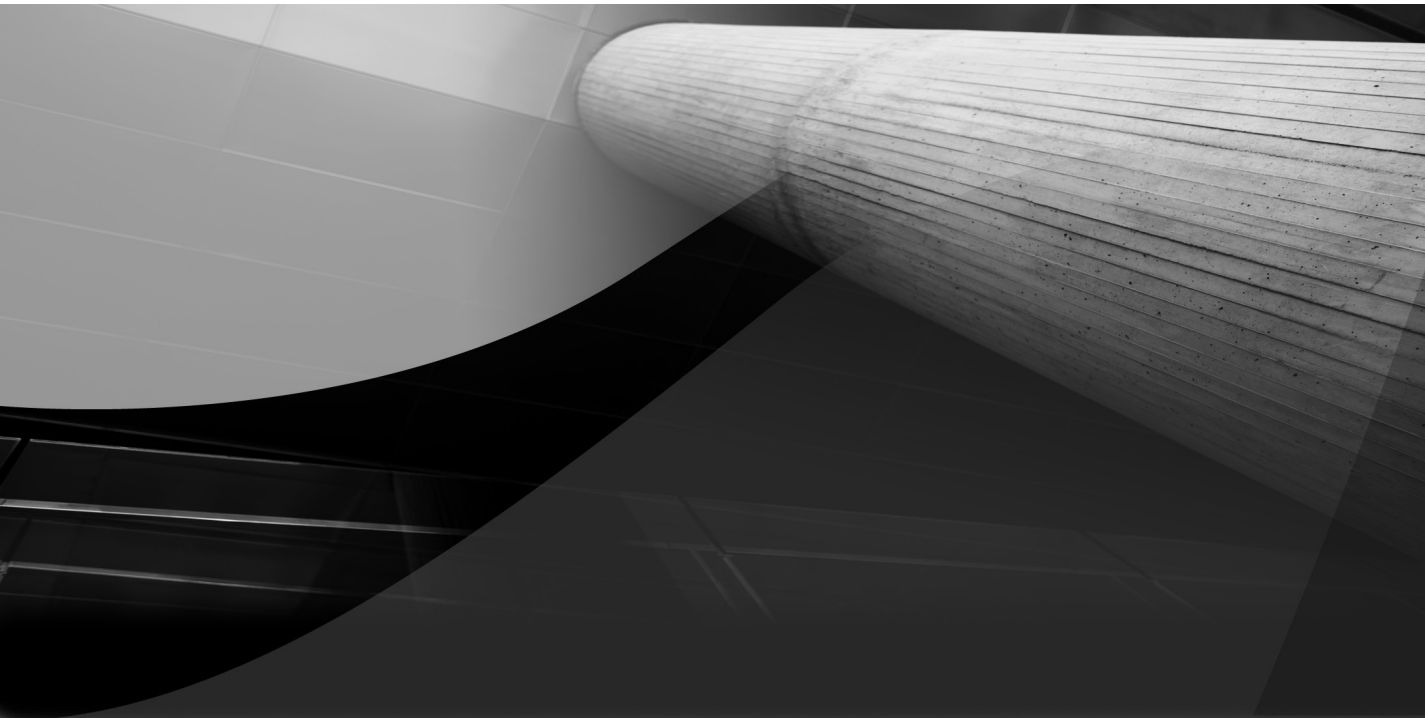


PART I

Installing and Deploying Grid Control



CHAPTER 1

What Is Grid Control?
Enterprise Manager
Concepts



know it will sound like a cliché, but you are at the beginning of an important journey to help guide you through planning, deploying, and using a product that has an impact on your entire IT organization. And as with all other big undertakings, planning is a critical phase.

This chapter is important because it deals with and explains the very basics of the software called Enterprise Manager. I hope that, in reading this chapter, the one notion you will gather is that Enterprise Manager is more than just some application reporting data from a bunch of machines.

There are tips and tricks throughout this book to help you use this product more effectively. But the most important thing I hope people will learn from this book is that Enterprise Manager is indeed focused on the Enterprise, and can have a positive impact on the entire enterprise, in all of its aspects, and all of its diverse branches.

Enterprise Management Software

In 2003, Oracle took an unprecedented direction over the landscape of IT and proclaimed—with uncanny foresight—that its software, most notably Enterprise Manager, would be used to address key challenges that many companies would soon come to face. By that time, the dust over Internet technologies had settled, and IT professionals began to find themselves struggling for an end-to-end management solution, as the concept of *computing as a commodity* began to catch on.

To address the challenges of companies whose business needs change faster than their IT departments can adapt, Oracle unveiled a suite of business software: Oracle Application Server 10g, Oracle Database 10g, and Oracle Enterprise Manager 10g Grid Control. The “g” in 10g made an explicit call to address *grid computing*, a deployment topology that integrates all IT resources—storage, servers, databases, application servers, and network peripherals—to provide database services to applications on demand.

Oracle technology is typically arranged as a multitier architecture (commonly referred to as *n-tier*), in which the presentation, the application processing, and the data management are logically separate processes. It is worth noting that the concepts of *layer* and *tier* are often used interchangeably, when in fact there is one fairly common distinction between the two: a *layer* represents a logical structuring mechanism for the elements that make up a software solution, while a *tier* represents the physical structuring mechanism that make up a system infrastructure. With that in mind, a typical deployment using Oracle technology would be divided as follows:

- The *presentation layer*, represented as Tier-1, is the entry point for all application client connections seeking data. These requests originate from client machines or handheld devices, and communicate with the other layers by rendering the results from these requests.
- The *application-processing layer*, represented as Tier-2, consists of Java Enterprise Edition (J2EE) application servers and HTTP web servers, which Oracle provides in Oracle Application Server 10g. This tier is considered the “glue” to the other tiers and controls most of the application’s functionality by performing detailed processing. If necessary, the communication to the database back end happens by means of various protocols and specifications such as Java Database Connectivity (JDBC) and Oracle Database Connectivity (ODBC).

- The *data management layer*, represented as Tier-3, consists of databases and database storage subsystems. In the grid, databases are utilizing Oracle’s Real Application Clusters (RAC) technology, which allows databases to run packages and custom applications unchanged across a set of clustered servers. The storage subsystem is usually a collection of low-cost disk devices that, when utilized against a solution such as Oracle’s Automatic Storage Management (ASM), can easily partition and distribute data storage throughout the disk array.

Enterprise Manager 10g Grid Control weaves itself into the mix by effectively monitoring these tiers, and is even architected in the same manner; that is, components of Grid Control are built on Oracle technology that is even capable of monitoring itself. Because of this, no company needs to be fully immersed in grid technology to take advantage of what Grid Control offers. This is a cursory overview of the subject and is by no means complete, but it does place Enterprise Manager Grid Control in the correct context—this is the solution to gravitate toward if you seek to effectively monitor and leverage assets of a data center.

The monitoring of these data centers can be done from locations all around the world, making this suite of products a truly enterprise-class application. As Figure 1-1 illustrates, Grid Control can be used on local company networks, on subnets protected by firewalls, and even with targets on the Internet (outside the company network).

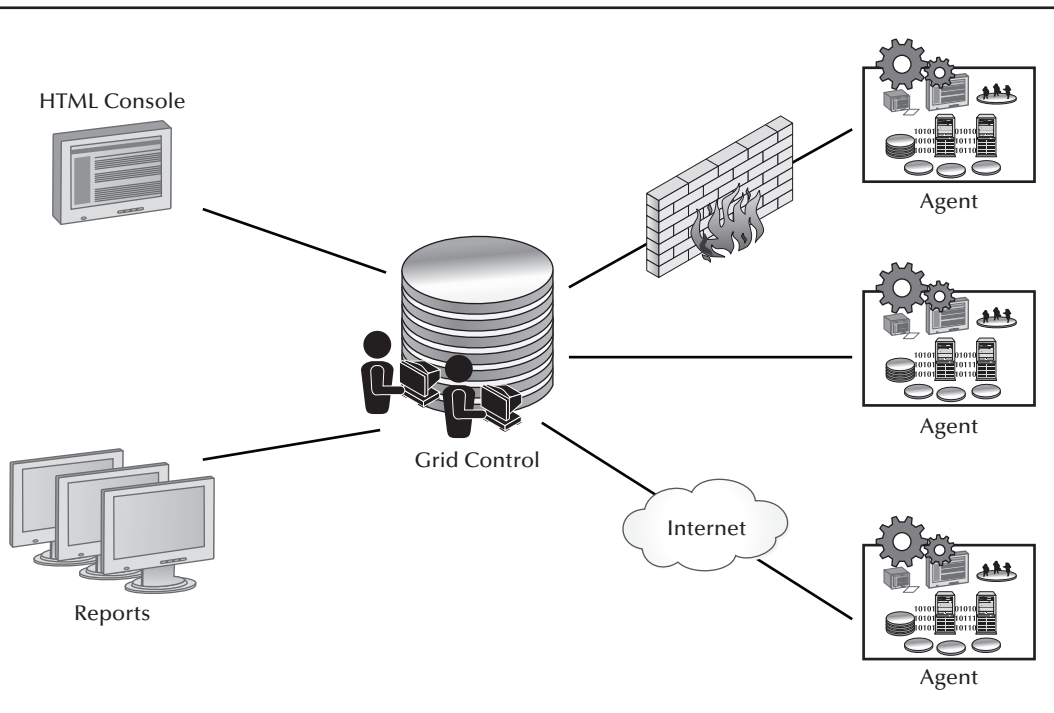


FIGURE 1-1. *The Grid Control architecture*

Administrators, who do not have to be in the same location as the management infrastructure, can use the information provided in Grid Control to manage and maintain their applications. With lights-out management capabilities, they can be alerted about outages and potential problems, giving Grid Control both proactive and reactive monitoring capabilities. Out-of-box, Grid Control supports monitoring of over 200 different types of targets and applications. On top of that, the infrastructure can be extended with additional monitoring plug-ins to support those types of business-critical targets that Grid Control is not aware of by default. This might sound like a lot, or even overkill. But the variety is required to even be considered a manageability solution in today's diverse IT environments. The days when a data center was nothing more than a few database machines put together in a single location are long gone.

Recall if you can, the days when you required specialized resources to maintain every implementation, and significant overhead was needed simply to make things work. The advent of Grid Control provides tremendous relief in this area, but requires a disciplined approach in changing the way we design and think about managing assets of a data center. Modern data centers have evolved into hubs of diverse IT infrastructure, ranging from simple staging machines to complex multimachine application setups and network devices needed to run today's enterprise-wide (and sometimes even world-wide) software.

The Various Management Tools

To respond to the rapid changes in a data center, a set of tools was introduced to manage the various components that make up a data center. Each of these tools manages and controls a set of Oracle products. They all follow the same multitier approach, with an Agent monitoring and gathering information, an application layer to collect and process the data, and a presentation layer to render the information to the administrator. While Grid Control is used to manage an enterprise and a variety of different things within that enterprise, there are other control tools managing some of the Oracle products. Two of those control tools, Database Control (DB Control) and Application Server Control (AS Control), are closely related to Grid Control, and often cause confusion in terms of setting them up and using them in conjunction with Grid Control.

DB Control

DB Control is a scaled-down and trimmed-down version of Grid Control, intended to manage a single database, or a single RAC installation. It contains a SYSMAN schema in the SYSAUX tablespace, and is monitored by an Agent designed specifically for that version of the database. Both the SYSMAN schema and the DB Control Agent have limited capabilities to monitor just that single installation. The software to run DB Control will be present in the *ORACLE_HOME* of the database. No additional software is needed to get the DB Control application—which includes the DB Control Agent—up and running.

If Grid Control is used to manage the enterprise, it replaces DB Control. The SYSMAN schema can then be removed from the target database and the DB Control Agent needs to be shut down, since it will no longer be used. The Grid Control Agent, installed on the same machine but outside of the database home, will then assume all monitoring and administration tasks, and report the data to the central Grid Control repository.

AS Control

AS Control is the main tool used to administer Oracle Application Servers. Like DB Control, AS Control consists of two parts: The main application, AS Control, and the AS Control Agent perform the basic monitoring and information gathering of each Oracle Application Server installation. No database schema is installed or used. This means that AS Control uses only real-time information, no historical information is kept, and no trending is done for any of the metric data. AS Control, like DB Control, is also installed together with the OAS software in the same *ORACLE_HOME*, requiring no further installation or configuration to get the tool up and running after the installation of the Application Server.

If Grid Control is used to monitor the Application Server, the Grid Control Agent will collect all the metrics directly from the Application Server components (bypassing the AS Control Agent), and upload that data to the central repository. The AS Control application is still needed to perform the main administration tasks like reconfiguring the Application Server, or deploying new applications.

Grid Control Concepts

Before we start diving deeper into the Grid Control application and what it can do, let's first review some basic concepts and explain some of the basic terminology used in the Enterprise Manager world.

Grid Control vs. Enterprise Manager

Take any piece of documentation, or any document or white paper written about Enterprise Manager, and you will see that the terms "Enterprise Manager" and "Grid Control" will be referenced multiple times. Besides the full term, there are also acronyms associated with each one: EM (Enterprise Manager) and GC (Grid Control). Historically speaking, the set of tools used to manage "things" (called "targets" in Grid Control—speak; see section "Metrics vs. Targets" later in this chapter) was called Enterprise Manager. Over the years the product has evolved and expanded into a suite of applications, with more than just the main management application. Hence the term Enterprise Manager nowadays points to the total infrastructure, with the entire suite of applications used to monitor and manage. Grid Control, on the other hand, is a relatively new term. It was first used as the management solution for the Grid Computing infrastructure. It refers to the main web-based management application, and the various components it needs to monitor and manage.

Going forward, the two will diverge even more: Enterprise Manager is the complete suite of applications used to manage and administer the enterprise, ranging from basic monitoring, top-down diagnostics, and performance monitoring of all the applications running in the enterprise, to rolling out and provisioning standardized environments defined as the corporate environment. It also includes tools like Oracle Application Diagnostics for Java (AD4J), Oracle Real User Experience Insight (RUEI), and Change Management Console, all add-ons installed on top of the base product. Grid Control, on the other hand, is still the same set of basic monitoring tools an Enterprise Administrator will get after installing the base software, designed to monitor and administer the enterprise.



NOTE

For the history buffs, a little side note: This web-based application was, in a previous incarnation, the Java application called Enterprise Manager. When the switch to Grid Computing and the web-based approach happened, the application was renamed to Grid Control, and the suite of tools and application kept the name Enterprise Manager.

In this book, we will focus primarily on the main application, and the IT infrastructure needed to use and maintain the product. As such, in the context of this book the terms Grid Control and Enterprise Manager mean and point to the same thing.

Acronym vs. Idiom

Every component in the Grid Control has its own acronyms and its own idioms:

- The OMA (Oracle Management Agent) is the Grid Control Agent. This is usually just referred to as the *Agent*.
- The OMS (Oracle Management Server) is the middle tier of infrastructure and is typically named the *Management Server*.
- The OMR (Oracle Management Repository) is the database used to store all the management data. Strictly speaking, the OMR only points to the schema used in the database that holds all the data (the SYSMAN schema), although the term “repository” is used several times to specify both the schema and the database the schema resides in.
- The last one in the list lacks a proper management acronym. The interface the users and administrators are using has several names: Console, UI, Browser. On the up side, each of these terms is descriptive enough to explain what it is pointing to.

For the sake of completeness, here are a few of the Application Server acronyms frequently used in documents and white papers:

- AS (Application Server) or iAS (Internet Application Server—the older term) points to an installation of the Oracle Application Server software.
- OC4J (Oracle Containers for Java) is a container that contains an entire application. It is deployed in an Application Server infrastructure. For Grid Control, there are two OC4J containers deployed: OC4J_EM (the main OMS application) and OC4J_EMPROV (Agent provisioning application). In the next release of Grid Control, Oracle has acquired BEA and plans to replace OC4J with JRockit as the new Java container.
- OHS is the Oracle HTTP Server, an OHS-based server used by OAS.

Metrics vs. Targets

Each enterprise asset is a *managed entity* in Grid Control (called a *target*) and provides the following information:

- State of this target (is it available for use?)
- Performance, resource, and usage indicators (how responsive is it?)
- Health statistics (is it working properly?)
- Configuration data (how is it configured?)

One should consider a target to be the management of just one thing—for instance, the DB Console is used to manage that one database, but it cannot also manage the environment in which that database exists. The ability to manage the combination of both is what is now commonly referred to as a *managed target*. When the monitoring capabilities expand to all technology elements required to keeping a business up and operational, then we consider that to be the management of a grid environment. The consumption of all such technology elements is what makes up a working ecosystem!

Each piece of information and each data point a target provides is modeled as a metric. A metric is computed through a command—or series of commands—that is executed on a regular basis to obtain data about the specified target. These commands can be specified as shell scripts, Perl scripts, SQL statements (or a PL/SQL block), SNMP requests, or even as Java commands. The metrics are combined into logical units called *collections* by the Agent. In most cases, a collection contains just one metric, but there are cases where several metrics are combined together to form one logical unit. An example of this is the storage information collected for a database: The storage collection consists of several metrics, gathering the tablespace and segment information for the database. All these collections are uploaded to a central location, where the information will be analyzed, to be used for trending and historical comparisons.

Metric data points can also have threshold conditions specified, to signal an abnormal condition of that metric on the target. These alerts, called *violations*, can then be used to notify the owner responsible for the target to take the appropriate action or artificially programmed to automatically address the issue.

The focus of Grid Control is to provide a centralized overview of all known targets and their metric information, to give the administrators managing those targets access to the current (and historical) state, and to interact with the targets for either regular maintenance, or to react to a problem condition found.

The two main parts in the management of targets are:

- **Monitoring** Proactively gathering state information, raising alerts if needed, and uploading the state information to a central location. This data can then be used to do historical trending and compare information between targets at various points in time.
- **Administration** Interaction with the targets via a centralized user interface to perform day-to-day operations. These tasks include regular maintenance and cleanup, as well as patching and upgrading targets.

When people log in to the Grid Control UI for the first time, the amount of data that is available usually overwhelms them. And it is true that there is a lot of information available. This is where this book can be a big help, to guide you through the initial installs, making available the things that you absolutely need, and giving suggestions on how to handle the continual stream of information.

Administrator vs. User

For any deployed application, an administrator maintains and monitors the application, while a user merely uses the product to do his or her daily work. And it's the same for Grid Control, although people tend to make the mistake of making users administrators. A Grid Control Administrator is someone who manages the (Enterprise Manager) infrastructure. These people typically do not manage the individual databases or application servers, or any other target discovered in Grid Control. A Grid Control User is a database administrator, a system operator,

a network administrator, or an application DBA; in short, someone who uses Grid Control to manage the targets he or she is responsible for. Managing targets in Grid Control does not necessarily mean managing Enterprise Manager itself (meaning the infrastructure and the various components of the Grid Control application).

To make things easier to relate to, users in terms of Grid Control should be referenced as Enterprise Users or Enterprise Administrators:

- An Enterprise User is a mere user of Grid Control, with no special privileges or requirements to manage the Enterprise Manager infrastructure.

These users only need access and privileges to manage and maintain the targets they are responsible for.

- An Enterprise Administrator is someone who manages and maintains Enterprise Manager, and will therefore need elevated privileges to be able to access the infrastructure components, and responds to alerts triggered by these targets.
- Every Grid Control administrator is by definition also a user of the application (but not every user is an administrator!). Some of these administrators will be super-users, with the necessary privileges to make changes to the infrastructure.

Grid Control Components

In order to provide a scalable architecture, which can grow and keep up with the growth of the IT environment in which it is installed, Grid Control has been split up into three different tiers, each with its own specific responsibilities, each performing a specific task. The high-level breakdown of these tiers is shown in Figure 1-2. The arrows in the figure describe the main communication flows between the various tiers in the architecture. The direction of the arrows is also important: The communication between the various components is strictly regulated, with only the central tier handling all the information requests, and dispatching the information as needed.

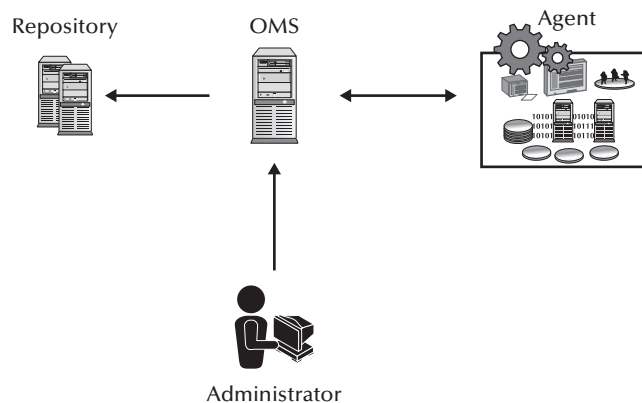


FIGURE 1-2. *The three tiers of Grid Control*

Grid Control Console

Resting on the presentation layer, the Grid Control Console is the entry point to the Grid Control product and is commonly accessed from any web-based browser, making it a lightweight means of easily managing the entire environment from any location. With no additional setup, you can also extend subsets of Enterprise Manager to be managed by handheld devices using functionality provided by Enterprise Manager's EM2Go. The URL to access the Grid Control Console is `http(s)://<OMS-FQDN>:<PORT>/em`. By default, the communication between the browser and the Console is secured using SSL (HTTPS). The URL port differs depending on whether you are using Webcache or accessing the Oracle HTTP Server directly and on the operating system, Linux or Windows, that is hosting the Oracle Management Server. In addition to the web console, there are several optional client components that work with Grid Control:

- **EM Command Line Interface** The EM CLI is for administrators who want to access the functionality of Grid Control, but use shell scripting or command-line interactivity in place of the Web Console.
- **Oracle Configuration Manager Client (OCM)** Previously known as the Remote Diagnostic Agent, OCM is a tool to collect host configuration information to assist Oracle Support Services in performing diagnostics and root-cause analysis of your installations. This is a standalone utility that can be downloaded from My Oracle Support. It is also bundled with many Oracle products, and is installed and configured during the installation of the Oracle software.

The Grid Control Agent

It all starts with the Agent. This is the worker bee of the infrastructure. Without Agents, there is no data, and an administrator or user will have nothing to work with. Every managed server machine will have an Agent installed, which gathers and uploads information to the middle tier about the state and condition of all the targets it is responsible for running on this machine.

An Agent by itself will not do anything, or report anything. Only after it has been told to start monitoring a target, or when an OMS instructs it to execute a command on behalf of one of the targets, will the Agent start performing management tasks.

As soon as the install of the Grid Control Agent completes, there are at least two entities defined that the Agent will use as managed targets:

- The Agent itself, to report diagnostics information about itself
- The host the Agent was installed on
- A third entity (a target of the type "cluster") will be added if the Agent was installed on a cluster machine, and an Oracle CRS installation was found on the box.

These targets are the essential and mandatory minimum ones needed for the monitoring. Additional targets can be added to the Agent in three ways:

- **Auto-discovery** After the install has added the Agent and Host targets, it will look for other common target types on the box. If any targets are found, they will be added to the list of managed targets.

The Agent will automatically look for these targets:

- Oracle Databases, RAC, and database listeners
- Oracle Application servers, and all Application Server components, such as HTTP Servers, Webcache servers, deployed OC4J applications, and so on.
- **A user requesting a rediscovery** When a user requests to add a target on an Agent in the Console (either via the Agent home page, or via the All Targets page), the Agent will rerun the discovery scripts of the standard types. If this rediscovery finds any new ones, these targets will be presented in the Console, so the user can decide to keep or ignore them.
- **Manual discovery of a target** If a standard target is not automatically discovered, or for those targets that do not have a discovery script, the user has the ability to manually add that target to the Agent via the Console.

When a target is discovered and visible in the Console, the Agent will monitor the target by providing useful data about it. At the same time, the Agent will accept requests from the Management Server and Grid Control users to execute administration tasks against the targets. For each target the Agent is monitoring, it has a set of collection data (the metrics and data points describing the state and condition of a target, as described earlier) that it will capture on a regular basis. These out-of-box collections are stored in the default collection files, one file for each type of target the Agent knows how to monitor.

By using these default collection files, the Agent will automatically start monitoring a specific set of metrics as soon as a target of that type is discovered. Each of the data points (called a *metric*) the Agent collects has the following characteristics:

- **A collection frequency** The agent will use this to schedule the metric on a repeating basis.
- **A metric category**
 - **Normal data metric** This is the most common type. The data is uploaded to the repository, where it gets rolled up, so the history of this metric can be shown in the Console. This rolled-up information can then be used for trending. Examples:
 - CPU utilization of a host
 - Tablespace usage of a database
 - **Configuration metric** These metrics are uploaded, but they are not rolled up, and they are not used for trending purposes. Differences between the current and previous values are kept, however, so changes to the configuration of the target can be tracked over time. Examples:
 - List of *ORACLE_HOME*s installed on a machine
 - List of network interfaces present on a host
 - List of OS patches installed on a host
 - **Real-time metric** These metrics are defined at the Agent, but without a formal schedule. They are run on demand, when a user specifically requests the data for this metric. They are also not uploaded to the repository, and no historical record is kept of the resulting values.

- **An optional condition** Only data metrics can have conditions. Configuration and real-time metrics do not have conditions. The condition is defined as a comparison between the data returned by the metric and the warning and critical thresholds defined in the condition. If the metric data point violates any of these threshold values, the Agent will generate an alert and send it to the OMS. This alert is sometimes also referred to as the *state* of the metric. Once this alert is uploaded to the repository, it can then be used to notify an user to take action.

Besides the metrics the Agent collects on a regular basis, the OMS can send information to run a specific command in the context of a target. This is the essence of a Grid Control Job: a command, or set of commands, sent to the Agent to be run at the appointed time in the context of a specific target.

The Management Server (OMS)

The Management Server (OMS) is the core of the infrastructure. It is the tier (and the only one) that communicates with all the other components. It rests in the application-processing layer and is installed as a standard Oracle Application Server, with two OC4J applications deployed (OC4J_EM, the main OMS server application, and OC4J_EMPROV to do Grid Control Agent provisioning). For deployments using version 10.2.0.5 or higher of Grid Control, a third OC4J application will be deployed: the OCMRepeater. This application is not part of the Management Server. It only serves to upload the OCM configuration data to Oracle.

The management server acts as an information broker in the infrastructure: It accepts all information the Agents are uploading, and stores it in the repository. When a user requests info about a specific target, the OMS will retrieve this information from the repository and report it back to the user. The OMS internally is broken up into a set of modules, each responsible for a specific aspect of the monitoring:

- **The Console** Responsible for retrieving and displaying all the information a user has requested in the browser.
- **The Job subsystem** Scheduling and dispatching of tasks and jobs defined either by the system itself, or by a user.
- **The Loader subsystem** All data uploaded by the Agents is parsed and inserted into the repository.
- **The Notification subsystem** Users and administrators need to be alerted of critical conditions happening in the environment. Once the data is loaded into the repository, and the triggered alert is detected by the notification system, all affected users and administrators will receive a notification of the violation.
- **Self-monitoring** This is often referred to as the MTM subsystem (Monitor The Monitor). This is a set of routines and automated tasks, which keep track of how the infrastructure itself is operating. If needed, alerts can be generated by the metrics the routines are collecting to alert an administrator of a potential problem.

The Grid Control Repository

The data management tier houses the data store in which all collected information from the managed targets resides. The size and resources used by this database will grow as the number of managed

targets in your enterprise grows. To be able to handle growth, any Oracle Enterprise Edition Database can be used as the repository: The initial install can be done using a single-instance database, which later on is converted into a multinode RAC database, with a Data Guard instance in a remote location for disaster recovery. The database has to be an Enterprise Edition database, with the partitioning and object options enabled: These two options are needed for the repository schema.

The Enterprise Manager data is stored in a single schema (the SYSMAN user). This user controls and handles all management data. Besides acting as the central data store for all data, the repository will also perform some tasks that need to be done centrally, and not per Management Server. This includes housekeeping tasks, like data rollup, purging of the old data, and the old partitions. These tasks are performed via DBMS_JOBS, and can function and execute without the intervention of OMS servers.

The Flow of Information

We'll start with a basic overview of the main data flows of an Enterprise Manager infrastructure. More details for each of these flows will follow in the following chapters. But in order to make the right decisions when planning the rollout of Grid Control, understanding who talks to whom, and in what direction the data flows, is essential to make the right decisions on where to place and install which component.

Agent and OMS Communication

The communication between the Agent and OMS always uses HTTPS (SSL-encrypted HTTP traffic). Although it is technically possible to unsecure the Agent and have just normal HTTP communication (without the SSL encryption), the out-of-box default will always be secure.

The Agent will initiate contact with the OMS to upload metric data, update the target and metric configuration, or send over the result of a job or administration task the Agent has executed on behalf of the OMS. These communications are not on a fixed schedule: As soon as the Agent has information available to be sent over to the OMS, it will initiate a connection and send the information over. For secure communication from the Agent to the OMS, the default destination port 1159 will be used (Figure 1-3). This will be used by almost all the communications from the Agent to the OMS. For unsecure traffic initiated by the Agent, the default destination port 4889 is used. With the default rollout, the only unsecure connection the Agent will make is the first connection the Agent makes after install to obtain the secure wallet from the OMS. As soon as that secure wallet is obtained, all further communication done by the Agent will use the secure port 1159.

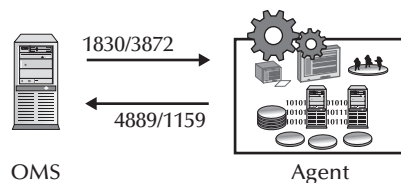
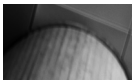


FIGURE 1-3. *Agent and OMS communication*

**NOTE**

To find out which ports are being used by the OMS for secure and unsecure communication, check the `emoms.properties` file, located in the `$ORACLE_HOME/sysman/config` directory:

```
oracle.sysman.emSDK.svlt.ConsoleServerPort=4889
oracle.sysman.emSDK.svlt.ConsoleServerHTTPSPort=1159
```

**BEST PRACTICE**

If there are multiple Management Servers running in the infrastructure, make sure all of them are configured to listen on the same ports. It is technically possible for each OMS to use different ports. But for obvious reasons, make the network administration a lot easier, and always use the same set of ports for each OMS. (Tips and tricks on how to force the install to use the same ports will be explained in detail in the next chapter, covering the install and setup of Grid Control.)

Similarly, the OMS can initiate a connection to the Agent to update the monitoring information of the Agent, or to send over job information with details on how to execute the various commands of the jobs. This type of communication is also not on a fixed schedule, and is initiated as needed.

Based on whether the Agent is running in secure or unsecure mode, the OMS will choose either HTTP or HTTPS to contact the Agent. The port it will use is the port the Agent has sent over with the initial configuration information of the Agent target that was discovered during the install. For 10gR1 installations, the port the Agent uses is 1830. For 10gR2 installations, port 3872 will be used by default.

**NOTE**

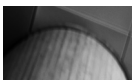
To find out which port the Agent is using to communicate with, check the `emd.properties` file, located in the `<EMHOME>/sysman/config` directory:

```
EMD_URL=https://myhost.acme.com:3872/emd/main/
```

The `<EMHOME>` directory is the state directory the Agent uses. For regular Agents, this directory will be the same as the `$ORACLE_HOME` of the Agent software. But for some types of Agents, the state directory will be different from the `$ORACLE_HOME` (more about that in Chapter 4).

To get the state directory the agent uses, run this command:

```
[oracle@agent ~]$ emctl getemhome
```

**BEST PRACTICE**

The same best practice applies to communication toward the Agents as well: Standardize on a single port to be used for the Agents. Having multiple ports in use for the Agent communication will make it a lot harder to configure firewalls. (The next chapter will shed some light on how to force the install to use the same port.)

An additional benefit of standardizing on a single port is the ability to spot the ugly duckling that got installed in a different way: Any Agent that is not using the standard port (3872 by default) can be easily identified.

The following query will list all Agents that have not used the default port 3872 during install. Run this as the owner of the Grid Control repository (SYSMAN user):

```
SELECT target_name
FROM   mgmt_targets
WHERE  target_type = 'oracle_emd'
      AND SUBSTR(target_name, INSTR(target_name, ':')+1) !=
      3872;
```

All information the OMS and the Agent exchange is structured in XML format. There are various types of information:

- The OMS server sends changes to the configuration of a target (called the *monitoring configuration*) and the way the target is being monitored (metric settings and thresholds) to the Agent.
- The Agent will upload any monitoring information, including metric data and the generated alerts.
- For synchronous job tasks, the OMS server will send over the command, and wait for the Agent to reply using the same connection.

For long-running operations, the OMS server will send over the information, and close the connection. When the Agent has finished the task, it will initiate a new connection back to the OMS server to upload the results.

Because of the nature of the HTTP (or HTTPS) protocol, it is possible to have Agents scattered around the network. There are no restrictions on where to deploy Agents. As long as the OMS can communicate with the Agent, and the Agent can communicate and upload XML files to the middle tiers, Grid Control can monitor the server machine through the deployed OMS servers.

With the central function of the OMS server, however, the Management Servers will have to be strategically positioned in the network, to allow these machines to communicate with all the machines in the enterprise that require monitoring. Especially in enterprise-wide deployments, with (sometimes multiple) firewalls involved, this will be a factor that needs to be discussed and verified with the network administrators to successfully install Grid Control.

OMS-to-Repository Communication

The OMS is the only tier that makes a connection to the repository database (Figure 1-4). The Agents will communicate only with the OMS, and the logged-in users will also make all requests to the OMS. The Management Server will then take those requests, store and retrieve the necessary information to and from the repository, and send the resulting data back to the requestor.



FIGURE 1-4. OMS and repository communication

The OMS server creates several thin JDBC connections to the repository database. These connections are split up in connection pools. Each pool is dedicated for a specific function. If a connection in a pool is not in use, it can be reused by another thread using the same connection pool in the OMS. If there are several OMS servers running, each middle-tier server will have its own connections for each pool.



NOTE

When you are checking the `gv$session` view in the database, you can find the connection pools by looking at OMS sessions, made from an OMS machine, which have the value of the `Module` column set to one of those pools.

Use the following SQL query to narrow this down:

```
SELECT inst_id, sid, serial#, machine, module, action
FROM   gv$session
WHERE  username = 'SYSMAN'
       AND program = 'OMS'
       AND module LIKE '%Pool';
```

There are four different connection pools the OMS uses, each with its own cryptic name. Each of the pools is part of a critical part (sometimes called a subsystem) of the OMS, performing either internal housekeeping tasks, or responding to requests made by the Administrators or the Agents:

- **OEM.BoundedPool** Set of connections used to upload the Alert information from the Agents. This information also includes any updates to the condition of the targets the Agent is monitoring, and the setup and definition of the collections and metrics the Agent is monitoring (more about this in Chapter 7). This special connection pool is set-aside especially for this purpose, to guarantee no interruptions or delays when loading this type of data.
- **OEM.CacheModeWaitPool** This pool is only used to process updates to the Grid Control jobs. More details about the Job system can be found in Chapter 10.
- **OEM.CacheModeWaitnPool** All the processing notifications to inform an Administrator about a new Alert are done using connections from this pool. In-depth coverage about these notifications can be found in Chapter 9.
- **OEM.CacheModeWaitrPool** Connections used by the Shared File System loader. This is a High-Availability (HA) feature. The details on how to use and set up this pool can be found in Chapter 3.
- **OEM.DefaultPool** Every user logged in to the Grid Control application will use a connection from this pool. Since the connections are pooled and reused, multiple Administrators logged in to the Console can be using the same connection on the OMS to process the requests.
- **OEM.SystemPool** This is used for all internal OMS operations and housekeeping tasks.

All data that needs to be persisted is saved in the repository. This can come from Agents in the form of metadata (target definitions, metric settings), state data (condition of the targets and metric run on that target), configuration data, and metric data.

Data can also come from Administrators who are logged in to the Console and updating configuration or providing new administration data.



NOTE

To find out which port the OMS server is using to create the database connections, check the `emoms.properties` file:

```
oracle.sysman.eml.mntr.emdRepPort=1521
```

In case of a RAC repository database, the port will be taken from the TNS descriptor used to connect to the database:

```
oracle.sysman.eml.mntr.emdRepConnectDescriptor=  
(...TNS descriptor... (PORT\=1521)...rest of TNS  
descriptor...)
```

An OMS server on startup will create several connections in the repository database. This includes the connections to perform the housekeeping tasks, handle the job requests, respond to the Administrators logged into the Console UI, and upload all the data send by the Agents.

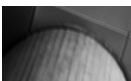
With the out-of-box settings, each OMS server will create a minimum of 14 connections to the repository. More connections can be made depending on the number of Agents uploading data and administrators logging in to the system.



NOTE

More information about how to tune and change the OMS server settings to manage the database connections can be found in Chapter 12.

Based on this information, in combination with the overview of how the Agents communicate with the OMS server, we can now draw a second conclusion on how to deploy and where to put OMS servers in an Enterprise Manager deployment. With the OMS creating several connections to the repository database, and the fact that the information loaded into the repository through those connections will come in bursts, either from Agents uploading XML files or from users requesting information via the Console UI, the OMS server should be installed in the close vicinity of the repository database. The network latency should be kept to an absolute minimum between the OMS servers and the repository database to reduce the response time of the operations an OMS makes into the repository database: Any network hop that can be eliminated should be eliminated.



BEST PRACTICE

There are several tools available to monitor network activity, or to test network connectivity. For a quick-and-dirty approach to test the responsiveness of a remote machine, the standard command `ping` can be used. Ideally, the response time returned by the `ping` command should be in the single digits, or the low double digits in milliseconds.

There will be severe performance implications for the OMS server responsiveness to user requests (Console performance) and Agent upload request (XML file throughput) when the response time reported by `ping` reaches 100 milliseconds and higher.

User (Web-Client) to OMS Communication

A user connecting to the Grid Control Console will have to use a web browser to do so. Web browsers by definition only contact the HTTP server, and will never be the recipient of a connection from the server.

There are two sets of ports to access the OMS:

- A direct access path to the OMS servers, using ports 1159 and 4889. These ports are used by the Agents to talk to the OMS.



BEST PRACTICE

For large enterprise-wide deployments, it is strongly advised to keep the traffic from the Agents separate from the communications between the users and administrators. The ports 1159 and 4889 should therefore only be used interactively (using a browser) to debug a problem, or when testing communication. All other UI-based activity done by users of Grid Control should use the Oracle HTTP Server (OHS) ports (or the Webcache ports if Webcache is used).

- Access to the OMS servers using ports 7777/7778 and 4444/8250 (Figure 1-5). These ports are the main ones used by the users connecting via a console. Ports 7778 (unsecure, HTTP) and 8250 (secure, HTTPS) are defined by Webcache, and forwarded to OHS on ports 7777 and 4444 respectively.



NOTE

Webcache is not a strict requirement for the application rollout. Given the dynamic nature of Grid Control, with pages getting constructed on the fly based on the specifics specified by the logged-in user, the benefit of using Webcache to speed up the web pages is very limited. It is needed only if end-user monitoring is used to track the usage of specific URLs of the application.

Users logging in to Grid Control can use either of these two sets of ports. The reason for the second set of ports is to separate the traffic from the Agents from traffic from the people using Grid Control.

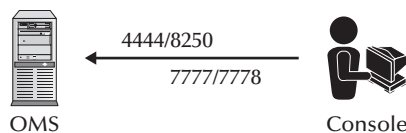


FIGURE 1-5. Console-to-OMS communication

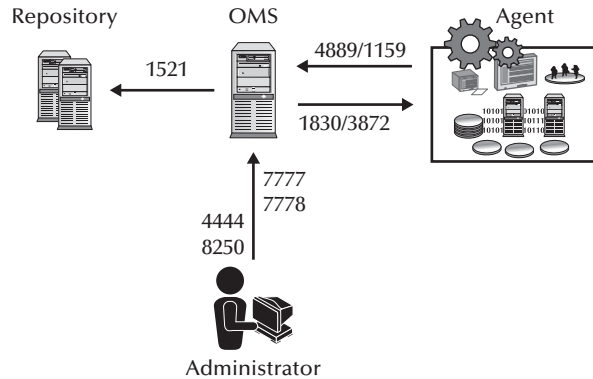


FIGURE 1-6. TCP ports used by each tier

TCP Ports Used by Enterprise Manager

Figure 1-6 shows all the TCP ports that Grid Control release 10.2.0.4 or higher uses. Table 1-1 describes the use of these ports.

Port	Used By	Communication To and From
1159	OMS Server	Used by the Agents to communicate to the OMS servers in a secure way.
1521	Repository	Used by the OMS servers to communicate to the repository database listener.
3872	Agents	Used by the OMS servers to communicate to the Agents. Can be either secure or unsecure.
4444	OMS Server	Used by the Administrator's browser UI to communicate to the OMS server (HTTP Server) in a secure way. The traffic will be redirected to port 8250.
4889	OMS Server	Used by the Agents to communicate to the OMS servers in an unsecure way (during securing requests).
7777	OMS Server	Used by the Administrator's browser UI to communicate to the OMS server (Webcache Server) in an unsecure way.
7778	OMS Server	Used by the Administrator's browser UI to communicate to the OMS server (HTTP Server) in an unsecure way. The traffic will get redirected to port 7777.
8250	OMS Server	Used by the Administrator's browser UI to communicate to the OMS server (Webcache Server) in a secure way,

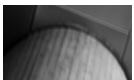
TABLE 1-1. TCP Port Details

The Different Types of Targets

After this brief explanation of Grid Control and its basic functions, it's time for a list of the different types of targets that can be monitored. The idea is not to go into much detail at this point, but to give an idea of the capabilities and possibilities that Enterprise Manager can offer. Most people still are under the impression that Grid Control is just a database-monitoring tool. Over the years, the monitoring capabilities have been extended far beyond merely databases, making it possible to monitor a lot more targets that are typically present in today's data centers. Each different type of target added to the environment has its own requirements to keep track of. And with each new monitoring capability comes new preparation that needs to be done to make the monitoring and administration possible.

Besides drawing the attention of administrators beyond that of database monitoring, the intention of this list is also to make people aware of the additional steps and preparation that need to be completed to make the monitoring possible.

A lot of these targets are provided out of box with each install of Grid Control. Others are plug-ins (extensions) provided by either Oracle or partners to extend the default set of monitoring capabilities of Grid Control.



NOTE

For a complete list of all available monitoring extensions, check the Grid Control Extensions Exchange web page on OTN (Oracle Technology Network):

<http://www.oracle.com/technology/products/oem/extensions/index.html>

Here is a brief overview of the type of targets to be monitored:

- **Host, Agent, and Cluster** This is the cornerstone of all monitoring. Each monitored machine will have an Agent and a host target defined. If CRS is installed, a third target of the type "Cluster" will also be added to the default list of targets the Agent has after install.
- **Databases** This goes beyond Oracle databases. Besides the single database instances, RAC databases, and Data Guard instances, other type of databases like Microsoft SQL Server, Times Ten In Memory Database, Sybase, or IBM DB2 databases can also be monitored.
- **Application Servers** Besides Oracle Application Servers, Grid Control can monitor BEA Weblogic, JBoss Application Server, Microsoft Internet Information Server (IIS), Internet Security and Acceleration (ISA), Biztalk and Commerce Server, and IBM Websphere middle-tier installations.
- **Application Suites** This includes Oracle Ebiz and Collaboration Suite, PeopleSoft and Siebel Application, Oracle Identity Management, and SAP installations.
- **Network devices** Includes F5 Server Load Balancers, Juniper NetScreen, and Nortel Alteon Application Switch.
- **Storage devices** Among the different storage devices are Netapp filers, Onara SANScreen, and EMC Celera, Clariion, and Symmetrix systems.
- **Grid Control-specific Container targets** This includes Groups (logical groups of targets), Systems (set of physical targets that run a specific application), Services (logical definition of service, an application), and Beacons (network point to test connectivity with applications).

This list is not definitive. Additional plug-ins and monitoring extensions are uploaded regularly on OTN, and support for new target types is added to Grid Control with each new release of the product.

Summary

Now that we have been acquainted with the range of products that makes up Oracle's Enterprise Manager, two things should be clear by now. First, we can no longer profit from the luxury of placing live bodies to monitoring every new deployed database or middle-tier server. This solution doesn't scale in today's IT environment where environments exist in multi-national locations and accompanied by increased complexity from the applications that get deployed into them. Second, anytime an environment is scaled out, there is significant expense in ensuring the environment is effectively utilized and available. This is because prior to enterprise management technologies, the length of the feedback loop for discovering system imbalances then implementing corrective actions was lengthy and manually intensive. Throughout the rest of the book we will look at these various architectural challenges and now with Grid Control as part of the solution, we will discuss why the technology is a good fit.