TeamQuest®

Advancements in Cross-platform Capacity Planning

Whether it's about maximizing throughput, minimizing response time, maximizing availability, or minimizing costs, a thorough capacity planning exercise requires the evaluation of alternative configurations. A change in configuration might be as simple as moving a heavily used file to a less active disk controller or adding a CPU. A good capacity planning tool makes it quick and easy to evaluate such alternatives.

One configuration alternative that deserves consideration is a change in processor architecture. TeamQuest's capacity planning tool, TeamQuest Model, has been enhanced to simplify predicting the affect on performance of using different processor types.

This paper shows how new capabilities in TeamQuest Model streamline the process of predicting the effect of cross-platform migrations, making it easy for you to evaluate alternative vendor configurations.

About the Author
David Burgart, Principal Engineer, has been with TeamQuest since its inception in 1991. Mr. Burgart specializes in TeamQuest's flagship capacity modeling technology and is one of the primary architects for the capabilities described in this paper.

TeamQuest Model is a capacity planning tool used to quickly evaluate alternative configurations in various situations:

- Planning a server consolidation project
- Budgeting for IT infrastructure needed to handle future workloads
- Searching for configurations that will reduce power or cooling requirements
- Predicting the affects of a new application rollout on production systems
- Creating a cost-effective disaster plan

When conducting a capacity planning exercise for situations such as those listed above, it is oftentimes necessary to evaluate and predict how a workload will perform after it has been migrated to new server configurations. TeamQuest Model has always been used for such cross-platform capacity planning, taking into account changes in I/O and CPU configuration; however, determining the relative power of different processor architectures was an exercise left to the user. TeamQuest Model simply supplied silos of CPU definitions by vendor, now called the "By Vendor Scale."

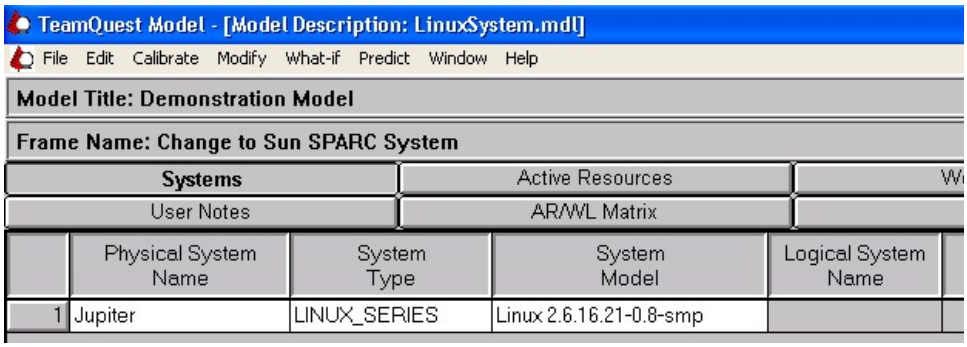
TeamQuest Model is predictive software that accounts for important non-linear behavior that is missed by simpler trend analysis capacity planning techniques. It can perform not only discrete event simulation, but also analytic modeling, which is simpler and less time-consuming, yet capable of producing very accurate results.

Now TeamQuest Model provides relative CPU performance measures that work across vendor platforms. Available right out of the box and based on the SPEC 2006 benchmark, TeamQuest Model's new "Common Scale" compares and normalizes performance across vendor CPUs, and the user interface makes it easy to incorporate them into its predictions.

If, for example, you are planning to consolidate workloads, you can easily evaluate options by modeling your consolidated workloads on alternative platforms using the Common Scale. These cross-platform predictions are quick and easy and no longer require you to research relative performance measures.

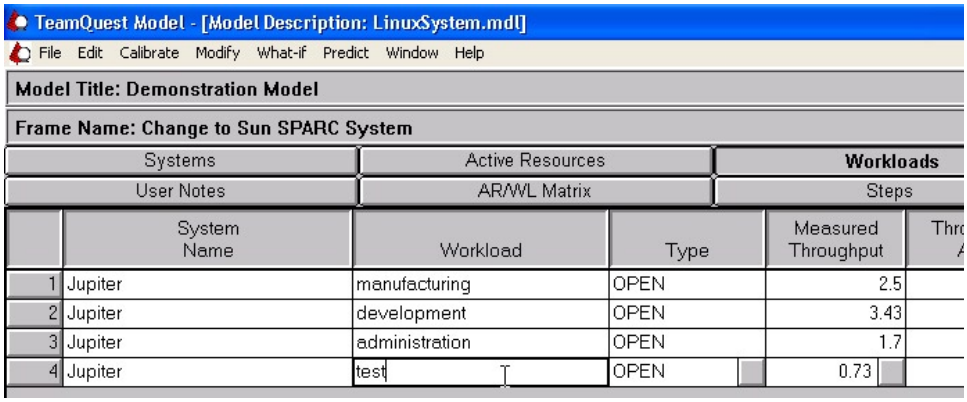
Step-by-step example

In this example, we will show how to predict performance when moving an entire systems workload from one architecture to another. (If you watch the **Vidcast**, we show a second example where just one application is moved from one architecture to another.)



Starting from a Linux System Called Jupiter

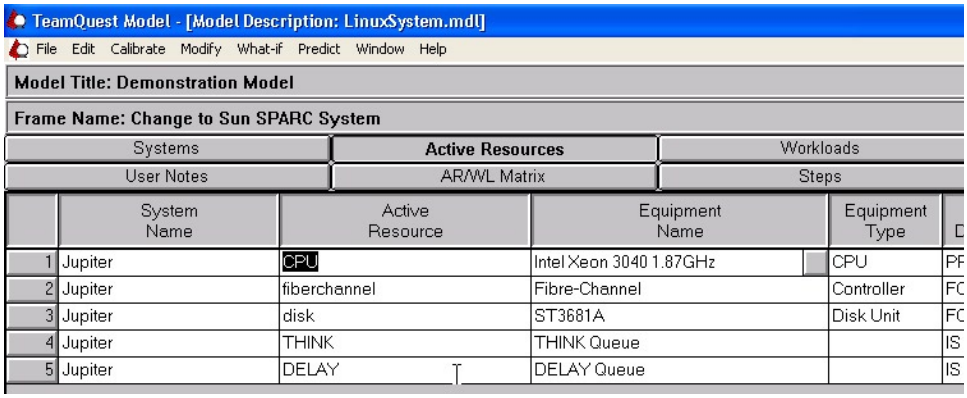
The window on the left shows a Linux system called Jupiter. What if we had a need to predict the performance of the same workloads on a particular Solaris SPARC system instead? The new Common Scale within TeamQuest Model makes such a model change much simpler.



Jupiter Runs Four Workloads

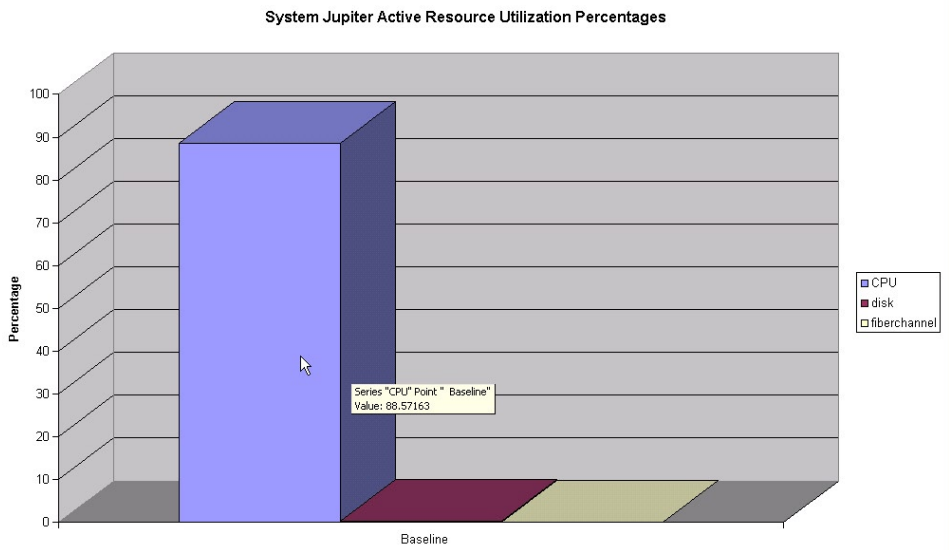
By pressing the Workloads button, we can see that this system is running four workloads:

1. manufacturing
2. development
3. administration
4. test



Jupiter Has an Intel Xeon 3040 CPU

By pressing the Active Resources button, we can see that we are currently running on an Intel Xeon 3040 CPU.



Jupiter Has Almost 90% CPU Utilization

Solving the model to check on the beginning state for resource utilization we see this situation.

TeamQuest Model - [Model Reports: LinuxSystem.mdl]

File Edit Calibrate Modify What-if Predict Window Help

Model Title: Demonstration Model

Frame Name: Change to Sun SPARC System

Step Name: Baseline **Solved at: 13:58:04.00 2008/07/**

Principal Results | AR Statistics | PR Statistics

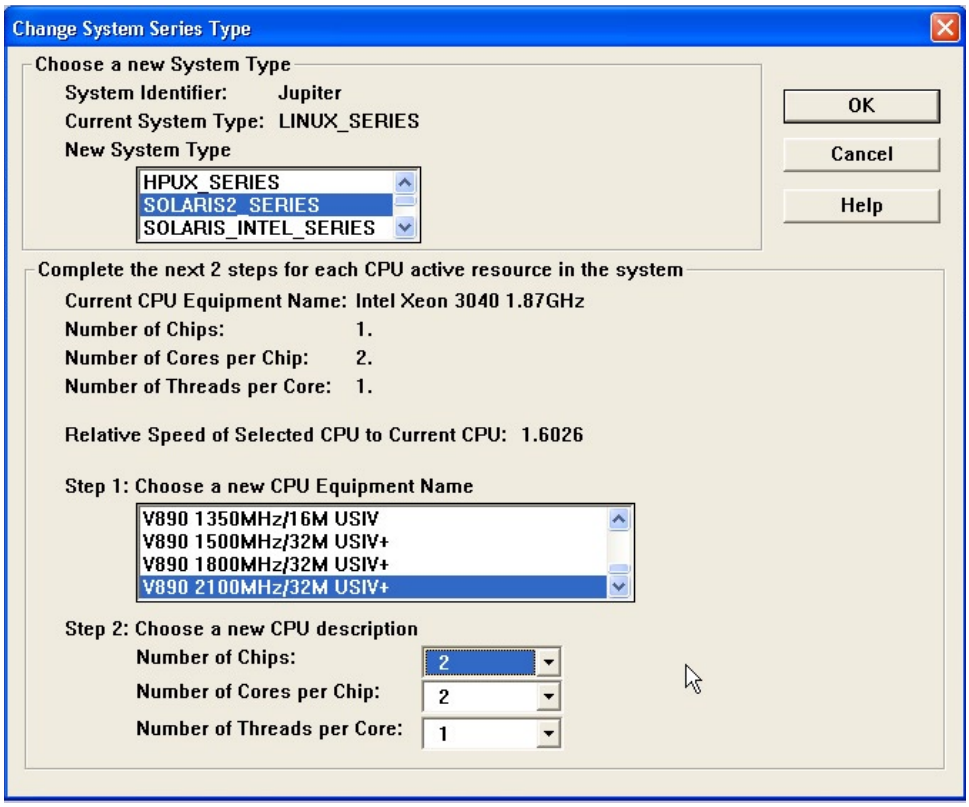
WL by AR Statistics | WL by PR Statistics | Measured vs Modeled

	System Name	Workload	Throughput	Response	Population	Wait for Passive R	Occupy Passive R
1	Jupiter	manufacturing	2.5	0.57754	78.002	0.	0.
2	Jupiter	development	3.43	1.9642	111.775	0.	0.
3	Jupiter	administration	1.7	0.004339	103.027	0.	0.
4	Jupiter	test	0.73	0.17106	22.48	0.	0.

Development Response Time is 1.96 Seconds

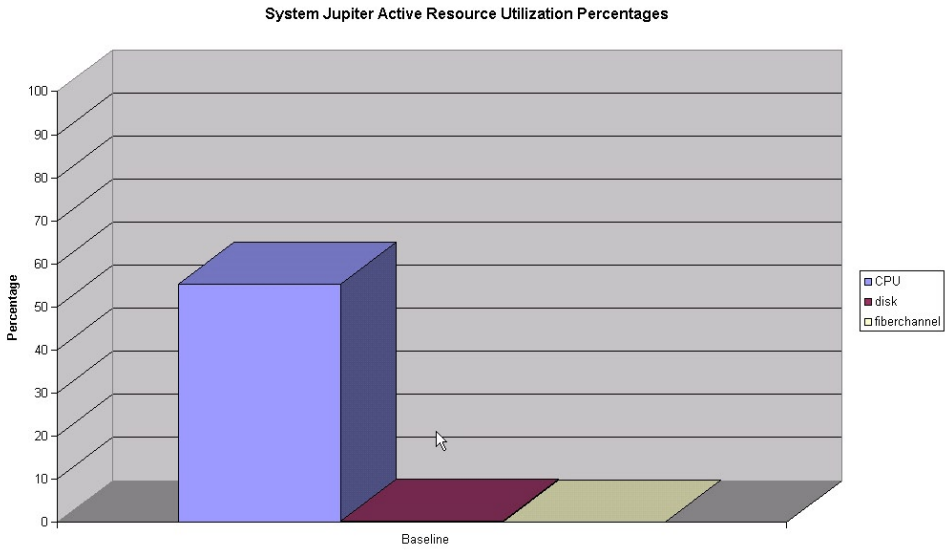
And response time for the development workload it is 1.96 seconds.

Now let's predict what performance will be like on a different configuration. To change the model description for a new system type, you select "Change to a Different System Type" under the "What-if" menu. This will cause the Change System Series Type dialog to be displayed.



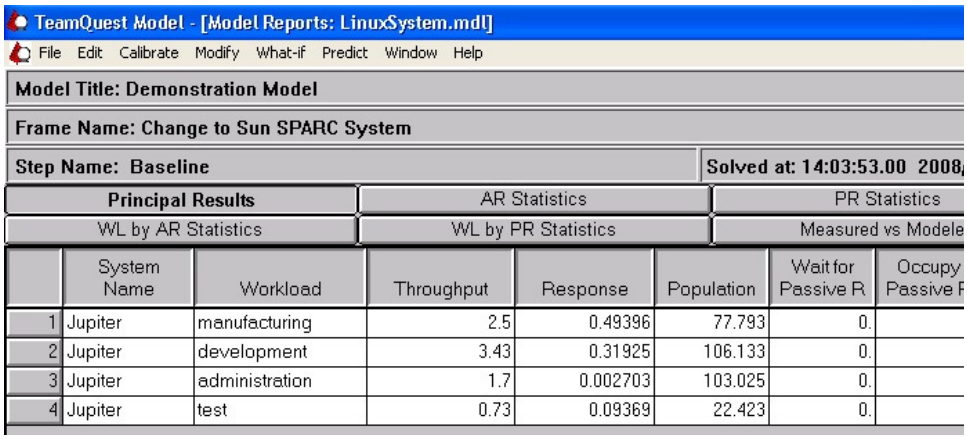
Change System to Solaris on a V890 with Two CPUs

In this example, we choose a Solaris system type and a V890 Sun Fire Server. TeamQuest Model preloads the CPU description with one chip, two cores per chip, and two threads per core, but we will add a second CPU, changing the number of chips to two. Notice the ease with which you can switch from a single threaded CPU to a multithreaded CPU. TeamQuest Model takes the nuances of multithreading into account when making such a switch.



We Now Have Respectable CPU Headroom (55% Utilization)

After re-solving the model, here are the results.



TeamQuest Model - [Model Reports: LinuxSystem.mdl]

File Edit Calibrate Modify What-if Predict Window Help

Model Title: Demonstration Model

Frame Name: Change to Sun SPARC System

Step Name: Baseline **Solved at: 14:03:53.00 2008/**

Principal Results | AR Statistics | PR Statistics

WL by AR Statistics | WL by PR Statistics | Measured vs Modele

	System Name	Workload	Throughput	Response	Population	Wait for Passive R	Occupy Passive F
1	Jupiter	manufacturing	2.5	0.49396	77.793	0.	
2	Jupiter	development	3.43	0.31925	106.133	0.	
3	Jupiter	administration	1.7	0.002703	103.025	0.	
4	Jupiter	test	0.73	0.09369	22.423	0.	

Development Response Time is 0.32 Seconds

We now have respectable CPU headroom, and the response time for the development workload has improved, moving from 1.96 seconds to 0.32 seconds.

Conclusion

TeamQuest is perhaps best known for expertise and leadership in the capacity planning and analytic modeling fields. New capabilities in TeamQuest Model further advance capacity planning techniques that simplify cross-platform migrations, allowing users to easily evaluate alternative vendor configurations.

For a demonstration of these new capabilities, please watch Principle Engineer David Burgart in the vidcast, **“Cross-platform Capacity Planning.”**

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