Distributed steady-state simulation of telecommunication networks with self-similar teletraffic

Hae-Duck J. Jeong a,*, Jong-Suk R. Lee b, Don McNickle c, Krzysztof Pawlikowski c

a Department of Information Science, Korean Bible University, Seoul, South Korea
b Department of Grid Technology Research, Supercomputing Center, Korea Institute of Science and Technology Information, Daejeon, South Korea
c University of Canterbury, Christchurch, New Zealand

Received 12 February 2003; received in revised form 9 August 2004; accepted 13 October 2004
Available online 8 December 2004

Abstract

Recent measurement studies of teletraffic data in modern telecommunication networks have shown that self-similar processes may provide better models of teletraffic than Poisson processes. If this is not taken into account, it can lead to inaccurate conclusions about performance of telecommunication networks. We show how arrival processes with self-similar input influences the run-length of a distributed steady-state simulation of queueing systems in telecommunication networks. For this purpose, the simulation run-length of SSM/M/1/∞ queueing systems in the method based on the batch means, conducted for estimating steady-state mean waiting times is compared with the results obtained from simulations of M/M/1/∞ queueing systems when a single processor and multiple processors are used. We also investigate speedup conducted stochastic simulation of SSM/M/1/∞ queueing systems on multiple processors under a scenario of distributed stochastic simulation known as MRIP (Multiple

* Corresponding author. Address: 205 Sanggye 7-dong, Nowon-gu, Seoul, South Korea 139-791. Tel.: +82 17 289 8848; fax: +82 2 950 5451.
E-mail addresses: joshua@bible.ac.kr, hdjeong@hanmail.net (H.J. Jeong), jsruthlee@kisti.re.kr (Jong-Suk R. Lee), don.mcnickle@canterbury.ac.nz (D. McNickle), krys@cosc.canterbury.ac.nz (K. Pawlikowski).
Replications In Parallel) in a local area network (LAN) environment on Solaris operating system. We show that, assuming self-similar inter-event processes (i.e., $SSM/M/1/\infty$ queueing systems), many more observations are required to obtain the final simulation results with a required precision, as the value of the Hurst parameter $H$ increases, than when assuming Poisson models, exhibiting short-range dependence (i.e., $M/M/1/\infty$ queueing systems) on a single processor and multiple processors. Our results show that the time for collecting many numbers of observations under the MRIP scenario is clearly reduced as traffic intensity and the value of the Hurst parameter increase, and as the engaged processor increases one to four. In particular, the value of $H$ influences much more the speedup than traffic intensity and the engaged processor.

© 2004 Elsevier B.V. All rights reserved.

Keywords: Long-range dependent self-similar process; Hurst parameter; Distributed steady-state simulation; Simulation run-length; Speedup

1. Introduction

There are two issues considered in this paper.

- First, we show how self-similarity of arrival processes influences the run-length of sequential stochastic simulation of queueing systems. For this purpose, the simulation run-length of $SSM/M/1/\infty$ queueing systems, \(^1\) conducted for estimating steady-state mean waiting times is compared with the results obtained from simulations of $M/M/1/\infty$ queueing systems. In both cases, simulations were stopped when the final estimates were achieved with the relative precision (see Section 4) not larger than 10%, for a 95% confidence interval (CI).
- Second, we investigate speedup conducted stochastic simulation of $SSM/M/1/\infty$ queueing systems on multiple processors under the MRIP scenario \([23,24,34]\).

Leland et al. \([25]\) claimed that a self-similar process is needed to model the behaviour of a time series well enough to capture its impact on queueing performance, while Ryu and Lowen \([41]\) claimed that the self-similarity of the arrival process needs to be taken into account only when the traffic intensity and the buffer size are very large. Heyman \([16]\) examined how the Markov-modulated Poisson process and FBM model proposed in the literature perform on two data sets of LAN traffic. Both models overestimated loss probabilities in the area of interest. The effects of a finite buffer of size $b$ could be estimated by calculating the probability that the infinite buffer has greater than $b$ items in it. This approximation may be fundamentally inaccurate for calculating small loss probabilities and modest values of $b$. Heyman \([17]\) also found that the approximation is more accurate for heavy-tailed distributions rather than those without power-law tails. Further investigation of the accuracy of this approximation is required.

\(^1\) Here, $SSM$ means a self-similar arrival process with an exponential marginal distribution.
Resnick and Samorodnitsky [38] and Dahl and Willemain [5] investigated that LRD arrival processes can significantly affect the performance of queueing systems. Neidhardt and Wang [32] pointed out that arrival processes with larger values of $H$ do not always cause longer queues. On the other hand, the results of Grossglauer and Bolot [14] showed that the behaviour of buffer can be more sensitive to the marginal distribution of arrival processes than to the $H$ value of these processes. Further, Boxma and Cohen [3] studied an approximation for the waiting time distribution for an M/G/1 queueing system with a heavy-tailed service time distribution by using the heavy traffic limit theorem. As the resulting approximation shows, there is a need to develop more accurate models of long-range dependent traffic and studying their impact on queueing systems and networks. Because of that, more work is needed to fully understand all the effects of self-similarity on queueing processes.

Jeong et al. [20] found that when self-similar input with $H \geq 0.6$ is assumed in a steady-state simulation of queueing models using the spectral analysis method, one needs many more observations to obtain the final simulation results with a relative precision not greater than 10%, at 95% confidence interval, than when assuming Poisson processes. The new contributions of this paper to the field of modelling of self-similar teletraffic for simulation in telecommunication networks are in the following. Influence of input $H$ values on the run-length of sequential distributed steady-state simulation of SSM/M/1/∞ queueing systems in the method based on batch means has been investigated when one and multiple processor(s) are used. We find that the speedup is clearly increased as traffic intensity and the value of the Hurst parameter increase, and as the engaged processor increases one to four. In particular, the value of $H$ influences much more the speedup than traffic intensity and the engaged processor.

The purpose of our investigations of these problems is to make a quantitative assessment of the influence of $H$ values on a single processor and multiple processors. A summary of the basic properties of self-similar processes is given in the next section. We study a method, based on Fractional Gaussian Noise and Daubechies Wavelets (FGN-DW) which has promising properties [19]. Generation and transformations of the FGN-DW processes are described in Section 3. In Section 4, we assess the simulation run-length for self-similar processes. Batch means methods of sequential simulation output data analysis of mean values are given in Section 5. Theoretical speedup in the MRIP Scenario is described in Section 6. We show the number of observations required in a distributed steady-state simulation of queueing models with self-similar input processes, and compare our results with simulation run-length of the same queueing models fed by short-range dependent processes such as the M/M/1/∞ queueing system before the final conclusions are formulated.

\footnote{Sequential procedures sequentially determine the length of a simulation run needed to construct an acceptable confidence interval for the parameter. With this approach, the analyst can automatically control the statistical error by specifying a stopping criterion [23].}
2. Self-similar processes and their properties

Self-similar (or fractal) long-range dependent stochastic processes occur in many natural and man-made systems. In particular, since such processes were discovered in the Internet and other multimedia telecommunication networks a decade ago [25,47] they have become the subject of numerous research investigations of their nature and consequences.

Theoretically, one can distinguish two types of stochastic self-similarity. A continuous-time stochastic process \( Y_t = \{Y_{t_1}, Y_{t_2}, \ldots\} \) is strictly self-similar with Hurst parameter \( H \), \( 0.5 < H < 1 \), if \( Y_{ct} \) and \( c^HY_t \) (the rescaled process with time scale \( ct \)) have identical finite-dimensional distribution for any stretching factor \( c, c > 0 \), [2,33,45]. This means that, for any sequence of time points \( t_1, t_2, \ldots, t_n \)

\[
\{Y_{ct_1}, Y_{ct_2}, \ldots, Y_{ct_n}\} \overset{d}{=} \{c^HY_{t_1}, c^HY_{t_2}, \ldots, c^HY_{t_n}\},
\]

where \( \overset{d}{=} \) denotes equivalence in distribution. Thus, a strictly self-similar process (or “self-similar in a narrow sense”) is self-similar in the sense of probability distribution.

Weak self-similarity of stochastic processes (or “self-similarity in a broad sense”) becomes relevant when one restricts analysis of stochastic processes to their first two moments only, i.e. to their means, variances and co-variances. Because of that, it is also known as second-order self-similarity. For example, let a sequence \( \{X_1, X_2, \ldots\} \) be a time-stationary stochastic process, defined at discrete time instances \( i = 1,2,3 \ldots \). Let \( E[X_i] = EX \), \( \text{Var}[X_i] = \text{Var}X \), and \( \rho_k = E[(X_i - EX)(X_{i+k} - EX)]/\text{Var}X \) denote its mean, variance and autocorrelation coefficient of lag \( k \), respectively. Having grouped this sequence of random variables into batches of size \( m, m \geq 1 \), we can define the aggregated process of \( X^{(m)}_i = \{X^{(m)}_1, X^{(m)}_2, \ldots\} \), at a given aggregation level \( m \), where

\[
X^{(m)}_i = \frac{1}{m} \left( X_{i-1} + \cdots + X_i \right), \ i \geq 1.
\]

If \( \{X_1, X_2, \ldots\} \) is a weakly self-similar time-stationary process, with \( 0.5 < H < 1 \), then

\[
\text{Var}[X^{(m)}_i] = m^{2H-2}\text{Var}X
\]

and

\[
\rho^{(m)}_k = \rho_k, k \geq 0,
\]

where \( \rho^{(m)}_k \) is the autocorrelation coefficient of lag \( k \) of the aggregated process \( X^{(m)} \). Eq. (3) means that the original process and its aggregated version have identical correlation structure. Further, it can be proved that in any weakly self-similar process with \( 0.5 < H < 1 \)

\[
\rho_k \sim H(2H - 1)k^{2H-2}
\]

as \( k \to \infty \), and

\[
\sum_{k=-\infty}^{\infty} \rho_k = \infty,
\]
see for example [33,40]. Thus the autocorrelation function decays very slowly (hyperbolically) with $k$. Because of these properties, these processes are also known as long-range dependent, or strongly correlated [4,28,29].

3. Teletraffic generation and simulation

We claim that the $FGN-DW$ transformation is sufficiently fast for generation of synthetic self-similar sequences, to be used as simulation input data [18,19]. Let us briefly introduce the $FGN-DW$ method itself.

3.1. Generation of self-similar teletraffic using $FGN-DW$

Step 1. Calculate a sequence of values $\{f_1, f_2, \ldots, f_n\}$ using Eq. (8) (following), where $f_i = f(\frac{2\pi}{n}; H)$, corresponding to the spectral density of an FGN process for frequencies $f_i$ ranging between $\xi$ and $\pi$.

For a fractional Gaussian noise (FGN) process, the spectral density $f(k, H)$ is given by

$$ f(k, H) = 2c_f (1 - \cos(\lambda)) \mathcal{B}(\lambda, H) $$

with $0 < H < 1$ and $-\pi \leq \lambda \leq \pi$, where

$$ c_f = \sigma^2 (2\pi)^{-1} \sin(\pi H) \Gamma(2H + 1), $$

$$ \mathcal{B}(\lambda, H) = \sum_{k=-\infty}^{\infty} |2\pi k + \lambda|^{-2H-1}, $$

and $\sigma^2 = \text{Var}[X_\lambda]$ and $\Gamma(\cdot)$ is the gamma function; see [2]. The main difficulty with using Eq. (6) to compute the spectral density is the infinite summation. The approximation of $f(\lambda, H)$ is given in [2] as

$$ f(\lambda, H) = c_f |\lambda|^{-2H} + O(|\lambda|^{\min(3-2H, 2)}), $$

where $c_f$ is Eq. (7) and $O(\cdot)$ represents the residual error. This formula was used in the generation of self-similar sequences proposed in this paper. Another generator of self-similar sequences based on FGN was also proposed by Paxson [37], but his method was based on a more complicated approximation of $f(\lambda, H)$. Eq. (8) can be used to determine $f(\lambda, H)$ for $\lambda \to \infty$, or for $n \to \infty$ at $\lambda = \frac{\pi}{n}$. For a large value of $\lambda$, $f(\lambda, H)$ can be calculated by Eq. (6).

Step 2. Multiply $\{f_i\}$ by realisations of an independent exponential random variable with a mean of one to obtain $\{\hat{f}_i\}$, because the spectral density estimated for a given frequency is distributed asymptotically as an independent exponential random variable with mean $f(\lambda, H)$ [1].

Step 3. Generate a sequence $\{y_1, y_2, \ldots, y_p\}$ of complex numbers such that $|y_i| = \sqrt{f_i}$, and the phase of $y_i$ is uniformly distributed between 0 and $2\pi$. This random phase technique, taken from Schiff [42], preserves the spectral density.
corresponding to \( \{ \tilde{f}_i \} \). It also makes the marginal distribution of the final sequence normal and produces the requirements for FGN.

**Step 4.** Calculate the two synthetic coefficients of orthonormal Daubechies wavelets that are used in the inverse DWT (IDWT) [12]. Then, generate the approximately self-similar FGN sequence \( \{ X_i \} \) in the time domain by using the IDWT from \( \{ y_i \} \).

Using the previous steps, the proposed FGN-DW method generates a fast and sufficiently accurate self-similar FGN process \( \{ X_1, X_2, \ldots, X_n \} \). It took 16 seconds to generate a sequence of 1,048,576 numbers on a Pentium II (233 MHz, 512 MB). Its theoretical algorithmic complexity is \( O(n) \). Moreover, the accuracy of Daubechies wavelets is slightly better than Haar wavelets, but there is no difference in the time taken to obtain the same number of coefficients. For more detailed discussion, see [18].

3.2. Transformation of self-similar processes with an exponential marginal distribution

Simulation studies of telecommunication networks require a mechanism to transform strictly and/or second-order self-similar processes into different processes with arbitrary marginal distributions [26,27,37]. Therefore, a mechanism is needed to transform self-similar processes representing the arrival counts, or differential arrival rates, into suitable sequences of inter-arrival times of packets, while preserving appropriate characteristics [18]. In this paper, we use a self-similar process with an exponential marginal distribution generated by FGN-DW, using the inverse cumulative distribution function (ICDF) transformation.

The ICDF transformation is based on the observation that given any random variable \( X_i \) with a cumulative distribution function (CDF) \( F(X) \), the random variable \( u = F(X) \) is independent and uniformly distributed between 0 and 1. Therefore, \( u \) can be obtained by generating uniform realisations and calculating \( X = F^{-1}(u) \). Here we only consider the exponential marginal distribution of output processes that is frequently used in simulation practice. The exponential distribution has the CDF

\[
F_X(u) = 1 - e^{-\lambda u},
\]

where \( \lambda \) is a scale parameter.

To generate processes with an exponentially marginal distribution \( Y \) from normal distribution \( X \), one applies the transformation:

\[
y_i = -\left( \frac{1}{\lambda} \right) \log(F_N(x_i)), \tag{9}
\]

where \( F_N(\cdot) \) is the CDF of the normal distribution [18,22].

4. Required run-length of simulation for mean waiting times of \( M/M/1/\infty \) queueing systems

In a typical simulation, a variance or a mean of parameters (such as response times, waiting times and queue lengths) is unknown. Nevertheless, simulation
practitioners would like to be able to plan a simulation and, in particular, estimate how long the simulation must be run so as to obtain a credible confidence interval [36]. Depending on which steady-state parameters are estimated in a queueing system, a different number of observations must be collected to reach the required confidence interval. Here, we considered only the mean waiting time from the $M/M/1/\infty$ queueing system. These were estimated with a specified error (or precision) for a 95% confidence interval. The theoretical number of observations required to estimate these parameters can be obtained as follows. The derivation of a formula that can calculate the theoretically required run-length for the $M/M/1/\infty$ queueing system is ascribed to Daley [6]. We begin by defining a way to measure the relative precision.

Let $\tilde{W}_q$ be the estimate for the mean waiting time in the queue $W_q$. The requirement that the estimate $\tilde{W}_q$ of $W_q$ is obtained with a given relative precision of 5%, for a 95% confidence interval, can be written as

$$\Pr(|\tilde{W}_q - W_q| \leq 0.05W_q) = 0.95$$

(10)

If $\rho = \lambda/\mu$ is the traffic intensity, where $\lambda$ is the arrival rate and $\mu$ is the service rate, then for an $M/M/1/\infty$ queue the theoretical mean waiting time in the queue is obtained by

$$W_q = \frac{\rho^2}{\lambda(1-\rho)}.$$  

(11)

The theoretical variance of the waiting time in the queue is also obtained by

$$\sigma^2(W_q) = \frac{\rho^3(2-\rho)}{\lambda^2(1-\rho)^2}.$$  

(12)

The Laplace transform of a function $f(t)$ is given by

$$B^*(s) = \int_0^{\infty} f(t)e^{-st}dt,$$

in Gross and Harris [13] and Kleinrock [21] (p. 184), and the Laplace-Stieltjes transform of probability density function (PDF) of exponential service times, $B^*(s)$, for the $M/M/1/\infty$ queueing system is defined by

$$B^*(s) = \int_0^{\infty} \mu e^{-\mu t}e^{-st}dt = \frac{\mu}{\mu + s}$$

(13)


The Laplace-Stieltjes transform of PDF of response times $W^*(s)$ in the $M/G/1/\infty$ queueing system is defined as:

$$W^*(s) = \frac{(1-\rho)sB^*(s)}{s - \lambda[1-B^*(s)]}$$

(14)

in Kleinrock [21] (p.199). This simplifies Eq. (13) to

$$W^*(s) = \frac{\mu(1-\rho)}{s + \mu(1-\rho)}$$

(15)
as in Kleinrock [21] (p. 202). The Laplace–Stieltjes transform of PDF of response times from the convolution property of transforms can also be written as

\[ W^*(s) = W^*_q(s)B^*(s) \]

where \( W^*_q(s) \) is the Laplace–Stieltjes transform of PDF of waiting times in the queue, since \( \text{Response Time} = \text{Waiting Time In The Queue} + \text{Service Time} \). Thus, the Laplace–Stieltjes transform of PDF of waiting times in the queue is given by

\[ W^*_q(s) = \frac{(1 - \rho)s}{s - \lambda[1 - B^*(s)]} \] (16)

in [13] (p.202), which simplifies Eq. (13) to

\[ W^*_q(s) = \frac{(s + \mu)(1 - \rho)}{s + (\mu - \rho)} \] (17)

Detailed discussions of Laplace–Stieltjes transforms can be found in Gross and Harris [13] and Kleinrock [21].

If the system has been operating for a long time, and one selects \( n \) observations of waiting times \( W_1, ..., W_n \), then the mean waiting time in the queue

\[ \hat{W}_q = \frac{\sum_{i=1}^{n} W_i}{n} \]

has, for sufficiently large \( n \)

\[ n\sigma^2(\hat{W}_q) \div \sigma^2(W_q) \left[ 1 + 2 \sum_{j=1}^{\infty} \rho_j(m) \right] \] (18)

where

\[ 1 + 2 \sum_{j=1}^{\infty} \rho_j(m) = \frac{1 + \rho}{1 - \rho} + \frac{\lambda(W_q'' - W_q''')}{(1 - \rho)(W_q'' - W_q''')}, \]

where \( W_q'', W_q''' \) and \( W_q'''' \) can be calculated by the first, second and third differentiations of the Laplace-Stieltjes transform of PDF of mean waiting time in the queue \( W_q^*(s) \), and \( \sigma^2(W_q) \) can be calculated by \( (W_q'' - W_q''')W_q''' \) to obtain the number of observations in the queue. For a more detailed discussion, see Daley [6] and Fishman [9].

From Eq. (10), we assume that \( \left( \frac{\hat{W}_q - W_q}{\sigma(W_q)} \right) \) is a normal \( N(0,1) \) distribution, then we have

\[ \Pr \left( \frac{\hat{W}_q - W_q}{\sigma(W_q)} \leq 0.05W_q \sigma(W_q) \right) = 0.95 \]

or

\[ \frac{0.05W_q}{\sigma(W_q)} = 1.96. \] (19)
Then, from Eqs. (18) and (19), we can obtain the following equation [30]. The number of observations needed to estimate the mean waiting times in an $M/M/1/\infty$ queueing system, with a relative precision of 5% for a 95% confidence interval, can be calculated theoretically as

$$n = \left( \frac{1.96}{0.05 W_q} \right)^2 A(\rho),$$

where

$$A(\rho) = \sigma^2(W_q) \left[ \frac{1 + \rho}{1 - \rho} + \frac{\hat{\lambda}(W''_q - W_q W''_q)}{(1 - \rho)(W''_q - W_q W''_q)} \right].$$

Thus, using Eqs. (11) and (12) we get

$$n = \left( \frac{1.96}{0.05 W_q} \right)^2 \left( 2 + 5\rho - 4\rho^2 + \rho^3 \right) = 1536.64 \left( 2 + 5\rho - 4\rho^2 + \rho^3 \right),$$

Table 1 shows the numbers of observations required in theory, when estimating mean waiting times in an $M/M/1/\infty$ queueing system with a relative precision of 5% and 10%, respectively; see also [39]. One can note that in this case we need to run the simulation for approximately four times longer if we want to achieve results twice as accurate.

5. Batch means methods of sequential simulation output data analysis of mean values

Various approaches based on the batch means (BM) have been proposed to discover the best options, such as the number of batches and batch sizes; see for example [10,46]. Automated sequential simulation analysis procedures for implementing the BM can be found in [23,34], and research on methods of BM under the MRIP
scenario can be also found in [31]. The classical estimator known as Non-Overlapping Batch Means (NOBM) (we consider only NOBM which is also commonly called BM) is most widely used in simulation practice to calculate interval estimators from a single (long) simulation run by weakening correlations existing between consecutive data. NOBM requires that sequences of analysed data are stationary. Thus the initial transient observations, collected during the initial transient period, should be discarded. This approach is based on the assumption that observations more separated in time are less correlated. Thus, for sufficiently long batches of observations, the batch means are (almost) uncorrelated. The sequence of \( n \) original observations \( x_1, x_2, \ldots, x_n \) is divided into non-overlapping batches \( (x_{11}, x_{12}, \ldots, x_{1m}), (x_{21}, x_{22}, \ldots, x_{2m}), \ldots \) of each batch size \( m \), sufficiently large so that the mean values over these batches are (almost) independent. Batch means \( \bar{X}_1(m), \bar{X}_2(m), \ldots, \bar{X}_b(m) \), where

\[
\bar{X}_i(m) = \frac{1}{m} \sum_{j=1}^{m} x_{ij},
\]

are used as (secondary) output data in the statistical analysis of the simulation results to obtain the mean and interval estimates of the process. The mean \( \mu_x \) is estimated by

\[
\bar{X}(b,m) = \frac{1}{b} \sum_{i=1}^{b} \bar{X}_i(m),
\]

where \( b \) is the number of batches. A 100(1 − \( \alpha \))% CI for the steady-state mean \( \mu_x \) obtained by applying the method of NOBM is given by

\[
\bar{X}(b,m) \pm t_{b-1,1-\alpha/2} \hat{\sigma}[\bar{X}(b,m)]
\]

where

\[
\hat{\sigma}[\bar{X}(b,m)] = \frac{\sum_{i=1}^{b} (\bar{X}_i(m) - \bar{X}(b,m))^2}{b(b-1)}
\]

is the estimator of the variance of \( \bar{X}(b,m) \), and \( t_{b-1,1-\alpha/2} \), for \( 0 < \alpha < 1 \), is the upper \((1 - \alpha/2)\) critical point from the Student \( t \)-distribution with degrees of freedom \( b - 1 \).

The popularity of NOBM among practitioners continues because of the simplicity of the theory, regardless of reports of relatively poor coverage using this method, especially in heavily loaded systems. This is probably because sometimes batch sizes are accepted, even though they are not sufficiently large enough to obtain uncorrelated batch means. For example, one can select batches of as few as eight observations. Song [44] showed a trade-off between bias and variance for a batch means estimator in accordance with batch sizes. Batches not having optimal numbers of observations can not guarantee that the final results are analysed properly within the NOBM method. Determination of the optimal batch size and the number of batches are definitely problems for the batch means estimator. A few algorithms

\footnote{The initial batch size in Akaroa-2 is 50.}
to determine the best number of batches $b$ and the best batch size $m$, so that the batch means can be assumed to be independent and normally distributed, have been developed; see [23]. Correlation between the batch means of the batch size $m$ can be measured by estimators of the autocorrelation coefficients

$$\hat{r}(k, m) = \frac{\hat{R}(k, m)}{\hat{R}(0, m)}$$

where

$$\hat{R}(k, m) = \frac{1}{b-k} \sum_{i=k+1}^{b} \left[ \bar{X}_i(m) - \bar{X}(n) \right] \left[ \bar{X}_{i-k}(m) - \bar{X}(n) \right]$$

is the estimator of auto-covariance of lag, $k = 0, 1, 2, \ldots$, in the sequence of batch means $\bar{X}_1(m), \bar{X}_2(m), \ldots, \bar{X}_b(m)$. The sequential approach using the method of NOBM has been implemented in Akaroa-2 [8] (Appendix A). An algorithmic description in sequential simulation with this method, implemented in Akaroa-2, can be found in [34].

6. Theoretical speedup in the MRIP scenario

Following Gunther [15], speedup is commonly associated with a measure of parallel numerical performance, and quantifies the reduction in elapsed time achieved by executing a fixed amount of work on a successively greater number of processors. The simplest way of describing the speedup is depicted in Fig. 1 as an ideal parallelism [15]. Ideal parallelism assumes that a total simulation time which runs on a uniprocessor in time $T(1)$ can be fully partitioned and executed on $P$ homogeneous simulation processors simultaneously in time $T(1)/P$ [15]. This can give a linear speedup.

![Fig. 1. Ideal parallelism (taken from [15]).](image-url)
However, most simulations cannot be partitioned in this ideal way because some portion of the simulation needs to be executed sequentially. Therefore, that portion can only be executed on a single processor. This simulation can be classified into two portions: one can execute in parallel and the other can only execute sequentially; see Fig. 2 [15]. In this case, defining the parameter $f < 1$, which is 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), the total simulation time by $P$ homogeneous simulation processors $T(P)$ is $f \cdot T(1)$ (for the sequential portion) plus $((1-f) \cdot T(1))/P$ (for the parallel portion). Therefore, we can write the time reduction, when assuming a simulation executes using $P$ homogeneous simulation processors, as

$$T(P) = f \cdot T(1) + \frac{(1-f) \cdot T(1)}{P}.$$  

The conventional definition of the speedup is given by

$$S(P) = \frac{T(1)}{T(P)}.$$  

Substituting Eq. (29) into Eq. (30), the speedup $S(P)$ is given by

$$S(P) = \frac{T(1)}{(f + \frac{1-f}{P})T(1)}$$  

or

$$S(P) = \frac{P}{1 + f \cdot (P - 1)}.$$  

Speedup achievable with Eq. (32) is based on Amdahl's law. If the value of the parameter $f$ vanishes ($f = 0.0$), then the speedup would follow the ideal linearly
increasing trajectory. Otherwise, depending on the value of $f$, the speedup falls away from the ideal trajectory. As $P \to \infty$, Eq. (32) has an asymptotic bound at $1/f$. Eq. (32) can be regarded as an upper limit of the speedup, since it assumes each parallel subtask is homogeneous with identical processing demands [15]. In reality, applications are less uniform. Therefore, the speedup will be inferior to that expected on the basis of Amdahl's law [7].

Following [35], to analyse the average speedup of sequential steady-state simulation runs under the MRIP scenario, let us note that each processor runs an independent replication of the simulation process. Therefore, it first generates data characterising the initial transient period (if there is such a period) and these data are discarded. Only later, having entered the steady-state region, does a simulation processor start its contribution to the steady-state analysis by submitting its data to a global analyser. Obviously, the best speedup is achievable if one launches simulation processors on an homogeneous set of processors. With heterogeneous processors speeding up the simulation may not even be possible. This case occurs when one of the processors is fast enough to generate the required number of observations before any of the slower processors reaches the first checkpoint. Therefore, we assume that a steady-state simulation is run on a set of $P$ homogeneous simulation processors, and the length of the simulation is measured by the total number of observations submitted by $P$ simulation processors to the global analyser before the simulation is stopped. Furthermore, assuming the very fine granularity of a stochastic simulation (the small distance between checkpoints), the speedup of a steady-state simulation in the MRIP scenario would be governed by the truncated Amdahl's law.

Following [35], let us assume that a sequential steady-state simulation under the MRIP scenario is stopped when $P$ homogeneous simulation processors have delivered the total number of observations $N_{\text{min}}$ needed to satisfy the stopping criterion. As the number of processors increases, we will reach a situation in which all $P$ processors reach their first checkpoint before the global analyser stops the simulation. Let $D$ be the location of the first checkpoint (i.e., the number of observations generated when the first checkpoint is reached), and let

$$P_{\text{min}} = \min\{P : D \cdot P \geq N_{\text{min}}\}. \tag{33}$$

Adding more than $P_{\text{min}}$ processors would not increase the speedup, since it has already reached its maximum speedup of

$$S_{\text{max}} = P_{\text{min}}. \tag{34}$$

The effect of adding more than $P_{\text{min}}$ processors is that the total number of observations when the simulation is stopped is greater than $N_{\text{min}}$. Having more observations (generated by $P > P_{\text{min}}$ processors) only improves the statistical error. Therefore, the upper limit of the maximum speedup can be rewritten as

$$S_{\text{max}} = \frac{N_{\text{min}}}{D}. \tag{35}$$

Linking Eqs. (31) and (35) leads to the following truncated Amdahl's law proposed by Pawlikowski and McNickle in [35]:
\[
S_p(P) = \begin{cases} 
\frac{1}{f+(1-f)/P} & \text{for } P < P_{\text{min}} = \frac{(1-f)N_{\text{min}}}{D}, \\
\frac{1}{f+D/N_{\text{min}}} & \text{for } P \geq P_{\text{min}} = \frac{(1-f)N_{\text{min}}}{D},
\end{cases}
\]

where \( f \) is the relative length of the initial transient period, which means the simulation cannot be parallelised, \( P \) is the number of processors \((P > 1)\), \( D \) is the location of the first checkpoint, \( N_{\text{min}} \) is the total number of observations needed, and \( S_p(P) \) is the speedup achievable with \( P \) homogeneous simulation processors.

As discussed in [35], one can draw the following conclusions from these results:

- To obtain maximum speedup under the MRIP scenario, \( P_{\text{min}} \) processors or more are needed to collect the required number of observations.
- The longer the relative length of the initial transient period, the smaller the speedup. As the value of parameter \( f \) increases, the speedup falls away from the theoretical trajectory.
- The truncated Amdahl’s law is valid for average speedup.
- If the length of the initial transient period is negligible in comparison with the total simulation run-length, or the length of the initial transient period has no role in the steady-state analysis, then the speedup should be linear with the number of processors engaged.

### 7. Numerical results

#### 7.1. Influence of input \( H \) values on the run-length of sequential distributed steady-state simulation of \( SSM/MI/1/\infty \) queueing systems

The number of observations required to estimate this parameter can be obtained theoretically. The derivation of a formula for the theoretically required run-length for the \( M/M/1/\infty \) queueing system was given by Daley [6]. For more detailed discussions of simulation run-length and defining a way of measuring the relative precision, see Section 4.

To investigate the influence of self-similarity on inter-arrivals, we need self-similar processes with an exponentially marginal distribution obtained from the FGN-DW method [18,19] (\( SSM/MI/1/\infty \)). However, the theoretically required run-length to estimate mean waiting times in the \( SSM/MI/1/\infty \) queueing system may not be known. We practically compare it with the simulation run-length of the same queueing models fed by Poisson processes.

Let us consider a queueing system with a self-similar arrival process with an exponential marginal distribution in a parallel and distributed steady-state simulation. We investigated how self-similar input processes influence the required length of simulation for a given relative precision of results. The \( M/M/1/\infty \) and \( SSM/MI/1/\infty \) queueing systems with a relative precision of 10%, for a 95% CI are simulated in Akaroa-2 (Appendix A) [8].
7.1.1. When one processor is used

Table 2 shows the number of expected observations, and the empirical mean number of observations needed to achieve the required relative statistical precision, for different values of traffic intensity $\rho$, when analysing mean waiting times in an $M/M/1/\infty$ queueing system. Each mean experimental run-length of the simulations is averaged over 30 replications. Confidence intervals of relative half-widths of 10% or less, for a 0.95 CI in the method based on batch means (BM) when one processor is used, are also shown in Table 2. For $\rho \leq 0.8$, the empirical mean number of observations is slightly higher than the number of theoretically required observations; see Table 1. The range of the mean numbers of empirical observations was between 13,750 and 175,125.

Table 2
Mean numbers of observations needed in the sequential analysis of steady-state mean waiting times in an $M/M/1/\infty$ and an $SSM/M/1/\infty$ queueing systems in the method based on batch means (BM) when one processor is used

<table>
<thead>
<tr>
<th>$\rho$</th>
<th>Mean numbers of observations</th>
<th>$M/M/1/\infty$</th>
<th>$SSM/M/1/\infty$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>.5</td>
<td>.6</td>
</tr>
<tr>
<td>0.1</td>
<td>19,428</td>
<td>18,633</td>
<td>23,080</td>
</tr>
<tr>
<td>0.2</td>
<td>14,977</td>
<td>14,341</td>
<td>19,675</td>
</tr>
<tr>
<td>0.3</td>
<td>13,750</td>
<td>14,574</td>
<td>19,750</td>
</tr>
<tr>
<td>0.4</td>
<td>15,670</td>
<td>15,490</td>
<td>23,494</td>
</tr>
<tr>
<td>0.5</td>
<td>19,291</td>
<td>16,543</td>
<td>29,455</td>
</tr>
<tr>
<td>0.6</td>
<td>21,482</td>
<td>21,059</td>
<td>36,517</td>
</tr>
<tr>
<td>0.7</td>
<td>29,552</td>
<td>29,286</td>
<td>55,796</td>
</tr>
<tr>
<td>0.8</td>
<td>52,897</td>
<td>52,861</td>
<td>129,890</td>
</tr>
<tr>
<td>0.9</td>
<td>175,125</td>
<td>145,376</td>
<td>483,704</td>
</tr>
</tbody>
</table>

Stars (*) mean that the results were not obtained because too many numbers of observations required to achieve the final estimates with relative precision not larger than 10%, for a 95% CI when both $\rho$ and $H$ are high.

7.1.1. When one processor is used

Table 2 shows the number of expected observations, and the empirical mean number of observations needed to achieve the required relative statistical precision, for different values of traffic intensity $\rho$, when analysing mean waiting times in an $M/M/1/\infty$ queueing system. Each mean experimental run-length of the simulations is averaged over 30 replications. Confidence intervals of relative half-widths of 10% or less, for a 0.95 CI in the method based on batch means (BM) when one processor is used, are also shown in Table 2. For $\rho \leq 0.8$, the empirical mean number of observations is slightly higher than the number of theoretically required observations; see Table 1. The range of the mean numbers of empirical observations was between 13,750 and 175,125.

In contrast, the results in Table 2 shows that analysis of an $SSM/M/1/\infty$ queueing system with self-similar input processes requires many more observations than an $M/M/1/\infty$ queueing system. The mean number of observations, or simulation run-length, significantly increases when $H > \frac{1}{2}$. For $H = \frac{1}{2}$, the mean number of observations needed in the $SSM/M/1/\infty$ queueing system is similar to the $M/M/1/\infty$ queueing system. For $\rho = 0.4$; and $H = 0.5,0.6,0.7$ and 0.8, the empirical mean numbers of observations for the $SSM/M/1/\infty$ queueing system are 15,490 ($-1.1\%$), 23,494

---

4 So far no simulation output data analysis methods for self-similar processes have been proposed. However, we believe the BM method can be used for models with self-similar input because they are used in simulation practice to calculate interval estimators from a single (long) simulation run by weakening correlations existing between consecutive data. The sequential approaches using BM have been implemented in Akaroa-2 (Appendix A) [8]. An algorithmic description in sequential simulation with these methods, implemented in Akaroa-2, can be found in [23,34]. More research on self-similar processes is required to fully understand the queueing properties.
(49.9%), 57,905 (269.5%), and 680,238 (4241.0%) in the BM method, while the $M/M/1/\infty$ queueing system needs 15,670 observations in the BM method. The relative increases of the run-length for the $SSM/M/1/\infty$ against the $M/M/1/\infty$ queueing system. In addition, for $\rho = 0.8$; and $H = 0.5, 0.6$ and 0.7, the empirical mean numbers of observations for the $SSM/M/1/\infty$ queueing system are 52,861 (−0.1%), 129,890 (145.6%) and 874,046 (1552.4%) in the BM method, while for $\rho = 0.8$, the $M/M/1/\infty$ queueing system needs 52,897 observations in the BM method.

Therefore, the results show that the mean number of observations in the method based on BM gradually increases as $\rho$ increases. The mean numbers of observations needed in the sequential analysis of mean waiting times in the $M/M/1/\infty$ queueing system also show that the difference is insignificant theoretically and empirically when compared with an $SSM/M/1/\infty$ queueing system.

7.1.2. When two processors are used

While for $\rho \leq 0.6$, the empirical mean number of observations shown in Table 3 is slightly higher than the number of theoretically required observations, for $\rho \geq 0.7$, the empirical mean number of observations is lower than the number of observations required theoretically. The range of the mean numbers of empirical observations was between 21,847 and 145,166 in the BM method.

Table 3 shows that for $\rho = 0.4$; and $H = 0.5, 0.6, 0.7$ and 0.8, the empirical mean numbers of observations for the $SSM/M/1/\infty$ queueing system are 22,627 (−8.4%), 24,621 (−0.3%), 47,248 (91.2%) and 580,707 (2,250.5%) in the BM method, while the $M/M/1/\infty$ queueing system only needs 24,706 observations in the BM method. In addition, for $\rho = 0.8$ and $H = 0.5, 0.6$ and 0.7, the empirical mean numbers of observations for the $SSM/M/1/\infty$ queueing system are 43,006 (−2.3%), 99,433 (125.8%) and 879,571 (1897.3%) in the BM method, while for $\rho = 0.8$, the $M/M/1/\infty$ queueing system needs 44,039 observations in the BM method.
7.1.3. When four processors are used

Our results shown in Table 4 are similar to Section 7.1.2. The range of the mean numbers of empirical observations was between 21,007 and 143,538 in the BM method. Table 4 shows that for \( \rho = 0.4 \); and \( H = 0.5, 0.6, 0.7 \) and 0.8, the empirical mean numbers of observations for the \( SSM/M/1/\infty \) queueing system are 21,726 (1.9%), 23,163 (8.6%), 51,826 (143.0%), and 317,495 (1388.8%) in the BM method, while the \( M/M/1/\infty \) queueing system only needs 21,325 observations in the BM method. In addition, for \( \rho = 0.8 \); and \( H = 0.5, 0.6 \) and 0.7, the empirical mean numbers of observations for the \( SSM/M/1/\infty \) queueing system are 49,420 (1.2%), 98,188 (101.1%) and 559,623 (1046.4%) in the BM method, while for \( \rho = 0.8 \), the \( M/M/1/\infty \) queueing system only needs 48,814 observations in the BM method.

Table 4
Mean numbers of observations needed in the sequential analysis of steady-state mean waiting times in an \( M/M/1/\infty \) and an \( SSM/M/1/\infty \) queueing systems with relative precision not larger than 10%, for a 95% CI in the method based on batch means (BM) when four processors are used

<table>
<thead>
<tr>
<th>( \rho )</th>
<th>( M/M/1/\infty )</th>
<th>( SSM/M/1/\infty )</th>
</tr>
</thead>
<tbody>
<tr>
<td>.5</td>
<td>21,423</td>
<td>23,328</td>
</tr>
<tr>
<td>.6</td>
<td>21,007</td>
<td>19,987</td>
</tr>
<tr>
<td>.7</td>
<td>21,147</td>
<td>20,004</td>
</tr>
<tr>
<td>.8</td>
<td>21,325</td>
<td>21,726</td>
</tr>
<tr>
<td>.9</td>
<td>22,007</td>
<td>22,648</td>
</tr>
<tr>
<td>1</td>
<td>23,592</td>
<td>26,136</td>
</tr>
<tr>
<td>2</td>
<td>28,651</td>
<td>31,544</td>
</tr>
<tr>
<td>3</td>
<td>48,814</td>
<td>49,420</td>
</tr>
<tr>
<td>4</td>
<td>143,538</td>
<td>127,983</td>
</tr>
<tr>
<td>0.1</td>
<td>24,173</td>
<td>23,854</td>
</tr>
<tr>
<td>0.2</td>
<td>22,903</td>
<td>23,163</td>
</tr>
<tr>
<td>0.3</td>
<td>42,512</td>
<td>51,826</td>
</tr>
<tr>
<td>0.4</td>
<td>69,053</td>
<td>317,495</td>
</tr>
<tr>
<td>0.5</td>
<td>112,232</td>
<td>558,660</td>
</tr>
<tr>
<td>0.6</td>
<td>228,246</td>
<td>559,623</td>
</tr>
</tbody>
</table>

![Empirical M/M/1/∞](image)

Fig. 3. For \( \rho = 0.1–0.9 \), speedup obtained from distributed simulations of the \( M/M/1/\infty \) queueing system with relative precision not greater than 10%, for a 95% CI.
Therefore, as we discussed above, the mean number of observations of the $SSM/M/1/\infty$ queueing system, shown in Tables 2–4, rapidly increases as $H$ increases, while the mean number of observations of the $M/M/1/\infty$ queueing system, shown in Tables 2–4, slowly increases. We may also face difficulty in the sequential distributed steady-state simulation of a queueing system when the $H$ value is especially high, since methods of bulk generation of self-similar input processes require a huge amount of time.

7.2. Influence of input $H$ values on the speedup of sequential distributed steady-state simulation of $SSM/M/1/\infty$ queueing systems

The usual measurement of effectiveness of parallel computation is the speedup. The relative speedups obtained from distributed steady-state simulations of $SSM/M/1/\infty$ queueing systems with relative precision not greater than 10%, for a 95% CI.

Fig. 4. For $H = 0.5, 0.6, 0.7$ and $0.8$, speedup obtained from distributed simulations of the $SSM/M/1/\infty$ queueing systems with relative precision not greater than 10%, for a 95% CI.
\( M/1/\infty \) queueing systems using \( P = 1, 2 \) and \( 4 \) processors under the MRIP scenario are presented in Figs. 3 and 4 as the engaged processor increases one to four. These speedup are calculated from the data set of Tables 2–4 by

\[
\text{Speedup} = \frac{\text{the number of replication collected using a single processor}}{\text{the number of replication collected using } P \text{ processor}}/P,
\]

where \( P \) is the number of processors used. For more detailed discussions, see [23]. Our results show that the time for collecting many numbers of observations in distributed stochastic simulations of \( SSM/M/1/\infty \) queueing systems is clearly reduced as \( \rho \) and the value of \( H \) increase. Figs. 3 and 4(a) show that the speedups of the \( M/M/1/\infty \) and \( SSM/M/1/\infty \) queueing systems are increased about three times. For \( H = 0.6 \) and \( 0.7 \), each of the speedup was increased about four and five times, respectively. For \( H = 0.8 \), it is increased about seven times. In particular, for \( \rho = 0.5 \) and \( H = 0.8 \), it is increased over nine times.

8. Conclusions

We have examined queueing behaviour in parallel and distributed stochastic simulations of \( SSM/M/1/\infty \) with self-similar inter-event input and \( M/M/1/\infty \) queueing models. We estimated the number of observations required in the sequential distributed stochastic simulation of telecommunication networks with self-similar input. As we have shown, assuming self-similar inter-event processes (i.e., \( SSM/M/1/\infty \) queueing systems), many more observations are required to obtain the final simulation results with a required precision, as \( H \) increases, than when assuming Poisson models, exhibiting SRD (i.e., \( M/M/1/\infty \) queueing systems) when a single processor (\( P = 1 \)) and multiple processors (\( P = 2 \) and \( P = 4 \)) are used.

We found that the speedup is clearly increased as traffic intensity and the value of the Hurst parameter increase, and as the engaged processor increases one to four. In particular, for \( \rho = 0.5 \) and \( H = 0.8 \), the speedup is increased over nine times. In other words, the value of \( H \) influences much more the speedup than traffic intensity and the engaged processor. To secure a predefined statistical precision of final simulation results, one must generate arbitrary long sequences of inter-event times, and this must be done sequentially.

Acknowledgments

The authors gratefully thank Murray McMillan for his valuable comments. We also thank Professor Christian Lavault of the Universite Paris-Nord in France and two anonymous referees for their constructive remarks. This work was partially supported by the Korean Bible University’s Research Grant.
Appendix A. Automated simulation package: Akaroa-2

A fully automated simulation package, Akaroa-2 (version 2.4.1), was used as a performance evaluation tool in this dissertation. Akaroa-2 is the latest version of a fully automated simulation tool designed to run parallel/distributed stochastic simulations under the Multiple Replications In Parallel (MRIP) scenario in a local area network (LAN) environment [8].

A.1. Architecture of Akaroa-2

The Akaroa-2 system has three main components: akmaster, akslave and akrun, plus three auxiliary components: akadd, akstat and akgui; more detailed discussion can be found in Ewing et al. [8]. Fig. 5 shows the relationships between the three main components of Akaroa-2. Each bold-outlined box represents one Unix process, and the connecting lines represent Transmission Control Protocol and Internet Protocol (TCP/IP) stream connections.

The akmaster is the master process that coordinates the activity of all other processes initiated by Akaroa-2. It launches new simulations, maintains information about the state running simulations, performs global analysis of the data produced by simulation engines, and makes simulation stopping decisions.

The akslave processes run on hosts that run simulation engines. The sole function of the akslave is to launch simulation engine(s) on its host as directed by the akmaster.

Once the akmaster and any desired akslaves are running, the akrun program is used to initiate a simulation. It first contacts the akmaster process, obtaining its host name and port number from a file left by the akmaster in the user’s home directory. For each simulation engine requested, the akmaster chooses a host from among those hosts on the LAN that are running akslave processes. It instructs the akslave on that host to launch an instance of the user’s simulation program, passing on any specified arguments. The first time the simulation program calls one of the Akaroa-2 library routines, the simulation engine opens a connection to the akmaster process and identifies the simulation to which it belongs, so that the akmaster can associate the connection with the appropriate simulation data structure.

Akadd is used to add more simulations to a running simulation. It can be used to replace simulation engines that have been lost for some reason, or to speed up the simulation if more hosts become available. Akstat is used to obtain information about the state of the Akaroa-2 system: which hosts are available, which simulations are running, and what progress each simulation is making. Akgui provides a graphical user interface to start and monitor simulations that can be used instead of, or in addition to, akrun and akstat.

---

5 The Akaroa-2 package can be freely downloaded for the purpose of teaching and non-profit research activities at universities and research institutes from http://www.cosc.canterbury.ac.nz.
Following the principles of sequential simulation [34], each engine in Akaroa-2 performs a sequential analysis of its own data to form a local estimate of each performance measure. At more or less regularly determined checkpoints, the engine sends its local estimates to the akmaster process, where the local estimates of each performance measure from all the engines are combined to produce a set of global estimates. Whenever a new global estimate is calculated, the relative half-width of its confidence interval is computed, and compared with the required precision. When the precision of all analysed performance measures becomes satisfactory, the akmaster terminates all the simulation engines and sends the final global estimate to the akrun process, which in turn reports them to the user.

A.2. Transient period detection in Akaroa-2

A number of ways to estimate the length of the initial transient period of steady-state simulations have been proposed. Basic problems related to the existence of initial transient periods can be found, for example, in [34]. The length of the initial transient period has traditionally been determined using various heuristic rules. More precise measures of the length of the initial transient period could be obtained by using various statistical tests to test the stationarity of data sequences. Each operates in a hypothesis testing framework, formally testing the null hypothesis that there is no initialisation bias in the output mean against the alternate hypothesis that initialisation bias in the output exists. In Akaroa-2, a method applied for automatic detection of the length of the initial transient period was proposed by Pawlikowski [34]. In the case of steady-state simulation, a fully automated sequential statistical test for detecting the initial transient period in Akaroa-2 follows the following steps:

1. A rough, first approximation of the number of initial observations that should be discarded is obtained by applying a heuristic rule of thumb (labelled R5 in [34]).
The initial transient period is over after \( n \) observations \( x_1, x_2, \ldots, x_n \) crosses the mean \( \overline{X}(n) \) \( k \) times, \(^6\) where \( \overline{X}(n) = \frac{1}{n} \sum_{i=1}^{n} x_i \).

2. Following the first rough selection of the transaction point for the initial data, the length of the initial transient period is more precisely determined sequentially by applying the statistical tests proposed by Schruben et al. in [43] for testing the stationarity of collected observations.

3. If the sequence of tested data cannot be considered stationary, it is discarded and the next sequence of observations tested. This process is repeated until the test determines that the system is free from the effect of the initial transient period, or some predefined upper limit on the simulation length is reached.

References


\(^6\) This heuristic rule is sensitive to the value of \( k \). The selection of \( k = 25 \) was adopted in Akaroa-2 as recommended in [11].