

Troubleshooting: High CPU Utilization (Doc ID 164768.1)

In this Document

[Purpose](#)

[Troubleshooting Steps](#)

[Prevention of High CPU utilization](#)

[Managing High CPU Usage with Resource Manager](#)

[What to look for when multiple processes are using the CPU intensively](#)

[What to look for when one process is using the CPU intensively](#)

[Which process is hogging the CPU?](#)

[Background Processes](#)

[PMON](#)

[SMON](#)

[LGWR & DBWR](#)

[Job processes \(CJQ0, Jn, SNPn\)](#)

[Advanced Queuing \(AQ, QMN\)](#)

[Parallel Query \(Pnn\)](#)

[Oracle \(User\) processes](#)

[OS Processes and defunct processes](#)

[Finding High CPU Utilization processes on Windows](#)

[Finding High CPU Utilization processes on Unix](#)

[Find Top Activity and Top CPU Session Using ASH Tab From Grid Control](#)

[References](#)

APPLIES TO:

Oracle Database - Personal Edition - Version 7.1.4.0 and later
Oracle Database - Standard Edition - Version 7.0.16.0 and later
Oracle Database - Enterprise Edition - Version 7.0.16.0 and later
Generic UNIX
Generic Windows

PURPOSE

The purpose of this document is to help in diagnosing the reason for Oracle related processes consuming high CPU

TROUBLESHOOTING STEPS

Prevention of High CPU utilization

High CPU utilization may not necessarily indicate that there is a problem; it could just mean that the system is being well utilized. However, high CPU usage means that any new operations may start to interfere with the current usage. Since there is no room for growth, they can start to exhibit signs of performance degradation.

You should most likely investigate the reason for the high CPU usage if:

- CPU usage is consistently high when the load on the system is low
- System performance is poor together with High CPU usage
- One or more processes are consistently hogging CPU at the expense of other processes

Other than collecting diagnostic information to determine the root cause of the problem, there is little or nothing to do to stop processes from using a lot of CPU once they started to do so. On the other hand, a lot can be done to prevent it from happening.

Oracle provides two ways to limit the CPU being used by individual users :

- User Profiles can be configured to terminate sessions that exceed a specified amount of CPU time per call or CPU time per session - See:

[Document 1016552.102](#) How to use PROFILES to limit user resources

https://docs.oracle.com/database/121/SQLRF/statements_6012.htm#SQLRF01310

- Resource Manager offers more flexibility than User Profiles in handling runaway database processes by switching potential resource hogging sessions into a different resource group with lower resource limits or to a resource group that will cancel a query (CANCEL_SQL group) or kill the session (KILL_SESSION group) based on how long a SQL has been executing (SWITCH_TIME) or even anticipated execution time (ESTIMATED_SWITCH_TIME). A parallel degree limit can also be set by group so that as CPU resources become tight only specific groups of users can use more or less PQ slave processes (therefore fewer CPUs) than other groups.

[Document 106948.1](#) Database Resource Manager samples

[Document 471265.1](#) Example: How to control CPU Resources using the Resource Manager

https://docs.oracle.com/cd/B19306_01/server.102/b14231/dbrm.htm

Managing High CPU Usage with Resource Manager

If individual sessions are using excessive CPU, then you might be able to throttle their activity using resource manager such that they do not affect other activities on the database. See:

[Document 1600965.1](#) Managing and Monitoring Runaway Query Using Resource Manager
White Paper: [Effective Resource Management Using Oracle Database Resource Manager](#)

What to look for when multiple processes are using the CPU intensively

Sometimes the CPU is fairly distributed across several processes. In that case, investigate if they share a common task like the execution of a particular PL/SQL package or SQL statement. We recommend that you take a few manual AWR or statspack snapshots while the CPU usage is at its peak and generate reports against those snapshots. From there, you can look at the "Top Timed Events" section to verify where the time was spent and investigate the SQL listed in the "Top SQL ordered by CPU" section.

If the problem only occurs when a particular application is active, ensure that the application is not overwhelming the underlying server hardware with more connections than it can handle. An application architecture, that is not designed to use persistent connections (connection pooling, shared servers) thus generating much connection and disconnection activity while using dedicated servers, wastes large amounts of CPU. They can cause an unstable "login / connection storm" effect.

What to look for when one process is using the CPU intensively

The approach to take here depends on the type of process involved. Determine the type of process and then act accordingly as outlined below:

Which process is hogging the CPU?

- Background process
- Oracle (user) process
- OS Process that is not related to Oracle
- Defunct process

Background Processes

Background processes should rarely be a source of runaway CPU consumption. A few known historical issues are outlined below and categorized by background process.

PMON

Generally reasons for high CPU usage by PMON are specific bugs related to cleaning up database processes that died abnormally or registering with the Oracle Net listener.

SMON

SMON is responsible for space consolidation and transaction recovery operations which can cause significant overhead if you are using dictionary managed tablespaces.

SMON can bring a database to a halt if a large table with many extents is dropped or truncated and the table resides within a dictionary-managed tablespace. Starting in 9i and later, locally-managed tablespaces are the default when a tablespace is created, and in Database 9i Release 2 (9.2.x) and higher, the system tablespace may be created as locally-managed as well.

Excessive tablespace extent consolidation can consume excessive CPU. See the following note for more information on this and how to troubleshoot.

[Document 61997.1](#) SMON - Temporary Segment Cleanup and Free Space Coalescing

Dead transaction recovery operations performed by SMON can also consume high CPU. The following note explains when the SMON is doing recovery and what to do about it:

[Document 414242.1](#) Database Hangs Because SMON is taking 100% CPU doing transaction recovery

SMON may do transaction recovery in parallel. This may result in considerable CPU consumption. In such cases you may consider disabling parallel recovery:

[Document 144332.1](#) Parallel Rollback may hang database, Parallel query servers get 100% CPU

LGWR & DBWR

These two processes are usually restricted by the speed of reads or writes (ie they are I/O bound), but when there is a problem on the OS, they may "spin" (wait) until the I/O operation completes. Spinning is a state whereby a process consumes CPU by repeatedly executing the same set of instructions until some event happens. Slowness or failures in async I/O (AIO) operations may also manifest themselves by high CPU consumption.

If LGWR appears to be intermittently taking up 100% CPU and AIO is setup, then the AIO configuration should be rechecked. As a temporary measure, the following parameter may be set time to prevent LGWR from spinning:

```
_lgwr_async_io=false
```

This parameter disables async i/o for LGWR leaving it intact for the rest of the database. For more information see:

[Document 813473.1](#) LGWR Uses Async I/O In spite Of Setting _lgwr_async_io=False
[Document 7385253.8](#) Bug 7385253 - Slow Truncate / DBWR uses high CPU / CKPT blocks on RO enqueue

Job processes (CJQ0, Jn, SNPn)

Job processes run user defined and system defined batch-like tasks. The high CPU usage should be investigated in a similar way as when investigating CPU used by a user process. Check the Views DBA_JOBS_*, DBA_SCHEDULER_*, DBA_AUTOTASK_* for information of what is being run. Even on their own these processes may consume a fair amount of CPU as they are in an infinite loop querying the job queue. See:

[Document 8531434.8](#) Bug 8531434 - Solaris: Excessive CPU by MMNL/CJQ0 when running multiple instances and cpus

Advanced Queuing (AQ, QMN)

The AQ processes send and receive messages mostly through tables. Excessive CPU utilization may be because the tables need to be purged or reorganized or other issues related to Advanced Queuing. See:

[Document 305662.1](#) Master Note for AQ Queue Monitor Process (QMON)
[Document 271855.1](#) Procedure to manually Coalesce all the IOTs/indexes Associated with Advanced Queueing tables to maintain Enqueue/Dequeue performance, reduce QMON CPU usage and Redo generation

Parallel Query (Pnn)

Parallel query processes are used specifically in order to do a lot of work and therefore may indeed high CPU. However it is advised to ensure that the system is set up optimally. The parallel query option is best for data warehouse type environments where only a small number of users will be executing queries at any given time. See:

[Document 203238.1](#) Using Parallel Execution.

Oracle (User) processes

Parsing large queries, procedure compilation or execution, space management and sorting are examples of operations that are CPU intensive. In order to collect more information on a process using high CPU see the following notes that may be of assistance:

[Document 352648.1](#) How to Diagnose High CPU Usage Problems to the Module Level
[Document 452358.1](#) How to Collect Diagnostics for Database Hanging Issues

An AWR or statspack report may also be of assistance in diagnosing which activities are using high CPU and what they are doing:

[Document 748642.1](#) How to Generate an AWR Report
[Document 94224.1](#) FAQ- Statspack Complete Reference
[Document 276103.1](#) PERFORMANCE TUNING USING 10g ADVISORS AND MANAGEABILITY FEATURES

If the problem is found to be a slow running query, then efforts should be made to tune the query in order that it may avoid consuming high CPU. If it is doing a number of hash joins and full table scans, efforts should be made to add indexes and get the indexes used.

The following notes assist in diagnosing problems with queries and assist in tuning them.

[Document 215187.1](#) SQLT (SQLTXPLAIN) - Tool that helps to diagnose SQL
[Document 199083.1](#) Master Document SQL Query Performance Overview

Real-time SQL monitoring is a 11g new feature that enables you to monitor the performance of SQL statements while they are executing. See:

[Document 1229904.1](#) Real-Time SQL Monitoring in 11g

Other tracing techniques might be useful to decide whether to allow a process to continue or not and help with analyzing the reason for high CPU usage.

[Document 376442.1](#) How To Collect 10046 Trace (SQL_TRACE) Diagnostics for Performance Issues

OS Processes and defunct processes

As OS processes are not related to Oracle we cannot help in diagnosing the cause behind the CPU usage. Please consult with your OS Vendor. The following notes include information on some known OS issues:

[Document 316533.1](#) AIX: Database performance gets slower the longer the database is running
[Document 217990.1](#) Hpux_sched_noage 178
[Document 580273.1](#) Processes Hang Waiting on 'cursor: pin S wait on X' (and other Wait Events) on HP-UX 11.23 and 11.31 Itanium Systems
[Document 7228334.8](#) More CPU utilization on IBM Linux on POWER
[Document 285026.1](#) Oracle Threads Not Showing In V\$Process and Consuming CPU Background (instance) process
[Document 361670.1](#) Slow Performance with High CPU Usage on 64-bit Linux with Large SGA

Finding High CPU Utilization processes on Windows

Following notes may help to collect information on which processes on Windows may be utilizing high CPU:

[Document 273646.1](#) How to diagnose the high CPU utilization of ORACLE.EXE in Windows environment
[Document 116236.1](#) Diagnosing High CPU Utilization on NT
[Document 433472.1](#) OS Watcher For Windows (OSFWF) User Guide

Finding High CPU Utilization processes on Unix

On UNIX systems there are two basic tools with the capability of evaluating and estimating the CPU usage on the system. These are vmstat and sar. The following articles explain how to use them and

other tools in diagnosing high CPU on Unix:

[Document 148466.1](#) Monitor Oracle Resource Consumption in UNIX

[Document 224176.1](#) How to use OS commands to diagnose Database Performance issues?

[Document 301137.1](#) OS Watcher User Guide

Find Top Activity and Top CPU Session Using ASH Tab From Grid Control

Review youtube video on how to use ASH tab to review top activities, high CPU users, etc.

This shows a graphical way to drill down from a top timed event to the session/SQL most responsible for each component:

<https://www.youtube.com/watch?v=awFDEYdLJTY>

- User Profiles can be configured to terminate sessions that exceed a specified amount of CPU time per call or CPU time per session - See:

Didn't find what you are looking for?