

## 9(IT INFRASTRUCTURE MANAGEMENT WEEK 12 ASSIGNMENT

### 1.Elaborate about how infrastructure's performance and tuning process are assessed?

#### Performance Planning

Before starting on the instance or SQL tuning sections of this guide, make sure you have read Part II, "Performance Planning". Based on years of designing and performance experience, Oracle has designed a performance methodology. This brief section explains clear and simple activities that can dramatically improve system performance. It discusses the following topics:

- Understanding Investment Options
- Understanding Scalability
- System Architecture
- Application Design Principles
- Workload Testing, Modeling, and Implementation
- Deploying New Applications

#### Instance Tuning

Part III, "Optimizing Instance Performance" of this guide discusses the factors involved in the tuning and optimizing of an Oracle database instance.

When considering instance tuning, care must be taken in the initial design of the database system to avoid bottlenecks that could lead to performance problems. In addition, you need to consider:

Allocating memory to database structures

Determining I/O requirements of different parts of the database

Tuning the operating system for optimal performance of the database

After the database instance has been installed and configured, you need to monitor the database as it is running to check for performance-related problems.

#### Performance Principles

Performance tuning requires a different, although related, method to the initial configuration of a system. Configuring a system involves allocating resources in an ordered manner so that the initial system configuration is functional.

Tuning is driven by identifying the most significant bottleneck and making the appropriate changes to reduce or eliminate the effect of that bottleneck. Usually, tuning is performed reactively, either while the system is preproduction or after it is live.

#### Baselines

The most effective way to tune is to have an established performance baseline that can be used for comparison if a performance issue arises. Most database administrators (DBAs) know their system well and can easily identify peak usage periods. For example, the peak periods could be between 10.00am and 12.00pm and also between 1.30pm and 3.00pm. This could include a batch window of 12.00am midnight to 6am.

It is important to identify these peak periods at the site and install a monitoring tool that gathers performance data for those high-load times. Optimally, data gathering should be configured from when the application is in its initial trial phase during the QA cycle. Otherwise, this should be configured when the system is first in production.

Ideally, baseline data gathered should include the following:

- Application statistics (transaction volumes, response time)
- Database statistics
- Operating system statistics
- Disk I/O statistics
- Network statistics

In the Automatic Workload Repository, baselines are identified by a range of snapshots that are preserved for future comparisons. See "Overview of the Automatic Workload Repository".

### **The Symptoms and the Problems**

A common pitfall in performance tuning is to mistake the symptoms of a problem for the actual problem itself. It is important to recognize that many performance statistics indicate the symptoms, and that identifying the symptom is not sufficient data to implement a remedy. For example:

#### **Slow physical I/O**

Generally, this is caused by poorly-configured disks. However, it could also be caused by a significant amount of unnecessary physical I/O on those disks issued by poorly-tuned SQL.

#### **Latch contention**

Rarely is latch contention tunable by reconfiguring the instance. Rather, latch contention usually is resolved through application changes.

#### **Excessive CPU usage**

Excessive CPU usage usually means that there is little idle CPU on the system. This could be caused by an inadequately-sized system, by untuned SQL statements, or by inefficient application programs.