

# CSIM 19 User Story

## Performance Prophet: Simulation of Parallel and Distributed Computing Systems using CSIM 19



By Sabri Pllana, Institute for Software Science,  
University of Vienna

---

### **Introduction**

High performance computing is often used to solve complex problems in science and engineering. But developing high-performance parallel and distributed programs can be time-consuming, error-prone, and expensive. Effective performance-oriented program development requires the programmer to understand the intricate details of the programming model, the parallel and distributed computer architecture, and the mapping of programs onto computer architectures.

We needed to build a tool that lets programmers develop and evaluate the performance model of a program at an early development stage. This early insight helps to improve design decisions and avoid time-consuming modifications of the code of an already implemented program.

Our vision was to support program development and performance analysis via graphical modeling and simulation – a parameterized simulation tool called *Performance Prophet*.

### **The Simulation Tool**

Performance Prophet is a performance prediction tool for parallel and distributed computing systems. Our approach was to reduce the time needed to evaluate the model, by simplifying the model and by combining mathematical modeling and discrete event simulation.

The model simplification is achieved by using grouping and neglecting techniques. For instance, in the process of building a model for a scientific program, several program statements are grouped and considered as a single element of the program model. Likewise, program

**Application Area:**  
*High Performance  
Computing*

**Platform:**  
*Sun – Solaris,  
PC – LINUX*

**CSIM's Challenge:**  
*Simulate the  
performance behavior  
of parallel and  
distributed programs  
on cluster and grid  
architectures*

statements that do not strongly influence the performance are neglected.

Figure 1 depicts the architecture of Performance Prophet. In order to develop the model of program, we employ our Java-based graphical editor Teuta, which is an integral part of Performance Prophet. We use Teuta to graphically develop the performance model of a program in the form of a Unified Modeling Language (UML) activity diagram; the program model is composed of existing building blocks based on the UML. Thereafter, Teuta transforms the program model from UML into a C++ representation. The CSIM-based simulator can now evaluate the performance of the program on the computer architecture selected by user.

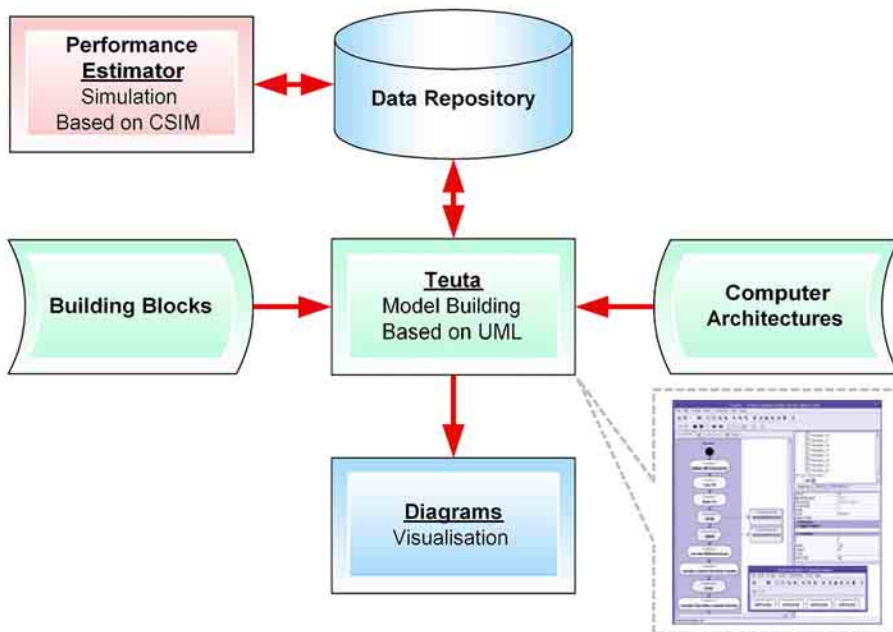


Figure 1: Performance Prophet Architecture

Modeling elements, which make up the program model and computer architecture model, are implemented as C++ classes based on CSIM. During the model-building phase, instances (i.e. objects) of these classes are used to create the parallel and distributed computing system model.

CSIM serves as the simulation engine for the Performance Estimator. During the simulation run, an object simulates the behavior of a component of the computing system over time. Furthermore, objects may interact with each other in order to simulate the behavior of the whole system.

### ***Our Biggest Simulation Hurdle:***

*Determining a level of abstraction for the computing system model that would be sufficiently accurate, but fast enough for evaluation by simulation.*

### Specific Requirements for CSIM

Our aim was to build high-level performance models of distributed and parallel computing systems. These performance models should offer comparative accuracy, since we usually use them to compare various parallelization strategies of large scientific programs. The time spent evaluating these models should be short, in order to explore a large set of possible parallelization strategies within a reasonable time. Therefore, the performance of the simulation engine was one of the key requirements. CSIM was able to provide the rapid execution that our application required.

A particular benefit of using CSIM was the ease of mapping CSIM concepts, such as process, facility, storage, event, and mailbox, onto elements of the simulation models of the computing systems. CSIM provided the right level of abstraction to allow straightforward mapping, which substantially reduced the development time.

### Results with CSIM

The key result of our simulation tool is the execution time of a program on a given computer architecture. This information helps us to evaluate the architecture and to improve the program under assessment.

For illustration purposes, we present simulation results for the program *LAPW0*. The Linearized Augmented Plane Wave (LAPW) method is among the most accurate methods for performing electronic structure calculations for crystals. The code of *LAPW0* program was written using FORTRAN90 and the Message Passing Interface (MPI). The *LAPW0* program consists of 100 file modules; a module is a file that contains the source code.

We carried out a model-based performance analyzes of *LAPW0* program, by following these steps:

- We developed a high-level, graphical performance model with UML in Teuta
- Teuta transformed the program model from UML into a C++ representation
- The CSIM-based simulator evaluated program performance



In this way, we can experiment with the model rather than with the real program.

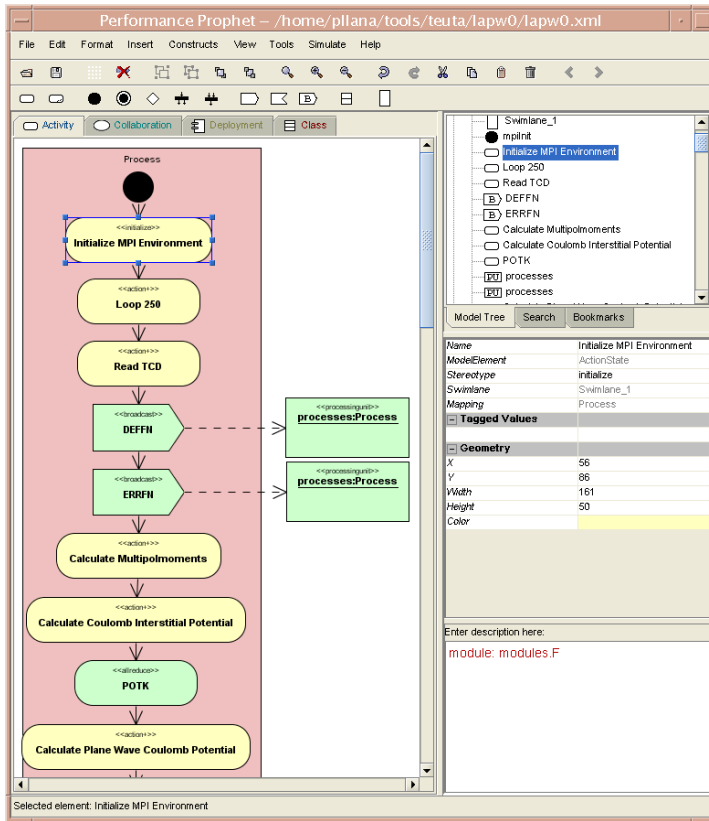


Figure 2(a)

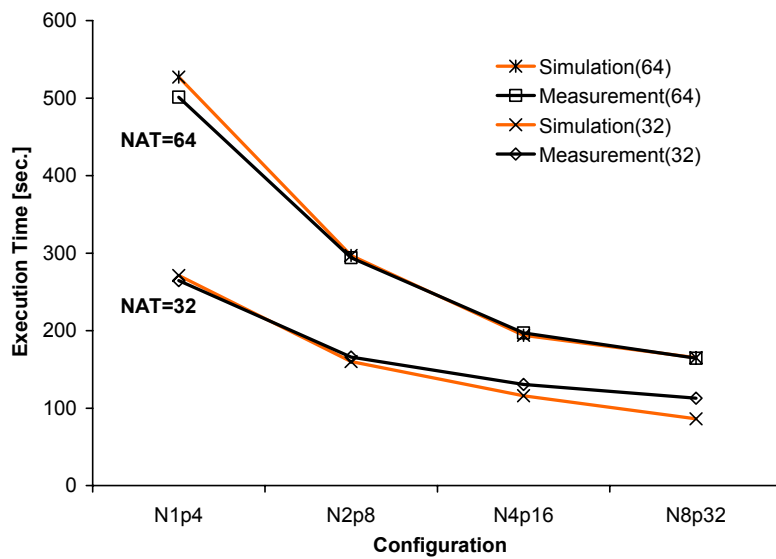


Figure 2(b)

Figure 2: The performance modeling and evaluation of LAPW0

Figure 2(a) depicts the performance model of LAPW0. The simulation and measurement results for two problem sizes and four machine sizes are presented in Figure 2(b). The problem size is determined by the parameter NAT, which represents the number of atoms in a unit of the material. The number of nodes of the cluster architecture determines the machine size. Each node of the cluster has four CPUs. One process of the LAPW0 program is mapped onto one CPU of the cluster architecture. We validated the simulation model by comparing the simulation results with the measurement results, and found that our simulation model provides performance prediction results with sufficient accuracy to compare various designs of the program LAPW0.



### **Conclusion**

System simulation is a domain where the saying “a little knowledge is a dangerous thing” is particularly true. Therefore, we wanted to base our research work on simulation software that comes from a knowledge-based company. This year, Mesquite Software is celebrating its 10<sup>th</sup> anniversary, but for over three decades, its members have contributed to research in system modeling and simulation, education of young simulation experts, and to the profession of simulation. Their excellent research record is supported by publications in many conferences and refereed scientific journals. We felt very secure trusting our simulation needs to CSIM.

You can find more details about Performance Prophet at <http://www.par.univie.ac.at/project/prophet/>

For more information about CSIM 19, please contact Mesquite Software:

Mesquite Software  
8500 N. MOPAC Expwy, Suite 825  
Austin, TX 78759 USA  
Phone: +1(512) 338-9153 or 1(800) 538-9158  
Email: [info@mesquite.com](mailto:info@mesquite.com)  
WWW: <http://www.mesquite.com>