

## Efficient Parallel MonteCarlo with FastFlow

## Time- and Space-Efficient Parallelisation of the StochKit Framework with FastFlow

## Abstract

The stochastic simulation of natural systems is a very informative but happens to be computationally expensive. We present StochKit-FF, an parallel version of StochKit, a reference toolkit for stochastic simulations that substantially improves StochKit performances on multi-core platforms.

## StochKit-FF

StochKit is an extensible simulation framework that aims at making stochastic simulation accessible to practicing biologists; it implements, among the others, the popular Gillespie algorithm (5). StochKit-FF extends StochKit with two main features: the support for the parallel run of multiple simulations on multicores, and the support for the online *reduction* of simulation results (*selective memory*).

StochKit-FF is semi-automatically derived from StochKit by means of the FastFlow, i.e. a pattern-based programming toolkit for multi-cores that make it possible the parallelisation existing, possibly complex C/C++ codes (4).

In particular, StochKit-FF exploits of the FastFlow *farm* pattern, which implement the master-worker (3). StochKit-FF exploits it by running on workers different simulations, which are either multiple instances or a parameter sweep of a single simulation (2).

The results of different simulation instances (a.k.a. trajectories) are combined in parallel via a *selective memory*, i.e. a data structure supporting the on-line *reduction* of trajectory data by way of user-defined associative and commutative functions. Selective memory distinguishes from standard parallel *reduce* operation because it works on (possibly unbound) streams of non-time-aligned simulation data.

The behaviour of selective memory is exemplified in Fig. 1. Trajectories from different simulations are first averaged at aligned simulation time points: the resulting trajectory is oversampled with respect a single simulation (Fig. 1 left) and can be reduced again along time axis (Fig. 1 right). Overall, selective memory produces a combined simulation that has been adaptively sampled along the two axes.

Fig. 1. – Selective Memory. Left) Alignment and averaging of different simulations via oversampling. Right) Sampling reduction along time axis.

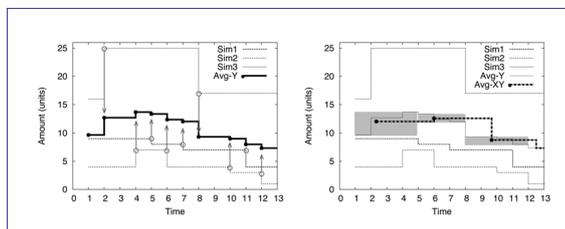
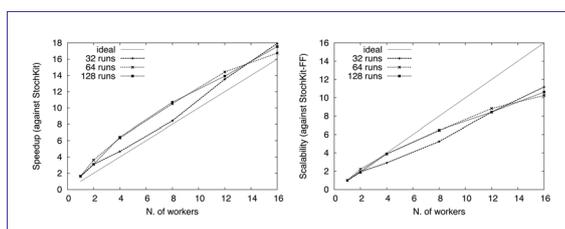


Fig. 2. – Left) Speedup of StochKit-FF against StochKit. Right) Scalability of StochKit-FF ( $n$ ) against StochKit-FF (1), where  $n$  is the number of workers.



Selective memory effectively mitigates the memory pressure of results logging when running many simulations on a multicore because substantially reduces output size, thus cache capacity misses and memory bus pressure.

## Experimental Results

We report on experiments performed on Sun X4600 16 cores platform hosted at EPCC, University of Edinburgh.

The performances of StochKit-FF have been evaluated on the HIV diffusion case-study (2, 1). A single run of the simulation with StochKit averagely produces  $\sim 150M$  simulation points for 4000 days of simulated time, which will turn into about 6 GBytes of generated data; multiple runs of the same simulation will need a linearly greater time and space. These simulations can be naively parallelised on a multicore platform by running several independent instances, which however, will compete for memory and disk accesses, thus lead to suboptimal performances in the case of high output pressure.

StochKit-FF mainly attacks these costs by online reducing the outputs of simulations, which are run in parallel. In Fig. 2, average and variance are used as combining functions. As shown in Fig. 2 (left), StochKit-FF exhibits a superlinear speedup with respect to StochKit in all tested cases, which is mainly due to the faster memory allocator provided by FastFlow. As shown in Fig. 2, StochKit-FF exhibits a good scalability also when compared with the sequential (one-thread) version of StochKit-FF. StochKit-FF is available as open source software under LGPLv3 licence (4).

**Acknowledgements.** The work was carried out under the HPC-EUROPA2 project (project number: 228398) with the support of the European Commission - Capacities Area - Research Infrastructures. We wish to thank Gianfranco Balbo, Andrea Bracciali and Pietro Liò for the many fruitful discussions.

## References

- (1) M. ALDINUCCI, A. BRACCIALI and P. LIÒ, *Formal synthetic immunology*. ERCIM News, 82, July 2010.
- (2) M. ALDINUCCI, A. BRACCIALI, P. LIÒ, A. SORATHIYA and M. TORQUATI, *StochKit-FF: Efficient systems biology on multi-core architectures*. TR-10-12, Università di Pisa, Dipartimento di Informatica, Italy, 2010.
- (3) M. ALDINUCCI, M. MENEGHIN and M. TORQUATI, *Efficient Smith-Waterman on multi-core with FastFlow*. In Proc. of Intl. Parallel Distributed and network-based Processing, Pisa, Italy, Feb. 2010. IEEE.
- (4) M. ALDINUCCI and M. TORQUATI, *FastFlow website* (2009). <http://mc-fastflow.sourceforge.net/>.
- (5) L. PETZOLD, *StochKit: stochastic simulation kit web page* (2009). <http://www.engineering.ucsb.edu/cse/StochKit/index.html>.