Multiple Replications in Parallel: Distributed Generation of Data for Speeding up Quantitative Stochastic Simulation

Gregory Ewing, Donald McNickle♦ and Krzysztof Pawlikowski
University of Canterbury
Christchurch, New Zealand

Abstract. Quantitative stochastic simulation has become a tool commonly used by engineers in various disciplines for studying and evaluating the performance of complex dynamic systems. Unfortunately, a simulation study can require a very long run length to obtain results with an acceptable accuracy. In this paper we consider the properties of a new scenario for stochastic simulation, named Multiple Replications in Parallel Time Streams (MRIP), that allows simulations to run in an automatic way, using distributed computing platforms such as a local area network of workstations. In this paper we look at the speedup of stochastic simulations run under MRIP. Both theoretical and experimental results of speedup analysis are presented.

1. Introduction

Over the last decade discrete-event simulation has become perhaps the tool most commonly used tool by engineers of various disciplines for studying and evaluating performance of various systems. This is the result of significant achievements in electronic and computer engineering that have led to the broad proliferation of powerful computers and computer networks, and significant achievements in software technology, that have resulted in very simple and efficient interactive human-computer interfaces. Today, it is natural for telecommunication engineers to study processes occurring in data communication networks by watching their animated, dynamic graphical representations on monitors, using data generated by computers during simulation runs. There is easy access to various user-friendly simulation packages in which traditional discrete-event simulation modeling is supported by various concepts of artificial intelligence. Some of these packages totally release users from burdens of programming, allowing then to construct simulation models from standard components appearing on screens as icons. This situation has created a climate for spreading a popular impression that simulation is mainly an exercise in computer programming.

Unfortunately, this is totally misleading impression, at least in the case of quantitative stochastic simulation used for quantitative assessment of the performance of dynamic systems. Any such simulation is equivalent to a statistical experiment, and may require quite sophisticated statistical techniques for producing statistically accurate final results. J.Kleijnen [KLEI79] warned that: "… *computer runs yield a mass of data but this mass may turn into a mess <if the random nature of such output data is ignored, and then>... instead of an expensive simulation model, a toss of the coin had better be used*". Unfortunately, this and similar warnings have remained unnoticed. B.Gaither, the Editor-in-Chief of the ACM Performance Evaluation Review, complaining about the way stochastic simulation is used in computer science and engineering, stated in his editorial that he did not know of "... *any other field of engineering or science where similar liberties are taken with empirical data …*" [GAIT90]. One way of escaping from this situation is through development of a methodology for automated sequential stochastic simulation; see for example [HEID83, PAWL90]. There are many problems which need to be solved before such methodology could be commonly used. One of them is the problem of credibility of the final simulation results; see for example [EWIN95, MCNI96].

Another inherent problem of stochastic simulation is that simulation of even moderately complex models is often computationally intensive and require very long simulation runs. Excessive runtimes hinder development and validation of simulation models, and can even totally prohibit some performance evaluation studies. The obvious solution is to speed up simulation by

♦ The correspondence author: Dr D.McNIckle, Department of Management, University of Canterbury, Christchurch, New Zealand

executing it on a multiprocessor, a distributed computer system, possibly using computers linked by a local area network. Traditionally, distributed or parallel[1] stochastic simulation has meant **Single Replication in Parallel** (**SRIP**), based on many processors cooperating to execute a single replication of a simulated system; see eg. [BILE85, FUJI90]. This scenario may give reasonable speedup, provided that the given simulation model is highly decomposable. In this paper we consider an alternative scenario for distributed stochastic simulation, **Multiple Replications in Parallel** (**MRIP**)**,** with multiple processors independently running their replications of the simulated system but cooperating with central analysers (one central analyser for each performance measure analysed) that are responsible for statistical analysis of output data [PAWL94].

We are concerned here exclusively with sequential stochastic simulation, which is the only way of producing final results with the required statistical accuracy since *"… no procedure in which the run length is fixed before the simulation begins can be relied upon to produce a c.i. that covers the true steady-state mean with the desired probability level"* [LAW83]. This means that each central analyser analyses the precision of its estimate (the half width of the confidence interval at a given confidence level) at consecutive checkpoints of the simulation, and the simulation is stopped when all estimates reach the required level of precision.

Without loss of generality of our discussion, we will focus on discrete-event steady-state simulation, ie. stochastic simulation applied for performance evaluation of dynamic stochastic systems modelled eg. by queueing networks, in regions of their (internal) stability. Additionally we will restrict our discussion to steady-state simulation, the most challenging application of stochastic simulation, conducted for studying behaviour of systems in their steady state, that theoretically can be reached after infinitely long period of time [PAWL90].

In Section 2 we discuss the two above-mentioned scenarios of concurrent quantitative stochastic simulation in more details. The theoretical limitations of the speedup of the sequential stochastic simulation achievable under MRIP, as well as some experimental results of speedup analysis from the steady-state simulation of simple queueing systems, are discussed in Section 3.

2. Two Scenarios for Concurrent Quantitative Stochastic Simulation

As mentioned, stochastic simulation of even moderately complex simulation models can require collecting large numbers of observations, to ensure that the final results are sufficiently accurate statistically. The obvious solution is to speed up simulation by executing it in a distributed computer system, possibly using computers linked by a local area network. Traditionally, distributed or parallel stochastic simulation has meant a simulation run according to the SRIP scenario, with a number of processors executing different parts of a single replication of a simulated system.

In this scenario, a simulation process and/or simulation model is partitioned between a number of processors. When a simulation process is distributed, this distribution is done at a functional level, and the logical topology of interprocessor connections may reflect different functional elements of the simulation (event set processing, input/output processing etc.); see eg. [BILE85]. It is obvious that this method cannot offer a substantial speedup in itself, since the degree of such distributiveness is limited. The second option for SRIP is to partition a given simulation model into a set of submodels to be simulated at different processors of a tightly or loosely coupled multiprocessors system. The processors responsible for running processes related to different simulation submodels occasionally have to synchronise the advance of simulated processes. Many different methods have been proposed to achieve such a synchronisation; a survey of the problems and solutions of this problem can be found eg. in [FUJI90]. Generally speaking, it is achieved by exchanging timestamped messages between participating processors. Reasonable speedup is possible, provided that a given simulation model is highly decomposable. Unfortunately, this characteristic is not frequently found in practice, thus the efficiency of this approach is strongly application-dependent [WAGN89]. The research into distributed/parallel processing, having successfully solved many related problems, has not led yet to a portable and efficient tool for distributed stochastic simulation.

In contrast, the MRIP scenario can be applied to any simulation model, regardless of the level of its internal decomposability. One can even parallelise any simulation model automatically, using

[1] For the purpose of this paper, the terms "concurrent", "parallel" and "distributed" are used as synonyms, meaning a simulation executed by a number of cooperating processors during the same time interval.

an ordinary sequential model as the basis; as done in Eclipse [SUND91] and AKAROA [YAUP93]. Under MRIP, replicated simulations of the same simulated system are run on multiple processors, or simulation engines, which work independently to produce samples of output data (one stream of samples for each performance measure). Observations generated by different simulation engines, but representing values of the same performance measure, are submitted to the global analyser responsible for analysing a given performance measure (both its point and interval estimates).

3. Speedup achievable under MRIP

To analyse speedup of sequential steady-state simulations run in the MRIP scenario, let us note that under MRIP each simulation engine runs an independent replication of the simulation process. Thus, first it generates data characterising the initial transient period (if any such period occurs) and these data are discarded. Only later, having entered steady state regime, a simulation engine starts its contribution in steady state analysis by submitting its data to a global analyser(s). Obviously, the best speedup can be achieved if one launches simulation engines on a homogeneous set of processors. With heterogeneous processors, in the worst case there may be no speedup. This case occurs when one of processors is fast enough to generate the required total number of observations before any of the slower processors reaches its first checkpoint.

In the following discussion we assume that steady-state simulation is run in the MRIP scenario on a set of P homogenous processors (P homogenous simulation engines), and the length of simulation is measured by the (average) total number of observations submitted by one simulation engine to the global analyser before the simulation is stopped.

Assuming very fine granularity of stochastic simulation (a small distance between checkpoints) the speedup of steady-state simulation in the MRIP scenario would be governed by the well-known Amdahl's law. Namely, if f represents a fraction of the simulation which cannot be parallelised (in the steady-state simulation, this corresponds to the relative length of the initial transient period) then the speedup S is given by $S=1/(f+(1-f)/P)$.

However, the simulation is stopped when P simulation engines have delivered to the global analyser the total number of observations needed for satisfying the stopping criterion (the half-width of confidence interval smaller than the required one). As the number of simulation engines increases, we will reach the situation in which all P simulation engines are only able to reach their first checkpoint before the global analyser stops the simulation because it has access to the N_{min} observations needed.

Let n_1 be the location of the first checkpoint (the number of observations to be generated by a simulation engine when it reaches the first checkpoint), and let $P_{min} = \min\{P: n_1 P \geq N_{min}\}$. Adding more then P_{min} processors would not increase the speedup since it has already reached its limit value $S_{max}=P_{min}$. The only effect of having more data (generated by $P>P_{min}$ processors) would be better-than-required precision of the final results. This gives the upper limit of the speedup under MRIP as $S_{max}=\lceil N_{min}/n_1 \rceil$.

Linking the two effects together leads to the following **truncated Amdahl's law** of speedup for MRIP scenario of steady-state simulation

$$S= \min (1/(f+(1-f)/P), \lceil N_{min}/n_1 \rceil) \qquad (1)$$

This formula is graphically depicted in Fig.1, assuming that $P_{min}=7$. Note that e.g. when simulating the M/M/1/∞ queuing system loaded at 90%, and estimating its mean queue length with

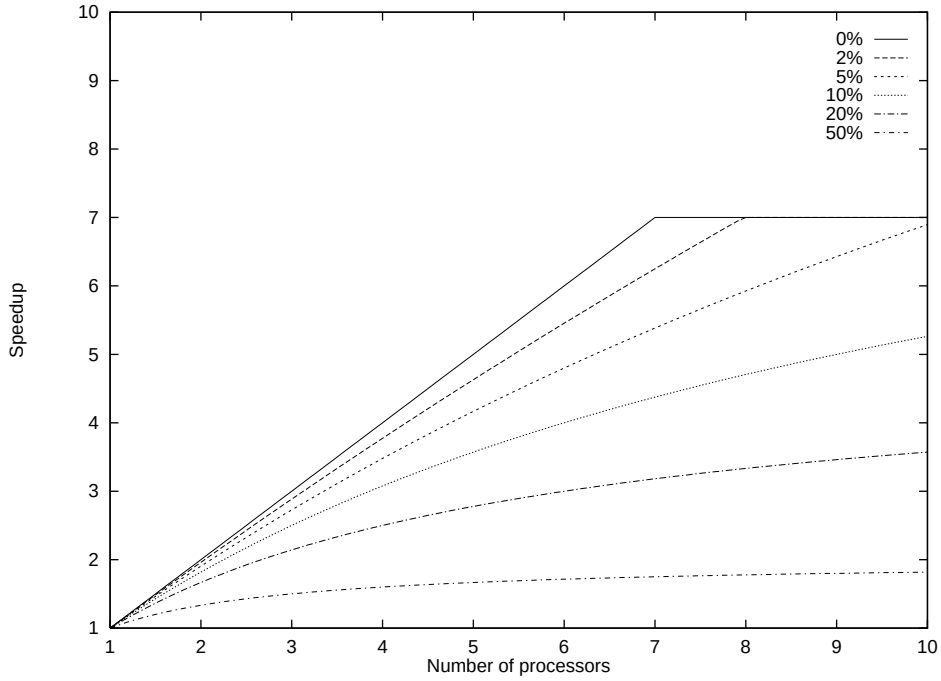


Figure 1. Speedup of MRIP achievable according to the truncated Amdahl's law

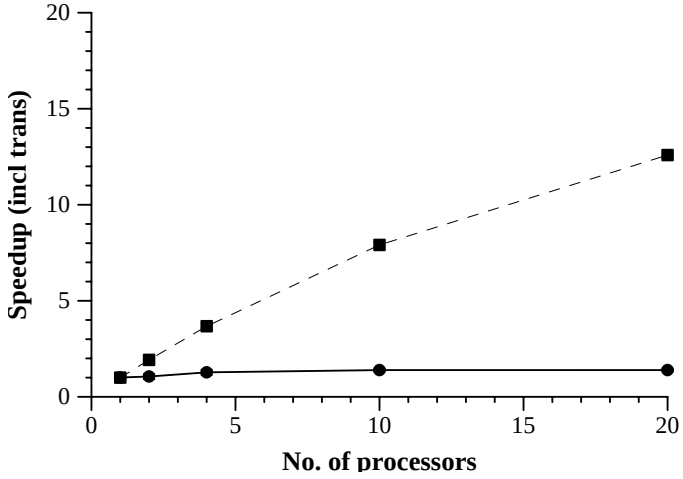
5% relative width of confidence interval at 0.95 confidence level, then mean values of N_{min} and n_1 are about 1000000 and 500, respectively. Thus in this case P_{min}=2000. On the basis of Eq.(1) one can draw the following conclusions

(1) The longer the relative length of the initial transient period, the smaller the speedup. Thus, steady-state simulation of highly dynamic stochastic systems, with long-lasting initial transients, can not be sped up as much as those with short transients.
(2) If the length of initial transient is negligible in comparison with the total simulation length, or the length of initial transient plays no role in steady-state analysis (as in regenerative simulation [PAWL90]), then the speedup can be linear with the number of processors engaged.
(3) Since f=0 in other-than -steady-state types of stochastic simulation, the achievable speedup for these types of simulation will be linear with the number of processors engaged. This includes e.g. terminating simulation or simulation of non-stationary processes.

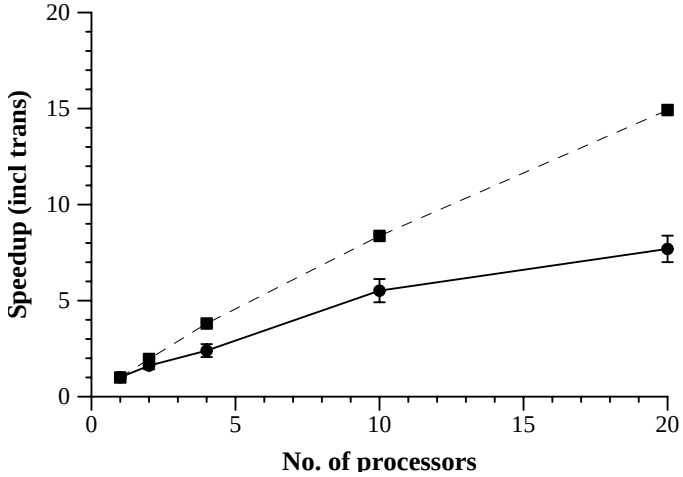
For more accurate analysis of speedup of steady-state simulation in the MRIP scenario, one needs to take into account specific computational requirements of the method used for analysis of steady-state estimates and their precision. For example, when estimating a steady-state mean value of, for example, delay in a telecommunication network, one can select one of about 10 different methods for estimating the variance of the mean. A survey of such techniques until 1990 can be found in [PAWL90].

Figs. 2 and 3 show the results of our experimental speedup analysis for the method of non-overlapping Batch Means (BM), and the method of Spectral Analysis (SA) in the version proposed by Heidelberger and Welch [HEID81]. All results were obtained by using AKAROA[2] II [PAWL94, EWIN96] (a simulation p**a**c**k**age for **a**utomatic generation and cont**r**ol of processes for parallel st**o**chastic simul**a**tion), version 1.2.1, developed at the Department of Computer Science, University of Canterbury, for running MRIP on local area networks of Unix workstations.

[2] Also: a nice spot on Banks Peninsula in the South Island of New Zealand.



(a)



(b)

Fig.2. Speedup for the method of (non-overlapping) batch means. M/M/1/∞ queuing system loaded at (a) 50%, (b) 90%; estimation of mean queue length; stopping with the relative width of confidence intervals ≤ 0.05, at 0.95 confidence level.

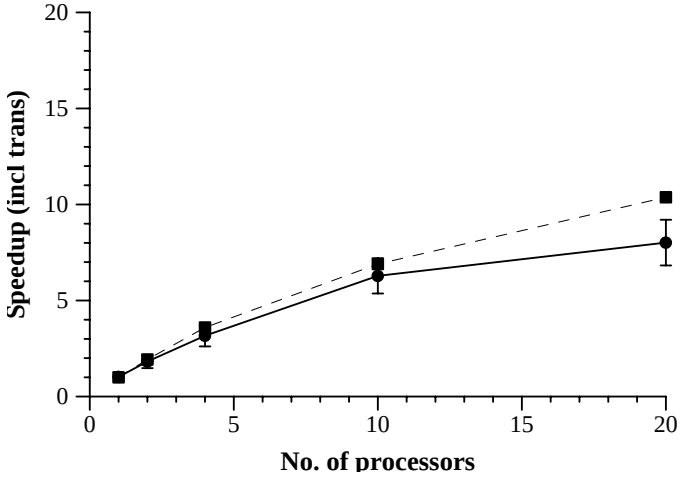


Fig.3. Speedup for the method of spectral analysis. M/M/1/∞ queuing system loaded at 50%; estimation of mean queue length; stopping with the relative width of confidence intervals ≤ 0.05, at 0.95 confidence level.

All results are based on simulations of M/M/1/∞ queuing system, and estimating mean queue length with 5% relative width of confidence interval at 0.95 confidence level. Broken lines in all these figures correspond to the speedup predicted by the truncated Amdahl's Law.

It is clear that, although neither of these methods provide as much speedup in practice as is predicted by Eq. (1), SA comes considerably closer to doing so than BM. In particular, hardly any speedup at all is obtained under BM when the simulated system is moderately loaded. This is surprising, because the nature of BM would lead one to expect a speedup that follows Amdahl's law very closely. If B independent batch means are required before stopping the simulation, then it should not matter which processors they come from, and with P processors only B/P batches should be required from each processor. The abysmal speedup depicted in Fig. 1 is therefore quite unexpected.

The reason for this result becomes apparent when we take into account the number of observations required to determine the batch size. In BM, after the transient phase is over, each processor goes through a *batch size determination (BSD) phase* in which observations are collected until a batch size can be found securing sufficiently uncorrelated batch means [PAWL90]. Only when this phase is over does the processor enter the *estimation phase* and begin sending estimates to the global analyser.

The BSD phase therefore constitutes an additional phase of the simulation which, together with the transient phase, cannot be parallelised. The importance of this phase can be seen from Table 1, which presents an average breakdown of the observations collected by the simulations of Fig. 2 into the three phases. At a load of 50%, the BSD phase represents a very large fraction of the total number of observations required, severely limiting the speedup obtainable from MRIP. As the load increases, the simulation run lengths become longer, but so does the BSD phase. Even at a load of 90%, the BSD phase is still very significant, limiting the speedup to about half of that which ought to be obtainable according to Eq. (1).

Table 1. Breakdown of observations collected from one simulation engine under BM into three processing phases.

	Number of Observations				
Load	Transient	BSD	Estimation	Total	BSD/Total
50%	303.9	11500	2250	14053.9	82%
90%	480.6	62500	280395	343375.6	18%

In contrast, SA suffers from no such limitation, because it does not have anything corresponding to the BSD phase. As soon as the transient phase is over, each engine is immediately able to begin sending estimates to the global analyser. As a result, the speedup obtained from SA, shown in Fig. 3, is very close to that predicted by Eq. (1).

We can conclude that, in general, BM is not well suited for parallelisation using MRIP. This limitation can also be expected to apply to other methods which involve batching, such as those based on Standardized Time Series [SCHR83]. Much better speedup can be obtained from MRIP using a method such as SA which does not have a large startup overhead.

4. Conclusions.

We have presented here the results of our ongoing research into the MRIP scenario of quantitative stochastic simulation. While this scenario is potentially able to deliver a speedup proportional to the number of processors, whether this is obtained in practice can depend heavily on the method used to estimate variance. We have found that the method of Batch Means is not well suited for use with MRIP, and that the method of Spectral Analysis is a better choice.

Many other aspects will also have to be carefully studied before MRIP can be safely used in simulation practice. These include the properties of the global estimators used in MRIP, which are weighted sums of local classical estimators.

Finally, let us note that if a simulation is distributed by SRIP, resulting in a speedup of S_{SRIP}, and one applies MRIP to such an SRIP-based model by launching P independent replications of it, then the total speedup will be PS_{SRIP}, as long as $P \leq P_{min}$.

References

BILE85 W.E.Biles, C.M.Daniels and T.J.O'Donnell. "Statistical Considerations in Simulation on a Network of Multicomputers". Proc. 1985 Winter Simulation Conf. (IEEE Press) 388-393

EWIN95 G.Ewing, D.McNickle and K.Pawlikowski. "Credibility of the Final Results from Quantitative Stochastic Simulation". Proc. 1995 EUROSIM Conf. (Elsevier Science Publ., 1995) 189-194

EWIN96 G.Ewing, K.Pawlikowski and D.McNickle. "Akaroa II Version 1.2.1. User's Manual". Technical Report TR-COSC 03/96, Department of Computer Science, University of Canterbury, Christchurch, New Zealand, 1996

FUJI90 R.Fujimoto. "Parallel Discrete Event Simulation". *Comm. of the ACM* **33**, 1990, 30-60

GAIT90 B.Gaither. "Empty Empiricism".*ACM Performance Evaluation Review* **18** (2) 1990, 2-3

HEID81 Heidelberger, P., and P.D. Welch. "A Spectral Method for Confdence Interval Generation and Run Length Control in Simulation". *Comms. of ACM* **25**, 1981, 233-245

HEID83 P.Heidelberger and P.D.Welch. "Simulation Run Length Control in the Presence of an Initial Transient". *Operations Res.*. **31**, 1983, 1109-1144

KLEI79 J.P.C.Kleijnen. "The Role of Statistical Methodology in Simulation". In: *Methodology in Systems Modelling and Simulation*, B.P.Zeigler et al. (eds), North-Holland, Amsterdam, 1979.

LAW83 A.M.Law. "Statistical Analysis of Simulation Output Data". *Operations Res.* **31** (6) 1983, 983-1029

MCNI96 D.C.McNickle, K.Pawlikowski and G.Ewing. "Experimantal Evaluation of Confidence Intervals Procedures in Sequential Steady-State Simulation". Proc. 1996 Winter Simulation Conf. (IEEE Press, 1996), in press

PAWL90 K.Pawlikowski. "Steady-State Simulation of Queueing Processes: A Survey of Problems and Solutions". *ACM Computing Surveys* **22** (2) 1990, 123-170

PAWL94 K.Pawlikowski, V.Yau and D McNickle. "Distributed Stochastic Discrete-Event Simulation in Parallel Time Streams". Proc. 1994 Winter Simulation Conf. (IEEE Press, 1994) 723-730

SCHR83 L.Schruben. "Confidence Interval Estimation Using Standardised Time Series". *Operations Res.*, 1983, 1090-1108.

WAGN89 D.B.Wagner, and E.Lazowska. 1998. "Parallel Simulation of Queueing Networks: Limitations and Potentials". *Performance Evaluation Review* **17**, 1989, 146-155.

YAUP93 V.Yau and K.Pawlikowski. "AKAROA: a Package for Automatic Generation and Process Control of Parallel Stochastic Simulation". Proc. 16th Australian Computer Science Conf., *Australian Computer Science Comms.* **15** (1) 1993, 71-82