PERFORMANCE CONSIDERATIONS RELATING TO THE DESIGN OF INTERCONNECTION NETWORKS FOR MULTIPROCESSING SYSTEMS

Earl Hokens and Ahmed Louri
Department of Electrical and Computing Engineering
The University of Arizona
Tucson, AZ 85721

Abstract – *Choosing the best interconnection scheme for a multiprocessor is no easy task. This paper presents a two phase analysis to assist in this task. The two phases of the analysis are analytical, and simulation. The analytical phase introduces a new metric called the network bandwidth requirement, or nbr. The nbr is an estimate for the interconnecting network link speed of a multiprocessor. This result is used in the second phase of the analysis, different multiprocessor configurations are simulated using stochastic activity networks to verify the results of the analytical phase, and analyze the effects of contention and memory design on performance.*

INTRODUCTION

Processing need is surpassing current limits of single processor technology [1]. This fact is pushing the envelope of computer design deeper into the realm of parallel processing. The single most important issue of parallel processing is the interconnection network [1]. This paper will give a computer designer insights on how to select an interconnection network for use in a tightly coupled multiprocessor. The analysis however, can also be applied to Multicomputers and computer networks.

Our approach to this problem is from a different perspective than previous research [2]; our analysis centers on the individual interconnecting links as part of a complete multiprocessor. Rather than calculating a bandwidth for the entire network, we calculate a bandwidth requirement in words per cycle for the interconnecting links. Furthermore, we do not isolate the interconnection network for the purpose of analysis. Processor characteristics, memory issues, data consistency issues are all considered in addition to interconnection network issues.

We propose a two step method for the analysis of this problem that results in an estimate for the interconnection link speed. The first step is an analytical method that provides an initial estimate for the interconnecting link speed. We refer to this estimate as a new metric called the network bandwidth requirement, or *nbr*. The units for the *nbr* are words per cpu cycle, or w/c. The second step is simulation.

Using the estimate for the interconnecting link speeds obtained in the analytical portion, models based on stochastic activity networks, or SANs [3] are used to simulate the multiprocessor. The processor, interconnection network and memory are modeled providing performance results for each part of these parts. The simulation is used to verify results obtained in the analytical step, and to explore contention and memory access characteristics.

EQUATION DEFINITIONS

The purpose of these analytical equations is to provide a speed estimate for the interconnection links. Four factors have been identified for this analysis: (1) cache line size, (2) memory cycle, (3) contention, and (4) data consistency issues. The analytical equations incorporate cache line size to account for transferring a cache line, and data consistency for the traffic generated to maintain the consistency of data in the system. Conspicuous in their absence are contention and memory cycle, both are deferred to the simulation phase. For equation development, the memory is assumed to have zero delay. To compensate for the effects of contention, the processors in the system are assumed to issue data requests 100% of the time, the actual probability of a data request should be somewhat less than one. In an actual network a percentage of these transactions will be rejected, but so long as the sum of the probability of rejecting a request and the actual probability of a request do not exceed one, the equations compensate for not factoring in contention and will provide an accurate estimate.

Overhead

Overhead represents the average number of transactions needed to maintain data consistency for

Proceedings of the International Conference on Parallel Processing (ICPP'93)
0-8493-8983-6/0-7 $20.00 © 1993IEEE

each memory transaction initiated. A representative scheme was developed through a survey of directory based coherency schemes [4] and used to develop these equations. The analysis separates consistency transactions from data transfers to simplify equation development. Hierarchical interconnection network characteristics are incorporated into the equations, but will not be discussed. Units for all the overhead equations are words per memory request, or w/mr.

Overhead generated by reading dirty shared data. When a processor attempts to read dirty shared data the dirty item may be stored in the processor cache, or it may be stored elsewhere. In either case a dirty item is considered unusable. For the reading of dirty shared data the hit ratio is unimportant, because an attempt to read dirty data contained in a processor's cache is a forced miss. In the representative scheme an attempt to read dirty shared data is accomplished in *three* steps. First, the processor cache pair issues a read to main memory. Second, upon receipt of read request, main memory recognizes that the data is dirty, and forwards the request to the processor with the clean copy, which sends the cache line containing the data to the processor that initially requested the data. This is a net of *three* transactions. However, since data transfers are considered separately, overhead due to reading dirty shared data is(oh_{rds}):

$$oh_{rds} = 2 \times P_r \times P_s \times P_d \quad \text{w/mr.} \qquad (1)$$

For each instruction issued, P_r is the probability an instruction is a read. Of the data being read, P_s is the probability it is shared and P_d is the probability that the shared data is dirty. For every read of this type, *two* one word transactions are generated.

Similarly, equations for overhead generated by: reading shared data (oh_{rs}), reading unshared data (oh_{ru}), writing shared data ($ohinv_i$), and writing unshared data (oh_{wu}) were developed.

$$oh_{rs} = P_r \times P_s \times (1 - P_d) \times (1 - h_i) \quad \text{w/mr.} \quad (2)$$

$$oh_{ru} = P_r \times (1 - P_s) \times (1 - h_i) \quad \text{w/mr.} \quad (3)$$

$$ohinv_i = (1-P_r)\times(2P_s(1+S*N))\times(1-\frac{C_i}{N}) \quad \text{w/mr.} \quad (4)$$

$$oh_{wu} = (1 - P_r) \times (1 - P_s) \times ((1 - h_i) + 2h_i) \quad \text{w/mr.} \quad (5)$$

h_i is the global hit ratio of the caches up to level i, S is the percentage of processors sharing data, N is the number of processors, and C_i is the number of processors serviced by a cache at level i.

By summing these equations a unique representation of the overhead at level i may be obtained. This results in

$$oh_i = oh_{rds} + oh_{rs} + oh_{ru} + ohinv_i + oh_{wu} \quad \text{w/mr.} \quad (6)$$

Data transfers. A data transfer must be initiated for every *miss* and also every *hit* on shared dirty data. For the shared dirty data case, *two* data transfers are initiated. First the processor sends a clean copy to the main memory, which forwards a copy to the requesting processor. Factoring in instruction fetches, which incur no overhead, and data transfers results in the following equation:

$$D_i = 2 \times ((1 - h_i) + h_i \times P_s \times P_d) \quad \text{t/mr.} \quad (7)$$

The units are data transfers per memory request, or t/mr.

Network bandwidth requirement - *nbr*.

We now introduce a new metric called the network bandwidth requirement, or *nbr*. The *nbr* is the average number of words per cpu cycle that each link should be capable of transferring to or from each source. For single level interconnection networks, the only source is the processor private cache pair. The equation for the *nbr* was derived through a realization of factors that will increase traffic on an interconnecting link. Each processor contributes $D_i L_i + oh_i$ transactions into the interconnection network where L_i is the cache line size in words. The overall network bandwidth requirement for each link is affected by: the number of sources per link (N_{L_i}), the average path length, P_{L_i}, that a transaction travels, the number of processors covered by the cache at this level, C_i, and the number of instructions each processor can execute per cycle (nbr_0). The P_{L_i} calculations are based on characteristics of the interconnection network [5]. This yields the following equation:

$nbr_i = nbr_0 \times P_{L_i} \times N_{L_i} \times C_i \times (D_i \times L_i + oh_i)$ w/c.

ANALYTICAL RESULTS

The results of the equations were generated using representative values for the different parameters found in literature [6, 7] and plotted. Networks evaluated include a bus, crossbar, Multistage Interconnection Network (MIN), and a 2D mesh. Analysis of the graphs showed that the curves maintain a consistent shape for each of the interconnection networks the *nbr* values are of course different. Figure 1 shows the typical shape of the curves for the *nbr* results. The influence each parameter has on interconnection network traffic relative to each other changes significantly as the multiprocessor size grows. The slopes of these lines increases and decreases with the number of processors. For small multiprocessors, the parameter P_r exerts the greatest influence on interconnection network performance. However, as the number of processors grows, the effect of P_s, and S on the interconnection network performance also grows, eventually becoming the dominant parameters. This information should be a clue to designers

Proceedings of the International Conference on Parallel Processing (ICPP'93)
0-8493-8983-6/0-7 $20.00 © 1993IEEE

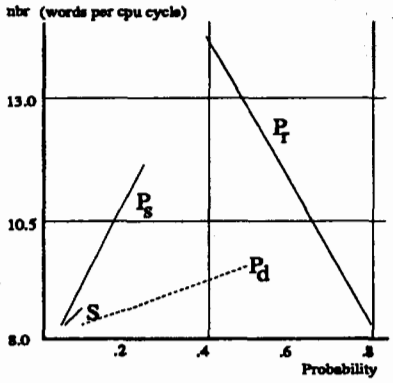

Figure 1: Typical curves for *nbr* vs. P_r, P_d, P_s, and S for a *64*-processor multiprocessing system.

as to which parameters need to be addressed in the design of multiprocessors.

The *nbr* results are graphed in Figures 2, and 3.

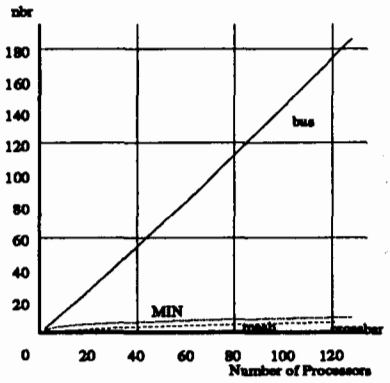

Figure 2: *nbr* vs. the number of processors. All network types are graphed.

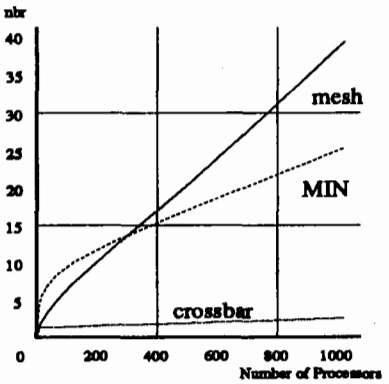

Figure 3: *nbr* vs. the number of processors. All network types are graphed.

As expected the bus is the worst performer, and the crossbar the best. The bus is the least capable interconnection strategy, but it is still a feasible choice for very small numbers of processors. The mesh is a better performer than the MIN up to about 300 processors, but the difference is not substantial. A designer must remember that performance is not the only factor to consider in choosing an interconnection network.

The conversion of these *nbr* results to an interconnecting link speed is very simple. The *nbr* represents the number of one word transactions each link must be capable of handling in a cpu cycle to sustain processor execution. Therefore, if the cpu has a 20MHz clock cycle, the interconnecting links should have a cycle of 20 x *nbr* MHz. Obviously, it is impossible to operate interconnecting links at some of the speeds indicated given current technological constraints, these configurations may be eliminated from further consideration. Once impossible configurations have been eliminated, simulation can be used to further investigate the remaining candidate networks.

SIMULATION RESULTS

To verify the results of the analytical section, the tool *UltraSAN* [3] was used. *UltraSAN* is a modeling and simulation tool based on stochastic activity networks, or SANs [3]. As a gauge for measuring the performance of the processors, a single processor bus was used. The interconnecting link speed used in this model evaluation was assumed to be equal to the processor speed. This uniprocessor system achieved a 66% processor utilization, a 12% memory utilization and a 3% link utilization. For all graphs of processor performance data a light line was included for the processor utilization of the uniprocessor bus model as a reference to this gauge. Furthermore, all results are graphed with their corresponding error bars. Some error bars may not be visible. This indicates that the simulation results are very accurate. Simulation models for all networks mentioned previously were developed and with the corresponding *nbr* values used to generate the results.

Performance results were obtained for processor, memory, and link utilizations. An accurate prediction of the *nbr* will result in a processor utilization that is equal to the gauge for all multiprocessor types and sizes.

Figures 4, 5, 6, 7, show the simulation results for the bus, crossbar, MIN, and mesh interconnection networks respectively. The *nbr* is a very good estimate, maintaining the processor utilization for all interconnection networks at a level close to the gauge. The memory utilization also remains at a fairly constant level of performance. The steady levels of performance of the processor and memory utilization demonstrate that the *nbr* value accounts for expected increases in the interconnection network traffic. If it had not, the utilization factor for both would drop since the interconnection network would necessarily have a higher utilization factor. The link utilizations for all models except the crossbar also remain essen-

Proceedings of the International Conference on Parallel Processing (ICPP'93)
0-8493-8983-6/0-7 $20.00 © 1993IEEE

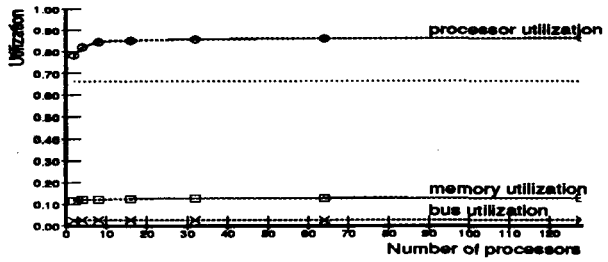

Figure 4: Simulation results for the bus multiprocessor with the capability of interleaving bus transactions.

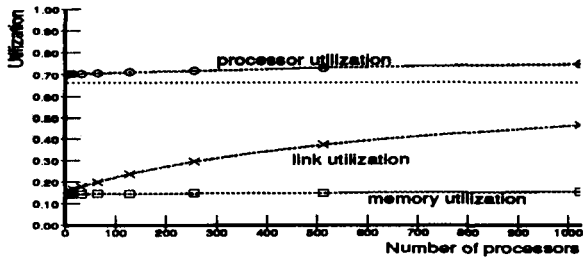

Figure 5: Simulation results for the crossbar multiprocessor model.

tially constant, which is in accordance with the previous statement. The link utilization in the crossbar increases with the number of processors, and is attributed to the invalidation/acknowledgement traffic and the method by which the link utilization is calculated.

CONCLUSIONS

The *nbr* calculations are shown to be accurate and flexible. Processors with different characteristics may be considered easily helping the designer to determine the best possible mating of processor to interconnection network strategy to obtain maximum performance within the current technological constraints. It can also be determined if the *nbr* of an interconnection network exceeds what is required by a given processing load.

Using the analytical and simulation phases in concert offers excellent aid to designers for the selection

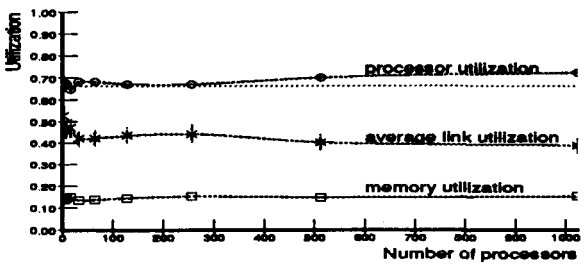

Figure 6: Simulation results for the MIN multiprocessor model.

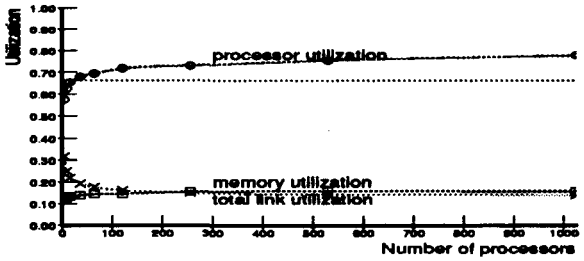

Figure 7: Simulation results of the mesh multiprocessor model.

of an interconnecting network and the determination of the interconnecting link speed requirements of a multiprocessing computer. The ease of use, the accuracy of the results, and the small time investment required to obtain the results makes this two-phase analysis an ideal starting block for multiprocessor design.

ACKNOWLEDGEMENT: We acknowledge the contributions of Dr. William Sanders and the *usan* group for assisting us in the use of *UltraSAN*.

References

[1] L. N. Bhuyan, Q. Yang, and D. Agrawal, "Performance of Multiprocessor Interconnection Networks," *IEEE Computer Magazine*, pp. 25–37, February 1989.

[2] A. Agrawal, "Limits on Interconnection Network Performance," *IEEE Transactions on Parallel and Distributed Systems*, vol. 2, pp. 398–412, October 1991.

[3] J. A. Couvillion, R. Freire, and et al., "Performability Modeling with UltraSAN," *IEEE Software Magazine*, pp. 69–80, Sept 1991.

[4] A. Agarwal, R. Simoni, and et al., "An Evaluation of Schemes for Cache Coherence," in *IEEE Computer Architecture Conference - 1988*, pp. 280–289, 1988.

[5] R. Duncan, "A Survey of Parallel Computer Architectures," *IEEE Computer Magazine*, pp. 5–16, February 1990.

[6] M. Tomašević and V. Milutinović, "A Simulation Study of Snoopy Cache Coherence Protocols," in *Hawaii International Conference on System Sciences*, pp. 427–436, 1992.

[7] J. Archibald and J.-L. Baer, "Cache Coherence Protocols: Evaluation Using a Multiprocessor Simulation Model," *ACM Transactions on Computer Systems*, vol. 4, pp. 273–298, November 1986.

Proceedings of the International Conference on Parallel Processing (ICPP'93)
0-8493-8983-6/0-7 $20.00 © 1993 IEEE