How Much Bigger, Better and Faster?

Dave Abbott, Corporate Computing and Networks, *Mike Timmerman*, Professional Services, and *Doug Wiedder*, Benchmarking Services, Cray Research, Inc., Eagan, Minnesota

ABSTRACT: *How much bigger, better and faster? This question is often asked when Data Centers begin to plan. The problem facing most Data Center Management is the quantification of resource usage and service levels for input into the development of the requirements for future hardware and software acquisitions or reallocations.*

This paper will deal with three aspects of determining resource requirements. First, what types of data needs to be collected for system evaluation. Second, how to classify workloads and usage patterns. Finally, the use of system modeling to predict the effects of changes in hardware and software configurations or the reallocation of workloads on system utilization and service levels.

1 Introduction

How much bigger, better, and faster do our data center computing resources need to be to meet our users demands while still staying within our budget? This question is often asked by data center management but seldom answered with any degree of certainty, until now. Cray Research's internal data center shares many of the same problems of other data centers, along with a few unique to the role of providing the development environment for a computer vendor. Almost all data centers share the common challenge of best matching resources with their users needs at the lowest cost. By quantifying the current resource usage and expected service levels, the data center manager is better able to develop requirements for future hardware and software acquisitions.

The question now becomes what would be "Biggest, best, and fastest" for our particular user environment? Bigger to one system might be more memory; to another, an aditional CPU, and to yet another, more disk space. Better could be more effective multitasking for one site or to another, multithreading. Faster for one system might be access to a faster PVP CPU or for another, many processors which may not be as fast. Typically the data center manager is looking for the best throughput and utilization of resources while providing an acceptable level of service to the end users. By collecting data on an environment and then modeling its workload, one can predict the outcome of adding various pieces of hardware and software to the current configuration and even predict how it will behave on another architecture. This becomes especially helpful when doing upgrade planning. Through this process the data center manager is better able to choose which acquisitions best fit with both the users' needs and budgets.

Historically, the reason these techniques were rarely employed was because of the overwhelming amount of data, effort, and complexity of the data gathering and modeling process. With the tools currently available, capacity planning has become more practical, accurate, and timely than ever before.

This paper will deal with three aspects of determining Cray system resource requirements:

- What data needs to be collected to provide the fundamental information to evaluate a system.
- How to classify and group both workloads and usage patterns.
- The use of system modeling to predict the effects of changes in hardware and/or software on system utilization and service levels.

To illustrate and further explain these issues, examples are shown from work completed in the Cray Research Corporate Computing and Networks (CCN) Data Center in Eagan, Minnesota.

2 Data Collection

In CCN we use three sets of utilities to gather data characterizing a systems performance.

The first is *sar*, the System Activity Monitor which accesses kernel table information. The *sar* routine is the standard UNICOS utility to monitor basic system performance and utilization. Table utilization, disk and channel load balancing, and memory and CPU utilization are easily monitored with *sar*.

The second utility used to collect data on system performance is Cray System Accounting (CSA). System accounting provides us with information that allows isolation of codes or individuals who impact the system in a detrimental way.

The third tool used is TeamQuest Baseline®, which is a third party performance analysis package from TeamQuest Corporation. The TeamQuest Baseline product provides a user friendly interface to *sar*, CSA, and many other forms of system data. The tool also provides a facility for historical analysis, long-term trend analysis, and management performance summaries. Baseline also keeps this information in a database making it convenient to access with the modeling software.

It is important to note that the time periods used to represent system activity should reflect the intended system usage. CCN systems are used heavily for interactive work, with most of this load coming between 08:00 and 17:00. This is the load data that we want to include in our analysis. For an environment that runs a continuous batch load, one or more full days worth of data should be considered.

3 Case Studies and Examples

The computing environments used as examples for analysis of the workloads and modeling are production computer resources in the Cray Research Corporate Computing and Network's (CCN) data center in Eagan, Minnesota.

USS (UNICOS Storage Server) is the main CCN file server running on a YMP 8E/8128-6. It currently holds about 3.9 terabytes of data for approximately 3000 users. Along with its file serving function, but at a lower priority, USS also has batch and interactive loads that use the majority of the available CPU cycles not used by file serving activities.

RAIN is the primary interactive production platform and is a C92/2128-2. This system supports compiler development as well as many other developmental efforts.

WIND is the marketing production platform and is a C98/8256-4. This system supports marketing production computing, benchmarking, and many external customers with both batch and interactive computer resources.

4 Workload Classification

Workloads are defined as the grouping of system users by common characteristics. The effect of grouping system users provides the ability to track growth by application, user group, account ID, or other usage parameters as desired. We use two selection methods for the classification of workloads:

- Functional groupings
- Organizational groupings.

Prior to making these groupings, we reduce the amount of data we deal with by using a reduction feature in the TeamQuest software that allows us to combine data points that meet certain criteria and reduce it to one data point. This reduction of data is used to combine into two groups; small processes owned by root and processes owned by users. In this we take processes with less than 1/10th of a second of CPU time, add them all up, and treat them as a single entity. These two small process groups typically account for about 90% of the process on the system but only about 2% of the total CPU utilization. Table 1 represents the reduction set definition we most commonly use in CCN.

The different workload classifications that we use on CCN production computing resources are summarized in Table 2 for organizational groupings and Table 3 for functional groupings.

4.1 Organizational Grouping

Organizational grouping involves the splitting of the workload on the system into groups that align them selves to organizational or departmental boundaries. We use the Fair Share Scheduler description, which is set up to put users into resource groups based on their department within the organization, to define the groupings of users. We break the users into six organizational groups:

- CCN
- Demos
- Hardware
- Marketing
- Software
- System

The System users are the group in which we put system administrators, and it is different from the system. We then break the six user groups into batch and interactive usage groups. The system functionality is grouped together into separate groups. Because much of the performance work is done first on USS, and this machine is a file server, we leave the DMF, Tape and NFS related groups as separate entities. We then can look at these independently.

Chart 1 shows the prime shift CPU utilization by groups on USS over the last few weeks of the summer and the first few weeks of the school year. Information grouped as this allows us to identify trends in workload based on external factors. We see that as the summer vacation season comes to a close and school starts, there is a definite increase in CPU utilization as employees return from vacation and resume a more normal schedule during the school year. We can see the decreased workloads on the weekends and the three day weekend the first part of September.

Chart 2 is a look at the CPU utilization of RAIN by groups during a normal workday. It is worthy to note the time of day that different groups get on the system, when batch jobs are run and when it is the best time to get CPU cycles. Although minimal, in this graph we can tell when traffic is heavy causing employees to get in late or what time lunch is by the loading of the system. We can also tell if the weather is really nice and everyone skips out to enjoy the day.

4.2 Functional Grouping

Functional grouping involves the splitting of the workload on the system into groups based off the tasks being performed. For example, in the case of our file server USS, when we want to look at the performance of the file server function separate from the batch and interactive processes, we break the workload into many different functional groups. These groups are normally groups of daemons that provide a common or related functionality to the users. They include DMF, tapes, accounting, NFS, networking and NQS. Users are grouped in to two groups, internal and external users.

Chart 3 is a look at the functional usage of the File server USS. We use this type of representation of work load to understand the resource requirements of providing different types of functionality to users. It is interesting to note the minimal system resources needed to run the file serving functions on USS. Due to the small amounts of time generated by the different functional areas we tend to use this type of grouping for performance problem isolation or the scheduling of system maintenance runs. We use organizational groupings for most of our performance analysis.

5 System Modeling

Having established the functional groups and workload characteristics, we next need to analyze the system for any performance problems that may need to be taken into consideration when preparing the system model. This phase consists of evaluating the following:

- Individual disks for excessive queuing and utilization.
- Disks in daisy chained configurations for queuing and excessive utilization.
- High memory utilization determined by the application environment.
- Swap activity and its characteristics.
- Voluntary and involuntary Job wait times.

Next the analyst has determined to what level of detail each active resource needs to be taken in the model. The active resources include memory, think time for interactive workloads, voluntary delays (such as tape mount times) for batch runs, and CPU's. A variable number of queues are used to model input/output subsystems. For example, for a disk subsystem may include an MUXIOP queue, EIOP queue, and a number of disks queues. This will account for any of the performance issues mentioned above. At this point a calibrated base system model can be constructed using analytical modeling techniques. This base model is done to insure that data collected for input to the model can accurately reproduce the existing environment. Table 4 represents the calibrated modeling results from the C90. The items reported are used in the determination of a calibrated system model. These statistics include the following:

- Population: The number of concurrently active users in a workload group.
- Throughput: The rate at which active users are serviced at an active resource. Also the rate at which transactions/jobs are completed.
- Response times: The elapsed time of a transaction/job.
- Utilization: The fraction of time an active resource spends servicing transactions/jobs.

Having the calibrated base system model completed, the next phase will be to try to answer such questions as:

- What would the current workload look like on our proposed J90 and T90 configurations?
- How much latent work will be appear on the new systems?
- What impact will the new systems have on (Job throughput, response time, etc.)?
- What will be the effect of the latent work on the system utilization?

With confidence in the calibrated system model, one can apply the characteristics of the proposed system configurations of the J90 and T90. Table 5. shows the CPU utilization changes for the J90 and T90 configurations. Table 6 gives throughput changes for both configurations and Table 7 gives projected changes in response time.

It is clear that the computer system characteristics do not follow a linear pattern. Each workload sees differing effects from the overall system change. By using system modeling the analyst is able to reveal these and other potential differences.

It is worth noting that while some workloads saw Process Throughput rates drop by as much as 50% when migrated to the J90, others such as interactive users were hardly impacted by the change. This as a result of the low CPU demand by the interactive users and also a result of the ability of the J90 to maintain a higher number of concurrent users due to the 16 CPU's. Examining the throughput changes for the T90 configuration reveals throughput increases for the Batch workloads as a result of latent work now being processed. The T90 result further shows negligible changes in interactive throughput. This again is a result of the low CPU demand by the interactive users.

These results clearly show the need for system modeling when making system projections. Since each workload class has its own characteristics when consuming CPU, I/O, and Memory resources, the effects on the resources and on each other can only be viewed using a modeling process.

5.1 Modeling summary

The results of the modeling can be summarized in Table 5. Table 5 has all of the selected workloads as shown Tables 6 and 7, but shows their expected usage for the proposed systems. It also shows that the CPU utilization for the J90 configuration would be 98.8 % indicating a saturated CPU situation. The T90 configuration shows a 86.3 % CPU utilization which includes

the additional work being processed as a result of the latent work.

6 Summary

How much bigger, better and faster? With proper data collection, workload classification and system modeling the answer to this question is available to data center management. This information can then be used to make business decisions that are prudent for the operation of the business and that take into account the effect on the productivity of the users of the computing environment.

Table 1: Reduction Set Definition

Set	Definition
Small_CPU_Root	totcpu < 0.1 AND uid = 0
Small_CPU_Users	totcpu < 0.1

Table 2: Organization Workload Definition

Workload	Definition
Idle: Idle Procs	command = "idle" && uid = 0
Small: CPU_Root	redname == "Small_CPU_Root"
Small: CPU_User	redname == "Small_CPU_Users"
Sys: Tapes	uid=0 && (command = "tpdaemon" || command = "stknet" || command = "avrproc" || command = "clsfile" || command = "fesreq" || command = "flush" || command = "openfile" || command = "opentdt" || command = "proceot" || command = "proceov" || command = "readerr" || command = "readvol" || command = "scratch" || command = "tppos" || command = "writeerr" || command = "writevol" || command = "rtidaemo")
Sys: NFS	uid=0 && (command="nfsd" || command="cnfsd" || command="mountd" || command="biod" || command="automoun")
Sys: DMF	uid=0 && (command="fsdaemon" || command="dmaudit" || command="dmconfig" || command="dmctcmpc" || command="dmctrbld" || command="dmdaemon" || command="dmdalter" || command="dmdbase" || command="dmdbval" || command="dmdebug" || command="dmdelete" || command="dmdidle" || command="dmdjournal" || command="dmdstat" || command="dmdstop" || command="dmdtext" || command="dmdump" || command="dmfill" || command="dmfree" || command="dmhit" || command="dminst" || command="dmmctl" || command="dmstamsp" || command="dmtpget" || command="dmtpmerge" || command="dmtpmsp" || command="dmtpput" || command="dmtpread" || command="dmtpsave" || command="dmvdbgen")
Sys: Daemons	uid = 0
Sys: Operator	login = "operator" || login = "backup"
Batch: Demos	resgrpid = 8367 and ttyname <> /tty.*/
Inter: Demos	resgrpid = 8367
Batch: CCN	resgrpid = 8354 and ttyname <> /tty.*/
Inter: CCN	resgrpid = 8354
Batch: Hardware	resgrpid = 8372 and ttyname <> "?"
Inter: Hardware	resgrpid = 8372
Batch: Marketing	resgrpid = 8381 and ttyname <> "?"
Inter: Marketing	resgrpid = 8381
Batch: Software	resgrpid = 8386 and ttyname <> /tty.*/
Inter: Software	resgrpid = 8386
Batch: System	resgrpid = 8389 and ttyname <> /tty.*/
Inter: System	resgrpid = 8389

Table 3: Functional Workload Definition

Workload	Definition
System Accounting	uid=0 && (command=/acct*/ || command=/csa*/ || command="chargefe" || command="devacct" || command="diskusg" || command="fwtmp" ||' command="getconfi" || command="setacid" || command="wtmpfix")
Operator	login = "operator" || login = "backup"
External User	login = /[cnv][0-9][0-9][0-9][0-9]/
Small_CPU_User	redname == "Small_CPU_Users"
Internal User	uid > 0
Dmf Daemons/Routines	uid=0 && (command="fsdaemon" || command="dmaudit" || command="dmconfig" || command="dmctcmpc" || command="dmctrbld" || command="dmdaemon" || command="dmdalter" || command="dmdbase" || command="dmdbval" || command="dmdebug" || command="dmdelete" || command="dmdidle" || command="dmdjournal" || command="dmdstat" || command="dmdstop" || command="dmdtext" || command="dmdump" || command="dmfill" || command="dmfree" || command="dmhit" || command="dminst" || command="dmmctl" || command="dmstamsp" || command="dmtpget" || command="dmtpmerge" || command="dmtpmsp" || command="dmtpput" || command="dmtpread" || command="dmtpsave" || command="dmvdbgen")
Tapes	uid=0 && (command = "tpdaemon" || command = "stknet" || command = "avrproc" || command = "clsfile" || command = "fesreq" || command = "flush" || command = "openfile" || command = "opentdt" || command = "proceot" || command = "proceov" || command = "readerr" || command = "readvol" || command = "scratch" || command = "tppos" || command = "writeerr" || command = "writevol" || command = "rtidaemo")
Small_CPU_Root	redname == "Small_CPU_Root"
Disk Quotas	uid=0 && (command="quadmin" || command="qudu" || command="quota" || command="quotamon")
Fair Share	uid=0 && (command=/shr.*/)
IdleProc	uid=0 && command = "idle"
USCP	uid=0 && (command="uscpcmd" || command="uscpcore" || command="uscpd" || command="uscpdevs" || command="uscpdump" || command="uscpfix" || command="uscplink" || command="uscpops" || command="uscpques" || command="uscpstat" || command="uscpstrs" || command="uscpterm" || command="uscptrace" || command="uscpxmon")
Monitoring	uid=0 && (command=/tq.*/ || command="logdaemo" || command="aird" || command="airping" || command="sar" || command="sadc" || command="crayperf" || command="netmon")
NFS	uid=0 && (command="nfsd" || command="cnfsd" || command="mountd" || command="biod" || command="automoun")
NQS	uid=0 && (command="nqsdaemo" || command="netdaemo" || command="qfdaemon")
Network Daemons	(uid=0 && (command="getty" || command="lpd" || command="portmap" || command="snmpd" || command="named" || command="sendmail" || command="gated" || command="inetd" || command="ftpd" || command="telnetd" || command="rshd" || command="rlogind" || command="rexecd" || command="uucpd" || command="comsat" || command="talkd" || command="ntalkd")) || (uid=12 && (command="fingerd" || command="tftpd"))
Security	uid=0 && (command="slogdemo")

Chart 1: USS CPU Utilization by Departments

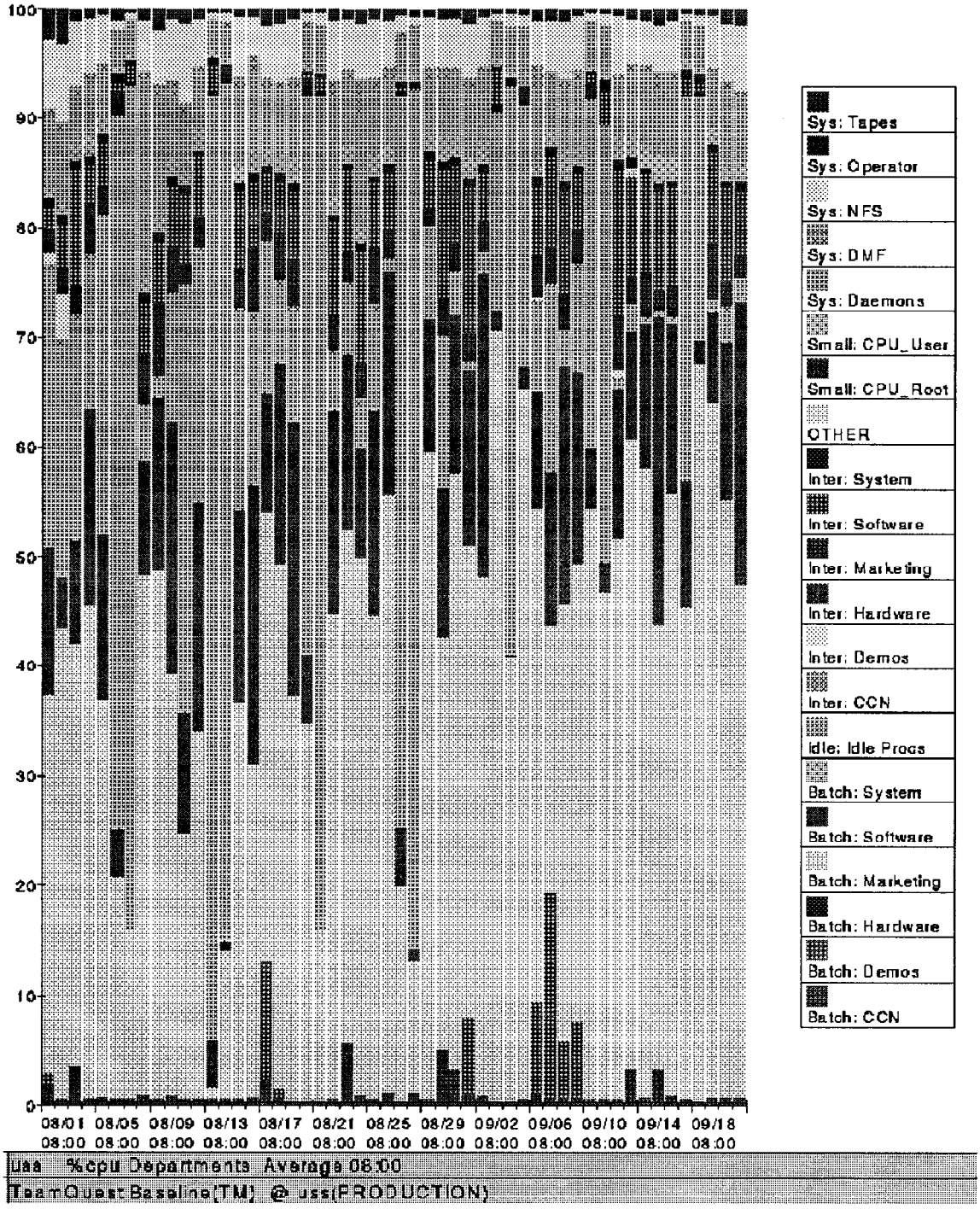

Chart 2: RAIN CPU Utilization by Departmnet

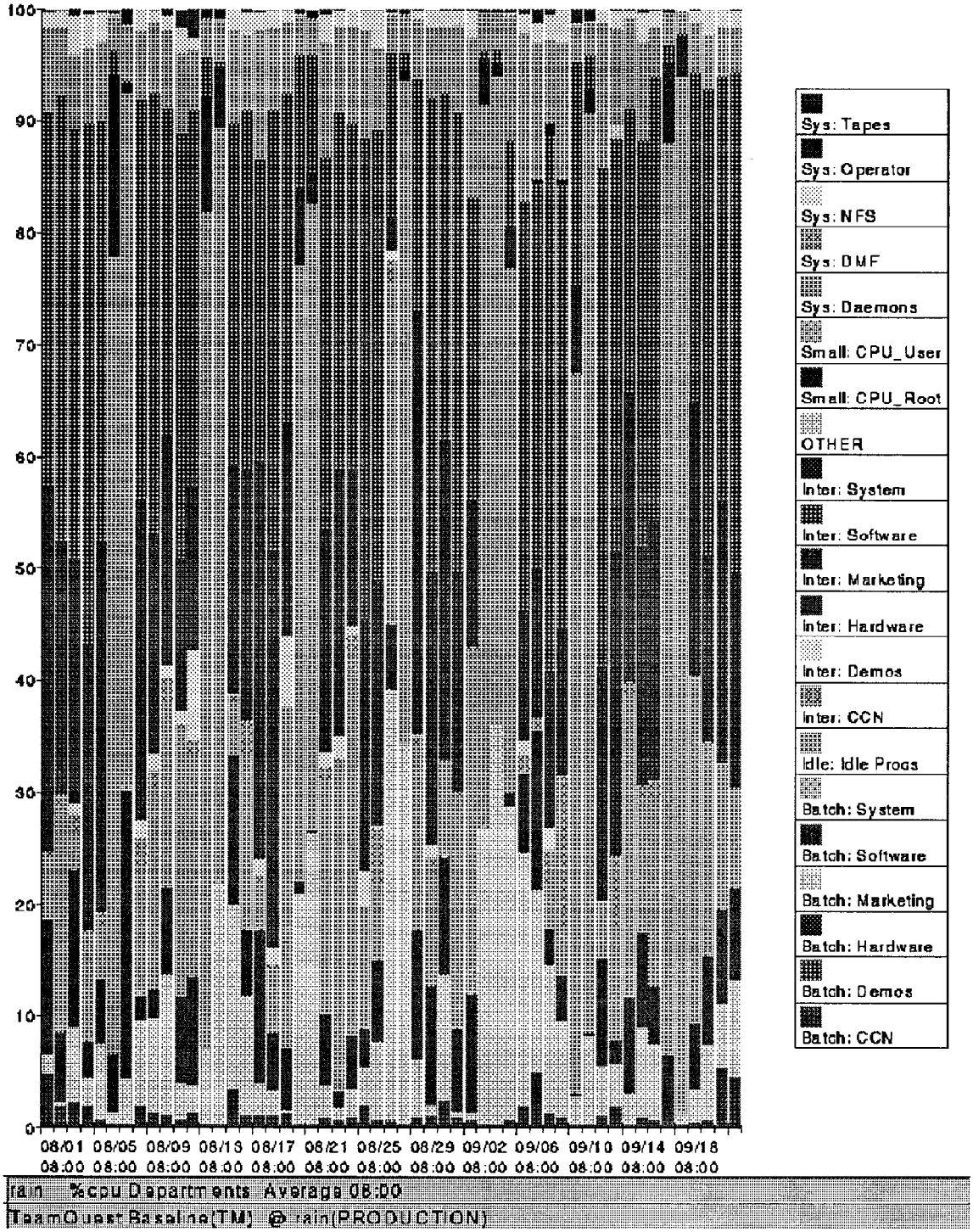

Chart 3: USS CPU Utilization by Function

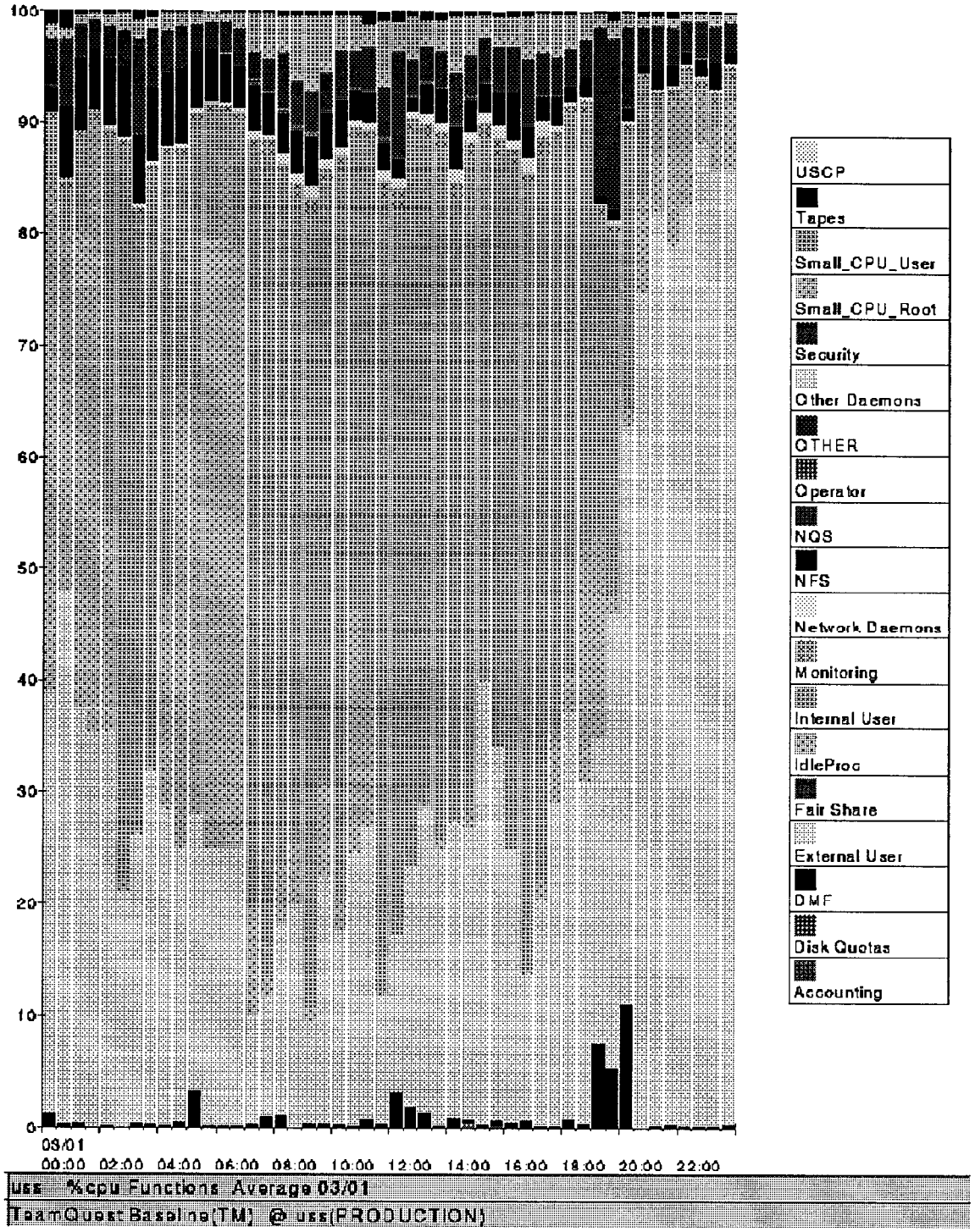

Table 4: Model Calibration Results

	Measured Population	Measured Throughput	Modeled Throughput	Measured Response	Modeled Response	Measured CPU %	Modeled CPU %
Batch Mktg	14.42	0.51	0.51	28.35	28.35	66.67	66.69
Interactive Mktg	189.30	9.12	9.12	20.76	20.76	16.05	16.05
Software	4.01	0.11	0.11	35.03	35.03	11.21	11.21
CCN/Hardware	0.47	0.02	0.02	23.85	23.85	0.74	0.74
System Utilities	323.40	6.59	6.58	49.11	49.12	3.47	3.47
Measured CPU %	98.14						
Modeled CPU %	98.16						

Table 5: Projected Cpu Utilization Changes

Batch Mktg	66.67%	54.67%	62.01%
Interactive Mktg	16.05%	26.23%	11.61%
Software	11.21%	11.28%	9.64%
CCN/Hardware	0.74%	0.90%	0.59%
System Utilities	3.47%	5.78%	2.50%
Total	**98.14%**	**98.86%**	**86.34%**

Table 6: Projected Throughput Changes

			J90		T90
	C90	J90	Delta	T90	Delta
Batch Mktg	1831	901	-50.8%	2364	29.1%
Interactive Mktg	32832	32191	-2.0%	32983	0.5%
Software	412	249	-39.6%	492	19.5%
CCN/Hardware	71	51	-27.2%	78	10.2%
System Utilities	23706	23670	-0.2%	23699	0.0%

Table 7: Projected Response Time Changes

			J90		T90
	C90	J90	Delta	T90	Delta
Batch Mktg	28.35	57.64	103.3%	21.95	-22.6%
Interactive Mktg	20.76	21.17	2.0%	20.66	-0.5%
Software	35.03	58.00	65.6%	29.34	-16.2%
CCN/Hardware	23.85	32.74	37.3%	21.64	-9.3%
System Utilities	49.11	49.19	0.2%	49.13	0.0%