Modeling Exascale Applications with SST/macro and Eiger

April 29, 2013

Half-day tutorial, held previously at ISPASS 2013.

- Eric Anger[1] <eanger@gatech.edu>, Georgia Institute of Technology
- Gilbert Hendry[1] <ghendry@sandia.gov>, Sandia National Laboratories
- Benjamin Allan <baallan@sandia.gov>, Sandia National Laboratories
- Sudhakar Yalamanchili <sudha@gatech.edu>, Georgia Institute of Technology

1 Abstract

In high performance computing (HPC), the importance of fast, large scale models of high fidelity are only increasing as we move towards the next frontier of exascale. Hardware/software codesign is viewed as a key methodology to reaching this end. The SST/macro toolkit[1] provides HPC engineers the ability to explore current and future hardware/software design constraints. Instead of costly (in time and user effort) cycle-accurate simulation, macro-scale simulation can provide valuable insight into the performance of large applications. The value of these tools lies in high quality application models for increasingly complex hardware designs.

The Eiger Performance Modeling Framework[2] generates models by applying statistical techniques from the field of machine learning on empirical performance data. While macro-scale simulation can provide a reasonable overview of system wide phenomena, Eiger can leverage data acquired from micro-scale sources to *inform* large scale simulations in SST/macro. Eiger provides an API and data store for aggregating data from micro-scale sources such as simulators, emulators, and runtime instrumentation.

This tutorial will present attendees with the techniques and methodologies to leverage SST/macro and Eiger for modelling large scale applications on upcoming supercomputer hardware. This presentation is geared towards domain experts and HPC hardware designers, as well as students and researchers whose work requires exploration of programming models, interaction between computation and communication, and data-driven modelling techniques for large scale systems. These tools are geared toward ease of use and rapid iteration, allowing area experts to generate verbose performance models without requiring intricate knowledge of every facet of the computing environment. This tutorial will require only a basic level of programming skill.

2 Outline

- Introduction [30 minutes]
  - Principals of codesign and motivating macro-scale
  - Application skeletons as the vehicle for modeling
  - Overview of SST/macro and Eiger
- SST/Macro [1 hour 15 minutes]
  - Goals, functionality, and use of SST/macro
  - Building and running the simulator
  - User interface and GUI
  - Validation
  - Visualization
  - Examples

[1] Tutorial presenter.

- Short Break [30 minutes]
- Eiger [30 minutes]
  - Instrumentation, analysis, and modeling with Eiger
  - Creating new transformation passes
  - Integration with other tools
  - Validation
  - Examples
- Putting SST/macro and Eiger together[45 minutes]
  - Adding instrumentation with the lwperf tool
  - Polling Eiger models in SST/macro
  - Visualization tools
  - Examples
- **Total Time: 3 hours 30 minutes**

3 Organizer Biographies

Eric Anger is a Ph.D student with the Computer Architecture and Systems Lab at Georgia Tech. His research focus is on the interactions between large scale applications and hardware, with particular focus on accelerator technologies like GPUs. His current work explores statistical methods for performance analysis and prediction.

Gilbert Hendry received a B.S. and M.S. in Computer Engineering from the Rochester Institute of Technology, and a Ph.D in Electrical Engineering from Columbia University. His dissertation focused on network design involving silicon nanophotonics for network-on-chip applications, as well as design automation techniques and software development in this area. He is currently a staff member in the Scalable Modeling and Analysis Systems group at Sandia National Labs in Livermore, CA ,and is the PI and technical lead for the SST/macro simulator project. Gilbert has over 20 publications in conferences and journals in the area of computer architecture and network simulation.

Benjamin Allan is a Principal staff member at Sandia National Laboratories conducting research in HPC and cyber security topics. He has authored and presented multiple tutorials on scientific computing software tools in international forums such as the annual Supercomputing Conference, PARA 2008, and DOE ACTS Workshops. He develops SSTmacro's compute modeling and validation capability.

Sudhakar Yalamanchili earned his Ph.D degree in Electrical and Computer Engineering in 1984 from the University of Texas at Austin. He was at Honeywells Systems and Research Center in Minneapolis before joining the ECE faculty at Georgia Tech in 1989 where he is now a Joseph M. Pettit Professor of Computer Engineering. His current research foci lie in addressing the software challenges of heterogeneous architectures and solutions to power and thermal issues in many core architectures and data centers. Since 2003 he has been a Co-Director of the NSF Industry University Cooperative Research Center on Experimental Computer Systems at Georgia Tech.

References

[1] Curtis L. Janssen, Helgi Adalsteinsson, Scott Cranford, Joseph P. Kenny, Ali Pinar, David A. Evensky, and Jackson Mayo. A simulator for large-scale parallel computer architectures. *IJDST*, 1(2):57–73, 2010.

[2] Andrew Kerr, Eric Anger, Gilber Hendry, and Sudhakar Yalamanchili. Eiger: A framework for the automated synthesis of statistical performance models. In *High Performance Computing*, 2012.