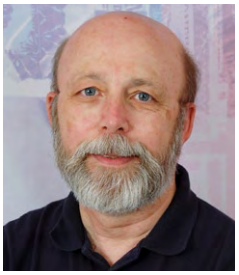
TeamQuest®

Multi-System Modeling

Ensuring realiable IT services doesn't just require provisioning a single system, but a series of interrelated layers each delivering a part of the end service. This includes the standard three-layer architecture — database, application server and web server. Portals and Service Oriented Architecture (SOA) adds to the complexity.

When you want to know how to improve service levels, it is not enough to model a single layer or each layer separately. Modeling multi-tiered applications requires modeling all the layers and the interaction between them. Only then can you be sure that they will work together to achieve the desired level of service to the end customer.

About the Author
Ron Potter is the Best Practices manager for TeamQuest Corporation. Ron's background includes more than 20 years in the IT industry, spearheading a successful ITIL implementation with a Fortune 500 insurance company, and discussing ITIL topics as a presenter at several conferences and trade shows.



Introduction to Multi-System Modeling

IT structures continue to grow in complexity. Predicting utilization and service levels was hard enough when it just involved an operator and a mainframe. Client/server added to the number of users and layers of servers. Service Oriented Architectures (SOA) take it to an even higher level.

With SOA, an IT end user is like a restaurant customer. In a restaurant, the customer places the order with the waiter and some minutes later the waiter delivers the order. What the customer doesn't see is the dozens of staff in the kitchen each doing their part to prepare, cook and present that food. Management must account for each of those back-end employees and their production to ensure the customer gets that meal on time. The process isn't linear, the order for each customer requires a different set of resources.

Increasingly, as IT determines the viability of businesses, the demand being placed on IT departments is to guarantee levels of service.

Do they want soup from the pot or salad from the refrigerator? A steak from the grill, or spaghetti from the stove? A baked potato from the oven or fries from the deep fryer? Is the drink coming from the wine cellar, the bar, the soda fountain or the coffee pot? Does dessert mean a scoop of ice cream out of the freezer, or something from the bakery? The food for each person at a table may involve completely different services, but all must arrive in the waiter's hands in a timely manner. And so it is with SOA, however many layers are involved in providing the service, those service levels must be maintained.

The Value of Modeling

Increasingly, as IT determines the viability of businesses, the demand being placed on IT departments is to guarantee levels of service. This involves not only understanding the current services and what it takes to create them, but also to be able to predict future resource needs as business operations continue to evolve.

This is where modeling comes in. When you are looking at capacity planning, you can take one of three approaches:

- Benchmarking — Running workloads on the actual hardware and measuring the performance to determine where bottlenecks occur. This can be done by building a complete, parallel infrastructure that fully duplicates the production environment or by running tests on the production equipment after hours. While accurate, this approach is also costly and time consuming.

- Trending — Graphing past performance and projecting this out into the future. This is certainly less costly than building a test environment, but is not very accurate. For one thing, business needs do not necessarily grow at an even rate. Applications get added or subtracted, new lines of business are developed, branch offices are opened or closed. The other is that machine capacity doesn't scale linearly, so doubling the number of servers won't necessarily double capcity. Capacity is determined by the point at which some element — CPU, memory, network — reaches its own limitations and thereby creates a bottleneck for the rest of the system. Trending won't tell at what point this will occur.

Capacity is determined by the point at which some element — CPU, memory, network — reaches its own limitations and thereby creates a bottleneck for the rest of the system.

- Modeling — Pilots train in a simulator before being given the keys to a 747 full of passengers, and IT needs to use simulations as well before making decisions on load allocation or hardware purchases. Simulation, or modeling, gives greater insight into both current and future operations at a fraction of the cost of benchmarking and with far greater accuracy than trending.

Multi-Tier Modeling Options

Modeling multi-tier systems involves unique challenges. In this case, the output of one set of servers acts as the input value for the next layer of servers. Specifying changes in workload intensity can only be done in one place — from the perspective of the end user — but any adjustments made to one tier to improve throughput also affects the performance of all the other layers. Identifying and coordinating the flow of units of work or transactions between the various servers is, therefore, critical in order to have a modeled unit of work that correctly represents a real-world scenario. The two factors of end user workload and changes in configuration at any tier must be coordinated to achieve accurate results.

The two factors of end user workload and changes in configuration at any tier must be coordinated to achieve accurate results.

The key benefit to modeling multiple systems is the ability to pinpoint at what tier the contention will occur and, more specifically, in which resource — CPU, disk, controller, etc.
Next comes the ability to change the configuration to see what the performance impact will be. Users can modify the hardware configuration to see, for example, whether adding more memory would solve the bottleneck rather than having to upgrade the entire server. This step can be repeated as many times as needed to determine the optimum configuration.

Before one gets into modeling, however, there are two other critical preparatory steps. The first is to gain a full understanding of the business issues involved and the application to be modeled. The ultimate goal is to support business functions, so the better these are understood, the better job that can be done modeling. This means getting realistic data from the business units, not just hopes or broad generalities.

The ultimate goal is to support business functions, so the better these are understood, the better job that can be done modeling.

The other is to gain accurate data on the existing systems. TeamQuest® Manager collects and stores the performance data on the monitored systems in a common database that can then be used for modeling. Without accurate input data, you will never get an accurate model.

Once the above steps are completed, TeamQuest Model offers customers three methods of modeling such data:

- Single Server — With this method, each server is modeled individually, and the output of one server is then used as the input value for the next layer. There are several advantages to this method. To begin with, it is the simplest method, particularly for someone who is already experienced at modeling single-tier applications. It also breaks down the whole job into small, easily-managed steps. This method could also be useful for tweaking the configuration on a particular layer, before inserting it into a complete multi-layer model. The limitation of this method is that it makes it harder to see the dependencies between layers and the overall impact on response time.

- Enterprise Model — This method takes the entire infrastructure and models it at one time. It solves the limitations of the single server method; however, it is a complex undertaking. Enterprise modeling is best suited to applications which involve a limited number of servers, but it doesn't scale well for large installations.

Enterprise modeling is best suited to applications which involve a limited number of servers, but it doesn't scale well for large installations.

- Simplified Enterprise Model — Betwen these two extremes lies the simplified enterprise model, which is well-suited for large server farms, and works well for single- or multi-tiered applications. In this case, rather than modeling all the servers in the data center, a representative subset of those servers is selected. The modeling is done on this subset and the results are then extrapolated to the rest of the servers. This method is easier to set up than the full enterprise model, and scenarios run much faster than if running computations using all the servers.

Why Use TeamQuest?

Whichever method of modeling you are considering, there are, of course, the usual considerations of ease of use and accuracy which pertain to both single and multi-system modeling. TeamQuest Model automatically retrieves parameters such as CPU utilization, I/O usage, disk space utilization, TCP/IP, workloads and hardware configuration, making it easy to model the existing setup.

Then there is the matter of which types of multi-tiered modeling the product can do. No single type of modeling is best for all applications, so the capacity planner should have a range of options to choose from.

TeamQuest Model automatically retrieves parameters such as CPU utilization, I/O usage, disk space utilization, TCP/IP, workloads and hardware configuration, making it easy to model the existing setup.

Finally, you must examine whether the software will analyze and model the hardware and software that comprises the company's most critical IT infrastructure. Each layer is likely to be served by a different class of server. Looking, for example, at a simple three-tiered architecture, one might have a Sun SPARC Enterprise Server, feeding a Windows application server which users access through a Linux/Apache web server. The modeling software must be able to capture the peculiarities of each of these layers as well as their interdependencies.

Platform
Hewlett-Packard HP-UX on PA-RISC systems
Hewlett-Packard HP-UX on Itanium systems
IBM AIX on POWER systems
Microsoft Windows Server on Itanium systems
Microsoft Windows Server on x64 systems
Microsoft Windows Server on x86 systems
Microsoft Windows on x64 systems
Microsoft Windows on x86 systems
Oracle Enterprise Linux on x64 systems
Oracle Enterprise Linux on x86 systems
Red Hat Enterprise Linux Server on Itanium systems
Red Hat Enterprise Linux Server on x64 systems
Red Hat Enterprise Linux Server on x86 systems
Sun Solaris on UltraSPARC systems V8 processor or higher
Sun Solaris on x64 systems
Sun Solaris on x86 systems
SuSE Linux Enterprise Server on Itanium systems
SuSE Linux Enterprise Server on x64 systems
SuSE Linux Enterprise Server on x86 systems
VMWare ESX server on x64 systems
VMWare ESX server on x86 systems

Table 1

To make this possible, TeamQuest Model accurately simulates a wide range of server platforms. See Table 1.

Using TeamQuest Model gives organizations the ability to see how their entire set of systems works together to deliver service to the end users. In doing so, it gives them a cost-effective method to tweak their existing systems for better performance as well as predict and prepare for future needs.

TeamQuest Corporation

www.teamquest.com

Americas

One TeamQuest Way
Clear Lake, Iowa 50428
USA
+1 641.357.2700
+1 800.551.8326
info@teamquest.com

Europe, Middle East and Africa

Box 1125
405 23 Gothenburg
Sweden
+46 (0)31 80 95 00
United Kingdom
+44 (0)1865 338031
Germany
+49 (0)69 6 77 33 466
emea@teamquest.com

Asia Pacific

Units 1001-4 10/F
China Merchants Bldg
152-155 Connaught Rd Central
Hong Kong, SAR
+852 3571-9950
asiapacific@teamquest.com