Quantitative Computer

Simulation as a Paradigm of Scientific Investigations

Krzysztof Pawlikowski

University of Canterbury

Christchurch


NORTH ISLAND
SOUTH ISLAND
STEWART ISLAND
Kaitaia
Kerikeri
Paihia
Whangarei
Dargaville
Warkworth
Helensville
AUCKLAND
Coromandel
Thames
Paeroa
Waihi
Katikati
Tauranga
Te Puke
Hicks Bay
Whakatane
Opotiki
Hamilton
Otorohanga
Rotorua
Tokoroa
Taupo
Waikaremoana
Gisborne
New Plymouth
Turangi
Wairoa
Waiouru
Hawera
Napier
Wanganui
Palmerston North
Woodville
Paraparaumu
Masterton
WELLINGTON
Collingwood
Motueka
Picton
Nelson
Blenheim
Westport
Reefton
Punakaiki
Springs Junction
Kaikoura
Greymouth
Lewis Pass
Hanmer Springs
Hokitika
Arthurs Pass
Waipara
Franz Josef
CHRISTCHURCH
Mount Cook
Haast
Ashburton
Akaroa
Timaru
Omarama
Milford Sound
Wanaka
Kurow
Queenstown
Cromwell
Oamaru
Alexandra
Te Anau
Manapouri
DUNEDIN
Gore
Milton
Invercargill
Half Moon Bay

Earthquake in Christchurch: 12.50 pm, 22 February 2011

University of Canterbury: Interdepartmental Network Research Laboratory:

Protocols, Distributed Processing and Simulation

(1) Simulation Research Group:
developing new methodologies of distributed and automated stochastic discrete-event simulation

(2) Network Research Group:
modelling and evaluating performance of the Internet pand other multimedia networks in wired and wireless technologies

Membership:

- Prof. K. Pawlikowski (CSSE), with
- Associate Prof. D. McNickle (MGMT)
- Dr A. Willig (CSSE)
- Dr A. McINnes (ECE)
- two postdoc research fellows
- Masters and PhD students

Currently: 5 PhD and 2 Masters students

see www.cosc.canterbury.ac.nz/research/RG/net_sim/

Computer simulation:

one of the most important factors enabling new developments in science & technology

Congress of the USA, 1996

Three paradigms of science:

- theory
- experimentation
- computer simulation

Ken Wilson, Nobel Prize in physics in 1982

SIMULATION as a discipline of Computing:

(i) computability = **simulatability**

(ii) **object-oriented simulation:**
an origin of object-oriented programming

(iii) **parallel simulation:**
an origin of parallel programming & processing

... we are a part of Universe simulated by a computer (a cellular automaton) ...

Prof. Edward Fredkin
Carnegie-Mellon University
Director of Computer Science Lab., MIT, 1971-74

More on simulated Universe:

e.g. Paul Davies, “The Mind of God”

Professor of Arizona State University, Center for Fundamental Concepts in Science.

Previously: Professor of Mathematical Physics, Imperial College of London and University of Adelaide, Australia

Quantitative stochastic discrete-event simulation

is used in performance evaluation studies of dynamic stochastic systems and processes

In particular, studies of **Telecommunication Networks**, motivated by:

- Enormous dependence of our civilisation on computers and their networks makes their performance evaluation mandatory before they are deployed.
- Modern multi-media networks have become so complex that their performance can be usually studied only by simulation.

Network Modelling & Simulation *among the most important 23 areas of Information Technology supported by Advanced Research Project Agency of US Department of Defense in 1995-2005*

Challenge:
to find solutions
(protocols, architectures, topologies)
which can ensure the best (or the worst) performance of networks

For example: Future Internet initiative

Issue of credibility of simulation studies:

- When can results obtained by quantitative stochastic simulation be regarded as **accurate** ?
- How to ensure **credibility** of quantitative analysis based on stochastic simulation ?

General guidelines for conducting valid performance evaluation studies based on stochastic simulation:

- use a valid simulation model:
  - correct functional representation of simulated mechanisms
  - appropriate stochastic characteristics of the model
- **execute valid simulation experiment:**
  - **use appropriate pseudo-random number generators (PRNGs)**
  - **apply appropriate method of analysis of simulation output data**

SOURCE(S) OF PRIMARY RANDOMNESS:

A good PRNG (of uniformly and independently distributed numbers):

- should generate numbers that satisfy the most rigorous statistical tests (of uniformity and independency),
- should be able to generate multiple independent streams of numbers,
- should be fast and accurate.

Until recently, the most popular PRNGs were
linear congruential generators
with cycle $L \leq 2^{48} - 1$
(with $L \leq 2^{31} - 1$ being the most popular)

How large cycles should PRNGs have
today (and in forseeable future)
to avoid repetition of randomness ?

Cycle length vs. computing technology

Proposition:

The duration of a typical simulation (measured by CPU time) does not become shorter as computing technology becomes more powerful: **users use faster computers for simulating more complicated processes within the available time**.

Conclusion 1:

PRNGs with adequately long cycles are needed to avoid repetitions of Pseudo-Random Numbers (PRNs) during a typical simulation.

Technology & generation rate of PRNs

Moore's law by Gordon Moore (1965)
see *research.microsoft.com/rgray/Moore_Law.html*

Law of accelerating returns

by Ray Kurzweil (2001)
see *www.kurzweilai.net/articles/art0134.html*

Singularity will be reached in 2034 (?)

On average, computing power of computers doubles each 1.5 - 2 years

At University of Canterbury in Christchurch, New Zealand:

In 2000, PCs with the CPU clock at 800 MHz

using rand (from C library): 10^6 PRNs in about 0.4 seconds:
2^{31} PRNs in about 13.5 minutes
2^{48} PRNs in about 3.5 years

In 2002, PCs with the CPU clock at 2.4 GHz

using rand (from C library): 10^6 PRNs in 0.13 seconds:

2^{31} PRNs in about 4.5 minutes

2^{48} PRNs in less than 1.1 years

In 2002, The Christchurch Press informed:

THE PRESS, Christchurch Tuesday, January 8, 2002 **13**

Pentium 4 fastest yet

Intel is set to unveil its fastest-ever microprocessor, a Pentium 4 chip that not only allows it to maintain its leadership over rival Advanced Micro Devices, but also introduces what analysts say is a critical improvement in Intel's memory chip technology.

The Pentium 4, running at 2.2 gigahertz, or 2.2 billion cycles per second, is the world's fastest in terms of clock speed.

PC makers, including Dell Computer and Compaq Computer, were to have PCs using the chip available today, Intel executives said in an advance briefing on the product.

Intel said its 845 chipset, which connects the microprocessor to the rest of the PC, now supports DDR, or double-data rate, memory.

Intel had been criticised for being late with DDR. Its principal rival in the processor business, AMD, has had DDR DRAM available since last year. "They've got to enable the DDR, which they're now doing, and I'd argue that's almost more important than this latest Pentium 4," Needham & Co analyst Dan Scovel said.

"Putting SDRAM on a Pentium 4 is like putting training wheels on a Porsche."

The latest microprocessor is the first Pentium 4 chip to use 0.13 micron fabrication process using copper interconnects, rather than aluminum.

Copper conducts electricity better than aluminum, boosting efficiency and performance.

Previous Pentium 4 chips have been made with a 0.18 micron manufacturing process.

The latest Pentium 4 processor will be available with and without the 845 chipset, Intel said.

—Reuters

At Intel's Develper Forum of 2002, Pat Gelsinger (Intel's chief technology officer) stated:

"We're on track, by 2010, for 30-gigahertz devices, 10 nanometers or less, delivering a tera-instruction of performance."

However, **it has not happened yet**.
As R.X. Cringley wrote
(in “Parallel Universe”, MIT Technology Review, Jan./Feb. 2009)

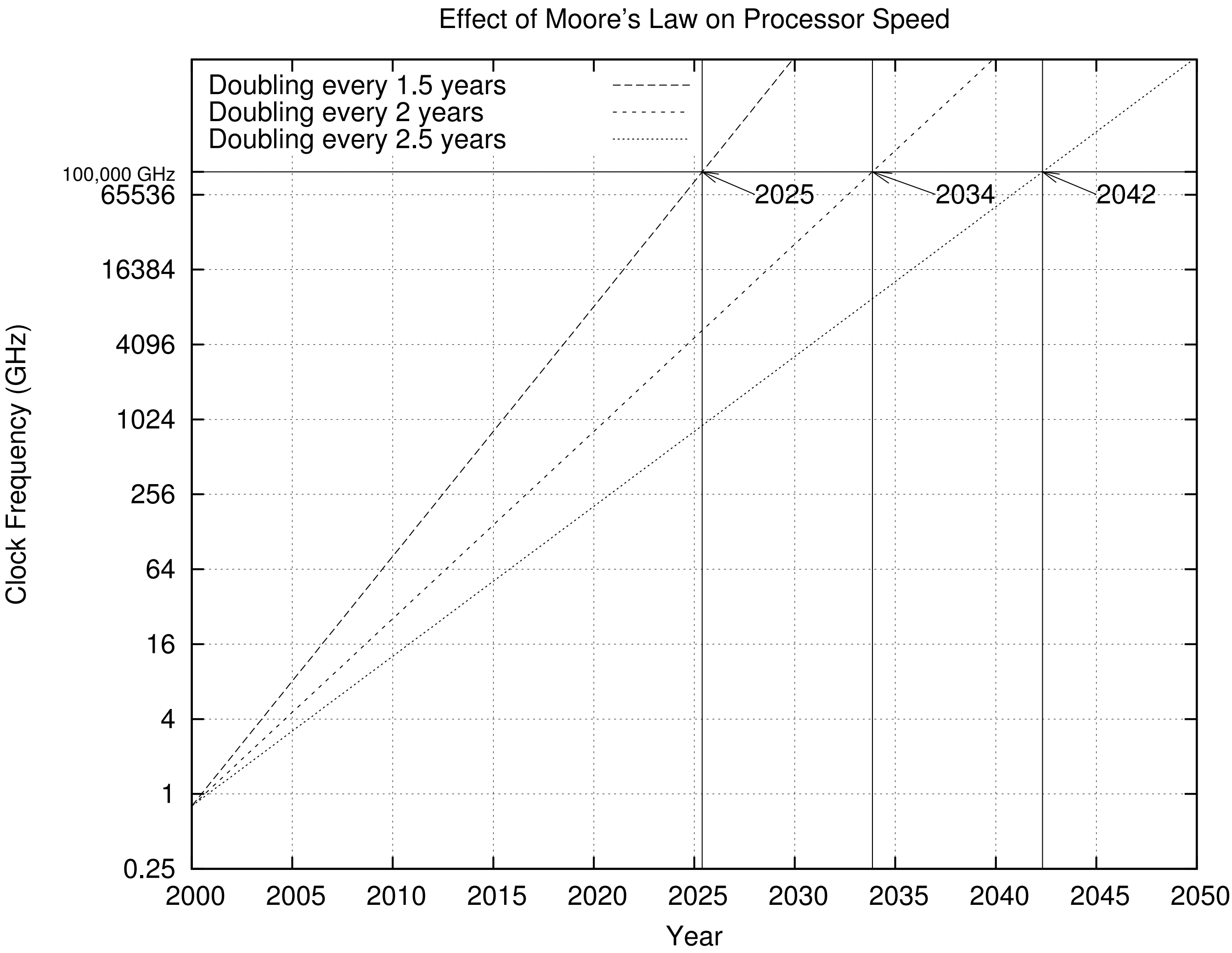
“Intel is still making processors that top out at less than 4 GHz, and 5 GHz has come to be seen as the maximum feasible speed for silicon technology. **Unexpected problems with heat generation and power consumption** *have put a (temporary) practical limit on processors’ clock speeds. New technologies, such as spintronics and quantum (or tunneling) transistors, may ultimately allow computers to run many times faster than they do now, while using much less power. But those technologies are at least a decade away from reaching the market.”*

So, in order to make the most of the technologies at hand, chip makers are increasing the number of processors inside a chip, in **“multicore” computers**.

In 2009, a 2-core PC (with Intel C2D at 2.66GHz)
using rand (from C library) generated 10^6 PRNs in 0.058 sec
or 2^{31} PRNs in about 2.08 minutes
or 2^{48} PRNs in less than 189 days

Let us assume the most critical scenario: that Moore Law will remain valid in future, regardless of technology.

How large cycles should PRNGs have then ?



If clock frequency continued to double each **1.5** years (or **2**, or **2.5** years), then we will have CPU running at over 100 THz in **2025** (or in **2034**, or in **2042**).

Cycle length of PRNGs vs. theory

(A) Large primes are needed for designing PRNGs with long cycles

1. Currently the largest prime number known is:

$$2^{43,112,609} - 1$$

discovered on 23 August 2008, by Edson Smith and his colleagues from the UCLA Mathematics Department.

This prime number (one of Mersenne primes) consists of **12,978,189 digits**. *The number qualified for Electronic Frontier Foundation's $100,000 award for discovery of the first 10 million digit prime number. The prime was verified by Tom Duell (Burlington, MA, USA) and Rob Giltrap (Wellington, New Zealand), both of Sun Microsystems, using the Mlucas program by Ernst Mayer of Cupertino California USA. The verifications on 8 dual-core SPARC64 and 4 quad-core SPARC64 took 13 days.*

See `www.mersenne.org/prime.htm` for details

2. In August 2002, Agrawal, Saxena and Kayal proved that testing numbers for primality is a problem of P complexity

See `www.cse.iitk.ac.in/news/primality.html` for details

(B) PRNs should pass tests of uniformity and statistical independence

Birthday Spacing Problem:
one of reference problems/tests for testing uniformity of PRNGs
see D.E.Knuth. Art of Computer Programming, vol.2

Recent finding:

Any linear congruential PRNG fails Birthday Spacing Test if one applies this test (in two or more dimensions) to $n \geq 16\sqrt[3]{L}$ numbers generated by that PRNG, where L is the length of its cycle.

L'Ecuyer and Simard;
in Mathematics and Computers in Simulation, 2001.

Conclusion 2:

During a simulation,
a PRNG should be used as a source of **no more than** $16\sqrt[3]{L}$ **numbers** (linear congruential PRNG), or
no more than L numbers (non-linear congruential PRNG).

Assumption:
the process of PRN generation takes no longer than 1% of simulation time.

Using popular linear congruential PRNGs on computers with 2.4 GHz CPU clock

Cycle length	n_{max}	**Max. total time of "stochastically safe" application of a good PRNG**
$L = 2^{31}$	20,594	less than 0.3 seconds
$L = 2^{48}$	1,048,576	less than 14 seconds

where $n_{max} = 16\sqrt[3]{L}$

Search for good PRNGs of the 21^{st} century

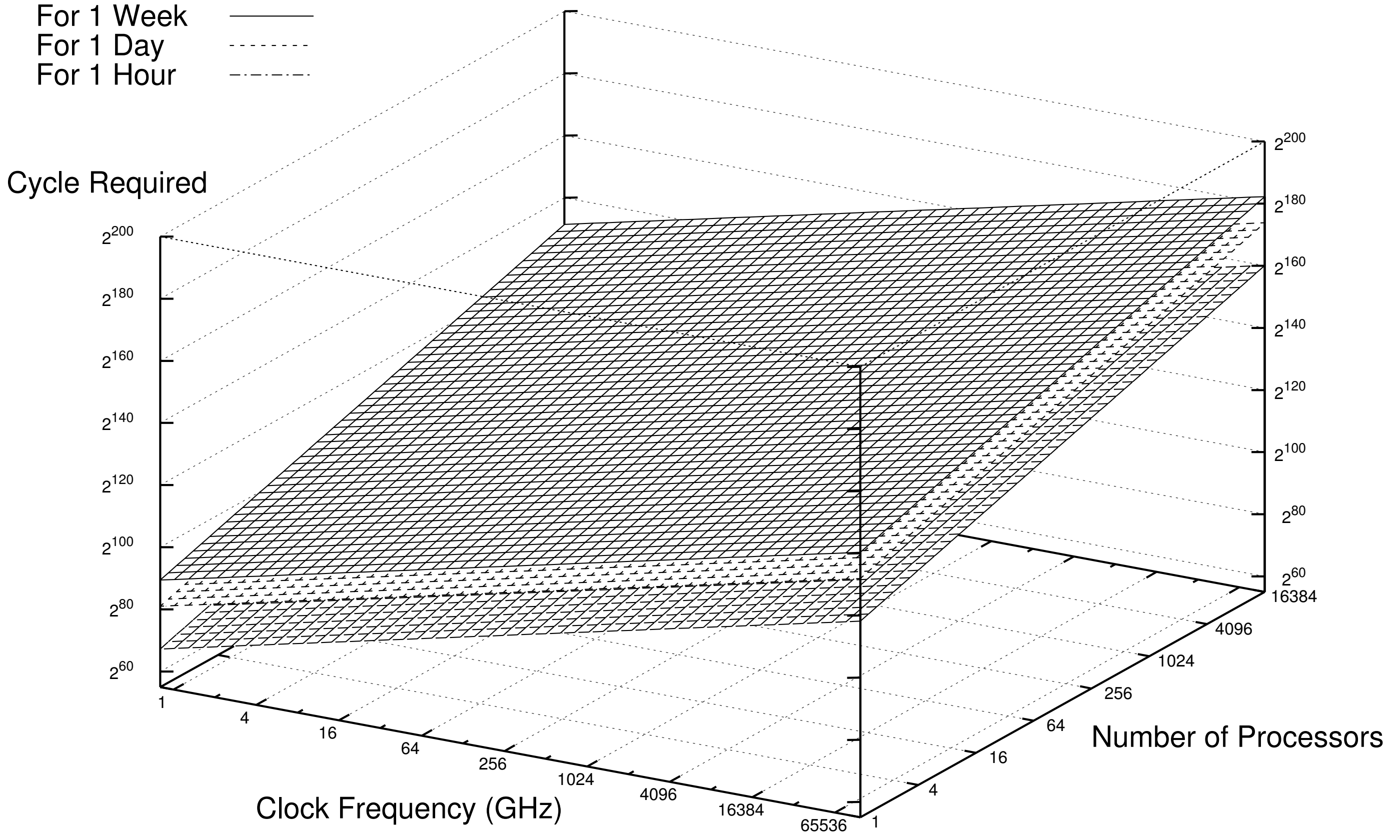
How long cycles of PRNGs are needed ?

Assumptions:

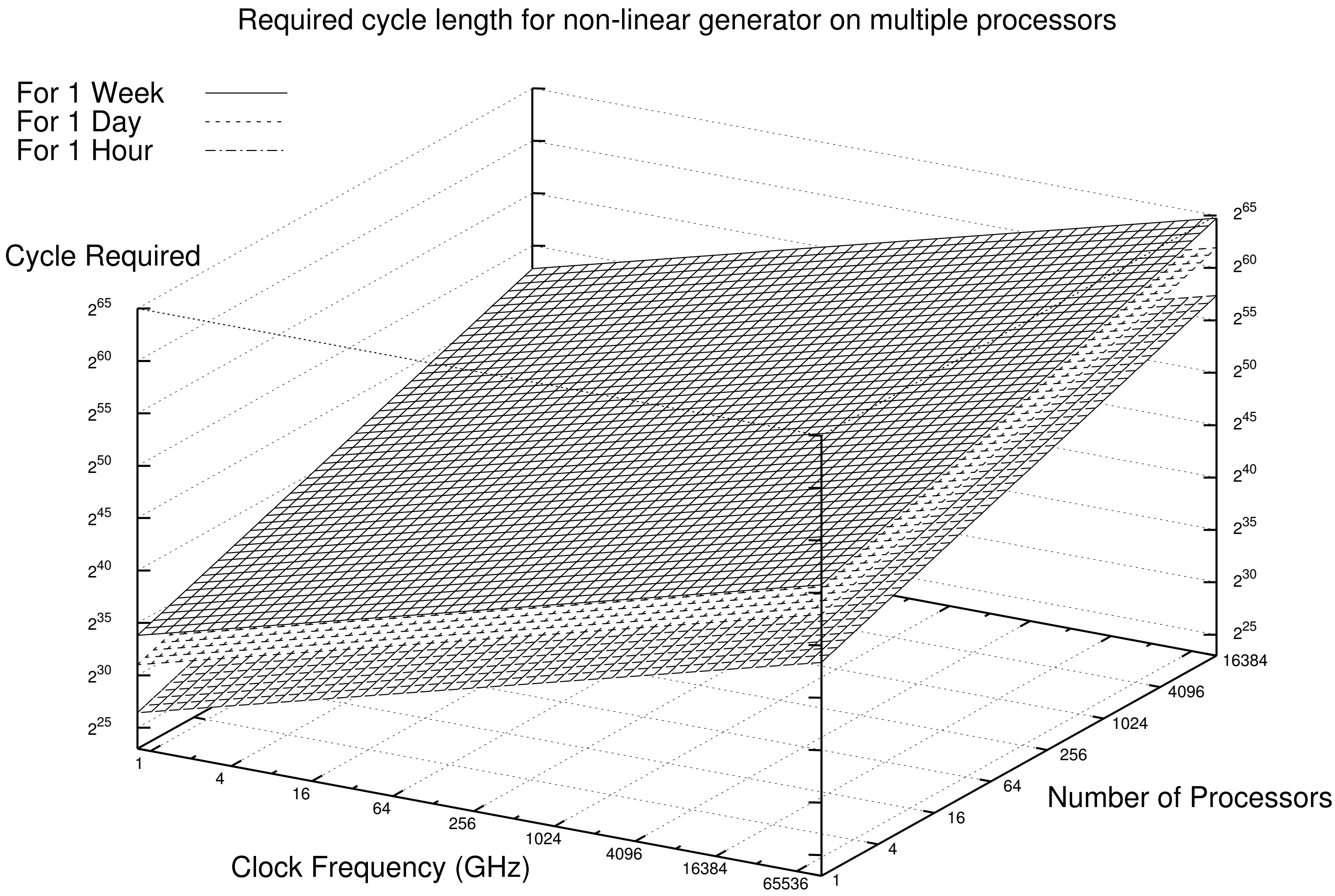
- **the process of PRN generation takes no longer than 1% of simulation time.**
- **computing power of computers continues its exponential growth, manisfesting itself both in the growth of the CPU clock frequency and in the number of cores**

K. Pawlikowski, M. Schoo & D. McNickle. "Modern Generators of Multiple Streams of Pseudo-Random Numbers". European Modeling and Simulation Symp. (EMSS'06), Barcelona, Spain, Oct. 2006, 553-59

Required cycle length for linear generator on multiple processors



Linear congruential PRNG on 2^{14} processors, with 100 THz CPUs, should have cycle of 2^{160} long if it operates for 1 hour, or 2^{182} for 1 week.



Distributed simulation on 2^{14} processors, with 100 THz CPUs, needs a nonlinear congruential PRNG with the cycle length of about 2^{57} if it lasts 1 hour, and the cycle length of about 2^{65} if it lasts 1.

Conclusion:

"Statistically safe" PRNGs
of the 21^{st} century should be able to generate numbers in cycles of length
$L > 2^{180}$ (linear PRNGs), or
$L > 2^{65}$ (non-linear PRNGs).

On the basis of the results of **exhaustive statistical testing** of various PRNGs, one can recommend a number of such "statistically safe" PRNGs.

See L'Ecuyer and Simard, *"TestU01: a C Library for Empirical Testing of Random Number Generators". ACM Trans. Mathematical Software, Aug 2007*

PRNGs of the 21^{st} century:

1. Combined Multiple Recursive Congruential PRNG (MRG32k3a):

$$r_{1,i} = 1403580 r_{1,i-2} - 810728 r_{1,i-3})$$
$$\text{mod } (2^{32} - 209)$$
$$r_{2,i} = (527612 r_{2,i-1} - 1370589 r_{2,i-3})$$
$$\text{mod } (2^{32} - 22853)$$
$$r_i = (r_{1,i} - r_{2,i}) \text{ mod}(2^{32} - 209)$$

$$L \approx 2^{191}$$

10^8 PRNs generated in 10 seconds,
on Intel Pentium processor with 2.8GHz clock

Multiple streams of PRNs generated by **Cycle Splitting**.

Proven good uniformity in up to 45 dimensions; see: *P. L'Ecuyer. Operations Research, 1996*.

2. Generalised Feedback Shift Register (GFSR) PRNGs:

WELL PRNGs

(Well Equidistributed Long-period Linear PRNGs)

Modifications of Mersenne Twister,
with quick escape from a ”statistically poor region”
(about 1000 times faster than Mersenne Twister).
Their cycles are between 2^{512} to 2^{44497}.

Generation of 10^9 PRNs varies
between 35.8s to 38.8s (cf. 30.9s for Mersene Twister),
on 2.8 GHz Intel Pentium 4.

See: Panneton & L'Ecuyer. ”Improved Long-Period Generators Based on Linear Recurrences Modulo 2”. *ACM Trans. on Mathematical Software, 2006.*

Mersenne Twister (MT19937) with the cycle $L = 2^{19937} - 1$

M. Matsumoto and T. Nishimura,
in *ACM Trans. Modelling and Computer Simulation, 1998*

Some properties of MT19937:

- uniformity in up to 623 dimensions;
- multiple streams of PRNs generated by **dynamic creation of multiple Mersenne Twisters**

Problem with Mersenne Twister:
if generated sequence enters a "statistically poor region", it can stay in that region for a long time, generating a long sequence of strongly correlated numbers (long sequences of potentially very similar or even identical numbers)

4. DX Multiple Recursive Generators

with **cycle lengths**

from about $2^{45,000}$ **to** $2^{280,000}$.

L.-Y. Deng and H. Xu. “A System of High-Dimensional, Efficient,Long-Cycle and Portable Uniform Random Number Generators”.
ACM Transactions on Modeling and Computer Simulation.
October 2003

L.-Y. Deng. “Efficient and Portable Multiple Recursive Generators of Large Order”.
ACM Trans. on Modeling and Computer Simulation. Jan 2005

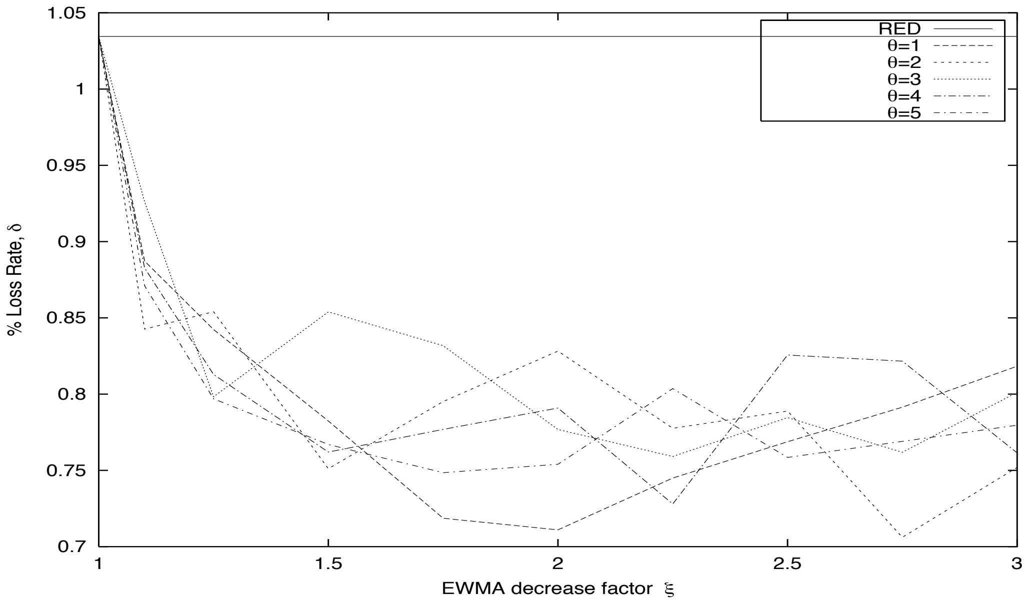
SIMULATION OUTPUT DATA ANALYSIS:

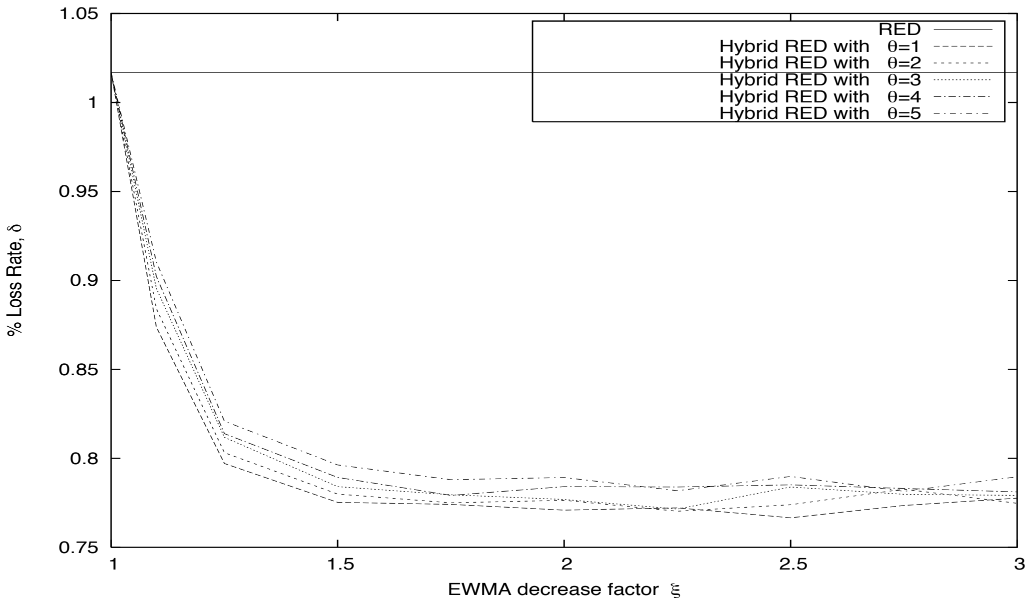
stochastic simulation	$\equiv$	simulated statistical experiment

Results of any quantitative simulation study can be credible only if they are obtained with acceptable (i.e. sufficiently low) statistical errors.

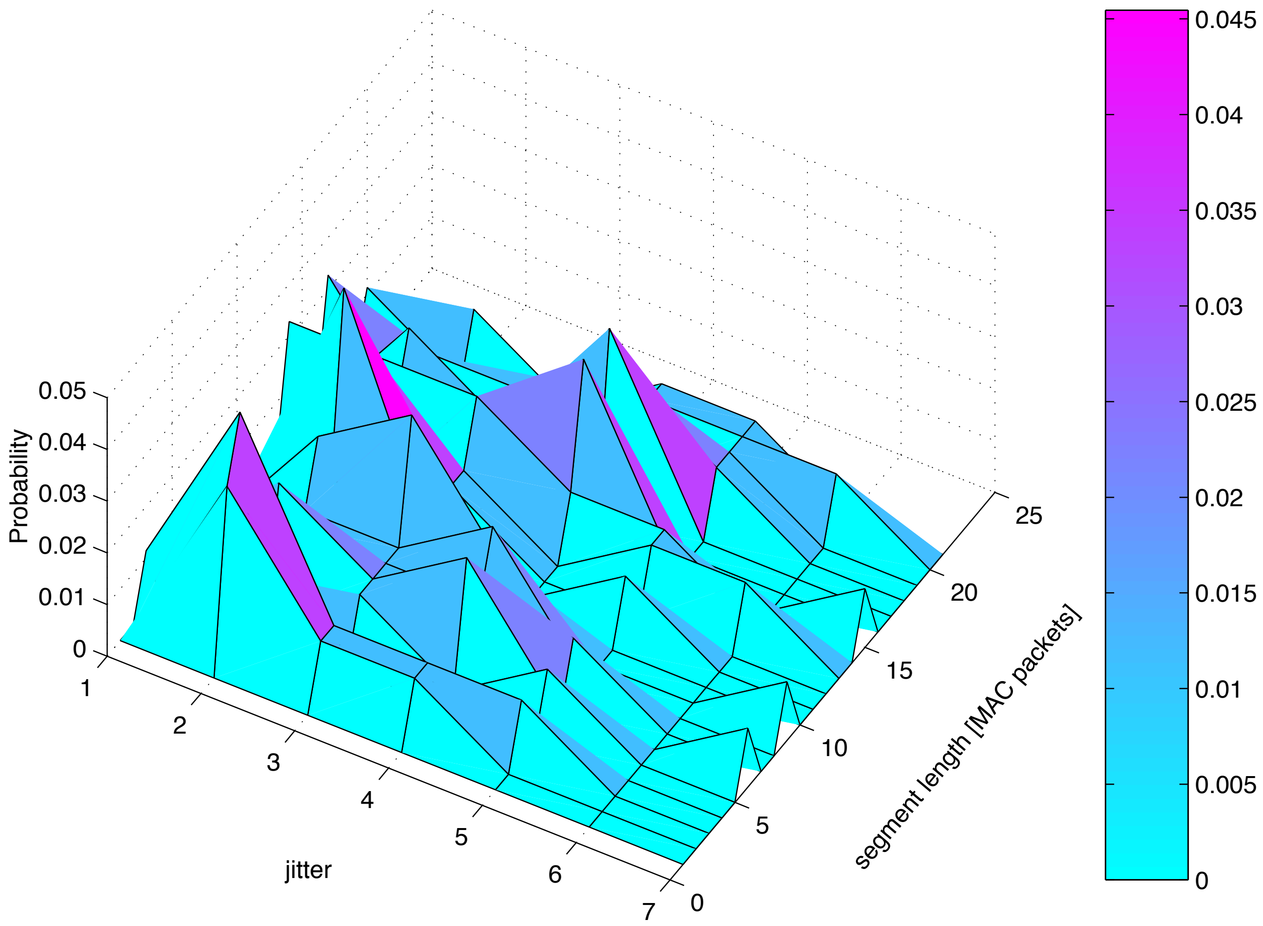
Otherwise ... the results can be meaningless !

Which of these two sets of results are more conclusive/credible ?

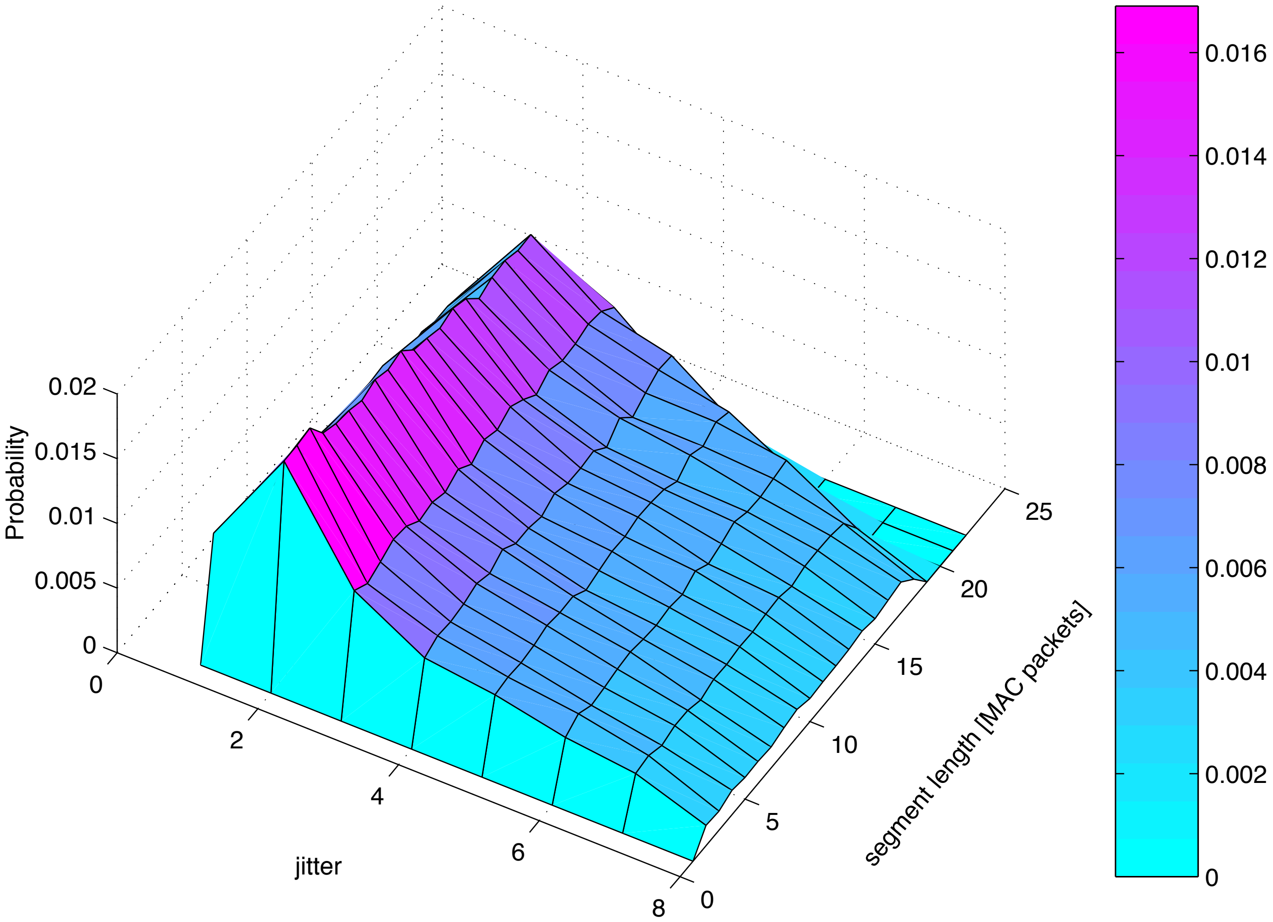




Example
Analysis of TCP: average results from one simulation vs average results over 100 replications



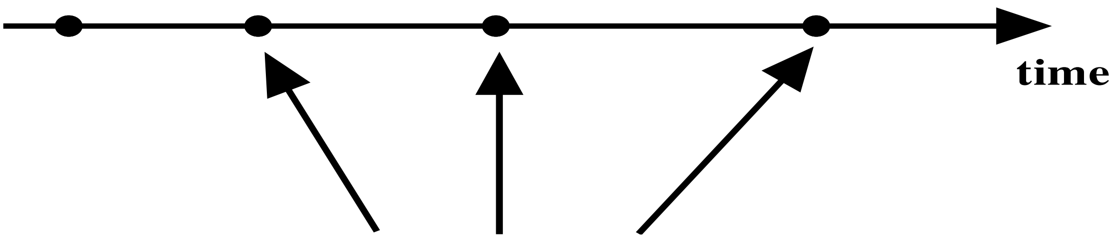
Evaluation of a Medium Access Control protocol of a mobile communication network. Results with **relative statistical errors not greater than 25%**, at 0.95 confidence level.



Evaluation of a Medium Access Control protocol of a mobile communication network. Results with **relative statistical errors not greater than1%**, at 0.95 confidence level.

The only practical solution for producing credible results from simulation:

sequential on-line data analysis of statistical errors at consecutive checkpoints of simulation



checkpoints
of output data analysis

the length of simulation is determined during simulation: the simulation is continued until statistical errors of results become satisfactorily small

Are simulation results reported in scientific literature appropriately analysed ?

Two surveys:

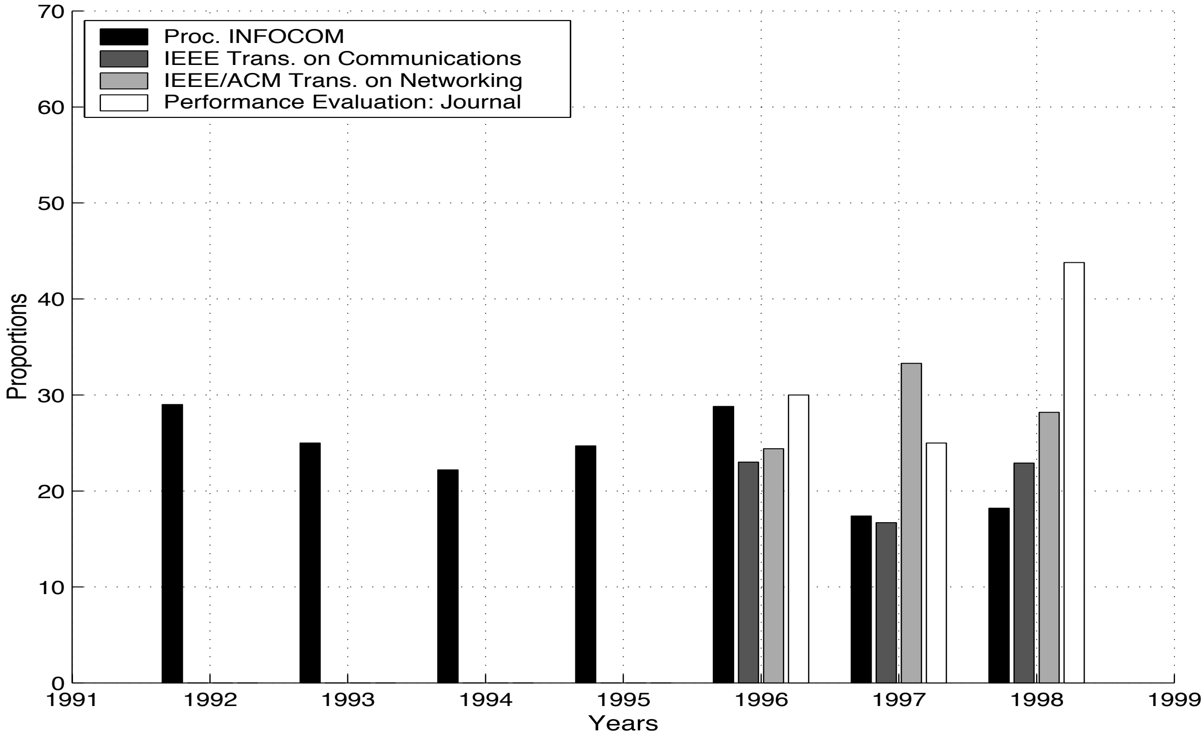
Survey A of 2246 publications on networks in Proc. IEEE INFOCOM, IEEE Trans. Comms., ACM/IEEE Trans. Networking, and Performance Evaluation J., published between 1992-1998,
see *[Pawlikowski et al., IEEE Comms., Jan. 2002]*

and

Survey B of 1772 publications on networks in Proc. IEEE INFOCOM, ACM/IEEE Trans. Networking, ACM SIGCOMM, and Performance Evaluation J., published between 2008-2010,
see *[Pawlikowski et al., 2011, submitted for publication]*
show that

over 51% of papers in 1992-1998 (Survey A) and over 58% of papers in 2008-2010 (Survey B) reported simulation results

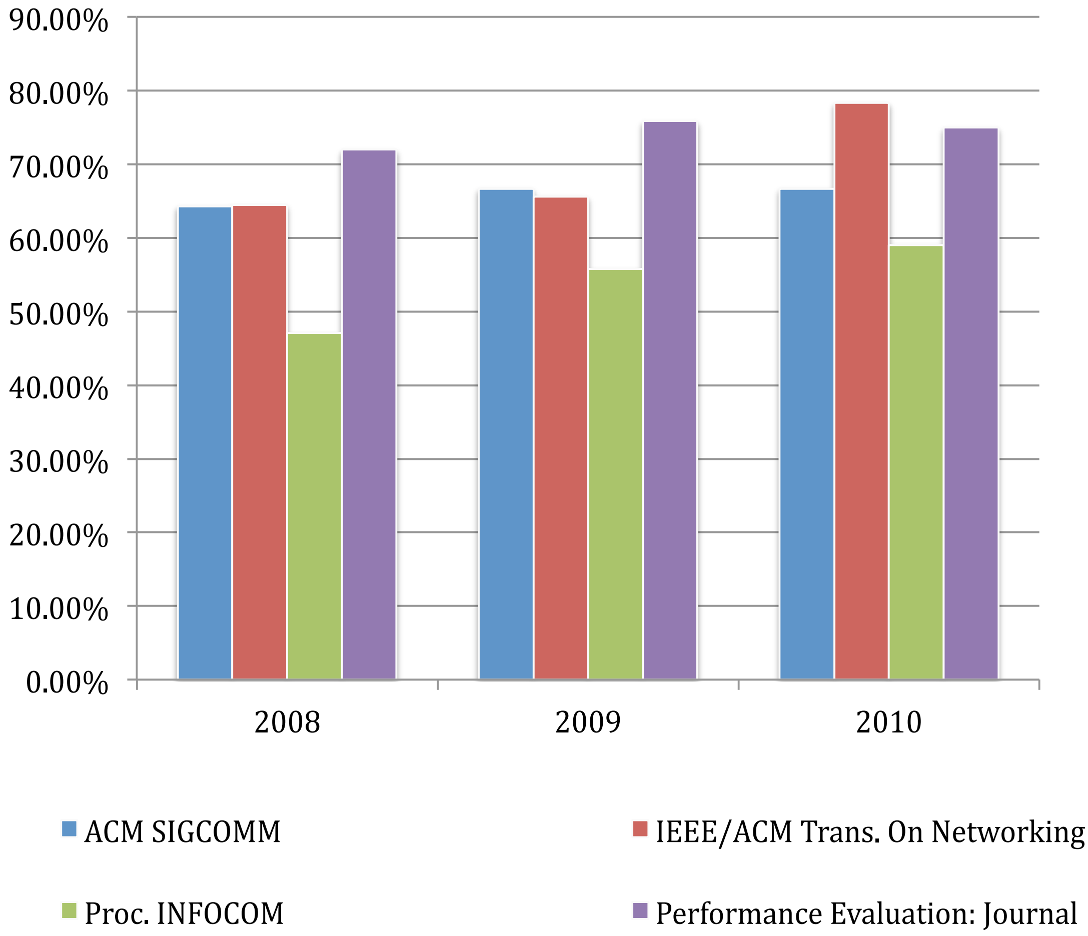
Proportion of papers reporting statistically analysed simulation results (1992-98, Survey A)



77% of papers that reported results based on simulation gave no evidence at all that simulation output data were statistically analysed.

Pawlikowski et al., IEEE Comms., Jan. 2002.

Proportion of papers reporting statistically analysed simulation results (2008-2010, Survey B)



about 66% of papers that reported results based on simulation gave evidence that simulation output data were statistically analysed.

Pawlikowski et al., 2011, submitted for publication

Unfortunately, the preliminary results of Survey B show too that, **in a significant number of papers (in 90% of papers of 2010), even if they stated that the results were statistically analysed, there was no information given what type of simulation was conducted (steady-state or terminating ?),**
nor which PRNG was used.

Both surveys indicate that the majority of papers either

- **did not give any details on the conducted simulation, or/and**
- **did not inform on how simulation data were analysed**

While all such papers reported non-repeatable simulation expriments, some of them reported purely random results only !

Another survey:
in all 151 papers published in Proc. the ACM Int. Symposium on Mobile Ad Hoc Networking and Computing (2000-2005):

- **75.5% of the papers reported results from simulation**
- **12% of papers reported results which were based on sound statistical analysis of simulation output data**
- **less than 15% of the papers reported repeatable studies of networks**

See: S. Kurowski, T. Camp and M. Colagrosso. "MANET Simulation Studies: The Incredibles".

Simulation studies of telecommunication networks:

- PSEUDO-SCIENCE ?
  (results based on personal beliefs of their authors)

- SCIENCE ?
  (results obtained and reported according to the principles of **the scientific method**)

... any new scientific hypothesis should be at least independently testable ...

Karl Popper (1901-1994), philosopher of science

Guidelines for a credible publication based on simulation results:

- make sure that your simulation program represents valid simulation model

- use well tested PRNG with sufficiently long cycle

- describe your statistical analysis of simulation output data in the paper, and/or refer to a technical report with the detailed documentation of your simulation

Resolution to the credibility crisis of quantitative simulation:

- by introduction of professional certification of simulation specialists ?
- by developing tools for automated sequential simulation ?

See *www.SimProfessional.org* for the current status of **Professional Certification in Modelling and Simulation**, administered by the Modeling and Simulation Professional Certification Commission of the Society for Computer Modelling and Simulation Int.

Problems of sequential quantitative simulation of telecommunication networks:

- **limitations of statistics:**

  limited spectrum of methods for sequential analysis of simulation output data; in particular, if data are strongly correlated

- **technological limitations:**

  prohibitively long simulation time can be needed for securing sufficiently low statistical errors

For example, estimation of mean steady-state delay in $M/M/1/\infty$ buffer, with max. error of 5%, at 0.95 confidence level, needs:

(a) **over 61 000 000** observations (ρ = 0.99),

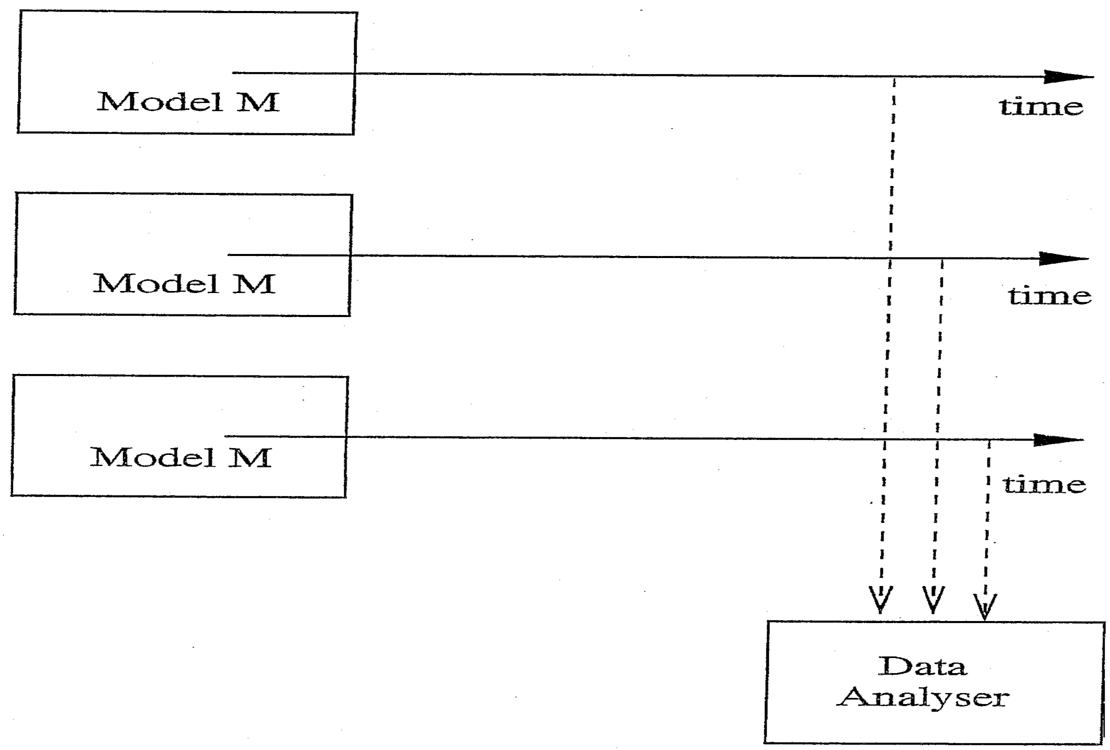
(b) **over 6 143 000 000** observations (ρ = 0.999)

Possible solution:

- **application of distributed processing** (distributed generation and collection of data for analysis, for speeding up simulation)
- **automated sequential output data analysis** (during simulation, for ensuring accurate results)

Multiple Replications in Parallel (MRIP):

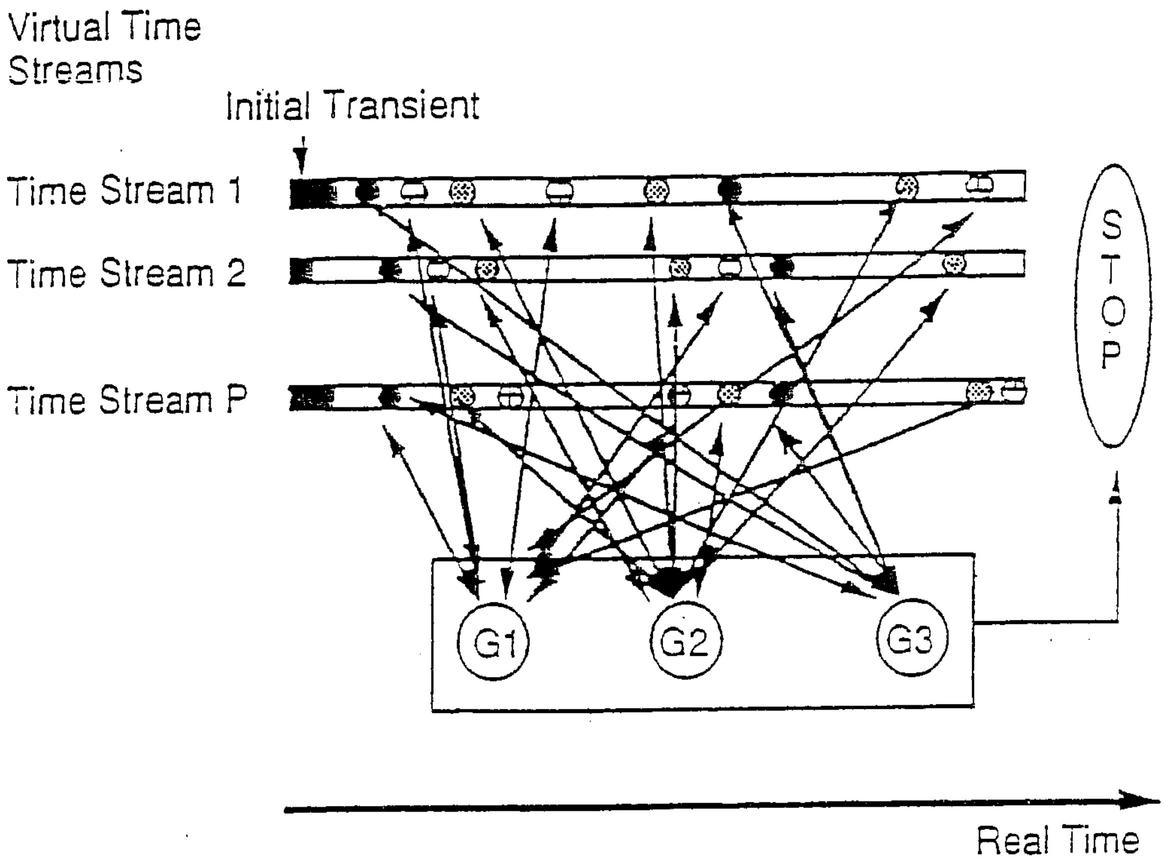
Multiple simulation engines independently produce statistical data by running their own simulation replications, but cooperate together in data analysis.



Main properties:

- **model-independent parallelisation**
- **no parallel programming needed**
- **possible automation of parallelisation**

MRIP as a (distributed) simulation in multiple time streams:



(⊖ = local checkpoint parameter 1,
⊛ = local checkpoint parameter 2,
● = local checkpoint parameter 3,
Gi = Global Estimation of parameter i, i=1,2,3.)

MRIP: estimating mean value of dependent observations

Assumptions:

- P simulation engines are used
- N_i observations have been generated by simulation engine i;
- $N = N_1 + N_2 + \cdots + N_P$ observations have been generated in total.

Global Analyser calculates:

$$\begin{aligned}
\overline{\overline{X}} &= \frac{N_1}{N}\overline{X}_1 + \frac{N_2}{N}\overline{X}_2 + \cdots + \frac{N_P}{N}\overline{X}_P \\
\hat{Var}(\overline{\overline{X}}) &= (\frac{N_1}{N})^2 \hat{Var}(\overline{X}_1) + \cdots + (\frac{N_P}{N})^2 \hat{Var}(\overline{X}_P),
\end{aligned}$$

where $\overline{X}_i$ = local estimate of EX by simulation engine i,

$i = 1, 2, \ldots, P$, and

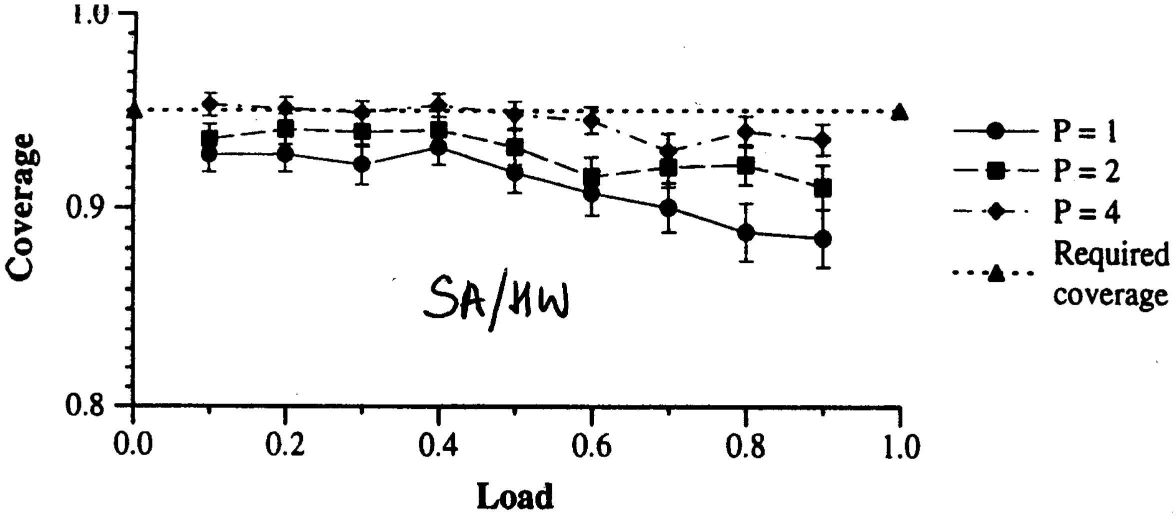
$\hat{Var}(\overline{X}_i)$ as local SA/HW estimator of variance of $\overline{X}_i$,

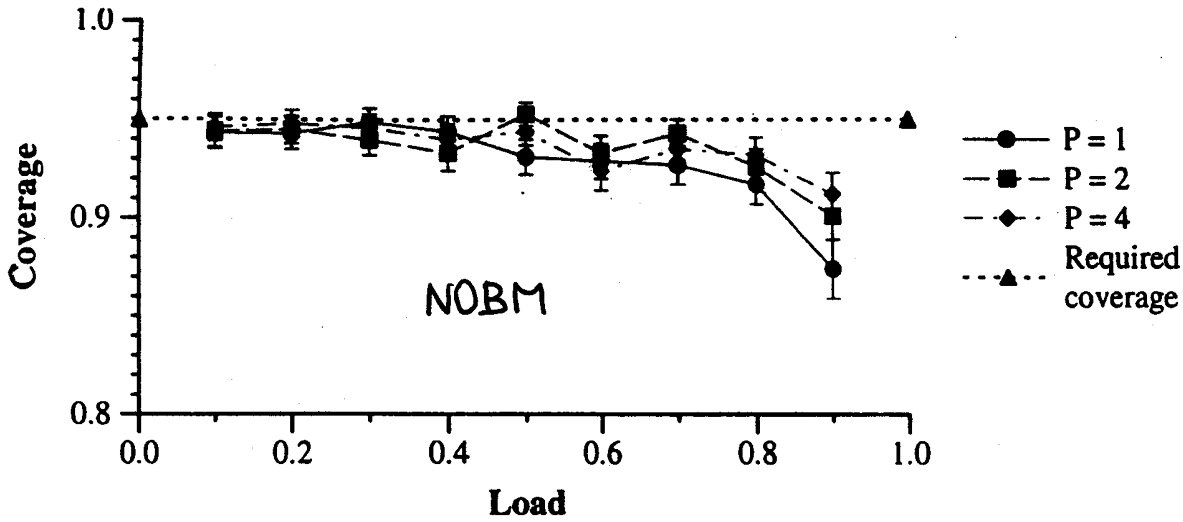
with t-distributed local error with d degrees of freedom

$\frac{\overline{\overline{X}} - EX}{\hat{Var}(\overline{\overline{X}})}$ **= approximately t-distributed, with df =** $d \cdot P$ **degrees of freedom (**d **= 7 or 16)**

Properties of MRIP estimators of errors of means: coverage analysis

Spectral Analysis (SA/HW) vs Non-overlapping Batch Means (the case of M/M/1/∞)





- Coverage improves with P and decreases with traffic intensity
- SA/HW offers better performance than NOBM as P increases

MRIP: Estimating variance of dependent observations

Variance as the mean of $Y = (X - EX)^2$,
by using secondary data $y_i = (x_i - \overline{x}_i)^2$

Assumptions:

- P simulation engines are used
- N_i observations obtained from simulation engine i;

Global Analyser calculates:

$\hat{Var}(X) = \frac{\sum_i N_i \hat{\sigma}_i^2}{\sum_i N_i}$ and

$\hat{Var}(\hat{Var}(X)) = \frac{\sum_i N_i^2 \hat{Var}(\hat{\sigma}_i^2)}{(\sum_i N_i^2)^2}$

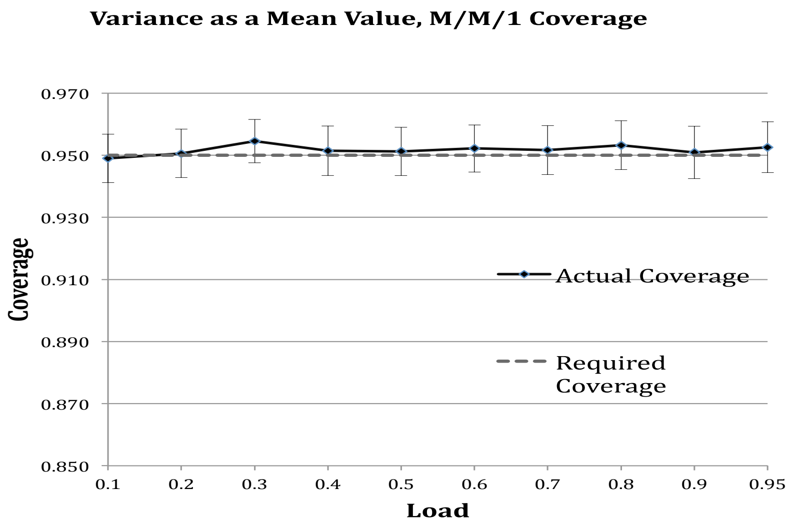
where $\hat{\sigma}_i^2$ = local estimate of of Var[X] by simulation engine i, $i = 1, 2, \ldots, P$, and

$\hat{Var}(\hat{\sigma}_i^2)$ as local SA/HW estimator of variance of $\hat{\sigma}_i^2$,

with t-distributed local error with d degrees of freedom

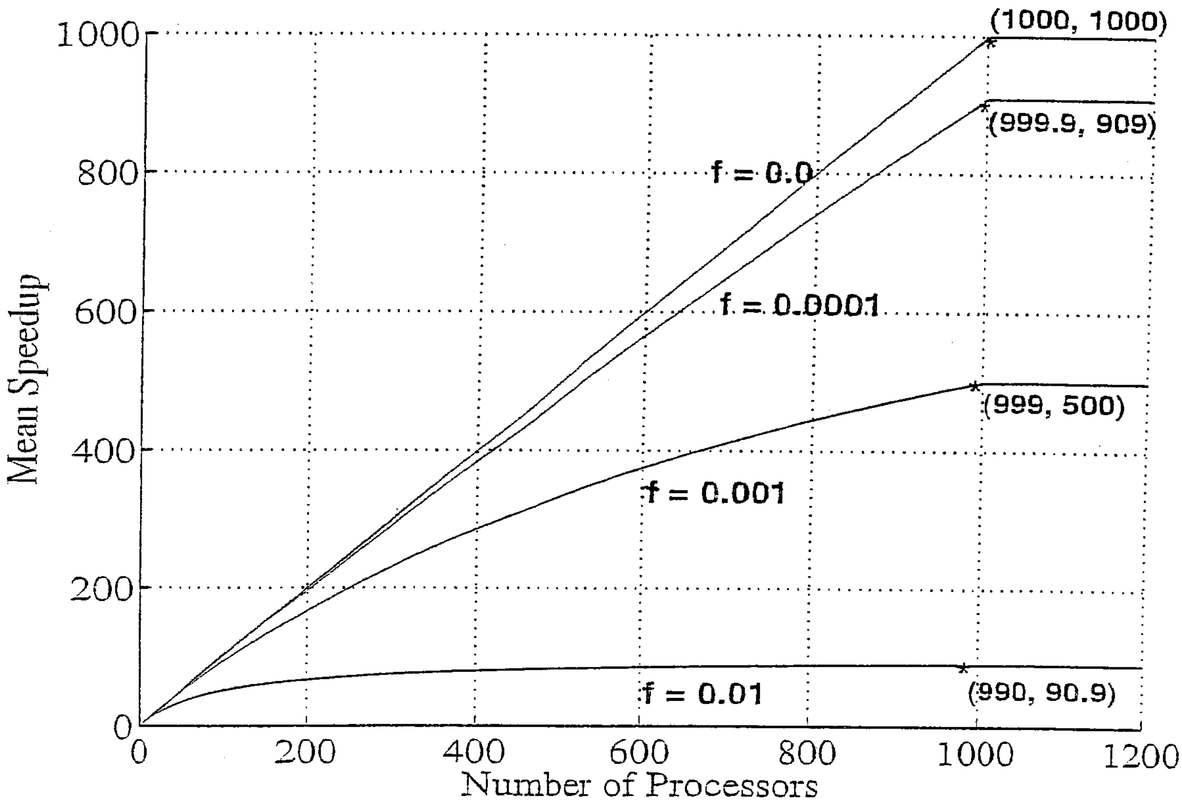
Properties of MRIP estimator of errors of variance: coverage analysis

The method of ”variance as the mean” ith SA/HW estimator of variance



Other methods are not satisfactorily robust
to be implemented in automated setting of MRIP

Truncated Amdahl's law of speedup in MRIP



Assumptions:

- 1000 checkpoints needed
- f = relative length of non-parallelisable stage of simulation (for example: the length of initial transient phase in steady-state simulation)

See *K. Pawlikowski and D. McNickle. Proc. ESS'2001*, Marseille, October 2001

IMPLEMENTATION OF MRIP

AKAROA2 = p**A**c**K**age for **A**utomatic gene**R**ation and process c**O**ntrol of p**A**rallel stochastic simulation (version **2**)

- written in C++
- tested on Sparc/SunOS4 & Solaris2, i386/Solaris2 and Linux
- automatic parallelisation
- automatic data collection and on-line analysis.

```
/*  *   mm1.C - M/M/1 Queueing System   */
#include "akaroa.H"
#include "akaroa/distributions.H"
#include "akaroa/process.H"
#include "akaroa/resource.H"
double arrival_rate; // Arrival rate of customers
double service_rate; //Service rate of customers
// Server, as a Resource with a capacity of 1 unit.
Resource server(1);

class Customer : public Process
{
public:
  void LifeCycle();
};
void Customer::LifeCycle()
{
  Time arrival_time, time_in_system;
  arrival_time = CurrentTime();
  server.Acquire(1);
  Hold(Exponential(1/service_rate));
  server.Release(1);
  time_in_system = CurrentTime()-arrival_time;
AkObservation(time_in_system)
};
// The main program.
// After getting the load from the
// command line and calculating arrival and service
// rates, we enter a loop generating new customers
// at the arrival rate.
int main(int argc, char *argv[])
{
  real load = atof(argv[1]);
  service_rate = 10.0;
  arrival_rate = load * service_rate;
  for (;;)
{
    new Customer;
    Hold(Exponential(1/arrival_rate));
}
```

Main functional blocks of AKAROA.2

akrun (shell command interface)

- **initiates simulation**
- **reports results to user**

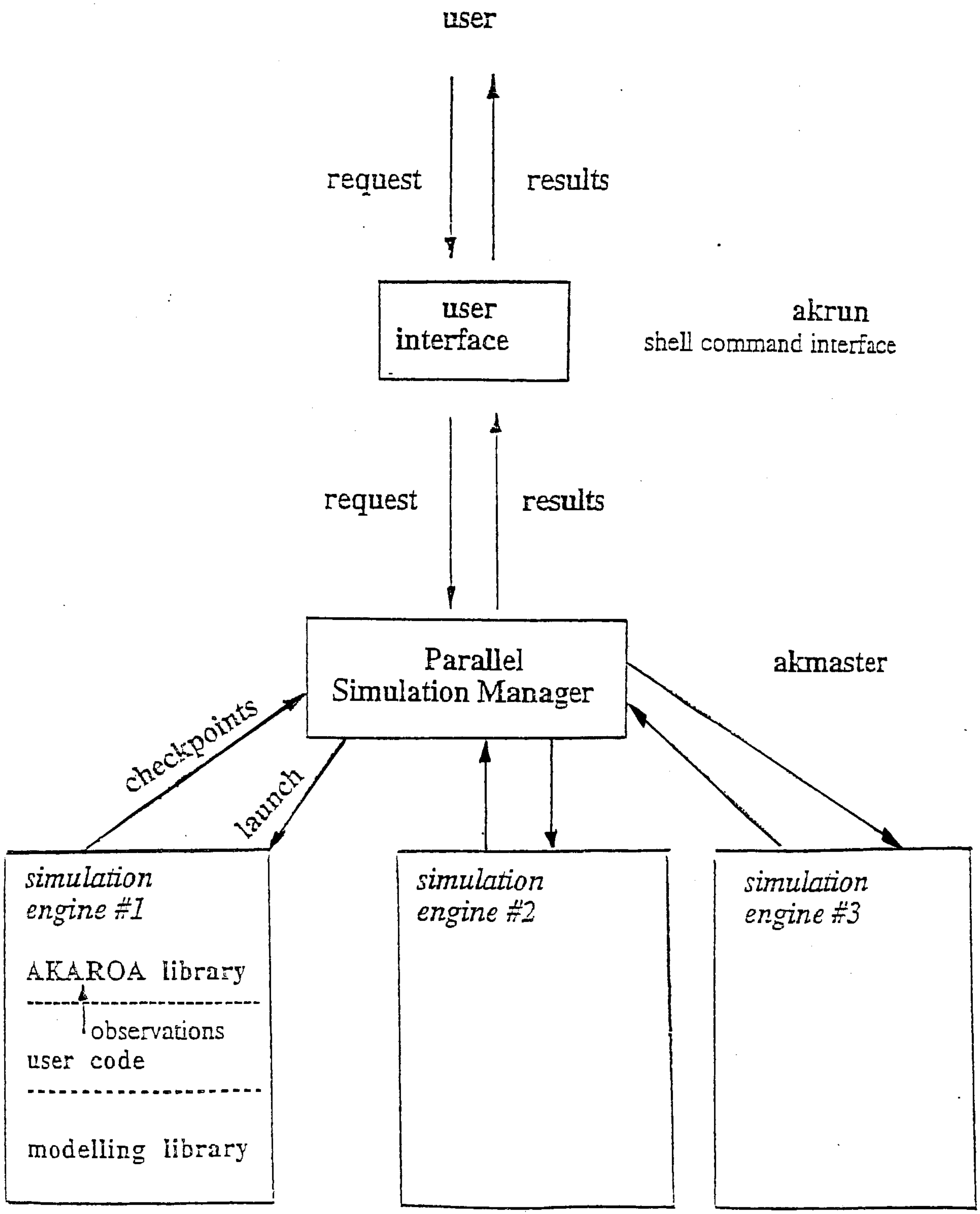
akmaster (Parallel Simulation Manager)

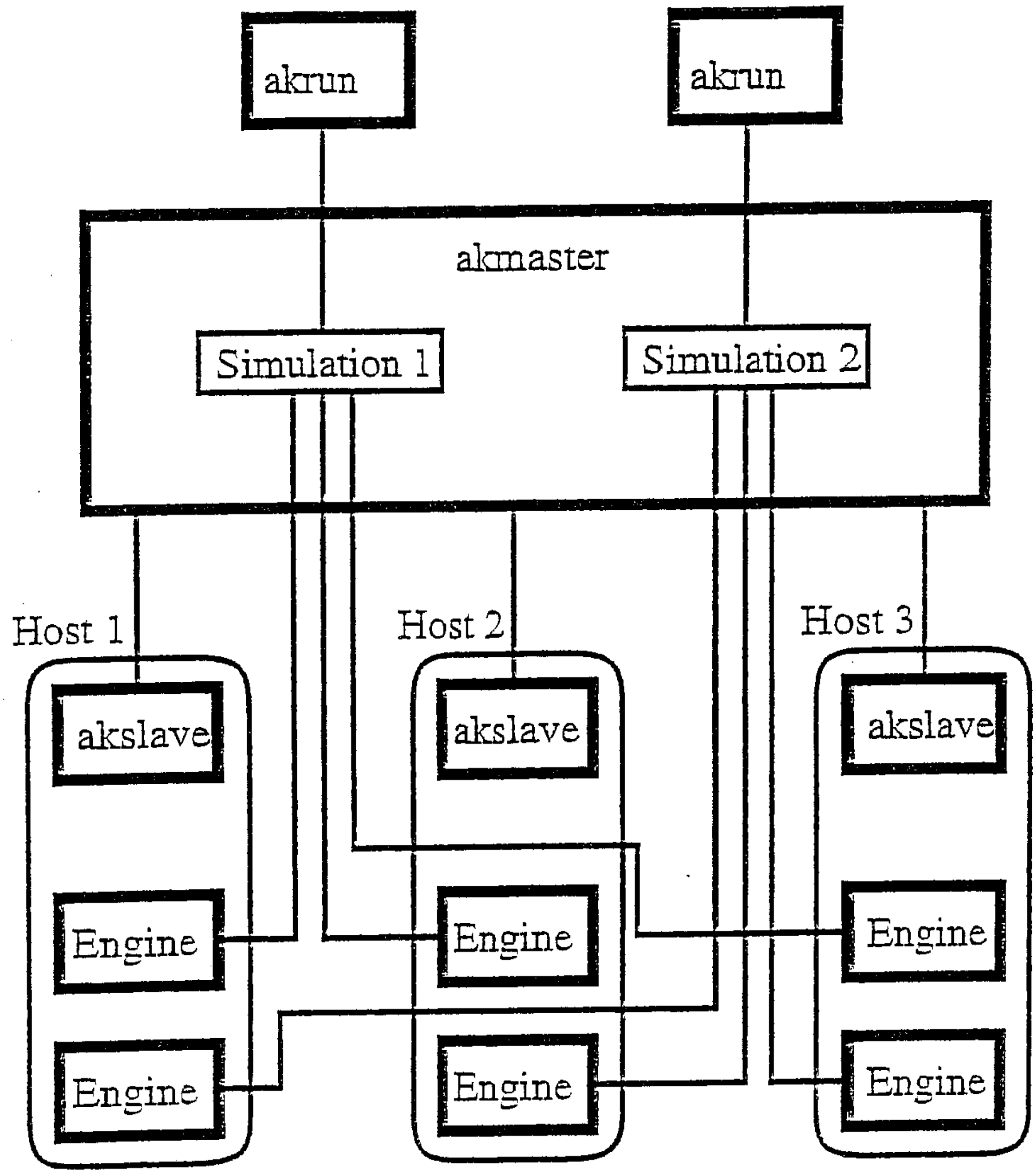
- **manages launching of simulation engines**
- **receives checkpoint data and calculates global estimates**
- **controls simulation run length**

akgui:

AKAROA.2 graphical user interface

Interprocess Communication
based on UNIX Internet-domain stream sockets

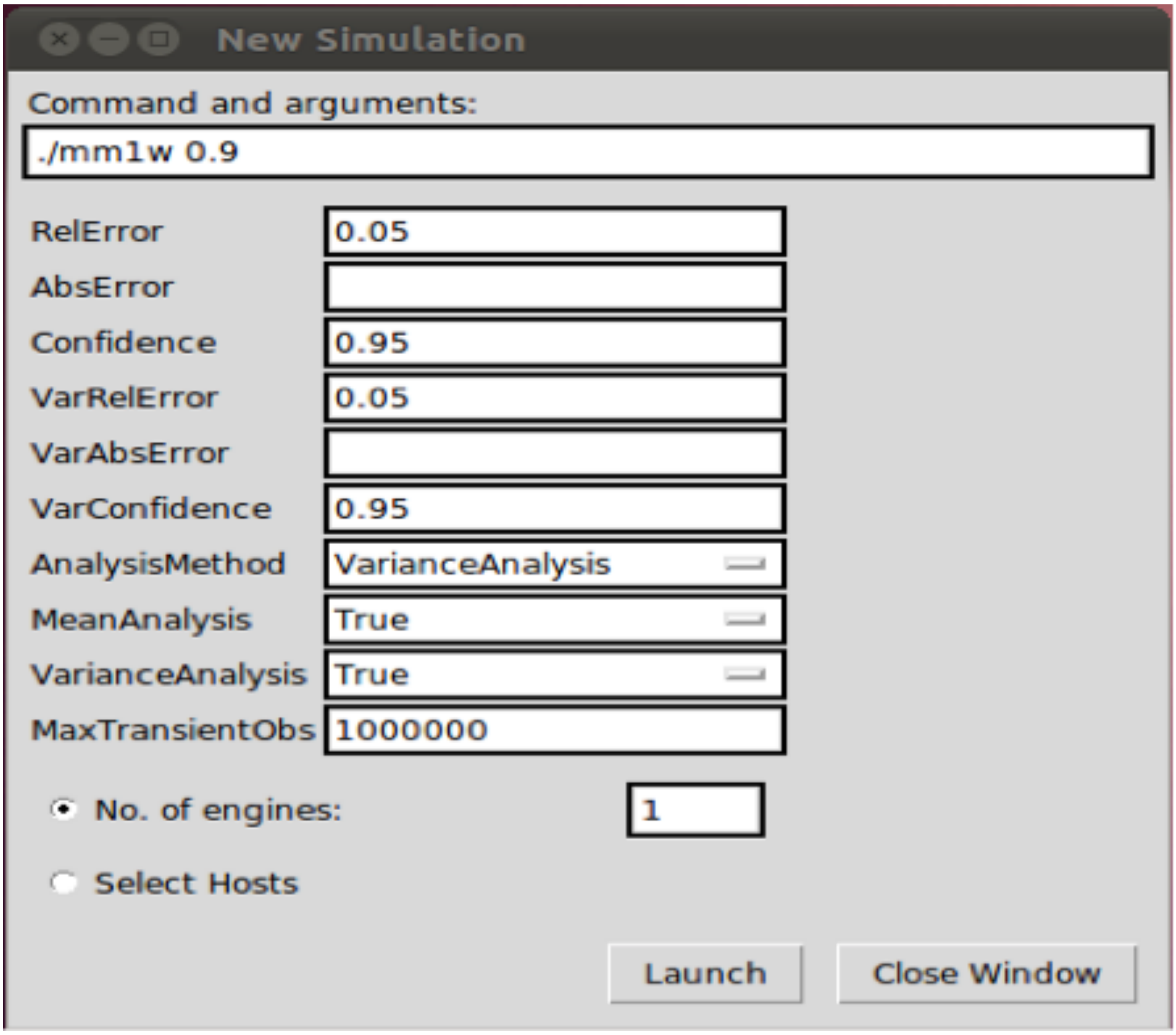

user
request
results
user interface
akrun
shell command interface
request
results
Parallel Simulation Manager
akmaster
checkpoints
launch
simulation engine #1
AKAROA library
observations
user code
modelling library
simulation engine #2
simulation engine #3



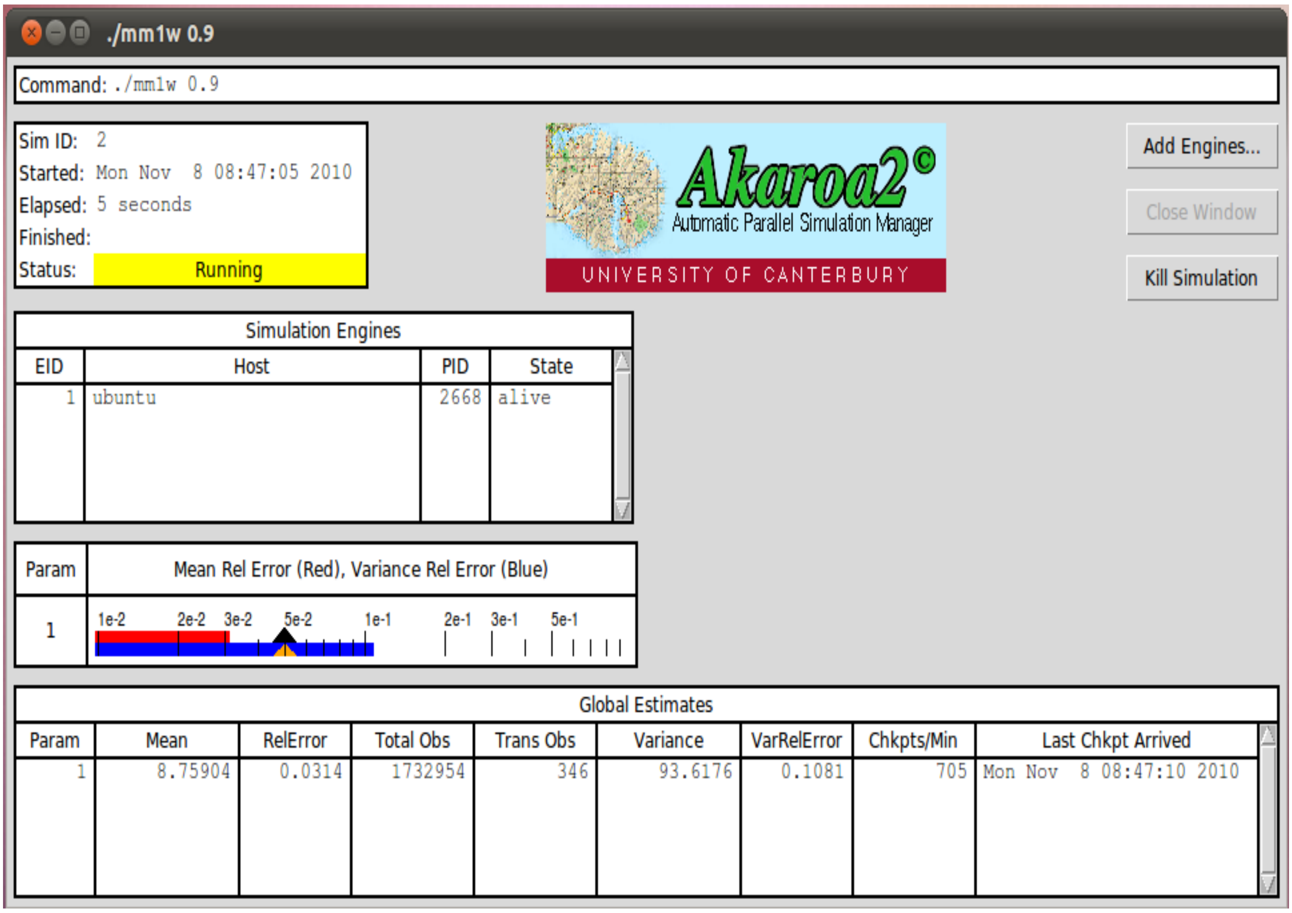
AKAROA.2 architecture

(2 simulations in progress, each one with
3 engines run on separate network hosts).

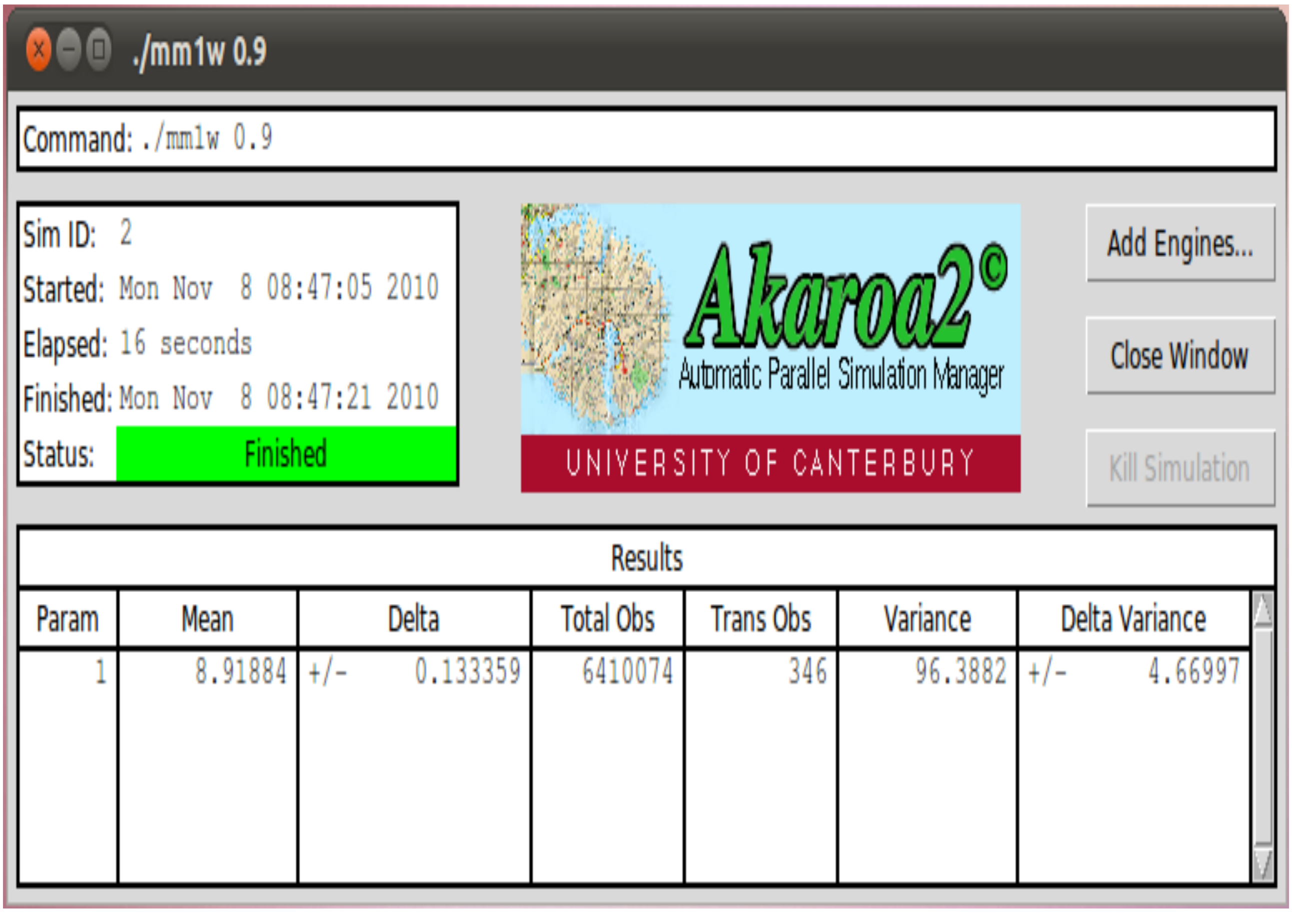
Akgui (mean and variance analysis):
before launching simulation



Akgui (mean and variance analysis): during simulation



Akguim (mean and variance analysis): end of simulation



Some special features of AKAROA2:

- dynamic parallelisation
- fault tolerance
  (*if simulation engine is lost, simulation continues with remaining engines*)

AKAROA 2 as MRIP controller of other simulation packages

- AKAROA2/Ptolemy interface
- AKAROA2/NS2 interface
- AKAROA2/OMNET++ interface

Simulation engines can be distributed:
- **on a LAN, or**
- **world-wide** , e.g. on PlanetLab; see

`www.planetlabnz.canterbury.ac.nz/docs/ieice09_yasmeen`

Akaroa 2 is offered as a free-ware
for non-profit research at universities.

See *www.akaroa.canterbury.ac.nz*

A counter on Akaroa2 webpage has recently shown:

over 200 000 visits, and

over 2700 downloads of the code

(May 2001 - March 2011)

Akaroa2 is used at universities in about 80 countries, including:

```
Australia                 Korea
Belarus                   Lithuania
Belgium                   Macedonia
Bosnia                    Malaysia
Brazil                    Mexico
Canada                    New Zealand
Chile                     Nigeria
China                     Norway
Colombia                  Pakistan
Cyprus                    Poland
Denmark                   Russia
England                   Singapore
Finland                   Slovakia
France                    South Africa
Germany                   Spain
Greece                    Switzerland
Guana                     Taiwan
Hong Kong                 Thailand
Hungary                   The Netherlands
India                     Trinidad
Indonesia                 Tunisia
Iran                      Turkey
Ireland                   Ukraine
Italy                     United Kingdom
Jakarta                   United States
Japan                     Vietnam
Jordan
```

Commercial users include **IBM Research Labs in Zurich**

Currently:

Akaroa2 can automatically produce accurate estimates of mean values, probabilities, proportions and variances.

Work in progress:

- quantiles and
- probabilities of rare events.

Future work:

- MRIP of distributed simulation models (distributed simulation on multiple clusters of workstations)

- distributed optimization based on MRIP scenario.

Thank you

Questions ?