Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

AN IMPORTANT FACTOR FOR OPTIMISTIC PROTOCOL ON DISTRIBUTED SYSTEMS: GRANULARITY

Eunmi Choi
Moon Jung Chung

Department of Computer Science
Michigan State University
East Lansing, MI 48824-1027, U.S.A.

ABSTRACT

Grain size, the amount of computations between communication points, is a quantity to be tuned appropriately depending on the characteristics of the underlying parallel and distributed machines, application problems, and simulation protocols. As target machines for optimistic protocol, the architectural characteristics of a cluster of DEC Alpha workstations are compared to the MasPar MP-2's in view of parallel logic simulation. We study the effects of varying grain size on several performance metrics when more than one logical processes are assigned to a physical processor on distributed systems. We obtain analytic formulas for the total number of simulation cycles and the total execution time, and find the optimal grain sizes for several benchmark circuits when the number of processors varies. Our experimental results show that the grain size greatly affects the performance of parallel logic simulation on distributed systems, and the effects vary depending on the machine independent factors.

1 INTRODUCTION

Logic simulation is a necessary step for verification in the context of VLSI design and development. *Parallel Discrete Event Logic Simulation* (PDELS) on massively parallel computers and distributed systems has been a researchers' choice for achieving significant reduction in simulation execution time. This approach can achieve a high degree of parallelism by concurrently processing events.

In this paper, two different types of parallel architecture are compared as platforms for gate-level PDELS. Due to the inherent characteristics of parallel gate-level logic simulation, such as highly parallel processing of a large number of gates and frequent communications between gates, a suitable platform provides many processors to execute events at the same time and fast interprocessor communications. *Massively parallel SIMD (Single Instruction stream over Multiple Data streams)* architectures are in this category. A problem of SIMD platform is that the size of the local memory is not enough for the event queue space when several gates are assigned to a processor. An alternative platform for PDELS of huge circuits is a *distributed system* or a *cluster of workstations.* In contrast to a PE in the SIMD machines, a workstation in distributed systems has a large local memory so that a large number of logic gates or *(logical) processes* can be assigned to a processor. In this paper, we call the number of logical processes allocated to a processor the *LP ratio.* This ratio means the ratio of the number of logical processes to the number of processors in the system.

One of the interesting research issues for optimistic protocol on distributed systems is to decide how many processes are to be processed between consecutive communications among the processes activated by events. This quantity is called in this paper *granularity* or *grain size.* Since the LP ratio on distributed systems is greater than one, processors may have various numbers of activated processes. If all activated processes are executed on every processor, highly loaded processors may be the bottleneck of the simulation, resulting in increasing the total execution time of the simulation. As restricting the maximum number of executed processes among the activated processes, proper tuning of the granularity can contribute to minimizing the total execution time and the simulation cost.

The primary goals of this paper are to help choose machine architectures for parallel logic simulation and to study the effects of granularity on the performance of optimistic protocol on distributed systems. We compare the architectural characteristics of the MasPar MP-2 to a DEC Alpha workstation cluster's. According to the comparison, the MP-2 gives approximately four times faster performance than the distributed system for small circuits, while the distributed system has the advantage of large memory for huge circuits that cannot be simulated on the MP-2. Based on the model described in Section 3, we examine the performance of large benchmark logic circuits, while varying grain size on a cluster of DEC Alpha workstations interconnected by DEC GIGAswitch. Parallel Virtual Machines (PVM) (Oak Ridge National Laboratory 1994) are employed for distributed parallel processing. To verify the experimental results and study the effects of machine-dependent and machine-independent parameters, we derive analytic formulas to predict performance metrics for various grain sizes. Both of experimental and analytical results present that the grain size is an important parameter to optimize the performance of distributed systems for PDELS of large circuits. In addition, we find optimal grain size for several benchmark circuits when the number of processors varies. For the study, we draw conclusions that distributed systems are promising systems for logic simulation with large circuits and granularity is an important factor to be considered.

Agrawal and Chakradhar (1992) analyzed the performance of the synchronized iterative algorithms based on the binomial distribution on their analytic model. In their analytic model, the grain size is fixed to the value of the LP ratio and the interprocessor communication time was ignored. Chung and Chung (1992) presented a model to predict parallelism and the number of simulation cycles for parallel logic simulation. The grain size of their model was fixed to one or the LP ratio.

The remainder of this paper is organized as follows. In the following section, the characteristics of two target machines are compared in view of PDELS. Section 3 describes the model that is used for experiments and analysis. In Section 4, we analyze the total execution time. Section 5 presents our experimental and analytical results on the effects of grain size. Concluding remarks are given in the last section.

2 CHARACTERISTICS OF TARGET MACHINES

The target machines used for our simulation are the MasPar MP-2 that has 4K PEs and a cluster of 6 DEC Alpha 3000 workstations interconnected by a DEC GIGAswitch through FDDI.

2.1 Interprocessor Communication and Global Synchronization

One of better characteristics of the MP-2's architecture is that it has built-in global router connections for interprocessor communications. The global router is a bidirectional communication instruction and a circuit-switched style network organized as a three-stage hierarchy. In contrast, a cluster of workstations uses a local area network (LAN) on which interprocessor communications are based on system softwares, such as operating systems and network protocols. The collective communication time is much longer than the interprocessor communication time. Table 1 shows a comparison of communication and synchronization performances between the machines. Processing time of 1-integer data is measured on the MP-2. For the interprocessor communication time on the Alpha cluster, the round trip time of 1-integer messages is measured. For the synchronization time, the master workstation sends a 1-integer multicasting message by using *pvm_mcast*() after collecting 2-integer messages from all five slave workstations. The measured time includes the time of *gettimeofday*().

Table 1: The Communication Performance and Array Access Times on the MP-2 and Alphas (in μseconds).

	Comm.	Syn.	Read	Write
MP-2	385	71	5.47	5.86
DEC Alpha	3900	11700	0.08	0.09

2.2 Computation Capability

Since the logic simulation requires many event manipulations, such as storing events, accessing events, comparing events, removing events, and checking the queue boundary, the simulation performance is significantly affected by the computation capability of processors. On the MasPar, most data movement within a processor occurs on the internal processor 4-bit nibble bus and the 1-bit bus. The distributed system is composed of DEC Alpha workstations whose clock rate is 133 MHz. Table 1 shows the processing performance of array read and write on the MP-2 and the DEC Alpha. The DEC Alpha shows much faster processing times than the MP-2 on array write and read operation.

2.3 Local Memory Space and Number of Processors

Most SIMD machines have small local memories. For example, the Connection Machine CM-2 has 8K memories, the MP-1 has 16K memories, and the MP-2 has 64K memories. The number of processors in the SIMD machines is quite large. The MasPar 1200 family has up to 16K PEs and the CM-2 has up to 64K PEs. In contrast, the distributed system has a larger memory, but a small number of processors. Most of workstations, such as the SUN Sparc 10 and the DEC

Alpha 3000, have a 32 MB memory. When the number of gates are larger than the number of processors, many gates should be assigned to a processor. In this case, however, the recent SIMD machines have the limitation of small local memory space. In contrast, the distributed systems can take the benefit of having a large memory for the simulation of huge circuits by accommodating many gates.

3 THE MODEL OF PDELS

The event-driven protocol used for PDELS is the most well-known optimistic protocol, Time Warp (Jefferson 1985). Time Warp processes events whenever there are unprocessed events. Since each process computes and propagates immature events, the protocol has to recover the earlier system state from the wrongly-advanced state when it receives genuine events. The *Rollback* situation occurs when an event called *straggler* arrives, which has a timestamp smaller than the current *Local Virtual Time* (LVT) of a process. To cope with the rollback situation, each process must keep all events that have already been sent. *Fossil collection* is needed to reuse the storage of queues; whenever queues are out of space, it removes from the queue the processed events that have lower timestamps than *Global Virtual Time* (GVT). As the mechanism to manipulate rollback situations, the immediate cancellation mechanism (Chung and Chung 1991) is used. In this mechanism, the process that gets a straggler starts the processing from the point of the straggler's timestamp, ignoring all messages sent so far in the event queue of input port.

In our simulation, the logical processes in a circuit are randomly partitioned and evenly allocated to the processors. Each logical process in the circuits contains 3 input ports and 2 output ports by preprocessing step of circuit reconfiguration. The output ports of a logical process are pre-determined to input ports of the destination processes according to the characteristic of the circuit. An input port has its own input queue for storing the arrived events. There is no queue for output ports.

To compute the GVT, processors are synchronized periodically after determining their PVT (Processor Virtual Time), determined from the minimum LVT's among assigned processes. Each processor repeats the same procedure that consists of the following steps: selecting some processes among activated processes, executing the selected processes, sending the generated messages to other processes, and a synchronization. This procedure is called a *simulation cycle*. Each processor selects some of the activated processes up to the grain size and executes the events of the selected processes independently.

4 ANALYSIS

To study the effect of granularity, in this section formulas are developed to predict performance metrics, such as the total execution time and the number of simulation cycles, when the grain size is k and the LP ratio is r. The *total execution time* of a logic circuit is defined as the total spent time in simulating the circuit. It can be computed as the average execution time per simulation cycle multiplied by the number of simulation cycles. Before proceeding further to the analysis of granularity, we present some necessary notation.

- N is the number of logical processes.
- P is the number of processors in the system.
- r is the LP ratio that is $\lceil \frac{N}{P} \rceil$.
- k is the grain size of the simulation, that is, the maximum number of processes that are allowed to be executed in a processor.
- n_i is the random variable that represents the number of activated processes in the ith processor.
- m_i is the random variable that represents the number of processes that are actually executed in the ith processor. It is the minimum value of n_i and k, i.e., $n_i \wedge k$.
- S_r^k is the total number of simulation cycles when the grain size is k and the LP ratio is r.
- τ^k is the execution time per simulation cycle when the grain size is k and the LP ratio is r.
- T^k is the total execution time of the simulation with k and r. That is, $T^k = S_r^k * \tau^k$.

n_i, m_i, τ^k, and T^k are all variables that are dependent on the LP ratio, but we explicitly state r only for S_r^k.

4.1 Execution Time per Simulation Cycle

The execution time in a simulation cycle is determined by the processor that executes the largest number of processes in the cycle. Let m_{max} be the largest number of executed processes in a simulation cycle over all processors, i.e., $m_{max} = Max(m_1, m_2, ..., m_P)$. Suppose the fanout of a process is w, i.e., each executed process generates w events that are to be sent to other processes. Then, the execution time per simulation cycle when the grain size is k, which is τ^k, is derived as follows.

$$\begin{aligned}\tau^k \;&=\; t_{proc} \;+\; m_{max}\;[\,t_{comp} + w\;(1-q)\;t_{move}\,]\\ &\quad +\;(P-1)\;t_{comm}\;+\;t_{syn} \qquad (1)\end{aligned}$$

where t_{proc} is the general processing time in a processor to select processes to be executed, t_{comp} is the computation time of a process, q is the probability that a generated event is transmitted to another processor, t_{comm} is the interprocessor communication time, t_{move} is the message moving time within a processor, and t_{syn} is the global synchronization time. Since N processes are randomly and evenly assigned to P processors, the probability q is $\frac{N-r}{N-1}$. The term, $(P-1)\ t_{comm}$, is the maximum interprocessor communication time for the all generated events in the simulation cycle. In our simulation model, the events generated within a processor are combined and sent together as a message if their destination processes are on the same processor. If the logical processes are randomly distributed and the LP ratio is large, then a processor may send to all other processors. The machine-dependent parameters, t_{proc}, t_{comp}, t_{comm}, t_{move}, and t_{syn} can be measured in experiments.

As a machine-independent parameter, we derive the expected value of m_{max} when the grain size is k. Let $f_{n_i}(x)$ and $F_{n_i}(x)$ be the probability mass function and the cumulative distribution function of n_i, respectively, where $0 \leq x \leq r$.

Theorem 1 *The probability mass function of m_i is given by*

$$f_{m_i}(x) = \begin{cases} f_{n_i}(x) & \text{if } x < k \\ 1 - F_{n_i}(x) & \text{if } x = k \end{cases} \tag{2}$$

Proof: Because m_i is the number of actually executed processes among the n_i active processes and the grain size k limits the maximum of m_i, $m_i = n_i \wedge k$. If $m_i < k$, the number of actually executed processes are the same as the number of active processes, that is, $m_i = n_i$. The condition that $m_i = k$ includes all the cases that the number of active processes is larger than k, i.e., $n_i \geq k$. Thus, $f_{m_i}(x) = Pr[n_i \geq x]$ if $x = k$. Note that m_i has the *truncated distribution* at k. □

The cumulative distribution function (c.d.f.) of m_i is given by

$$\begin{aligned} F_{m_i}(x) &= 1 - Pr[m_i > x] = 1 - Pr[n_i \wedge k\ > x] \\ &= \begin{cases} 1 - Pr[n_i > x] = F_{n_i}(x) & \text{if } x < k \\ 1 - 0 = 1 & \text{if } x \geq k \end{cases} \end{aligned}$$

If $m_1, m_2, \ldots, m_P$ have the identical distribution, say $F_m(x)$, and they are independent, then the c.d.f. of m_{max} is computed as

$$\begin{aligned} F_{m_{max}}(x) &= Pr[m_{max} \leq x] \\ &= Pr[\bigcap_{i=1}^{P}\{m_i \leq x\}] = F_m(x)^P \end{aligned} \tag{3}$$

Since m_{max} is a non-negative integer value, $E(m_{max}) = \sum_{x=0}^{\infty}(1 - F_{m_{max}}(x))$ (see Resnick 1992). Based on the c.d.f. of m_{max}, the expectation of m_{max}, $E(m_{max})$, is obtained as follows.

$$E(m_{max}) = k - \sum_{x=0}^{k-1}(F_m(x)^P) \tag{4}$$

Based on Equations (1) and (4), the expected execution time per simulation cycle is as follows:

$$\begin{aligned} E(\tau_{max}^k) &= t_{proc} + E(m_{max})\ [t_{comp} + w\ (1-q)\ t_{move}] \\ &\quad + (P-1)\ t_{comm}\ +\ t_{syn} \end{aligned} \tag{5}$$

4.2 Number of Simulation Cycles

Another component of total execution time is the number of simulation cycles, S_r^k. In this subsection, S_r^k is derived when the grain size is k and the LP ratio is r. As a preliminary step, we define the *workload* of parallel simulation.

Definition 1 *Let W_r^k be the workload of the simulation when the grain size is k and the LP ratio is r. Then, W_r^k is defined as the total number of executed events by all processors during the simulation.*

By definition, W_r^k is computed by multiplying the total number of simulation cycles by the total number of executed processes in a simulation cycle, that is,

$$W_r^k \ = \ S_r^k\ (\sum_{i=1}^{P} m_i) = S_r^k\ P\ \bar{m} \tag{6}$$

where $\bar{m}$ is the average of $m_1, m_2, \ldots, m_P$. Let $E(\bar{m})$ be the expectation of $\bar{m}$. When P is large, $\frac{\bar{m}}{E(\bar{m})}$ is near 1. Hence, we can use $E(\bar{m})$ for $\bar{m}$. Since $m_1, m_2, \ldots, m_P$ have the same distribution, we can use $E(m)$ for $E(\bar{m})$. Applying Equation (4) with $P = 1$, we have

$$E(m) = k - \sum_{x=0}^{k-1} F_m(x) \tag{7}$$

Note that the total number of executed events during a simulation is not dependent on the values of k and r. This property is called *the workload conservation law.*

Definition 2 *Workload Conservation Law: The workload is preserved for all values of r and k, i.e.*

$$W_1^1 \ = \ W_r^k \ \ \textit{for any } k \textit{ and } r,\ 0 \leq k \leq r$$

By using Equation (7) and the workload conservation law, the number of simulation cycles can be derived as follows.

Theorem 2 *When grain size is k and the LP ratio is r, the number of simulation cycles, S_r^k, is computed as follows:*

$$S_r^k \approx \frac{S_1^1 \; a \; r}{E(m)|_{(r,k)}} \qquad (8)$$

where a is the average activation probability of a logical process.

Proof: Assume $r = 1$, $k = 1$, $P = N$, and that all active processes are executed. Therefore,

$$W_1^1 \;=\; S_1^1 \; N \; a \qquad (9)$$

If $r > 1$ and $k \geq 1$, then $P = \lceil \frac{N}{r} \rceil$ and

$$\begin{aligned} W_r^k &= S_r^k \; P \; \bar{m}|_{(r,k)} \\ &\approx S_r^k \; \lceil \frac{N}{r} \rceil \; E(m)|_{(r,k)} \end{aligned} \qquad (10)$$

Due to the workload conservation law, $W_1^1 = W_r^k$ implies $S_1^1 N a \approx S_r^k \; \lceil \frac{N}{r} \rceil \; E(m)|_{(r,k)}$. Thus,

$$S_r^k \;\approx\; \frac{S_1^1 \; N \; a}{\lceil \frac{N}{r} \rceil \; E(m)|_{(r,k)}} \;=\; \frac{S_1^1 \; a \; r}{E(m)|_{(r,k)}}$$

□

Therefore, the total execution time is as follows:

$$T^k \;=\; S_r^k \; E(\tau_{max}^k) \qquad (11)$$

5 RESULTS

This section presents the experimental results of circuit simulation that are performed on the MP-2 and a cluster of six DEC Alpha workstations interconnected by a DEC GIGAswitch through FDDI. The GIGAswitch supports a peak data transfer rate of 200Mbits per second. As benchmark circuits, we use several circuits from ISCAS89 benchmark suite (Brglez et al. 1989). We use PVM 3.3 for parallel processing on the distributed system. Time Warp is implemented on the distributed system as a master-slave model. For the MP-2, MPL (Massively Parallel Language on the MasPar) (MasPar 1990) is used.

5.1 Comparison of the SIMD machine and the Distributed System

As shown in Section 2, event queue manipulation is the major part of computations. We describe the implementation and data structure for queue manipulation in order to interpret the performance results of the target machines.

For the MP-2, a circular array data structure called *Circular Binary Search Queue* (CBSQ) is used. Since a PE of the MP-2 has a small local memory, input queues are to be implemented by the data structure that can reduce memory usage. The CBSQ not only saves the memory space due to the circular structure, but also makes queue handling efficient by using binary searches on rollback situations and fossil collections. The access time to CBSQ is the same as the read and write time shown in Table 1.

On the distributed system that has a few processors, the LP ratio must be larger than 1. The large memory of a workstation should be shared by many processes efficiently. For this purpose, the input queues are implemented by the linked data structure in our simulation on the DEC Alpha cluster. The enqueuing time is measured 8.19 μseconds when *malloc*() time is included. The dequeuing time takes 3.18 μseconds. To improve the queue manipulation speed, a *free space pool* is used. Instead of freeing the reserved space when dequeuing, the free space pool temporarily holds the released space for the future usage without calling malloc routines. When the free space pool is used, the enqueue operation time is reduced to 4.81 μseconds. Although the computation time of DEC Alpha workstations is much faster than that of the PEs in the MP-2 as shown in Table 1, the queue manipulation times on the two machines do not make a big difference because of the different implementation and manipulation of input queues.

Table 2: The Experimental Results on the MP-2 and a Cluster of DEC Alpha Workstations (in seconds).

Circuit	No. of Gates	Total Execution Time (T)		Performance Ratio
		MP-2	Alphas	$\frac{T_{Alpha}}{T_{MP-2}}$
S953	715	11.57	47.09	4.07
S1196	967	16.41	70.16	4.28
S1238	988	16.62	71.44	4.30
S1423	1161	11.98	52.79	4.41
S1488	1456	9.43	33.12	3.51
S1494	1462	9.48	33.21	3.50

Table 2 shows the experimental results of Time Warp on the MP-2 and the cluster of DEC Alpha workstations. Since the MP-2 has 4K PEs, the benchmark circuits whose numbers of gates are less than 4K are selected from the ISCAS89 suite. The number of input event vectors for the simulation is 1000. For the distributed system, the maximum grain size is used. The MP-2 produces about four times faster performance than the cluster of DEC Alpha workstations on the experiments. Since queue manipulation times

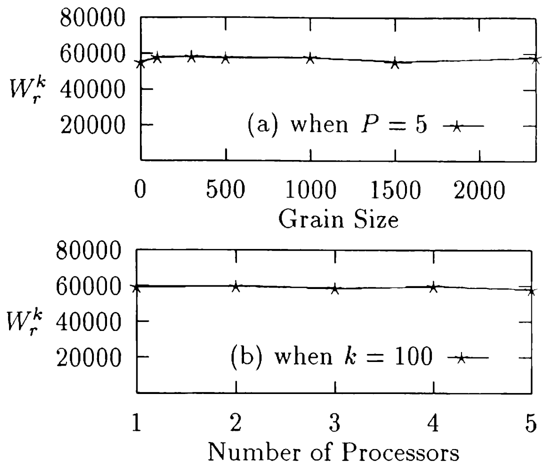

Figure 1: (a) The Workload when Varying Grain Size with $P = 5$. (b) The Workload when Varying the Number of Processors with $k = 100$.

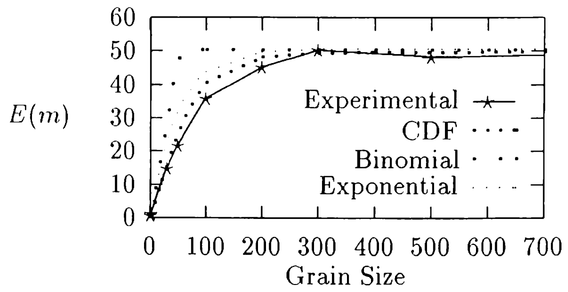

Figure 2: The $E(m)$ vs. Grain Size k that Varies from 0 to the LP Ratio.

on those machines are similar as explained above, the major difference of the results is caused by the difference in communication and synchronization time.

5.2 The Workload Conservation Law

In Section 4, we used the workload conservation law to compute the number of simulation cycles. The workload conservation law suggests that the total number of executed events by all processors during the simulation are preserved for all values of r and k. To verify the correctness of the law, the total number of executed events during the simulation are counted. The S13207 circuit with 100 input event vectors is used. Figure 1 (a) shows the workload at $P = 5$, varying k values, and Figure 1 (b) presents the workload with a fixed k, 100, varying P values up to 5. Both figures show the almost same values of the total number of executed events for all r and k and support the workload conservation law.

5.3 Comparison of Experimental and Analytical Results

This subsection verifies the analysis in Section 4 by comparing experimental results on a cluster of DEC Alpha workstations with analytical results. The total execution time and its related components are compared for the S13207 circuit with 100 input event vectors. Among the six workstations, one machine works as the master and the other five workstations work as slaves; P is 5. The S13207 circuit has 11689 gates; N is 11689. Since 11689 gates are assigned to five workstations evenly, the LP ratio, r, is 2338. The grain size can vary from 1 to r.

Figure 2 shows the experimental and analytical results of $E(m)$. Because $E(m_{max})$ behaves similarly to $E(m)$, but has different scale, we omit the figure of $E(m_{max})$. As k increases, $E(m)$ increases with k until it reaches a bound. Once it reaches the bound, the value of $E(m)$ is steady even as k increases. The bound of curves is delimited by the number of executed processes; it is not limited by k but the number of activated processes. The CDF in the figure uses the $F_n(x)$ from experiments by measuring the number of executed gates at each simulation cycle on processors and computing the p.d.f. and c.d.f. of n. Figure 2 also presents the curves of $E(m)$ that are computed by the binomial distribution and the exponential distribution of n. The result of exponential distribution is closer to the experimental one than that of the binomial distribution, but both are still far from the experimental result.

Figure 3 (a) shows the experimental results of S_r^k compared with the analytical results as a function of k. The dotted lines represent the analytical values, and the solid lines represent the experimental values. To obtain the values of parameters needed for Theorem 2, we simulated the S13207 circuit by a sequential simulator of Time Warp with the LP ratio 1. The value, 222, is obtained for S_1^1, 0.021457 for the activation probability, a, and 1.286383 for the average fanout of a gate, w. Since S_1^1 is obtained when the LP ratio is 1, the simulation parallelism is maximum and S_1^1 should be the low bound of S_r^k as shown in the figure. The results say that as k becomes small, due to the limitation of grain size, processors postpone the execution of some activated processes to the next cycle, increasing the number of simulation cycles.

The experimental and analytical results of T^k are shown in Figure 3 (b) as a function of k. When k is near 1, T^k becomes very high because the time spent for frequent communications dominates the total execution time. This implies that very small grain

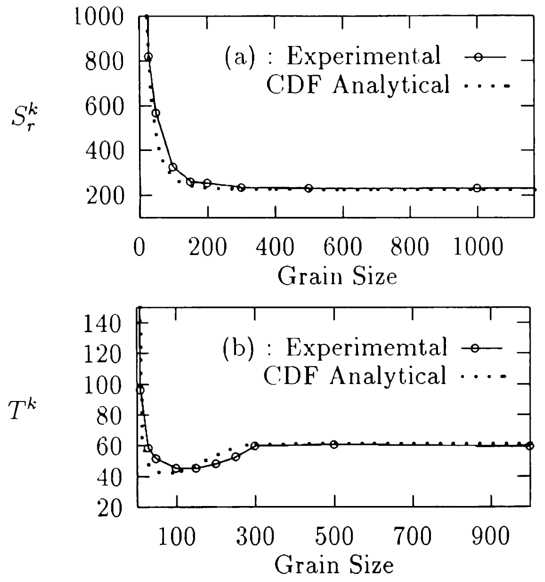

Figure 3: Comparisons of (a) S_r^k and (b) T^k.

sizes can deteriorate the simulation performance significantly when the communication time is relatively large compared to the computation time. In contrast, in the SIMD machines, which have short communication and synchronization times, the simulation performance may not be affected much with even small grain sizes when the LP ratio is larger than 1. In Figure 3 (b), when k is around 100, the simulation shows the optimal performance. It is shown that good performances can be achieved on distributed systems by using at least a certain amount of grain size. When k becomes around 300, the simulation reaches to the steady performance that is much higher than the optimal performance. This result implies that when k is larger than the optimal size, the variation of the execution times per simulation cycle among processors is large. The steady curve after $k = 300$ is delimited by the number of activated processes in a simulation cycle. The analytical curve that is compared to the curve of experimental results is shifted to the left somewhat. It is caused by the differences of curves on S_r^k as shown in Figure 3 (a). By adjusting the grain size, thus, the unbalanced load among the processors can be adjusted by reducing idle times of processors, and the simulation performance can be improved.

5.4 Effects of Granularity on Distributed Systems

In the previous subsection, we have studied how performances could be affected by adjusting granularity for a given circuit when the LP ratio is fixed.

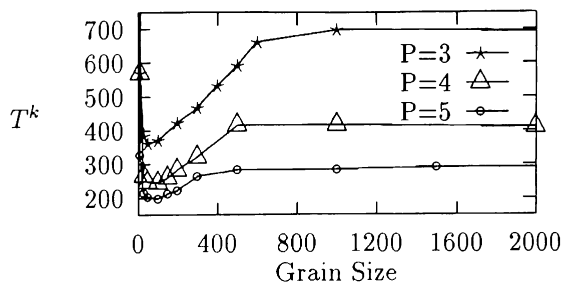

Figure 4: The Experimental Total Execution Times with S38417 Circuit.

Even on the same distributed system, however, the effects of granularity may vary as different machine-independent factors are given. Varying the number of processors, we have performed similar experiments to examine the performance effects of granularity for several circuits from ISCAS89 benchmark suite.

Figures 4 shows T^k for S38417 circuit, when the number of slave workstations varies from 3 to 5. From experimental results, we observe that the amount of improvement in T^k due to adjusting the grain size differs between circuits. Table 3 gives more detailed information in this aspect for four different circuits. Total execution times are compared for each circuit when k is the LP ratio and the optimal size that yields the minimum total execution time. The *Improved Rate (IRate)* is computed as $\frac{(T^k \ when \ k=r)}{(T^k \ when \ k=optimal)}$. Depending on circuits, the experimental results show various IRates up to 2.05. Even on the same circuit, the IRate varies as P increases. As P increases, less gates are assigned to a processor and more parallelism can be achieved. Therefore, T^k decreases and the speedup increases. The speedups in T^k are given in Figure 5 for the S13207 and S38417 circuits. The figure shows the speedups when the grain sizes are equal to the LP ratio and the grain sizes are optimal. As the number of processors increases, almost linear speedup is achieved up to five processors.

6 CONCLUSION

In this paper, we have studied the effects of granularity on the performance of optimistic protocol on distributed systems for benchmark circuits. We summarize the contributions made by this research. First, the architectural characteristics of a cluster of DEC Alpha workstations has been compared to the MP-2's in view of PDELS. Next, analytic formulas have been developed to predict several performance metrics, such as the number of simulation cycles and the

Table 3: The Experimental Results on a Cluster of DEC Alpha Workstations (in seconds).

P	*k*	S13207	S35932	S38417	S38584
2	LP ratio	286.28	8493.81	1501.87	5175.72
	Optimal	239.30	7153.13	731.59	4432.14
	IRate	1.20	1.19	2.05	1.17
3	LP ratio	147.81	3992.81	698.05	2829.21
	Optimal	131.96	3539.53	361.57	2469.10
	IRate	1.12	1.13	1.93	1.15
4	LP ratio	98.15	2281.65	413.34	1921.02
	Optimal	71.15	1965.55	244.72	1672.98
	IRate	1.38	1.16	1.69	1.15
5	LP ratio	61.77	1459.43	284.52	1244.56
	Optimal	45.21	1326.58	198.35	995.01
	IRate	1.37	1.10	1.43	1.25

total execution time. Finally, the effects of grain size on distributed systems have been observed. According to the results, by adjusting the grain size, the improved rate of total execution time has increased up to 2.05.

ACKNOWLEDGMENTS

We gratefully acknowledge the cooperation of the Scalable Computing Laboratory of the Department of Energy's Ames Laboratory and the ACS Laboratory of MSU in allowing the use of their MP-2 and DEC Alpha machines, respectively. We also thank Dr. James Hannan in Statistics and Probability Department for his valuable comments.

REFERENCES

The Oak Ridge National Laboratory. 1994. *PVM 3 User's Guide and Reference Manual.*

Agrawal, V. D., and S. T. Chakradhar. 1992. Performance Analysis of Synchronized Iterative Algorithms on Multiprocessor Systems. *IEEE Transactions on Parallel and Distributed Systems* 3:739–746.

Brglez, F., D. Bryan, and K. Kozminski. 1989. Combinational Profiles of Sequential Benchmark Circuits. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, 1929–1934.

Chung, M. J., and Y. Chung. 1992. Performance Estimation based on Gate-to-Processor Ratio in Parallel Logic Simulation. In *Proceedings of the 1992 International Conference on Parallel Processing*, 246–253.

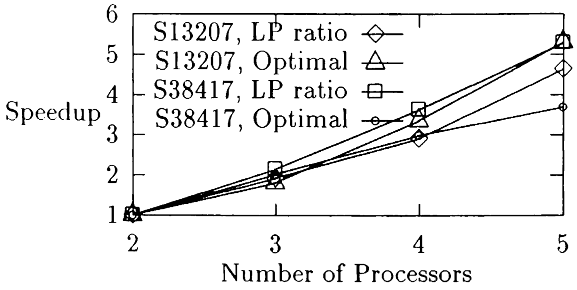

Figure 5: The Speedup of Total Execution Time.

Chung, Y., and M. J. Chung. 1991. Time Warp for Efficient Parallel Logic Simulation on a Massively Parallel SIMD Machine. In *Proceedings of the Tenth Annual IEEE International Phoenix Conference on Computers and Communications*, 183–189.

Jefferson, D. 1985. Virtual Time. *ACM Transactions on Programming Languages and Systems* 7:404–425.

MasPar Computer Corporation. 1990. *MasPar MP1.*

Resnick, S. I. 1992. *Adventures in Stochastic Processes.* Birkhauser, Boston.

AUTHOR BIOGRAPHIES

EUNMI CHOI is a PhD candidate in computer science at Michigan State University. Her current research interests include parallel asynchronous protocols, parallel logic simulation on parallel and distributed systems, and parallel and distributed algorithms. She received an MS in computer science from MSU in 1991, and a BS from Korea University in 1988. She is a member of ACM and the IEEE Computer Society.

MOON JUNG CHUNG is an Associate Professor in the Department of Computer Science at Michigan State University. His research interests include algorithms in design automation, design process management, and parallel algorithms. Prior to joining MSU, he was a faculty member at Rensselaer Polytechnic Institute. He has received a senior associateship of the National Research Council for 1994-1995. During this tenure, he is visiting the US Army Research Laboratory. Dr. Chung received a BS from Seoul National University, Korea; an MS from Korea Advanced Institute of Science and Technology; and a PhD from Northwestern University. He is a member of ACM and the IEEE Computer Society.