Performance Prediction of Conservative Parallel Discrete Event Simulation

Gábor Lencse
and
András Varga

The Aim of our Work

- Parallel Discrete Event Simulation is done to achieve speed-up compared to sequential simulation
- PDES is not an easy task
- Before investing work in parallelisation, one would like to predict if it worth doing so
- We need a method to determine whether a simulation model has a potential for good speed-up

The Topics Covered

- PDES synchronisation methods
- The method for assessing available parallelism in a model
- Hardware software and environment
- Simulation model for testing the method
- Results
- Conclusions

Parallel Discrete Event Simulation

- Parallelisation of DES
  - the simulated system is divided into partitions
  - the partitions are assigned to processors
  - the processors execute the partitions **maintaining causality** (synchronization method)
  - the achievable speed-up depends on the method used for inter-processor synchronization
- Synchronization Methods for PDES
  - **Conservative (Null Message Algorithm)**
  - Optimistic (Time Warp)
  - Statistical Synchronization

The Method for Assessing Available Parallelism in a Model

- The method was proposed in
  Varga, A., Y. A. Sekercioglu and G. K. Egan. 2003. "A practical efficiency criterion for the null message algorithm". *Proceedings of the European Simulation Symposium (ESS 2003)*, (Oct. 26-29, 2003, Delft, The Netherlands.) SCS International, 81-92.

  Will be referred as: (Varga et. al. 2003)
- Our aim is to test it for higher number of CPUs
- The method uses quantities that can be easily measured on sequential simulations

The Method for Assessing Available Parallelism in a Model (Parameter #1)

- ***P*** *performance* represents the number of events processed per second (*ev*/*sec*).
- *P* depends on the performance of the hardware and the amount of computation required for processing an event.
- *P* is independent of the size of the model.

The Method for Assessing Available Parallelism in a Model (Parameter #2)

- ***E*** *event density* is the number of events that occur per simulated second (*ev/simsec*).
- *E* depends on the model only, and not on the hardware and software environment used to execute the model.
- *E* is determined by the size, the detail level and also the nature of the simulated system.

The Method for Assessing Available Parallelism in a Model (Parameter #3)

- ***R*** *relative speed* measures the simulation time advancement per second (*simsec*/*sec*).
- $R = P/E$.

The Method for Assessing Available Parallelism in a Model (Parameter #4)

- ***L*** *lookahead* is measured in simulated seconds (*simsec*).
- When simulating telecommunication networks and using link delays as lookahead, *L* is typically in the *microsimsec–millisimsec* range.

The Method for Assessing Available Parallelism in a Model (Parameter #5)

- $\boldsymbol{\tau}$ *latency* (sec) is the latency of sending a message from one Logical Process (LP) to another.
- τ is usually in the μs-ms range, and is largely determined by the hardware and software on which the simulation runs.

The Method for Assessing Available Parallelism in a Model (Parameter #6)

- λ *coupling factor* can be calculated as the ratio of *LE* and *τP:*

$$\lambda = \frac{L \cdot E}{\tau \cdot P}$$

- The paper (Varga et. al. 2003) states that the chance of the good speed-up of the PDES using the conservative synchronisation method can be predicted on the basis of the magnitude of λ.

Hardware Environment

- A cluster of 12 PCs with
  - *AMD Athlon 64 X2 Dual Core 4200*+ CPU
  - *2*1GB DDR2 667MHz* (dual channel) RAM
  - *NVIDIA nForce® 500 SLI™ MCP*
  - *built-in Gigabit Ethernet* NIC
- *3Com 2948-SFP* Gigabit Ethernet switch

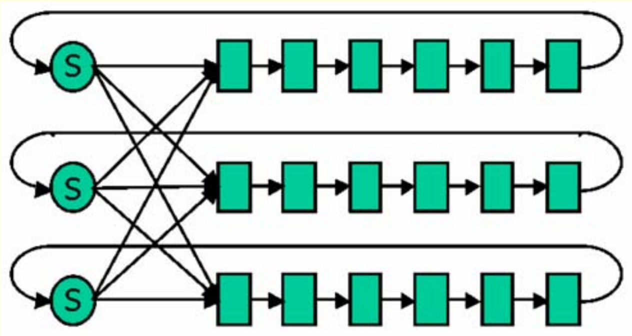
  communication latency (*L*) about *25μs*
- *SUN Fire X4200 M2* NFS server for home directories

Software Environment

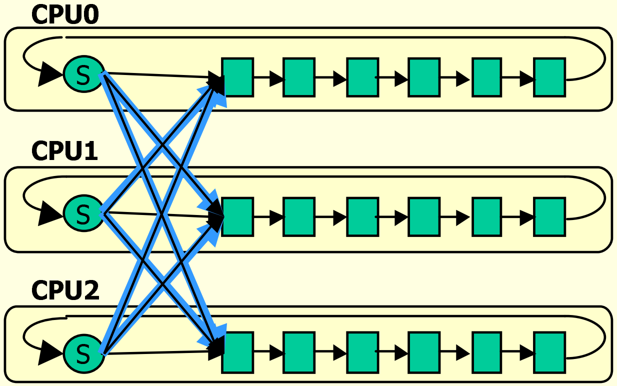
- Debian Squeeze GNU/Linux host OS
- Private IP address space
- LAM/MPI 7.2.1 cluster software
- OMNeT++ 4.0p1 simulation environment

Simulation Model

- Parallel Closed Queueing Network simulation sample program of OMNeT++



M=3 Tandem Queues
with *k*=6 Single Server Queues
in Each Tandem Queue



Partitioning the CQN Model

Simulation Model Parameters

- *M*=24 tandem queues
- *k*=50 queues in each tandem queue
- exponential service time of the queues with expected value of 10 seconds
- Parameters tuned:
  - N – number of logical processes
  - L – delay between the tandem queues
- All the other parameters were left unchanged.

Estimation for the λ Parameter

- The parameters for the calculation of λ were measured in the sequential simulation for L=100ms, and we got:

$$\lambda = \frac{L \cdot E}{\tau \cdot P} = \frac{100 \cdot 156}{25 \cdot 10^{-6} \cdot 250000} \approx 2500$$

- The value of λ decreases with the number of LPs. If we use *N* number of LPs, then:

$$\lambda_N = \frac{\lambda}{N}$$

Vacationing Jobs

- As *L* increases, a higher proportion of the jobs will be "buffered" in the long-delay links among the tandems, that is, they are effectively removed from the queueing inside the tandems.

The influence of the Vacationing Jobs on λ

- The Values of λ in the Function of L

  (measured vs. calculated from the initial estimation)

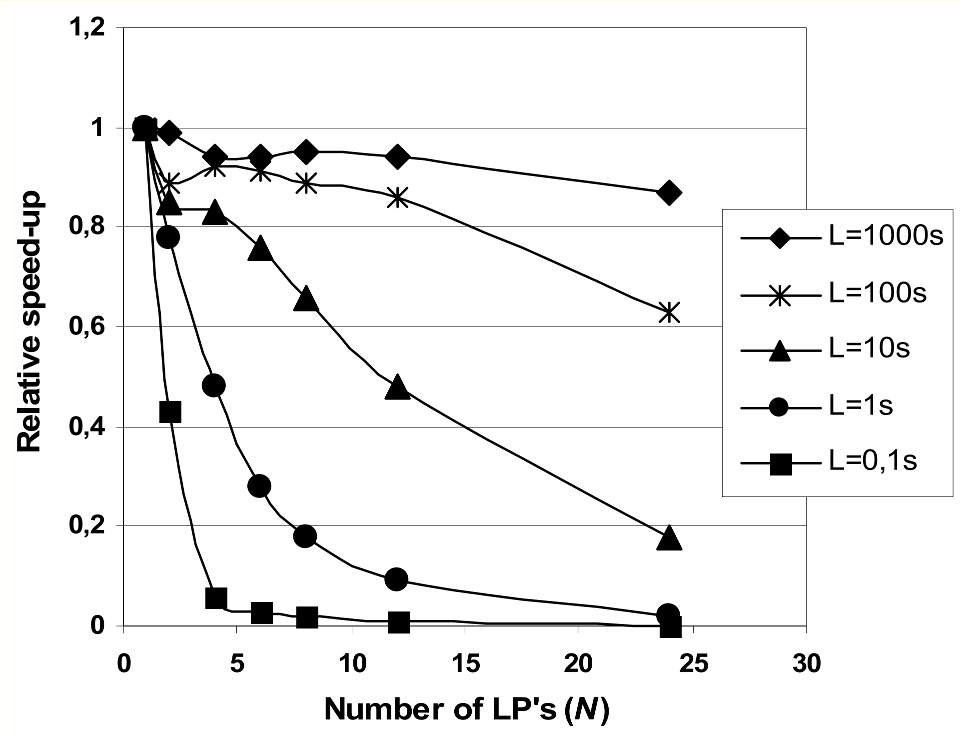
***L** [simsec]*	**0.1**	**1**	**10**	**100**	**1000**
# events	138122606	138091806	137816386	134885378	102957082
exec. t.*[sec]*	524.18	521.36	523.09	516.54	415.73
P [ev/sec]	263502.24	264868.43	263465.92	261132.49	247653.72
E [ev/simsec]	159.86	159.83	159.51	156.12	119.16
meas.'d λ	**2.43**	**24.14**	**242.17**	**2391.39**	**19246.76**
λ_0*L/L_0	2.50	25.00	250.00	2500.00	25000.00

Results for the Speed-up

- To fully explore the effect of the magnitude of λ on the available speed-up, we conducted a series of experiments for some values of *L: L*=100ms, 1s, 10s, 100s and 1000s.
- The results can be found in the proceedings.
- For further discussion, we use the value of the relative speed-up, that is the value of the speed-up divided by the number of CPUs used.

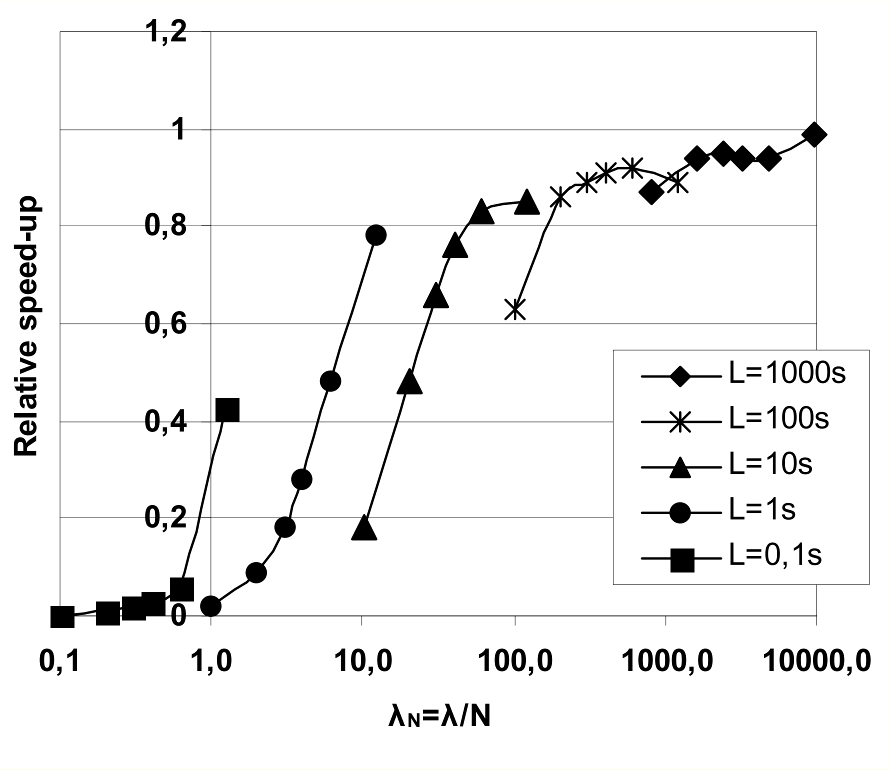
Results for the Relative Speed-up

- Relative Speed-up in the Function of *N* for *L*=0.1s, 1s, 10s, 100s, 1000s Lookahead



Relative Speed-up in the Function of λ_N

- Relative Speed-up in the Function of λ_N for Different Values of *L*, and for *N*=24, 12, 8, 6, 4, 2



Summary

- We have used a closed queueing network as simulation model, and run it on up to 24 CPU cores.
- We have experimentally verified that a coupling factor of $\lambda >> 1$ is a necessary precondition of getting a good speed-up with conservative parallel simulation.
- The 10..100 range of λ_N can provide an acceptable speed-up, and there is a high chance for a good speed-up if λ_N is above that range.
- The results confirm that with our model, a $\lambda_N = \lambda / N$ (N being the number of LPs) value near or below 1 practically prohibits good parallel performance.

Conclusion

- We conclude that the criterion for λ provides a quick and convenient way to determine whether it makes sense to experiment with parallelizing a particular simulation model or not, before actually investing work in the parallelization.
- We have tested this method and we have found that it works for even higher number of processors up to 24 CPU cores.