World-Wide Distributed Multiple Replications in Parallel for Quantitative Sequential Simulation

Mofassir Haque[1] , Krzysztof Pawlikowski [1], Don McNickle [2], Gregory Ewing [1],

[1] University of Canterbury, Department of Computer Science,
Christchurch 8140, New Zealand
[2] University of Canterbury, Department of Management,
Christchurch 8140, New Zealand
Mofassir.Haque@pg.canterbury.ac.nz
{Krys.Pawlikowski, Don.McNickle , Greg.Ewing,}@ canterbury.ac.nz

Abstract. With the recent deployment of global experimental networking facilities, dozens of computer networks with large numbers of computers have become available for scientific studies. Multiple Replications in Parallel (MRIP) is a distributed scenario of sequential quantitative stochastic simulation which offers significant speedup of simulation if it is executed on multiple computers of a local area network. We report results of running MRIP simulations on PlanetLab, a global overlay network which can currently access more than a thousand computers in forty different countries round the globe. Our simulations were run using Akaroa2, a universal controller of quantitative discrete event simulation designed for automatic launching of MRIP-based experiments. Our experimental results provide strong evidence that global experimental networks, such as PlanetLab, can efficiently be used for quantitative simulation, without compromising speed and efficiency.

Keywords: Multiple Replications in Parallel, Experimental networking facilities, Akaroa2, PlanetLab, Sequential quantitative stochastic simulation. Open queuing network.

1 Introduction

Quantitative stochastic simulation of complex scenario can take hours or days to complete. SRIP (Single Replication in Parallel) and MRIP (Multiple Replication in Parallel) are two methods used to reduce simulation time. In SRIP, the simulation program is divided into smaller logical parts and run on different computers. In MRIP, multiple processors run their own replications of sequential simulation, but cooperate with central analyzers (one central analyzer for each performance measure analyzed) that are responsible for analyzing the results and stopping the simulations when the specified level of accuracy is met [1]. The MRIP technique can significantly speed up simulation if replications are launched on a larger homogeneous set of computers [2, 3].

In last few years, a large number of experimental networking facilities have been, or are being developed across the globe: e.g. PlanetLab, GENI, OneLab, G-Lab, Akari, Panlab, etc. [4]. These global networks often consist of thousands of

computers. Thus they provide a viable alternative for running distributed stochastic simulations in the Multiple Replications in Parallel scenario (MRIP). We selected PlanetLab as the provider of distributed computing resources for investigating various aspects of MRIP simulations, since it is a continuously evolving computing platform with thousands of nodes [5]. These nodes can be easily accessed for running MRIP without investing in infrastructure. However, before using such a globally distributed networking facility for sequential stochastic simulation on multiple computers, factors such as load at selected nodes and potential communication overhead between them have to be carefully considered, as these computers can be shared by a large number of users and some of them are thousands of miles apart. Load generated by these users can vary significantly and quickly. Thus, it can adversely affect performance of computers, and the simulations running on them.

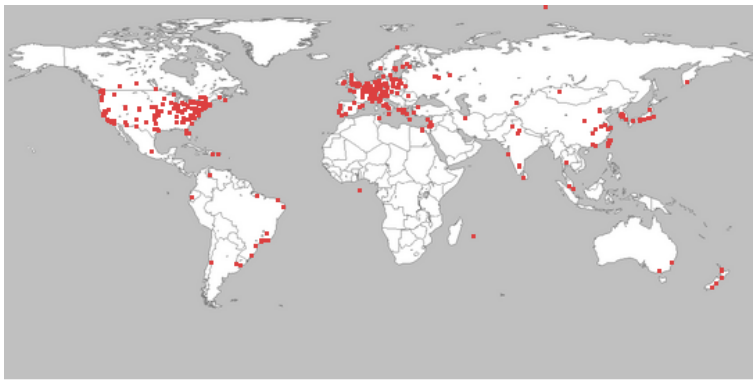

Fig. 1. PlanetLab with deployed nodes around the world [5]

We did extensive experimentation to determine the suitability of PlanetLab nodes for MRIP simulations. Our simulations were run with Akaroa2, a universal controller of quantitative discrete event simulation, designed for automatic launching of MRIP-based experiments. Experiments were designed to measure times needed to produce final simulation results over various sets of PlanetLab computers. Results obtained from the experiments executed over PlanetLab nodes were compared with the results obtained from running MRIP simulations on a local area network at the University of Canterbury. This has allowed us to conclude that a global networking facility such as PlanetLab can be effectively utilized for running MRIP.

The rest of the paper is organized as follows. Section 2 spells out the procedure for running Akaroa2 on PlanetLab. Sections 3 explains in detail the experimental set up and evaluation metric. Section 4, presents experimental results, and conclusions are in Section 5.

2 Akaroa2 on PlanetLab

In Akaroa2, multiple independent replications of a stochastic simulation are run on different processors, which play the role of independent simulation engines producing

statistically equivalent output data during one simulation. Multiple simulation engines cooperate with the global analyzer that processes streams of output data coming out from different simulation engines, and stops the simulation once the required accuracy of the results has been achieved. The accuracy is typically measured by the relative statistical error of the results. Two main processes of Akaroa2 are: Akmaster and Akslave. The Akslave process initiates simulation engines on multiple processors, while Akmaster controls sequential collection of output data and their analysis. It collects local estimates from all running Akslaves, calculates final global estimates, displays results, and then terminates the simulation when the stopping criterion is reached [6]. Both steady-state simulations and terminating simulations are supported. In the former case, the procedures for sequential mean and variance analysis are described in [1, 7-8], while the procedure adopted for terminated simulation is presented in [2]. Akaroa2 is widely used for simulations executed on local area networks, as its records of the last 10 years (in July 2011) show over 3100 downloads of the software by users from over 80 countries [9].

In order to run Akaroa2 on PlanetLab, first we need to copy and install Akaroa2 on all the nodes which will be used for running simulation engines. Copying and installing software on hundreds of machines is an intricate task. Either the CoDeploy program [10] provided by PlanetLab or, alternatively, simple shell scripts for automating copying, installation and running of Akaroa-2 on PlanetLab can be used. The shell script we used can be downloaded from the PlanetLab New Zealand web site [11]. For proper execution of MRIP-based simulation, the path variable should be correctly set in the bash profile file of all participating PlanetLab nodes, and simulation program should be copied in the directory specified in the path. The detailed procedure with step by step instructions for running Akaroa-2 on PlanetLab using Linux or Windows operating system can be downloaded from PlanetLab New Zealand web site [11].

3 Experimental Setup

To study the feasibility of running Akaroa2 on PlanetLab, we conducted a large number of experiments, considering different strategies for selecting participating nodes of the network. The aim was to measure the times to produce simulation results, from the time instant when the simulation was launched until the time instant when the final results were obtained, to find out how using differently selected sets of PlanetLab nodes can affect users' quality of experience, in comparison with simulations executed on local computers only.

We compared two of many possible strategies for selection of PlanetLab nodes for MRIP simulations. We assumed that the computers are either distributed over a restricted geographical region (so they operate in the same or close time zones), or they are distributed globally (so computers participating in MRIP simulations work in very different time zones).

3.1 Computing Setup CS1:

In this computing setting, while operating from New Zealand, we installed Akaroa2 on PlanetLab nodes spread over the European Union. The Akmaster was installed in Italy and simulation engines were located in France, UK, Belgium, Italy, Hungary and Poland. PlanetLab nodes were carefully selected using the CoMon utility [12] to avoid currently heavily loaded nodes. The CoMon utility is provided by PlanetLab for monitoring of resource utilization of all PlanetLab nodes. In CS1 our aim was to assess response times of MRIP-based simulation experiments. The experiments were run on Friday, beginning at 2pm British Standard Time.

3.2 Computing Setup CS2:

In this computing environment, simulation engines of Akaroa2 were installed world-wide, so they operated in very different time zones. Again, while operating from New Zealand, we installed the Akmaster in Italy, and the simulation engines were launched in Europe, USA, Canada, New Zealand and Asia; see Figure 2. Nodes were again carefully selected using the CoMon utility, avoiding nodes which were heavy loaded. This setup was used to study and verify effect of communication overhead when simulation engines are thousands of miles apart. The experiments were run on Friday, beginning at 2pm USA Central Standard Time.

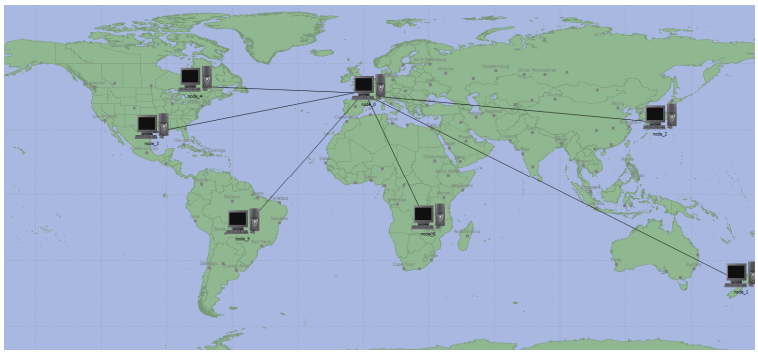

Fig. 2. Global distribution of Akaroa2 in CS2

Note that the nodes of PlanetLab used by Akaroa2 represented a homogenous set of computers, as the computers of PlanetLab have to satisfy some minimum common technical requirements. For comparing the quality of users' experience in such distributed simulation environments, we have also measured the waiting times for simulation results in the traditional local processing environment of Akaroa2, where its simulation engines are located around a local area network.

3.3 Computing Setup CS3:

Here, the simulation experiments were run on computers linked by a local area network in a computer laboratory of the Department of Computer Science and Software Engineering, at the University of Canterbury, in Christchurch. Akmaster and Akslave were installed in this controlled local area network environment, the original home location of Akaroa2. The results were used as the reference for comparison with the results obtained from the two other, distributed computing environments. The experiments were run on Friday, beginning from 2pm, New Zealand time. The nodes of the local area network, physically located in one laboratory, constitute a homogenous set of computers. Laboratory and PlanetLab nodes are equipped with quad processors and both use the Fedora operating system based on the Linux Kernel. However, the computers available on PlanetLab are of slightly higher technical standards in terms of memory and clock frequency than those available in our CS3 setting.

3.4 Simulation Setting and Evaluation

We ran the same sequential stochastic simulation in MRIP scenario in all three computing setups: CS1, CS2 and CS3. For our study, we simulated a simple open queuing network, consisting of a CPU and two disk memories with unlimited buffer capacities, depicted in Figure 3. We estimated steady-state mean response (mean time spent by a customer in this system), assuming that arriving customers form a Poisson process with $\lambda = 0.033$ tasks per second. All service times are exponentially distributed with mean service time at the CPU of 6 seconds, mean service time at Disk 1 and mean service time at Disk 2 both of 14 seconds. This makes the servers to the CPU, Disk 1 and Disk 2 loaded at 96%, 92.4% and 92.4%, respectively.

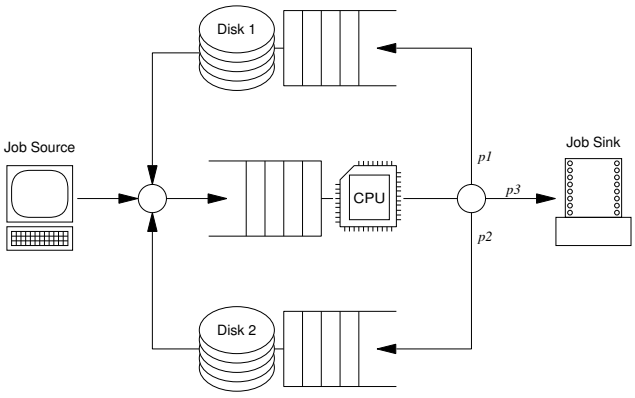

Fig. 3. Simulated open queuing network

The simulation processes on all computers were to stop when the estimate of steady-state mean response reached a relative statistical error not greater than 5%, at a

confidence level of 0.95. This should require about 20 million observations. Running simulation in Multiple Replications in Parallel scenario allowed us to collect this sample of output data faster, as it is produced by multiple simulation engines.

To demonstrate that this attractive feature of MRIP remains practical also in case of globally distributed simulation engines, we assessed speedup and relative efficiency of MRIP simulations in setup CS1 and CS2, and compared the results with those from locally distributed simulation engines in CS3. The performance of our MRIP simulations was assessed by measuring response time (RT) of a given simulation setting, defined as the time interval between the time of launching the simulation until the time when the final results are delivered to the user. Then, the speedup of simulation at $P > 1$ simulation engines can be found as:

$$S(P)=\frac{Mean\_RT(1)}{Mean\_RT(P)} \quad \textbf{(1)}$$

where Mean_RT (P) is mean response time of P simulation engines running MRIP simulation with $P \geq 1$. Alternatively, we looked at the relative speedup of MRIP simulation, defined as

$$SR(P)=\left(\frac{Mean\_RT(1)-Mean\_RT(P)}{Mean\_RT(1)}\right)*100\ \% \quad \textbf{(2)}$$

for P= 1, 2, 3, …. Note that, due to the truncated Amdahl law for MRIP formulated in [2, 3], there exists a limit on the number of processors which would increase the speedup of MRIP simulation. It is also known that the largest speedup can be obtained in homogeneous computing environments. In the extreme case, if one simulation engine uses a very fast processor and remaining processors are slow, a simulation will not benefit at all from MRIP at all, as the fastest simulation engine can produce the entire sample of required observations needed for stopping the simulation, before any of the remaining slower simulation engines is able to reach its first checkpoint.

Another performance measure which we considered is the efficiency of distributed processing during MRIP simulation, or speedup per simulation engine:

$$E(P) = \frac{S(P)}{P} \quad \textbf{(3)}$$

In an ideal situation, the efficiency would be equal to one. However, in practical applications of parallel processing it is usually much smaller. E (P) measures how well the contributing processors are utilized for solving a given problem, despite their mutual communication and synchronization activities.

4 Experimental Results

In this section, we present our experimental results obtained under computing setups CS1, CS2 and CS3. We use mean response time as the measure of quality for testing

our hypothesis that the MRIP scenario can also be efficiently used in the case of world-wide distributed simulation engines. The mean response times obtained for CS1, CS2 and CS3, measured in seconds, are given in Table 1. Each reported result is an average over 10 independent measurements/simulations. The relative statistical errors of these estimates are not larger than 1% for CS3 and not larger than 6 % for CS1 and CS2, at 0.95 confidence level.

Table 1. Mean response time for scenario CS1, CS2 and CS3

Number of Nodes	CS1	CS2	CS3
2	88.13	97.53	59.78
4	61.94	75.48	47.53
6	52.25	64.74	37.08
8	45.23	59.34	32.81
10	39.8	46.81	29.98
12	34.32	43.21	28.62
15	27.14	36.35	15.67

Fig. 4 compares mean response times of CS1, CS2 and CS3. The histogram clearly shows that mean response time reduces as the number of nodes increases. The PlanetLab nodes are being shared by a large number of users and are located hundreds of miles apart. Conversely, laboratory nodes are used by only one person and are located close to each other. The mean response times in case of CS3 are therefore smaller than in the case of PlanetLab nodes both in CS1 and CS2. In order to obtain good performance, PlanetLab nodes should be carefully selected, avoiding heavily loaded nodes and busy working hours.

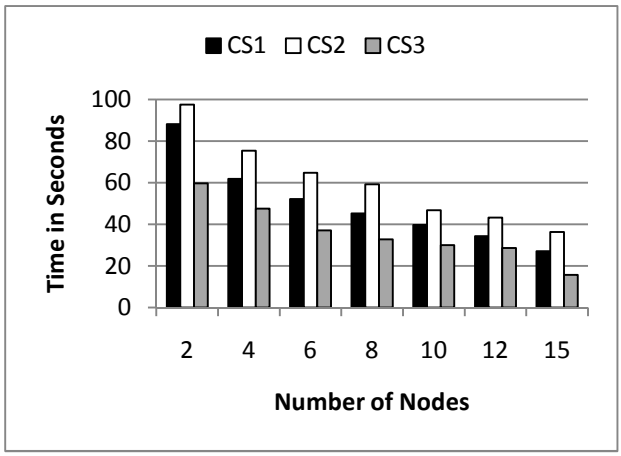

Fig. 4. Comparison of mean response times in CS1, CS2 and CS3

Comparison of the mean response times for CS1 and CS2 shows that these mean response times are much shorter if all the PlanetLab nodes are selected from one area

(continent), for example within Europe rather than from all over the world. This is primarily because of communication overhead. When controller and simulation engines are located thousands of mile apart, the time used for exchanging data between simulation engines and controller directly effects the mean response time. We also ran the same experiment by selecting PlanetLab nodes from North America only and found results similar to the setup CS2.

Speedup for distributed scenario of CS1 and CS2 is calculated using Equation (1) and given in Table 2. Speedup has been calculated using mean response time of two nodes as a reference. In spite of the longer distance between nodes, speedup offered by PlanetLab nodes in the case of CS1 is better than in CS3, because of the slightly better hardware of PlanetLab nodes.

Table 2. Speedup for distributed scenario of CS1 and CS3

Number of Nodes	CS1	CS3
2	1	1
4	1.42	1.26
6	1.69	1.61
8	1.95	1.82
10	2.21	1.99
12	2.57	2.09
15	3.24	3.17

Efficiency in the case of CS1 and CS3 has been calculated using Equation (3) and is shown in table 3. In this case, there is only small difference between the results. The efficiency decreases as the number of processors increases. This is due to the fact that processor communication is usually slower than computation and exchange of local estimates between Akslaves and Akmaster results in frequent communication.

Table 3. Efficiency for scenario CS1 and CS3

Number of Nodes	CS1	CS3
2	0.5	0.5
4	0.31	0.35
6	0.26	0.28
8	0.22	0.24
10	0.19	0.22
12	0.17	0.21
15	0.20	0.21

These results allow us to conclude that it has become practical to use distributed computing resources of global experimental networks for fast quantitative stochastic simulation, paying only a small penalty in the form of a minor worsening of response times, speedup and efficiency of the simulation as comparing with the same

simulations run on a local area network. The advantage of using globally distributed computing resources is that they can be substantially larger than the ones available locally.

We conducted experiments using two different ways of selection of computers in PlanetLab for simulation engines and compared their performance with performance of simulation run on computers of a local area network. The performance of MRIP in CS1 appears to be better that in CS2. Thus, for best results selection of computers from closer geographical location, avoidance of both heavily loaded nodes and busy hours is recommended.

5 Conclusions

In this paper we have shown that the distributed computing resources of global experimental networks, such as PlanetLab, can be effectively used for running quantitative stochastic simulations in MRIP scenario. Only a small penalty (in the form of a minor worsening of performance) is paid for using globally distributed resources instead of local ones. Launching and on-line control of globally distributed simulations can be done by using for example Akaroa2.

It is encouraging news for those who need to run time-consuming quantitative simulations to get accurate final results, but do not have access to sufficiently large number of computers for launching multiple simulation engines. Recently, there has been a surge in development of global and regional experimental networking facilities, see Table 4 [13]. Most of these networks offer free membership and can be effectively used for conducting simulation experiments under control of Akaroa2.

Table 4. Selected experimental networking facilities, with size and accessibility

Name	Purpose	Size	Access
OneLab	Multipurpose	Regional	Free membership
Panlab	Multipurpose	Regional	Planned to be on Payment
Federica	Multipurpose	Regional	Free membership
PlanetLab	Multipurpose	Global	Free membership
GENI	Multipurpose	Regional	Free membership
JNB 2	Multipurpose	Regional	Free membership
CNGI	Multipurpose	Regional	Free membership

In future, we plan to investigate the upper bounds for speedup of globally distributed sequential stochastic simulation, such as those in the MRIP scenario. This will require running experiments at full scale, employing hundreds of PlanetLab nodes as simulation engines, with simulations requiring extremely large samples of output data for producing accurate simulation results, in particular if the simulated processes are strongly correlated.

Acknowledgments. This work was partially supported by REANNZ (2010/2011 KAREN Capability Build Fund).

References

1. Pawlikowski, K., Yau, V., McNickle, D., Distributed stochastic discrete-event simulation in parallel time streams. In: 26th conference on Winter simulation, pp. 723--730. Society for Computer Simulation International, Orlando, Florida, United States(1994)
2. Pawlikowski, K., Ewing, G., McNickle, D.: Performance Evaluation of Industrial Processes in Computer Network Environments. In: European Conference on Concurrent Engineering, pp. 129-135. Int. Society for Computer Simulation, Erlangen, Germany (1998)
3. Pawlikowski, K., McNickle, D.: Speeding Up Stochastic Discrete-Event Simulation. In: European Simulation Symposium, pp. 132-138. ISCS Press, Marseille, France (2001)
4. Lemke, M.: The Role of Experimentation in Future Internet Research: FIRE and Beyond. In: 6th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities, Berlin, Germany (2010)
5. PlanetLab, http://www.planet-lab.org/
6. Ewing, G., Pawlikowski, K., McNickle, D.: Akaroa-2: Exploiting Network Computing by Distributing Stochastic Simulation. In: 13th European Simulation Multi-Conference, pp. 175-181. Warsaw, Poland (1999)
7. Ewing , G., Pawlikowski, K.: Spectral Analysis for Confidence Interval Estimation Under Multiple Replications in Parallel. In: 14th European Simulation Symposium. pp. 52-61, ISCS Press, Dresden, Germany (2002)
8. N. Shaw, McNickle, D., Pawlikowski, K.: Fast Automated Estimation of Varience in Sequential Discrete Event Stochistic Simulation. In: 25th European Conference on Modelling and Simulation, Krakow, Poland (2011)
9. Akaroa2, http://www.cosc.canterbury.ac.nz/research/RG/net_sim/simulation_group/akaroa/about.chtml
10. CoDeploy, http://codeen.cs.princeton.edu/codeploy/
11. PlanetLab NZ, http://www.planetlabnz.canterbury.ac.nz/
12. CoMon, http://comon.cs.princeton.edu
13. Haque, M., Pawlikowski, K., Ray, S.: Challenges to Development of Multipurpose Global Federated Testbed for Future Internet Experimentation. In: 9th ACS/IEEE International Conference on Computer Systems and Applications, Sharm El-Sheikh, Egypt (2011)