Parallel Computing Platform for the Agent-Based Modeling of Multicellular Biological Systems

Toh Da-Jun, Francis Tang, Travis Lee, Deepak Sarda, Arun Krishnan, and Andrew Goryachev*

Bioinformatics Institute, Matrix L7, 30 Biopolis St, Singapore

Abstract. Agent-based simulation of large multicellular biological systems has become a viable option owing to affordable parallel computers, such as Beowulf-style clusters. We describe a scalable modular software platform that (i) provides for an easy integration of different solvers computing internal dynamics of the cells, and (ii) dynamically adapts to the changing loads on the cluster nodes using a load balancing algorithm. Simulations of systems of about 100,000 bacterial cells have been shown to be feasible using the platform.

1 Introduction

The overwhelming complexity of biological systems challenges our attempts to understand their function. The staggering variety of the "parts" that constitute biological systems and the highly nonlinear laws that define their interaction leave computational simulation as, perhaps, the only practical approach to the "understanding" of living matter. Steady improvements in computing hardware and recent advances in computational cell biology have given researchers powerful tools for simulation of intracellular metabolic, genetic and signal transduction networks [4]. Considerably less attention, however, has been given to the development of computational tools suitable for the biologically realistic simulation of systems comprised of many (100–100,000) interacting biological cells.

The necessity to model large ensembles of interacting cells commonly arises in the studies of disease pathology, cancer progression, embryonic development, wound healing and many other highly important processes involving creation and reorganization of biological tissue. A number of approaches based on the continuous representation of tissue with partial differential equations has been proposed in the literature [3]. They, however, do not explicitly address the discrete, cellular organization of the tissue and cannot be easily generalized to satisfy the needs of various applications.

In this work we present a computational platform capable of multipurpose simulation of large ensembles of interacting biological cells whose number varies

* Author to whom correspondence should be addressed.

K.-M. Liew et al. (Eds.): PDCAT 2004, LNCS 3320, pp. 5–8, 2004.

due to the processes of cell division and cell death. The natural parallelism of biological systems is captured by implementing the cells as largely independent agents whose intracellular dynamics is computed in a parallel way on a cluster-based hardware. We demonstrate that large multi-cellular ensembles of practical importance can be simulated in a reasonable time on contemporary cluster hardware.

2 Platform Design

The platform is designed for the Beowulf-style clusters, since they provide the scalability and computing power required for the simulation of thousands of interacting cells at affordable cost. We employ agent-based simulation paradigm with three types of agents: Cell, Medium and Master. We allocate one processor each for the Medium and the Master, while the cells are allocated to the remaining processors.

A Cell encapsulates a number of variables describing the environment of a typical biological cell, such as age, cell cycle phase, and concentrations of various chemical species. It also performs autonomous calculations of the intracellular dynamics of the metabolic, genetic and signal transduction networks. This function is performed by a dedicated third-party software, such as CellWare [2]. This core engine can be readily replaced to address the specific goals of the particular simulation problem. Based on their environment and the internal state variables, cells autonomously select predefined developmental programs, such as cell division or death. Cells do not communicate directly with each other, but only through interaction with the common Medium.

The Medium models the extracellular environment where the cells are allowed to move and with which they are chemically connected through the exchange of a subset of intracellular molecules. These chemicals are allowed to freely diffuse in the Medium and be "sensed" by other cells, thus enabling the cell-cell communication. Finally, the Master acts as the administrator that manages the system clock and schedules cooperation of the Medium and the Cells. It also maintains the dynamic cell collection by satisfying the requests of individual cells for division and termination, as well as migrates cells from node to node to balance their computational load.

2.1 Load Balancing and Scheduling

Since each cell independently proliferates or dies, some processors may end up with more cells than others (see Fig. 1, left). Also, since the underlying cluster is typically a shared resource, its load may depend on other users' jobs. We created a load balancing module that monitors in real time the node performance and determines where the new cells should be created as well as proposes cell migrations should changes in the node performance be detected (Fig. 1, right).

The load balancing is achieved through a list scheduling algorithm which uses the expected run-times of cells derived from the run-times observed in the previous time steps. To reduce the amount of statistics, similar cells are grouped

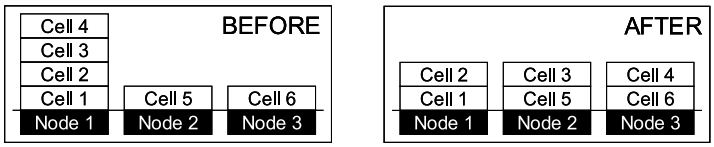

Fig. 1. Load Balancing. The load balancing algorithm dynamically redistributes the work

together. Assuming heterogeneity in the performance of processors, allows us (i) to model interference from other users' jobs, and (ii) to incorporate a cost penalty for migrations. In the case of heterogeneous processors we used a modified list scheduling algorithm based on that proposed by Davis and Jaffe [1]. The algorithm is known to give a solution at most $2.5\sqrt{m}$ times worse than the optimal for m processors.

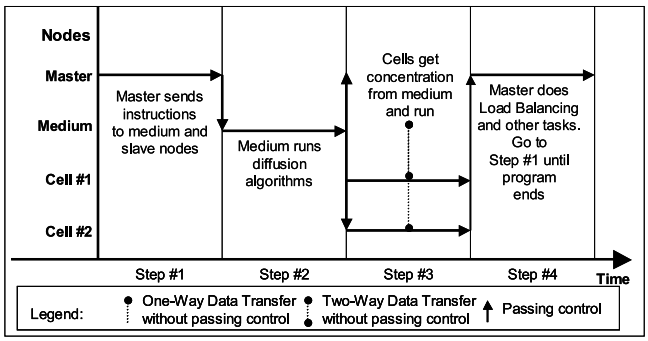

Fig. 2. Scheduling. For each time step, the Master instructs the Medium to run the diffusion algorithm. Then it instructs each cell to request the latest extracellular chemical concentrations from the Medium and run the intracellular simulation. Each cell may decide to divide, die or otherwise change its behavior, passing this decision to the Master. The Master creates, destructs and relocates the cells according to the load balancing method

In biological systems, the communication between the cells and the medium is continuous. By analogy with the Forward-Euler Method, we simulate this process by alternating the parallel intracellular dynamics of cells with the Medium communication step. Reduction of the time step length increases accuracy at the expense of the performance. The Master provides the necessary time step coordination (see Fig. 2).

2.2 Implementation and Results

The platform was implemented using a layered reference software architecture (see Fig. 3), which allows for easy upgrading of program functions, and task

allocation among developers. It was developed using the Intel C/C++ compiler and MPI, on a 32-node Myrinet-connected RedHat Linux cluster. Each node had two 1.4GHz Pentium III CPUs and 2Gb memory.

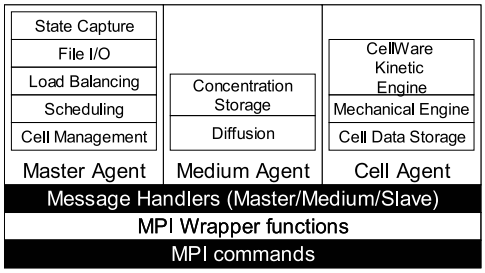

Fig. 3. Layered Reference Architecture

We have tested the platform by simulating a bacterial population in a two-dimensional medium. We used the Brownian motion model of bacterial movement, neglecting direct interactions between the cells. Cell-cell communication was modeled by assuming permeability of cell walls to certain diffusible molecules. Although the exponential population growth poses a significant computational challenge, we have performed simulations with several hundred thousand cells using as many as 22 nodes.

3 Conclusion

In this paper we have introduced a highly flexible parallel computational platform suitable for the multipurpose agent-based simulation of large multicellular ensembles of biological cells. On the example of a loosely chemically connected bacterial culture we demonstrated the feasibility of biologically realistic simulations involving hundreds of thousands of bacterial cells. We thus validated the potential of our platform to address complex biological problems and will further improve its simulation realism and computational efficiency in the future work.

References

1. E. David and J. M. Jaffe. Algorithms for scheduling tasks on unrelated processors. *J. ACM*, 28:721–736, 1981.
2. P. Dhar, Tan C. M., S. Somani, Li Y., K. Sakharkar, A. Krishnan, A. B. M. Ridwan, M. Chitre, and Zhu H. Grid Cellware: The first Grid-enabled tool for modeling and simulating cellular processes. *Bioinformatics*, 2004.
3. J. D. Murray. *Mathematical biology II: Spatial models and biomedical applications.* Springer-Verlag, 2003.
4. B. Slepchenko et al. Computational cell biology: Spatiotemporal simulation of cellular events. *Ann. Rev. Biophys. Biomol. Struct.*, 31:423–441, 2002.