

HuskySim: A Simulation Toolkit for Application Scheduling in Computational Grids

Mohamed Kerasha
University of Connecticut
Computer Science and Engineering
369 Fairfield Road, Storrs CT 06269-2155
mohamed@engr.uconn.edu

Ian Greenshields
University of Connecticut
Computer Science and Engineering
369 Fairfield Road, Storrs CT 06269-2155
ian@engr.uconn.edu

Categories & Subject Descriptors: Add here

General Terms: Add here

Keywords: Discrete Event Simulation, Computational Grids, Adaptive Scheduling, Performance prediction.

ABSTRACT

Grid computing-- the assemblage of heterogeneous distributed clusters of computers viewed as a single virtual machine-- promises to serve as the next major paradigm in distributed computing. Since Grids are assemblages of (usually) autonomous systems (autonomous clusters, supercomputers, or even single workstations) scheduling can become a complex affair which must take into consideration not just the requirements (and scheduling decisions) made at the point of the job's origin, but also the scheduling requirements (and decisions) made at remote points on the fabric, and in particular scheduling decisions made by a remote autonomous system onto which the local job has been scheduled. The current existing scheduling models range from static, where each of the programs is assigned once to a processor before execution of the program commences, to dynamic, where a program may be reassigned to different processors, or a hybrid approach, which combines characteristics of both techniques [1,4,5].

To address this issue, we have developed a JAVA based discrete event Grid simulator toolkit called HuskySim. The HuskySim toolkit provides core functionalities (e.g., compute objects, network objects, and scheduling objects) that can be used to simulate a distributed computing environment. Furthermore, it can be used to predict the performance of various classes of Grid scheduling algorithms including:

- Static scheduling algorithms
- Dynamic scheduling
- Adaptive Scheduling

In our design, we adopted an object-oriented design, which allows an easy mapping and integration of simulation objects into the simulation program. This approach simplifies the simulation of multitasking, and distributed data processing

model. Our model of multitasking processing is based on an interrupt driven mechanism.

As shown in Figure 1, the simulator works by relaying messages between the core engine and the simulation modules through the message handling sub-system. Once the architecture, the load distribution, and the scheduling algorithms are defined, the object registration subsystem sends a NEW OBJECT REQUEST MESSAGE to the object class libraries and builds a skeleton for the requested simulation experiment.

Workloads traces can be generated using probabilistic models. The currently supported distributions are: Uniform, Poisson, Exponential, Normal, Erlang, and Power Tailed. It is also possible to use real world load traces. Moreover, we augmented the Simulator with a statistical module. Using the statistical module provided with the HuskySim, the core simulation engine can send messages to perform various type of analysis on the performance data including: variance reduction, regression, time series analysis, clustering, and data mining.

In order to quantify the system performance, the simulator provides various performance metrics including: CPU utilization, disk utilization, application turnaround time, latency, make span, host to host bandwidth, jammed bandwidth, and TCP/IP traffic data. These measurements are handled through the measurement sub-system.

Furthermore, the HuskySim can be used to simulate the classes of algorithmic and parametric adaptive Grid schedulers. In which, the scheduling algorithm may not be fixed in advance. Simply, the scheduling algorithm is selected at run time based on the current workload on the Grid fabric in order to operate at near optimal level.

CONCLUSION

This paper has described the basic architecture of the HuskySim toolkit, currently under development at the University of Connecticut. The simulator allows fast comparisons of Grid scheduling algorithms in realistic conditions.

HUSKYSIM

GRID Distributed Discrete Event Scheduling Simulator

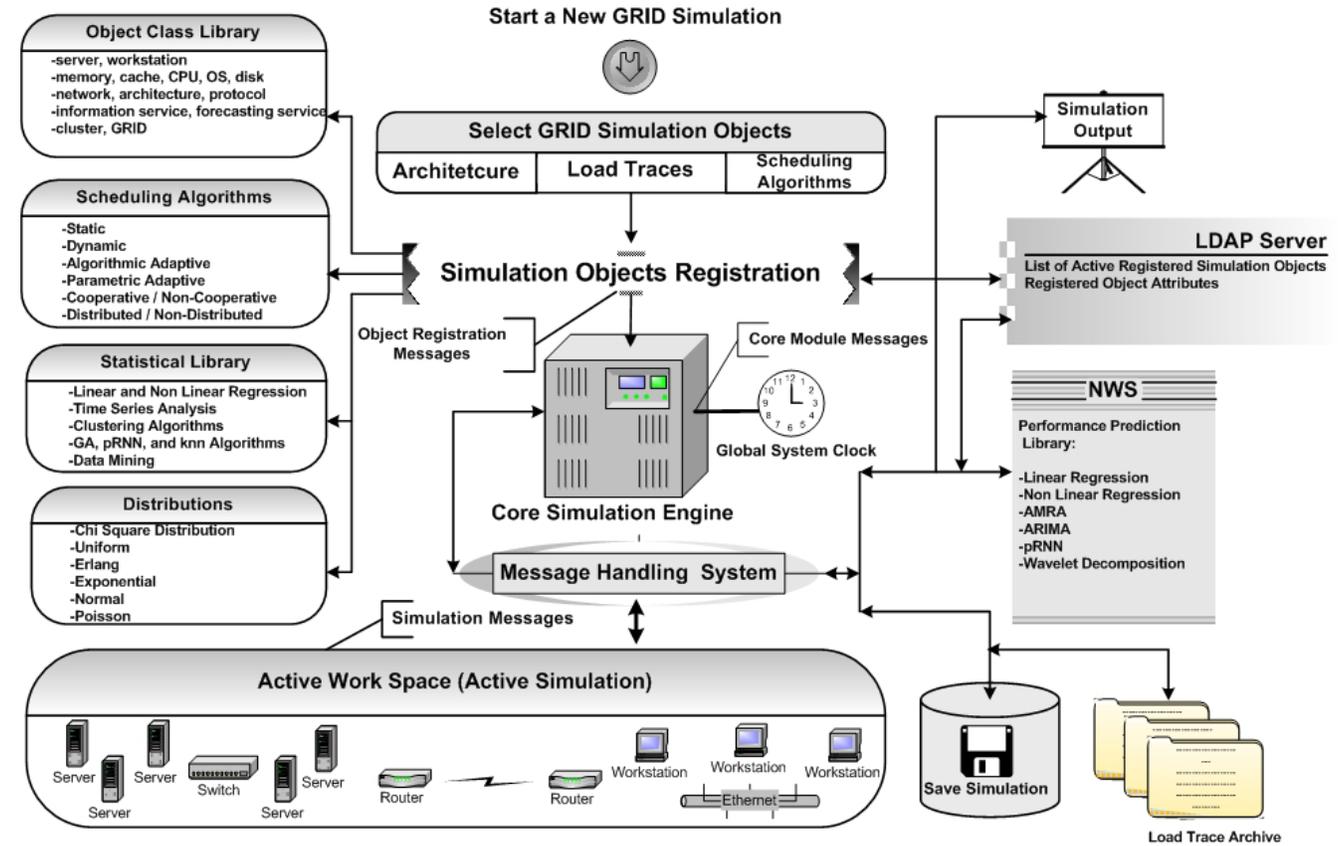


Figure 1. Architecture of the HUSKY Grid Simulator

REFERENCES

1. H. Casanova. Simgrid: A toolkit for the simulation of application scheduling. In Proceedings of the IEEE Symposium on Cluster Computing and the Grid CCGrid'01). IEEE Computer Society, May 2001.
2. Homi Bodhanwala Luis. A General Purpose Discrete Event Simulator. Symposium on Performance Evaluation of Computer and Telecommunication Systems, Orlando, USA , July 2001
3. Paul Bratley: A Guide to Simulation. Springer-Verlag
4. Heath A. James, Scheduling in Metacomputing Systems, Ph. D. Thesis, Department of Computer Science, University of Adelaide, Adelaide Australia, 1999.
5. Dongmin Kim et al., Virtual Computing Concept Design and Evaluation, Kluwer Academic Publishers, 2001
6. Marcio Faerman, Alan Su, Rich Wolski and Francine Berman, Adaptive Performance Prediction for Distributed Data-Intensive Applications, Proceedings of the ACM/ IEEE Supercomputing '99 Conference on High Performance Networking and Computing, Portland Oregon, 13-18 November 1999
7. Globus Project, <http://www.globus.org>
8. Portable Batch System (PBS), <http://pbs.mrj.com/>