On Parallelizing On-Line Statistics for Stochastic Biological Simulations*

Marco Aldinucci[1], Mario Coppo[1], Ferruccio Damiani[1], Maurizio Drocco[1], Eva Sciacca[1], Salvatore Spinella[1], Massimo Torquati[2], and Angelo Troina[1]

[1] Department of Computer Science, University of Torino, Italy
{aldinucci,coppo,damiani,drocco,sciacca,spinella,troina}@di.unito.it
[2] Department of Computer Science, University of Pisa, Italy
torquati@di.unipi.it

Abstract. This work concerns a general technique to enrich parallel version of stochastic simulators for biological systems with tools for on-line statistical analysis of the results. In particular, within the FastFlow parallel programming framework, we describe the methodology and the implementation of a parallel Monte Carlo simulation infrastructure extended with user-defined on-line data filtering and mining functions. The simulator and the on-line analysis were validated on large multi-core platforms and representative proof-of-concept biological systems.

Keywords: multi-core, parallel simulation, stochastic simulation, on-line clustering.

1 Introduction

The traditional approach to describe biological systems relies on deterministic mathematical tools like, e.g., Ordinary Differential Equations (ODEs). This kind of modelling becomes more and more difficult when the complexity of the biological systems increases. To address these issues, in the last decade, formalisms developed in Computer Science for the description of stochastically behaving computational entities have been exploited for of biological systems [15].

Biochemical processes, such as gene transcription, regulation and signalling, often take place in environments containing a (relatively) limited number of some reactants, or involve very slow reactions, and thus result in high random fluctuations, determining phenomena like transients or multi-stable behaviour. Stochastic methods can give an exact account of the system evolution in all situations and are playing a growing role in modelling biological systems. Stochastic modeling keeps track of the exact number of species present in a system and all reactions are simulated individually. These methods can be highly demanding in terms of computational power (e.g., when a large number of molecules or species

* This research has been funded by the BioBITs Project (Converging Technologies 2007, Biotechnology-ICT, Regione Piemonte). The authors acknowledge the HPC Advisory Council (www.hpcadvisorycouncil.com) University Award spring 2011.

M. Alexander et al. (Eds.): Euro-Par 2011 Workshops, Part II, LNCS 7156, pp. 3–12, 2012.

is involved) and data storage (e.g., when the amounts of each species for each time sample of a simulation have to be tracked).

A single stochastic simulation represents just one possible way in which the system might react over the entire simulation time-span. Many simulations are usually needed to get a representative picture of how the system behaves on the whole. Multiple simulations exhibit a natural independence that would allow them to be treated in a rather straightforward parallel way. On a multicore platform, they might exhibit serious performance degradation due to the concurrent usage of underlying memory and I/O resources.

In [2] we presented a highly parallelized simulator for the Calculus of Wrapped Compartments (CWC) [5] which exploits, in an efficient way, the multi-core architecture using the FastFlow programming framework [8]. The framework relies on selective memory [1], i.e. data structure designed to perform the online alignment and reduction of multiple computations. A stack of layers progressively abstract the shared memory parallelism at the level of cores up to the definition of useful programming constructs supporting structured parallel programming on cache-coherent shared memory multi- and many-core architectures.

Even in distributed computing the data processing of hundreds (or even thousands) simulations is often demoted to a secondary aspect in the computation and treated as off-line post-processing tools. The storage and processing of simulation data, however, may require a huge amount of storage space (linear in the number of simulations and the observation size of the time courses) and an expensive post-processing phase, since data should be retrieved from permanent storage and processed.

In this paper, we adapt the approach presented in [2] to support concurrent real-time data analysis and mining. Namely, we enrich the parallel version of the CWC simulator with on-line (parallel) statistics tools for the analysis of results on cache-coherent, shared memory multicore. To this aim, we exploit the FastFlow framework, which makes it possible not only to run multiple parallel stochastic simulations but also combine their results on the fly according to user-defined analysis functions, e.g. statistical filtering or clustering. In this respect, it is worth noticing that while running independent simulations is an embarrassingly parallel problem, running them aligned at the simulation time and combining their trajectories with on-line procedures definitely is not as merging high-frequency data streams. This, in turn, requires to enforce that simulations proceed aligned according to the simulation time in order to avoid the explosion of the working set of the statistical and mining reduction functions.

2 The CWC Formalism and Its Parallel Simulator

The Calculus of labelled Wrapped Compartments (CWC) [5,2] has been designed to describe biological entities (like cells and bacteria) by means of a nested structure of ambients delimited by membranes.

The terms of the calculus are built on a set of *atoms* (representing species e.g. molecules, proteins or DNA strands) , ranged over by a, b, ..., and on a set

of *labels* (representing compartment types e.g. cells or tissues), ranged over by $\ell, \ldots$. A *term* is a multiset $\overline{t}$ of *simple terms* where a simple term is either an atom a or a compartment $(\overline{a} \rfloor \overline{t'})^{\ell}$ consisting of a *wrap* (a multiset of atoms $\overline{a}$), a *content* (a term $\overline{t'}$) and a *type* (a label ℓ).

Multisets are denoted by listing the elements separated by a space. As usual, the notation $n * t$ to denotes n occurrences of the simple term t. For instance, the term $2 * a \ \ (b \ \ c \rfloor d \ \ e)^{\ell}$ represents a multiset containing two occurrences of the atom a and an ℓ-type compartment $(b \ \ c \rfloor d \ \ e)^{\ell}$ which, in turn, consists of a wrap with two atoms b and c on its surface, and containing the atoms d and e[1].

Interaction between biological entities are described by rewriting rules written as $\ell : P \mapsto O$ where P and O are terms built on an extended set of atomic elements which includes variables (ranged over by X, Y,...) and ℓ represents the compartment type to which the rule can be applied. An example of rewrite rule is $\ell : \ a \ b \ X \mapsto c \ X$ that is often written as $\ell : \ a \ b \mapsto c$ giving X for understood to simplify notations.[2] The application of a rule $\ell : P \mapsto O$ to a term $\overline{t}$ consists in finding (if it exists) a subterm $\overline{u}$ in a compartment of type ℓ such that $\overline{u} = \sigma(P)$ for a ground substitution σ and replacing it with $\sigma(O)$ in $\overline{t}$. We write $\overline{t} \mapsto \overline{t'}$ to mean that $\overline{t'}$ cam be obtained from $\overline{t}$ by applying a rewrite rule.

The standard way to model the time evolution of biological systems is that presented by Gillespie [9]. In Gillespie's algorithm a rate function is associated with each considered chemical reaction which is used as the parameter of an exponential distribution modelling the probability that the reaction takes place. In the standard approach this reaction rate is obtained by multiplying the kinetic constant of the reaction by the number of possible combinations of reactants that may occur in the region in which the reaction takes place, thus modelling the law of mass action. In this case a stochastic rule is written as $\ell : P \stackrel{k}{\longmapsto} O$ where k represent the kinetic constat of the corresponding reaction.

The CWC simulator [6] is an open source tool under development at the Computer Science Department of Turin University, implements Gillespie's algorithm on CWC terms. It handles CWC models with different rating semantics (law of mass action, Michaelis-Menten kinetics, Hill equation) and it can run independent stochastic simulations, featuring deep parallel optimizations for multi-core platforms on the top of FastFlow [8].

3 On Line Statistical Tools

Most biological data from dynamical kinetics of species might require further processing with statistical or mining tools to be really functional to biologists. In particular, the bulk of trajectories coming from Monte Carlo simulators can exhibit a natural unevenness due to the stochastic nature of the tool and are typically represented with many and large data series. This unevenness, in the form

[1] For uniformity we assume that the term representing the whole system is always a single compartment labelled $\top$ with an empty wrap.

[2] We force *exactly* one variable to occur in each compartment content and wrap. This prevents ambiguities in the instantiations needed to match a given compartment.

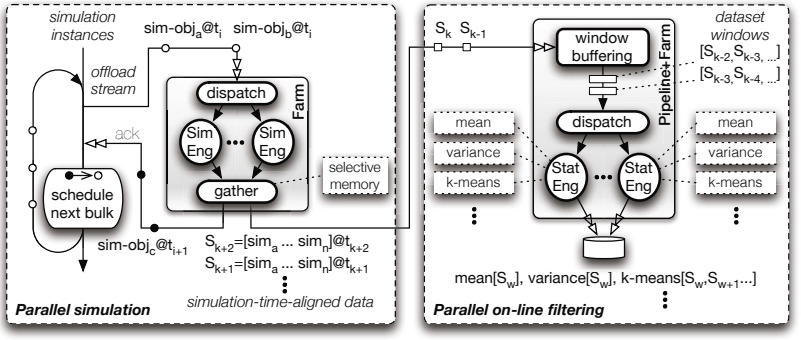


Fig. 1. CWC simulator with on-line parallel filtering: architecture

of deviant trajectories, high variance of results and multi-stable behaviours, often represents the real nature of the phenomena that is not captured by traditional approaches, such as ODEs.

Several techniques for analysing such data, e.g. principal components analysis, linear modelling, canonical correlation analysis have been proposed. We envision next generation software tools for natural sciences as able to perform this kind of processing in pipeline with the running data source, as a partially or totally on-line process because: 1) it will be needed to manage an ever increasing amount of experimental data, either coming from measurement or simulation, and 2) it will substantially improve the overall experimental workflow by providing the natural scientists with an almost real-time feedback, enabling the early tuning or sweeping of the experimental parameters.

On-line data processing requires data filtering and mining operators to work on streamed data and, in general, the random access to data is guaranteed only within a limited window of the whole dataset, while already accessed data can be only stored in synthesized form. When data filtering techniques, requiring to access the whole data set in random order, cannot be used, on-line data filtering and mining requires novel algorithms. The extensive study of these algorithms is an emerging topic in data discovery community and is beyond the scope of this work, which focuses on the design of a parallel infrastructure with the following general objectives: 1) efficient support for data streams and its parallel processing on multi-core platforms, and 2) easy engineering of battery of filters, that can be plugged in the tool without any concern for parallelism exploitation, data hazards and synchronisations.

These issues will be demonstrated by extending the existing CWC parallel simulator with a sample set of parallel on-line statistical measures computation including mean, variance, quantiles and clustering of trajectories (according to different methodologies such as K-means and Quality Threshold). The flexibility given by the possibility of running many different filters is of particular interest for the present work, as in many cases the searched pattern in experimental results is unknown and might require different kind of analysis tools.

The CWC parallel simulator, which is extensively discussed in [2] and sketched in Fig. 1 (left box), employs the selective memory concept, i.e. a data structure supporting the on-line reduction of time-aligned trajectory data by way of one or more user-defined associative functions (e.g. statistic and mining operators). Selective memory distinguishes from standard parallel reduce operation because it works on (possibly unbound) streams, and aligns simulation points (i.e. stream items) according to simulation time before reducing them: since each simulation proceed at a fixed time step, simulation points coming from different simulations cannot simply be reduced as soon as they are produced [1].

In this work, we further extend the selective memory concept by making it parallel via a FastFlow accelerator [8], which make it possible to offload selective memory operators onto a parallel on-line statistical tools implementing the same functions in parallel fashion. The pipeline has two stages: 1) statistic buffering, and 2) a farm of statistic engines. The first stage creates *dataset windows* (i.e. a number of arrays of simulation-time-aligned trajectory data from different simulation). The second stage farms out the execution of one or more filtering or mining functions, which are independently executed on different (possibly overlapping) dataset windows. Additional filtering functions can be easily plugged in by simply extending the list of statistics with additional (reentrant) sequential or parallel functions (i.e. adding a function pointer to that list). Overall, the parallel simulation (Fig. 1, left box) and parallel on-line filtering (Fig. 1, right box), work, in turn, in a two-stage pipeline fashion.

3.1 Typical Patterns for Biological Trajectories

Monostable Systems Analytical mathematical methods for steady-state analysis of deterministic models give insights on the dynamic equilibrium of a biological system over time. In the case of stochastic models are usually performed statistics on the mean and standard deviation of the system comparing the results with the correspondent deterministic mathematical model. Another useful analysis is the one based on quantiles calculation which approximate the distribution of simulation trajectories data over time.

Multi-stable Systems. Multi-stable biological systems play a significant role in some of the basic processes of life. The core behavior of these systems is based on genetic switches. Stochastic effects in these systems can be substantial as noise can influence the convergence to different equilibria.

Deterministic modeling of multi-stable systems is problematic. Bifurcation analysis of ODE based models traces time-varying changes in the state of the system in a multidimensional space where each dimension represents a particular concentration of the biochemical factor involved.

The effect of molecular noise in stochastic simulations causes the switching between the two stable equilibria if the noise amplitude is sufficient to drive the trajectories occasionally out of the basin of attraction of one equilibrium to the other. When stochastic simulations are performed a useful mining tool to capture these multi-stable behaviors is represented by curves clustering techniques. In

the presence of stochasticity in the data, direct clustering methods on aligned simulation results is not reliable. In order to keep the structure of the molecular evolution over time, we propose to apply the clustering procedure on data stream portions filtering numerically the data from the noise of the stochastic simulation and calculating the relative local trends.

In this work we employed two clustering techniques: K-means [10] and Quality Threshold (QT) [11] clustering. The clustering procedure collects the filtered data contained into the constant sliding time window Δ_W centered in the current data point $x_i \equiv f(t_i)$ where $t_i \equiv t_0 + i\Delta_S$ (where Δ_S is a constant sampling time) for all simulation trajectories and the extrapolated forecast point x_i^E referred to a future trend in time using the information of the Savitzky-Golay filter. Savitzky-Golay filter f_{SG} replaces the data value x_i by a linear combination of itself and some number of equally spaced nearby neighbors to the left (n_L) and to the right (n_R) of the data point x_i: $x_i^{SG} = f_{SG}(x_i) = \Sigma_{j=-n_L}^{n_R} c_j x_{i+j}$. The idea of the numerical filter is to find the coefficients c_j to approximate the underlying function within the sliding time window by a polynomial of degree M. The extrapolated forecast point x_i^E is calculated at a chosen time step Δ_F exploiting the derivatives coming from the filter in a Taylor series truncated at third term. The couple (x_i^{SG}, x_i^E) represents the trend of the curve at time t_i. A weighted metric distance employed by the clustering procedures on these couples phrase the similarity of behaviour between curves at time t_i using the information of data stream portions contained in the sliding time window Δ_W. This method is comparable with other curve clustering techniques (traditionally performed off-line) that partition the data keeping their functional structure.

Oscillatory Systems. Many processes in living organisms are oscillatory (e.g. the beating of the heart or, on a microscopic scale, the cell cycle). In these systems molecular noise plays a fundamental role inducing oscillations and spikes. We are currently working on statistical tools to synthesize the qualitative behavior of oscillations through peak detection and frequency analysis [16].

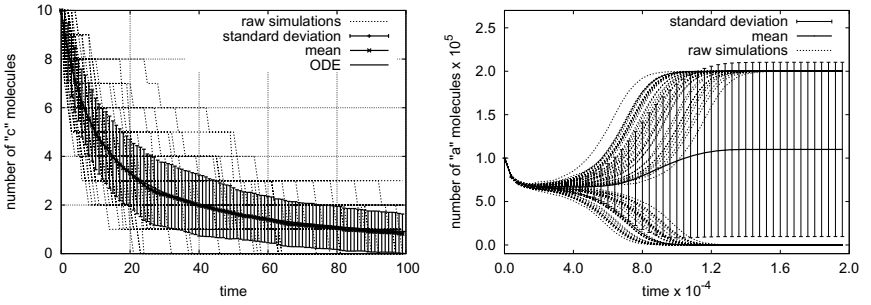
4 Examples

We now consider two motivating examples that illustrate the effectiveness of the presented real-time statistical and mining reduction functions.

Simple Crystallization. Consider a simplified CWC set of rules for the crystallization of species "a":

$$\top : 2 * a \xmapsto{1e-7} b \qquad \top : a\ c \xmapsto{1e-7} d$$

We here show how to reconstruct the first two moments of species "c" using the on-line statistics based upon 100 simulations running for 100 time units using a sampling time $\Delta_S = 1$ time unit. The starting term was: $T = 10^6 * a\ \ 10 * c$. Figure 2(a) shows the on-line computation of the mean and standard deviation



(a) Simple crystallization (b) Stable switch

Fig. 2. Mean and standard deviation on the simple crystallization and on the stable switch. The figures report also the raw simulation trajectories.

for species c. Notice that in these cases of mono-stable behaviors, the mean of the stochastic simulations overlap the solution of the corresponding deterministic simulation using ODEs.

Switches. We here consider two sets of CWC rules abstracting the behavior of a stable and an unstable biochemical switch [4] showing how to reconstruct the equilibria of the species using the on-line clustering techniques on the filtered trajectories. The stable switch with two competing agents a and c is based on a very simple population model (with only 3 agents) that computes the majority value quickly, provided the initial majority is sufficiently large. The essential idea of the model is that when two agents a and c with different preferences meet, one drops its preference and enters a special "blank" state b; b then adopts the preference of any non-blank agent it meets. The rules modeling this case are:

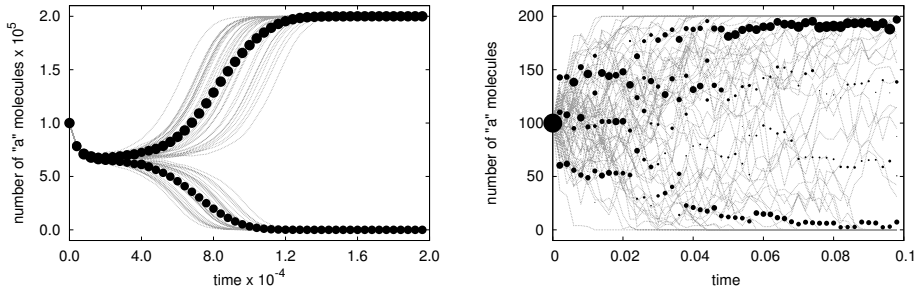
$$\top : a\ c \overset{10}{\longmapsto} c\ b \qquad \top : c\ a \overset{10}{\longmapsto} a\ b \qquad \top : b\ a \overset{10}{\longmapsto} a\ a \qquad \top : b\ c \overset{10}{\longmapsto} c\ c$$

The unstable switch is based on a direct competition where a species a catalyzes the transformation of another species b into a and, in turn, b catalyzes the transformation of a into b. In this example any perturbation of a stable state can initiate a random walk to the other stable state. The set of CWC rules modeling this case are:

$$\top : a\ c \overset{10}{\longmapsto} a\ a \qquad \top : c\ a \overset{10}{\longmapsto} c\ c$$

In these cases, simple mean and standard deviation are not significant to summarize the overall behavior. For instance in Fig. 2(b) the mean is not representative of any simulation trajectory.

Figures 3 a) and b) show the resulting clusters (black circles) computed on-line using K-means on the stable switch and QT on the unstable switch for species a over 60 stochastic simulations. The stable switch was run for $2 \cdot 10^{-4}$ time units with $\Delta_S = 4 \cdot 10^{-6}$. The number of clusters for K-means was set to 2. The starting term was: $T = 10^5 * a\ \ 10^5 * c$. The unstable switch was run for 0.1 time units with $\Delta_S = 2 \cdot 10^{-3}$. The threshold of clustering diameter for QT



(a) K-means clustering on the stable switch (b) QT clustering on the unstable switch

Fig. 3. On-line clustering results (black circles) on the stable and unstable switches using K-means and QT, respectively. The figures report also the raw simulations.

was set to 100. The starting term was: $T = 100 * a \ \ 100 * c$. Circles diameters are proportional to each cluster size.

K-means is suitable for stable systems where the number of clusters and their tendencies are known in advance, in the other cases QT, although more computationally expensive, can build accurate partitions of trajectories giving evidence of instabilities with a dynamic number of clusters.

Figure 4 shows the speedup of the simulation engines equipped with mean, standard deviation, quantiles, K-means, and QT filters on a 8 cores Intel platform against number of Simulation Engines with one and two Statistic Engines, respectively, on varying number of simulations and sampling rates. The first experiments show the ability of selective memory of reducing the I/O traffic as the speedup remain stable with increased number of simulations, thus output size. In the second experiment, the speedup decreases while the number of samples increases highlighting that the bottleneck of the system is in the data analysis stage of the pipeline: any further increase of Simulation Engines does not bring performance benefits.

5 Related Work

The parallelisation of stochastic simulators has been extensively studied in the last two decades. Many of these efforts focus on distributed architectures. Our work differs from these efforts in three aspects: 1) it addresses multicore-specific parallelisation issues; 2) it advocates a general parallelisation schema rather than a specific simulator, 3) it addresses the on-line data analysis, thus it is designed to manage large streams of data. To the best of our knowledge, many related works cover some of these aspects, but few of them address all three aspects.

The Swarm algorithm [14], which is well suited for biochemical pathway optimisation has been used in a distributed environment, e.g., in Grid Cellware [7], a grid-based modelling and simulation tool for the analysis of biological pathways that offers an integrated environment for several mathematical representations ranging from stochastic to deterministic algorithms.

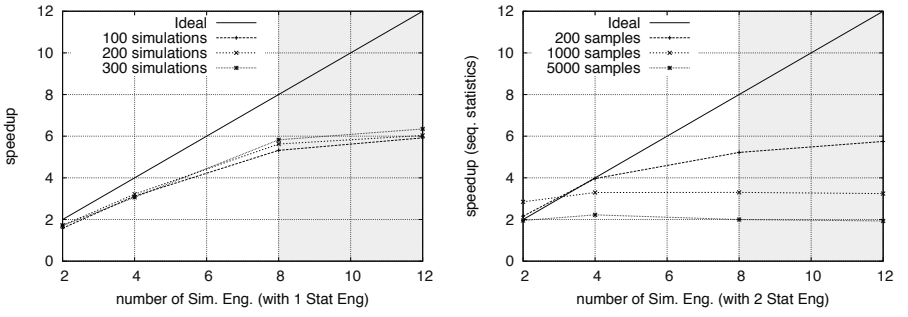


Fig. 4. Speedup on the stable switch simulation with 1 Statistic Engine for different number of parallel simulations and 200 samples (left), and with 2 Statistic Engines for different sampling rates and 200 simulations (right). The grey region delimits available platform parallelism (Intel x86_64 with 8 cores).

DiVinE is a general distributed verification environment meant to support the development of distributed enumerative model checking algorithms including probabilistic analysis features used for biological systems analysis [3].

StochKit [13] is a C++ stochastic simulation framework. Among other methods, it implements the Gillespie algorithm and in its second version it targets multi-core platforms, it is therefore similar to our work. It does not implement on-line trajectory reduction that is performed in a post-processing phase. A first form of on-line reduction of simulation trajectories has been experimented within StochKit-FF [1], which is an extension of StochKit using the FastFlow runtime.

StochSimGPU [12] exploits GPU for parallel stochastic simulations of biological systems. The tool allows to compute averages and histograms of the molecular populations across the sampled realizations on the GPU. The tool leverages on a GPU-accelerated version of the Matlab framework that can be hardly compared in flexibility and performance with a C++ implementation.

6 Conclusions

Starting from the Calculus of Wrapped Compartments and its parallel simulator we have discussed the problem of the analysis of stochastic simulation results, which can be complex to interpret also due to intrinsic stochastic "noise" and the overlapping of the many required experiments by the Monte Carlo method.

At this aim, we characterised some patterns of behaviour for biological system dynamics, e.g. monostable, multi-stable, and oscillatory systems, and we exemplified them with minimal yet paradigmatic examples from the literature. For these, we identified data filters able to provide statistically significative information to the biological scientists in order to simplify the data analysis.

Both the simulations and the on-line statistic filters, which are both parallel and pipelined, can be easily extended with new simulation algorithms and filters

thanks to FastFlow-based parallel infrastructure that exempt the programmer from synchronization and orchestration of concurrent activities.

Preliminary experiments demonstrated a fair speedup on a standard multi-core platform. We plan to further investigate the performance tuning of the simulation pipeline on larger problems and platforms.

Acknowledgements. We wish to thank Luca Cardelli for the inspiring talk and the discussion on multi-stable biological systems and switches, and Andrea Bracciali for the discussion on data filtering for biological simulations. We also thank M. Mazumder and E. Macchia of Etica Srl for the simulator GUI implementation.

References

1. Aldinucci, M., Bracciali, A., Liò, P., Sorathiya, A., Torquati, M.: StochKit-FF: Efficient Systems Biology on Multicore Architectures. In: Guarracino, M.R., Vivien, F., Träff, J.L., Cannataro, M., Danelutto, M., Hast, A., Perla, F., Knüpfer, A., Di Martino, B., Alexander, M. (eds.) Euro-Par-Workshop 2010. LNCS, vol. 6586, pp. 167–175. Springer, Heidelberg (2011)
2. Aldinucci, M., Coppo, M., Damiani, F., Drocco, M., Torquati, M., Troina, A.: On designing multicore-aware simulators for biological systems. In: Proc. of Intl. Euromicro PDP 2011: Parallel Distributed and Network-Based Processing, pp. 318–325. IEEE, Ayia Napa (2011)
3. Barnat, J., Brim, L., Safránek, D.: High-performance analysis of biological systems dynamics with the divine model checker. Briefings in Bioinformatics 11(3), 301–312 (2010)
4. Cardelli, L.: On switches and oscillators (2011), `http://lucacardelli.name`
5. Coppo, M., Damiani, F., Drocco, M., Grassi, E., Troina, A.: Stochastic Calculus of Wrapped Compatnents. In: QAPL 2010, vol. 28, pp. 82–98. EPTCS (2010)
6. CWC Simulator website (2010), `http://cwcsimulator.sourceforge.net/`
7. Dhar, P.K., et al.: Grid cellware: the first grid-enabled tool for modelling and simulating cellular processes. Bioinformatics 7, 1284–1287 (2005)
8. FastFlow website (2009), `http://mc-fastflow.sourceforge.net/`
9. Gillespie, D.: Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. 81, 2340–2361 (1977)
10. Hartigan, J., Wong, M.: A k-means clustering algorithm. Journal of the Royal Statistical Society C 28(1), 100–108 (1979)
11. Heyer, L., Kruglyak, S., Yooseph, S.: Exploring expression data: identification and analysis of coexpressed genes. Genome Research 9(11), 1106 (1999)
12. Klingbeil, G., Erban, R., Giles, M., Maini, P.: Stochsimgpu: parallel stochastic simulation for the systems biology toolbox 2 for matlab. Bioinformatics 27(8), 1170 (2011)
13. Petzold, L.: StochKit: stochastic simulation kit web page (2009), `http://www.engineering.ucsb.edu/~cse/StochKit/index.html`
14. Ray, T., Saini, P.: Engineering design optimization using a swarm with an intelligent information sharing among individuals. Eng. Opt. 33, 735–748 (2001)
15. Regev, A., Shapiro, E.: Cells as computation. Nature 419, 343 (2002)
16. Sciacca, E., Spinella, S., Genre, A., Calcagno, C.: Analysis of calcium spiking in plant root epidermis through cwc modeling. Electronic Notes in Theoretical Computer Science 277, 65–76 (2011)