like Logging
- Deduplication

•

For more information on Oracle SecureFiles, see http://www.oracle.com/technology/products/database/securefiles/index.html

Storing Medical and Life Science Images in Oracle Database

The advantages of using Oracle Database to build large, high performance repositories and archives for medical and life science content include:

- Integration of medical images with patient data to avoid scattering of patient-centric health records across heterogeneous systems
- Data transfer and sharing of medical and life science content without custom-built infrastructure
- Consistent enforcement of privacy and security policies for all archive content
- Management of an evolving DICOM archive (incorporation of new modalities, new DICOM standards) with no application changes or downtime

PURPOSE

The purpose of this benchmark is to demonstrate that Oracle Database 11g with DICOM support, is a high performing, low cost, and functionally complete platform for large repositories and archives of Medical and Life Science images. This was achieved with HP computers and storage, and Intel processors.

A secondary purpose was to construct a representative image workload based on input from PACS and modality vendors, Hospitals, Life Science researchers and Pharmaceutical companies, which would be clear to and accepted by them, as we generated and published results.

We were specifically requested to report on Cardiac CT Images (512k bytes); in addition we added 5 other modalities to the benchmark. The database size was 2 terabytes comprised of 2.4 million images. New DICOM files with unique metadata and varying image content were generated to ensure that each study has a unique Study Instance UID, Series Instance UID and patient name. The patient name follows the name distribution from the US Census. Each image has a unique Media Storage SOP Instance.

The benchmark measures the rate at which images can be inserted into Oracle Database and retrieved from Oracle database.

A Workload for Medical Imaging

This benchmark defines a workload with two main components: a model for the DICOM image dataset, and a set of operations to perform on the data. The two components are described in the following sections. Medical Imaging modalities are used, since they are well defined; however Life Science applications can easily relate their image sizes to the Medical modalities, and estimate performance.

•

Modeling the Data

A diagnostic medical imaging device produces a group of digital images known as a study. The size of individual images and the number of images in a study varies with the objective of the clinical test and with the various imaging devices. The images that make up a study are commonly stored, retrieved and processed as a group.

This benchmark uses the dataset model shown in Table 1 to simulate six common modalities of medical imaging. The model does not represent any specific medical device. The intent is to create a set of modeled modalities that broadly represent actual modalities in terms of the number of image files per study and the size of the individual files. Along these dimensions, the modeled modalities range from **M02,** which contains exactly one file ranging in size from 1MB to 100MB, to **M03, which** can contain a maximum of 4096 images with an average image size of .5MB. This variation in count and size allows us to study storage and retrieval performance over a widely varying dataset.

Table 1: Dataset Model for DICOM Images

Type	Modeled Modality	Images / Study			Size of Each Image (MB)			Avg Study size (MB)
		Min	Max	Avg	Min	Max	Avg	
M01	MR / CT	16	1024	64	.06	4.2	.36	22
M02	Ultrasound	1	1	1	1	108	28	28
M03	Cardiac CT	128	4096	2051	.5	5	.5	1031
M04	Visible light	1	64	16	.02	4	1.6	25
M05	Mammography	4	4	4	13.6	44	26	106
M06	Pathology	4	4	4	644	2086	1319	5276

A dataset of 20,080 studies, with approximately 2.4M images and totaling 2 TB in size was built according to the characteristics of the model. The dataset was built by expanding a small set of genuine DICOM data files using the DICOM image and metadata manipulation features of Oracle Multimedia. New DICOM files with unique metadata and varying image content were generated to ensure that each study has a unique Study Instance UID, Series Instance UID and patient name. The patient name follows the name distribution from the US Census. Each image has a unique Media Storage SOP Instance.

•

Table 2 DICOM Dataset at 2 TB Scale

Type	#Studies	#Images	%Total Images	Size (GB)	%Total Size
M01	3000	191360	7.92%	64	3%
M02	4000	4000	0.17%	108	5%
M03	1000	2051328	84.94%	1006	49%
M04	10000	160000	6.63%	248	12%
M05	2000	8000	0.33%	206	10%
M06	80	320	0.01%	412	20%
Total	20080	2415008		2044	

Modeling the Operations

Our benchmark defines two tasks to model the common operations that occur in a medical imaging application. These operations are study retrieval and study insertion.

Study Retrieval

The study retrieval operation models what occurs when a radiologist reviews a study at a viewing workstation. A set of images is retrieved from the centralized database.

In our workload, a study is chosen at random, with equal probability, from all the studies stored in the database for a given modality. Once this study is identified, all the images for the study are retrieved from the database and delivered to the client. Image retrieval can occur in parallel. When all the images have been retrieved, performance statistics such as throughput and response time are computed for the study.

Study Insertion

A study insertion operation models what occurs when an imaging device creates a new study and the images are uploaded to a repository.

In our workload, a study is chosen at random, with equal probability, from a pre-selected set of candidate studies chosen to represent a given modality. These candidate studies reside on a file system accessible only to the client machines. Once this study is identified, a new study record is created in the database. Then a new image record is created for each image in the study and the images are uploaded from the file system into the database. Image insertion can occur in parallel. When all the images have been inserted, performance statistics such as throughput and response time are computed for the study.

SYSTEM CONFIGURATION

The benchmark was run using two pairs of client – server systems. For the single instance tests, only one client server pair was used. Both pairs were used for the dual instance tests. This section provides information about the hardware and software used to perform the tests.

Database Servers

The database server system was configured using two HP BL480c Server Blades. The major hardware components used for each server are show in Table 3. Note that each server contains a Dual Port 4Gb FibreChannel HBA for connectivity to the storage system. This provides each server with up to 8Gb of throughput to the storage system. Each server also contains five 1GigE network ports for connectivity to the client machines. This provides each server with more than 500 MB/sec of network bandwidth to the client application.

Table 3: Database Server Systems –2 x HP BL480c

Component	Description	
CPU	2 x 3.2 GHz Intel Xeon X5640 Quad Core Processors, 8 cores total per server	
Memory	48 Gigabytes	
I/O Controller	QLogic QMH2462 4Gb Dual Port FC HBA	
Network	5 Gigabit Ethernet port.	
Operating System	Oracle Enterprise Linux 5.1	
Kernel Parameters	/dev/shm	33GB
	kernel.shmmax	68719476736
	kernel.shmall	4294967296
	kernel.sem	500 32000 256 256
	net.ipv4.ip.local_port_range	1024 65000
	net.core.rmem.default	4194304
	net.core.rmem.max	4194304
	net.core.wmem.default	4194304
	net.core.wmem.max	4194304
Database	Oracle Database 11g Release 1 (11.1.0.6) Enterprise Edition	
	Options	Real Application Clusters Partitioning

•

	Patches	bug 6277860 bug 6991044
Database Parameters	db_block_size	8192
	memory_target	34628173824
	db_writer_processes	1
	SDU	32768
	TDU	32768
	SEND_BUF_SIZE	1048576
	RECV_BUF_SIZE	1048576

Database Storage

The storage for the database was provided by an HP StorageWorks EVA 8100 and managed by Oracle ASM. All 112 disks were configured in a single EVA disk group. This disk group was then configured to provide 8 VRaid1 LUNs. All 8 LUNs were configured into a single ASM disk group to yield 8TB of database storage. Note that storage redundancy is achieved externally to ASM using VRaid1. Manual path distribution was employed to ensure that the I/O was balanced equally over all the Fibre Channel links. The details of the storage system are shown in Table 4.

Table 4: Database Storage –HP StorageWorks EVA 8100

Component	Description
Disk Drives	112 x 146GB FC 15k RPM HDD
Cache	8 GB automatic allocation management for read vs write
Host Interface	8 x 4Gb FC ports
Drive Interface	8 x 2Gb FC ports

Client Systems

Two HP BL465c Blade Servers were used as client systems to drive the tests. Each client contained five 1GigE network ports for connectivity to the server machines. This provided each client with approximately 500 MB/sec of network bandwidth to the database system. Two HP StorageWorks MSA1000 Arrays provided storage for the client systems. Each array contained 14 72GB drives to yield about 2 TB of available storage. This storage was used to hold DICOM images that were used for the Study Insertion tests.

•

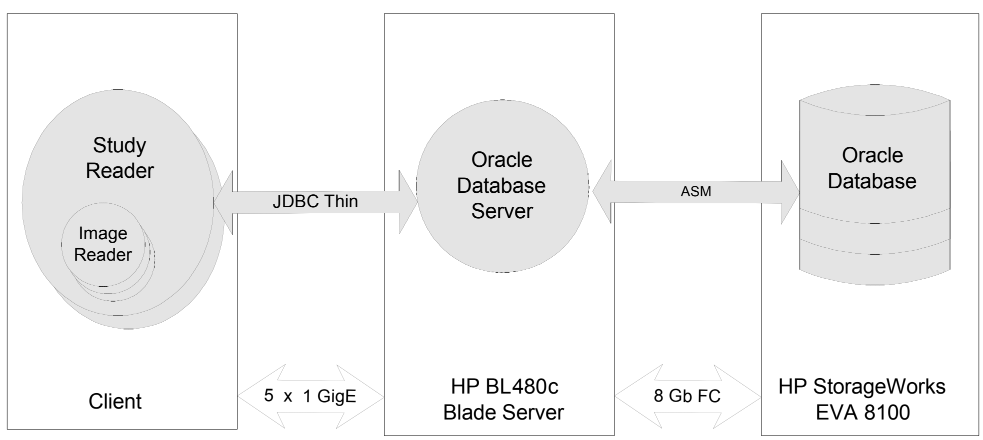
Client system details are show in Table 5.

Table 5: Client Systems 2 x HP BL465c

Component	Description
CPU	2 x 2.8 GHz quad core processors 8 cores total per client
Memory	64 GB
I/O Controller	QLogic QM2462 Dual Port FC HBA
Network	5 Gigabit Ethernet port.
Operating System	Oracle Enterprise Linux 5.1

Figure 1 shows a high level schematic of the application implementation for retrieving studies. The application was implemented as a multi-threaded Java program. The primary application component was a Study Reader. Each Study Reader had a pool of independent image readers that were used to fetch the image content from the database. The image readers enable images to be read in parallel for an individual study. A Study Reader randomly selects a study, with equal probability, from all the studies stored in the database for a given modality. Once this study is identified, all the images for the study are retrieved from the database. If multiple Study Readers are instantiated, then multiple studies can be retrieved in parallel.

Figure 1 Application Implementation for Study Retrieval



SINGLE INSTANCE PERFORMANCE RESULTS

In this section we present the performance test results for study retrieval and study insertion for a single instance database. The single instance database contained more than 20,000 studies and 2.4 million images comprising 2 TB of image data (See Table 2). For these results, six Study Readers were instantiated for each modality. The performance for each modality was tested independently.

Two primary performance metrics are used. The first measures the average rate that images that are retrieved from or written to the database. The measurement unit is images per second (images/sec). Modalities with small average image sizes will exhibit larger image rates then modalities with large image sizes. The second metric measures the average rate of data volume for retrieval and insertion operations. The measurement unit is megabytes per second (MB/sec).

Finally, we note that these metrics capture throughput rates for the entire system, not for individual study reader or writer processes.

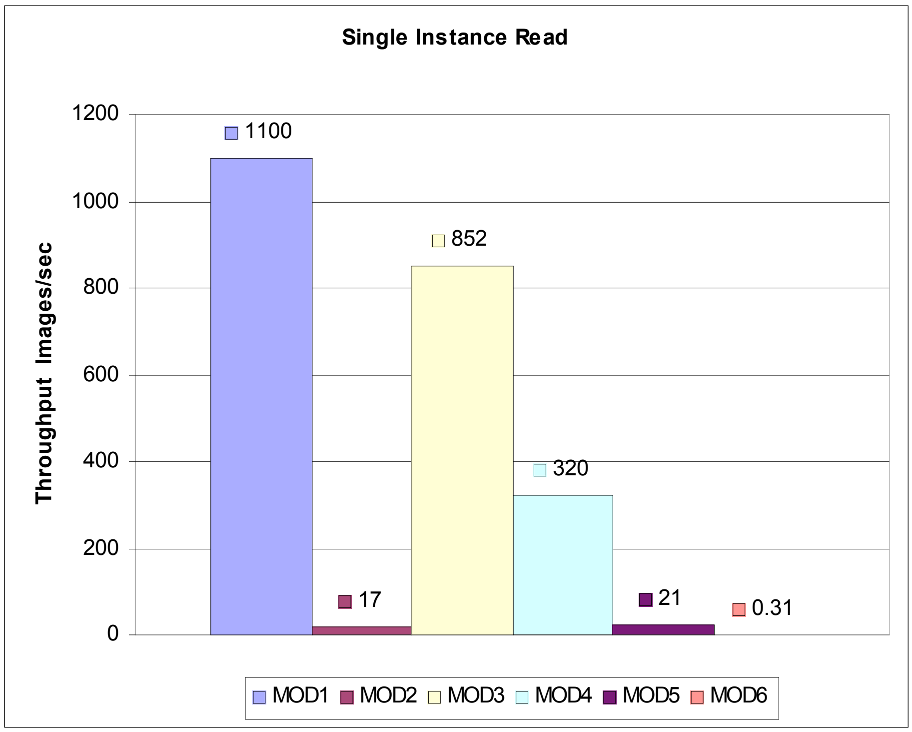
•

Single Instance Study Retrieval Performance by Modality

Figure 2 shows the sustained image retrieval rate for a single node database instance. Note that the average image size has a large impact on this metric. Modalities with relatively small image sizes, MR/CT (MOD1), Cardiac CT (MOD3) and Visible Light (MOD4) can achieve large image retrieval rates. Modalities with large average image sizes, Ultrasound (MOD2), Mammography (MOD5) and Pathology (MOD6) have comparatively low values for this metric.

At an average rate of 852 images/sec, the system was retrieving approximately 1497 Cardiac CT studies per hour from the Oracle database.

Figure 2 Single Instance Study Retrieval - Throughput (images/sec)

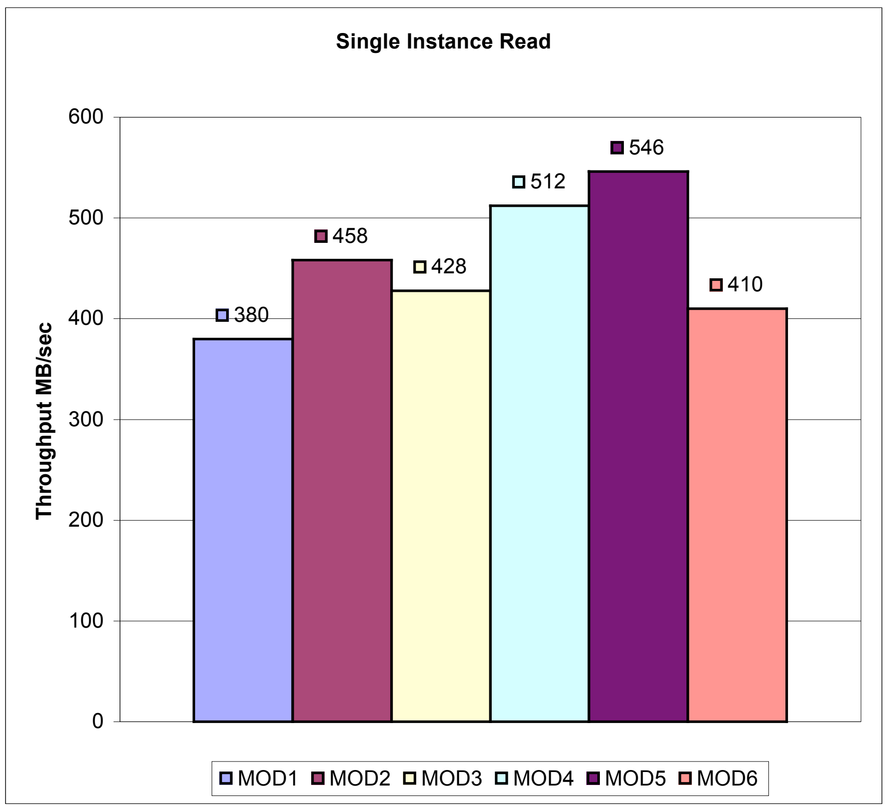


•

Figure 3 presents the results of the same experiment using a data volume metric, MB/sec. Note that for all but one of the modalities, the system achieved sustained rates over 400 MB/sec. Modality MOD1 has a very small average image size (360K bytes). The time needed to locate the individual image and begin the transfer is proportionally a much larger overhead than for the other modalities with larger average image sizes. Modalities with large images (MOD5) can achieve very high throughput rates. Once the image is located and the transfer begins, the system streams data from the database to the client at a rate that saturates the network.

Finally, we explain why the Pathology modality (MOD6) appears to perform lower than expected, as it is comprised of four large images per study. The reason is due to the variability of image size within a study. The smallest image is 644MB while the largest is more than three times greater at 2086 MB. Throughput statistics are computed at the study level and the response time for the study is recorded when the last image is completely retrieved. This means that study retrieval performance is bounded by the performance of retrieving a single large image. The application is able to retrieve multiple images in parallel but a single image is retrieved in a sequential operation.

Figure 3 Single Instance Study Retrieval -Throughput (MB/sec)



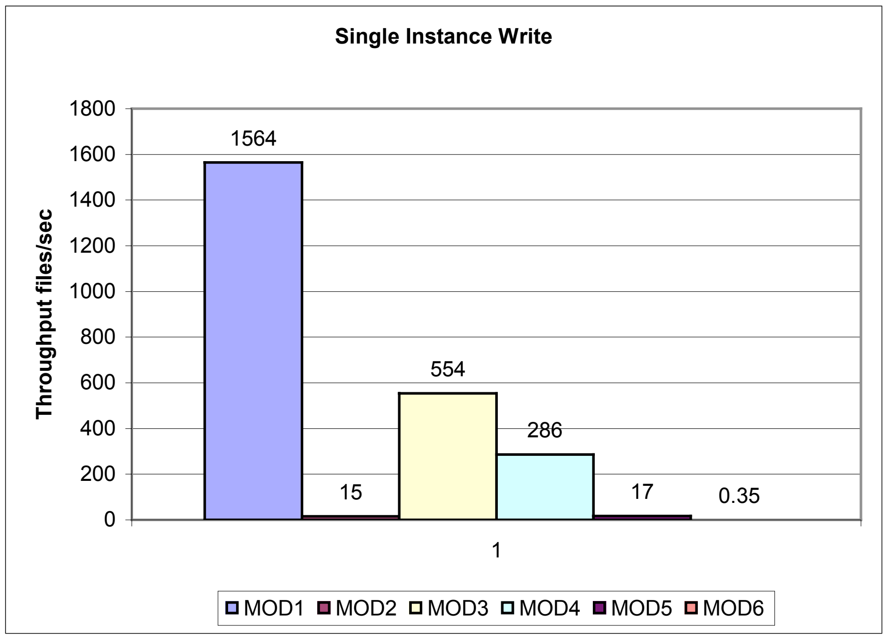
•

Single Instance Study Insertion Performance by Modality

Figure 4 shows the sustained image insertion rates for a single node database instance. Note that the average image size has a large impact on this metric. Modalities with relatively small image sizes, MR/CT (MOD1), Cardiac CT (MOD3) and Visible Light (MOD4) can achieve large image insertion rates. Modalities with large average image sizes, Ultrasound (MOD2), Mammography (MOD5) and Pathology (MOD6) have comparatively low values for this metric.

At an average rate of 554 images/sec, the system was inserting approximately 970 new Cardiac CT studies per hour into the Oracle database.

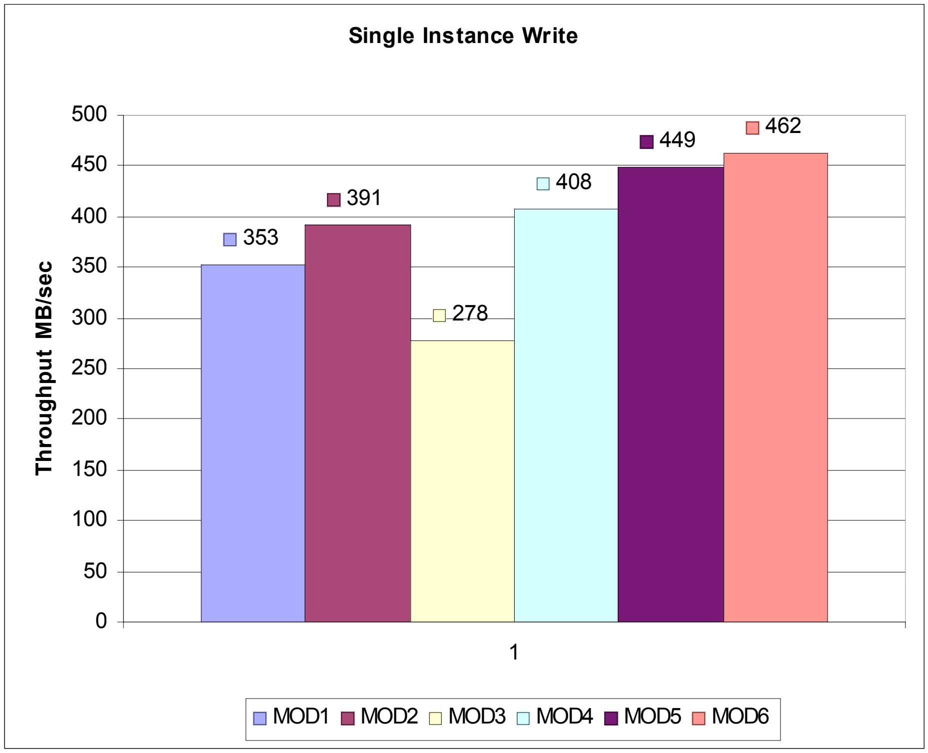
Figure 4 Single Instance Study Insertion - Throughput (files/sec)



•

Figure 5 presents the results of the same experiment using a data volume metric, MB/sec. Note that for very large image sizes, the system achieved sustained rates over 450 MB/sec.

Figure 5 Single Instance Study Insertion - Throughput (MB/sec)



•

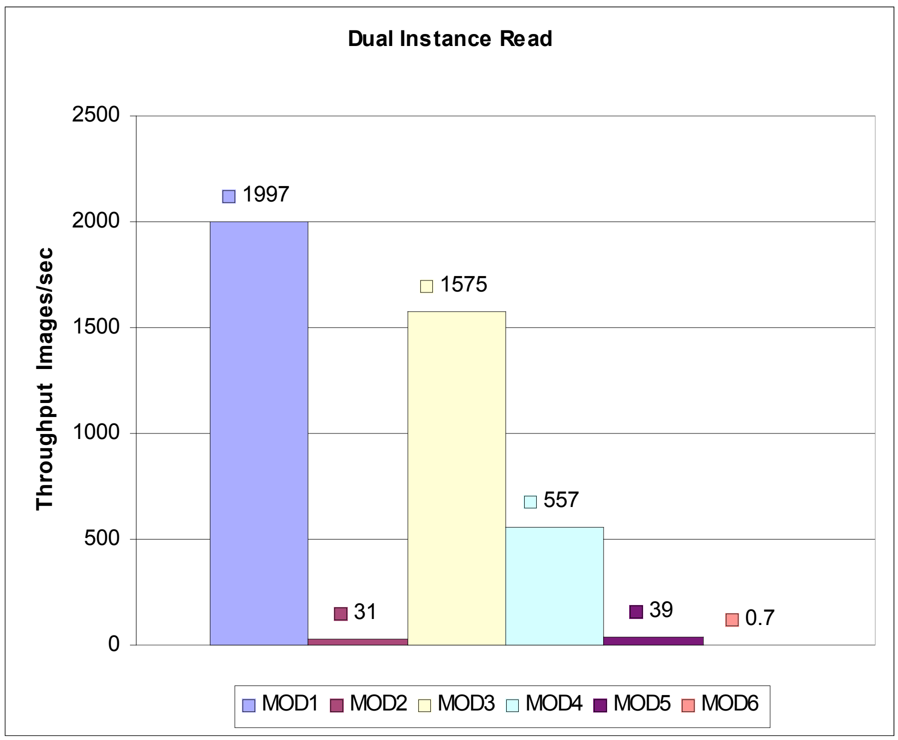
DUAL INSTANCE PERFORMANCE RESULTS

Adding a second server and a second client node expanded the system to a two-node RAC cluster. The database size was doubled, bringing the total archive size to 40,000 studies and 4.8 million images comprising 4 TB of image data.

Dual Instance Study Retrieval Performance by Modality

Figure 6 shows the sustained image retrieval rate for a two-node database system. At an average rate of 1575 images/sec, the system was retrieving approximately 2768 Cardiac CT studies per hour from the Oracle database.

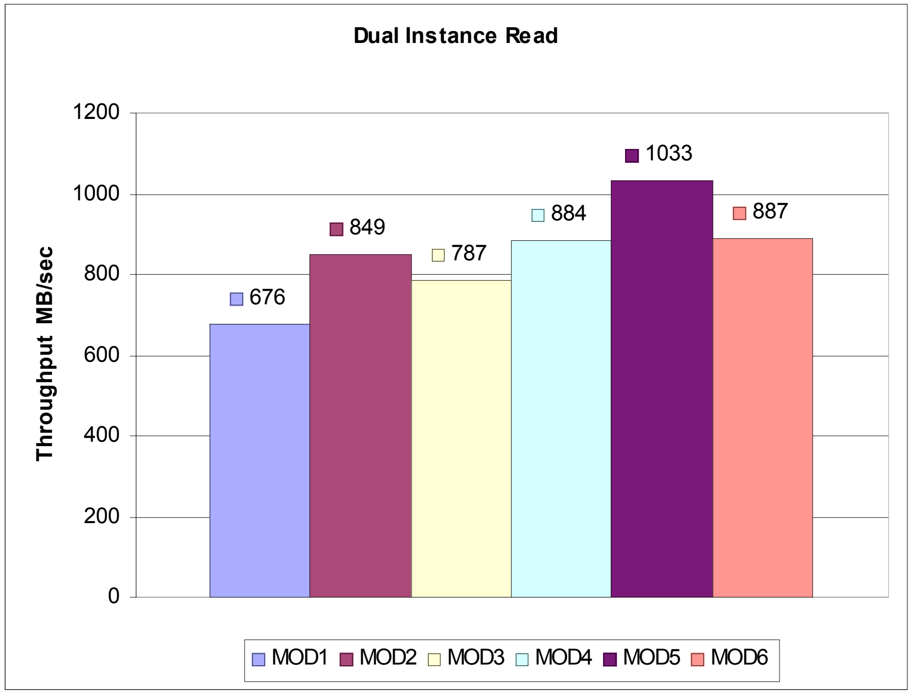
Figure 6 Dual Instance Study Retrieval – Throughput (files/sec)



•

Figure 7 Shows dual instance read results measure in data volume throughput. Note that we achieved over 1 GB/sec in throughput for reading Mammography images (MOD5).

Figure 7 Dual Instance Study Retrieval – Throughput (MB/sec)7



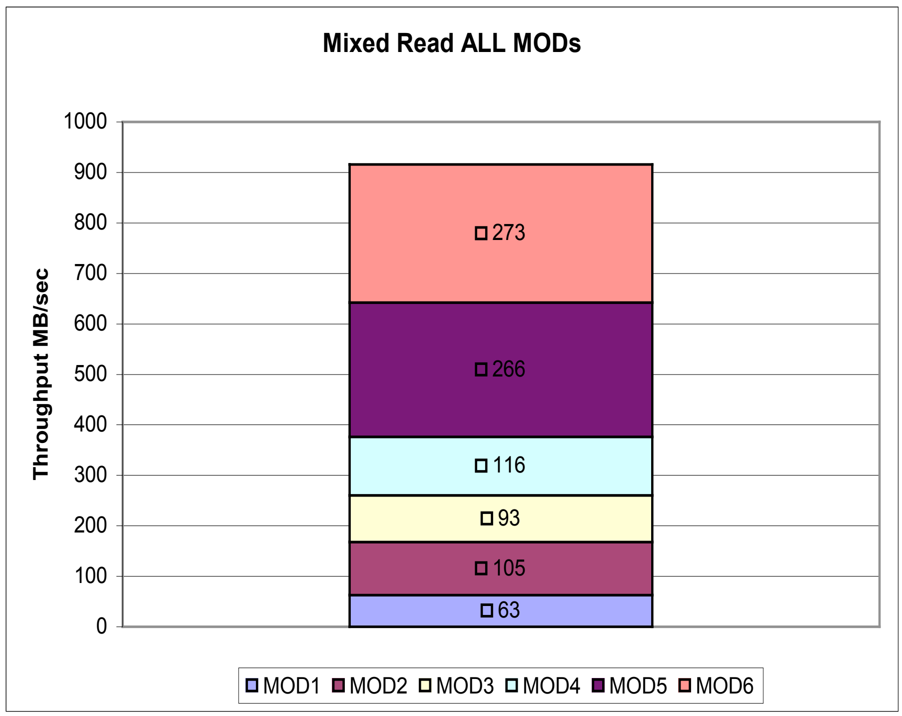
•

Dual Instance Study Retrieval Performance, All Modalities at Once

Figure 8 Shows results for a test where all modality types were read from the database. One Study Reader for each modality was instantiated on each client machine. Nothing was done to tune the performance of a particular modality.

The system achieved a cumulative throughput performance of over 900 MB/sec. with more than 10% of the data coming from Cardiac CT (MOD3). This clearly demonstrates that Oracle Database can achieve very high performance for a mixed workload.

Figure 8 Dual Instance Study Retrieval – Throughput (MB/sec)

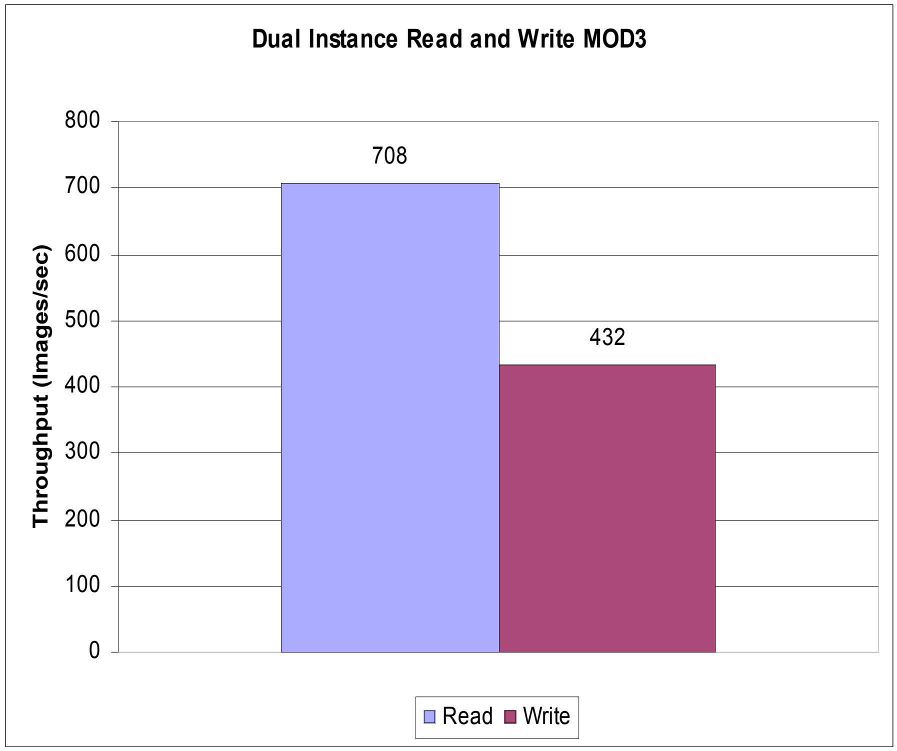


•

Dual Instance Simultaneous Retrieval and Insertion of Cardiac CT

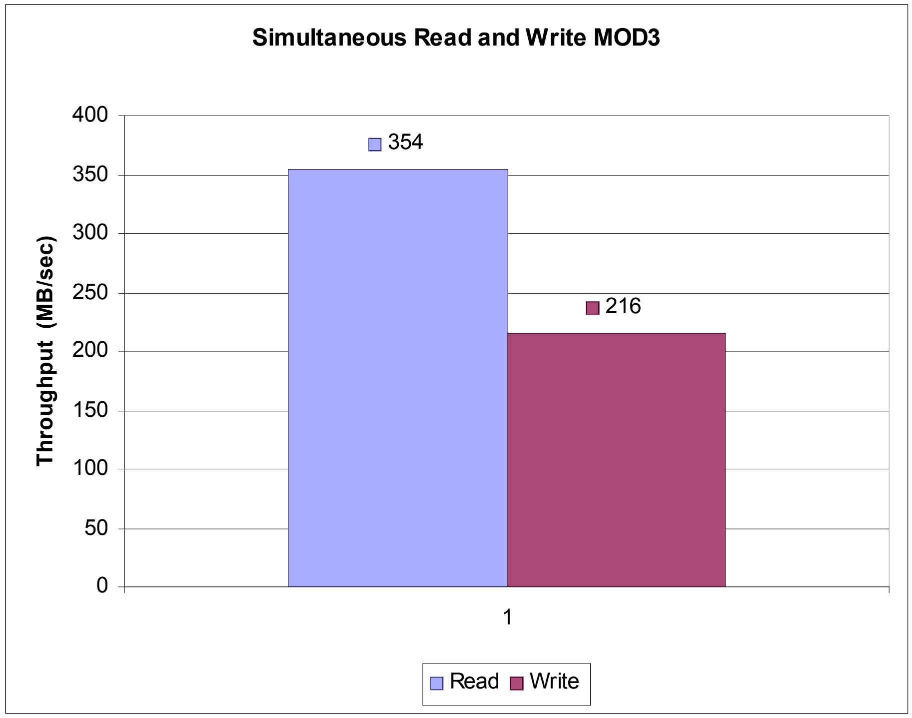
In a final experiment, we tested the simultaneous retrieval and insertion of Cardiac CT studies (MOD3) in a dual instance system. Figure 9 shows the performance in Images/sec and Figure 10 shows the performance in MB/sec. At these rates, the system was reading over 1200 Cardiac CT studies per hour while simultaneously inserting 760 new studies.

Figure 9 Dual Instance Study Retrieval and Insertion - Throughput (Images/sec)



•

Figure 10 Dual Instance Study Retrieval and Insertion – Throughput (MB/sec)



•

APPENDIX - IMPLEMENTING THE WORKLOAD

There are a few simple guidelines to follow in order to achieve the best possible performance for medical image storage and retrieval using Oracle Multimedia and SecureFiles in Oracle Database 11g. The guidance is organized into four areas: database storage, table design, application design and network configuration.

Database Storage

- Use Oracle ASM to manage your disk storage.
- Storage image content in a tablespace with automatic space management using local uniform extents – 64 MB in size.
- Consider using Oracle Partitioning to control the maximum size of storage segments. This benchmark used 1 TB table partitions.

Table Design

DICOM image content is stored using the ORDDicom object data type. which is a complex data type containing two BLOB attributes, (**SOURCE.LOCALDATA**, **EXTENSION**) and an XMLType attribute (**METADATA**) (See Table 6). Careful attention to storage parameters is necessary to achieve compact, efficient storage and high performance for inserts and retrievals

One important fact to understand is the two-step process that is usually used to load and initialize an ORDDicom object. In the first step, a record is created with an empty ORDDicom object and the image data is loaded into the **SOURCE.LOCALDATA** BLOB attribute. In the second step, the ORDDicom object is selected from the table and the **setProperties()** method is invoked to extract metadata and set object attributes. Then the row is updated with the new ORDDicom object. Since this initialized object has increased in size, it requires more space on the page. In this benchmark, using the table definition in Table 8, we observed that an ORDDicom object increased in size approximately 190 bytes after metadata extraction and object attribute initialization.

Determining a good estimate for the amount of increased space required for updates is critical to setting a proper value for the PCTFREE table parameter. This parameter controls how much space is left free on the page during row insertion in order to reserve space for rows that will grow in size when they are updated. If there is insufficient space on the page to store an updated row, the row must migrate to another database page. This leads to row chaining and can be a significant detriment to performance.

Another important item in the table definition is the storage specification for the **METADATA** attribute. This attribute is an XMLType and can be stored in a LOB or decomposed using structured storage. Structured storage will place the data in line on the page and greatly increase the average row length. DICOM metadata documents can be large and this would require a larger database page to accommodate the larger rows. For this benchmark, we chose to store the metadata attribute using SecureFile CLOB storage and disabled storage in row to prevent small documents from consuming page space. We also specified the NOCACHE attribute since the metadata documents are not read as part of this workload.

Table 6 ORDDicom Object Type

Attribute		Type	
SOP_INSTANCE_UID		VARCHAR2(128)	
SOP_CLASS_UID		VARCHAR2(64)	
STUDY_INSTANCE_UID		VARCHAR2(64)	
SERIES_INSTANCE_UID		VARCHAR2(64)	
SOURCE		ORDSYS.ORDDATASOURCE	
	LOCALDATA		BLOB
	SRCTYPE		VARCHAR2(4000)
	SRCLOCATION		VARCHAR2(4000)
	SRCNAME		VARCHAR2(4000)
	UPDATETIME		DATE
METADATA		SYS.XMLTYPE	
CONTENTLENGTH		NUMBER	
FLAG		NUMBER	
EXTENSION		BLOB	

Finally, the **EXTENSION** attribute, a BLOB type for internal use, is stored as a SecureFile with NOCACHE option. This attribute is usually less than 4000 bytes in size and would be stored in line on the page unless the disable store in row attribute is specified.

The DICOM image dataset was stored in two tables using a master-detail relationship. One record for each study was stored in the DICOM_IMAGE_GROUP table. One record for each image was stored in the DICOM_IMAGE table. The DDL for these tables is show in Table 7 and Table 8 respectively.

The DICOM_IMAGE table is partitioned using the **I_DATE** column, with an interval of one week per partition. Values for this date column were fixed so that each partition contained 1 TB of data.

•

Table 7: Table definition for DICOM_IMAGE_GROUP

```
create table dicom_image_group (
    ig_id integer,               -- image group id
    ig_modid integer,            -- modality id
    ig_date  date,               -- date of load
    ig_patient_name varchar2(200),
    ig_description  varchar2(400),
    constraint ig_id_u unique(ig_id)
    using index (create unique index ig_id_ix
       on dicom_image_group(ig_id) reverse)
);
```

Table 8: Table Definition for DICOM_IMAGE

```
create table dicom_image (
    i_id  integer,               -- image id
    i_ig_id integer,             -- image group id
    i_date date,                 -- load date
    i_image ordsys.orddicom,     -- dicom image
    constraint i_ig_id_u unique(i_ig_id, i_id)
      using index (create unique index i_ig_id_ix
        on dicom_image(i_ig_id, i_id) ) )  )
--
--  metadata extraction expands the ORDDicom object
    pctfree 60
--
-- lob storage
    lob(i_image.source.localdata) store as SecureFile
        ( nocache filesystem_like_logging
          retention none ),
--     disable in row storage for the extension
--     so that it does not consume page space
--     it is usually < 4k in size
    lob(i_image.extension) store as SecureFile
        ( nocache disable storage in row ),
--     store the metadata as a CLOB,
--     disable storage in row
    xmltype i_image.metadata store as SecureFile clob
        ( nocache disable storage in row )
--
-- store 1 TB of image data per partition
--
partition by range(i_date)
(
  partition image_04122008 values less than
       (to_date('04/12/2008', 'MM/DD/YYYY')),
  partition image_04192008 values less than
       (to_date('04/19/2008', 'MM/DD/YYYY')),
  partition image_04262008 values less than
       (to_date('04/26/2008', 'MM/DD/YYYY')),
  partition image_05032008 values less than
       (to_date('05/03/2008', 'MM/DD/YYYY')),
  partition image_05172008 values less than
```

•

```
        (to_date('05/17/2008', 'MM/DD/YYYY'))
;
```

Application Design

- If the application is written in Java, use the JDBC thin driver for best performance. If the application is written in C or C++, the OCI driver also provides good performance
- If the application is written in Java, use the methods **`getBytes()`** and **`setBytes()`** from the **`java.sql.Blob`** class to read and write from the SecureFile. These methods were introduced in JDBC 3.0. and they are the most efficient way to read and write to BLOBs.

  An alternative method for reading and writing is to obtain a Java stream class object: **`java.io.InputStream`** or **`java.io.OutputStream`**, and use the **`read()`** and **`write()`** methods of the stream class. This choice will not yield the best performance.
- Read and write large buffers to the database. This benchmark used a maximum buffer size of 10MB to read and write large pieces of the image to the database.
- Balance your application traffic over the available network links. This benchmark used a round-robin method for creating connections over five Ethernet links. With this scheme, we were able to achieve a balanced and full utilization of all network links.

Network Configuration

- Enable TCP flow control on the network switches. This is vital for ensuring consistent performance when a network link has a high utilization.
- Consider using jumbo frames. A larger packet will reduce CPU utilization on both client and server and allow a slightly higher maximum throughput. This benchmark used an MTU setting of 9000. Using the **netperf** tool, we measured a maximum network throughput of 118 MB/sec. For the default network setting MTU=1500, we measured a maximum throughput of 112 MB/sec.

ACKNOWLEDMENTS

This benchmark was a joint collaboration among Oracle Corporation, Hewlett-Packard Company and Intel Corporation. The Oracle team extends our gratitude to our colleagues at HP and Intel who provided the key resources and support to complete this project.

•

A Performance Evaluation of Storage and Retrieval of DICOM Content in Oracle Database 11g
Using HP Blade Servers and Intel Processors
July 2008

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com