International Journal of Trend in Scientific Research and Development (IJTSRD)
Volume 5 Issue 1, November-December 2020 Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

IO Management in Oracle

Vijay Kumar Tiwari

IT Consultant, HCL America Inc, Texas, United States

How to cite this paper: Vijay Kumar Tiwari "IO Management in Oracle" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-5 | Issue-1, December 2020, pp.1466-1467, URL: www.ijtsrd.com/papers/ijtsrd38276.pdf

IJTSRD38276

I/O tuning in Oracle Database

I/O operations in UNIX and Linux systems typically go through the file system cache. Although this doesn't represent a problem in itself, this extra processing does require resources. Bypassing the file system cache reduces CPU requirements, and frees up the file system cache for other non-database file operations. Operations against raw devices automatically bypass the file system cache.

When a synchronous I/O request is submitted to the operating system, the writing process blocks until the write is complete before continuing processing. With asynchronous I/O, processing continues while the I/O request is submitted and processed. This allows asynchronous I/O to bypass some of the performance bottlenecks associated with I/O operations.

Checking your Server for direct I/O

Methods for configuring the OS will vary depending on the operating system and file system in use. Here are some examples of quick checks that anyone can perform to ensure that you are using direct

- **Linux Operating System** - Linux systems support direct I/O on a per-filehandle basis (which is much more flexible). Also with 10g and beyond, this feature is already working which means that is does not require any patch. For Oracle, the DBA will need to download - Abstract: DIRECT IO SUPPORT OVER NFS. To enable direct I/O

check these settings:

- Set the filesystemio_options parameter in the parameter file to DIRECTIO (filesystemio_options = DIRECTIO)
- If the asynchronous I/O option is in use, the filesystemio_options parameter in the parameter file should be set to SETALL.

Oracle can take advantage of direct I/O and asynchronous I/O on supported platforms using the FILESYSTEMIO_OPTIONS parameter, whose possible values are listed below.

- ASYNCH - Enabled asynchronous I/O where possible.
- DIRECTIO- Enabled direct I/O where possible.
- SETALL- Enabled both direct I/O and asynchronous I/O where possible.

NONE - Disabled both direct I/O and asynchronous I/O.

Most of the performance tuning issues can be related to I/O in any database. Oracle provides only two main parameters to control I/O behaviour these are filesystemio_options and disk_asynch_io

filesystemio_options alllows you to specify synchronous and asynchronous read/write and direct and indirect read write. By default it is none it means operating systen's default I/O mode is selected which is synchronous read/write and indirect read write that is file system cached read write operations. But since disk_asynch_io parameter defaults to true Oracle supports asynchronous read write by default.

Oracle recommends to set parameter filesystemio_options to value 'setall' but it is not always good practise especially when SGA is small. setting it to setall lets your Oracle DB perform I/O operations without going to file system cache and it saves overhead of double caching but if SGA is smaller and DB host machine has large free memory then it is not good to set this parameter to value setall. In this case you should increase DB_CACHE_SIZE and only then set filesystemio_options to setall.

Five years ago I was setting up Oracle database to handle large I/O of new Dataware house project. I set filesystemio_options to setall. This speeded up load operations aproximately by 5% whose caching in file system was not useful as these were insert statements and not repeated. At the same time increased SGA to quite high to cache I/O as much as possible so that repetitive select queries can benefit from SGA as in this case when filesystemio_options was set to setall then file system cache was not available.

So summary is be prudent when setting filesystemio_options to setall to enable direct read/write and asynchronous operations.

Other parameters to affect write (as well as read) is dbwriter_processes. When asynchronous I/O operations are slower in operating system in comparison to synchronous I/O then turn off asynchronous I/O by setting disk_asynch_io to false and set multiple db writer processes by increasing dbwriter_processes values from 1 to 2,3 or 4 suitable value to your system. Alternate is incrase dbwr_io_slaves from 0 to 2,3,4 suitable value.

You would be keen to disable asynchronous I/O when you see high average_wait on event db_file_parallel_wait. Other reason for turning it off will be synchronous I/O is more reliable.

SQL> select event, average_wait from v$system_event where event like 'db file parallel write';

This is not a very good ASYNCH I/O. Try Synchronous I/O Note 1: Asynchronous I/O operations are more prone to block corruptions than synchronous operations so many DBAs disable it and follow practice as mentioned in above paragraph. So if you do not have standby databases and oracle 11g then which automatically recovers corrupted block on primary then you would not want asynchronous I/O