An Oracle White Paper
August 2010

Oracle Database Auditing:

Performance Guidelines

Introduction

Database auditing has become increasingly important as threats to applications become more sophisticated. In fact, the use of Oracle database auditing has steadily increased over the past decade and today is mandatory in many organizations. Surveys conducted with the Independent Oracle User Group (IOUG) have consistently shown over 50% of the respondents use some level of native Oracle database auditing.

Database auditing is generally used to:

- Provide proof of monitoring internal controls to auditors
- Provide reports on changes to the database environment to auditors
- Act as a deterrent to unauthorized activity
- Assist with investigations of data breaches or other suspicious activity
- Detect when an attempt is made to bypass a security control

Database auditing monitors and records activity that occurs in the database. Oracle database auditing has been enhanced with each successive release of the database and today provides highly customizable auditing that can be fine tuned to specific security requirements. Oracle9*i* Database introduced the Fine Grained Auditing (FGA) feature, enabling audit policies to be associated with application tables. Oracle Database 11g enhancements include the ability to audit based on connection factors such as IP address and new built-in manageability features that eliminate the storage and administration costs associated with audit data on the production server.

Perhaps the most common question that arises when it comes to any security solution is "*What about performance overhead?*" This is especially true in the case of commercial organizations where response times and availability impact the bottom line. As with any security control, database auditing does require additional system resources. This paper helps you estimate impact on application throughput and CPU usage when enabling auditing policies in the Oracle database. To understand the impact of Oracle database auditing, let's first review how Oracle auditing is configured.

Database Auditing Overview

Oracle auditing can be divided into two basic categories: standard auditing and FGA. Standard auditing provides the ability to audit based on user, privileges, schemas objects, and statements. For example, it can be based on a specific type of SQL statement (create, alter, update, delete,…). FGA provides the ability to audit access to specific application table columns conditionally based on factors such as IP address or the program name used to connect to the database.

Starting with Oracle Database 11g, the Oracle Database Configuration Assistant (DBCA) can automatically configure Oracle recommended minimum audit settings for compliance and internal controls. These audit settings are associated with important security relevant SQL statements and privileges and are listed in the Oracle security documentation. After creating a database with DBCA, the database will audit the following privileges and SQL statements by default:

ALTER ANY PROCEDURE	CREATE ANY TABLE	GRANT ANY OBJECT PRIVILEGE
ALTER ANY TABLE	CREATE EXTERNAL JOB	GRANT ANY PRIVILEGE
ALTER DATABASE	CREATE PUBLIC DATABASE LINK	GRANT ANY ROLE
ALTER PROFILE	CREATE SESSION	PROFILE
ALTER SYSTEM	CREATE USER	PUBLIC SYNONYM
ALTER USER	DATABASE LINK	ROLE
AUDIT SYSTEM	DROP ANY PROCEDURE	SYSTEM AUDIT
CREATE ANY JOB	DROP ANY TABLE	SYSTEM GRANT
CREATE ANY LIBRARY	DROP PROFILE	
CREATE ANY PROCEDURE	DROP USER	

Table 1 – Oracle Database 11g Default Audit Settings

Oracle Database Audit Trails

The Oracle database can write audit records to a database table or an operating system file. If you choose to write the audit record to an operating system file, you can direct it to be text based, XML formatted, or written directly to the SYSLOG on Unix and Linux or the Event Viewer on Windows. The Oracle database OS audit files can be text based with an extension of '.aud' or XML format with an extension of '.xml'.

Configuring the Oracle Database Audit Trail

The database parameters (<sid>init.ora) that direct standard audit records are as follows:

Parameter	**Value**	**Description**
audit_trail	DB	Write the standard audit content to sys.aud$ table
	DB, EXTENDED	Write standard audit content to sys.aud$ along with the SQL text and bind variable content that was executed for that SQL
	OS	Write the standard audit content to text files
	XML	Write the standard audit content and FGA audit content to an XML formatted file
	XML, EXTENDED	Write the standard audit content and FGA content to an XML formatted file along with SQL text and bind variable content.
audit_sys_operations	TRUE/FALSE	Audits all top-level SYSDBA and SYSOPER activity. These audit records are only written to OS files irregardless of the audit_trail parameter setting.
audit_syslog_level	*<FACILITY_CLAUSE. PRIORITY_CLAUSE>*	Provides level information for the syslog.
audit_file_dest	*<OS_DIRECTORY>*	Specifies the OS directory location to write the OS and XML audit files

Table 2 – Oracle Database Auditing Parameters

FGA enables you to create policies that define specific conditions that must take place for the audit to occur. To add and remove fine-grained auditing, you use the `DBMS_FGA` package. An FGA audit policy allows you to apply it to the specific operations or objects you want to monitor. FGA enables you to monitor data access based on session context or data values.

FGA can provide:

- Boolean condition check. If the Boolean condition you specify is true, for example, a table being accessed on a Saturday, then the audit takes place
- SQL text and bind variables that triggered the audit
- Extra protection to sensitive columns. You can audit specific relevant columns that may hold sensitive information, such as salaries or social security numbers
- Event handler feature to execute additional processing

Audit Trail Management

It is important to manage your audit records properly in order to ensure efficient performance and disk space management. The `DBMS_AUDIT_MGMT` package enables you to efficiently manage your audit trail records by providing the following capabilities.

- Perform cleanup operations on all audit trail types. Audit trails can be cleaned based on the Last Archive Timestamp value. The Last Archive Timestamp represents the timestamp of the most recent audit record that was securely archived.
- Move the database audit trail tables out of the `SYSTEM` tablespace. This improves overall database performance by reducing the load on the SYSTEM tablespace and dedicates an optimized tablespace for audit records.
- Manage your operating system and XML audit files. You can define properties like the maximum size and age of an audit file. This enables you to keep the file sizes of OS and XML audit files in check.

For optimal performance, Oracle recommends the following "auditing best practices" for auditing when writing to database tables and OS files.

- Optimize Space Management - create an audit specific, user-defined tablespace, for example AUDSYS, with pre-created sized (1GB) extents for audit trail tables.
- Load Distribution - move audit trail tables, sys.aud$ and sys.fga_log$ to this new user-defined tablespace.
- Reduce number of OS audit files - set larger file-size (100MB) for OS audit files using the DBMS_AUDIT_MGMT package.

Performance Testing Results

To demonstrate the resources used by the Oracle database with auditing turned on, testing was performed based on the destination where the audit records are being written (database table sys.aud$ or operating system file (OS)) with a TPC-C like workload generating approximately 250 audit records per second. A TPC-C like workload is used since it provides standardized OLTP database activity for complex application environments. The workload is characterized by simultaneous execution of multiple transaction types that span breadth of complexity that are common across all industries.

Machine Configuration

The Oracle Database Release 11.2.0.1 was installed on hardware with the following configuration:

- 4 x 3.40 GHz Xeon CPUs
- 4 GB memory
- x86_64 Linux

The database also included the following patches:

- Remove Oracle database OS flush call after every XML audit write and rely on the Operating System to flush the writes to disk. (9078032)
- Remove XML Index file. (8880803)

Audit Performance Overhead

To simulate a real-world scenario, a standard workload was created to use 50% of system resource before auditing was initiated. For each test run, the following results were recorded:

- Throughput: Additional time used by the transaction after auditing was turned on
- Additional CPU Usage: Measured additional CPU after auditing was turned on

For standard database auditing, a test was created to generate approximately 250 audit records per second using the Oracle database standard audit command.

Audit Trail Setting	Additional Throughput Time	Additional CPU Usage
OS	1.39%	1.75%
XML	1.70%	3.51%
XML, Extended	3.70%	5.26%
DB	4.57%	8.77%
DB, Extended	14.09%	15.79%

Table 3 – Oracle Database 11.2.01 Standard Audit Trail with 50% CPU System Load

For FGA, a test was created to generate approximately 200 audit records per second using the DBMS_FGA package. The condition of the audit policy creates an audit record when an UPDATE or SELECT occurs on the TPCC.ORDL table and the client_identifier value is equal to NULL.

```
dbms_fga.add_policy (
object_schema   => 'TPCC',
object_name     => 'ORDL',
policy_name     => 'Config_A',
audit_condition =>
 'SYS_CONTEXT(''USERENV'',''CLIENT_IDENTIFIER'') IS NULL',
statement_types => 'UPDATE, SELECT',
audit_trail     => DBMS_FGA.XML +DBMS_FGA.EXTENDED);
```

Audit Trail Setting	Additional Throughput Time	Additional CPU Usage
XML	3.66%	4.35%
XML, Extended	4.62%	9.09%
DB	6.60%	11.11%
DB, Extended	9.61%	20%

Table 4 – Oracle Database 11.2.01 Fine Grained Audit Trail with 50% CPU System Load

Based on the results, it shows that writing audit records to OS files, whether that be character based or XML based, has the least impact to system resources. As part of the test, 200+ audit records per second were generated. In general, this is a higher number of audit records than most environments generate. When writing less than 200 audit record per second, you will use less system resources. Results can vary from system to system and should be tested in your environment.

Conclusion

Threats to applications have become more sophisticated and database auditing plays an important part not only in helping detect suspicious behavior but providing proof of controls to auditors. Oracle database auditing has minimal impact on performance even for very high audit trail loads. For optimal performance, Oracle recommends writing database audit records to the operating system (OS) and setting larger file sizes for the OS audit files. Oracle security documentation provides best practice recommendations on minimal audit settings for both compliance and internal controls. Auditing inside the database, should be part of your defense-in-depth architecture.

Oracle is committed to developing practices and products that help protect the environment

Oracle Database Auditing: Performance Guidelines
August 2010
Author: Tammy Bednar
Contributing Authors:

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