

# Optimizing Autodesk Topobase for Performance and Scalability

By turning to tested optimization techniques, organizations that use Autodesk® Topobase™ software to design and manage infrastructure assets can increase system responsiveness and serve more concurrent users.

## Introduction: Enhancing Topobase Performance and Scalability

Infrastructure asset management and design are mission-critical processes. Organizations that have invested in advanced asset management solutions, such as Autodesk Topobase software and the Oracle® Spatial database, want to derive optimal performance and scalability from their solutions to maximize value over the long term.

For Topobase, efficient configuration and maintenance of the various components of the system are crucial in helping ensure optimum performance and system capacity. To develop recommendations for enhancing Topobase performance and scalability, Autodesk formed the Topobase Performance Team (TPT). The TPT had three goals:

- **Analyze:** Test the performance of Topobase and its Oracle Spatial database in a variety of real-world situations
- **Improve:** Determine approaches for enhancing software and database configuration for typical Topobase environments
- **Document:** Share recommendations with the people who implement and depend on Topobase and Oracle Spatial

The TPT developed a set of best practices for enhancing Topobase performance, improving scalability, and determining system capacity needs. The team worked with a number of organizations that use Topobase to develop, test, and refine recommended practices in real-world environments, helping to ensure that recommendations would apply to diverse Topobase implementations.

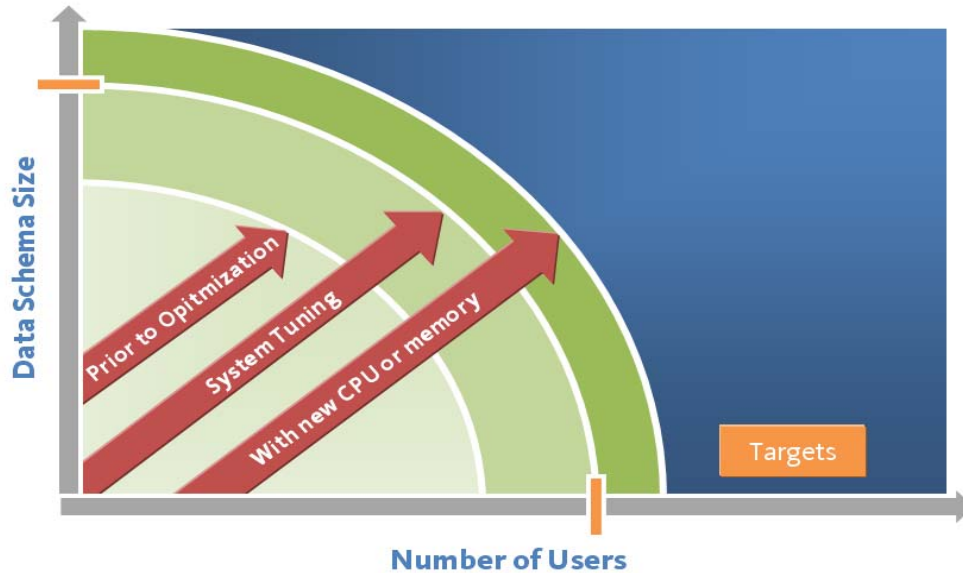


Figure 1: By optimizing Topobase, organizations can improve performance and scalability.

The TPT found that by tuning Topobase and Oracle, organizations can dramatically increase the number of concurrent users and improve system response time—often without adding memory or CPU capacity. They also developed recommendations to help organizations ascertain whether or not they need additional system capacity to meet or exceed performance and scalability goals.

Organizations using Topobase can refer to this paper to learn about tested approaches for maximizing the performance and scalability of their Topobase solution. Specifically, this paper:

- Presents a number of recommended options for configuring Topobase so that it generates graphics more efficiently
- Offers an examination of recommended techniques for configuring Oracle Spatial
- Outlines results organizations may expect from optimizing Topobase

## Performance Factors

As with any sophisticated technology, a number of factors impact the performance of the Topobase solution. Let's take a look at these factors as they apply to Topobase, Oracle Spatial, and Topobase solution users.

**Topobase** software performance is affected by the software module, the operations being performed, and the options selected by the administrator. Additionally, the memory and CPU speed of both the client and server-side machines impact system responsiveness.

**Oracle Database** performance is impacted by SQL statements—such as select, insert, update, and delete—executed against the database. Data schema size, configuration choices, and database hardware dictate how quickly the statements are executed. For example, a complex select statement will run more slowly—and affect concurrent users—in environments with a large data schema size or insufficient CPU capacity.

The number of **concurrent users** is the most vital parameter that impacts performance for organizations using Topobase. If the system has more concurrent users than the configuration and hardware can efficiently support, response times may slow to unacceptable levels.

### Meeting Real-world Demands: Testing Parameters

#### The Average Workload

A typical Topobase user digitizes up to 450 new features per day. Factoring in an average of 200 additional daily feature operations, such as updating the feature attributes or creating coordinate geometry (COGO) features, Topobase users execute as many as 650 feature operations per day. Assuming an eight hour day, that is approximately 1.4 features per minute.

#### A Robust Test Environment

To test Topobase under real-world conditions, the TPT simulated varying numbers of users working in parallel with the Topobase Client. Each test user digitized between 20 and 100 features every 20 to 60 seconds, which is 1.5 features per minute and user. Tests involved 25, 50, 100, and 200 users. Although specific tasks varied, the following workflow process was executed iteratively:

- Select an existing job or create a new job
- Generate graphic
- Digitize from 20 to 100 new point, line, and polygon features
- Set job to live
- Start network tracing

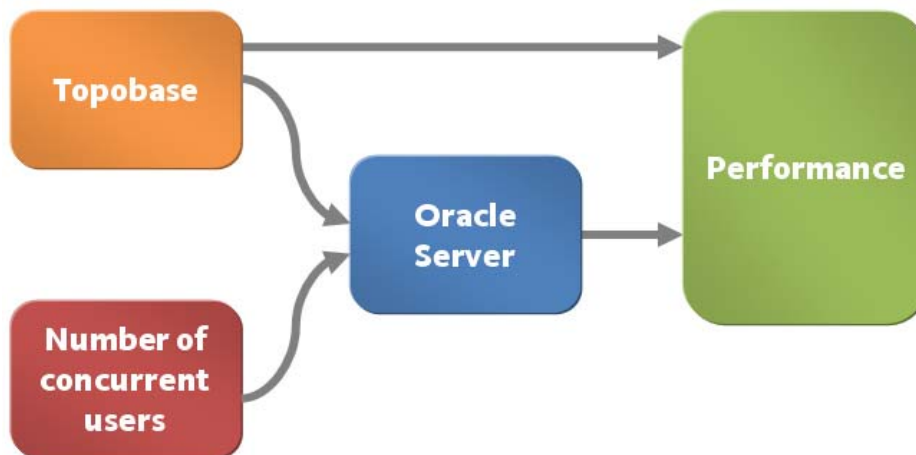


Figure 2: A number of factors impact the performance of Topobase.

By following the practices outlined in this paper, organizations can improve the performance of Topobase and Oracle Spatial while increasing the number of concurrent users they can support.

## Tuning Topobase

From a Topobase user's perspective, efficiency improvements in the generation and manipulation of graphics can have a significant impact. Actions taking place on the database are crucial to graphics generation speed, but there are a number of simple ways to accelerate graphics generation that do not involve database configuration.

To speed up graphic generation:

- Remove layers that are not important to the operation being performed. Graphic generation time is directly dependent on the number of layers displayed.
- Avoid unnecessary coordinate system transformation. Changing the coordinate system involves a number of complex calculations on the server side, which can slow graphics generation significantly.
- Use database views instead of client-side joins. Though essentially the same from a functional perspective, client-side joins take longer to generate than database joins, especially for joins consisting of more than two tables.
- Limit the number of columns in database views. Including unnecessary columns in views slows the system's ability to generate graphics efficiently.
- Use viewports that load the area of interest rather than the whole dataset when generating graphics.
- Optimize the display model. For example, use display rules to filter data by attribute. This helps ensure that the filtering is done on the client prior to rendering, which reduces the load on the server.

## Tuning Oracle

The TPT tested and refined a number of strategies for tuning Oracle Spatial configurations to support Topobase. While no two Oracle environments are exactly alike, organizations can adapt the TPT's recommendations to improve performance and scalability in a variety of circumstances. In addition to its tested recommendations, the TPT also suggests that organizations always apply the latest service packs and patches for Topobase and Oracle Spatial. The subsequent sections of this paper explore Oracle Spatial and the following recommendations in greater detail:

- **Use the Oracle Enterprise Manager to configure the Oracle database.** The Oracle Enterprise Manager streamlines a number of routine database management processes, making it easier for Topobase administrators without extensive Oracle database knowledge to maintain a high performance level.
- **Confirm that sufficient SGA memory is available.** Administrators can monitor and manage SGA memory using the Oracle Enterprise Manager. However, the TPT recommends using Automatic Shared Memory Management to manage SGA memory, especially for organizations with small-to-medium size datasets.

### General Oracle Server Recommendations

#### 1. Processors

Number of Concurrent Users	Recommended Number of CPU Cores
1 - 50	2
51 - 200	4
201 - 500	8 (to be verified)
501 - 1000	16 (to be verified)

#### 2. Memory

Calculate needed memory using the following formula:

Size of memory = size of tablespace + OS

#### 3. Use Oracle Real Application Clusters (RAC)

Oracle RAC provides a highly scalable and available solution for business applications.

- **Optimize execution plans based on system usage statistics.** By regularly gathering and applying statistics, administrators can keep the system tuned appropriately.
- **Configure Oracle to support appropriate numbers of sessions and open cursors.** The TPT developed a calculation that helps organizations determine the number of user sessions needed. For open cursors, the TPT recommends setting the maximum allowable to 300.
- **Maintain a buffer cache hit ratio of 95 percent.** Keeping the buffer cache hit ratio high improves performance significantly. Organizations without sufficient system capacity to maintain a high hit ratio should consider purchasing additional memory.
- **Place frequently used data into a keep pool.** By developing and maintaining an effective keep pool, organizations can realize a high hit ratio for important tables and indices. This helps improve the cache hit ratio in situations where additional memory is needed, but purchasing it is not an option. Only experienced Oracle database administrators (DBA) should implement this recommendation.
- **Find slow indices.** Slow indices are not able to quickly deliver information within their underlying tables. The TPT recommends identifying slow indices and reducing their clustering factors so that they can deliver data faster. Only experienced Oracle DBAs should implement this recommendation.

### **Overview of the Primary Components of Oracle**

Organizations running Topobase use Oracle Spatial to serve data to users. Oracle servers have two components: the Oracle database and the Oracle Instance. An Oracle Instance accesses the Oracle database and consists of memory and background process structures. Many of the configurations discussed in this white paper impact the Oracle Instance and are meant to enable the system's memory structure to support efficient access to information stored within the Oracle database.

Oracle's memory structure resides within two types of memory areas:

- **System Global Area (SGA):** This is memory that is shared by all server and background processes. SGA is allocated at instance startup from the system memory, and it is a fundamental component of each Oracle Instance. Much of the tuning covered in this paper concerns the management of and access to SGA data.
- **Program Global Area (PGA):** This is memory that is private to each server and background process; there is one PGA for each process. It is allocated when the server process is started.

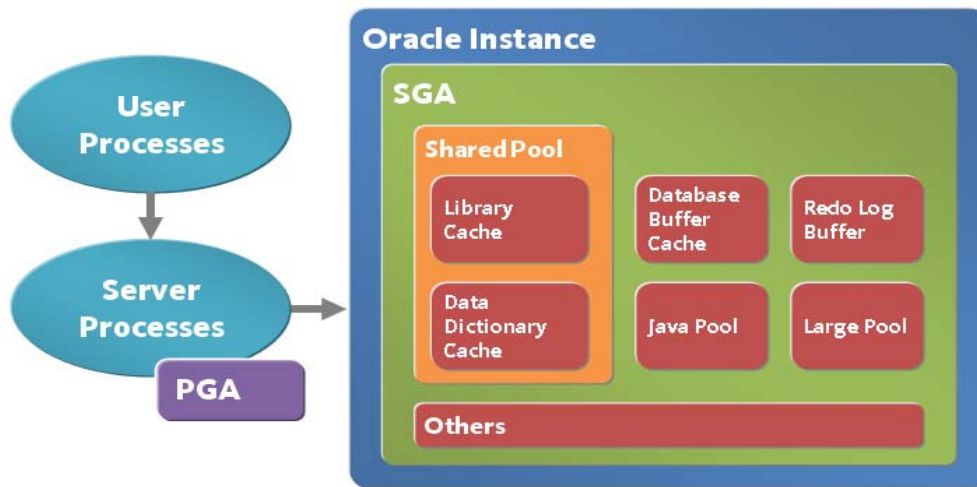


Figure 3: The Oracle Instance contains the SGA memory. Users connect to the Oracle Instance by way of the server processes that reside in the PGA.

### Work with the Oracle Enterprise Manager to Configure Oracle

The Oracle Enterprise Manager streamlines many aspects of Oracle database configuration and management. Although the same tasks can be executed using SQL statements, using the Oracle Enterprise Manager when possible is easier and faster for most functions. The Oracle Enterprise Manager is useful for managing a number of functions, including CPU load monitoring, memory allocation and usage, SQL statement statistics, and hard disk performance.

#### Quick Tip: Get to Know the Oracle Enterprise Manager

The Oracle Enterprise Manager contains significant useful functionality that is outside the scope of this paper. Learn more at <http://www.oracle.com/ent>

#### CPU Load Monitoring

To monitor CPU load, users can access the Top Activity main page within the Oracle Enterprise Manager (Figure 4). The chart at the top shows how the Oracle server is loaded over a selected time period. The page also gives an overview of which SQL statements had the highest impact on CPU load within the selected time frame. This can help Topobase administrators investigate the reasons behind periods of slow performance.

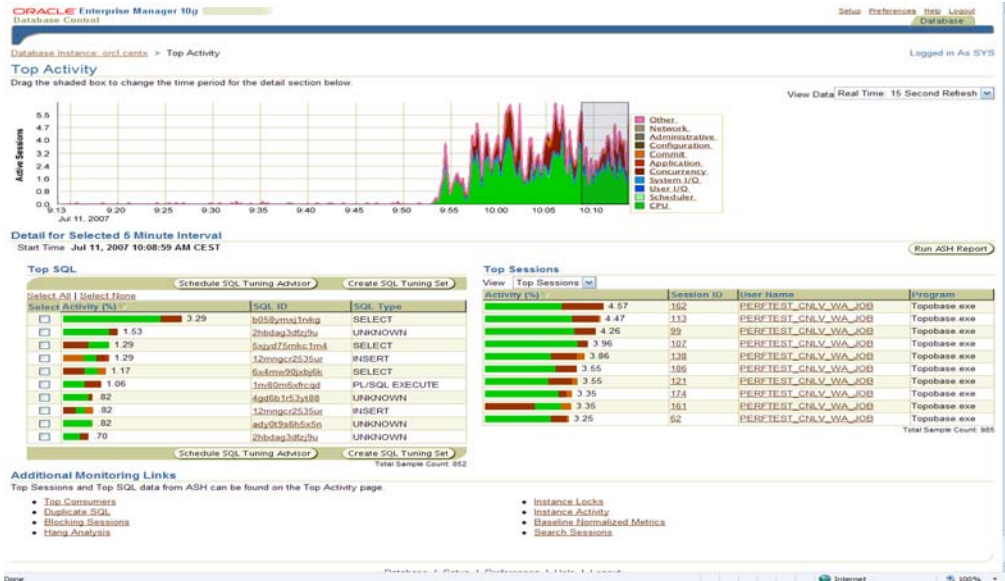


Figure 4: The Top Activity main page in the Oracle Enterprise Manager displays CPU load by activity.

### Memory Allocation

SGA memory must be sufficient for Topobase sessions to run efficiently. Using the Oracle Enterprise Manager, an organization can monitor and modify available SGA memory. The Memory Parameters interface within the Oracle Enterprise Manager provides a quick overview of how much memory is allotted to SGA. It also shows a history of SGA allocation over a given time period, which is useful for monitoring spikes in allocations across memory pools. Note that Automatic Shared Memory Management is **enabled** in the screenshot (Figure 5). A new feature within the Oracle 10g database, Automatic Shared Memory Management helps the system adapt automatically to varying workloads, eliminating the need to manually tune SGA parameters. Automatic Shared Memory Management is enabled by default, and the TPT recommends keeping it enabled.

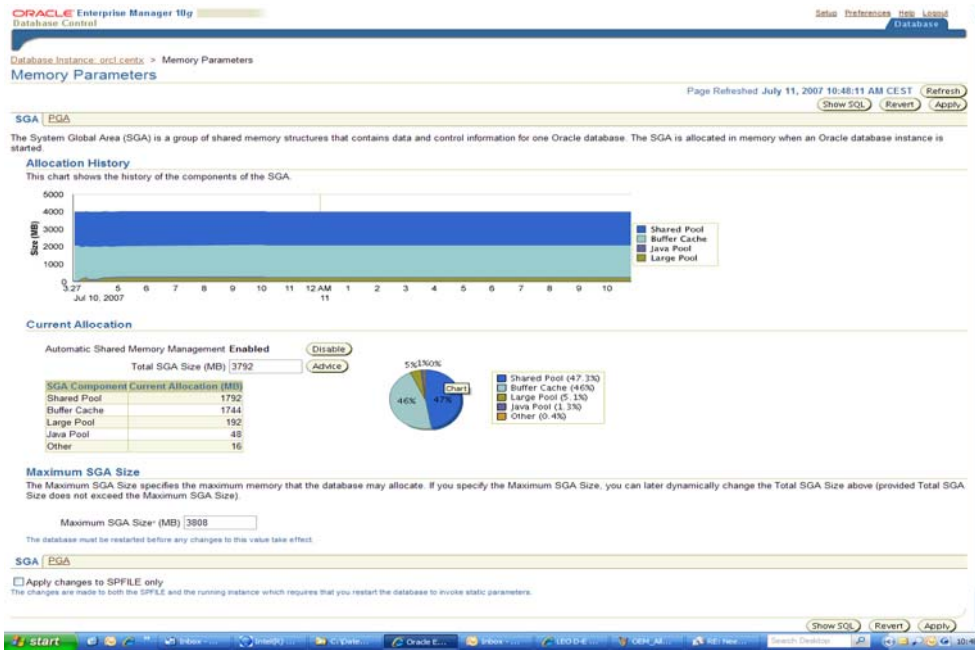


Figure 5: The Memory Parameters delivers SGA memory allotment information.

### Memory Usage

Organizations should monitor SGA usage, even when using Automatic Shared Memory Management. The SGA Pool Wastage screen in the Oracle Enterprise Manager shows the percentage of available SGA memory. If any of these values are near 0 percent, increase the allocated SGA memory to improve performance. SGA memory can be increased using the Memory Parameters interface. Organizations that do not have enough available memory to optimize SGA allocations should consider purchasing additional memory.

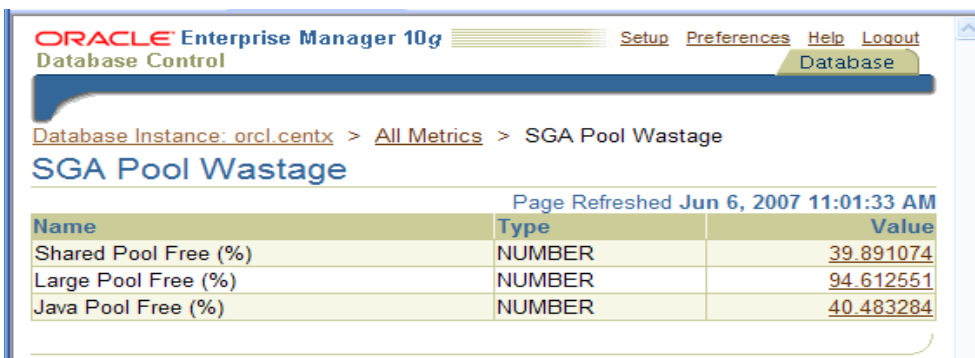


Figure 6: SGA Pool Wastage shows the amount of capacity available in SGA memory pools.

## SQL Statement Statistics

By selecting an individual SQL statement from the Top Activity main page, administrators can access the SQL Details page for detailed statistics about individual SQL statements (Figure 7). The SQL Details page shows statement activity as a percentage per session. It also allows administrators to schedule tasks recommended by SQL tuning advisor, which is on this page, to tune the execution of specific SQL statements and to check execution plan effectiveness. Optimizing execution plans is covered in the next section of this paper.

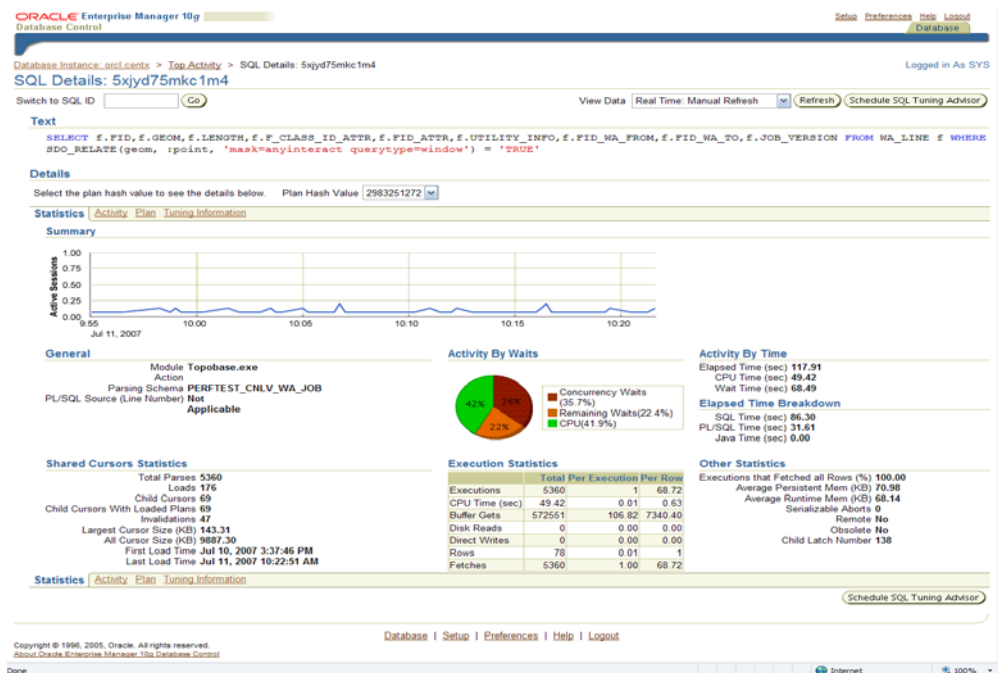


Figure 7: The SQL Details page shows execution parameters for selected SQL statements.

## Hard Disk Performance

By visiting the Database Files page in the Oracle Enterprise Manager, administrators can monitor the speed of read/write access to database files. Slow access may indicate that the hard disk needs to be defragmented or replaced. Alternatively, database files can be distributed on different disks. Some file systems, such as ext3 on the Linux<sup>®</sup> platform, allow access to the hard disk directly and asynchronously. If allowed, this technique may speed up the access time by a factor of 2. The setting that enables this type of access is called "filesystemio\_options".

ORACLE Enterprise Manager 10g Database Control [Setup](#) [Preferences](#) [Help](#) [Logout](#) [Database](#)

Database Instance: orcl.centx > [All Metrics](#) > Database Files

### Database Files

Page Refreshed Jul 11, 2007 10:39:44 AM

File Name	Average File Read Time (centi-seconds)	Average File Write Time (centi-seconds)
/usr/oracle/oradata/data/orcl/system01.dbf	1.200000	4.268293
/usr/oracle/oradata/data/orcl/undotbs01.dbf	2.000000	0.919492
/usr/oracle/oradata/data/orcl/sysaux01.dbf	0.403509	15.295154
/usr/oracle/oradata/data/orcl/performance-test	1.200000	14.994048
/usr/oracle/oradata/data/orcl/MO_FRIBOURG.dbf	1.000000	18.121951
/usr/oracle/oradata/data/orcl/PROVIDER_UNIT_TESTS.dbf	2.666667	2.333333
/usr/oracle/oradata/data/orcl/very_large	3.666667	2.000000
/usr/oracle/oradata/data/orcl/users01.dbf	1.500000	12.199667

[Database](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)  
 Copyright © 1996, 2005, Oracle. All rights reserved.  
[About Oracle Enterprise Manager 10g Database Control](#)

Figure 8: The Database Files page shows read/write time to files in the database.

### Gather Statistics to Optimize Execution Plans

By gathering statistics about Topobase usage, an organization can configure Oracle Spatial to better support routine processes and typical numbers of concurrent users. Users with administrative access to Topobase can gather statistics about the entire database schema from within the Topobase Administrator. Using these statistics, the Topobase Administrator module can generate optimal execution plans with the Optimize command. The TPT found that running the Optimize command whenever there are extensive changes to the database, such as data imports or major updates, helps to keep execution plans up-to-date.

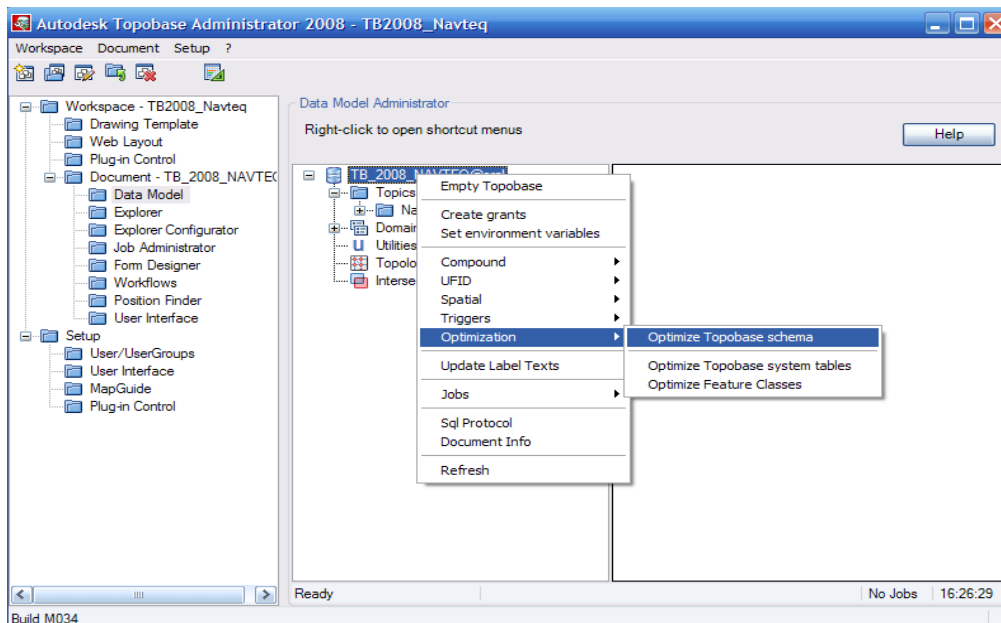


Figure 9: Run the Optimize command within the Autodesk Topobase Administrator after gathering relevant usage statistics.

## Configure Oracle to Support the Appropriate Number of Users

Oracle Spatial should be configured to support the maximum number of potential concurrent users within the organization. To do this, determine the maximum number of sessions required by calculating the number of connections users need to documents within the database. The TPT recommends calculating the number of connections using the following formula:

Oracle Connections =  $Concurrent\ Users \times (Topobase\ Documents \times 3 + 1)$   
Oracle Sessions =  $Oracle\ Connections \times 1.1$

For example, an organization with 200 users accessing an average of 2 Topobase documents simultaneously would need 1540 sessions:

Oracle Connections =  $200\ Concurrent\ Users \times (2\ Topobase\ Documents \times 3 + 1) = 1400$   
Oracle Sessions =  $1400\ Oracle\ Connections \times 1.1 = 1540$

When the number of sessions is determined, configure Oracle Spatial to support it. Set this value in the Oracle Enterprise Manager on the Initialization Parameters page (Figure 10) or use the following SQL statement:

```
ALTER SYSTEM SET sessions = 1540 SCOPE=SPFILE;
```

The screenshot shows the Oracle Enterprise Manager 10g Database Control interface. The page title is "Initialization Parameters" for the instance "orcl.centx". The "sessions" parameter is highlighted in the table below.

Select	Name	Help	Revisions	Value	Comments	Type	Basic	Dynamic	Category
<input type="radio"/>	java_max_sessionspace_size					Integer			Processes and Sessions
<input type="radio"/>	java_soft_sessionspace_limit					Integer			Processes and Sessions
<input checked="" type="radio"/>	sessions		1	1540		Integer	<input checked="" type="checkbox"/>		Processes and Sessions

Figure 10: Use the Initialization Parameters page to set sessions and open cursors.

## Set Maximum Open Cursors to at Least 300

In addition to configuring the system to support an appropriate number of users, administrators must also set the maximum number of open cursors. Database cursors are opened when SQL statements such as insert, update, delete, or select commands are executed against the database. If the maximum number of allowed open cursors is exceeded, users will receive the ora-1000 error message, which may also indicate an error in the

### Quick Tip: Reduce Network Latency

Use Oracle analysis tools to monitor and improve data throughput on the network at remote offices.

For more information, see [http://www.oracle.com/technology/deploy/availability/pdf/MAA\\_WP\\_10gR2\\_DataGuardNetworkBestPractices.pdf](http://www.oracle.com/technology/deploy/availability/pdf/MAA_WP_10gR2_DataGuardNetworkBestPractices.pdf)

application or in a Topobase plug-in.

The TPT found that it is best not to tune open cursors. Instead, set it high enough that it will not impact performance. If set higher than needed, there is no performance loss. In tests, the TPT did not encounter any errors with open cursors set to 300. Due to concerns about open cursors taking up too much memory in the shared pool, some administrators may be reluctant to set open cursors so high, but cursors are open on an as-needed basis. Every session will not utilize the maximum allowable number of cursors, but with a high number of cursors allowed, users will not encounter error messages during normal operations.

By selecting cursors as the parameter under the current tab, administrators can use the Oracle Enterprise Manager Initialization Parameters page (Figure 10 above) to view and set the maximum number of open cursors allowed.

### **Improve the Database Buffer Cache Hit Ratio**

If insufficient memory has been allocated to SGA, the database buffer cache hit ratio will be too low, slowing system response times significantly. Ideally, every time a user requests data, it will already reside in memory (i.e., the buffer cache) and be immediately available. For most databases, it is not practical to have enough memory to achieve that goal, but modest improvements in the cache hit ratio result in dramatic performance gains. For example, improving the cache hit ratio from 90 to 95 percent can nearly double system performance in some hardware environments. The TPT recommends a database buffer cache hit ratio of 95 percent or higher.

The following SQL statement provides the buffer cache hit ratio:

```
SELECT (1-(PHY.VALUE - LOB.VALUE - DIR.VALUE)/SES.VALUE)*100
       "CACHE HIT RATIO"
FROM   V$SYSSTAT SES, V$SYSSTAT LOB,
       V$SYSSTAT DIR, V$SYSSTAT PHY
WHERE  SES.NAME = 'session logical reads'
       and DIR.NAME = 'physical reads direct'
       and LOB.NAME = 'physical reads direct (lob)'
       and PHY.NAME = 'physical reads'
```

## **Oracle Tuning for Database Experts**

The two following optimization techniques are best executed by experienced Oracle DBAs. Successfully implementing these recommendations may require database knowledge that is outside the scope of this paper. Oracle beginners should not attempt to develop a keep pool or try to improve slow performing indices without the assistance of a database expert.

### **Develop an Effective Keep Pool**

If an organization cannot maintain a buffer cache hit ratio of 90 percent or above, it is likely that frequently used data is being left out of the buffer. The database must then turn to disk I/O to retrieve the data. Although this is a simplification, disk I/O can be thought of as the transfer of data to or from the hardware within the database. The process is slow, and virtually all Oracle tuning is intended to reduce disk I/O. Data that users access frequently can be manually added to the buffer by creating a keep

pool. The keep pool helps to keep selected data in the buffer, enabling a higher hit ratio for frequently used data. This reduces disk I/O when users access the data.

The following SQL statement places a table in the keep pool:

```
alter table <USERNAME>.<TABLE_NAME> storage (buffer_pool keep);
```

For example, the following statement puts the TB\_UFID table into the keep pool:

```
alter table <USERNAME>.TB_UFID storage (buffer_pool keep);
```

### Find Slow Performing Indices

Specific data indices or groups of indices within an Oracle database can slow system performance. This happens when the clustering factor is high. The clustering factor is the relationship of the index structure to the data table within it. Any indices with a clustering factor above 60 percent should be optimized. Strategies for lowering clustering factors include re-sequencing and using a hash cluster table or a single table cluster.

Use the following SQL statement to generate a table showing the performance of indices within Oracle:

```
SELECT substr(i.table_name,1,25) "Table",
       substr(i.index_name,1,30) "Index",
       substr(t.num_rows,1,10) "Rows",
       substr(i.clustering_factor,1,6) "Clust Factor",
       (round(100/t.num_rows*i.clustering_factor, 1)) "%"
FROM   user_indexes i, user_objects o, user_tables t
WHERE  (100/t.num_rows*i.clustering_factor) > 70
       and t.blocks > 0 and t.num_rows > 10
       and i.index_name = o.object_name
       and i.table_name = t.table_name
       and (t.table_name not like ('%_TBD'))
       and t.table_name not like ('%_TSER'))
       and i.clustering_factor is not null
ORDER BY "%" desc;
```

Table	Index	Rows	CluFct	%
TB_JOB_VERSION	TB_JOB_VERSION_IDX1	349525	334866	95.8
WA_NNOD	WA_NNOD_I	11677	10585	90.6
WA_NNOD	WA_NNOD_P	11677	10542	90.3
WA_NNOD	WA_NNOD_FID	11677	10540	90.3
TB_GN_MENU_ITEM	IDX_GN_MENU_ITEM_ID	406	362	89.2
TB_JOB	TB_JOB_NAME_U	465	375	80.6
TB_JOB	TB_JOB_P	465	372	80
WA_WELL	WA_WELL_P	5388	3780	70.2
WA_WELL	WA_WELL_FID	5388	3780	70.2

Figure 11: A table showing clustering factors for indices within Oracle.

## The Result: Improved Performance and Scalability

By tuning Topobase and Oracle Spatial, organizations can expect to see improved performance while serving more concurrent users. In the TPT's tests, participating organizations were able to more than double the number of concurrent users without adding memory or CPU capacity.

### Oracle Tuning

As the chart below shows, when Oracle is not tuned correctly, as few as 25 users can consume 70 percent of the CPU capacity. However, 50 concurrent users cannot be served effectively. With tuning, the system is able to perform well with approximately 90 users and to serve over 100 concurrent users with an acceptable degree of speed.

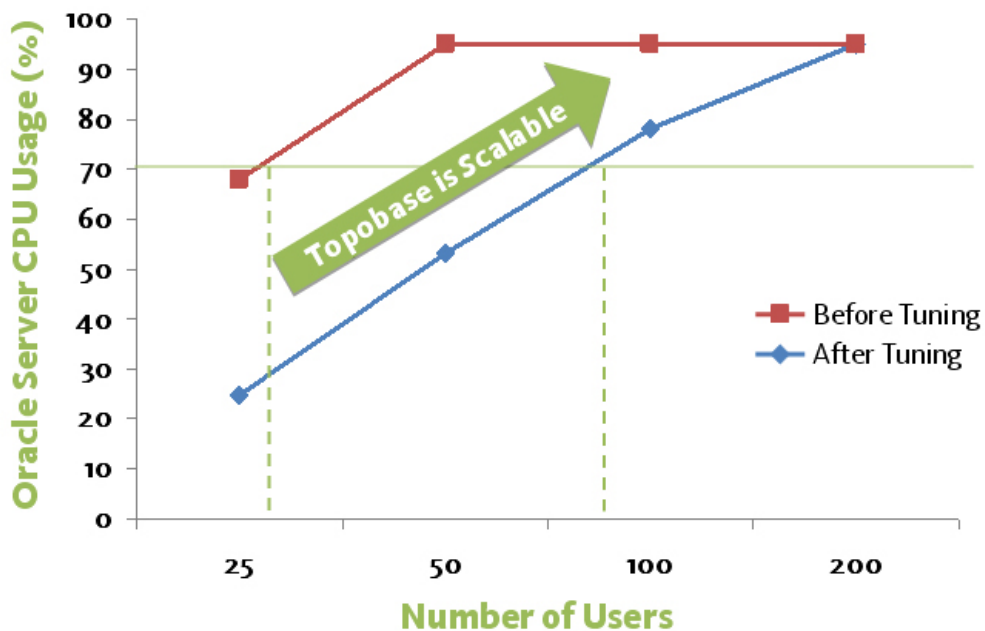


Figure 12: With tuning, Topobase is highly scalable.

### Oracle Server Hardware

For some organizations, the performance and scalability gains realized from system tuning will not be adequate to meet asset management needs. Tuning Oracle Spatial and Topobase is still advisable for these organizations because an optimized system will need less additional capacity to achieve their goals. The chart below shows the difference a small investment in hardware makes in a tuned system. The difference is so striking that even organizations meeting performance goals may want to consider the productivity gains likely to result from small server hardware investments.

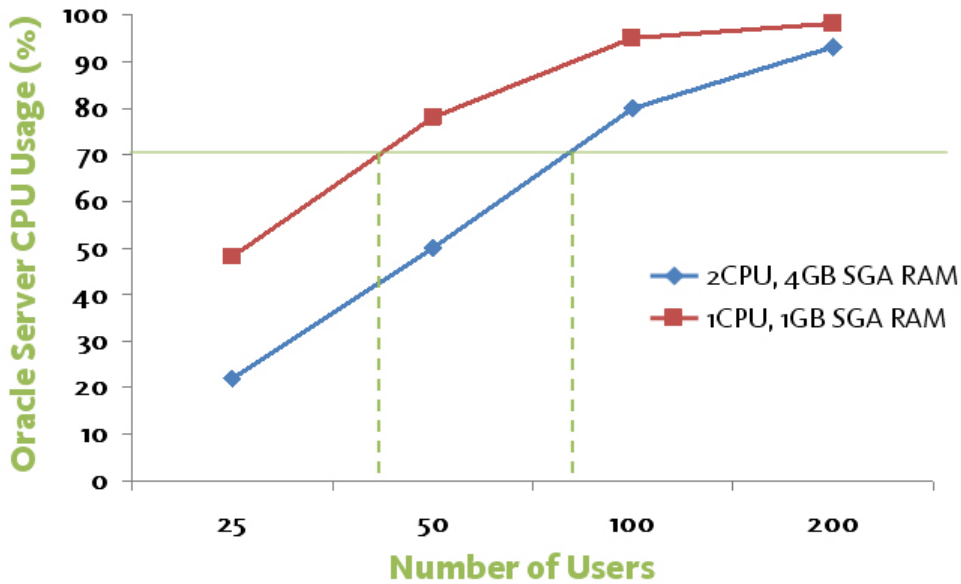


Figure 13: By making small investments in server hardware, organizations can double the number of concurrent Topobase users.

### Versioning

Many Topobase Administrators have asked how much of a burden the jobs functionality, also known as long transactions, places on the CPU. A job is a complex project or series of tasks that take place over time. When jobs functionality is enabled, Topobase allows users to keep modifications, a history, and multiple versions of a project separate from the base network information until the project is complete. While jobs functionality is not specifically a tuning issue, the TPT explored the question of how much jobs functionality impacts CPU load. The team found that jobs functionality increases CPU load by only 10 percent, even with large numbers of concurrent users.

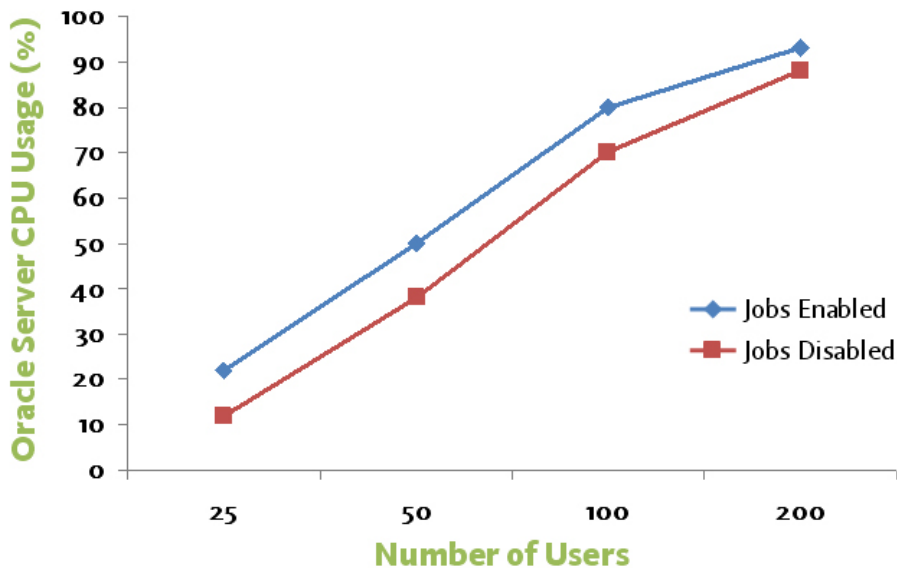


Figure 14: Enabling jobs functionality in Topobase increases server load by only 10 percent.

## Conclusion: Tune Topobase and Oracle and Improve Performance

Organizations can help keep their Topobase solution performing well and effectively serving concurrent users by implementing the TPT's system tuning recommendations. In summary, the TPT recommends that organizations:

- Configure Topobase so that it generates graphics efficiently
- Use the Oracle Enterprise Manager to monitor system performance
- Allocate sufficient memory to SGA pools
- Gather database statistics regularly to optimize execution plans
- Strive for a database cache hit ratio of 95 percent
- Add memory or CPU capacity to the system if necessary
- Place frequently accessed data in a keep pool (for experienced Oracle DBAs)
- Improve slow performing indices (for experienced Oracle DBAs)

By following these recommendations, organization can as much as double the number of concurrent users served by Topobase. Additionally, users will be able to perform routine functions faster.

### Learn more

For additional information about Autodesk Topobase software, visit the Topobase product center at: <http://www.autodesk.com/topobase>.

To learn more about enhancing the performance of Topobase, visit the Topobase Insiders Blog at: <http://topobaseinsiders.typepad.com>.

Additional information about tuning Oracle can be found at [www.oracle.com](http://www.oracle.com).

