Performance Modeling & Simulation Of Complex Systems

(A Systems Engineering Design & Analysis Approach)

Laverne Hall, Systems Engineer

Jet Propulsion Laboratory, MS: 168-414
4800 Oak Grove Drive
Pasadena, CA 91109-8099
Phone: (818)354-8310
FAX: (8 18)393-6962
E-mail: Laverne_Hall @IPLmail.jpl.nasa. gov

ABSTRACT

Modeling of the Multi-mission Image Processing System (MIPS) will be described as an example of the use of a modeling tool to design a distributed system that supports multiple application scenarios. This paper examines (a) modeling tool selection, capabilities, and operation (namely NETWORK 11.5 by CACI), (b) pointers for building or constructing a model and how the MIPS model was developed, (c) the importance of benchmarking or testing the performance of equipment/subsystems being considered for incorporation the design/architecture, (d) the essential step of model validation and/or calibration using the benchmark results, (e) sample simulation results from the MIPS model, and (f) how modeling and simulation analysis affected the MIPS design process by having a supportive and informative impact.

INTRODUCTION

There is a need for sophisticated, yet simple-to-use methods and tools for modeling the effects of processing workloads on today's distributed multi-node and multi-network data processing systcm architectures prior to their actual implementation. In support of system engineering and analysis activities, today's computer-based modeling tools allow evaluation of system performance characteristics for systems that are too complex to be modeled by more traditional analysis methods.

Modeling and simulation is conducted to specify and analyze the performance of distributed computer systems and local area network configurations. The models simulate total system operation combining the interactions of workload volumes, processing modules, and hardware elements and provide performance predictions for the complete system architecture. The objective is to study the effects of system loads and required performance levels on candidate system hardware configurations in order to develop a system design that will meet performance requirements.

Performance modeling is an iterative process involving computer-based system descriptions that are used to analyze and compare options for the system architecture and to estimate performance measures in areas such as: (1) CPU utilization, (2) network contention and data transfer efficiency, (3) software execution and resource utilizations, and (4) mass storage utilization/contention. The results are used to help determine an optimal design, to help verify that the system will meet its prerequisite performance requirements, and to aid in the formulation of derived requirements and component specifications.

Use of a model makes it possible to perform system trade-off analyses by removing or adding system hardware elements and software functions (*system reconfiguration analysis*). System modeling can also be used to aid in the planning and scheduling of system and/or network processing activities. The essential benefit of system modeling is to experiment with a computer-based model before incurring the risk and cost of committing to a proposed or modified system design. The objective is to identify any potential system problem areas and/or deficiencies which in turn gives guidance to direct or redirect engineering efforts to areas where further work is required.

TOOL SELECTION, 'CAPABILITIES, & OPERATION

It is important to select the best and most applicable tool(s) for the system environment to be modeled or simulated. But first, it is necessary to establish a set of tool selection criteria that are pertinent to your needs and that include prioritized items such as applicable tool capabilities (existing and future), overall costs, user interface features (user friendliness), and resource requirements (e.g. computer resources and tech-support personnel).

The MIPS performance modeling task team selected and uses the tool Network 11.5 by CACI Products Company. This computer-based tool can specify and analyze computer systems and local area networks by simulating total system operation between workload volumes, processing modules, and hardware elements. [Other competitive tools surveyed via on-site free trial and evaluation periods were Block-Oriented Network Simulator (BONeS) by Comdisco and Optimized Network (OPNET) Engineering Tool by MIL3.]

The modeling tool used should provide for a reasonably straightforward mapping of required and proposed system components into a model description that will be understood by the tool. This feature is inherent in Network 11.5. Figure 1 gives a pictorial representation of this concept of transforming proposed system architecture information into a modeling tool description for computer-based simulation. Network 11.5 allows for a "layered" approach to defining or describing the model, thus permitting more details to be included where and when needed.

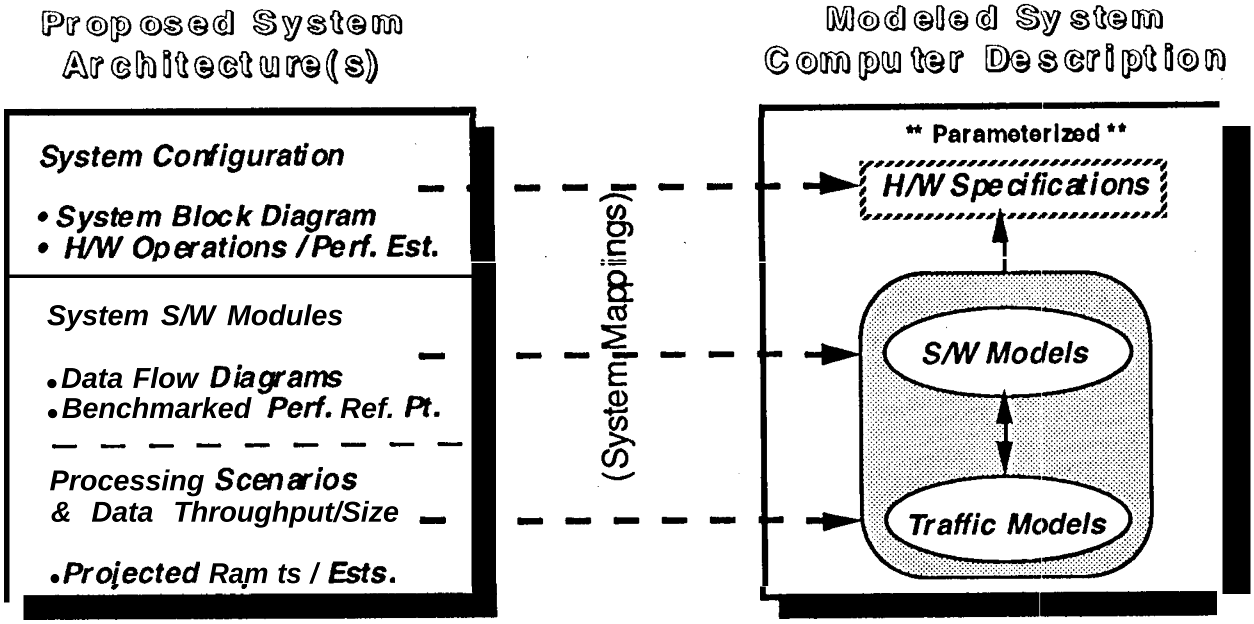

Figure 1. Mapping System Architecture Into Model Description

Network 11.5's user interface allows for graphical layouts and specification of the hardware elements that make-up the system under consideration. Details of parameter/instruction settings for 3 key kinds of hardware (processing elements, storage devices, and communications links / transfer devices) are input via interactive windows. In addition, processing modules with instructions, global parameters and functions, file specifications, and routing information can be defined via windows to reside and/or execute on these hardware elements. The data flow and processing within the simulation runs are controlled by event-driven scripted scenarios whose activation can be specified by statistical distributions, fixed/set times, ardor message triggered.

The accuracy of the simulation output depends heavily on the accuracy of its input. However, the results will give a good indication of system performance and of gross deficiencies or bottlenecks within the proposed architecture. The objective is to minimize simulation error in order to reduce the total margins that would be added onto the performance results due to inaccuracies in input estimates, human factors (unpredictable behavior) on system, and imposed system requirement margins.

The output from the Network 11.5 simulation tool includes various plots and detailed statistics files containing information about hardware utilizations and/or contentions (such as percent of time busy for CPUS and storage devices), software executions and interrupts, and message/data transfers.

Network 11.5 also provides a built-in graphical simulation animation feature that can be used to debug the simulation or for demonstration purposes. Also, in keeping with its graphical capabilities, Network 11.5 provides the ability to use pre-designed or custom-built icons to represent different hardware elements within the system being modeled. Last but not least, it also gives the option of building and executing models in a non-graphics or text mode which (a) can save on simulation time and resource utilizations and/or (b) can be used for remote access via modem.

Figure 2 depicts a subset of possible interactive windows used by Network 11.5 to specify system parameters and script operational sequences when setting-up a system model. Window 1 is the starting point showing a graphical high-level description of the topology of the system to be modeled. Window 2 shows the parameter and instruction specifications for a processing element which represents a CPU within a compute-server. Window 3 gives a list of software modules that are allowed to run or execute on the selected CPU. Window 4 shows the functional flow and requirements for a given software module.

POINTERS FOR MODEL CONSTRUCTION

The following guidelines are provided to help system modelers focus on important ingredients for constructing a computer network system model when using a modeling tool:

- Use a computer-based modeling tool to describe and analyze a system when it becomes too complex to hand calculate system performance overtime.

- Distribute the labor effort according to the following time categories in order to minimize reworking the model.

    - ≈60% effort to formulate through-put volumes and performance specifications for I/O traffic, H/W devices, and S/W processing functions, and to decide on the level of model detail and relevant contents. (Note: If the system to be modeled is complex and not well documented or understood, the collection of model input data on system operations can be extremely time consuming. However, the process of defining required model inputs can help guide/force system decisions and understanding.)

- ≈40% effort to build and test/debug the model, run final simulations, and analyze results.

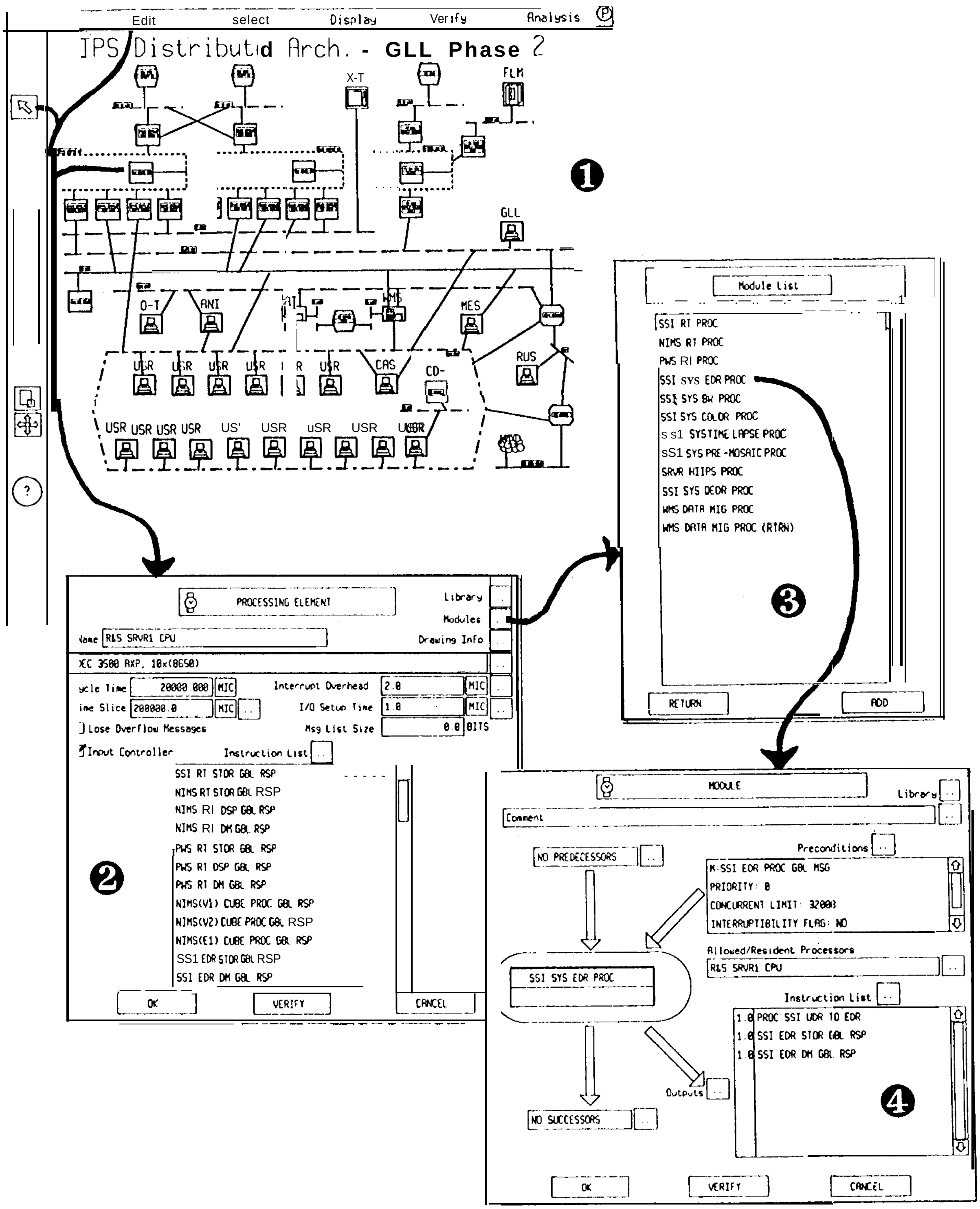

Figure 2. Sample Network 11.5 Interactive Graphics Windows

- Predetermine, as accurately as possible, model input estimates for the following system elements:

  - Device specifications such as CPU speed, disk I/O rates, link bandwidths, and overhead values.

  - Software CPU execution times for such items as mission procedures, database management functions, and network file service routines.

  - System communications traffic or 1/0 sizes and rates for data items to be processed or routed through the system.

  - Candidate system architecture(s) consisting of hardware configuration(s) and operating scenarios.

- Group, as a single entity, major system components (H/W or S/W) whenever a reasonable high-level representation of those components can be achieved which gives the same net operational effect as detailed representations. (This is an important ingredient for efficient modeling in order to minimize simulation execution times and costs.)

- Exclude portions of the system configuration that contribute little to the total system performance.

- Other Efficiency Tips:

  - Use optimization features and suggestions that are specific to the tool being used. (The objective is to limit the amount of work required by the simulator.)

  - Where feasible, separate large complex systems being modeled into multiple smaller models at clean interface points.

- Add margins to simulation results to accommodate for deficiencies in above-mentioned estimates, human factors (such as error and/or behavior predictions), and imposed system requirement margins.

Additional modeling efforts should be performed in order to establish the confidence level of the produced simulation results. These additional efforts should include: (a) calibration of model using a ***similar*** benchmarked system and (b) system reconfiguration simulations to test for the most efficient design. Iterative detailed model review meetings and model refinements will also improve the quality of the results. However, model refinements should be limited in that too much meticulous detail typically will not give a worthwhile percentage improvement to an already fairly well-estimated system.

It is important to remember that ***NO*** simulation package is capable of creating miracles. It will not improve upon poor estimates of its inputs for software timings, traffic loads, or hardware performance. This modeling characteristic leads into the discussions in the next two sections on using benchmarks and validation/calibration to strengthen confidence in estimates used.

IMPORTANCE OF’ BENCHMARKING

Emphasis should be placed on the importance of benchmarking (testing) the performance of components and subsystems of hardware and software being considered for incorporation into the system's architectural design. A benchmarking team should be identified and composed of personnel with the necessary skills to perform or complete the following activities:

(1) develop a test plan identifying system configuration and processing needs, schedule, and expected accomplishments,

(2) perform system administration functions for systern/software execution and data recording,

(3) possess familiarity with the software procedures being run in order to assure proper execution and results,

(4) possess familiarity with the hardware configuration and any on/off features, settings, or attachments that might affect the desired isolation effects.

For the MIPS, extensive benchmarking activities were performed for all major hardware subsystems (centralized computational servers, distributed workstations, mass and distributed storage, data catalog servers, film recording and hardcopy servers, and communications) and all major software data processing procedures (real-time, systematic, data management, interactive workstation). Benchmarking involved various methods from on-site testing of software and hardware to off-site testing at vendor test centers or facilities.

Note: Since the recorded performance timing of hundreds of MIPS software modules existed from the current system, the technique of using a conservatively derived scaling factor for projecting performance in the proposed system can be used to avoid the large task of re-testing all software on potentially changing proposed systems.

MODEL VALIDATION & CALIBRATION

The essential step of model validation and calibration using the benchmark results should be performed in order to build or enhance the confidence level in the modeled approach and of produced simulation results. The validation of model should be done using a benchmarked system similar to the one being modeled. This in turn will help determine boundaries for calibrating (refining) the simulation description and parametric inputs. The model validation and calibration exercise is done all in preparation for modeling and analyzing a final candidate design.

A simple model calibration and tool validation exercise was used for comparing benchmarked system results to similar modeled system results. This comparison was in turn used to help gauge the accuracy of the more detailed and complex model of the MIPS architecture design which is described in the next segment of this paper.

The model calibration and tool validation exercise used an image processing procedure typical of MIPS processing from the Voyager spacecraft era. This procedure is composed of a sequence of several software routines, each with its own required CPU timing and volumes of input and output data (1/0). The procedure was benchmarked on a candidate system which was used in the MIPS upgrade design. The procedure was also benchmarked on a system currently in use by MIPS, the VAX-8650. The model of this calibration exercise consisted of the following implementation and analysis concept:

(a) The resulting performance values from benchmarking the Voyager procedure on the current system were resealed for use as estimated or predicted software performance values for the modeled proposed system, along with its hardware performance parameter settings for the mainframe and disk storage cluster.

(b) The modeled (proposed) system and its 1/0 performance results were compared to the actual benchmarked system performance results for calibration of the MIPS model, validation of Network 11.5's performance prediction capabilities, and for substantiation of the modeling approach or methodology.

The results of this calibration/validation exercise confirmed that the tool and MIPS modeling approach were reasonably accurate. For instance, the overall elapsed time for completing a single stream of the modeled procedure was within 3.4% of the actual benchmarked system. Similar tests were run on the modeled system running multiple copies (8) of the Voyager procedure stream into the system for simultaneous execution and accessing of multiple disk drives. The modeled results for the multi-stream test-case were within 9.6% of the actual benchmarked results. Even though CPU performances estimates are fairly straightforward, the increased inaccuracies in simulation results were due to the lack of an accurate estimate for the nonlinear performance improvements of the disk cluster as the number of drives used is increased. This observation supports the inferred analysis that, when the storage configuration is modeled as a high-level cluster, a different I/O access rate (including overhead delays) must be derived for each disk cluster architecture.

SAMPLE MODEL & SIMULATION RESULTS

MIPS SIMULATION MODEL

The Multi'mission Image Processing System (MIPS) modeling and simulation work was conducted to specify and analyze the collective performance of the computer systems, storage devices, and local area networks that make-up the MIPS configuration.

Detailed modeling activities continue for the MIPS in the areas of upgrades and/or add-ens to the selected distributed design using Network 11.5. The system topology modeled is composed of distributed processing, storage, and communications operations. The major hardware components of the MIPS distributed design model (depicted in Figure 3, a Network 11.5 diagram) are listed below:

- Two DEC Alpha compute-servers (for Real-Time and Systematic processing) cross-sharing clusters of storage devices

- One FDDI LAN (Fiber Optic Local Area Network) ring connecting all compute-servers, user-stations, and other miscellaneous elements

- Twenty user-stations (18 of which are active, 2 reserved for film recording activities) performing various functions such as mission instrument processing, browsing, and quality control viewing of images

- One bridge-router connecting the FDDI and most Ethernet communications links

- One MIPS Local Ethernet interfacing office terminals and the animation subsystem to the rest of the system via the bridge router

- More Ethernet communications links (Operational Science Area LAN delivering telemetry data to the MIPS, Cluster Ethernet interfacing Alpha compute-sewers and the film recording/hardcopying subsystem, and R/T Ethernet connecting Alpha compute-servers and R/T displays)

- Two data management servers (Catalog Server and Working Mission Storage Server) sharing mass storage devices

• Onc cluster of office terminals batching systematic jobs into the system via the MIPS local Ethernet

• One cluster of Home Institution Image Processing System (HIIPS) remote sites at universities/etc. requesting network file serviced (NFS) data via the ILAN FDDI and NASA Science Internet

• One animation subsystem requesting processed time-lapsed NFS data.

The current model of the MIPS system design is too detailed for complete discussion in this paper. However, reference is made in order to give an idea of how well the tool handles the modeling of complex computer and network systems whose simulation results can aid in making better design decisions.

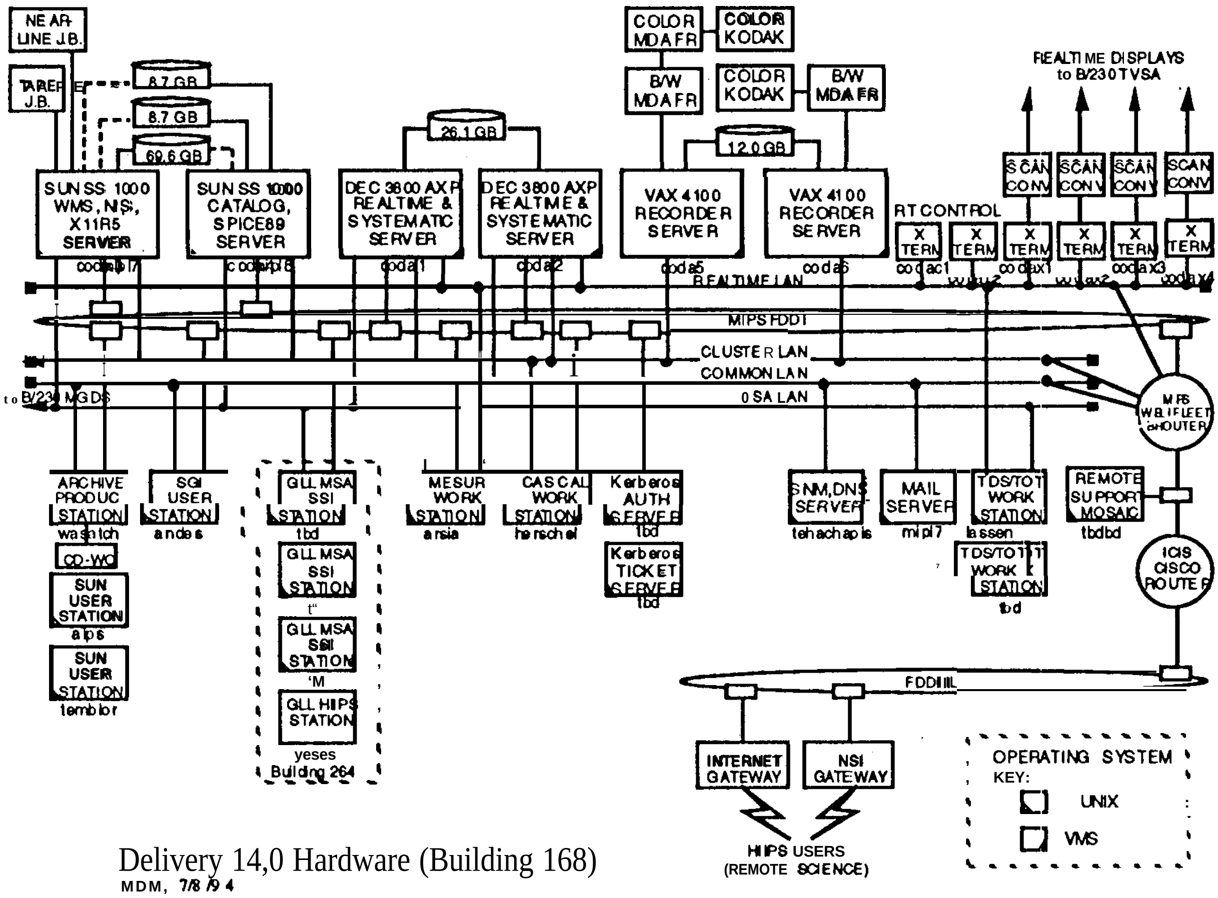

Figure 3. MIPS Modeled Configuration Diagram

MIPS SIMULATION RESULTS

The MIPS model was run for both real-time and systematic processing for a 32-hour simulation time period, starting from midnight to midnight for one day of processing and lapping over into the first 8 hours of the next day, to allow for possible processing of back-logged batch jobs. The key concern was that the required amount of worst-case processing be completed within a given day before a similar processing load is imposed on the system the following day.

In Network 11.5 plot scales, multiple sample data points taken during given simulated time intervals are averaged together to condense the number of points represented (limit of 100) on the plots. However when plots are built using smaller time intervals, less sample points are averaged together, thus revealing spikier plots with more maximum or peak values (see inset plot in Figure 6 shown later).

Typical sample resource utilization plots from Network 11.5's MIPS simulation output are shown in Figures 4, 5, and 6. The x-axis represents the simulation time in seconds times 10 to get the actual seconds. For example, zero seconds is midnight and 57600 (576OX1O) seconds is 4 p.m. or 16 hours past midnight. Depending on the type of system resource being measured, the y-axis represents the percent of time that resource is being utilized, busy, or allocated.

In Figure 4, the CPU of one of the DEC Alpha compute-servers was selected and the plot shows overall CPU utilization by all scheduled processing and I/O being run on that particular CPU over the above specified simulation time. Figure 5 shows the percent of time that Working Mission Storage (a centralized file server) is busy performing I/O operations. Figure 6 shows the percent of time the fiber-optic communications ring is busy transmitting and/or allocated between processing/storage elements. Basically, all plots showed that the given configuration of MIPS elements did not indicate any performance problems using the projected processing loads that were imposed on the system.

Depending on how complex the system being simulated is, Network 11.5 can produce hundreds of resource utilization plots for hardware processing elements, software modules, storage devices, communications links, messages/data being routed, and so on. Also, a detailed statistics file is generated with each simulation execution.

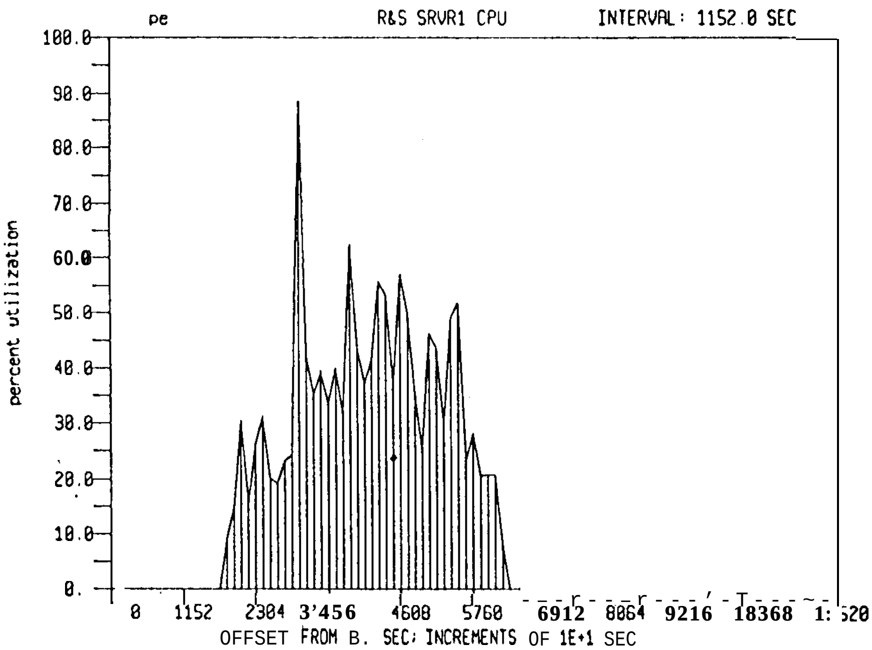

Figure 4. CPU Utilization Over All Procedure Routines

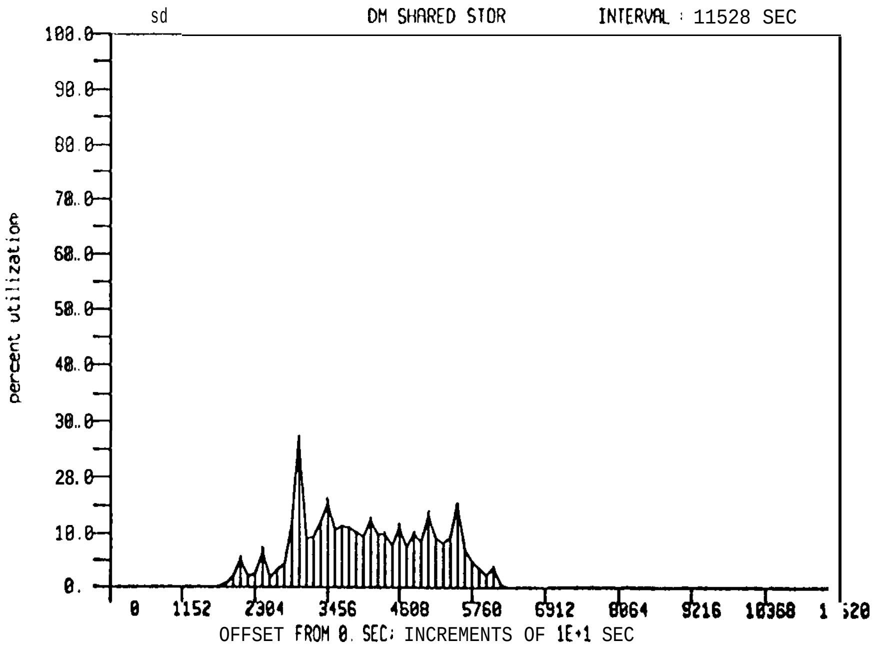

Figure 5. Working Mission Storage Percent Of Time Busy

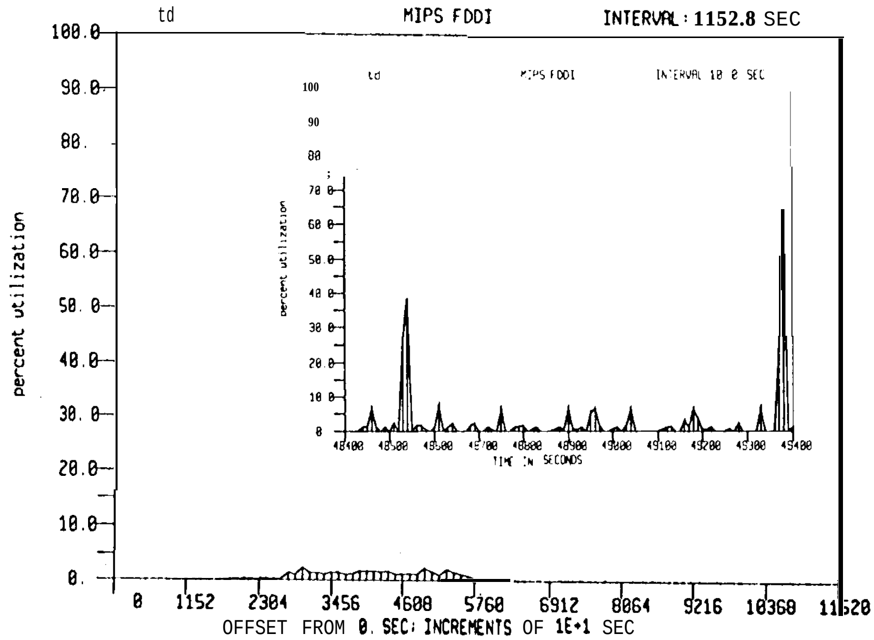

Figure 6. Fiber-Optic Link Percent Of Time Busy/Allocated

IMPACT ANALYSIS

This last section explains how the modeling and simulation analysis affected the MIPS design process by having a supportive and informative impact. As aresult ofvarious modeling and analyses of two MIPS topologies (centralized and distributed), the apparent potential problem to be concerned about with any given architecture was I/O contention and not insufficient CPU resources, given the MIPS real-time, systematic (batch), and interactive image processing tasks are I/O intensive.

One key operational configuration recommendation that resulted from the simulation activities was to not mix the tasks of network file/data serving and high-volume I/O intensive computational processing on the same machine (i.e., compute-server). In addition, the distribution of functions and storage across smaller machines were recommended and taken into consideration to relieve storage 1/0 contention problems. This analysis caused the redirection from a centralized architecture to a distributed approach for the MIPS design and modeling effort.

The MIPS distributed architecture is currently under phased implementation and continued modeling activities are performed to test the effects of add-ens or changes to the system. These on-going refinements to and maintenance of the model, based on developments, are conducted to allow and enable continuing technical decisions on the design and operations of an evolving complex system.

ACKNOWLEDGEMENTS

Special thanks goes out to the MIPS design team for their invaluable support in the system modeling process.

BIOGRAPHY

Laverne Hall is a resident in the area of Los Angeles, California and is a life-time member of the Los Angeles Council of Black Professional Engineers (LACBPE). She received a B.S. in Applied Mathematics with a minor in Computer Science from Tuskegee University in 1979, and a M.S. in Computer Engineering from the University of California at Los Angeles (UCLA) in 1981. Laveme is currently a systems engineer at the Jet Propulsion Laboratory in Pasadena California in the Image Processing Systems Group and a mathematics instructor at L.A. Southwest College. She has served several roles in the LACBPE, including V.P. of the organization and Chairperson of it's Computer Center. She has written, presented, and published several papers in the areas of algorithm development and of system architecture modeling and simulation, both for satellite-based onboard computing.