3-1999

Load Dependent Single Chain Models of Multichain Closed Queueing Networks

Amy Apon
Clemson University, aapon@clemson.edu

Lawrence Dowdy
Vanderbilt University

Load Dependent Single Chain Models of Multichain Closed Queueing Networks[1]

Amy W. Apon
Dept. of Computer Science
University of Arkansas
Fayetteville, Arkansas 72701
aapon@comp.uark.edu

Lawrence W. Dowdy
Dept. of Computer Science
Vanderbilt University
Nashville, TN 37253
dowdy@vuse.vanderbilt.edu

Abstract: Often a multichain product form queueing network is used to model a complex computer system. In many cases, the multichain demands are unavailable or are difficult to obtain. In contrast, load dependent demands are often directly measurable. This paper investigates the use of a load dependent single chain model as an approximate model of an actual multichain system. In restricted cases the load dependent single chain counterpart model of an actual multichain system is exact. In random unrestricted cases, it is shown that the load dependent model is a good approximation to the actual multichain system. It is demonstrated that the load dependent model can also be used effectively for predictive purposes. An experimental validation on a dual-processor PowerPC 604 workstation illustrates the applicability of the load dependent model of an actual multichain system.

Index Terms: load dependent models, multichain closed product form queueing networks, performance prediction, parameter measurements, approximation errors, workload characterization

1. Introduction

Multiple chains of customers, sometimes referred to as workload or job classes, are often used to describe the workload of a computer system. The parameters that characterize each customer chain (e.g., the resource demands that each customer places on each hardware device) typically are either assumed to be given, or come from a clustering analysis of measurement data. However, constructing the chains from measurement data is difficult. The number of chains and the chain demands can vary, depending on the assumptions made during the clustering process. Further, multichain measurements are difficult to obtain, since most systems lack the ability to track the device demands of individual customers. Operating system tasks that are spawned on behalf of individual customers are also part of the overall workload, and these tasks complicate the overall workload characterization process.

From a measurement perspective, single chain measurements are easier to obtain. For instance, the overall average demand placed on a device is the ratio of the device utilization and the device throughput, both of which can be measured easily. No clustering analysis is needed, since all customers are placed in the same chain. The single chain measurements may be either load independent or load dependent. Load independence assumes that the demands placed on a device are independent of the number of customers presently at the device. Load dependence assumes that the demands may be dependent on the current queue length at the device. The load dependent single chain measurements are often no more difficult to obtain than the load independent single chain measurements. To obtain the load dependent demands, the queue length at the device must be noted at the same time when the device utilization and throughput are measured. The load independent demands can be calculated directly from load dependent measurement data.

The purpose of this paper is to analyze the effect of making a load dependent single chain (LDSC) model of a system that is actually multichain. In many cases the actual system characteristics cannot be determined, and the analyst must construct a model based on assumptions about the actual system. In this paper, the model will be described as being exact if it provides the same aggregate performance metrics as the actual system. It is possible in some cases to construct a load dependent single chain model that exactly matches

[1] This work was partially supported by sub-contract 19X-SL131V from the Oak Ridge National Laboratory managed by Martin Marietta Energy Systems, Inc. for the U.S. Department of Energy under contract no. DE-AC05-84OR21400. The equipment used in the experimental validation was purchased through Research Instrumentation Grant #9617384 from the National Science Foundation.

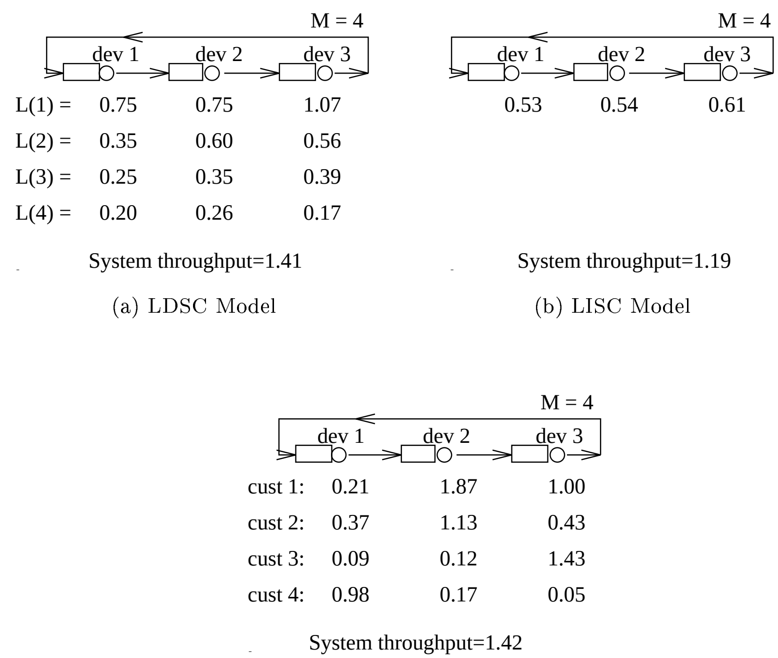
the performance metrics of a multichain system. The focus of this paper is on the use of a load dependent single chain model of a system, based on measurements of the system, as a tool for the description and prediction of an actual multichain system. Thus, given a system that may be multichain, but whose parameters are unknown, this paper investigates the error that is possible when a load dependent single chain model is constructed. This work is related to research in the characterization and construction of multiclass workload models [3, 4, 8, 11, 12] and to the operational analysis of stochastic closed queueing networks [5].

Previous work in this area has focused on the error that occurs when the load independent single chain counterpart model is constructed of an actual multichain system [7]. The performance metrics of the actual multichain system are pessimistically bounded by the performance metrics of its load independent single chain counterpart model. Furthermore, the maximum underestimate of the throughput given by the load independent single chain counterpart model is also bounded as function of the number of devices and the number of customers in the system. This paper extends these results by considering the load dependent counterpart model.

The remainder of the paper is as follows: Section 2 gives an introductory example. Section 3 summarizes the notation and assumptions for the remainder of the paper. Section 4 gives new results for when a load dependent single chain model is constructed of a multichain system. Section 5 demonstrates the usefulness of the load dependent single chain counterpart model when used to predict the performance of a multichain system. Section 6 decribes an experimental validation of the load dependent counterpart model of a multichain system. Section 7 summarizes the results and presents future research directions.

2. Example

Consider the example system shown in Figure 1. This system has $M = 4$ customers and $N = 3$ devices. Suppose that a system monitor collects the load dependent measurements of the system.



(a) LDSC Model

(b) LISC Model

(c) Actual Multichain System

Figure 1: First Example System

For example, measurements are collected for each of the devices as shown in Figure 1(a). $L(i)$ represents the demand (i.e., loading) that is placed on the device when the present queue length is i. Without any additional information, a system analyst might construct either the load dependent single chain (LDSC) model shown in Figure 1(a) or the load independent single chain (LISC) queueing model shown in Figure 1(b). However, it is possible that the actual underlying system is a multichain system, with four distinct customer chains, with the demands as shown in Figure 1(c). That is, *if* the actual system were multichain as illustrated in Figure 1(c), the load dependent and load independent single chain counterpart models that would be constructed from the measurement data are illustrated in Figures 1(a) and 1(b), respectively.

The throughput of the actual multichain system is 1.42 jobs per unit time. The calculated throughput using the load independent single chain counterpart model is 1.19 jobs per unit time. The throughput relative error is calculated as:

$$\frac{\text{actual throughput} - \text{approximate throughput}}{\text{actual throughput}} \times 100\%.$$

The load independent single chain throughput is in error by 16.2%. In contrast, the calculated throughput using the load dependent counterpart model is 1.41 jobs per unit time, which is in error by only 0.7%. In this example the calculated single chain performance of the system is much closer to the actual performance of the system when a load dependent model is used, rather than a load independent model. This motivates further investigation of the accuracy of load dependent single chain models of multichain systems in which multichain measurements are not available.

3. Assumptions and Notation

The actual systems being considered are closed multichain queueing networks. There are N devices in the network. There are M closed routing chains. Without loss of generality, the assumption is made that there is exactly one customer in each closed routing chain, so that the terms "customer" and "chain" may be used interchangeably. Without loss of generality, the state of the system may be described by the presence or absence of each customer at each center in the system. All service time distributions are assumed to have a rational Laplace transform and all queueing disciplines are assumed to be processor sharing. These networks are known to have a product form solution [1]. The product form solution for the probability of being in state s of the original multichain closed queueing network is described as follows:

- $v_i(s)$ — a vector describing the current state of device i in state s, where $v_i(s) = I_{i1}(s)I_{i2}(s)\ldots I_{iM}(s)$. For example, $v_2(s) = 001$ means that the chain 3 customer is the only customer currently at device 2 in state s.
- $I_{ik}(s)$ — an indicator variable to describe the state of center i in state s. $I_{ik}(s) = 1$ if the chain k customer is present at center i, and 0 otherwise.
- n_i — the total number of customers at device i.
- d_{ik} — the multichain service demand at center i for chain k.
- G_{NM} — the normalization constant for the multichain queueing network with N devices and M customers.
- $\mathrm{P}(s)$ — the steady state probability of being in state $s \in \mathcal{S}$. For the assumed multichain system, the product form solution [1] for $\mathrm{P}[s]$ is given by: $\mathrm{P}[s] = \frac{1}{G_{NM}} \prod_{i=1}^{N} n_i! \prod_{k=1}^{M} d_{ik}^{I_{ik}(s)}$.

The counterpart load dependent and load independent single chain queueing network models can be constructed for each multichain system. The load dependent and load independent demands can be calculated from the probabilities of the underlying state diagram of the multichain model. Then the solution of the single chain models can be calculated in a straightforward manner, using product form techniques [1, 10]. These are the same models that would be constructed using measurement data from the actual system if single chain (load dependent or load independent, respectively) measurements were taken at each device.

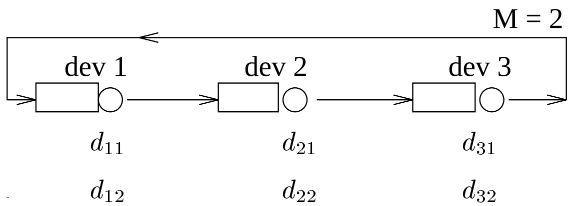


Figure 2: Second Example System

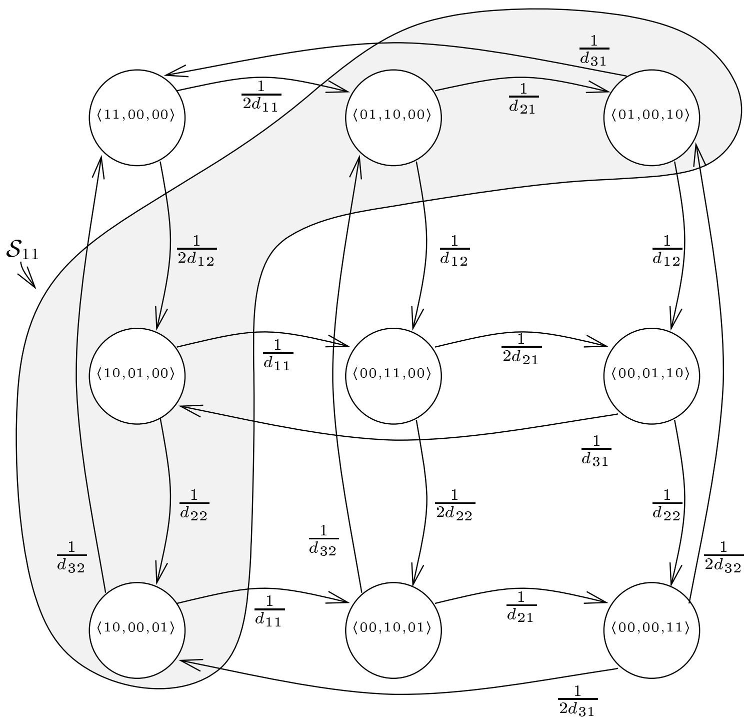


Figure 3: Markov Diagram for the Second Example System

As an example of the procedure for calculating the load dependent demands from the actual multichain system, consider the example system shown in Figure 2. This system has three devices and two customers, with multichain demands as labeled. The multichain state diagram for this system is shown in Figure 3. The nine states in Figure 3 are labeled $\langle 11,00,00\rangle$, $\langle 01,10,00\rangle$, $\langle 01,00,10\rangle$, $\langle 10,01,00\rangle$, $\langle 00,11,00\rangle$, $\langle 00,01,10\rangle$, $\langle 10,00,01\rangle$, $\langle 00,10,01\rangle$, and $\langle 00,00,11\rangle$. Using this notation, $\langle 11,00,00\rangle$ means that customer 1 and customer 2 are both present at device 1, $\langle 01,10,00\rangle$ means that customer 1 is at device 2 and customer 2 is at device 1, and so on. $\mathcal{S}_{im}$ is the set of states from the multichain system in which there are m customers at device i. The set of states in which there is 1 customer at device 1, $\mathcal{S}_{11}$, is shaded. The arcs are labeled with the corresponding transition flow rates between the states. For instance, the flow rate from state $\langle 11,00,00\rangle$ to state $\langle 01,10,00\rangle$ is $\frac{1}{2d_{11}}$. When both customers are at device 1 the rate at which the chain 1 customer finishes at device 1 and proceeds to device 2 is $\frac{1}{2d_{11}}$, which is one half of the chain 1 service rate (i.e., the inverse of the service demand) at device 1. The factor of $\frac{1}{2}$ is due to the processor-sharing discipline, since device 1 is equally shared between the two customers in state $\langle 11,00,00\rangle$.

The service demand at device 1 when there is 1 customer present, $L_1(1)$, is the ratio of the probability of being in $\mathcal{S}_{11}$ (i.e., the utilization of device 1 when there is a single customer at the device) to the flow rate out of $\mathcal{S}_{11}$ as a result of completed service from device 1 (i.e., the throughput of device 1 where there is a single customer at the device). So, $L_1(1) =$

$$\frac{\mathrm{P}[\langle 01,00,10\rangle] + \mathrm{P}[\langle 01,10,00\rangle] + \mathrm{P}[\langle 10,01,00\rangle] + \mathrm{P}[\langle 10,00,01\rangle]}{\mathrm{P}[\langle 01,00,10\rangle]\frac{1}{d_{12}} + \mathrm{P}[\langle 01,10,00\rangle]\frac{1}{d_{12}} + \mathrm{P}[\langle 10,01,00\rangle]\frac{1}{d_{11}} + \mathrm{P}[\langle 10,00,01\rangle]\frac{1}{d_{11}}}.$$

The product form solution for the probability for each state in Figure 3 gives,

$$\begin{aligned} L_1(1) &= \frac{\frac{1}{G_{32}}(d_{12}d_{31} + d_{12}d_{21} + d_{11}d_{22} + d_{11}d_{32})}{\frac{1}{G_{32}}(d_{12}d_{31}\frac{1}{d_{12}} + d_{12}d_{21}\frac{1}{d_{12}} + d_{11}d_{22}\frac{1}{d_{11}} + d_{11}d_{32}\frac{1}{d_{11}})} \\ &= \frac{d_{12}d_{31} + d_{12}d_{21} + d_{11}d_{22} + d_{11}d_{32}}{d_{31} + d_{21} + d_{22} + d_{32}}. \end{aligned}$$

4. Accuracy of the Load Dependent Single Chain Counterpart Model

The load dependent single chain counterpart model can be highly accurate when used to model an actual multichain computer system. With respect to device utilizations and system throughput, a load independent multichain queueing network that has $N = 2$ devices can be modeled exactly by its load dependent single chain counterpart model. This observation is proven in Appendix A from the stochastic analysis viewpoint and is shown by comparing the state space of the original multichain system with the state space of the corresponding load dependent single chain model. The proof is based on the idea that the states in the multichain network can be partitioned into sets according to the number of customers at device 1. The probability of being in this set of states in the multichain queueing network is equal to the probability of being in the corresponding state in the Markov diagram of the load dependent single chain model. This is not true in the load independent case.

In systems with three or more devices, the LDSC counterpart model does not exactly model the actual multichain system. Analytic error bound analysis is difficult. Thus, an experimental approach was used to estimate the errors that can occur when an actual multichain system is modeled by its LDSC counterpart model. The number of customers was allowed to range from 2 to 7, and the number of devices was allowed to range from 3 to 7. For each of the thirty combinations, one hundred random multichain networks with uniformly distributed demands were generated. The range of the demands generated was 0.0 to 20.0 [2]. Each of the multichain systems was solved analytically using Markovian analysis, and the load dependent and load independent single chain counterpart models were constructed and solved using the queueing network solution package QNAP [13]. The relative errors between the single chain models and the actual multichain systems were calculated. Figure 4 shows the distribution of the

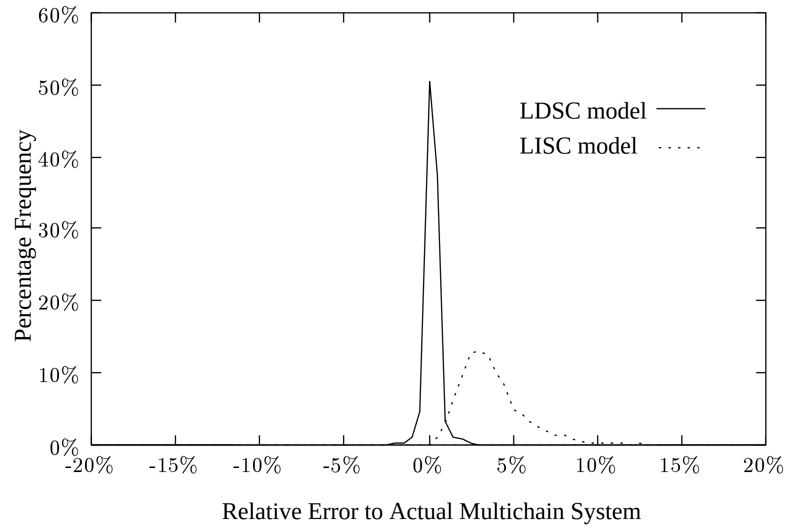


Figure 4: Distribution of Errors for 3000 Random Networks

relative errors that were obtained for the two single chain counterpart models, LDSC and LISC. Figure 4 shows that the relative error for the load independent single chain model is always positive, and tends to be in the range of 0 to 10 percent. The relative error for the load dependent single chain model can be positive or negative, but the magnitude of the error is less than 2 percent for more than 99 percent of the random networks generated.

In the special case of multichain segregated systems, a system in which each customer receives all of its service from a disjoint subset of devices, the load dependent counterpart model is exact. The load dependent solution is exact because in a segregated system the load dependent demand at each device when one customer is present is the same as the demand of the chain that visits the device, and demand when more than one customer is present is 0. As a note, this happens to be type of system for which the greatest error is obtained for the load independent counterpart model [7].

Given that a particular system satisfies the assumptions for a

2 It should be noted that the distribution of relative errors is the same for any range of uniformly distributed random demands.

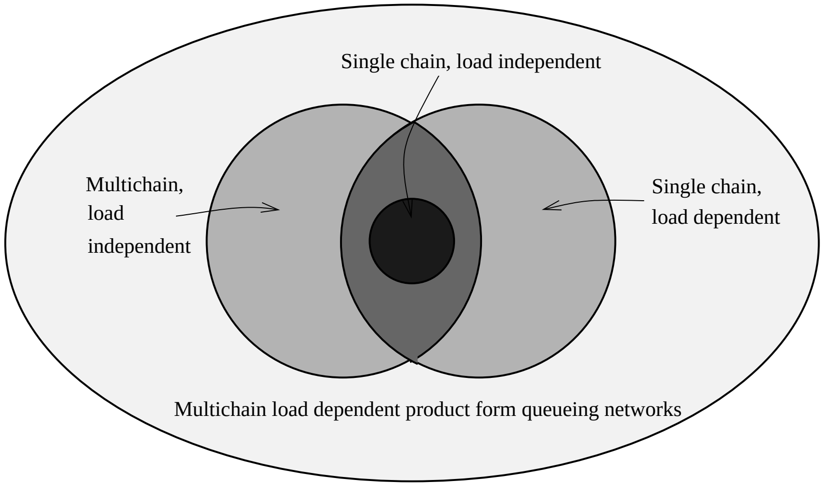
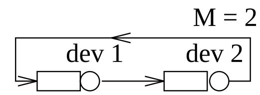


Figure 5: Product Form Models

product form queueing network model, a system analyst must still decide if the system model is to be single chain or multichain, load independent or load dependent. If multichain performance metrics are required, then a multichain model must be used. If only the aggregate performance metrics are required, then a single chain model may suffice.

Some observations can be made about the relationship between multichain and single chain queueing networks, as illustrated by the Venn diagram shown in Figure 5. First, although a unique load dependent single chain counterpart model exists for every load independent multichain queueing network, the counterpart may not exactly model the multichain network (e.g., the example in Figure 1). Second, there exist product form queueing networks that can be modeled exactly by either a load independent multichain queueing network, or by a load dependent single chain network, but that are not modeled exactly by the corresponding load independent single chain counterpart network (e.g., members of the set of 2-device queueing networks have a load-dependent single chain counterpart model that is exact, but have a load-independent counterpart model that may have up to 33% error [7]).



$L_1(1) = 1 \qquad L_2(1) = x$

$L_1(2) = 1 \qquad L_2(2) = 2$

Figure 6: Load Dependent Queueing Network Example

A third observation is that there exist load dependent single chain product form queueing networks that are not counterpart models for any load independent multichain queueing network. The load dependent queueing network in Figure 6 is such an example. If this queueing network is the counterpart model of a load independent multichain queueing network, then the demands of the multichain network must satisfy:

$$L_1(1) = 1 = \frac{d_{11}d_{22} + d_{12}d_{21}}{d_{22} + d_{21}} \quad (1)$$

$$L_1(2) = 1 = \frac{d_{11} + d_{12}}{2d_{11}d_{12}} \quad (2)$$

$$L_2(1) = x = \frac{d_{11}d_{22} + d_{12}d_{21}}{d_{11} + d_{12}} \quad (3)$$

$$L_2(2) = 2 = \frac{d_{22} + d_{21}}{2d_{22}d_{21}} \quad (4)$$

where d_{11}, d_{12}, d_{21}, and d_{22} are non-negative real numbers. However, it is relatively straightforward to show that if $x < 2$, d_{21} and d_{22} must be complex numbers. This observation implies that there are product form queueing networks that can be measured and modeled by load dependent single chain models, but have no underlying load independent multichain queueing network.

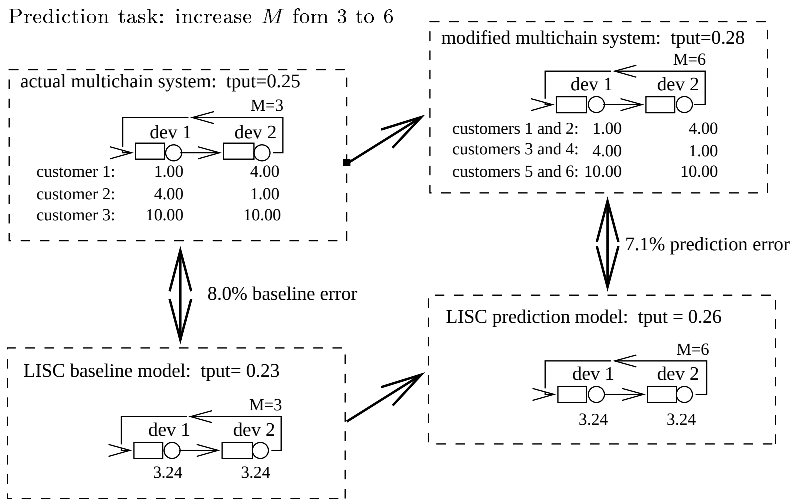


Figure 7: Prediction of Doubled M using the LISC Model

5. Accuracy of the Load Dependent Single Chain Predictive Model

A primary use of computer system models is for performance prediction. For example, suppose that a model of a computer system is constructed and used to predict the performance of the system when the multiprogramming level is doubled. It is assumed that the actual underlying system is a closed product form multichain queueing network, and that the number of customers in each chain will double. Figure 7 illustrates the steps involved in using a LISC model. The actual multichain system is represented in the upper left box. From measurement data, the LISC *baseline* model (lower left box) is constructed. The baseline model is typically in error since models rarely capture all aspects of the actual system being modeled. Then, the baseline model is modified to form the *prediction* model (lower right box). In the Figure 7 example, the prediction model is formed by simply changing M from 3 to 6. The prediction model is solved. The relative error between the performance indicated by the

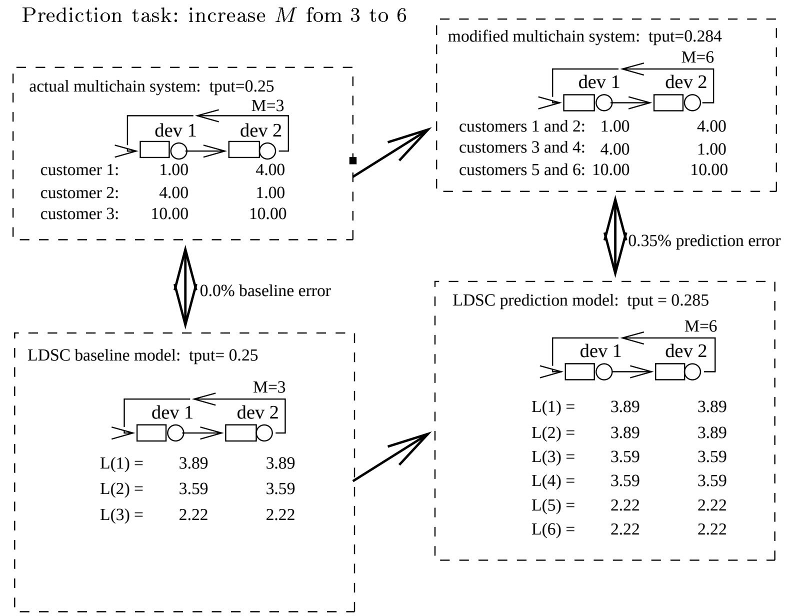


Figure 8: Prediction of Doubled M using the LDSC Model

prediction model, and the performance that occurs when the actual system is modified (upper right box) is the prediction model error. As seen in the Figure 7 example, the LISC baseline model has an 8.0 percent error and the corresponding LISC prediction model has a 7.1 percent error.

Instead of using the counterpart LISC model, it is possible to use the LDSC model to predict the performance of the system as the multiprogramming level doubles uniformly in all chains. The primary difficulty with using a load dependent model for this prediction task is the determination of the load dependent demands for an increased number of customers. For example, when a load independent model is used to predict the performance as the multiprogramming level doubles, the only change required to the model is the change to the multiprogramming level (see Figure 7). However, in the load

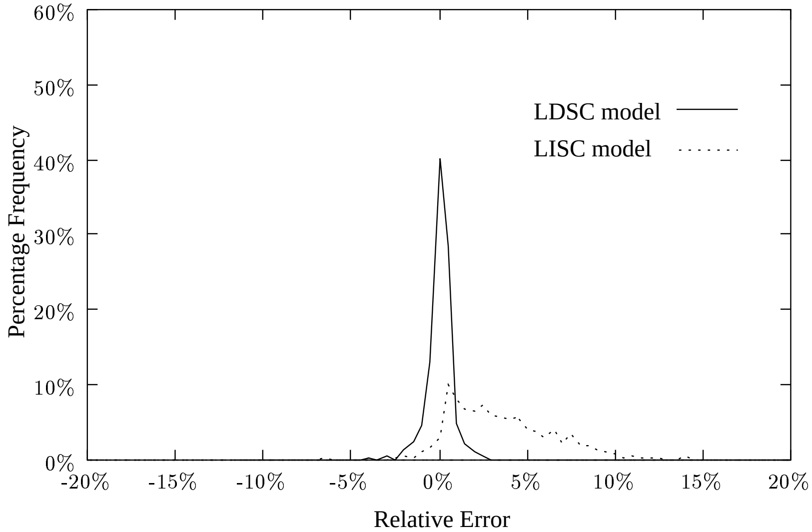


Figure 9: Distribution of Prediction Errors when M is Doubled

dependent model, it is necessary to not only increase the multiprogramming level, but also to specify the load dependent demands for the increased number of customers.

Reconsider the prediction example of Figure 7, but replace the LISC baseline and prediction models by the LDSC counterparts as shown in Figure 8. Since the system has only two devices (and consistent with the theorem in Section 5), the baseline error is 0.0 percent. To construct the prediction model, the load dependent demands must be specified for each number of customers from 1 to the new multiprogramming level, $M = 6$. An exact calculation of the demands would require knowing the new distribution of customers in each state in the Markov diagram of the modified system. However, a reasonable hueristic is to specify that the load dependent demand for p customers in the prediction model be the same as the demand for $\lceil \frac{p}{2} \rceil$ customers in the baseline model. For the example system in Figure 8, this leads to a 0.35 percent prediction error.

To investigate the prediction potential of LDSC models for a wider range of systems, an experimental approach was used. For each number of customers in the range from 2 to 6, and for 2 or 3 devices, one hundred random multichain networks were generated. Each of the multichain systems was solved analytically using Markovian analysis, and the load dependent and load independent counterpart models were constructed and used as the baseline and prediction models. Figure 9 shows the distributions of the relative errors that were obtained for predictions using the two types of single chain models. The figure shows that the magnitude of the error when the load dependent single chain model is used for prediction tends to be significantly smaller than for the load independent single chain model. As an indicator, 37 percent of random networks had prediction errors of less than 2 percent using the LISC model, while 96 percent of the networks had prediction errors of less that 2 percent using the LDSC model.

Another common prediction task is to predict the performance of the system when the speed of the bottleneck device is increased. The bottleneck device of a computer system is the device which is the most heavily utilized. If this device is replaced by an equivalent, but faster one, the performance of the entire system will improve. Modifications to the baseline models are straightforward for this prediction task. For instance, if the bottleneck device is replaced by one that is twice as fast, the demands at the bottleneck device are halved in the prediction model.

As before, an experimental approach was used to investigate the errors in larger systems. For each number of customers in the range from 2 to 7, and for each number of devices in the range from 2 to 7, one hundred random multichain networks were generated. Figure 10 shows the distributions of the relative errors that were obtained for predictions using the two types of single chain models. The magnitude of the prediction errors for the LISC model is less than 2 percent for only 10 percent of the random networks generated. In contrast, the magnitude of the prediction errors for the LDSC model is less than 2 percent for more than 81 percent of the random networks generated. These two prediction studies indicate that the load dependent single chain counterpart model may be useful in predicting the behavior of actual multichain systems.

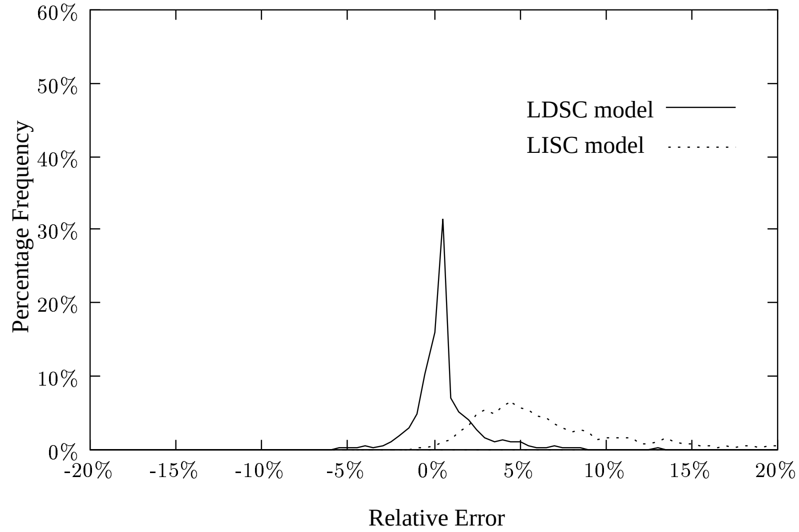


Figure 10: Relative Error Distribution Predicting Bottleneck Speedup

6. Experimental Validation

A experimental case study was performed in order to illustrate the accuracy and applicability of the load dependent counterpart model. The workstation used in this study is a dual-processor VME bus-based workstation with a Motorola MVME3600 VME processor module. The MVME3600 is a processor/memory module consisting of two 200MHz PowerPC 604 processors. Each processor accesses the same large shared memory but has separate L1 and shared L2 cache. HBench-OS [2] was used to measure memory access latencies. The L1 cache size is 64 KB, split into 32 KB for data and 32 KB for instructions. The measured latency for the L1 cache is 7.8 ns. The L2 cache size is 256 KB and has a measured latency of 210.3 ns. Main memory consists of 64 MB ECC DRAM. The bus interface runs at speeds greater than 66 MHz, and consists of a 64-bit data bus and a 32-bit address bus.

System software on the workstations is the IBM AIX 4.1.5.0 operating system. The MPICH implementation, version 1.1, of MPI (Message Passing Interface) [9] is chosen as the application platform for spawning processes on the individual processors. A call to MPI_Reduce is used to synchronize the processes prior to the execution of the application code. Since each process in MPI has its own memory space, interprocess synchronization and cache coherency is not an issue in this experiment.

Standard benchmarking techniques were used in the study. All runs were executed for at least fifty trials. The timings obtained were cleared of outliers by eliminating the bottom and top twenty percent of readings. No other applications were running on the workstations at the time. Each call was executed a few times before the actual timed call was executed in order to eliminate time taken for setting up connections, loading of the cache, and other housekeeping functions.

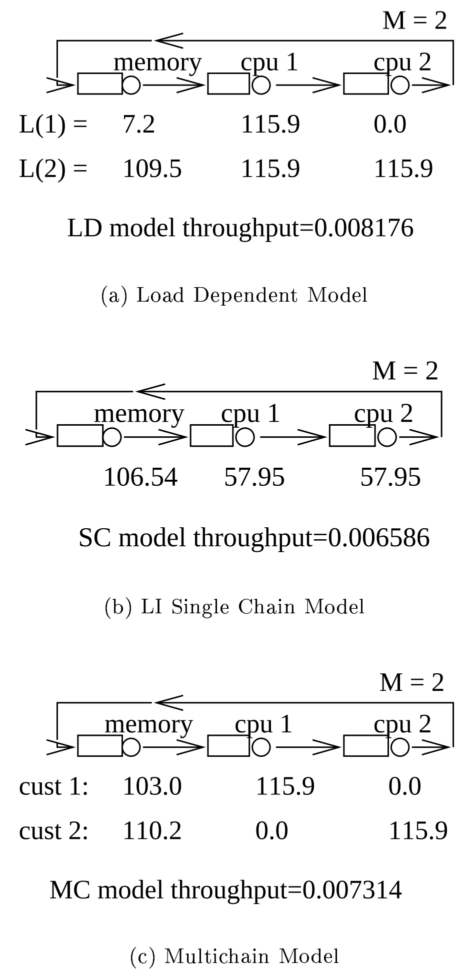
The application code chosen is an implementation of the summation of a large array of doubles. This type of application appears frequently in scientific codes such as the calculation of molecular or atomic structure data. In this code the elements of the large array must be accessed sequentially. An array size is chosen that is larger than will fit into the L1 and L2 caches. The array size for all experiments is 160,000 doubles, or approximately 640KB. To eliminate caching, the array is dynamically allocated in each run of the experiment and filled with randomly generated positive doubles in the range of 1.0E-04 to 1.0E+03. The array is filled randomly in order, and then accessed during the summation code, also, in order. With this setup, successive access to the array elements will cause a cache miss each time a new cache line is accessed.

In order to illustrate the effects of multichain data, two types of access are used in the experiment. Chain 1 accesses the data one element at a time, in order, for a total of 80,000 elements. This method of access gives maximum use of data in every cache line and gives the best memory access performance from main memory that is possible for this type of application. Chain 2 accesses every other element of the data, in order, for a total of 80,000 elements. This method of access will have a cache miss after only half of the elements in the cache have been accessed. The time for cpu processing of the array element summation and loop increments is the same for both

Chain 1 and Chain 2 access. However, Chain 2 access places a higher demand on the memory system than Chain 1 access.

The queueing network models for this system contain three nodes, one for the memory system and one for each cpu, as illustrated in Figure 11. A set of experiments was run in order to determine the demands for each node in the queueing network model. In order to determine the demands placed on the memory system by Chain 1 and Chain 2 access, measurements were performed both with and without array access. A first experiment was run to measure the cost of summing and assigning an element in a loop, by a single executing process, in which the same element is summed each time. This access pattern eliminates the cost of memory access since the element will stay in a register for the duration of the loop. This number is the cpu demand for when a single process is executing on the cpu, and is measured to be 115.9ns. A second set of baseline experiments measures the cost of summing and assigning the array elements for Chain 1 and Chain 2 access with a single process in the system. The difference between these sets of measurements and the measurements in the first baseline experiment gives the load independent demand to the memory system for Chain 1 and Chain 2 access. Figure 11(c) illustrates the measured multichain demands for Chain 1 and Chain 2. The calculated throughput of the multichain model of the system is 0.007314 summations per nanosecond. The calculated utilizations of the memory, cpu1 and cpu2 are .779, .43, and .41, respectively. In this system the memory is the bottleneck device. The actual measured throughput of the system is 0.0088456 summations per nanosecond. The multichain model is in error of the actual system by 17.3%.

The single chain load independent queueing model of the same system with two processes in the system is illustrated in Figure 11(b). Note that the routing probability to each cpu is .5, so that the single chain demand at each cpu in this case is half of the original demand for each chain. The system throughput of the single chain model is 0.006586 summations per nanosecond. For this system the single chain model is in error of the actual system by 25.5%. A model that is in error to this degree would not likely be used directly for performance prediction, but rather would be calibrated first [6].



(a) Load Dependent Model

(b) LI Single Chain Model

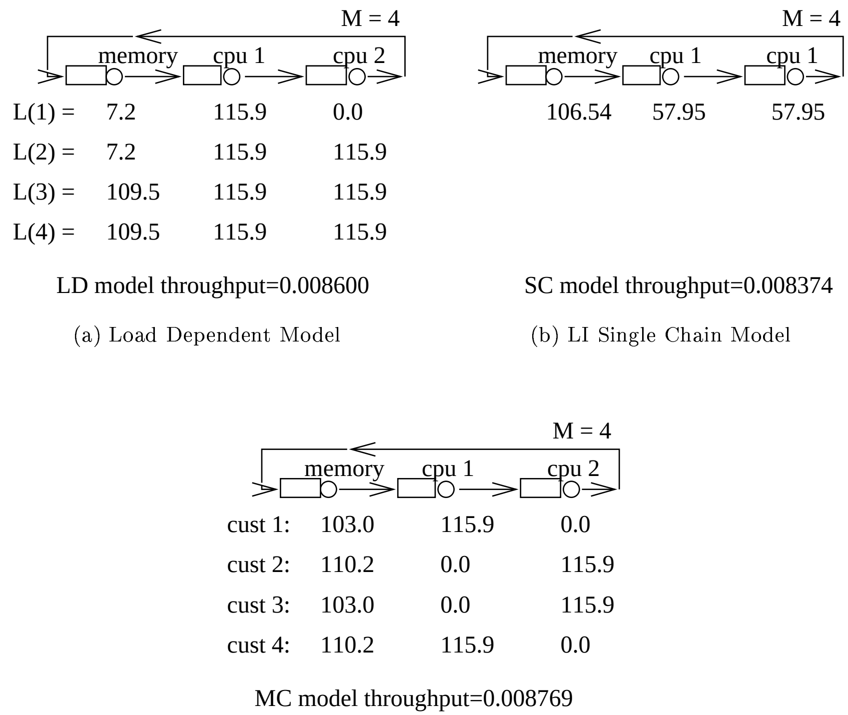
(c) Multichain Model

Figure 11: Baseline models for the system

Load dependent single chain measurements can be obtained by considering the completion times of Chain 1 and Chain 2 access when both processes are executing in the system at the same time (i.e., when the multiprogramming level of the system is two). Measurements are required for the mean demand for each device when two processes are executing at the device, and when a single process is executing at the device. Due to the synchronization at the start of each experimental run, Chain 1 and Chain 2 processes begin at the same time and each execute for 80,000 iterations. Chain 1 completes summation, on the average for each element, after 218.9ns. Chain 2 completes summation, on the average for each element, after 226.1ns. Thus, for 218.9ns the system is executing with two active processes. Because of overlapping execution of the two processors by each process and the access to the single main memory by each process, and because memory access is the bottleneck for this system, this time is a reasonable estimate of the time to perform memory access when two processes are in the system. Thus, the load dependent demand for memory access for when there are two processes in the system is 109.45ns per process. The remaining time of 7.2ns there is a single process executing, so that the load dependent demand when there is one process executing is 7.2ns. The demand at the processors is the same for both one and two processes, except that when a single process is in the system then only one processor is utilized.

Figure 11(a) illustrates the measured load dependent demands for Chain 1 and Chain 2 and the calculated throughput for the load dependent queuing network model. The throughput for the load dependent baseline model of this system is 0.008176 summations per nanosecond, which is in error of the actual measured system throughput by 7.6%. For this system, the baseline load dependent model is a more accurate description of the actual system than either the multichain or the single chain load independent models.

Each of the queueing network models can be used to predict the performance of the system for when four processes are executing, two each of Chain 1 and Chain 2. Figure 12 illustrates the prediction models for four processes in the system. The measured throughput of the modified system is 0.00813 summations per nanosecond. The multichain model predicts a throughput of 0.008769 when four processes are in the system. This is in error of the actual system through-



(a) Load Dependent Model

(b) LI Single Chain Model

(c) Multichain Model

Figure 12: Prediction models for the system

put by 7.8%. In spite of the large error in the baseline model, the single chain model predicts a throughput to the modified system of 0.008374. This is in error of the actual modified system throughput by 3.00%.

The demands for the load dependent model are illustrated in Figure 12(a). The same technique for calculating the load dependent demands is used here as in earlier sections, except that when two processes are executing in the system then the same demand is placed on both processors rather than on just one. The load dependent model predicts the system throughput to be 0.008600. This is is error of the actual system throughput by 5.8%. The load dependent model gives a system throughput that is closer to the throughput of the actual modified system than the multichain model of the modified system. Although it is not closer than that predicted by the single chain model, it is likely to give a more confident estimate of the predicted performance of the actual modified system since the baseline model also matches the system closely. Had the single chain model been calibrated the predicted throughput would likely be more in error of the actual modified system throughput than it is.

7. Summary and Future Work

The load dependent single chain (LDSC) counterpart model is an effective tool for describing and predicting the performance of actual multichain computer systems. In the case where the number of devices is limited to two, the LDSC model gives the same performance metrics as the actual system. In random cases where the number of devices is greater than two, the LDSC model is a good approximation to the actual multichain system.

The LDSC counterpart model can also be used effectively for predictive purposes. For two types of prediction, doubling the multiprogramming level and increasing the speed of the bottleneck device, the LDSC model is shown to be effective at accurately predicting the performance of the actual multichain system.

Several interesting issues remain regarding the use of load dependent single chain models.

- In Figure 4, the distribution of the relative errors appears to be symmetric about 0.0. Also, in Figure 6, when $x > 2$ the load dependent model is the counterpart for exactly two multichain networks. This indicates that is may be possible to show analytically that the errors obtained from the LDSC counterpart model of a multichain system are symmetric about 0.0. Thus, analytic error analysis may be possible.
- In the case of the LISC counterpart model of a multichain system, known error bounds exist as a function of the number of devices and customers. It may be possible to extend such error bound results to the LDSC counterpart model.
- Further analysis is required to determine how to change the load dependent demands under arbitrary performance prediction tasks.
- The experimental validation study described here is limited in scope. Much more extensive actual system validation is required. The application of LDSC models for prediction of actual computer systems is needed.

Appendix A

With respect to device utilizations and system throughput, a load independent multichain queueing network that has $N = 2$ devices can be modeled exactly by its load dependent single chain counterpart model. This can be proven by examing the underlying Markov diagrams in the original multichain system and its load dependent counterpart model, as follows:

When the number of devices is $N = 2$, the state of the system can be described by a single vector, indicating the presence or absence of each chain's customer at device 1. In this system, each state can be described by $s = \langle p_1 p_2 ... p_M \rangle$, where $p_k = 1$ means that the chain k customer is at device 1, and $p_k = 0$ means that the chain k customer is at device 2. $\mathcal{S}_m$ is the set of states from the multichain system in which there are m customers at device 1 (and $M - m$ customers at device 2). To show that the device utilizations and system throughput match between the actual multichain system and its LDSC counterpart model for any number of customers, M, it is sufficient to show that the probability of being in state $\langle n_1, n_2\rangle$ is equal to the probability of being in $\mathcal{S}_{n_1}$.

Consider the load dependent single chain counterpart model. The probability of being in state $\langle M, 0\rangle$ is calculated as $\mathrm{P}[\langle M,0\rangle] =$

$$\frac{1}{1+\left(\frac{L_2(1)}{L_1(M)}\right)+\left(\frac{L_2(1)}{L_1(M)}\right)\left(\frac{L_2(2)}{L_1(M-1)}\right)+\cdots\left(\frac{L_2(M)}{L_1(1)}\right)}. \quad (5)$$

By definition of $L_2(1)$ as the loading of device 2 with 1 customer present, $L_2(1)$ is the ratio of the probability of a single customer being at device 2 (i.e., $M-1$ customers being at device 1) to the throughput of device 2 when a single customer is present at device 2. That is,

$$\begin{aligned}
L_2(1) &= \frac{\mathrm{P}[\mathcal{S}_{M-1}]}{\text{throughput from device 2 out of } \mathcal{S}_{M-1}} \\
&= \frac{\mathrm{P}[\mathcal{S}_{M-1}]}{\frac{1}{d_{21}}\mathrm{P}[\langle 011\cdots 1\rangle]+\frac{1}{d_{22}}\mathrm{P}[\langle 1011\cdots 1\rangle]+\cdots+\frac{1}{d_{2M}}\mathrm{P}[\langle 111\cdots 10\rangle]} \\
&= \frac{\mathrm{P}[\mathcal{S}_{M-1}]}{\frac{(M-1)!}{G_{2M}}\left(d_{12}d_{13}\cdots d_{1M}+d_{11}d_{13}\cdots d_{1M}+\cdots+d_{1,M-1}\right)}.
\end{aligned}$$

Similarly,

$$\begin{aligned}
L_1(M) &= \frac{\mathrm{P}[\mathcal{S}_{M}]}{\text{throughput from device 1 out of } \mathcal{S}_{M}} \\
&= \frac{\mathrm{P}[\mathcal{S}_{M}]}{\mathrm{P}[\langle 11\cdots 1\rangle]\left(\frac{1}{M}\frac{1}{d_{11}}+\frac{1}{M}\frac{1}{d_{12}}+\cdots+\frac{1}{M}\frac{1}{d_{1M}}\right)} \\
&= \frac{\mathrm{P}[\mathcal{S}_{M}]}{\frac{M!}{G_{2M}}\frac{1}{M}\left(d_{12}d_{13}\cdots d_{1M}+d_{11}d_{13}\cdots d_{1M}+\cdots\cdots+d_{1,M-1}\right)}.
\end{aligned}$$

Thus, dividing $L_2(1)$ by $L_1(M)$ cancels the denominators, giving

$$\frac{L_2(1)}{L_1(M)} = \frac{\mathrm{P}[\mathcal{S}_{M-1}]}{\mathrm{P}[\mathcal{S}_{M}]}.$$

In general,

$$L_2(k) = \frac{\mathrm{P}[\mathcal{S}_{M-k}]}{\left(\frac{1}{G_{2M}}\right)\left(\frac{1}{k}\right)\sum_{s\in\mathcal{S}_{M-k}}\sum_{j=1}^{M}\frac{I_{2j}(s)}{d_{2j}}(M-k)!k!\prod_{i=1}^{M}d_{1i}^{I_{1i}(s)}d_{2i}^{I_{2i}(s)}} \quad (6)$$

$$L_1(M-k+1) =$$

$$\frac{\mathrm{P}[\mathcal{S}_{M-k+1}]}{\left(\frac{1}{G_{2M}}\right)\left(\frac{1}{M-k+1}\right)\sum_{s\in\mathcal{S}_{M-k+1}}\sum_{j=1}^{M}\frac{I_{1j}(s)}{d_{1j}}(M-k+1)!(k-1)!\prod_{i=1}^{M}d_{1i}^{I_{1i}(s)}d_{2i}^{I_{2i}(s)}} \quad (7)$$

Consider the denominators of equations 6 and 7. Each term of the denominator of equation 6 corresponds to a particular arc leaving a particular state $s_a \in \mathcal{S}_{M-k}$, going to another state $s_b \in \mathcal{S}_{M-k+1}$. There is a matching arc that returns from state $s_b \in \mathcal{S}_{M-k+1}$ to state $s_a \in \mathcal{S}_{M-k}$, which corresponds to a term in equation 7. This is true, since if $s_a = \langle I_{11}I_{12}\cdots I_{1c}\cdots I_{1M}\rangle$, then $s_b = \langle I_{11}I_{12}\cdots I_{2c}\cdots I_{1M}\rangle$, indicating that the chain c customer moves between device 1 and device 2. Thus, the term corresponding to the transit of customer c in state s_a in the denominator of equation 6,

$$\frac{(M-k)!(k)!}{G_{2M}(k)}\prod_{i=1,i\neq c}^{M}d_{1i}^{I_{1i}(s_a)}d_{2i}^{I_{2i}(s_a)}$$

is equal to

$$\frac{(M-k+1)!(k-1)!}{G_{2M}(M-k+1)}\prod_{i=1,i\neq c}^{M}d_{1i}^{I_{1i}(s_b)}d_{2i}^{I_{2i}(s_b)}$$

which is the term corresponding to the transit of customer c in state s_b in the denominator of equation 7. Therefore, in general

$$\frac{L_2(k)}{L_1(M-k+1)} = \frac{\mathrm{P}[\mathcal{S}_{M-k}]}{\mathrm{P}[\mathcal{S}_{M-k+1}]}.$$

Using this to simplify the denominator in equation 5,

$$\begin{aligned}
\mathrm{P}[\langle M,0\rangle] &= \frac{1}{1+\frac{\mathrm{P}[\mathcal{S}_{M-1}]}{\mathrm{P}[\mathcal{S}_M]}+\frac{\mathrm{P}[\mathcal{S}_{M-1}]}{\mathrm{P}[\mathcal{S}_M]}\frac{\mathrm{P}[\mathcal{S}_{M-2}]}{\mathrm{P}[\mathcal{S}_{M-1}]}+\cdots+\frac{\mathrm{P}[\mathcal{S}_{M-M}]}{\mathrm{P}[\mathcal{S}_{M-M+1}]}} \\
&= \frac{1}{1+\frac{\mathrm{P}[\mathcal{S}_{M-1}]}{\mathrm{P}[\mathcal{S}_M]}+\frac{\mathrm{P}[\mathcal{S}_{M-2}]}{\mathrm{P}[\mathcal{S}_M]}+\cdots+\frac{\mathrm{P}[\mathcal{S}_0]}{\mathrm{P}[\mathcal{S}_M]}} \\
&= \frac{\mathrm{P}[\mathcal{S}_M]}{\mathrm{P}[\mathcal{S}_M]+\mathrm{P}[\mathcal{S}_{M-1}]+\cdots+\mathrm{P}[\mathcal{S}_0]} \\
&= \frac{\mathrm{P}[\mathcal{S}_M]}{1} \\
&= \mathrm{P}[\mathcal{S}_M].
\end{aligned}$$

Likewise, the solution of $\mathrm{P}[\langle k, M-k\rangle]$ in terms of $\mathrm{P}[\langle M,0\rangle]$ shows that $\mathrm{P}[\langle k, M-k\rangle] = \mathrm{P}[\mathcal{S}_k]\ \forall\ k$.

Since $\mathrm{P}[\mathcal{S}_k] = \mathrm{P}[\langle k, M-k\rangle]$, any Markov reward function that has equal rewards for states in $\mathcal{S}_k$ and the single state, $\langle k, M-k\rangle$, will be equal in the two state space diagrams. This is true for the aggregate utilization and throughput of the actual multichain system and the utilization and throughput of the load dependent single chain counterpart model. Thus, for 2-device networks, the LDSC counterpart model exactly matches the performance of the actual multichain system. □

References

[1] Baskett, F., Chandy, K. M., Muntz, R. R., and Palacios, F. G. Open, closed, and mixed networks of queues with different classes of customers. *Journal of the Association for Computing Machinery 22*, 2 (1975), 248–260.

[2] Brown, A. B., and Seltzer, M. I. Operating system benchmarking in the wake of Lmbench: A case study of the performance of NetBSD on the Intel x86 architecture. In *Proc., ACM Sigmetrics Conference on Measurement and Modeling of Computer Systems* (1997), pp. 214–224.

[3] Calzarossa, M., and Serazzi, G. Workload characterization: A survey. *Proceedings of IEEE Special Issue on Performance Evaluation 81*, 8 (Aug. 1993), 1136–1150.

[4] Cheng, W. C. *Optimization, performance bounds, and approximations in queueing networks.* PhD thesis, University of California, Los Angeles, 1992.

[5] Dallery, Y., and Cao, X.-R. Operational analysis of stochastic closed queueing networks. *Performance Evaluation 14* (1992), 43–61.

[6] Dowdy, L., and Flowers, F. J. A comparison of calibration techniques for queuing network models. In *International Conference on Management and Performance Evaluation of Computer Systems (CMG '89)* (Reno, Nevado, 1989), pp. 644–655.

[7] Dowdy, L. W., Carlson, B. M., Krantz, A. T., and Tripathi, S. K. Single class bounds of multi-class queueing networks. *Journal of the Association for Computing Machinery 39*, 1 (1992), 188–213.

[8] Ferrari, D. *Computer Systems Performance Evaluation.* Prentice-Hall, 1978.

[9] Gropp, W., Lusk, E., Doss, N., and Skjellum, A. A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard. Tech. Rep. Preprint MCS-P567-0296, Argonne National Laboratory, March 1996.

[10] Reiser, M., and Lavenberg, S. S. Mean-value analysis of closed multichain queueing networks. *Journal of the Association for Computing Machinery 27*, 2 (Apr. 1980), 313–322.

[11] Rose, C. A. A measurement procedure for queueing network models of computer systems. *Computing Surveys 10*, 3 (Sept. 1978), 263–280.

[12] Serazzi, G., Ed. *Workload Characterization of Computer Systems and Computer Networks.* Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1986.

[13] Simulog Corp. *QNAP2 Reference Manual*, 1989.