Economides, A.A. & Dubois, M.: **Transient models of bus-based multiprocessors.** *Proceedings 1990 International Conference on Parallel Processing*, pp. I.153-I.160, 1990.

TRANSIENT MODELS OF BUS-BASED MULTIPROCESSORS [1]

Anastasios A. Economides and Michel Dubois

Electrical Engineering - Systems Department
University of Southern California
Los Angeles, CA 90089-0781

Abstract– In this paper, the effect of the initial transient of parallel processes on the performance of bus-based multiprocessor systems is investigated. Each processor has a local memory or a cache and also accesses a centralized shared memory via a single bus. An application program is forked into parallel processes, each one running on a different processor. During the initial execution time, each process loads its working set from the shared memory and therefore the miss rate in the local memory is very high. This initial burst of memory accesses may saturate the bus interconnection and cause serious performance degradation.

We present simulations and three analytical models to evaluate the effect of the transient. The main question is to determine whether the transient effect can be neglected. The models are applied to a family of miss rate functions based on fractal statistics. We compare the precision of the three models with the simulations. Cases where the transient effect can be neglected are identified.

1. INTRODUCTION

Faster computation requirements can be satisfied by a faster single processor or/and by many processors working concurrently. It is known in queueing theory [7], that a single processor with speed C performs faster than P processors, each with speed C/P. When a workload is decomposed in multiple processes, these processes must communicate and synchronize their activities via message passing or shared memory. In shared-memory multiprocessors, a local memory or cache is needed to reduce the penalty and conflicts in accessing the global store (Fig. 1). Cache memories significantly enhance the performance of computer systems and in particular of bus-based shared-memory multiprocessors [2, 3, 6, 10]. The cache memory is a small high-speed memory between the main memory bus and the processor. It holds copies of recently referenced main memory locations. Every time a process running on a processor needs a memory word, it may find a copy in the cache with a very high probability (hit rate). If the reference is not in the cache, then a miss occurs and the processor is blocked while accessing the bus interconnection and the shared-memory (Fig. 2). This causes bus interference, since many processors may simultaneously access the bus interconnection.

[1] This work was supported by NSF grant No CCR 8709997.

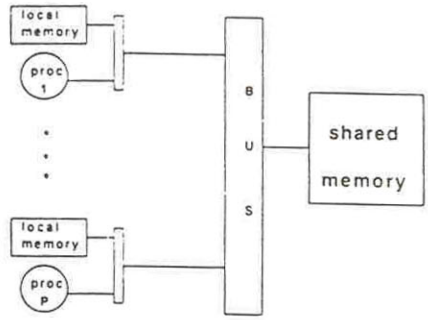


Fig. 1. Multiprocessor System Model.

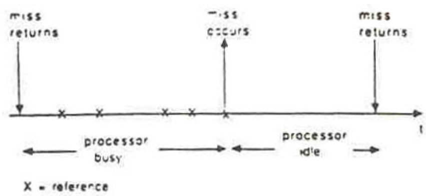


Fig. 2. Process Behavior.

Previous studies [4, 5, 8, 9, 14] for bus interference consider the steady-state behavior of the system. It is generally assumed that memory bus accesses are uniformly distributed over time (Poisson process) and Markov Chains or Queueing Network models [7] can be applied. However, when a new process starts running, its working set is not in the cache of its processor and therefore it accesses the global memory very often. The access rate on the bus interconnection may be so high as to cause saturation. So, although the average miss rate may not be enough to saturate the bus interconnection, the transient miss rate at the beginning of process execution may saturate it and cause severe performance degradation.

In the following, the effect of this process transient on the performance of bus-based multiprocessor systems is investigated. The prediction of three analytical models that consider this transient behavior of parallel processes are compared against simulations for a family of miss rate functions proposed in [13].

2. MULTIPROCESSOR MODELS

The architecture model is a multiprocessor system composed of P processors, each with a local cache interconnected via a single bus with a centralized shared memory (Fig. 1).

At time $t = 0$, the caches are empty and a parallel algorithm to be executed on the multiprocessor is forked into P concurrent processes that run simultaneously. Each process is assigned to one processor. Processors access their cache for instruction fetches, operand fetches or operand stores. Let τ be the average time between two successive references of a process, in the ideal case where no miss occurs in the cache. Throughout the paper, τ will be the time unit, i.e. $\tau = 1$. If the reference misses in the cache then a request is made to the shared memory via the memory bus. This request (miss) waits in a common queue for all processors, until it receives service from the memory bus according to a first-come-first-served discipline. We denote by $m(r)$ the instantaneous miss rate for a process at its r^{th} reference, and by $\bar{m}(r)$ the average miss rate for the first r references. The instantaneous miss rate $m(r)$ for a process at its r^{th} reference is the probability that the r^{th} reference misses, and we have $\sum_{i=1}^{r} m(i) = M(r)$, where $M(r)$ is the mean number of misses in a process up to its r^{th} reference. The average miss rate at r is $\frac{M(r)}{r}$. The service time D of the bus is constant and the bandwidth BW of the memory bus is the maximum number of misses that can be serviced per time unit.

Therefore, if a process has a total of R references, its total execution time $T(R)$ will not be $R * \tau$, but $R * \tau$ plus the extra time spent for requests (misses) on the memory bus. Let Z be the average time taken by each miss on the memory bus (response time); then the total average execution time will be $T(R) = R * \tau + M(R) * Z$. However, during the initial process execution period, the time that a miss spends on the memory bus is very large because of high contention, and later it decreases, as the local caches are filled with the working set of each process. The problem is to estimate $T(R)$ when Z varies with time.

We make the following additional hypotheses :

1) The P processes are homogeneous at all times. This means that the instantaneous miss rate $m(r)$ is the same for all processes.

2) The instantaneous miss rate $m(r)$ is a non-increasing function of r.

3) Cache sizes are infinite.

4) We neglect all coherence traffic.

5) The memory bus is the bottleneck; either the bus is circuit switched or, if it is packet switched, there are enough interleaved memory modules so that a memory access is never rejected at the memory module.

In practical situations the models and the simulation results presented in this paper are applicable to the parallel execution of a Fortran Do-loop by forking processes on different processors. In this case, the first hypothesis is often verified because the processes execute the same code but on different data. The second hypothesis may not be strictly verified in particular cases; however it corresponds to observed behavior of average programs. The infinite cache size hypothesis permits us to ignore replacements and the cache organization. The finite cache effects will be discussed further in Section 5. Since we neglect all coherence traffic, the results are more valid for *Doall* loops (with no dependencies between statements executed on different processors) than for *Doacross* loops (with dependencies among processes).

The first model is directly derived from a steady-state model.

2.1 QUEUEING MODEL

We approximate the behavior of the multiprocessor system using a closed queueing network model in steady-state (Fig. 3) [7]. Each processor executes locally for a time interval of mean $1/\bar{m}$ and then issues a request to the memory bus, where the mean service time is $1/\mu$.

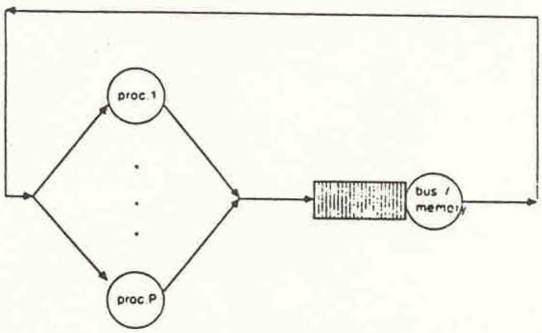


Fig. 3. Multiprocessor System Queueing Model.

For single-bus multiprocessors, with exponentially distributed interarrival time of requests for each processor with rate $\bar{m}$ and generally distributed service times on the bus, with mean service time $1/\mu$ and first-come first-served scheduling, the probability that the bus is idle is given by [7]

$$\pi_0 = \frac{1}{1 + P * \frac{\bar{m}}{\mu} + P * \frac{\bar{m}}{\mu} * \sum_{k=1}^{P-1} \binom{P-1}{k} * g(k)}$$

where $g(k) = \prod_{i=1}^{k} \frac{1 - L_{bus}(\bar{m} * i)}{L_{bus}(\bar{m} * i)}$, and $L_{bus}(s)$ is the Laplace transform of the service time distribution on the bus. For constant service time $\frac{1}{\mu} = D$, $L_{bus}(s) = e^{-sD}$.

Then the mean number of bus requests served per time unit is $X = (1 - \pi_0) * \mu$ and the response time of the bus is
$$Z = \frac{P}{X} - \frac{\tau}{\bar{m}} = \frac{P * D}{1 - \pi_0} - \frac{\tau}{\bar{m}}$$

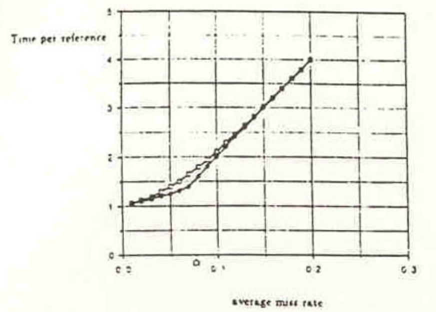


Fig. 4. Time per reference versus average miss rate, for $A = 2$, $\theta = 1.5$, $P = 4$.

Finally, the time required by each process to execute one reference is

$$t^q = \tau + \bar{m} * Z = \bar{m} * \frac{P * D}{1 - \pi_0}$$

In Fig. 4, we show the time per reference versus the average miss rate for a process.

Then, the total required time for each process to execute R references is $T^q(R) = R * t^q$.

Let also define the prediction error of the total time to execute R references using this model, $T^q(R)$, as

$$\frac{T^q(R) - T^s(R)}{T^s(R)},$$

where $T^s(R)$ is the total required time for each process to execute R references from the simulation.

The above queueing model is for a single bus (server) multiprocessor. Although there are closed form solutions for closed queueing networks with multiple servers, they require exponentially distributed service times (for first-come-first-served scheduling). Unfortunately, this assumption is not accurate for multiprocessors, where the memory access time is constant.

2.2 ASYMPTOTIC MODEL

The asymptotic model is based on a simple throughput analysis similar to the one in [5]. One interesting property of this model is that it yields an upper bound on the throughput and therefore a lower bound on the execution time.

Since the miss rate is a non-increasing function of r, bus saturation (if any) occurs at program start. If the bus is saturated at program start, either it remains saturated during the whole execution or there is a critical reference r^* such that the bus leaves saturation. In this asymptotic model, it is assumed that after r^*, bus conflicts can be neglected.

When the bus is saturated, the processors have their misses serviced in turn on the bus, at a rate per processor of $\frac{BW}{P}$. This means that the time between two consecutive misses in one processor is $\frac{P}{BW}$ and therefore, the total time to reach reference r^*, at which the bus leaves saturation is $\frac{P}{BW} * M(r^*)$.

When $r > r^*$, bus conflicts are neglected. $(R - r^*)$ references remain. Among these references, $M(R) - M(r^*)$ miss in the cache and each miss takes time D. Therefore the remaining processing time is $(R - r^*) * \tau + [M(R) - M(r^*)] * D$.

Finally, the total time required by a process to complete the execution of its R references is

$$T^1(R) = \begin{cases} \frac{P}{BW} * M(r^*) + (R - r^*) * \tau + [M(R) - M(r^*)] * D & \text{for } R > r^* \\ \frac{P}{BW} * M(R) & \text{for } R \leq r^* \end{cases}$$

We note that the time to execute a set of consecutive references between r_1 and r_2 is always lower bounded by $\frac{P}{BW} * [M(r_2) - M(r_1)]$, which is the time needed to execute $M(r_2) - M(r_1)$ misses in each processor if the bus is 100% utilized. $(r_2 - r_1) * \tau + [M(r_2) - M(r_1)] * D$ is also a lower bound, because the processor is blocked during the execution of the $M(r_2) - M(r_1)$ misses. Therefore the expression for $T^1(R)$ above is always a lower bound on the execution time, whatever the value of r^*. Note that if we apply this model without accounting for the transient as was done in [5], then we simply select r^* as 1 or ∞, depending on the average miss rate $\bar{m}$. In the presence of a transient, we have found that a tighter lower bound can be obtained by computing r^* as follows.

To evaluate r^*, we equate the times required by reference r^* for the saturated and the no conflict cases. When the bus is saturated, the number of accesses executed per processor is $\frac{BW}{m(r) * P}$, where BW is the bus system bandwidth in access per time unit and $m(r)$ is the instantaneous miss rate at reference r. Therefore, $\frac{P}{BW} * m(r^*) = \tau + m(r^*) * D$.

Therefore, for $BW * D < P$, $m(r^*) = \frac{BW * \tau}{P - BW * D} = \alpha$.

This equation can be solved graphically for r^* once we have defined a way of estimating $m(r^*)$. We note that for $BW * D \geq P$ the bus interconnection is never saturated, even when the miss rate is equal to 1. Those cases are not interesting because in practice the memory bus has been designed for the average traffic corresponding to the average miss rate, which is much smaller than 1. Since $0 < m(r^*) < 1$, we must have $D < \frac{P - BW * \tau}{BW}$.

The prediction error of the total time required to execute R references using this model, $T^1(R)$, is $\frac{T^1(R) - T^s(R)}{T^s(R)}$.

2.3 LINEAR APPROXIMATION MODEL

In the previous model, the execution time is always underestimated, for all r. The linear approximation model is similar to the asymptotic model, but the response time at all r is estimated by a linear approximation of the response time of the queueing network in steady-state. Consider the curve of the time between references as a function of the instantaneous miss rate. Then, we approximate the curve of the time per reference when the miss rate ranges from $m(1) = 1$ to $m(r^*) = \alpha$ by a straight line

$$t^2_{1,r^*}(r) = t^q(\alpha) + \frac{t^q(1) - t^q(\alpha)}{1-\alpha} * (m(r) - \alpha)$$

where $t^q(\alpha)$ and $t^q(1)$ will be found from the queueing model, for $\bar{m} = \alpha$ and 1.

Summing over all r's, from $r = 1$ to $r = r^*$, we obtain:

$$T^2_{1,r^*} = t^q(\alpha) * r^* + \frac{t^q(1) - t^q(\alpha)}{1-\alpha} * (M(r^*) - \alpha * r^*)$$

Similarly, we approximate the curve of the time per reference when the miss rate is from $m(r^*) = \alpha$ to $m(R)$ by a straight line

$$t^2_{r^*,R}(r) = t^q(m(R)) + \frac{t^q(\alpha) - t^q(m(R))}{\alpha - m(R)} * (m(r) - m(R))$$

where $t^q(\alpha)$ and $t^q(m(R))$ will be found from the queueing model, for $\bar{m} = \alpha$ and $\bar{m} = m(R)$.

Therefore the total time from reference r^* to R is obtained by summing $t^2_{r^*,R}$ over all r's from $r = r^*$ to $r = R$.

$$T^2_{r^*,R} = t^q(m(R)) * (R - r^*) + \\ + \frac{t^q(\alpha) - t^q(m(R))}{\alpha - m(R)} * [M(R) - M(r^*) - m(R) * (R - r^*)]$$

It follows that the total required time to execute R references is

$$T^2(R) = \begin{cases} t^q(\alpha) * r^* + \frac{t^q(1) - t^q(\alpha)}{1-\alpha} * (M(r^*) - \alpha * r^*) + t^q(m(R)) * (R - r^*) & \\ \quad + \frac{t^q(\alpha) - t^q(m(R))}{\alpha - m(R)} * [M(R) - M(r^*) - m(R) * (R - r^*)] & for \ R > r^* \\ t^q(\alpha) * R + \frac{t^q(1) - t^q(\alpha)}{1-\alpha} * (M(R) - \alpha * R) & for \ R \leq r^* \end{cases}$$

The prediction error of the total time required to execute R references using this model, $T^2(R)$, is $\frac{T^2(R) - T^s(R)}{T^s(R)}$.

3. APPLICATION TO A SYNTHETIC WORKLOAD

In the previous section, we have introduced approximate models for the total execution time of parallel processes. The models incorporate the variable miss rate due to the initial transient. To test the models and estimate the transient effect, we need to know the function $M(r)$. This function can be obtained from traces of actual programs. However, several analytical models of cache transient behavior have been proposed, for example by Strecker [11], by Agarwal, Horowitz & Hennessy [1], by Thiebaut, Stone & Wolf [13] and by Thiebaut [12]. The advantage of such models is that the behavior of many programs with various locality and transient characteristics can be easily generated. Thiebaut's model [12] is particularly simple to use because there are only two parameters. The model in the form presented here is valid for infinite caches only.

In Thiebaut's model, for infinite cache size, the number of misses for a process until its r^{th} reference is given by a fractal process [13] $M(r) = A * r^{\frac{1}{\theta}}$, where θ is the fractal dimension of the process (it can also be viewed as a measure of the locality of reference). When $\theta \rightarrow 1$, almost every reference misses, i.e. the process has very little locality. When $\theta \rightarrow 2$, the locality of process references is very high. A is a small number.

Note, that $M(r) \leq r$, i.e. the number of misses must be less than the number of references [13]. So, when a new process is loaded, it misses on every reference up to a critical reference $r_c = A^{\frac{\theta}{\theta-1}}$. Therefore, $M(r) = \begin{cases} r & \text{if } r \leq r_c \\ A * r^{\frac{1}{\theta}} & \text{if } r > r_c \end{cases}$

In general r_c is very small, and therefore we neglect this problem in the analytical model and we consider only the general form of the curve $M(r) = A * r^{\frac{1}{\theta}}$.

The average miss rate for the first r references is also given by $\bar{m}(r) = \frac{M(r)}{r} = A * r^{\frac{1}{\theta-1}}$.

To apply the model, we need to compute the instantaneous miss rate $m(r) = \frac{\partial M(r)}{\partial r} = \frac{A}{\theta} * r^{\frac{1}{\theta}-1}$. Finally, $m(r^*) = \alpha \Rightarrow r^* = (\frac{\alpha * \theta}{A})^{\frac{\theta}{1-\theta}}$. From this value r^*, the asymptotic and linear approximation models are easily derived.

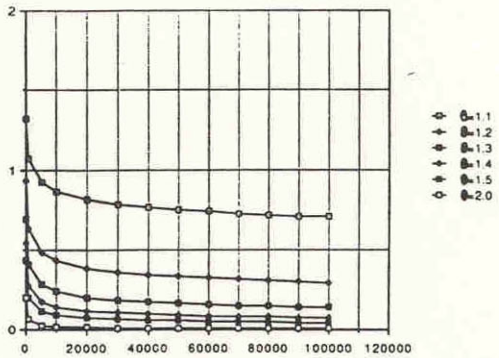


Fig. 5. The average miss rate $\bar{m}(r)$ versus the number of references r for $A = 2$ and different values of θ.

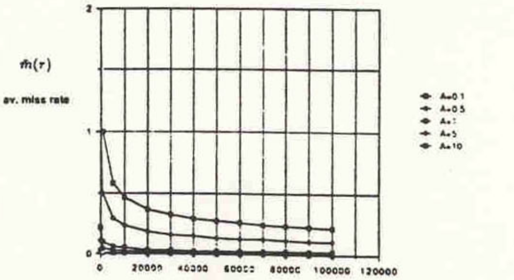


Fig. 6. The average miss rate $\bar{m}(r)$ versus the number of references r for $\theta = 1.5$ and different values of A.

In Fig. 5, we show the average miss rate $\bar{m}(r)$ versus the number of references r for $A = 2$ and different values of θ. In Fig. 6, we show the average miss rate $\bar{m}(r)$ versus the number of references r for $\theta = 1.5$ and different values of A.

By varying θ in the model, we can modify the shape of the transient. For each value of θ, the value of A can be chosen to change the miss rate uniformly as a function of time. The value of A simply "scales" the miss rate curve.

4. SIMULATION

In the simulations, there are P identical processors connected to a single circuit-switched bus, so that $BW = 1/D$. The simulation mimics the execution of a parallel program with P parallel identical processes, each one running on a different processor. At $t = 0$, the P processes are forked and their access pattern to the memory is approximated by Thiebaut's model. The intereference time for each process is the time unit ($\tau = 1$). The service time of each bus request is constant ($D = 5$). If the response time of the memory bus, D, is different from 5, the results of the simulation can still be used by simply changing the time unit and the value of A. For example, if the service time of the memory bus is equal to 10 intereference times, then we change the unit τ to be the time for 2 consecutive processor references. We also have to change $M(r)$ to $M'(r)$ so that

$$M'(r) = M(2r) = A * (2 * r)^{\frac{1}{\theta}} = A * 2^{\frac{1}{\theta}} * r^{\frac{1}{\theta}} = A' * r^{\frac{1}{\theta}}$$

With $A' = A * 2^{\frac{1}{\theta}}$ as the new value of A, the simulation results with $D = 5$ are applicable to the case $D = 10$.

In the simulations, for the first r_c references each process misses at every reference. After r_c, the time between misses for a process is calculated as follows: let the M^{th} miss for a process occur at its r^{th} reference. Then its processor sends a request to the bus, where the request waits in a single first-come-first-served queue. After this miss has been served, the next miss for this process will occur exactly after $\lceil (\frac{M+1}{A})^{\theta} - r \rceil$ references, where A and θ are the parameters of the miss rate function. Note that although during the initial execution period, all the processes miss at the same time, this synchronization has no effect on the performance, since the memory bus is always saturated. The processes become asynchronous because of the contention on the memory bus.

Our first objective is to find how closely our analytical models of section 2, predict the total execution time of a process compared to the simulation results, for different miss rate functions. Before we present our results, let us briefly discuss the overall picture, since we do not show all of our results due to space limitation.

First consider the case of small A and θ, for example $A \leq 1$ and $\theta \leq 1.6$ when $P = 4$, or $A \leq 0.5$ and $\theta \leq 1.4$ when $P = 8$. When the average miss rate is greater than α, ($\alpha = 0.0666$ when $P = 4$, $\alpha = 0.02857$ when $P = 8$), then the linear approximation model is better. When the average miss rate is around α, then the queueing model is better. When the average miss rate is smaller then α, then the asymptotic model is better. Finally, when the average miss rate becomes very small, then there is hardly any bus contention and all three models predict very accurately the simulation results.

Let consider next the case of large A. When the average miss rate is greater than α, then the linear approximation model is better. It is still better when the miss rate is close to α; when the miss rate becomes smaller than α, then the queueing model is better. For very small miss rate again all models predict very accurately the simulation results.

Finally, for the case of larger θ (higher locality), the linear approximation model tends to be better for almost all average miss rates. Again, for very small miss rate all models predict very accurately the simulation results.

In Fig. 7 and 8, we report some results on the prediction error of all three models for $P = 4$ and 8. This is the relative error on the total execution time as defined in section 2. Different values of A are picked between 0 and 5. The values of θ are set so that the average miss rate for $R = 100,000$ references does not result in bus saturation (i.e. is less than α). By varying R from 500 to 200,000 we are therefore able to observe the transient effect under various average conditions, from deep saturation of the memory bus to very light loading of the bus, and for different shapes of the transient curve.

Fig. 7 and 8 show the relative error of each analytical model versus the ratio $\bar{m}/\alpha$. Note that each curve corresponds to a trace and r increases from right to left ($r = 500$ corresponds to the largest value of $\bar{m}$ and $r = 200,000$ corresponds to the smaller value of $\bar{m}$). The maximum relative error is less than 20% in all cases and for all three models. Of the three models the linear approximation model is better most of the time. In Fig. 9 and 10, the relative error for the three models is plotted versus A, for $r = 10,000$ and 100,000 respectively.

The effect of the transient could be measured by the relative error of the queueing model, because similar models have been shown to predict the steady-state behavior with high accuracy. From all the curves that we have derived, it appears that this effect is never more than 20% of the total execution time. So, provided the *average miss rate* is used in the queueing model, it is good within 20% . While more sophisticated models than the queueing model can be derived, dramatic improvement on the precision cannot be expected. The interest of the asymptotic model is that it is quite accurate, always yields a lower bound on the execution time and is applicable to other types of interconnection besides the single bus. The interest of the linear approximation model is that it is the most accurate in most cases.

From Fig. 7 and 8, it appears that the transient is negligible if the average miss rate in the trace is less than α. If the average miss rate is higher than α, the transient effect can contribute between 10 to 20 % to the total execution time. In those cases, the linear approximation model is always the best. Note that these conclusions hold for all possible shapes of the transients and for all memory bus service times (for different service times we simply change the unit and the value of A as explained above).

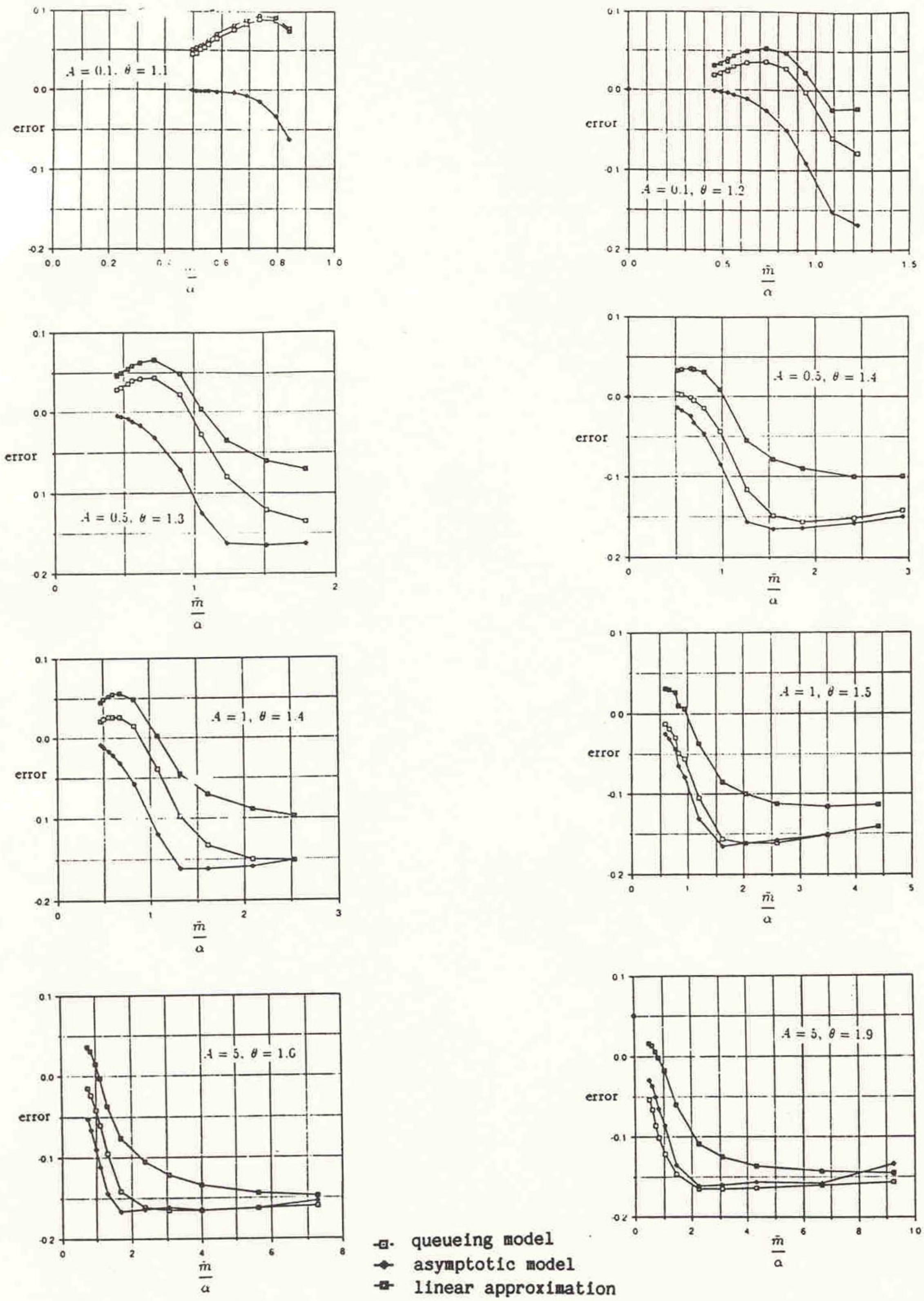
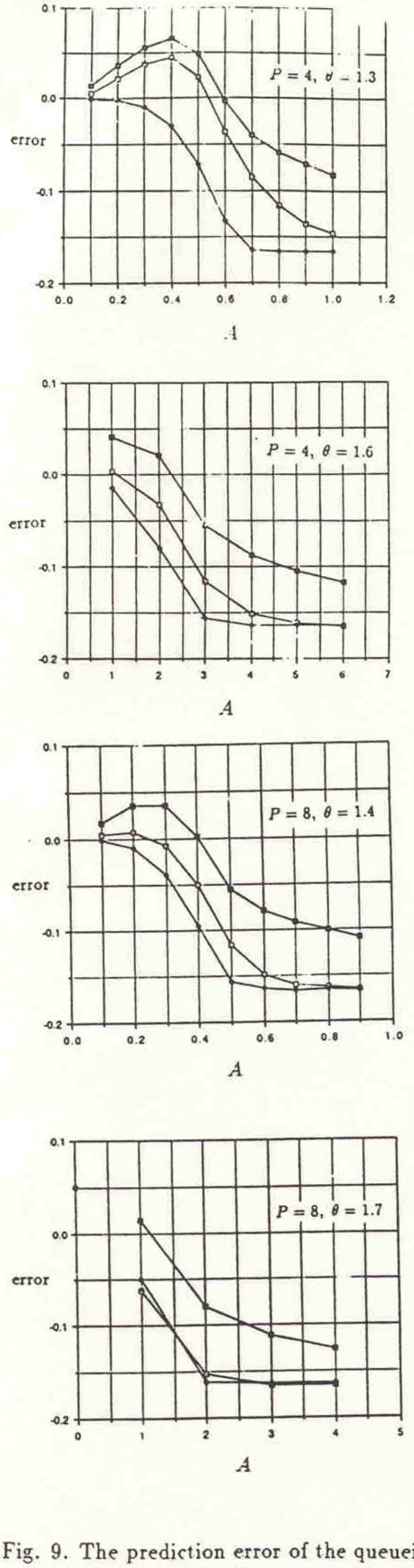
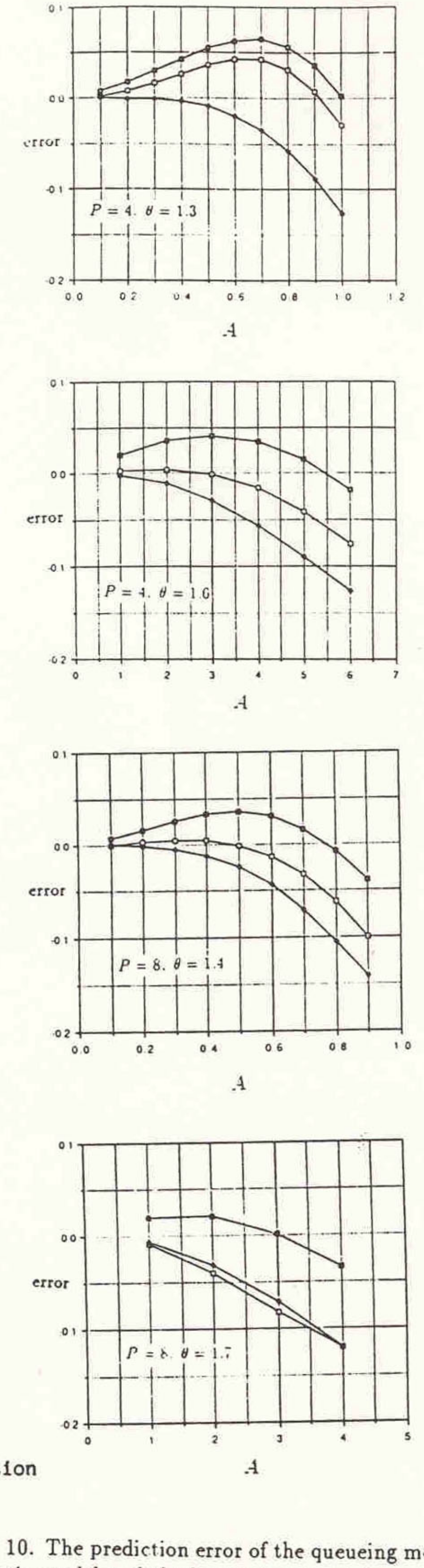


Fig. 7. The prediction error of the queueing model, the asymptotic model and the linear approximation model versus the average miss rate over α, for $P = 4$ ($\alpha = 0.0666$).

Fig. 8. The prediction error of the queueing model, the asymptotic model and the linear approximation model versus the average miss rate over α, for $P = 8$ ($\alpha = 0.028571$).





queueing model
asymptotic model
linear approximation

Fig. 9. The prediction error of the queueing model, the asymptotic model and the linear approximation model versus A, for $R = 10.000$.

Fig. 10. The prediction error of the queueing model, the asymptotic model and the linear approximation model versus A, for $R = 100.000$.

5. FINITE CACHE EFFECTS

In all the models, we have assumed infinite cache sizes. This has simplified the analysis because there was no cache replacement and the results are independent of the cache organization and replacement policy. In practice, this approximation means that the cache size is large with respect to the number of different cache blocks accessed by the process.

If the cache is very small, then the working set of the process will fill the cache before the end of the execution. Once the cache is filled, a steady-state behavior is reached for the hit rate. However, caches in today's systems are so large that multiple contexts can reside in cache and one given process cannot fill up the cache during its time quantum. In these situations, when a new process runs, it displaces some blocks of previous contexts, i.e. some replacements will occur during the transient because of the finiteness of the cache size. This effect is very difficult to predict because it depends on the behavior of previously running contexts. Clearly to include such effects, the definition of $M(r)$ must be replaced in the three models by the total number of bus access, $A(r) = M(r) * [1 + w(r)]$, where $w(r)$ is the probability that a miss displaces a dirty block, and $m(r)$ must be replaced by the instantaneous bus access rate $a(r) = \frac{\partial A(r)}{\partial r}$ in the computation of r^*.

6. CONCLUSIONS

In this paper, we have investigated the *effect of the transient behavior of parallel processes on the performance of a shared memory single-bus multiprocessor system.* Because the miss rate is variable with time, the basic hypotheses behind the steady-state queueing model are violated.

We have *introduced two transient models* for the total process execution time, which consider the burstiness of process miss rates during the beginning of process execution. We have *compared all analytical models to simulation, using a dynamic miss rate model* which allows us to explore various shapes of the transient curves. For most miss rate functions, the proposed models are very close to the simulation. The asymptotic model is always a lower bound on the execution time and can be applied to systems other than single-bus multiprocessors.

Finally, we can conclude that if the multiprocessor has been designed such that the average miss rate of the typical workload is much less than the critical miss rate α, then the transient effect can be neglected; on the other hand, if the computation time is short, so that the average miss rate of a process is larger than the critical miss rate α of the multiprocessor, then the total process execution time will be longer than that predicted using steady-state queueing models. In these cases, the linear approximation model predicts more accurately the process execution time.

Further work is needed to generalize these conclusions to finite caches, including replacements. Also the results of tracing parallelized do loops would be useful to corroborate our conclusions.

References

[1] A. Agarwal, M. Horowitz, and J. Hennessy. An analytical cache model. *ACM Transactions on Computer Systems*, Vol. 7, No. 2, pp. 184-215, May 1989.

[2] J. Archibald and J. L. Baer. Cache coherence protocols : evaluation using a multiprocessor simulation model. *ACM Trans. on Computer Systems*, 4(4), pp. 273-298, Nov. 1986.

[3] F. A. Briggs and M. Dubois. Effectiveness of private caches in multiprocessor systems with parallel-pipelined memories. *IEEE Trans. on Computers*, Vol C-32, No 1, pp. 48-59, Jan. 1983.

[4] G. Chiola, M. A. Marsan, and G. Balbo. Product-form solution for the performance analysis of multiple-bus multiprocessor systems with nonuniform memory references. *IEEE Trans. on Computers*, Vol C-37, No 5, pp. 532-540, May 1988.

[5] M. Dubois. Throughput analysis of cache-based multiprocessors with multiple buses. *IEEE Trans. on Computer*, Vol C-37, No 1, pp. 58-70, Jan. 1988.

[6] M. Dubois and F. A. Briggs. Effects of cache coherence in multiprocessors. *IEEE Trans. on Computers*, Vol C-31, No 11, pp. 1083-1099, Nov. 1982.

[7] S. S. Lavenberg. *Computer Performance Modeling Handbook.* Academic Press, 1983.

[8] M.A. Marsan and M. Gerla. Markov models for multiple bus multiprocessor systems. *IEEE Tr. on Computers*, Vol. C-31, No. 3, pp. 239-248, March 1982.

[9] T.N. Mudge and H.B.Al-Sadoun. A semi-markov model for the performance of multiple bus systems. *IEEE Tr. on Computers, Vol. C-34, No. 10, pp. 934-942*, Oct. 1985.

[10] J. H. Patel. Analysis of multiprocessors with private memories. *IEEE Trans. on Computers*, Vol C-31, No 4, pp. 296-304, Apr. 1982.

[11] W. D. Strecker. Transient behavior of cache memories. *ACM Trans. on Computer Systems*, Vol 1, No 4, pp. 281-293, Nov. 1983.

[12] D. Thiebaut. On the fractal dimension of computer programs and its application to the prediction of the cache miss ratio. *IEEE Trans. on Computers*, Vol. 38, No. 7, pp. 1012-1026, July 1989.

[13] D. F. Thiebaut, H. S. Stone, and J. L. Wolf. A theory of cache bahavior. *IBM Research Report 13309(59338)*, 11/24/1987.

[14] D. Towsley. Approximate analysis of multiple bus multiprocessor systems. *IEEE Tr. on Computers*, Vol. C-35, No. 3, pp. 220-228, March 1986.