ORACLE SECUREILES
A Filesystem Architecture in Oracle Database Server

Niloy Mukherjee, Amit Ganesh, Krishna Kuchithapadam and Sujatha Muthulingam
Oracle SecureFiles Team, Oracle Corporation, 500 Oracle Parkway, Redwood Shores, CA 94065, U.S.A.

Keywords: Oracle SecureFiles, Filesystems, Database Management Systems, Performance, Storage Utilization, Consistency, and Durability.

Abstract: Over the last decade, the nature of content stored on computer storage systems has evolved from being relational to being semi-structured, i.e., unstructured data accompanied by relational metadata. Average data volumes have increased from a few hundred megabytes to hundreds of terabytes. Simultaneously, data feed rates have also increased with increase in processor, storage and network bandwidths. Data growth trends seem to be following Moore's law and thereby imply an exponential explosion in content volumes and rates in the years to come. We introduce Oracle SecureFiles System, a storage architecture designed to provide highly scalable storage and access execution of unstructured and structured content as first-class objects within the Oracle relational database management system. Oracle SecureFiles breaks the performance barrier that has been keeping unstructured content out of databases. The architecture provides capability to maximize utilization of storage usage through compression and deduplication and preserves data management robustness through Oracle database server features such as transactional atomicity, durability, availability, read-consistent query-ability and security of the database management system.

1 INTRODUCTION

Traditionally, database management systems have been designed to provide maximum throughput of storage and access to relational data in transaction processing and data warehouse environments. However, the rapid growth of Internet has caused a huge increase in the amount of semi-structured information generated and shared by organizations in almost every industry and sector. As data volumes and ingestion rates step up, a number of challenges have risen in the area of database management (Blumberg, 2003) include provision for maximum throughput of storage and access operations, scalability, utilization of storage usage, highest degree of availability and security of critical data, and information lifecycle management (Hobbs, 2007) data volumes.

Filesystems have been preferred over database management systems for providing storage solutions for unstructured data while databases have been preferred to manage accompanying relational data for indexing and querying purposes. While filesystems provide better throughput of storage and access operations, they lack secure data management features such as atomicity, consistency, durability, manageability and availability (Vijayan, 2006)

We present Oracle SecureFiles System, a consolidated data storage architecture within Oracle 11g database server (Oracle, 2008) that bridges the gap between unstructured and relation data management by providing a clustered filesystem-like or better throughput and scalability for unstructured content while preserving the same for relational content. Consolidated content management as first-class database objects within the Oracle database kernel provides the advantages of rich database features such as transaction consistency, durability, metadata indexing, and query-ability using SQL standards. Besides providing support for advanced database features, Oracle SecureFiles provides advanced filesystem features such as compression and deduplication (Biggar, 2007) for optimising storage usage as well as encryption for maximum content security.

The rest of the paper is organized as follows. The design of the SecureFiles architecture and components is detailed in section 2. The following section details the set of database features associated with SecureFiles. A section on conclusion follows

Mukherjee N., Ganesh A., Kuchithapadam K. and Muthulingam S. (2008).
ORACLE SECUREILES - A Filesystem Architecture in Oracle Database Server.
In *Proceedings of the Third International Conference on Software and Data Technologies - SE/GSDCA/MUSE*, pages 60-63
DOI: 10.5220/0001885700600063

Section 4, which presents description of in-house throughput experiments conducted on SecureFiles and their evaluations.

2 DESIGN

The design of SecureFiles consists of two major components, namely, the schema or the structural component and the architectural component. The rest of the section will describe each of these components in details.

2.1 Schema

The structural design of SecureFiles is similar to that of filesystems.

Unstructured data associated with semi-structured content is stored as SecureFile objects. A SecureFile object is a collection of variable sized pages or chunks allocated from and stored in the Oracle database using the Oracle SecureFiles. Each chunk is a set of contiguous database blocks. The base table is an Oracle table that stores relational metadata associated with SecureFile objects. Besides the columns containing relational metadata, the table consists of one or more columns that hold locators providing reference pointers to the associated SecureFile objects. Each row-column intersection provides a distinct pointer to the very first block of an individual SecureFile object. Users can create unique and secondary indexes on the relational columns in the base table.

2.2 Architecture

SecureFiles architecture is layered into six major components, namely, Write Gather Cache, Transformation Management, Inode Management, Space Management and the I/O Management.

2.2.1 Write Gather Cache

SecureFiles uses a new cache that buffers data up to 64MB during write operations before flushing or committing to the underlying storage layer. This buffering of in-flight data allows for large contiguous space allocation and large disk I/O. Write performance is greatly improved due to reduced disk seek costs. The write gather cache is allocated from the database buffer cache and maintained on a per-transaction basis.

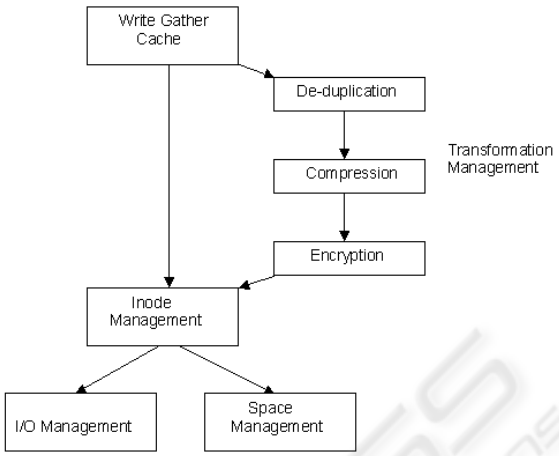

Figure 1: Architecture of SecureFiles.

2.2.2 Transformation Management

The advanced data transformation management comprises of three subcomponents, compression, encryption and de-duplication. Oracle SecureFiles provide the option to enable/disable all possible combinations of these features.

De-duplication: For every SecureFile object that has de-duplication enabled, a secure hash is generated for a subset of the object (prefix hash) and also for the whole object (full hash). During streaming writes, once generated, the prefix hash is compared to a set of prefix hashes stored in an index. If there is a prefix match, then the SecureFile object associated with the original prefix hash (master version) is read and byte-by-byte comparison is performed across the buffered data and the master version

Compression: Compression is performed on write gather cache buffers when sufficient amount of data is buffered. SecureFiles compression allows for random reads and writes to SecureFile data. Oracle SecureFiles architecture provides varying degrees of compression that represent a trade-off between storage savings and CPU costs.

Encryption: Oracle SecureFiles uses Transparent Data Encryption (TDE) syntax for encryption of SecureFile objects along with the accompanying relational metadata.

2.2.3 Inode Management

The inode management layer is responsible for initiating on-disk storage and access operations on SecureFile object buffers being communicated by the upper layers in the SecureFiles architecture. As a client of the space management layer, the inode

manager requests on-disk free space to store the amount of data being flushed by the write gather cache. Based on the array of chunks returned by the space management layer, the inode manager stores the metadata either in the row-column intersection of the base table associated with the object, or in the most current header block of the SecureFile object. The metadata information includes start block address and length of a chunk as well as the start and end offsets of the object being mapped to mana the chunk. The metadata structures are transactional managed similar to relational data and are recoverable after process, session and instance failures.

2.2.4 Space Management

The space management layer supports allocation of sets of variable sized contiguous data blocks or chunks up to 64M for on-disk storage of SecureFile objects. With SecureFile objects being cached in the Write Gather Cache, the space management layer is able to meet larger space requests from the inode manager through more contiguous layout on disk, therefore providing more efficient read and write access. Although space metadata is managed in-memory, the metadata changes are consistent across transactions, instance failures as well as media failiures

Operations such as full overwrites / rewrites, updates and deletes in SecureFiles follow ‘copy-on-write’ semantics resulting in de-allocation of space previously occupied by the offsets affected by the operation. Space freed during the de-allocation operations is not reused until it is retained for a certain period of time to achieve read consistency correctness for qureies.

2.2.5 I/O Management

During writes, the Inode Manager communicates the set of chunks obtained from the space layer as well as the write gather cache buffers to the I/O Manager. Based on a user parameter, the I/O Manager either copies the write gather cache buffers to database cache buffers or schedules asynchronous disk writes for the set of chunks.

The I/O Manager supports read-ahead or pre-fetching data from disk. It keeps track of access patterns of SecureFile objects and issues intelligent pre-fetching of chunks before the request is actually made. Read latency is reduced by overlapping the network and storage throughput.

3 FEATURES

Being stored as first-class objects within the database, Oracle SecureFiles has been designed to inherit most of the data management features such as transaction support, read consistency and data durability provided by the Oracle database server that are not provided by traditional filesystems.

3.1 Transactions and Read Consistency

Oracle SecureFiles is a transactional data store. Operations of Oracle SecureFiles generate undo records for relational data as well as metadata operations in the delta update, inode and space management components. SecureFile objects undergo 'copy on write' semantics on data manipulation operations and hence alleviate the requirement to store previous images for rollback purposes. Oracle SecureFiles achieves read consistency through 'copy-on-write' semantics thus enabling SecureFile segments to retain previous versions of SecureFile objects up to a certain period. A query on a SecureFile object issued at a point in time within the retention period is guaranteed to return the most consistent version of the object as of that point in time.

3.2 Data Durability

Oracle SecureFiles System design supports a range of data durability options. The design provides choice to the users to either use the database buffer cache to stage writes on SecureFile object buffers or to use the underlying storage for direct writes of SecureFile object buffers. Direct writes prevent pollution of buffer cache for large I/Os. Direct write operations can also be logged for media recovery purposes. The accompanying relational data, inode metadata and on-disk space metadata changes modify Oracle data blocks in the buffer cache itself and are logged in Oracle Redo logs.

4 PERFORMANCE EVALUATION

The evaluation experiment simulates a real world DICOM application consisting of digital diagnostic images accompanied by patient metadata. We compare read and write throughput of SecureFiles to that of NFSv3 filesystem. In both cases patient metadata is stored in the Oracle database. In the case of filesystem, the images are stored on Ext3 FS file servers that are accessed using NFSv3. In case of

SecureFiles, images are stored as SecureFile objects within the database.

4.1 Dataset and Hardware Set-up

The dataset consists of images ranging from 10 KB to 100 MB. The experiment consists of tests individually run on sizes averaging 10 KB, 100 KB, 1 MB, 10 MB and 100 MB. For tests on sizes 100 MB, 10MB and 1 MB, the total amount of unstructured data inserted as files as well as SecureFile objects is 100 GB.

A Dell 2650 consisting of 2 hyper threaded Intel Xeon 2.8 Ghz processors with 0.5 MB processor cache each, 6GB of RAM and using Red Hat Enterprise Linux 4.0 was used as the client. A Dell 2850 consisting of 2 hyper threaded Intel Xeon 3.2 GHz processors with 2 MB processor cache each, 6GB of RAM, Red Hat Enterprise Linux 4.0 and 2Gbit Fibre Channel SAN Host Adapter was used as the server.

Storage drives are allocated as two identically configured 2TB Raid 5 arrays. One of the units was allocated to Oracle and was managed using Oracle Automatic Storage Management. The other was configured as Ext3 FS made available to the client using NFSv3.

Table 1: Throughput Comparison of NFSv3 and Oracle SecureFiles in MB/sec on read and write operations.

File Sizes (MB)	.01	.1	1	10	100
NFS Performance for Reads	3	8	22	61	75
SecureFiles Performance for Reads	3	12	40	79	88
NFS Performance for Writes	2	13	40	79	74
SecureFiles Performance for Writes	18	61	81	80	80

4.2 Experiment and Results

Table 1 demonstrates the throughput comparison between SecureFiles and NFSv3 on the dataset on single stream reads and writes.

SecureFiles outperforms the NFSv3 access for all sizes with respect to read and write performance. Gains for the smaller file sizes are also due to reduced roundtrips where metadata and data is accessed in one roundtrip unlike the NFSv3 case where metadata and file is accessed in separate roundtrips. Read performance for larger file sizes is contributed by intelligent pre-fetching, larger I/O sizes due to better contiguous space allocations and network optimisations.

5 CONCLUSIONS

Current content management applications use filesystems to store unstructured data due to provision of better throughput of data and access operations across all sizes and types and use database systems to manage accompanying relational metadata for indexing and querying purposes. This dichotomy in storage creates a need for compromises in one or more of high availability, scalability, performance or functionality. With Oracle 11g Database Server's SecureFiles capabilities we now have a next generation unified data management platform without compromises. Performance evaluations demonstrate that Oracle SecureFiles unstructured and relation data management provides optimal execution throughput and scalability for unstructured content while preserving the same for relational content

REFERENCES

Blumberg, R., Atre, S. *The Problem with Unstructured Data*. DM Review Magazine, Feb. 2003.

Hobbs, L. *Information Lifecycle Management with Oracle database 11g*. An Oracle White Paper, June 2007

Vijayan, P. *Iron File Systems.* Thesis Submitted for Doctor of Philosophy in Computer Sciences, University of Wisconsin-Madison, 2006.

Oracle Database 11g Product Family. An Oracle White Paper, January 2008.

Biggar, H. *Experiencing Data De-Duplication: Improving Efficiency and Reducing Capacity Requirements*. A SearchStorage.com White Paper, Feb 2007.