Scaling-Out with Oracle® Grid Computing on Dell™ Hardware

A Dell White Paper

J. Craig Lowery, Ph.D.
Enterprise Solutions Engineering
Dell Inc.
August 2003

Increasing computing power by adding inexpensive processing and storage nodes to an existing infrastructure is a concept Dell refers to as "scalable enterprise computing" or "scaling out." Successful scale-out architectures manage workloads such that component utilization is optimized, cost is minimized, and availability and performance are maximized. Oracle's grid computing concept is an example of this kind of architecture. As Oracle continues to add new features to its products that automate workload redistribution and load balancing, it becomes increasingly clear that Dell's scale-out strategy and Oracle's grid computing concept are complementary, and capable of achieving many of the objectives of the "virtual" data center.

Optimal resource allocation: the new driving goal

Until recently, the driving design goals in the data center were availability and performance. As businesses became more reliant on computer systems to support critical operations – especially those facing customers, such as e-commerce – the costs associated with viable solutions were far outweighed by the consequences of not deploying them. To that end, computer vendors endeavored to deliver clustering solutions that provided fault-tolerant, load-balanced computing environments with cost playing a lesser role. Dell, for example, certifies both high availability (HA) and high performance computing (HPC) clustering solutions. Although not suitable for every application, clustering technology has done much to bridge the performance and availability requirement gaps in the data center.

Now that the primary goals of availability and performance have largely been achieved, many data center managers are returning to the issue of cost. They realize that each of the clustered systems they have deployed is sized to accommodate peak rather than average demand. The cost associated with underutilized resources is an obvious target for improving the bottom line. Ideally, computing resources or units such as servers and storage should be moved *between* application clusters, according to *current*

demand. Since it is unlikely that all applications experience peak demand simultaneously, if units are transferred via software between clusters as needed, the total number of units should be able to be reduced, thereby reducing cost. In the unlikely event that demand exceeds total available resources, prioritization of resources can ensure that critical systems are not starved, and if the increased demand persists, simply connecting new units to the existing infrastructure can increase capacity. These concepts form the basis for scalable enterprise computing, also known as the *virtual data center*, and have been described elsewhere.[1]

The statistical multiplexing of hardware across application clusters is ultimately at the core of the scalable enterprise computing concept. In the virtual data center, hardware is physically configured only once. Software is used to create logical associations between hardware components as needed. For example, virtual local area networks (VLANs) can be configured through software that is resident on a network switch. The abstraction of hardware to pools of similar, easily "relocated" components is what makes the virtual data center possible, and it is this characteristic that uncovers the new goal of optimal, real-time, dynamic resource allocation purely through software.

The virtual data center

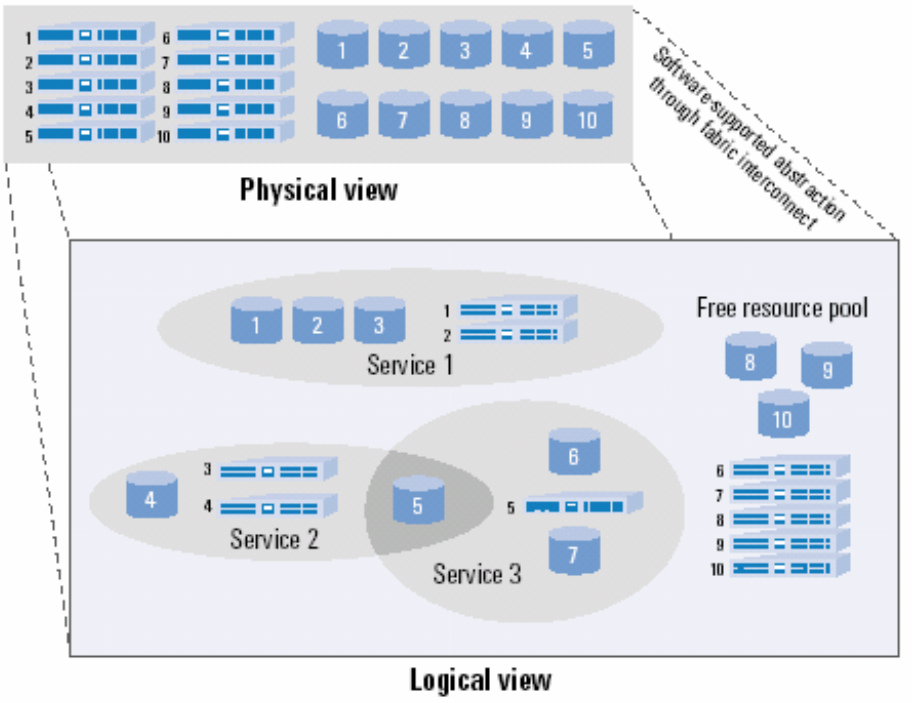

Figure 1 - A Virtual Data Center

[1] For more information, see "Building the Virtual Data Center" by J. Craig Lowery, Ph.D., in *Dell Power Solutions*, February 2003, and "Managing the Virtual Data Center" by J. Craig Lowery, Ph.D., in *Dell Power Solutions*, August 2003.

A virtual data center is depicted in Figure 1. Viewed physically, the data center is simply a collection of low-cost, standards-based, nearly homogeneous computing resources – namely servers and remote storage. By "nearly homogeneous," we mean that the components are similar enough to be interchangeable when projected into the logical view. For example, servers that are based on Intel architecture that implement the same level of remote management features and interconnection facilities could be considered nearly homogeneous, even if they are different models. The logical view is one created entirely by software. By using the facilities of a high performance interconnection network, this software is able to create logical associations between various components that change on demand over time, without the need for physical reconfiguration.

Figure 1 shows three logical groupings, or clusters, of resources, designated "Service #1," "Service #2," and "Service #3." Service #1 is self-contained, whereas Service #2 and Service #3 share some mutual data through the remote storage. Those resources not participating in a service are held in reserve and can be used to create new services, replace failed resources in existing services, or add additional computing power to existing services. The decisions to make these logical configuration changes are ideally automated, and are based on reactions to changes in demand for the various services.

Balancing loads and resources

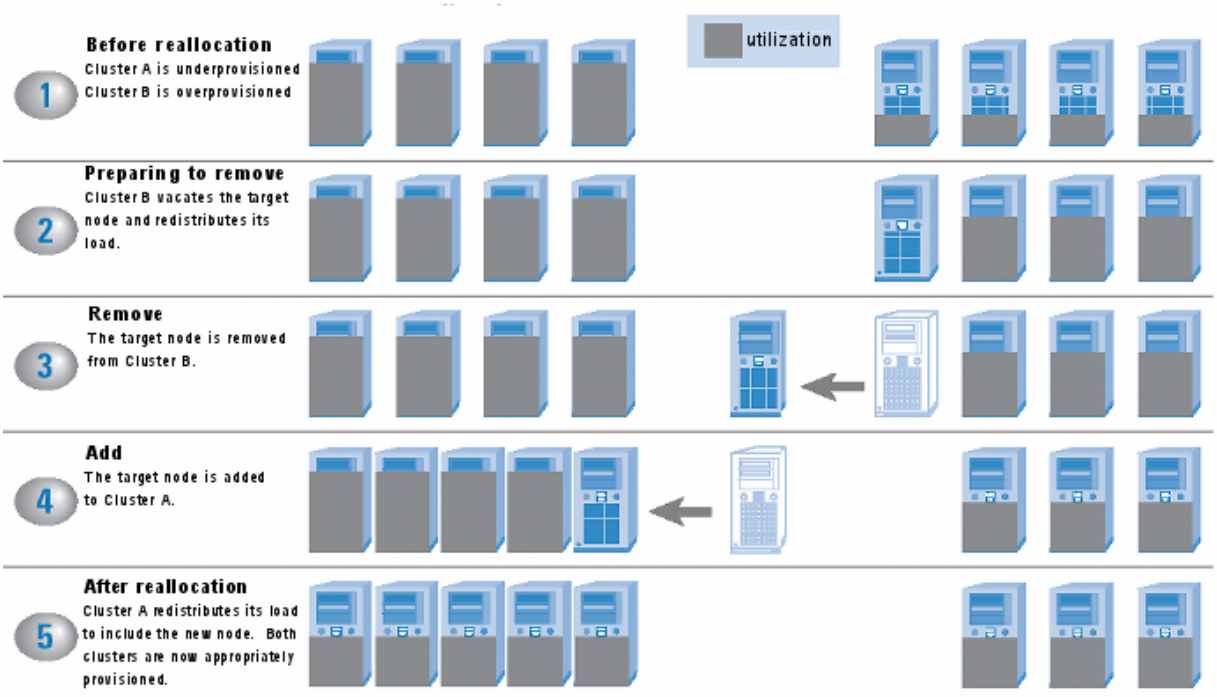

Figure 2 - Load Balancing and Resource Reallocation

Two mechanisms work in tandem to achieve maximum performance and optimized utilization in the virtual data center:

- **Load Balancing**. This is the ability to redistribute work across the nodes in a cluster, proportionate to each node's work capacity.
- **Resource Balancing.** This is the ability to move nodes between clusters, increasing and decreasing cluster sizes and, thus, their capacities for work.

The application of these two mechanisms is shown in Figure 2, and can be implemented in three operations: *redistribute* work, *remove* node, and *add* node. There are two clusters, A and B, running potentially very different applications. In step 1 of the figure, cluster A is experiencing heavy demand and is nearing a saturation point; it is *underprovisioned* because it requires additional hardware resources. Cluster B, on the other hand, is experiencing a light load and has spare capacity; it is *overprovisioned* because it has an abundance of hardware resources. In step 2, a node in Cluster B is identified for transfer to Cluster A. The identification algorithm can be selected to minimize time until the target node is released, minimize impact to clients, or other arbitrary functions. Cluster B uses workload redistribution methods such as session migration to *vacate* this node. In step 3, the target node is vacant and is removed from Cluster B. In step 4, the target node is added to Cluster A, which begins to redistribute the work of the cluster across the new membership. By step 5, a steady state workload has been established in Cluster A, and both clusters are appropriately provisioned.

Load balancing is occurring within the clusters; resource balancing is occurring between the clusters. Achieving this symmetry is the main goal of the virtual data center. Unfortunately, while simple in concept, the implementation is non-trivial.

As illustrated, the *redistribute*, *add*, and *remove* operations allow hardware resources to be reallocated according to changes in demand. *Add* is an easy operation to implement because it does not require an immediate reaction from the affected cluster; new nodes are naturally integrated in a non-disruptive fashion through *redistribute*. *Remove* is also not difficult if one simply redistributes work to vacate the target prior to removal. Clearly, of the three operations, *redistribute* is the most critical.

Oracle software provides hardware transparency and optimization

The concepts of load balancing, resource balancing, and workload redistribution are hardly new; they have been described in academic distributed operating system literature for at least 20 years.[2] Much research has been done into creating such systems, leading to many of the advances now incorporated into the virtual data center concept. One of the key differentiating characteristics of a distributed operating system as opposed to traditional ones is *transparency*. In other words, users of the system do not know, nor do they need to know, what components of the network are cooperating to service their requests, or how the components go about accomplishing it.

Since the virtual data center is a form of distributed operating system, transparency is not surprisingly a primary goal. For the virtual data center to be truly successful, it must

[2] E.g., Andrew S. Tanenbaum and Robbert Van Renesse. Distributed Operating Systems. ACM Computing Surveys, Vol. 17, No. 4, December, 1985.

host applications that are not affected by details of the underlying hardware, such as location, and require no special programming to accommodate migration. Application developers should not need to worry about synchronizing with re-configuration events. Ideally, the virtual data center presents applications with a virtual machine that provides continuity of execution at all times, making no special demands on applications to accommodate reconfiguration below this virtualization layer.

The Oracle grid architecture[3] is an excellent example of a virtualized application execution environment, including the facilities for workload and resource redistribution. Oracle database and application clustering products include this kind of technology. Oracle grid computing is realized through the use of one or more of the following Oracle product features:

- **Oracle Real Application Clusters (RAC):** Oracle RAC supports adding and removing server nodes. The nodes cooperate to support a service by using shared storage, such as a storage area network (SAN).
- **Transportable Table Spaces:** This facility allows for bulk data migration between Oracle database systems. Transportable table spaces makes possible to relocate data within the data center, moving it closer to the computing nodes where it is most needed.
- **Streams:** This facility provides a means for distributing data across multiple nodes, and for keeping multiple copies synchronized by communicating changes in the data rather than entire tables.
- **Distributed SQL and Transactions:** SQL queries can be run across nodes, in parallel, and the results combined.
- **Generic Connectivity:** Non-Oracle databases can be included in many distributed operations by using industry-standard access interfaces.

The Oracle software stack provides a consistent virtualization layer above the hardware and operating system. A key tenet of Oracle grid computing is the use of low-cost standard modular hardware components – servers and storage – for building the grid. Because of their low cost, Dell's industry standard servers running a Linux operating system are particularly suited to Oracle's grid vision.

Momentum is gathering

Achieving cluster reconfiguration and load balancing so that it is completely transparent to applications is extremely difficult. Experimental attempts have been made to create these environments, but no standards-based, commercially viable solution that meets all of the requirements of the virtual data center currently exists. Oracle's grid

---

[3] "Oracle and the Grid: An Oracle White Paper", November 2002, Oracle Corporation. http://otn.oracle.com/products/oracle9i/grid_computing/OracleGridWP.pdf.

computing architecture is one example of an emerging implementation that leverages standard hardware and software components. The Oracle grid architecture continues to evolve along with Oracle's product line, and customers – drawn to the improved cost efficiencies – are taking notice of grid computing and similar technologies. Dell hardware components combined with industry *de facto* standard operating systems, most notably Linux, are a good choice for building Oracle grids.

Copyright ©2003 Dell Inc. All rights reserved.