PERFORMANCE EVALUATION OF INDUSTRIAL PROCESSES IN COMPUTER NETWORK ENVIRONMENTS

K. Pawlikowski, G. Ewing and D. McNickle
University of Canterbury
Christchurch, New Zealand
E-mail: krys@cosc.canterbury.ac.nz

KEYWORDS
performance evaluation, network computing, quantitative stochastic simulation, distributed simulation, speedup

ABSTRACT

In this paper we look at an application of distributed computing power of computer networks in simulation studies of industrial systems modelled by dynamic stochastic processes. Stochastic simulation is an invaluable tool for studying such processes but, unfortunately, obtaining the final results with an acceptable level of precision, even in the case of moderately complex systems can be computationally very intensive. In industrial practice, the most commonly executed simulation studies belong to the so-called terminating simulation, during which the time horizon of simulated processes is well defined. For example, one can be interested in assessing throughput of a production line in a factory during eight hours of work. In this paper we look at a novel scenario for running such simulation experiments, known as Multiple Replications in Parallel (or MRIP), and show that MRIP offers a speedup equal to the number of processors employed as simulation engines, providing that this number of processors does not exceed some, usually very large, number. This is illustrated by the numerical results of two exemplary simulation studies. The results were obtained by running simulations under control of AKAROA-2, an automatic controller of distributed simulation, designed by the authors at the University of Canterbury. The main features of AKAROA-2 are also discussed.

1. INTRODUCTION

Continuous technological progress has led to more and more complicated industrial systems. Performance quality of industrial processes has become probably the most important factor deciding about the position, or even existence, of a company at today's competitive international market, and simulation of industrial systems has become probably the most commonly used method of their performance evaluation (Law and Kelton 1991, p.2; Banks et al. 1996, p.6). Continuous monitoring of performance quality of processes in business, manufacturing, transport, telecommunication and other industries, as well as performance evaluation studies of their alternatives, would not be possible without significant achievements in computer and telecommunication technology, which have led to proliferation of local networks connecting numerous computer workstations in powerful, distributed data processing systems. Computing power of these fast growing distributed network environments has not been yet widely realised. There is the need for proposing new applications of networked computers, both those already connected by existing local networks and those which will be connected by emerging global AAA network, able to offer Any type of information service, Anywhere in the world and at Any time.

In this paper we explore applications of distributed processing power of networked computers in simulation studies of industrial systems modelled by dynamic stochastic processes. Stochastic simulation is an invaluable tool for studying such processes but, unfortunately, simulation of even moderately complex systems can become computationally very intensive and can require very long runs in order to obtain the required precision of final results. This is specially relevant if one takes into account that the only effective way of controlling the final precision of results of such performance evaluation studies is to run simulations sequentially (Law and Kelton 1991, p.563), i.e. checking precision of the results at consecutive checkpoints of simulation. This allows the simulation to be run until the required precision of results is obtained; see (Pawlikowski 1990) for more detailed analysis of sequential simulation stopping rules.

In industrial practice, the most commonly executed performance evaluation studies are connected with the so-called terminating simulation, during which the time horizon of simulated processes is well defined. For example, one can be interested in assessing throughput of a production line in a factory during eight hours of work, or in estimating roundtrip times of buses in a city during rush hours.

Execution of simulated processes can be sped up by applying distributed processing. We will briefly characterise the "traditional" scenario of distributed terminating simulation, known as SRIP (Single Replication in Parallel) (Yau and Pawlikowski 1993; Pawlikowski et al. 1994) and will focus our attention on MRIP (Multiple Replications in Parallel), a newer scenario for distributed terminating simulation; see (Yau and Pawlikowski 1993; Pawlikowski et al. 1994) for a discussion of MRIP in the context of steady-state simulation. We will consider theoretical properties of MRIP and illustrate them by the numerical results of two exemplary simula-

tion studies. The results were obtained by running simulations under control of AKAROA-2, an automatic controller of distributed stochastic simulation designed by the authors at the University of Canterbury (Ewing et al. 1997). The main features of this controller, when it is used for executing terminating simulation of industrial processes in the MRIP scenario, will be also discussed.

2. TWO SCENARIOS OF QUANTITATIVE SIMULATION IN NETWORK ENVIRONMENT

To shorten the duration of simulation of any industrial process one can try to divide the simulated model into submodels and run them separately on different computers of a network, while maintaining the necessary exchange of data between submodels. Such a scenario of distributed simulation, which in the context of quantitative stochastic simulation is known as *SRIP* (Single Replication in Parallel) (Yau and Pawlikowski 1993; Pawlikowski et al. 1994), has been discussed in detail in (Fujimoto 1990) and (Bagrodia 1996). In this case, a number of computers co-operate in executing global computing task in the same way that a number of factories co-operate in manufacturing different parts for the same global product. Possible applications of this scenario are limited by the fact that it can work well only when studying performance of loosely-coupled industrial systems, naturally divisible in autonomous subsystems.

These limitations are irrelevant for the second scenario of quantitative stochastic simulation, known as *MRIP* (Multiple Replications in Parallel); see (Yau and Pawlikowski 1993; Pawlikowski et al. 1994) for discussion of its properties in the context of steady-state simulation. Following our analogy with co-operating factories, we can see it as a co-operation at integrated manufacturing level: to increase the production rate of a final product, the same product is produced by a number of factories. This, in the case of terminating stochastic simulation, means that a number of networked computers operate as simulation engines producing statistically identical output data, which are submitted to a global analyser. The global analyser is responsible for checking precision of results whenever new data is available, and for stopping the simulation when all final results have achieved the required precision. Speedup of simulations run in MRIP scenario is due to increased rate of production of simulation output data because of parallel generation of these data by a number of networked computers.

Since any stochastic simulation should be regarded as a statistical experiment (although simulated on a computer) and, as such, generates random output results (or observations), one has to collect a sufficient number of such observations to ensure that the final estimates have acceptably low statistical error. Such errors are usually measured by relative precision of the estimates, defined as the relative half-width of their confidence intervals at a given confidence level; see (Pawlikowski 1990) for alternative measures of statistical error for simulation results. Output data collected during terminating simulation should be analysed by using the method of Independent Replications (IR); see e.g. (Law and Kelton 1991). Usually one replication produces a single data item, or observation, per performance measure. These observations are used to calculate the final estimate(s) of the analysed performance measure(s).

Having accepted that the sequential quantitative stochastic simulation is the only effective way of controlling the final precision of results (Law and Kelton 1991, p.563), one should determine a sequence of checkpoints at which the precision of simulation results is analysed. Following available recommendations (see e.g. (Law and Kelton 1991, p.563), as well as relying on the authors' own experience, it is justified to have the first checkpoint when the results of first three to five replications are available. Subsequent checkpoints could be determined whenever a new replication is completed. The length of any such performance evaluation study based on such a sequential stochastic simulation can be measured by the number of replications required for obtaining the final results with an acceptable level of relative precision. Speedup of distributed terminating simulation run under the MRIP scenario on computers of a local area computer network is discussed in the next section.

3. SPEEDUP OF TERMINATING SIMULATION IN MRIP SCENARIO

Let us assume that a stochastic terminating simulation is run on P identical, uniprocessor computers, or workstations, linked by a local area computer network (LAN). Thus, randomness in parallel streams of simulation output data generated by different computers is caused by randomness of simulated processes only. The (average) speedup S of distributed simulation in the MRIP scenario can be defined as the ratio of the (average) simulation length run on a single computer and the (average) simulation run length on P computers. In a typical situation, each replication of simulation adds a single observation to analysis of a given performance measure. Thus, the simulation run length can be measured by the number of independent replications needed for collecting the minimum number of observations required for estimating a given performance measure with the acceptable level of precision. Assuming a fine granularity of sequential simulation, i.e. a small distance between consecutive checkpoints, the speedup obtainable when running an MRIP simulation on P computers should be governed by Amdahl's law, see e.g. (Kant 1991), which states that

$$S = 1/\left(f+(1-f)/P\right) \qquad (1)$$

where f represents the fraction of the simulation process which cannot be parallelised. In the case of terminating simulation f=0, since, contrary to steady-state simulation, there is no warm-up stage (Pawlikowski 1990). Thus, one can expect that the speedup is linear, equal to the number of computers operating as simulation engines. There is a limit, though, caused by the stopping rule of such simulation. Namely, according to the prin-

ciples of sequential stochastic simulation, the simulation is stopped when the number of collected observations is sufficient for achieving the required precision of the final results. Let:

N_{min} be the (average) total number of observations needed for stopping the simulation with the required relative precision, at a given confidence level; and

Δ be the number of observations collected by a simulation engine between two consecutive checkpoints.

Trying to use more and more simulation engines, one will reach a situation in which all computers are able to reach their first checkpoints only and the simulation stops, since N_{min} observations have been collected already. Let the average number of computers used at this moment be

$$P_{max} = \min\{P : P\Delta \geq N_{min}\} \qquad (2)$$

Adding more computers than P_{max} will not increase the speedup since it has already reached its limit value $S_{max}=P_{max}$. The only effect of having more observations (generated by $P>P_{max}$ processors) would be a better-than-required precision of the final results. Since terminating simulation is based on IR, it is natural to assume that $\Delta=1$, i.e. that the precision of results is checked whenever new replication is finished. This gives us the following *truncated Amdahl's law* regulating the (average) speedup of distributed terminating simulation in MRIP scenario, when it executed on P computers linked by a LAN:

$$S = \begin{cases} P & \text{, for } P < P_{max} \\ N_{min}/P_{max} & \text{, for } P \geq P_{max} \end{cases} \qquad (3)$$

This formula is graphically depicted in Fig.1.

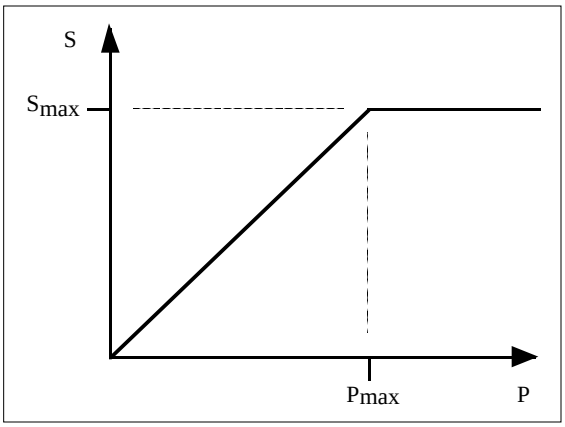

Fig. 1. Speedup obtainable in MRIP scenario of distributed simulation

Obviously, the best speedup can be achieved if one launches simulation engines on homogeneous computers. In the case of heterogeneous computers, faster simulation engines will execute more replications than slower ones. In an extreme case, one computer, much faster than others, could generate all N_{min} replicationsneeded and the simulation would be finished before any of other computers is able to reach its first checkpoint.

Let us note that SRIP and MRIP are not mutually exclusive scenarios of distributed terminating simulation. SRIP is a natural solution for simulating large simulation models, too large for being executed on a single computer. Such an SRIP simulation can be accelerated by replicating clusters of computers, each cluster engaged in simulating the same industrial process(es) in SRIP scenario. Let us assume that using N_1 computers for an SRIP simulation one can achieve speedup S_1. Then, replicating this SRIP simulation N_2 times following the MRIP scenario, i.e. using N_2 clusters of N_1 computers each, one can speedup SRIP simulation S_2 times. The total speedup of such MRIP of SRIP, or MRIP/SRIP, simulation would be S_1S_2..

4. IMPLEMENTATION OF SEQUENTIAL TERMINATING SIMULATION IN AKAROA-2

The rules of MRIP scenario have been implemented in AKAROA-2 (a simulation package for automatic generation and control of processes for parallel stochastic simulation) which accepts ordinary (non-parallel) simulation programs, and, fully automatically, creates the distributed environment required for running MRIP on computers of a local computer network. Our main considerations when selecting a development language and designing programming interface were: simplicity, space and code efficiency, as well as compatibility with our existing sequential simulation programs. Recognising the naturalness of the object-oriented approach in constructing simulation models by means of hierarchically encapsulated classes of objects, AKAROA-2 is written in C++. A user is required to add only one extra line of code to his/her ordinary sequential simulation program written in C or C++ for linking the program with AKAROA-2. Then the program is transparently taken over by control processes of AKAROA-2 and executed on multiple computers of a LAN. Thus, users do not even need to be aware of the existence of multiple (parallel) simulation engines and global analysers, since their creation, locations (machine and port addresses), co-operation, and inter-machine inter-process communication, are hidden from users. Launching of simulation engines, collection of distributed output data and their analysis is under full control of a central controlling process called *akmaster*. Interprocess communication is based on UNIX Internet-domain stream sockets[*] .

[*] In the previous version of AKAROA, the interprocess communication was based on UNIX Internet-domain datagrams (Pawlikowski et al. 1994).

5. NUMERICAL RESULTS

In this section we present the results of two case studies conducted by means of terminating simulation run in MRIP scenario, and executed under control of AKAROA-2. All final results were obtained with the relative precision not worse than 5%, at the 0.95 confidence level.

In Case A we studied a single-machine-tool system considered by (Law and Kelton 1991) and (Clark 1994). The analysis was done from the point of view of a company which considered the purchase of a new machine tool from a vendor. The expected time spent on processing a part by the machine, taking into account its possible breakdowns, was analysed over the period of eight working hours, assuming the advertised performance characteristics of the machine. Those led to the assumption of times between breakdowns being exponentially distributed with mean equal 540 minutes, processing times of all jobs being identical and equal 1 minute, and repair times being governed by lognormal distribution with mean and variance equal 60 and 400, respectively. The conducted simulation can be classified as a terminating simulation of an M/D/1/∞ queuing system with a server taking (lognormally distributed) vacation times. The estimated mean time spent by the simulated machine on processing a part (including the machine's repair times) as a function of the input load, is depicted in Fig. 2.

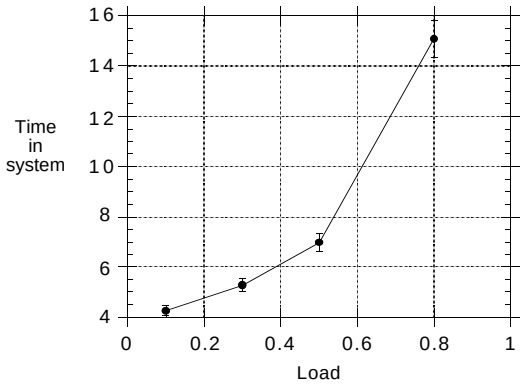

Fig.2. Case A: Mean time spent on processing a part by the simulated machine, including machine's down times. Relative precision < 5%

In Case B we studied performance of an automatic call-waiting telephone system during busy two hours of mid-day. The purpose of this study was to decide about the numbers of telephone operators needed for providing satisfactory quality of service for customers. One of the performance measures considered was the number of customers who were put on hold but did not wait until a connection was offered and hung off. The conducted simulation can be classified as a terminating simulation of an M/M/c/∞ queuing system with impatient customers leaving the queue if they waited for service too long. Both lengths of periods of "patience" and lengths of connections were assumed to be exponentially distributed with mean of 2 and 0.5 minute, respectively. The estimated mean number of impatient customers, who resigned from being connected, for c=1 telephone operator, as a function of the input load, is depicted in Fig. 3.

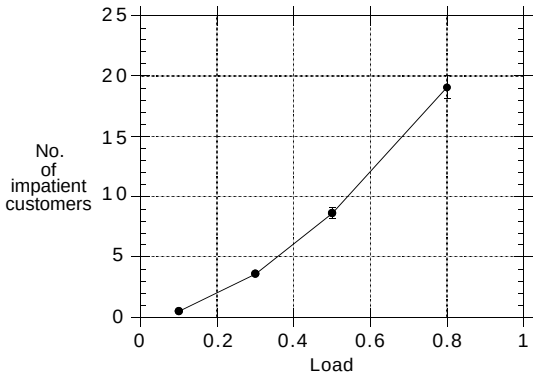

Fig.3. Case B: Mean number of impatient customers in a 2 hour period of time. Relative precision < 5%

In both cases, simulations were run at P= 1, 5, 10 and 20 processors. The average number of replications executed per one of P simulation engines in Case A (for input load equal 0.8) and in Case B (for input load equal 0.1) is shown in Table 1. These and other results obtained from running MRIP simulations of Case A and B on P= 5, 10 and 20 computers gave mean speedups equal about 5, 10 and 20, respectively. Thus, our experimental results confirmed the correctness of Eq.(3).

Table 1: Mean number of replications per engine (averaged over 100 simulations)

P	Case A	Case B
1	2350	2970
5	475	611
10	236	307
20	119	152

Table 2. Mean total number of replications needed

Load	0.1	0.3	0.5	0.8
Case A	1850	2017	2329	2350
Case B	2970	577	265	129

The mean total number of replications needed for stopping simulation with the required level of relative precision (of 5% or better) in Case A and B is shown in Table 2. One can see that depending on the performance measure, the length of simulation can either increase or decrease with the input load. The table gives us approximate values of S_{max}, the maximum possible speedup in each case, c.f. Eq. (3) and Fig.1. For example in Case A, when studying the mean processing time of a part at the input load of 0.3, one can expect that, when running simulation on 2017 or more identical computers, the simulation can be (on average) at most 2017 times shorter than on a single computer. Such speedup can be of course practically significant only if the time duration of a single replication is practically significant.

6. CONCLUSIONS

Computer processing power distributed in computer networks has not been fully utilised yet and there is a need for proposing applications which could benefit in such distributed computing environments. In this paper we have discussed a new scenario for executing performance evaluation studies of industrial processes by means of terminating stochastic simulation, called MRIP (Multiple Replications in Parallel) We have shown that speedup of MRIP simulations is ruled by the Truncated Amdahl's Law. This has been confirmed by the experimental results obtained from two performance evaluation studies in which simulations were run under control of AKAROA-2, a user-friendly package designed at the University of Canterbury. AKAROA-2 automatically creates parallel simulation engines, distributes them over different computers of a network, and controls execution of simulated processes until they are automatically stopped when the precision of results reaches its acceptable level. For statistical properties of estimators used in MRIP in more sophisticated simulations than terminating ones, see (Pawlikowski et al. 1998).

REFERENCES

Bagrodia, R.L. 1996 "Perils and Pitfalls of Parallel Discrete Event Simulation". *Proc. 1994 Winter Simulation Conf.*, IEEE Press, 136-143

Banks, J., J.S.Carson II and B.L.Nelson. 1996. *Discrete-Event System Simulation*. Prentice Hall

Clark., G.M. 1996. "Introduction to manufacturing applications". *Proc. 1994 Winter Simulation Conf.*, IEEE Press, 15-21

Ewing, G., K.Pawlikowski and D.McNickle. 1997. "Akaroa 2.4.2. User's Manual". Tech. Report TR-COSC 07/97, August 97, Department of Computer Science, University of Canterbury, Christchurch, New Zealand

Fujimoto, R. 1990. "Parallel discrete event simulation". *Communications of the ACM* **33**, 30-60

Kant, K. *Introduction to Computer System Performance Evaluation*. McGraw-Hill

Law, A.M. and W.D.Kelton. 1991. *Simulation Modelling and Analysis*. McGraw-Hill

Pawlikowski, K. 1990. "Steady-state simulation of queueing processes: a survey of problems and solutions". *ACM Computing Surveys* **22,** 123-170

Pawlikowski, K., V.Yau and D McNickle. 1994. "Distributed stochastic discrete-event simulation in parallel time streams". *Proc. 1994 Winter Simulation Conf.*, IEEE Press, 723-730

Pawlikowski, K., D.McNickle and G. Ewing. 1998. "Coverage of confidence intervals in steady-state simulation". *J. Simulation Practise and Theory, Vol.6, No.3, 1998, pp.255-267*

Yau, V. and K.Pawlikowski. 1993. "AKAROA: a package for automatic generation and process control of parallel stochastic simulation". *Proc. 16th Australian Computer Science Conf., Australian Computer Science Comms.* **15,** 71-82

KRZYSZTOF PAWLIKOWSKI is an Associate Professor (Reader) in Computer Science at University of Canterbury, Christchurch, New Zealand. He received his PhD degree in Computer Engineering from the Technical University of Gdansk, Poland. The author of over 80 research papers and four books. His research interests include stochastic simulation, distributed processing, ATM and optical telecommunication networks, and teletraffic modelling. Senior member of IEEE.

GREG EWING is a research associate in the Department of Computer Science at Canterbury. He has completed a Ph.D. on geographic information systems. His research interests include programming languages, 3D graphics, graphical user interfaces and parallel processing.

DON MCNICKLE is a Senior Lecturer in Management Science on the Department of Management, University of Canterbury, New Zealand. His research interests include queuing theory and statistical aspects of simulation. He received a PhD in Mathematics from the University of Auckland, New Zealand. Member of ORSA.