Modelling multiprocessor systems in PEPA

Andrea Marin
Università Ca' Foscari Venezia
DAIS

marin@dais.unive.it

1 Model description

The model is inspired from [2] (see also [1, 3]). Consider a multiprocessor system with a shared memory. Processes running on this system have to compete for access to the common memory: to gain access and to use the common memory they need also to acquire the system bus which is released when access to the common memory is terminated; for simplicity the bus will not be explicitly represented in the following. Processes are mapped onto processors. The processors are not explicitly represented but they determine the rate of activities in the associated processes, i.e. all processes have the same functional behaviour, but actions progress at different speeds depending on the processor on which they are running, and the number of processes present on the processor. It is the modeller's responsibility to select rates appropriately. A protocol which is not completely fair, but simply prevents one processor from monopolising the memory, imposes that after each access of a processor to the memory, some other processor must gain access before the first can access again. A process running on the ith processor is represented as P_i:

$$P_i = (\text{think}, \lambda_i).(\text{get}_i, g).(\text{use}, \mu_i).(\text{rel}, r).P_i\,.$$

In this case, in order to impose the protocol, the memory is modelled as remembering which processor had access last Access for this processor is disabled.

$$Mem_i = \sum_{\substack{i=1\\ i\neq j}}^{N} (\text{get}_j, \top).(\text{use}, \top).(\text{rel}, \top).Mem_j$$

2 Modelling challenge

According to the specifications given above, model a quad-core system. Find an appropriate PEPA model for the quad-core system in which processors $P_1, \ldots, P_3$ are identical and run 2 identical processes each. Processor P_4 runs only one process.

3 Getting performance bounds

We define the throughput of a process as the expected number of cycles in the think state it performs in steady-state. The total system throughput is the sum of the throughput of the processes.

- Derive the maximum throughput of one process assuming that the memory is always available
- Why is the index above important in the performance analysis?

4 Instance of the model

Write the PEPA equations for the model in Eclipse. For the identical processes running on the identical processors $P_1, \ldots, P_3$ we take $\lambda_i = 1$, and $\mu_i = 0.55$ for $i = 1, 2, 3$ and leave P_4 spare. Take $g = 7$ and $r = 7$.

- Derive the total system throughput. Would you say that the memory is a bottleneck of the system?
- Study the ratio between the throughput derived with PEPA plugin and the maximum (assuming always available memory) throughput of one process for λ going from 0.2 to 5 (steps of 0.4). Before trying it, when do you expect to find values closer to 1?

5 Malicious behaviours

We wish to test if the protocol that prevents a processor to monopolize the memory works well under malicious processes. We assume that the unique process in P_4 alternates small thinking times $\lambda = 10$ and long memory usage times $\mu = 0.5$ (while for all the other processes $\lambda = 1$ and $\mu = 0.55$). Compare the throughput of one of the processes running in P_i with $i = 1, 2, 3$ with the policy describe above and without. Notice that the latter case requires

to change the model. Is the policy efficient? How could you change it to improve the performances?

References

[1] S. Gilmore and J. Hillston. The PEPA Workbench: A tool to support a process algebra based approach to performance modelling. In *Proc. of Seventh Int. Conf. on Modelling Techniques and Tools for Comp. Perf. Eval., LNCS 794*, pages 353–368, Vienna, 1994. Springer.

[2] S. Gilmore, J. Hillston, and M. Ribaudo. An Efficient Algorithm for Aggregating PEPA Models. *IEEE Trans. on Software Eng.*, 27(5):449–464, 2001.

[3] M. Tribastone, A. Duguid, and S. Gilmore. The PEPA Eclipse Plug-in. *Perf. Eval. Review*, 36(4):28–33, 2009.