10

Analyzing Sampled Data

This chapter describes how to use sampled data to identify transient performance problems in Oracle Database and contains the following topics:

- About Active Session History
- Generating Active Session History Reports
- Interpreting Results from Active Session History Reports

About Active Session History

The Active Session History (ASH) is a diagnostic tool that records the information about all the active sessions in an Oracle database.

The Automatic Database Diagnostics Monitor (ADDM) analysis may not show transient performance problems because they are short-lived. The ASH diagnostic tool captures transient performance problems by taking samples of active sessions every second and storing the sampled data in a circular buffer in the shared global area (SGA). Any session that is connected to the database and is waiting for an event that does not belong to the Idle wait class is considered as an active session. By capturing only active sessions, a manageable set of data is represented with its size being directly related to the work being performed, rather than the number of sessions allowed on the system.

ASH enables you to examine and perform detailed analysis on the sampled session activity using the V\$ACTIVE_SESSION_HISTORY view. The data present in ASH can be rolled up in various dimensions that it captures over a specified duration and gathered into an ASH report.

Note:

ADDM tries to report the most significant performance problems during an analysis period in terms of their impact on DB time. Whether a performance problem is captured by ADDM depends on its duration compared to the interval between AWR snapshots.

If a performance problem lasts for a significant portion of the time between snapshots, it will be captured by ADDM. For example, if the snapshot interval is set to one hour, then a performance problem that lasts for 30 minutes should not be considered as a transient performance problem because its duration represents a significant portion of the snapshot interval and will likely be captured by ADDM.

If a particular problem lasts for a very short duration, then its severity might be averaged out or minimized by other performance problems in the analysis period, and the problem may not appear in the ADDM findings. Using the same example where the snapshot interval is set to one hour, a performance problem that lasts for only 2 minutes may be a transient performance problem because its duration represents a small portion of the snapshot interval and will likely not show up in the ADDM findings.



See Also:

- "Active Session History Statistics" for information about ASH
- Oracle Multitenant Administrator's Guide for information about how manageability features, such as ASH, work in a multitenant container database.

Generating Active Session History Reports

ASH reports enable you to perform analysis of:

- Transient performance problems that typically last for a few minutes
- Scoped or targeted performance analysis by various dimensions or their combinations, such as time, session, module, action, or SQL identifier

ASH reports are divided into multiple sections. The HTML report includes links that can be used to navigate quickly between sections. The content of the report contains ASH information used to identify blocker and waiter identities, their associated transaction identifiers, and SQL statements for a specified duration.

This section describes how to generate ASH reports and contains the following topics:

- User Interfaces for Generating ASH Reports
- · Generating an ASH Report Using the Command-Line Interface

User Interfaces for Generating ASH Reports

The primary interface for generating ASH reports is Oracle Enterprise Manager Cloud Control (Cloud Control). Whenever possible, generate ASH reports using Cloud Control.

If Cloud Control is unavailable, then generate ASH reports by running SQL scripts. The DBA role is required to run these scripts.



Oracle Database 2 Day + Performance Tuning Guide for information about generating ASH reports using Cloud Control

Generating an ASH Report Using the Command-Line Interface

This section describes how to generate ASH reports by running SQL scripts in the command-line interface.

This section contains the following topics:

- Generating an ASH Report on the Local Database Instance
- Generating an ASH Report on a Specific Database Instance
- Generating an ASH Report for Oracle RAC



Generating an ASH Report on the Local Database Instance

The ashrpt.sql SQL script generates an HTML or text report that displays ASH information for a specified duration on the local database instance.

To generate an ASH report on the local database instance using the command-line interface:

1. At the SQL prompt, enter:

```
@$ORACLE HOME/rdbms/admin/ashrpt.sql
```

2. Specify whether you want an HTML or a text report:

```
Enter value for report type: text
```

In this example, a text report is chosen.

3. Specify the begin time in minutes before the system date:

```
Enter value for begin time: -10
```

In this example, 10 minutes before the current time is selected.

Specify the duration to capture ASH information in minutes from the begin time.

```
Enter value for duration:
```

In this example, the default duration of system date minus begin time is accepted.

5. Enter a report name, or accept the default report name:

```
Enter value for report_name:
Using the report name ashrpt 1 0310 0131.txt
```

In this example, the default name is accepted and an ASH report named ashrpt_1_0310_0131 is generated. The report will gather ASH information beginning from 10 minutes before the current time and ending at the current time.

Generating an ASH Report on a Specific Database Instance

The ashrpti.sql SQL script generates an HTML or text report that displays ASH information for a specified duration on a specified database and instance. This script enables you to specify a database and instance for which the ASH report will be generated.

To generate an ASH report on a specific database instance using the command-line interface:

1. At the SQL prompt, enter:

```
@$ORACLE HOME/rdbms/admin/ashrpti.sql
```

Specify whether you want an HTML or a text report:

```
Enter value for report type: html
```

In this example, an HTML report is chosen.

A list of available database IDs and instance numbers are displayed:

```
Instances in this Workload Repository schema

DB Id Inst Num DB Name Instance Host
```



3. Enter the values for the database identifier (dbid) and instance number (inst num):

```
Enter value for dbid: 3309173529
Using 3309173529 for database id
Enter value for inst num: 1
```

4. To generate an ASH report on a physical standby instance, the standby database must be opened read-only. The ASH data on disk represents activity on the primary database and the ASH data in memory represents activity on the standby database.



This step is applicable only if you are generating an ASH report on an Active Data Guard physical standby instance. If this is not the case, then skip this step.

Specify whether to generate the report using data sampled from the primary or standby database:

```
You are running ASH report on a Standby database.

To generate the report over data sampled on the Primary database, enter 'P'.

Defaults to 'S' - data sampled in the Standby database.

Enter value for stdbyflag:

Using Primary (P) or Standby (S): S
```

In this example, the default value of Standby (S) is selected.

5. Specify the begin time in minutes before the system date:

```
Enter value for begin_time: -10
```

In this example, 10 minutes before the current time is selected.

6. Specify the duration to capture ASH information in minutes from the begin time.

```
Enter value for duration:
```

In this example, the default duration of system date minus begin time is accepted.

7. Specify the slot width in seconds that will be used in the Activity Over Time section of the report:

```
Enter value for slot_width:
```

In this example, the default value is accepted. For more information about the Activity Over Time section and how to specify the slot width, see "Activity Over Time".

- **8.** Follow the instructions in the subsequent prompts and enter values for the following report targets:
 - target session id
 - target sql id
 - target_wait_class
 - target service hash
 - target module name



- target action name
- target client id
- target plsql entry
- 9. Enter a report name, or accept the default report name:

```
Enter value for report_name:
Using the report name ashrpt 1 0310 0131.txt
```

In this example, the default name is accepted and an ASH report named <code>ashrpt_1_0310_0131</code> is generated. The report will gather ASH information on the database instance with a database ID value of <code>3309173529</code> beginning from 10 minutes before the current time and ending at the current time.

Generating an ASH Report for Oracle RAC

The ashrpti.sql SQL script generates an HTML or text report that displays ASH information for a specified duration for specified databases and instances in an Oracle Real Application Clusters (Oracle RAC) environment. Only ASH data that is written to disk will be used to generate the report. This report will only use ASH samples from the last 10 minutes that are found in the DBA HIST ACTIVE SESS HISTORY table.

To generate an ASH report for Oracle RAC:

1. At the SQL prompt, enter:

```
@$ORACLE_HOME/rdbms/admin/ashrpti.sql
```

Specify whether you want an HTML or a text report:

```
Enter value for report type: html
```

In this example, an HTML report is chosen.

A list of available database IDs and instance numbers are displayed:

Instances in this Workload Repository schema

DB Id	Inst Num	DB Name	Instance	Host
3309173529	1	MAIN	main	examp1690
3309173529	1	TINT251	tint251	samp251
3309173529	2	TINT251	tint252	samp252
3309173529	3	TINT251	tint253	samp253
3309173529	4	TINT251	tint254	samp254

Enter the values for the database identifier (dbid) and instance number (inst num):

```
Enter value for dbid: 3309173529
Using database id: 3309173529
Enter instance numbers. Enter 'ALL' for all instances in an Oracle RAC cluster or explicitly specify list of instances (e.g., 1,2,3). Defaults to current instance.
Enter value for inst_num: ALL
Using instance number(s): ALL
```

4. Specify the begin time in minutes before the system date:

```
Enter value for begin_time: -1:10
```

In this example, 1 hour and 10 minutes before the current time is selected.

5. Specify the duration to capture ASH information in minutes from the begin time.

```
Enter value for duration: 10
```

In this example, the duration is set to 10 minutes.

6. Specify the slot width in seconds that will be used in the Activity Over Time section of the report:

```
Enter value for slot width:
```

In this example, the default value is accepted. For more information about the Activity Over Time section and how to specify the slot width, see "Activity Over Time".

- 7. Follow the instructions in the subsequent prompts and enter values for the following report targets:
 - target_session_id
 - target sql id
 - target wait class
 - target service hash
 - target module name
 - target action name
 - target client id
 - target plsql entry
- 8. Enter a report name, or accept the default report name:

```
Enter value for report_name:
Using the report name ashrpt_rac_0310_0131.txt
```

In this example, the default name is accepted and an ASH report named <code>ashrpt_rac_0310_0131</code> is generated. The report will gather ASH information on all instances belonging to the database with a database ID value of <code>3309173529</code> beginning from 1 hour and 10 minutes before the current time and ending at 1 hour before the current time.

Interpreting Results from Active Session History Reports

After generating an ASH report, review its contents to identify possible causes of transient performance problems.

The contents of the ASH report are divided into the following sections:

- Top Events
- Load Profile
- Top SQL
- Top PL/SQL
- Top Java
- Top Sessions
- Top Objects/Files/Latches
- Activity Over Time



See Also:

Oracle Real Application Clusters Administration and Deployment Guide for information about sections in the ASH report that are specific to Oracle Real Application Clusters (Oracle RAC)

Top Events

The Top Events section describes the top wait events of the sampled session activity categorized by user, background, and priority. Use the information in this section to identify wait events that may be causing a transient performance problem.

The Top Events section contains the following subsections:

- Top User Events
- · Top Background Events
- Top Event P1/P2/P3

Top User Events

The Top User Events subsection lists the top wait events from user processes that accounted for the highest percentages of sampled session activity.

Top Background Events

The Top Background Events subsection lists the top wait events from backgrounds that accounted for the highest percentages of sampled session activity.

Top Event P1/P2/P3

The Top Event P1/P2/P3 subsection lists the wait event parameter values of the top wait events that accounted for the highest percentages of sampled session activity, ordered by the percentage of total wait time (% Event). For each wait event, values in the P1 Value, P2 Value, P3 Value column correspond to wait event parameters displayed in the Parameter 1, Parameter 2, and Parameter 3 columns.

Load Profile

The Load Profile section describes the load analyzed in the sampled session activity. Use the information in this section to identify the service, client, or SQL command type that may be the cause of a transient performance problem.

The Load Profile section contains the following subsections:

- Top Service/Module
- Top Client IDs
- Top SQL Command Types
- Top Phases of Execution



Top Service/Module

The Top Service/Module subsection lists the services and modules that accounted for the highest percentages of sampled session activity.

Top Client IDs

The Top Client IDs subsection lists the clients that accounted for the highest percentages of sampled session activity based on their client ID, which is the application-specific identifier of the database session.

Top SQL Command Types

The Top SQL Command Types subsection lists the SQL command types—such as SELECT or UPDATE commands—that accounted for the highest percentages of sampled session activity.

Top Phases of Execution

The Top Phases of Execution subsection lists the phases of execution—such as SQL, PL/SQL, and Java compilation and execution—that accounted for the highest percentages of sampled session activity.

Top SQL

The Top SQL section describes the top SQL statements in the sampled session activity. Use this information to identify high-load SQL statements that may be the cause of a transient performance problem.

The Top SQL section contains the following subsections:

- Top SQL with Top Events
- Top SQL with Top Row Sources
- Top SQL Using Literals
- Top Parsing Module/Action
- Complete List of SQL Text

Top SQL with Top Events

The Top SQL with Top Events subsection lists the SQL statements that accounted for the highest percentages of sampled session activity and the top wait events that were encountered by these SQL statements. The Sampled # of Executions column shows how many distinct executions of a particular SQL statement were sampled.

Top SQL with Top Row Sources

The Top SQL with Top Row Sources subsection lists the SQL statements that accounted for the highest percentages of sampled session activity and their detailed execution plan information. You can use this information to identify which part of the SQL execution contributed significantly to the SQL elapsed time.



Top SQL Using Literals

The Top SQL Using Literals subsection lists the SQL statements using literals that accounted for the highest percentages of sampled session activity. You should review the statements listed in this report to determine whether the literals can be replaced with bind variables.

Top Parsing Module/Action

The Top Parsing Module/Action subsection lists the module and action that accounted for the highest percentages of sampled session activity while parsing the SQL statement.

Complete List of SQL Text

The Complete List of SQL Text subsection displays the entire text of the SQL statements shown in the Top SQL section.

Top PL/SQL

The Top PL/SQL section lists the PL/SQL procedures that accounted for the highest percentages of sampled session activity.

The PL/SQL Entry Subprogram column lists the application's top-level entry point into PL/SQL. The PL/SQL Current Subprogram column lists the PL/SQL subprogram being executed at the point of sampling. If the value of this column is SQL, then the % Current column shows the percentage of time spent executing SQL for this subprogram.

Top Java

The Top Java section describes the top Java programs in the sampled session activity.

Top Sessions

The Top Sessions section describes the sessions that were waiting for a particular wait event. Use this information to identify the sessions that accounted for the highest percentages of sampled session activity, which may be the cause of a transient performance problem.

The Top Sessions section contains the following subsections:

- Top Sessions
- Top Blocking Sessions
- Top Sessions Running PQs

Top Sessions

The Top Session subsection lists the sessions that were waiting for a particular wait event that accounted for the highest percentages of sampled session activity.

Top Blocking Sessions

The Top Blocking Sessions subsection lists the blocking sessions that accounted for the highest percentages of sampled session activity.



Top Sessions Running PQs

The Top Sessions Running PQs subsection lists the sessions running parallel queries (PQs) that were waiting for a particular wait event, which accounted for the highest percentages of sampled session activity.

Top Objects/Files/Latches

The Top Objects/Files/Latches section provides additional information about the most commonly-used database resources and contains the following subsections:

- Top DB Objects
- · Top DB Files
- Top Latches

Top DB Objects

The Top DB Objects subsection lists the database objects (such as tables and indexes) that accounted for the highest percentages of sampled session activity.

Top DB Files

The Top DB Files subsection lists the database files that accounted for the highest percentages of sampled session activity.

Top Latches

The Top Latches subsection lists the latches that accounted for the highest percentages of sampled session activity.

Latches are simple, low-level serialization mechanisms used to protect shared data structures in the System Global Area (SGA). For example, latches protect the list of users currently accessing the database and the data structures describing the blocks in the buffer cache. A server or background process acquires a latch for a very short time while manipulating or looking at one of these structures. The implementation of latches is operating system-dependent, particularly regarding if and how long a process waits for a latch.

Activity Over Time

The Activity Over Time section is one of the most informative sections of the ASH report. This section is particularly useful for analyzing longer time periods because it provides in-depth details about activities and workload profiles during the analysis period.

The Activity Over Time section is divided into 10 time slots. The size of each time slot varies based on the duration of the analysis period. The first and last slots are usually odd-sized. All inner slots are equally sized and can be compared to each other. For example, if the analysis period lasts for 10 minutes, then all time slots will 1 minute each. However, if the analysis period lasts for 9 minutes and 30 seconds, then the outer slots may be 15 seconds each and the inner slots will be 1 minute each.

Each of the time slots contains information regarding that particular time slot, as described in Table 10-1.



Table 10-1 Activity Over Time

Column	Description
Slot Time (Duration)	Duration of the slot
Slot Count	Number of sampled sessions in the slot
Event	Top three wait events in the slot
Event Count	Number of ASH samples waiting for the wait event
% Event	Percentage of ASH samples waiting for wait events in the entire analysis period

When comparing the inner slots, perform a skew analysis by identifying spikes in the Event Count and Slot Count columns. A spike in the Event Count column indicates an increase in the number of sampled sessions waiting for a particular event. A spike in the Slot Count column indicates an increase in active sessions, because ASH data is sampled from active sessions only and a relative increase in database workload. Typically, when the number of active session samples and the number of sessions associated with a wait event increases, the slot may be the cause of a transient performance problem.

To generate the ASH report with a user-defined slot size, run the ashrpti.sql script, as described in "Generating an ASH Report on a Specific Database Instance".

