TeamQuest®

Modeling Multi-Threaded Processors

Symmetric multi-processing (SMP) has been employed by computer makers for some time. Multiple processors are connected to a common memory pool and a combination of hardware and operating system functions permit work to be balanced across the entire unit. Each processor had a single "thread" that processed programming instructions.

Recently chip manufacturers added additional threads to the processors to further increase efficiency of the chips. The added capacity of multi-threaded processors is welcomed but they provide some challenges for the capacity planner. As work is added to a system, multi-threaded CPU cores perform differently from multiple single-threaded CPU cores in a symmetric multi-processing (SMP) environment. The meaning of per-process CPU time measurements depend upon chip technology, therefore measurement results and expected future performance may not be intuitively obvious any more.

This paper explains a conceptual architecture developed for modeling multi-threaded processors and the new functionality provided in TeamQuest Model to support multi-threaded processors. A modeling example using Linux on Intel chips will be provided to aid your understanding.

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.

Single-threaded Versus Multi-threaded Symmetric Multiprocessing

Let's begin by comparing older single-threaded SMP technology with newer multi-threaded technologies. In both cases, the CPU chip is the hardware component that provides the instruction processing capability. Within each chip there may be multiple CPU cores, and each core contains the complete functionality of what we used to consider a CPU.

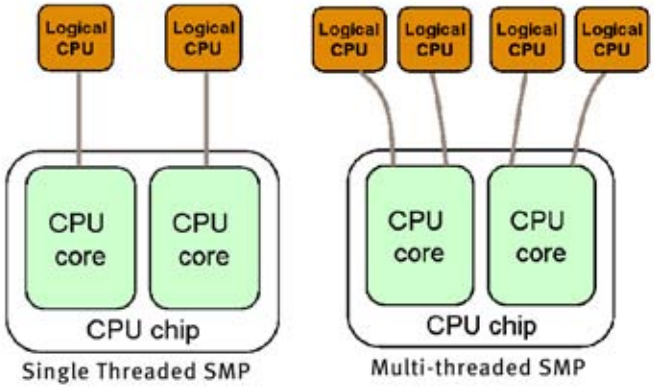


Figure 1

In traditional symmetric multiprocessing, each CPU core supports a single hardware instruction thread that interfaces with the operating system (diagram on left in Figure 1). When activating multi-threading, each core supports multiple hardware instruction threads, each interfacing with the operating system. Each hardware instruction thread is recognized by the operating system as a logical CPU.

Older SMP systems exhibited performance limitations as more CPU's were added to a configuration. For those of us familiar with the history of mainframe computers, we saw that each incremental processor added a lesser amount of additional capacity. In fact, one vendor, Amdahl Corporation, increased the computational power of the last two processors in their 12-way computer in order to overcome the SMP shortfall. These limitations resulted from hardware and operating system architectures designed to ensure data integrity through the use of various tactics such as signaling and locks. Over the years, all the major vendors have made significant improvements in this area. As a result, most SMP systems today have near linear performance scaling in the hardware and operating systems.

In a multiprocessing architecture, there are two approaches to providing additional processing power. Each additional core, bearing a single logical CPU, delivers a nearly equal quantity of CPU capacity. In most of today's architectures, this results in a commensurate increase in capacity when cores are added. The multi-threading option adds multiple threads to each core. Each thread adds some additional amount of CPU capacity. However, because these threads share the CPU core resources, the addition of a thread typically delivers only a portion of the capacity of a single-threaded core.

Examples of multi-threaded chips include Sun UltraSPARC T1 and T2, SPARC64 VII, Intel Xeon, Intel Itanium2, Intel Pentium 4, IBM POWER5 and IBM POWER6.

Performance Scaling in Multi-threaded Systems

When more threads are added to cores in multi-threaded systems, performance depends upon chip technologies. All deviate from a linear growth line graph once you get beyond the point where a single thread is active on each core and core resources are shared. Chip performance differences as seen during TeamQuest testing can be seen in Figure 2.

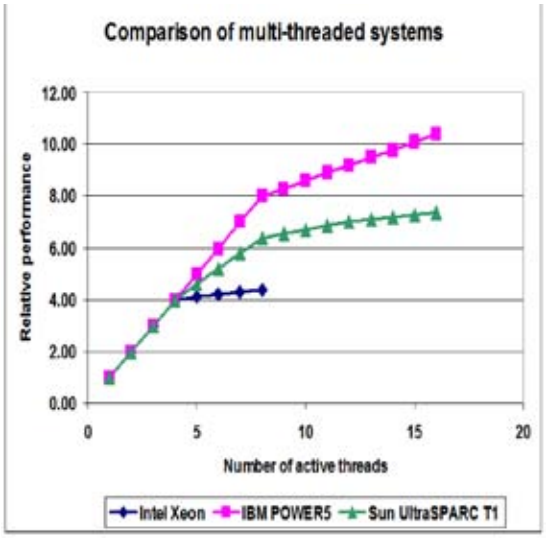


Figure 2

Intel Xeon (2 chips, 2 cores per chip, 2 threads per core) delivers minimal performance gain once four threads are exceeded.

Sun UltraSPARC T1 (1 chip, 4 cores per chip, 4 threads per core) shows a linear increase in performance up to four threads, slightly degraded performance when there are two threads per core active, and then only nominal gain after core sharing increases as more than two threads become active.

IBM POWER5 (4 chips, 2 cores per chip, 2 threads per core), shows linear gain up to eight threads (one per core) and then the gain per thread drops from that point forward. Where performance becomes non-linear, it is because more than one thread has become active on a CPU core.

Similarly, when you view transaction behavior with multi-threaded CPUs you see some interesting results. Figure 3 displays testing results from a transaction that executes a fixed number of instructions and a CPU core that supports four threads per core. If only one thread is active, each transaction will complete in one second. If two threads are active per core it will take 1.25 seconds for the same transaction. If three threads are active, each transaction will take about 1.6 seconds. If four threads are active, each transaction will take about 2.1 seconds. The behavior of transaction, therefore, depends on how many simultaneous logical CPU threads are active on a core. The results show that the best performance for a single transaction comes when there is only one CPU hardware thread is active on the core on which it is consuming resources.

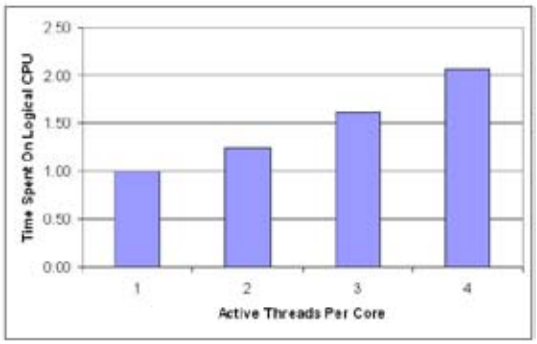


Figure 3

However, even though each logical CPU runs slower when multiple threads (logical CPUs) are active, there is a greater total capacity. It is the same whether two active threads on a core or more. The best throughput (transactions completed per second) occurs when all CPU hardware threads are active.

Revised Modeling Approach

When modeling multi-threaded environments the statistics are collected when multiple threads in the core are active, but the model tool requires as input the CPU service requirement when a single thread is active in the CPU. This will require some computation on the collected statistics.

The ideal statistics needed for precise modeling of multi-threaded environments are:

- Number of active threads per core
- Time spent on CPU
- Change in workload performance

Then we could use the formula here to compute a service time that reflects how a transaction would behave if a single thread was active.

$$\text{Service} = \text{time spent on CPU} \times \frac{\text{workload perf. change}}{\text{\# active threads per core}}$$

Unfortunately, vendors don't provide this level of detail in multi-threaded chips. To overcome the problem, we use available CPU statistics such as CPU seconds and logical CPU % busy. Then we solve a small model, with no user intervention, when the model is loaded to compute the single thread CPU service requirements.

TeamQuest Model has included the concept of a load-dependent server for some time, but it is now being put to a new use; it has been employed to model the CPU Active Resource Queue. An LDS server is where the rate of service varies depending on the queue length. When modeling CPU Active Resource Queue in a multi-threaded environment, LDS is activated and CPU Active Resource Type is displayed as "THREAD."

The LDS requires the use of speedup factors. Since there are diminishing returns when adding threads to a core, speedup factors are necessary to maintain accuracy when adding threads to cores within the LDS. TeamQuest Model contains default speedup factors for many popular CPUs. The default speedup factors that we provide come from a combination of industry literature and our own testing experience. If you have performed application scalability testing in such environments and your application scales differently, Model provides a way for you to change the default values to better reflect your environment.

The TeamQuest Model GUI now has several changes which make it easy to model today's multi-threading architectures. Instead of only being able to define CPU hardware using a CPU equipment name and the Number of CPUs, The hardware dialog box now expands the options to include a CPU equipment name, number of chips, number of cores per chip, and the number of threads per core. Most of the time TeamQuest Model knows the details for commonly used chips.

Speedup factors are also easy to adjust using the new GUI. A dialog box appears with the default speedup factor for the System Type selected and the number of threads per core that was specified. This is where you can adjust the default speedup factors with data derived from your own application tests or other credible sources.

On Windows, Linux, HP-UX, and VMware platforms with Intel hyper-threaded CPU hardware, you can control whether multi-threading is on or off at the CPU or hardware level. The Physical CPU Settings dialog box (accessed via the Active Resources spreadsheet by pressing the button in the Number of Servers column) allows you to control hyper-threading on or off just as you can with the real hardware. The hardware is capable of two threads per core, but if you set hyper-threading to "Off" the resulting number of servers per core will be reported as one.

The IBM AIX platform also has controls to turn Simultaneous Multi-threading on or off, but it is controlled at the Logical CPU level, not the Physical CPU or hardware level. Press the button in the logical CPU Number of Servers column and you will see the Logical CPU Settings dialog box. Here you can control the number of virtual CPUs that are assigned to this logical system and whether multi-threading is on or off at the logical system.

Sun Solaris platforms do not have any controls to turn multi-threading on or off. It is always on.

TeamQuest Model Reporting Changes

In a traditional single-threaded SMP CPU architecture, the Active Resource statistic Service Time is always equal to the Effective Service Time. Therefore, when there are not any other bottlenecks in the system the CPU utilization will grow at a linear rate.

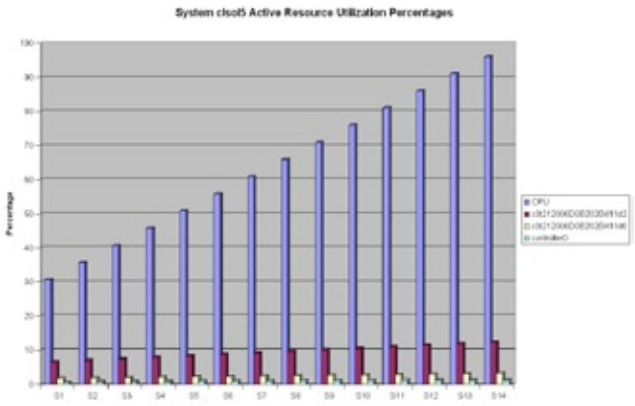


Figure 4A: Single-threaded architecture

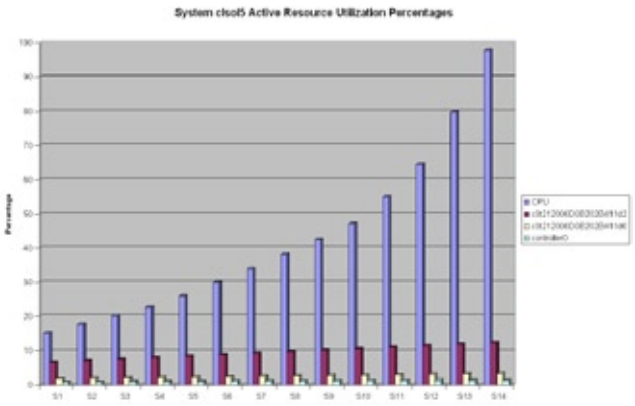


Figure 4B: Multi-threaded architecture

This is no longer true with a multi-threading CPU architecture. When the CPU has more than a minimal utilization, the Effective Service Time will sometimes be greater than the Service Time. Figure 4B shows how multi-threading deviates from a traditional SMP straight-line trend line (Figure 4A).

In the case of this Solaris system using a Sun UltraSPARC T1 processor with four threads per core, the service time is indeed less than the effective service time. At low utilizations when there is only one thread per core active, however, it is likely that the service time will equal the effective service time and the utilization growth will be linear until more than one thread per core is active. But as the processor gets busier and more threads become active within the core, effective service will get progressively larger than the service time and the utilization growth will be larger than linear.

A Simple Example

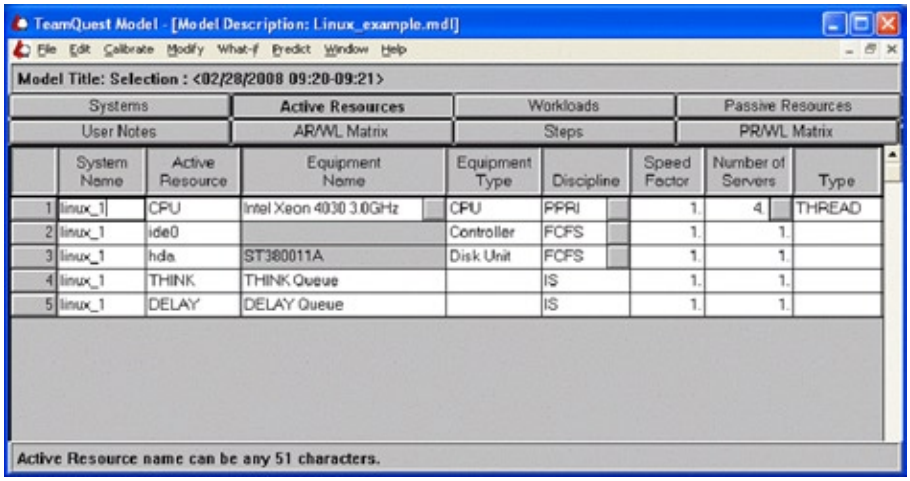


Figure 5

To show the power of the new modeling capabilities of TeamQuest Model, we will use a Linux system running a Xeon CPU with 2 chips, 2 cores per chip and 2 threads per core for a total of eight logical CPUs. We first model this system with multi-threading is turned off. This model will act like an SMP processor with four servers.

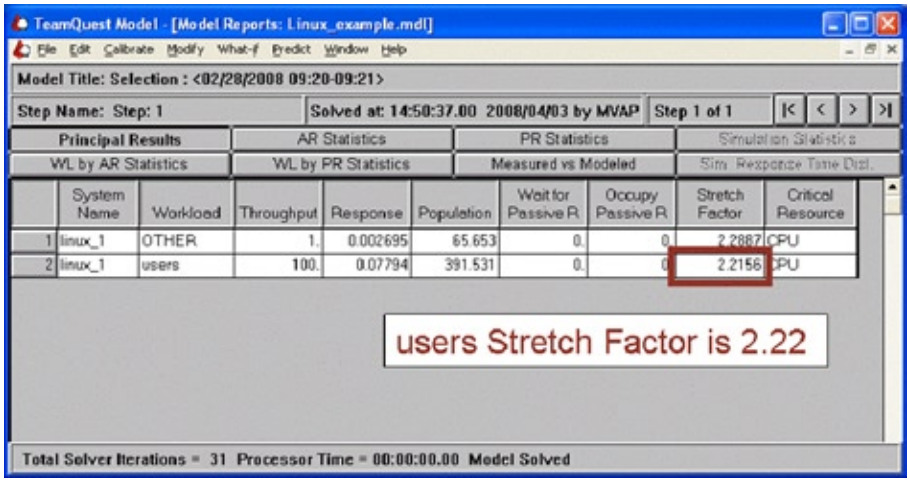


Figure 6

When this model is solved with multi-threading turned off, the stretch factor for a mission critical workload is calculated to be 2.2. Stretch factor is an industry-accepted way to portray normalized response time delays. As a rule-of-thumb, it makes sense to consider changes whenever stretch factor is greater two.

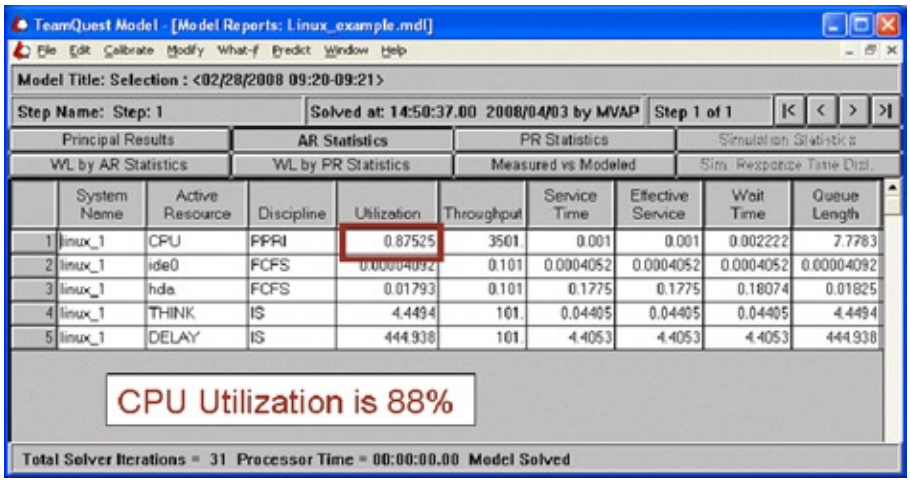


Figure 7

CPU Utilization is calculated to be 88%.

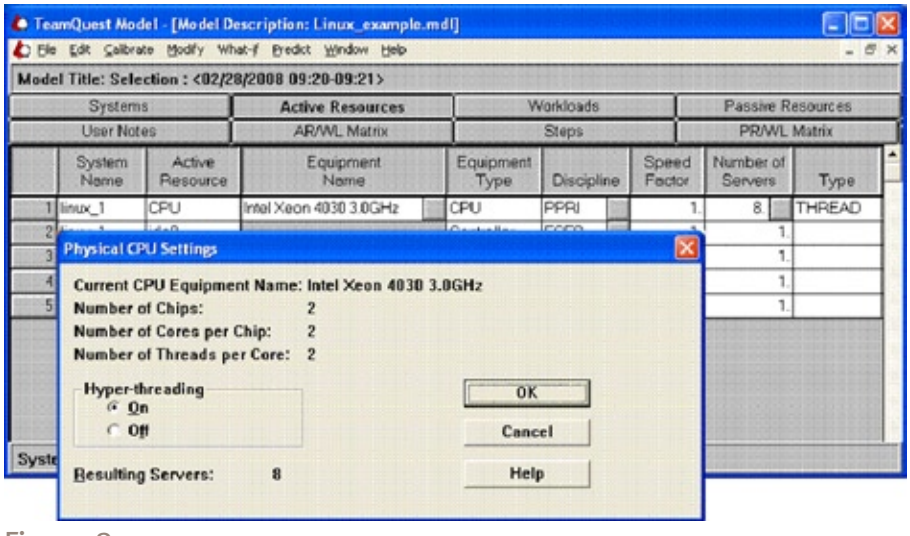
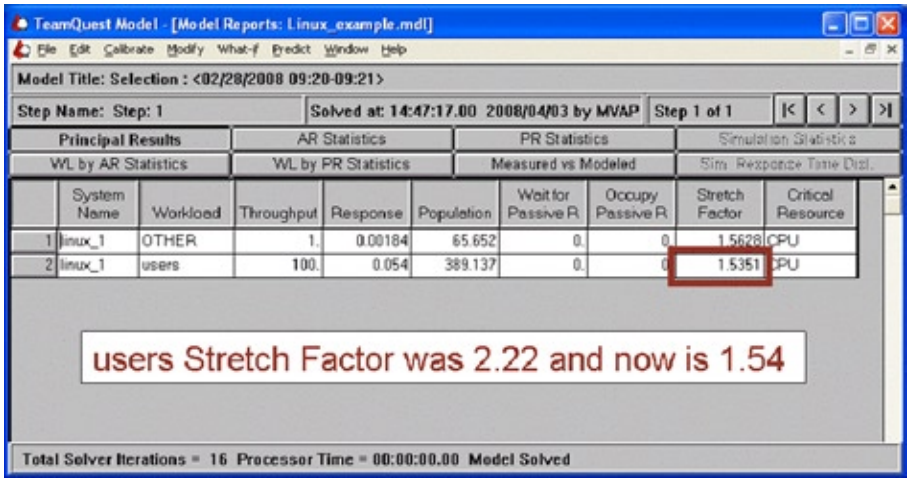


Figure 8

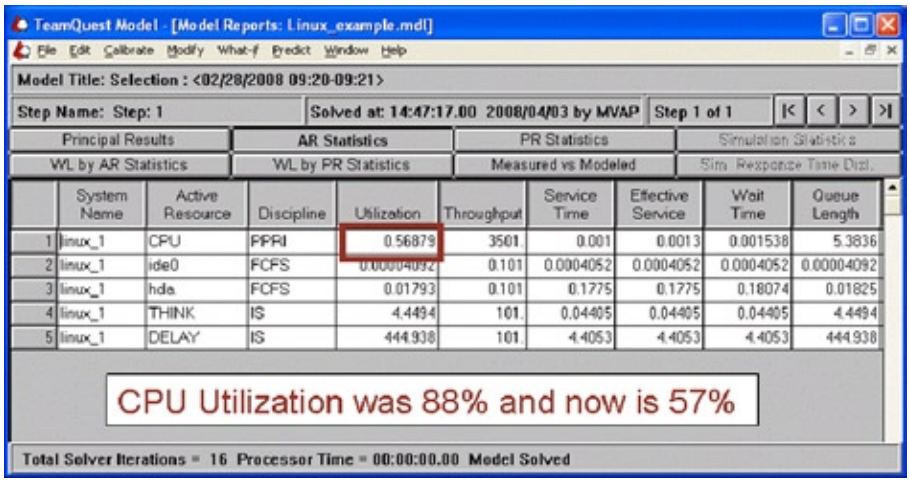
Now we are going to change the Intel CPU so that hyper-threading is on, resulting in the number of servers increasing from four to eight. As previously mentioned, we will press the button in the Number of Servers column for the physical CPU. The Physical CPU Settings dialog appears.

We chose the “On” radio button in the Hyper-threading group box and press OK. The Number of servers is now eight. Now let’s solve this model and see how the results change.



The stretch factor drops approximately 1.54 compared to 2.22 when hyper-threading was turned off.

Figure 9



Similarly, CPU utilization when multi-threading is turned on has reduced to approximately 57% from 88% when hyper-threading was turned off.

Figure 10

As you can see, this application benefits from operating in a multi-threaded environment. CPU utilization was reduced dramatically, making headroom for more work and extending the life of the asset.

Summary

As you can see, there are substantial differences between single-threaded and multi-threaded architectures. In order to deal with the changes, TeamQuest has incorporated a number of changes to facilitate your capacity planning activities when employing modeling techniques for multi-threaded architectures. The new capabilities combined with the easy-to-use TeamQuest Model interface makes it easy to predict application and service performance on the new multi-threaded architectures.

You need to be cognizant of the differences between the single- and multi-threaded technologies and understand their impacts on your applications and services. Armed with that information and TeamQuest Model's new capabilities, you will be able to easily predict future performance, no matter what technology you choose.

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