An Oracle White Paper
June 2011

Recovery Manager (RMAN) Performance Tuning Best Practices

Introduction

A complete high availability and disaster recovery strategy requires dependable data backup, restore, and recovery procedures. Oracle Recovery Manager (RMAN) provides a comprehensive foundation for efficiently backing up and recovering the Oracle database. It is designed to work intimately with the database server, providing block-level corruption detection during backup and restore. RMAN optimizes performance and space consumption during backup with file multiplexing and backup set compression, and integrates with Oracle Secure Backup, as well as third party media management products, for tape backup.

RMAN takes care of all underlying database procedures before and after backup or restore, freeing dependency on OS and SQL*Plus scripts. It provides a common interface, via command line and Enterprise Manager, for backup tasks across different operating systems and offers features not available through user-managed methods, such as block-level incremental backup, parallelization of backup/restore data streams, database-aware retention policies, and detailed history of all backups.

As database sizes continue to grow unabated, RMAN performance tuning is more critical than ever to satisfy backup and recovery windows. This paper presents an inside look at RMAN backup, restore, and recovery operations, including a step-by-step tuning methodology to diagnose performance bottlenecks and implement the database and system changes to achieve the desired performance characteristics. This paper assumes the reader has an understanding of RMAN concepts and features. All content in this paper assumes Oracle Database 11*g* Release 2, unless otherwise noted.

Backup Data Flow

An RMAN backup or restore job can be divided into separate phases or components, consisting of the read phase, copy phase, and write phase. The slowest of these phases causes the *bottleneck* and will determine the theoretical maximum backup throughput.

The goal of RMAN performance tuning is to identify the bottleneck for a given job and use RMAN commands, initialization parameters, or adjustments to physical media to improve performance.

Before diving into a hands-on diagnosis of performance bottlenecks, it is helpful to first understand the components involved in the overall backup data flow.

The work of a backup is performed by one or more channels. A *channel* represents a stream of bytes to a storage device. A byte stream passes from the input buffers in memory (either the PGA or SGA) to the output buffers. From there, the backup data is written to the storage device. A diagram of the process is shown below.

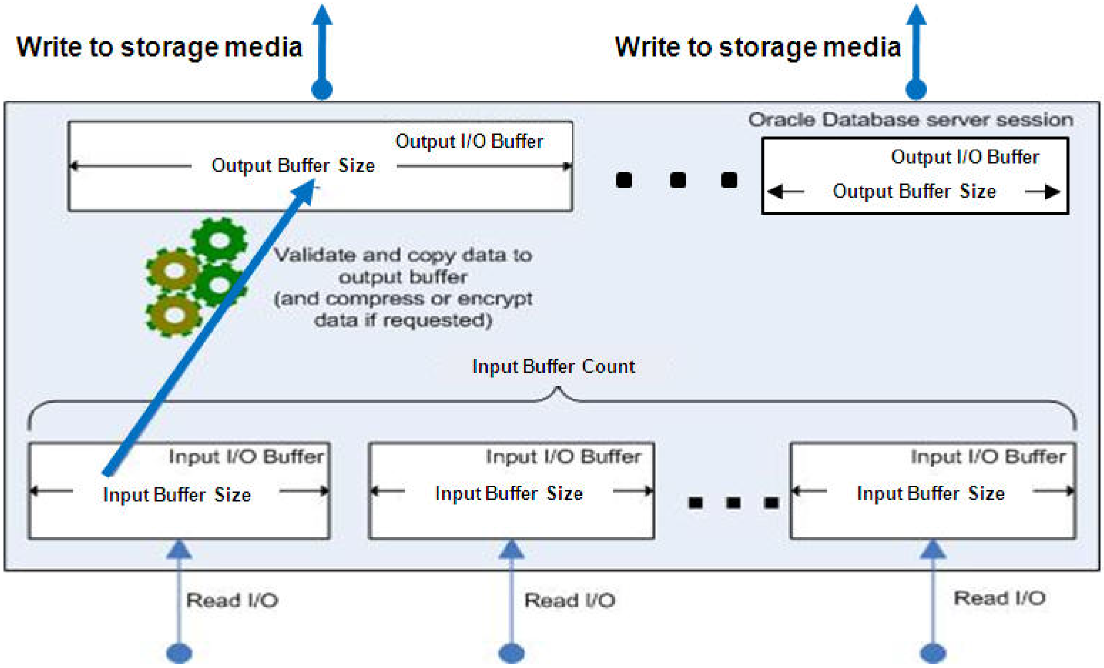

Figure 1 RMAN Backup Data Flow

The work of each channel, whether of type `DISK` or System Backup to Tape (`SBT`), is divided into the following distinct phases:

- Read Phase
    - The channel reads blocks from the data files into input I/O buffers.

- Copy Phase
  - The channel copies blocks from input buffers to output buffers and performs additional processing on the blocks, including validation and optional compression and/or encryption.
- Write Phase
  - The channel writes the blocks from output buffers to storage media. In the case of SBT, the media manager handles the writes to the tape device.

With an understanding of the backup phases, performance tuning principles for each phase are now discussed

Performance Tuning Principles

Prior to examining the core principles for tuning each backup phase, it is worth noting that `V$BACKUP_ASYNC_IO` is a key RMAN backup performance statistics view. This view assists in determining bottlenecks when the underlying operating system directly supports asynchronous I/O or asynchronous I/O is simulated by enabling database slave processes. For each backup, the view contains one row for each input datafile, one row for the aggregate total performance of all datafiles, and one row for the output backup piece. This data is not preserved when the database instance is re-started. More information on the view can be found in the Oracle Database Reference Guide (http://download.oracle.com/docs/cd/E11882_01/server.112/e17110/dynviews_1047.htm#REFRN30018).

In addition, Automatic Workload Repository (AWR) and Enterprise Manager can be used to monitor system CPU and I/O utilization during backup operations - these tools have the advantage of preserving historical performance data, no matter if the instance is re-started. More information on these tools can be found in the Oracle Database Performance Tuning Guide (http://download.oracle.com/docs/cd/E11882_01/server.112/e16638/autostat.htm).

For the rest of this paper, “input disk” refers to the disk subsystem that RMAN is reading from during the backup and “output device” refers to the disk or tape device to which the backup is being written.

Read Phase

The objective of tuning the read phase is to maximize RMAN read I/O on the production disk.

Principle #1: Determine the maximum throughput for input disk

It is helpful to determine the raw input disk throughput as a starting point in the tuning analysis, as it will typically be the highest performing component in the backup data flow.

The Oracle Orion Calibration tool can be used to simulate various database workloads (e.g. OLTP, DSS) and measure performance metrics, such as IOPS, MBPS, and I/O latency. More details on using

Oracle Orion can be found in the Oracle Performance Tuning Guide (http://download.oracle.com/docs/cd/E11882_01/server.112/e16638/iodesign.htm#CACJEEDI).

Principle #2: Configure disk subsystem for optimal performance

Once the baseline disk performance is determined, ensure that Oracle I/O can most efficiently utilize the underlying disk subsystem. Automatic Storage Management (ASM) is the recommended method to configure Oracle database storage. More details on deploying and configuring ASM can be found in the Automatic Storage Management Administrator's Guide (http://download.oracle.com/docs/cd/E11882_01/server.112/e16102/toc.htm).

To ensure that production workload I/O does not conflict with backup workload I/O, it is advised to create separate disk groups, one for data files (e.g. DATA) and one for backups (e.g. FRA), and to separate the physical disks between the disk groups. If the physical disks cannot be separated, then it is advised to configure the outer sectors of the physical disks for the DATA disk group, which provides excellent random I/O performance for data files, and configure the inner sectors for the FRA disk group, which provides good sequential I/O performance for writing backups.

If ASM is not deployed, then all data files should be striped across available disks with a 1 MB stripe size.

Principle #3: Tune RMAN to fully utilize disk subsystem and tape

Configuring RMAN to fully consume available disk and tape I/O is the next objective to focus on.

Ideally, the underlying operating system supports asynchronous I/O. If so, RMAN automatically takes advantage of this. However, if the operating system does not, the database can simulate asynchronous I/O with `DBWR_IO_SLAVES`. By setting this parameter to a nonzero value, 4 slave processes will be allocated per RMAN session. This is especially important for backups to disk, in order to fully take advantage of available I/O.

By default, backup to tape is a synchronous process, i.e. the SBT write operation must be acknowledged before the copy-to-output buffer process continues. To reduce any possible impact, it is recommended to set `BACKUP_TAPE_IO_SLAVES` to `TRUE` so that SBT write operations can proceed asynchronously with respect to the copy-to-output buffer process. The only exception is if the media management product documentation states otherwise.

The next task is to properly allocate channels for optimal backup parallelism. For disk backups, this means allocating as many channels as are needed to get acceptable performance – each channel will use memory, CPU, and I/O just like a standard server process, so monitor these resources carefully. For image copies, where RMAN creates a one-to-one backup copy of each datafile, one channel can only process one data file at a time, so parallelism becomes an especially key factor in performance tuning. Typically, there will be a point at which adding more channels will not result in much more overall performance gains.

For tape backups, one channel should be allocated per tape drive. This recommendation stems from the fact that allocating a higher number of channels than tape drives will result in tape-side file multiplexing where two or more backup pieces are interleaved on the same tape. Because of this behavior, restore time will degrade due to channel contention, i.e. one channel will attempt to read its backup file from tape while another attempts to read its backup file from the same tape. However, in many cases, increasing the number of channels beyond the number of tape drives appears to increase overall backup throughput and so, is considered prudent 'tuning'. This practice typically masks a performance problem that is not directly related to number of channels being used. Rather, the actual bottleneck may be the input buffers used during the read phase or the write phase, which may include the media management environment. Instead of suffering degraded restore time when the number of channels is greater than the number of tape drives, it is better to invest the time and analysis to determine where the bottleneck is actually located.

A simple test is to first run `BACKUP .. VALIDATE` with the higher number of channels than tape drives and note the duration of this operation – this is the ***read phase time A***. Then, set the number of channels equal to number of tape drives and repeat the same test, noting this duration as ***read phase time B***.

If read phase time A is less than B, then the bottleneck is most likely in the read phase itself and must be further tuned. This will be discussed in the next section "Read Phase Tuning".

If read phase time A equals B (i.e. read phase times are about the same), then the bottleneck is more likely in the write phase, which may include the media management environment. Specific recommendations for this case are in the section "Diagnosing Performance Bottlenecks – Copy/Write Phase".

Read Phase Tuning

After the input disk has been optimally configured and its baseline throughput been determined according to the above steps, we can turn to the RMAN-specific aspects of tuning. These primarily consist of file multiplexing and memory buffer usage during the backup. RMAN multiplexing is defined as the number of files read concurrently by one channel during a backup. Multiplexing is defined by the minimum of `MAXOPENFILES` parameter (default of 8) in the `ALLOCATE CHANNEL` or `CONFIGURE CHANNEL` command and `FILESPERSET` parameter (default of 64) in the `BACKUP COMMAND`. `MAXOPENFILES` specifies the maximum number of data files that can be opened for reading by a single channel at any time. `FILESPERSET` specifies the maximum number of data files that will be placed into each backup set.

A diagram where RMAN multiplexing=3, as specified by `MAXOPENFILES=3,` is shown below.

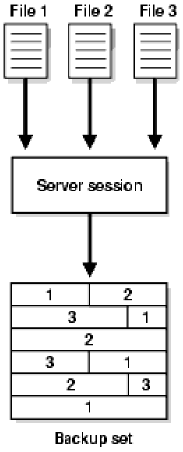

Figure 2 RMAN Backup with `MAXOPENFILES=3`

The multiplexing value dictates the number and size of input memory buffers utilized during the backup, as shown below:

- **MULTIPLEXING ≤ 4**
  - Each buffer = 1MB, total buffer size for channel is up to 16MB
- **4 < MULTIPLEXING ≤ 8**
  - Each buffer = 512KB, total buffer size for channel is up to 16MB. Number of buffers per file will depend on number of files.
- **MULTIPLEXING > 8**
  - Each buffer = 128KB, 4 buffers per file, so each file will have 512KB buffer

The actual number and size of buffers used during a backup can be found in `V$BACKUP_ASYNC_IO` (assuming asynchronous I/O or if disk slaves are enabled), under the `BUFFER_COUNT` and `BUFFER_SIZE` columns. All buffers are allocated from the PGA, unless disk or tape I/O slaves are enabled. If slaves are enabled, then buffers are allocated from the SGA[1] .

For ASM and striped storage, if the default `MAXOPENFILES` setting does not yield adequate read phase performance, it is recommended to reduce this value, in order to allocate more buffers and larger size buffers per file. For example, by setting `MAXOPENFILES=1`, the most number of (16) and largest sized (1 MB) buffers per file will be allocated during the backup. Since the data files are striped across all disks (i.e. data blocks are spread evenly across disks), using as many input buffers as possible will allow the overall disk I/O utilization to be maximized.

[1] If slaves are enabled, RMAN buffers will use `LARGE_POOL`, if set.

For non-striped storage, it is recommended to use the default setting of MAXOPENFILES=8 (i.e. 4 buffers per file with each sized 512 KB). Since each file resides entirely on one disk, this will allow each channel to read concurrently from multiple disks, to more effectively spread out I/O usage.

Once the appropriate number of channels and MAXOPENFILES has been set, re-run BACKUP VALIDATE and monitor the EFFECTIVE_BYTES_PER_SECOND column of V$BACKUP_ASYNC_IO.

If EFFECTIVE_BYTES_PER_SECOND is still less than the expected baseline disk performance and there is available CPU and memory, then manually increase the number and size of buffers. With Oracle Database 11*g* Release 11.1.0.7 and earlier versions, increase the *Input Buffer Count* (as shown in Figure 1), using _BACKUP_KSFQ_BUFCNT (default 16). This will set the number of input buffers allocated per file. Carefully monitor memory usage as different settings are tested, e.g. setting this parameter to 500 for 500 physical disks may exceed tolerable memory consumption.

Set the *Input Buffer Size*, using _BACKUP_KSFQ_BUFSZ (default 1048576 bytes, i.e. 1 MB), equal to the allocation unit size for ASM environments. It can be remain at the default 1 MB value for non-ASM environments.

With Oracle Database 11g Release 2, these parameters are expanded to address input and output buffers separately as well as buffers used for DISK and SBT backups. Refer to Support Note 1072545.1 for details on these parameters.

Note that with Oracle Database 11g Release 2, no buffer parameter changes should be needed when backing up from or to ASM, as all disk buffers are automatically sized for optimal performance. However, if the desired EFFECTIVE_BYTES_PER_SECOND is not achieved, then the buffers can still be manually adjusted.

Copy Phase

During the copy phase, the input buffer blocks are validated for physical corruptions (default) and logical corruptions (optional). The blocks are then compressed and/or encrypted (if specified for the backup) and then copied to the output buffers, as shown below:

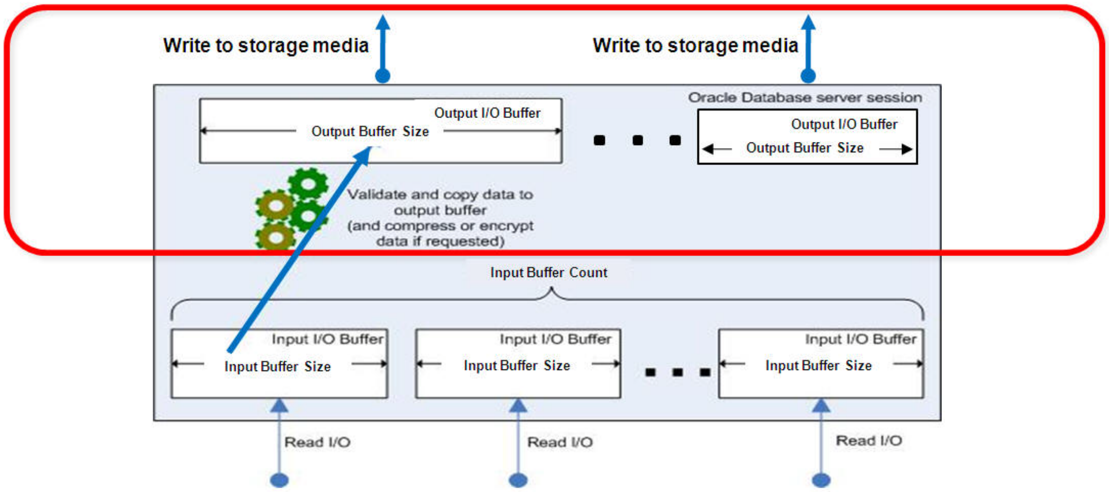

Figure 3 RMAN Backup Data Flow – Copy and Write Phases

Generally, the in-memory copy operation utilizes minimal resources, but the copy phase can become a bottleneck due to high CPU utilization when compression and/or encryption are requested. Therefore, the primary goal of tuning the copy phase is to minimize the resource impact due to compression and encryption.

Principle #4: Tune RMAN backup compression and encryption to minimize resource contention

The basic tuning options are to (a) increase CPU resources, (b) use a faster compression (LOW/MEDIUM) or encryption setting (AES128), and/or (c) allocate number of channels equal to the number of CPUs/cores, bearing in mind the potential impact to user workloads, as more channels are allocated.

In particular, if Transparent Data Encryption (TDE) is utilized in the database and encrypted and/or compressed backups are specified, CPU utilization can be affected, depending on whether TDE column or tablespace encryption is used, as shown below.

- TDE column encryption
  - If compressed & encrypted backups are specified, data is compressed first, then encrypted. Encrypted columns are treated as if they were not encrypted - thus, CPU utilization can be higher during the backup.
  - If only encrypted backups are specified, data is double encrypted – thus, CPU utilization can be higher during the backup.
- TDE tablespace encryption
  - If compressed & encrypted backups are specified, TDE-encrypted tablespaces are decrypted, compressed, then re-encrypted – thus, CPU utilization can be higher during the backup

- If only encrypted backups are specified, TDE-encrypted blocks pass through the backup unchanged – thus, there is no additional effect to CPU utilization.

Write Phase

The goal of tuning the write phase is to ensure that the output buffers are written out as quickly as possible to the backup device. Backups to DISK channels are written out by the RMAN channels themselves while backups to SBT channels are written via the media management software.

Principle #5: Determine baseline throughput of disk backup storage and/or tape device

If backups are taken to disk, the raw throughput of the disk backup storage (e.g. Fast Recovery Area) should be determined via Oracle Orion Calibration tool or other I/O measurement tool.

Secondly, the network throughput should be evaluated between all connection points, e.g. between the database host and media server. TCP/IP performance measurement tools, such as `qperf`, are helpful for this task. The media management environment (e.g. media server) network throughput to the tape system should also be evaluated.

Finally, the tape device throughput should be validated. In many cases, the number of available tape drives limits overall backup performance. Tools such as the `DD` operating system command can be utilized to measure tape drive throughput. In many cases, the advertised tape device throughput specifications are theoretical and do not take into account other operational factors, such as load/unload time, rewind/seek time, etc. – remember to always test with actual workloads.

Principle #6: Tune RMAN output buffers to increase write performance

By default, 4 output buffers are allocated per channel, whether `DISK` or `SBT`. The default *Output Buffer Size* is 1 MB (for `DISK`) and 256 KB (for `SBT`). For `SBT`, this buffer size can be modified using the `BLKSIZE` channel parameter.

For disk backups, monitor `EFFECTIVE_BYTES_PER_SECOND` in `V$BACKUP_ASYNC_IO`, where type=`'OUTPUT'`. This should be close to the disk backup storage throughput. If it is not and assuming the read phase has been properly tuned (as previously discussed), the output buffer count & size can be manually adjusted per Support Note 1072545.1.

Note that with Oracle Database 11g Release 2 & backing up to ASM disk, the output buffers are automatically set and sized for optimal performance.

Diagnosing Performance Bottlenecks

With an understanding of the RMAN backup data flow and its components, in addition to key tuning principles, we can now discuss the performance bottleneck diagnosis and tuning methodology.

The first step is to query `EFFECTIVE_BYTES_PER_SECOND` column for the ‘`AGGREGATE`’ row in `V$BACKUP_ASYNC_IO`, during the backup. If `EFFECTIVE_BYTES_PER_SECOND` does not match the expected output device throughput, then determine the read phase throughput by running `BACKUP VALIDATE` and re-monitoring the `EFFECTIVE_BYTES_PER_SECOND` column.

Read Phase Bottlenecks

If the BACKUP VALIDATE throughput is much less than the input disk throughput, then the read phase is the most likely bottleneck and should be tuned appropriately, as discussed in the Read Phase Tuning section. In addition to the guidance given for RMAN multiplexing and buffer usage parameters, there may be a possibility that some files have slower access patterns than others, in non-ASM environments. This can be determined by finding the data files with an above average (LONG_WAITS/IO_COUNT) ratio from V$BACKUP_ASYNC_IO**.** Consider moving these 'slow' files to a new disk and re-run BACKUP .. VALIDATE.

Copy and Write Phase Bottlenecks

If the observed read phase throughput is close to the expected input disk throughput, but the EFFECTIVE_BYTES_PER_SECOND value for the 'AGGREGATE' row is less than the output device throughput, then look to the copy or write phase as the most likely bottleneck.

If compression and/or encryption is used, refer to Principle #4: Tune RMAN compression and encryption to minimize resource contention

In the case of a disk backup, consider increasing the number of output buffers as prescribed in Support Note 1072545.1.

For tape backup, verify and tune the media manager performance, if it does not match the tape device performance itself. Refer to the media management vendor's performance tuning guide for specific tuning guidelines.

Restore Performance Tuning

While the paper thus far has discussed backup-specific tuning guidelines, the restore process is just as, or even more, important to optimize. The restore data flow is an inverse of the backup data flow. For example, any adjustments to the RMAN buffer parameters to speed up backups will also be leveraged on restore. Be aware that optimal restore performance assumes that the required storage media (e.g. tape drives) are immediately available, as they were during the backup. Any delays, such as wait times for accessing the device, loading tape, etc. must be factored in for the restore process.

Recovery Performance Tuning

While most tuning analysis can be spent on backup and restore operations, recovery performance should also be evaluated. The following guidelines can help increase recovery performance:

- Minimize archive log application by using more frequent incremental backups
- Use block media recovery for isolated block corruptions
- Keep adequate number of archived logs on disk
- Tune the database for adequate I/O, DBWR performance, and CPU utilization

- o Refer to redo apply best practices in the MAA Active Data Guard 11g Best Practices paper (http://www.oracle.com/technetwork/database/features/availability/maa-wp-11gr1-activedataguard-1-128199.pdf)

Conclusion

The key question to answer when embarking on RMAN performance tuning is: "Where is my bottleneck?" Then, endeavor to eliminate it (or shift it to a different part of the system) to achieve the desired performance.

The backup phase should be analyzed to determine its maximum throughput.. For disk backups, the read phase throughput by itself should ideally consume most or all available disk I/O, when measured using `BACKUP VALIDATE.` In this way, RMAN buffers are continuously being filled and there is no wait in reading data from disk. RMAN multiplexing and memory buffer parameters can be used to tune this phase.

The copy phase should only be of concern if RMAN backup compression and/or encryption are used, since these operations are CPU-intensive. This is due to the higher-than-normal CPU consumption for these operations. Thus, the available CPU becomes the bottleneck for the backup processing. Adding CPUs, allocating more channels, and/or using higher performance compression settings are recommended for tuning the copy phase. If backup data only needs to be encrypted and not compressed, consider using TDE tablespace encryption – encrypted data will pass through RMAN unchanged and thus consume no additional CPU. Also consider whether one or both of the compression and encryption operations can be offloaded to the tape hardware itself.

Finally, if tuning read and copy phases do not sufficiently improve backup performance, the write phase should be investigated. For disk backups, ensure the backup storage throughput performs as expected without RMAN's involvement. Then, adjust the output buffer count and size, as needed.

For backups to tape, the ideal scenario is where the tape drives are performing at a significant percentage (i.e. 80% or better) of their rated speed. If they are not performing at this level, check with the media management vendor for tuning recommendations - this can involve verifying that buffer sizes, network connectivity, and media server hardware are performing as expected to support the RMAN backup data streams. The DBA should have a basic understanding of the media management environment, in order to properly convey RMAN backup performance diagnosis and findings to the storage administrator.

With that said, it is important to weigh all backup tuning changes against any potential impact to user workloads. With each change made, monitor the impact of the change to system resources, while concurrent workloads are running.

Remember - at the end of the day, what ultimately matters is not the individual backup environment components, but the end-to-end backup performance that can be achieved.

ORACLE®

Recovery Manager (RMAN)
Performance Tuning Best Practices
June 2011
Author: Timothy Chien
Contributing Authors: Steve Wertheimer

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