19

THE COMPUTE UTILITY

The evolution in distributed computing is leading us toward a business focus where we have to consider how systems are designed and delivered. This line of thinking is a shift toward service-oriented architecture (SOA). The philosophy of software packaging is taught in every Computer Science 101 class starting with a compartmentalization of logical units of work with a well-defined interface. Just about every programming language supports some form of macros, subroutines, functions, objects, and services. Each is a progression in granularity of function starting with fine grains of work in macros to a more coarse grain definition of objects with a service as the coarsest or broadest in scope. An example of a service is a payment system. The payment system can be composed of 10s or 100s of individual objects, each specific in scope and function to deliver a piece of the larger payment system service. Only the service made up of the user interface to the payment system is visible to the user or consumer, not the finer-grained objects of which the complete payment system is composed.

As businesses offer more and more services to the user community, a change in the way the services are delivered and packaged results. It is important to note, as mentioned earlier, that siloed data centers do not lend themselves to service delivery. What was once an application for a specific user group that maintained its own hardware within a data center (a silo) now becomes a corporationwide service to be leveraged across the business units on an on-demand basis as well as a client service available externally. The shift in offering services immediately implies a different business model for information technology. The technology is required to transform to a consumer–producer model. The user groups are the consumers purchasing the

Distributed Data Management for Grid Computing, by Michael Di Stefano

services that are offered by information technology. Therefore, the consumer satisfaction rating of the service is measured by the functions provided by the service and the availability/performance of the service as the service is utilized. This implies that the infrastructure required to deliver the service to the customer now becomes an integral part in defining the quality of the service. This requires a change in the way the services are delivered. With this model, in order to meet the supply/demand curves that the consumer will levy on the infrastructure and its ability to deliver the service, the data centers need to be

- *Flexible*—responsive to change
- *Fungible*—interchangeable, substitutable
- *Scalable*—growing with business demands

There has been an emergence of standards from various corporations and research groups for the definitions of how to deliver the business services. Some examples are IBM's On Demand Business,[32] Sun's N1-Grid, Stanford's Compute Utility Architecture, The Global Grid Forum,[33] "cluster on demand,"[34] and the HP Adaptive Enterprise.[35] All the standards describe the architecture required to deliver services and use buzzwords such as "the compute utility" and "the virtual data center." We will examine the basic architecture for a compute utility, including its components and objectives, the architecture, and the interaction of the components so that the compute utility can be as efficient as a manufacturing plant or factory. On the broad scale, the user demands on the manufacture of a product or to supply the services are known to the plant manager; however, like many things in life, the factory production line will undergo daily changes based on user demand that they need to be addressed.

The architecture that will be discussed in this chapter originates primarily from two sources: a white paper that was released by Hewlett-Packard in March 2001 on virtual data centers and their operating environments, and an extension of this architecture that was a derived by the partnership work of Integrasoft, Platform Computing, and Corosoft.[36] Each component or layer of the compute utility is vital to the compute utility as a whole and equally complex in function, deserving of more attention than can be provided in the forum of distributed data management. Our discussions will concentrate on the role of data management within the compute utility.

OVERVIEW

The compute utility has many objectives, which are reflected in its design. To summarize some of these objectives, one must consider the purpose of the compute utility, which includes

- Reduce risk exposure.
- Lower operational costs.

- Achieve controlled and predictable costs for new implementations and services.
- Improve responsiveness of information technology to the business units that they support.

These goals can be achieved without cannibalizing the current data centers, which will not be an option in any real scenario. Many companies cannot justify reinvesting and doing away with their current data center investments in order to build and offer utility services. Therefore, the new architecture needs to reuse as much of the existing infrastructure as possible by at least leveraging the computer servers to increase utilization rates and build on the intra/inter–data center networking infrastructure. The key milestones must be the reduced complexity in the data center from an operational and maintenance perspective through more and better automation, providing an elastic infrastructure support and cost transparency to the business units.

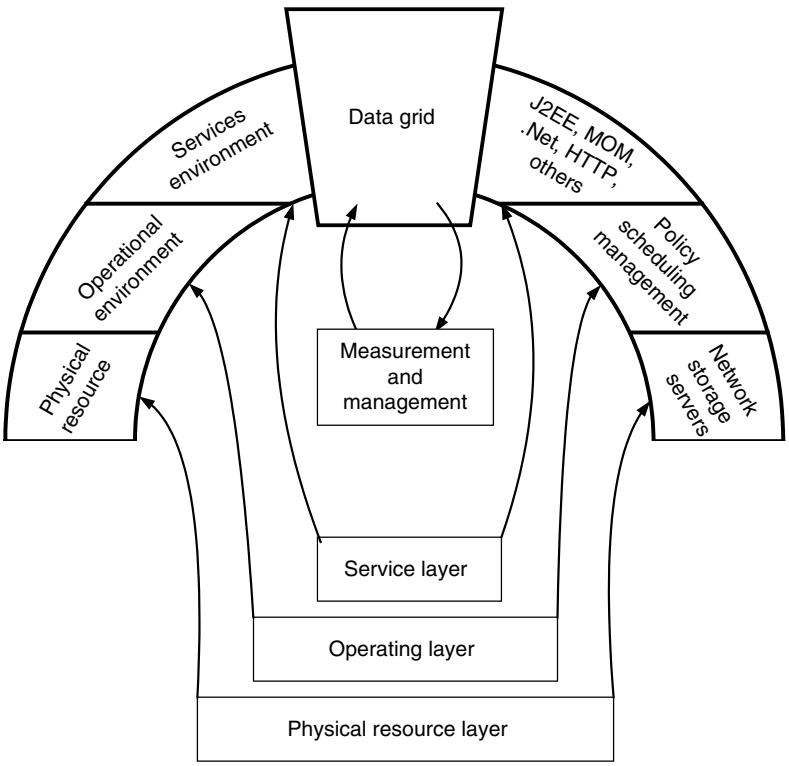
In order to meet these objectives, one needs to determine what is required in data centers. Well, the data grid is the answer. The role of the data grid within the compute utility is to provide visibility into the system. Visibility into the compute utility extends the current thinking of what visibility is in a siloed IT infrastructure. "Visibility" into the utility refers to the information infrastructure needed to deliver a service that has been perfected in industries other than computer software. Examples or models that IT can use when architecting the compute utility and considering visibility into the utility are the telephone line, power (electricity), and energy (natural gas). These respective industries have vast infrastructure for delivering service to the consumer on a supply and demand basis. A key ingredient in delivering the service is the *timely availability of accurate metered data.* So the key to a utility service is accurate metered data in order for that service to meet the demands. The system as a whole records and tracks information about itself, including its state, and feeds the data back to a control center. The metered data must include usage information. This information will allow the utility to change state in such a way as to accommodate the increase of demand as usage peaks occur. Similarly, as the usage decreases, the system can change its state to adjust to the lower service demand. The system will always be in the state to best offer the service to the customer on an as-needed basis.

Today in IT, data centers and the applications, that they support do not monitor or track any required metered data. The data grid implemented as part of the existing data centers architecture that can be used as the focal point to collect metered data and make them available to all users in a timely fashion, thus enabling broad and deep visibility into the state of the compute utility. For example, user demand is placed on the system by the consumer. With timely, accurate, and visible metered information describing the state of the compute utility, smart decisions can be made and appropriate actions taken to keep the compute utility in a predictable steady state. The steady state of a compute utility is just like the steady state of a factory, which I touched on above. It needs to be to efficient, in a time- and cost-effective manner, and to deliver service to the consumer when the consumer

demands the service. The quality of the service is equally evaluated as to how well the service functions when the demand is imposed on it. The data grid is not only the collection focal point of metered data; it is also the event channel to deliver control commands issued by the command mechanism. The data grid will deliver the instructions to the various components within the compute utility to complete the command/control management loop.

ARCHITECTURE

The architecture for the compute utility consists of five basic components: the physical resource, the operational environment, the service environment, the data grid, and the measurement and management (command and control). Figure 19.1 shows the compute utility architecture in an arch rather than the traditional block



Four base layers supporting a SOA with the data grid as the keystone of the compute utility arch

Figure 19.1. Compute utility architecture.

diagram format. The arch is one of the oldest and strongest infrastructures known. Its beauty is in its simplicity.

The architecture for the compute utility mirrors the simplicity and strength of the arch. At the foundation, the physical resource layer, is the physical resource, which includes the network, the storage, and the servers. The next layer up is the operating layer or environment. The operational environment is a logical view of the physical resource layer. It is a management layer that is policy-driven to keep track of, allocate, and provision the physical resources. The next layer of the arch is the service layer or environment. These are the typical services that we have become dependent on in distributed computing such as message-oriented middleware (MOM), J2EE, microsoft.Net, and HTTP. As with any arch, the strength of the arch comes from the keystone; the keystone to the compute utility arch is the data grid. The data grid's primary function is to provide the mechanism for collecting the metered information from within each of these layers and deliver this information or data to the measurement-and-management component of the compute utility. It is the interaction between all the layers of the compute utility through their metered data that allows this utility to deliver services on a supply/demand basis. With the absence of the data grid, the collection of metered information becomes cumbersome and the responsiveness of the utility to the users diminishes, thus lowering the quality of service that is delivered to the customer.

We have already addressed the major aspects and components of data management of the compute utility in our discussions on geographic boundary problems and command and control.

Geographic Boundary

The beauty of the data grid to manage and meter data is such that the physical resources do not need to be contained within a single data center. The operational environment abstracts physical locality of the servers and data centers to the layers above it (the service environment and ultimately the consumer). Recall from previous discussions that the limitations of geographic boundaries are not distance but are rather the network bandwidth. The data grid and the compute grid are the tools in the operational environment that provide the geographic independence and establish a policy-driven view of the physical resource.

Command-and-Control Systems

In our use case for command and control, we discussed how the data grid is essential for the collection of vast amounts of data from numerous dispersed sources. Each layer of the compute utility (the physical, operational, and service environments) must generate metered information. Each subsystem, software program, server, router, hub, and other component is generating new bits and pieces of information that individually tend to be smaller in size but taken as a whole end up as vast amount of data moving very quickly in time. In the command/control analysis it is the data grid that provides the mechanism for the quick and easy collection as

well as access to these data. The data grid can also provide the distribution backbone for the commands issued by the command and control. Figure 19.2 expands on the use of compute utility architecture for command and control.

In the architecture shown in Figure 19.2, the role of the data grid in the compute utility is twofold. The first part is getting the application to run in the environment. For example, some delivered services include credit risk and anti–money laundering. The services themselves rely on the data grid to distribute the data across the compute utility to the physical locations (the physical compute nodes of the compute grid) where the service is located. Please refer to the section on compute-intensive applications in Chapter 13 and Figure 15.3, on data warehousing with the data grid, for more in-depth discussion on how the data grid functions in these business leveling services.

The second area where the data grid plays a key role in the compute utility is in the management of the utility itself and in the command-and-control aspects of maintaining the compute utility. Figure 19.3 shows the interactions between the

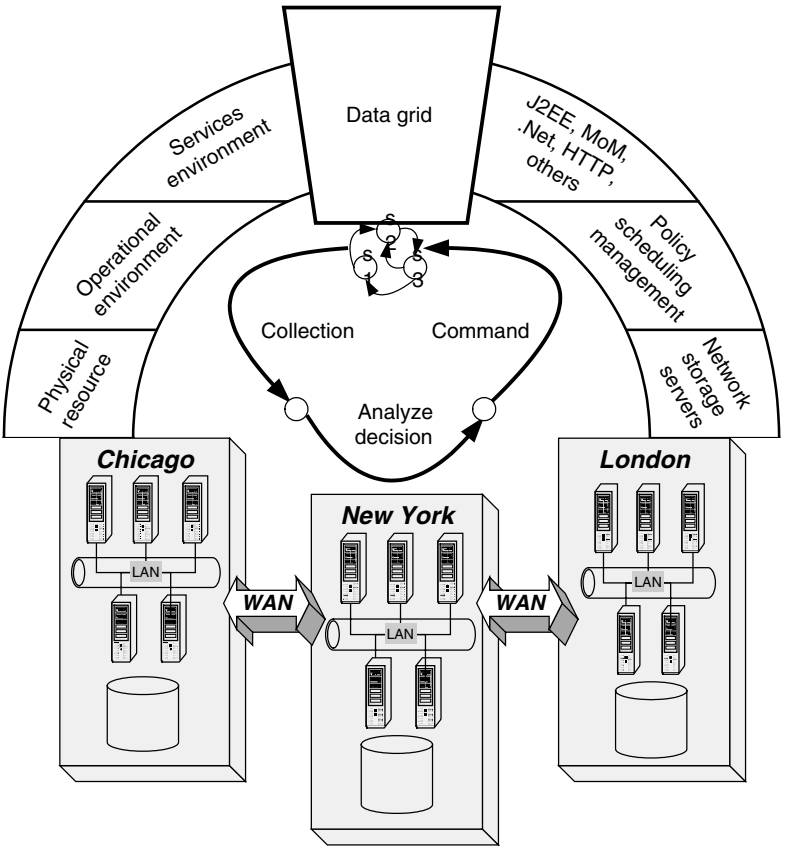


Figure 19.2. A broader view of the compute utility.

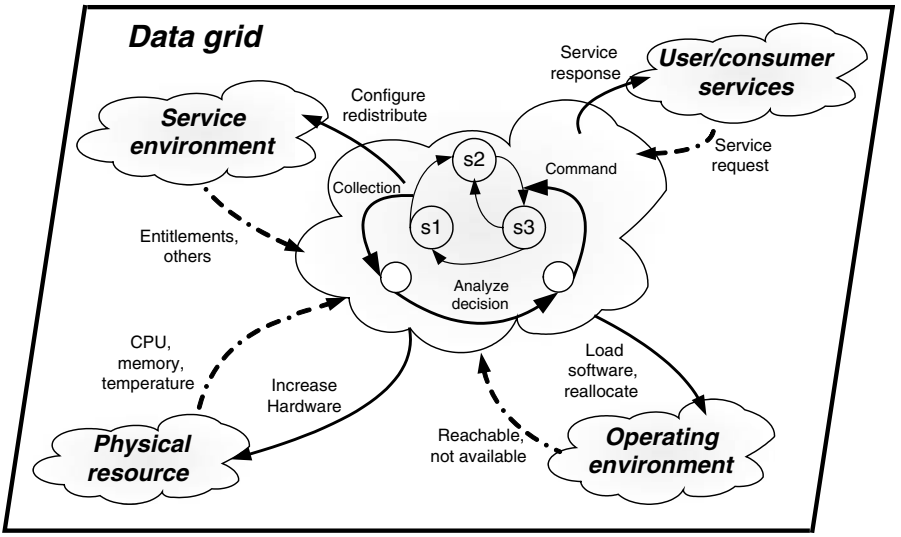


Figure 19.3. Compute utility interlayer data interaction.

various layers of the compute utility from the metered information that is generated by each layer and the command/control information that flows back to each layer.

Each layer has its own data region within the data grid: the physical resource, the operating environment, and the service environment, as well as the consumer services. The fifth data region is the command/control region. Each layer will generate metered information that flows into the command/control system via the data grid. At the physical resource layer, metered information can be CPU usage, memory utilization, temperature, and various networking statistics. The operating environments can tell us whether the physical resources are accessible or available. The service environment can give us information about entitlements. And, of course, the consumer services provide information usage such as workload, service, and user requests of the consumer community.

Macro/Microscheduling

Managers of an electric utility infrastructure or the power grid that delivers electrical power to the customer know that certain areas of the country have high usage during the summer months to deal with the hot temperatures. Similarly, natural-gas companies know that their peak usage periods are in the cold winter months. The managers can plan and adjust their utility infrastructures to handle the seasonal peaks and valleys of consumer supply and demand. However, even though electrical power demand in July and August is expected to be high in New York City, of brownouts and even blackouts still occur. Just the other evening, more than just the typical hot, hazy, and humid mid-August evening, by 10 P.M. it had managed to cool down to 90°F; I was at Yankee Stadium watching the Anaheim Angels shut out my beloved

Yankees (but that is another story) in the sixth inning, when parts of the Bronx experienced a blackout. The stadium scoreboard went dark, but for some reason, the stadium lights stayed on, allowing the game to continue. The point to note here is that even though the utility infrastructure is geared to deliver more power to places such as New York City in the summer, there are microscale peaks and valleys that augment tie macroscale peaks and valleys. Should the microscale peaks amplify or occur in such a way that exceeds even the best of expectations, the utility must respond. If it cannot respond quickly enough, or most likely if the demand peaks are above the ability of the infrastructure to deliver, circuitbreakers trip, causing brownouts or blackouts. The tripping of a circuitbreaker protects against wide-scale damage to the system. Remember that the entire northeast coast of the United States as well as parts of southeastern Canada lost power because the power grid tripped breakers to protect itself from a bigger potential danger. The only way this can occur is through measurement and analysis of metered data that describe the state of the utility. The metered data describe not only the state of the system but also the microscale peaks and valleys of user demand.

The command/control system of the compute utility performs a quick analysis of the metered information. The purpose of the analysis is not for macroscheduling but rather for microscheduling since the manager of the utility has a good idea of the larger cycles of consumer usage patterns on the utility. It is understood that there will be peaks and valleys for certain services throughout the day, the week, the month, and the year. However, as the user community's demand varies up or down (as a result of any number of reasons, e.g., variations in community size, external unpredicted events, local or global), you will see minipeaks and minivalleys, thus creating a microscheduling pattern that augments the macroscheduling of services. Think of macroscheduling as the carrier frequency of a radio signal and microscheduling as modulation of the carrier frequency. Figure 19.4 highlights micro- and macroscheduling and how to maintain a steady state for the compute utility.

Figure 19.4 shows three curves: the macroscheduling curve that the manager of the utility uses for long-term planning, the microscheduling curve describing unexpected peaks and valleys in usage demand, and a third curve that describes how the commute utility has to adjust to accommodate both the macro- and microscheduling demands. Both types of scheduling are needed to deliver the service.

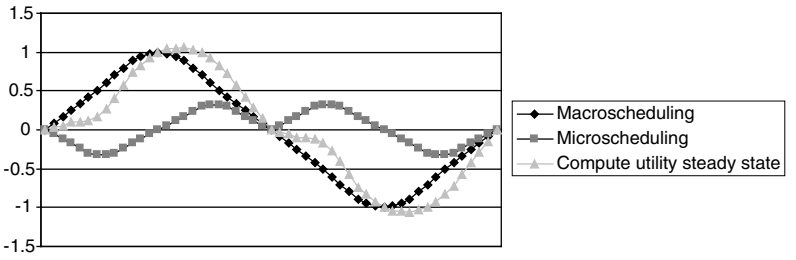


Figure 19.4. Macroscheduling, microscheduling, and the compute utility steady state.

The command/control system will issue commands back to the physical resource level to increase certain hardware components or reprovision servers with various software components. A top-down analysis starting from the customer service usage will show a chain of interdependence of metered information at each layer affecting commands to the next. The service environment will be issued commands to reconfigure or redistribute a service or services that will force commands down into the operating environments to load software or reallocate servers to supply the needed services, which will in turn cause commands to the physical resource layer to increase hardware or reprovision of servers.

We see two areas of information flow for the compute utility command and control. The two flows are metered information flowing up within each layer and the interdependency of information flow between each of the layers and command information flowing down.