

# Development of Distributed Cell Simulations Middleware with E-Cell System

Masahiro Sugimoto<sup>1,2</sup>  
msugi@sfc.keio.ac.jp

Kouichi Takahashi<sup>1</sup>  
shafi@e-cell.org

Daniel<sup>3</sup>  
p0217127@sp.edu.sg

Muhammad Nurazhar Bin Taher<sup>3</sup>  
p0216551@sp.edu.sg

Lu Yaomin<sup>3</sup>  
p0216056@sp.edu.sg

Novia Siswanto Be<sup>3</sup>  
p0256939@sp.edu.sg

Tracy Wong<sup>3</sup>  
p0215998@sp.edu.sg

Ng Yan Hong<sup>3</sup>  
Y.Ng@sp.edu.sg

Masaru Tomita<sup>1</sup>  
mt@sfc.keio.ac.jp

<sup>1</sup> Institute for Advanced Biosciences, Keio University, Tsuruoka, Yamagata, 997-0811, Japan

<sup>2</sup> Bioinformatics Department, Mitsubishi Space Software Co. Ltd., Amagasaki, Hyogo, 661-0001, Japan

<sup>3</sup> Singapore Polytechnic, School of Info-Communications Technology, 139651, Singapore

**Keywords:** distributed computing, cell simulation, E-Cell, SMP, SGE, globus, Genetic Algorithm

## 1 Introduction

E-Cell Project[1,2] is an international research project aiming to model and reconstruct biological phenomena *in silico*, and developing necessary theoretical supports, technologies and software platforms to allow precise whole cell simulation. We have developed a Python scripting environment that can transparently execute, and where possible, automatically parallelize those mathematical analysis sessions on any of single-processor workstations, Shared-memory Multiprocessor servers, workstation clusters and computational grid environments. This computational framework is called E-Cell SessionManager (ESM)[3] built upon E-Cell System. Here we introduce the ESM architecture and show results from benchmark experiments. We also developed a front-end GUI of genetic algorithm (GA) as an application of ESM, which enable users to easily conduct parameter estimation, setup configuration and access the simulated results.

## 2 Methods and Results

### 2.1 Design of E-Cell Session Manager

The fundamental design of ESM is comprised of three classes, the SessionManager-, SessionProxy-, and SystemProxy class. The SessionManager class provides the users with a basic API to create and run simulation sessions. SystemProxy generates SessionProxy instances and communicates to the computing environment. SessionProxy corresponds to a job on distributed environment (Globus[4] on grid or SGE[5] on cluster) or a process on local PC and SMP. Users can script using flat API of SessionManager, and also describe procedures in object-oriented programming style (Figure 1). This flexible scripting environment provides a benefit that users easily create their own application on the distributed middleware.

### 2.2 GUI of E-Cell Session Manager

The objective of the project is to design a GUI for Session Manager to support users to setup the procedure of GA and ESM, provide running status such as searched parameters, and display simulated time-courses.

### 2.3 Benchmark

We executed benchmark experiments that addressed 2 typical computationally intensive biological problems, a parameter estimation session of a small hypothetical metabolic pathway and simulations of a stochastic *E. coli* heat-shock model with different random number seeds to obtain the statistical characteristics of the

stochastic fluctuations. The results of the numerical experiments presented in the previous section clearly show the benefits of using distributed computation in computational cell biology.

```

MODEL_FILE = 'model.eml'
ESS_FILE = 'runsession.py'

aJobIDList = []
for VALUE_OF_S in range(0,100):
    aParameterDict = {'MODEL_FILE': MODEL_FILE,
                     'VALUE_OF_S':VALUE_OF_S}
    aJobID = registerEcellSession(ESS_FILE,
                                 aParameterDict, [MODEL_FILE,])
    aJobIDList.append(aJobID )

run()

for aJobID in aJobIDList:
    print getStdout(aJobID)

```

Figure 1: An example of ESM script. 100 E-Cell jobs with different parameters are registered and executed in this script file.

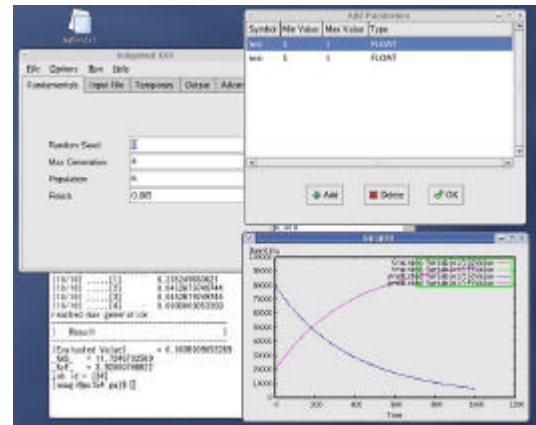


Figure 2: A screenshot of ESM GUI. Configuration of GA and parameter space to be searched is setup in main window and “add parameter” window. The training- and simulated time-courses are displayed in gnuplot window.

## Acknowledgments

This work was supported by a grant from the Ministry of Education, Culture, Sports, Science and Technology, a Grant-in-aid for the 21st century Center of Excellence (COE) program entitled ‘Understanding and control of life’s function via systems biology’ (Keio University), a grant from the New Energy and Industrial Technology Development and Organization (NEDO) of the Ministry of Economy, Trade and Industry of Japan (Development of a Technological Infrastructure for Industrial Bioprocess Project), a grant from the Leading Project for Biosimulation, Ministry of Education, Culture, Sports, Science and Technology, and a grant from the Japan Science and Technology Agency (JST).

## References

- [1] Takahashi, K., Sakurada, T., Kaizu, K., Kitamaya, T., Arjunam, S., Ishida, T., Berezcki, G., Ito, D., Sugimoto, M., Komori, T., Seiji, O., and Tomita, M., E-CELL System Version 3: A Software Platform for Integrative Computational Biology. *Genome Informatics 14* 294-295, 2003
- [2] Tomita, M., Hashimoto, K., Takahashi, K., Shimizu, T., Matsuzaki, Y., Miyoshi, F., Satio, K. Tanida, S., Yugi, K., Venter, J.C. and Hutchison, C, E-Cell: software environment for whole cell simulation. *Bioinformatics*, 15(1):72-84, 1999
- [3] Sugimoto, M., Takahashi, K., Kitayama, T., Ito, D. and Tomita, M, Distributed Computing Support of E-Cell System, *Proc. 1st International workshop on Life Science Grid*. 17-28, 2004
- [4] Foster, I. and Kesselman, C., Globus: a metacomputing infrastructure toolkit, *Int. J. Supercomput. Appl*, 11(2):115-128, 1997
- [5] <http://www.sun.com/software/gridware/>