GETTING SPEEDUP WITH ACCURACY IN SIMULATION EXPERIMENTS OF COMMUNICATION NETWORKS

Edjair Mota, Adam Wolisz
Technical University of Berlin, Germany
Krzysztof Pawlikowski
University of Canterbury, Christchurch, New Zealand
e-mail: mota@ft.ee.TU-Berlin.DE

Keywords: Efficient simulation, multiple replications, parallelization, computer networks, MAC protocols

ABSTRACT

With the advent of new communication technologies, modern networks tend to be more complex and there is a necessity of efficient tools for enhancing the computational effort of their simulations. Multiple Replications in Parallel (MRIP) is an attractive approach for this purpose, since it can decrease the simulation cost and increase the accuracy of the results. Akaroa.2, a user-friendly package is an MRIP implementation, and has been found easy to install, use and extend. We present some issues concerning Akaroa’s statistical properties, which indicate that it is a good candidate for simulating communication networks with efficiency.

1 INTRODUCTION

Nowadays we witness a great challenge in the computing area. As the communication technologies advance and computer processing power increases, new kind of applications arise, e.g., integrated services. As a consequence, the interest of new users is aroused, which implies that new requirements and applications, and thus other users, will certainly appear.

Dynamic increasing of the complexity of the networks and the growth of the number of users require efficient tools for analysing them and improving their performance. Straightforward simulation of such complex systems as communications networks takes frequently a prohibitive amount of time despite the increasing of processing speed of modern computers. It’s not unusual a simulation experiment to take hours, even days, for yielding reasonable information.

Reason for that, among others, is the statistical nature of the simulation experiments. Most simulation models contain stochastic input variables and, thereby, stochastic output variables are generated, the last ones being used for estimating the characteristics of the real system. In order to obtain an accurate estimation it is necessary to collect a substantial amount of simulation output data, and this can require very long simulation run length.

This paper addresses the necessity of tools for enhancing the computational efficiency of computer communication network models, and particularly presents our works on enhancing functionality of Akaroa.2, a user-friendly implementation of Multiple Replications in Parallel — MRIP, which launches ordinary sequential simulation in parallel and averages the results (Pawlikowski et al. 1994). As it is known, the MRIP approach can decrease the simulation cost and automatically control the accuracy of the results.

Section 2 overviews main problems associated with simulation experiments of communication networks and their possible solutions. Section 3 presents an MRIP implementation in Akaroa.2. Section 4 deals with the problem of accuracy of the results and discusses some confidence interval procedures (CIPs) proposed for this purpose. Section 5 compares the performance of two CIPs and comments...
on what can be expected when applying them in sequential stochastic simulation together with automatic parallelization in MRIP scenario. Section 6 contains final remarks and directions to further research.

2 PROBLEM OVERVIEW

Suppose we are conducting a simulation experiment to investigate the behavior of a new medium access control (MAC) protocol and we come across one of the following problems:

1. The experiment takes very long time to conclude.
2. We need to analyse several variants of this protocol.
3. We want to have higher confidence in the credibility of the results.

A possible solution to the first two issues would be to increase the statistical efficiency of our simulation experiment by means of (i) applying procedures which can reduce the variability of estimators being analysed or (ii) making use of more computational resources.

Alternative (i) is generally known as Variance Reduction Techniques — VRT, a set of experimental design and analysis techniques used to achieve a desired precision in shorter simulation time (Law and Kelton 1991). Although their efficacy is by all means recognized (e.g. Frost 1988; Izydorczyk et al. 1984), they can sometimes require great additional effort from the analyst in terms of statistical background. Alternative (ii) has been commonly seen as a pool of cooperative hosts, each one processing a part of the same model (submodel) and, therefore, synchronization among these parts is a relevant issue.

One can also take advantage of the available computational resources of computer networks by applying an approach called Multiple Replications in Parallel (see Pawlikowski et al. 1994 for a discussion of its properties in the context of steady-state simulation). Here, each processor runs an instance of the entire simulation model (hereafter called simulation engine), and the intermediate results are analysed by a central process. A faster processor can run more than one independent simulation engine.

Despite of its simplicity, MRIP approach answers satisfactorily the first two problems listed at the beginning of this section, and, moreover, offers a quite simple solution for the third and last problem, concerned with the credibility of the final simulation results.

3 AKAROA.2

Akaroa.2 is an MRIP implementation designed at the Department of Computer Science of the University of Canterbury in Christchurch, New Zealand, to run stochastic discrete-event simulation on Unix multiprocessor systems or networks of heterogeneous Unix workstations. It makes uses of an efficient interprocess communication (IPC), which facilitates the exchange of messages among the simulation engines and the central process.

Parallelization process is completely transparent from the point of view of the analyst, who does not even need to be aware of the existence of such parallel structure. She/he is just required to add one extra line to code her/his sequential simulation model before Akaroa.2 takes care of its parallel execution.

Akaroa.2 provides its own sources of randomness and ensures that the simulation engines are using non-overlapping sequences of random numbers, thus cross-observations are independent and identically distributed. The question of bias produced by the initial transient, pointed out by (Heidelberger 1986) as the greatest challenge for this kind of approach, is taken into account by applying an algorithm conceived by Schruben (Schruben 1982). For a revision of this and other problems related to steady-state simulation readers are referred to survey (Pawlikowski 1990).

Crash of a simulation engine, or a too slow behavior of some processors imposes no obstacle to Akaroa.2’s task, since it demands no order of arrivals for the intermediate results yielded by different simulation engines. Just the overall simulation run length takes a little longer, proportional to the number of lost simulation engines. It works as if the pool of processors had fewer processors. On the other hand, it is obvious that MRIP works better in an homogeneous environment.

Akaroa.2 is also equipped with a library of routines suitable for modeling communication networks, but it can also be used in an integration with other simulation tools written in C or C++, or at least capable of calling routines in those languages. Akaroa.2 has been linked with such simula-
T he accuracy of the results

During analysis of simulation output data we can make use of one of several procedures found in the literature, which construct confidence intervals of mean value at a confidence level assigned by the analyst, in order to assess the accuracy of the results. These procedures are hereafter called Confidence Interval Procedures, or CIPs.

Actually, Akaroa implements two CIPs, based on Nonoverlapping Batch Means (NOBM) and a version of Spectral Analysis proposed by Heidelberger and Welch (SA/HW). The former is the Confidence Interval Procedure that Akaroa has been most widely used for. The one of Spectral intervals is based on Nonoverlapping Batch Means (NOBM), although not implemented in the current version of the program. The procedure is intended to replace the confidence interval procedure that is used in the model generated by the parallel processing system under the scheduling of Confidence Interval Procedures.

These CIPs were originally conceived for a single processor environment, and it's interesting to investigate their behavior under a parallel computing environment created by Akaroa, by applying them on models with known analytical solution. Preliminary investigations showed us that adding more processors doesn't involve exactly proportional decrease in the number of observations, but the framework implemented by Akaroa can collect more observations during a shorter interval of time, which implies a reasonable speedup. In other words, Akaroa is a natural variance reducer, without any additional effort from the analyst.

Considering the popularity of Batch Means approach due to its simplicity of understanding and implementation, we considered as relevant implementing sequential versions of other variants of Batch Means, for instance Overlapping Batch Means (OBM) proposed by Mekelet and Schmeiser (Mekelet /1984/), and Spaced Batch Means (SBM) proposed by Fox et al. (Fox /91/). It is out of the scope of this work to discuss them in detail, but it is worth mentioning that OBM is an approach in which the only additional information passed by the analyst is the degree of overlap. SBM takes into account the fact that the usual strong correlation among the observations in the stationary phase can be reduced by discarding some observations between batches.

Fig. 1: Queueing models with increasing coefficient of variation

<table>
<thead>
<tr>
<th>Model</th>
<th>X</th>
<th>0.25</th>
<th>1</th>
<th>1.414</th>
</tr>
</thead>
<tbody>
<tr>
<td>M/D/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M/E4/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M/M/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M/H2/1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Suppose that during simulation of a communication network we have linked Akaroa with SMURPH. In the Telecommunication Network Group at the Technical University of Berlin, we have linked Akaroa with SMURPH.

Theoretical context

With SINR/W, we have linked Akaroa with SMURPH. The communication network was simulated at the Technical University of Berlin at all 1998. The current version of SMURPH contained the model of a single processor environment. Some observations in the literature have shown that the assumption of independence of observations is not always valid. Therefore, we have decided to investigate the behavior of Akaroa under the conditions of parallel computing.

Theoretical context

With SINR/W, we have linked Akaroa with SMURPH. The communication network was simulated at the Technical University of Berlin at all 1998. The current version of SMURPH contained the model of a single processor environment. Some observations in the literature have shown that the assumption of independence of observations is not always valid. Therefore, we have decided to investigate the behavior of Akaroa under the conditions of parallel computing.

Theoretical context

With SINR/W, we have linked Akaroa with SMURPH. The communication network was simulated at the Technical University of Berlin at all 1998. The current version of SMURPH contained the model of a single processor environment. Some observations in the literature have shown that the assumption of independence of observations is not always valid. Therefore, we have decided to investigate the behavior of Akaroa under the conditions of parallel computing.

Theoretical context

With SINR/W, we have linked Akaroa with SMURPH. The communication network was simulated at the Technical University of Berlin at all 1998. The current version of SMURPH contained the model of a single processor environment. Some observations in the literature have shown that the assumption of independence of observations is not always valid. Therefore, we have decided to investigate the behavior of Akaroa under the conditions of parallel computing.

Theoretical context

With SINR/W, we have linked Akaroa with SMURPH. The communication network was simulated at the Technical University of Berlin at all 1998. The current version of SMURPH contained the model of a single processor environment. Some observations in the literature have shown that the assumption of independence of observations is not always valid. Therefore, we have decided to investigate the behavior of Akaroa under the conditions of parallel computing.

Theoretical context

With SINR/W, we have linked Akaroa with SMURPH. The communication network was simulated at the Technical University of Berlin at all 1998. The current version of SMURPH contained the model of a single processor environment. Some observations in the literature have shown that the assumption of independence of observations is not always valid. Therefore, we have decided to investigate the behavior of Akaroa under the conditions of parallel computing.

Theoretical context

With SINR/W, we have linked Akaroa with SMURPH. The communication network was simulated at the Technical University of Berlin at all 1998. The current version of SMURPH contained the model of a single processor environment. Some observations in the literature have shown that the assumption of independence of observations is not always valid. Therefore, we have decided to investigate the behavior of Akaroa under the conditions of parallel computing.
are true, then
\[ P(\bar{X}(n) - H \leq \mu \leq \bar{X}(n) + H) = 1 - \alpha \]
where
\( \bar{X} \) is the mean for the \( n \)-sized sample, defined as
\[ \bar{X}(n) = \frac{1}{n} \sum_{i=1}^{n} x_i \]
and
\( H \) is the half-length of the confidence interval estimator of \( \mu \) defined as
\[ H = t_{n-1,1-\frac{\alpha}{2}} \sigma[\bar{X}] \]
where
\( \sigma[\bar{X}(n)] \) is the standard deviation of \( \bar{X}(n) \), and
\[ t_{n-1,1-\frac{\alpha}{2}} \] is the upper \((1 - \frac{\alpha}{2})\) critical point of the \( t \) distribution with \( n-1 \) degrees of freedom.

When the assumptions are satisfied, the coverage \( \eta \) has a uniform distribution. (Schruben 1980) suggests the construction of an empirical distribution function \( G_{\eta'}(\eta) \) with the observed values of confidence level \( \eta' \) for a sample size \( n \). This empirical function presents more information than a single value of coverage, e.g. at \( \eta = 0.9 \), a common practice to summarize empirical results in coverage analysis. Ideally, \( G_{\eta'}(\eta) = \eta \) but usually two other situations can arise:

- \( G_{\eta'}(\eta) < \eta \), occurs when positive serial correlation has been ignored, which means that the final \( \eta \) was overestimated.
- \( G_{\eta'}(\eta) > \eta \), occurs when negative serial correlation has been ignored, which can lead to wasting the time in sequential procedures, since it may result in collecting more observations than necessary for achieving a desired precision.

We have applied the sequential coverage analysis proposed in (Pawlikowski et al. 1998), since it gives us a more exact information than that obtained from fixed-sample versions, to investigate the coverage function proposed by (Schruben 1980). Our analysis stopped when the relative precision of the mean value of the coverage compared to the half-length of its confidence interval was less than 5%.

To assess the performance of the CIPs we used queueing models with known analytical solution, 90% loaded. As a matter of fact, we chose queueing models with increasing coefficient of variation of its service times (See Figure 1) \( \text{for } M/D/1 \) to \( \sqrt{2} \) (for \( M/H_2/1 \)). Table 1 summarizes the results of such sequential analysis by using 2, 4, 6, 8 and 10 processors, at a 90% of confidence level.

After finding sequentially the optimal batch size \( M \), which guarantees an acceptably low correlation among the batch means at a significance level \( \beta \), the sequential procedure based on \text{OBM} divides each batch into four sub-batches, each of which initiates an overlapping batch of size \( M \). Originally, (Meketon 1984) suggested that each observation should initiate an overlapping batch, but we found that satis-

Fig. 2: Coverage function under MRIP, for a sequential version of the method \text{OBM} using \( P \) processors \((M/M/1, \rho = 0.9)\).
factory variance reduction can be achieved with this computationally simpler overlapping scheme.

Since the sequential determination of the optimal batch size, SBM discards an amount s of observations between consecutive batches. Although greater values of s should improve a little bit the final coverage, we adopted a parsimonious approach and fixed s equal 20% of the initial batch size provided by the user. Recent investigations, however, showed us that SBM seems to perform somewhat better when s is chosen dynamically, based on the correlation structure of the underlying process being simulated.

Both CIPs produced acceptable coverages, except in 1 of 40 experiments, when the coverage was much bellow than 80%, but it seems to be a statistical variation that does not invalidate our conclusions.

E[T] stands for the average duration (in seconds) of each experiment. Here, Akaroa.2 shows an attractive time reduction in both CIPs. For OBM this reduction increased with the increasing coefficient of variation of the service times.

E[O] stands for the average total number of observations (x10^6), required for achieving the desired precision of the final results. Following this criterion, OBM overcame SBM (at least in the actual implementation, with no attempt of tuning), and required a little bit more observations.

Figures 2 and 3 give some examples of the coverage function for the CIPs investigated. With little exception, both procedures performed closely to the ideal case, but OBM seems to present less fluctuations. Better results should be obtainable after tuning each of these CIPs, and this is a subject of our current research.

6 FINAL REMARKS

Multiple Replications in Parallel is a very promising methodology for enhancing the computational efficiency of computer communication network simulation. The provided speedup is limited by number of processors involved, and no additional effort is required from the analyst for launching parallel simulation engines, if one uses such package as Akaroa.2. Interesting issues for further research includes the implementation of other CIPs and the analysis of the asymptotic behavior of these CIPs to know how they will perform as the run length of a real simulation model increases.
Table 1: Performance comparison of OBM and SBM under MRIP, for \( P = 2, 4, 6, 8 \) and 10 processors.

<table>
<thead>
<tr>
<th>Processor</th>
<th>OBM</th>
<th>SBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>M^2</td>
<td>M^2/2</td>
<td>M^2/2</td>
</tr>
<tr>
<td>2</td>
<td>0.875</td>
<td>0.266</td>
</tr>
<tr>
<td>4</td>
<td>0.865</td>
<td>0.865</td>
</tr>
<tr>
<td>6</td>
<td>0.864</td>
<td>0.864</td>
</tr>
<tr>
<td>8</td>
<td>0.861</td>
<td>0.861</td>
</tr>
</tbody>
</table>

References


