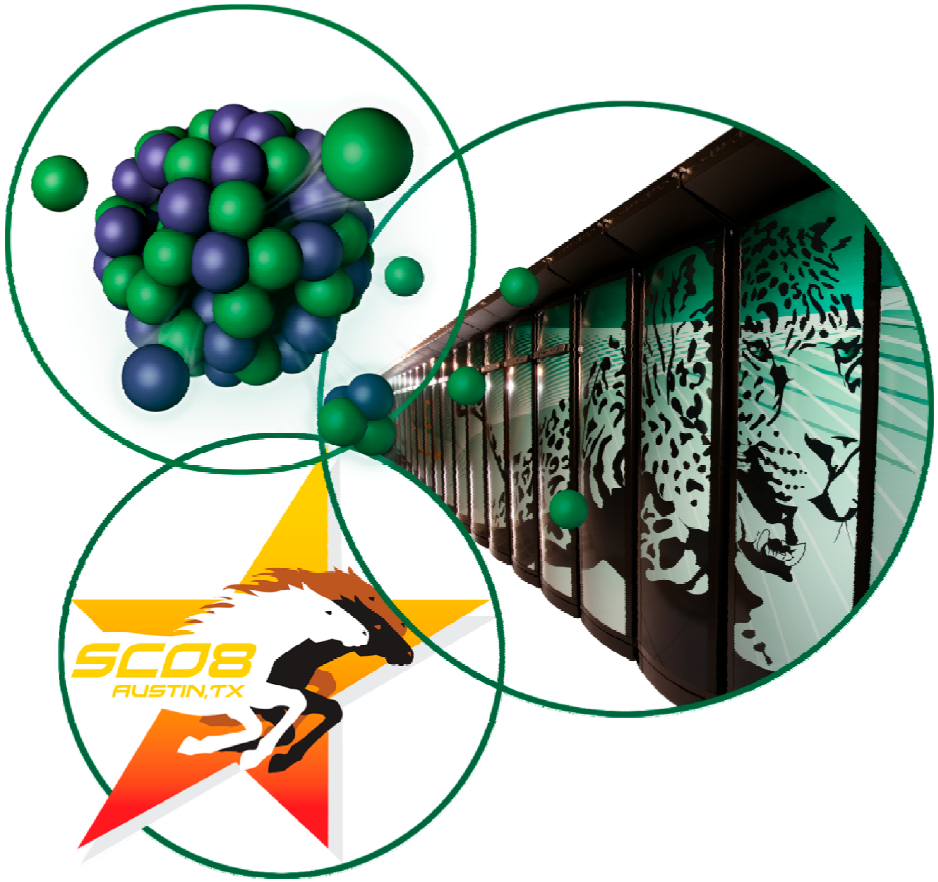
Modeling Assertions: Symbolic Models for Petascale Applications and Platforms



Presented by

Jeffrey S. Vetter (PI),
Sadaf R. Alam,
Heike Jagode,
Charles Lively

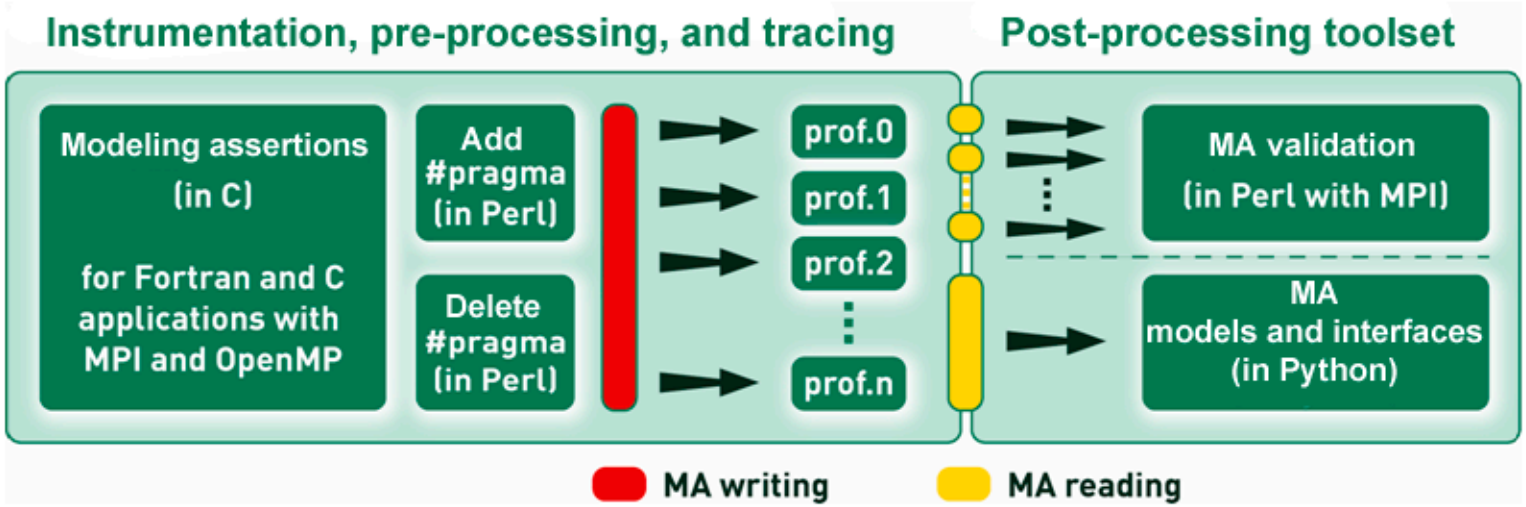
Future Technologies Group
Computer Science and Mathematics Division

Modeling assertions (MA)

- **What?**
  - Portable and extensible workload and performance modeling framework
- **Why?**
  - Provide an integrated solution to develop, validate, and experiment with symbolic performance models of large-scale applications
  - Provide an efficient mechanism for sensitivity analysis and workload project growth rates for future problem and system configurations
- **Components**
  - A portable API for code annotation
  - Pre- and post-processing tools
  - Synthetic MPI trace generation with OTF compatible format

Distributed architecture of MA (1/2)

- **MA instrumentation API**
  - Source code annotations to describe computation, communication and memory operations behavior
  - Annotations interpreted by the runtime system and recorded in trace files for post-processing tools
- **MA PRE-processor**
  - Insert and delete pragma statements in front of MA function calls
  - Facilitate model generation and development processes



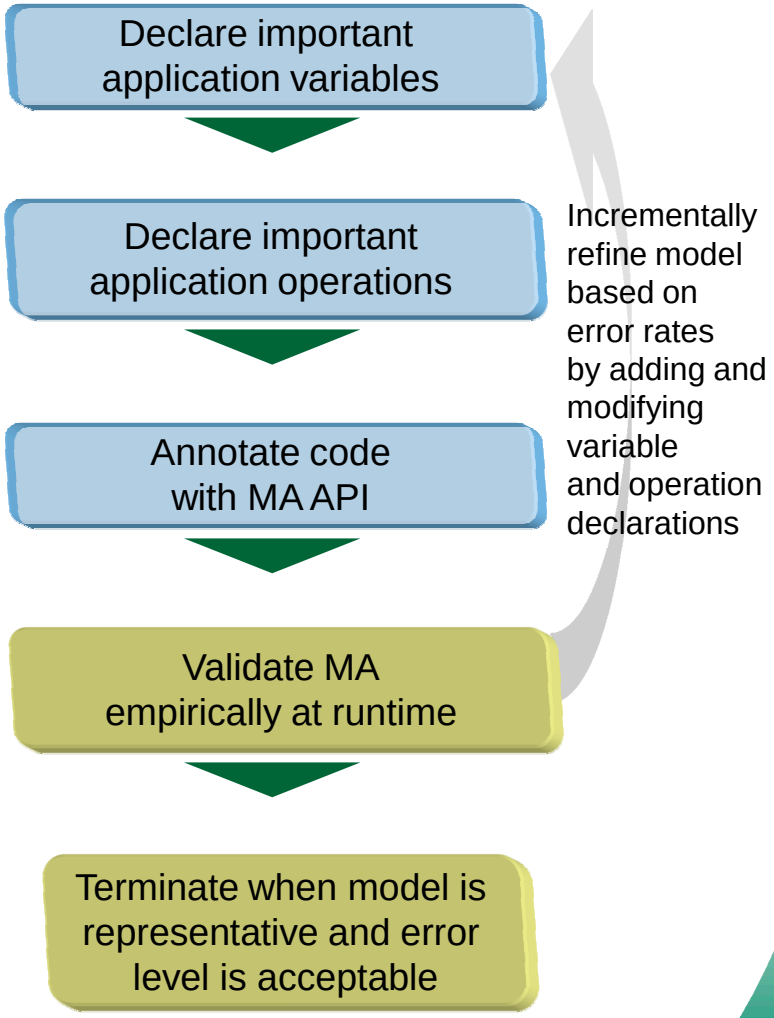
Distributed architecture of MA (2/2)

- **MA POST-processor: Validation**
  - Implements parallelized validation algorithms to enable fast validity check of complex performance modeling data
  - Large data volumes can be verified because of close interaction with state-of-the-art parallel production environments

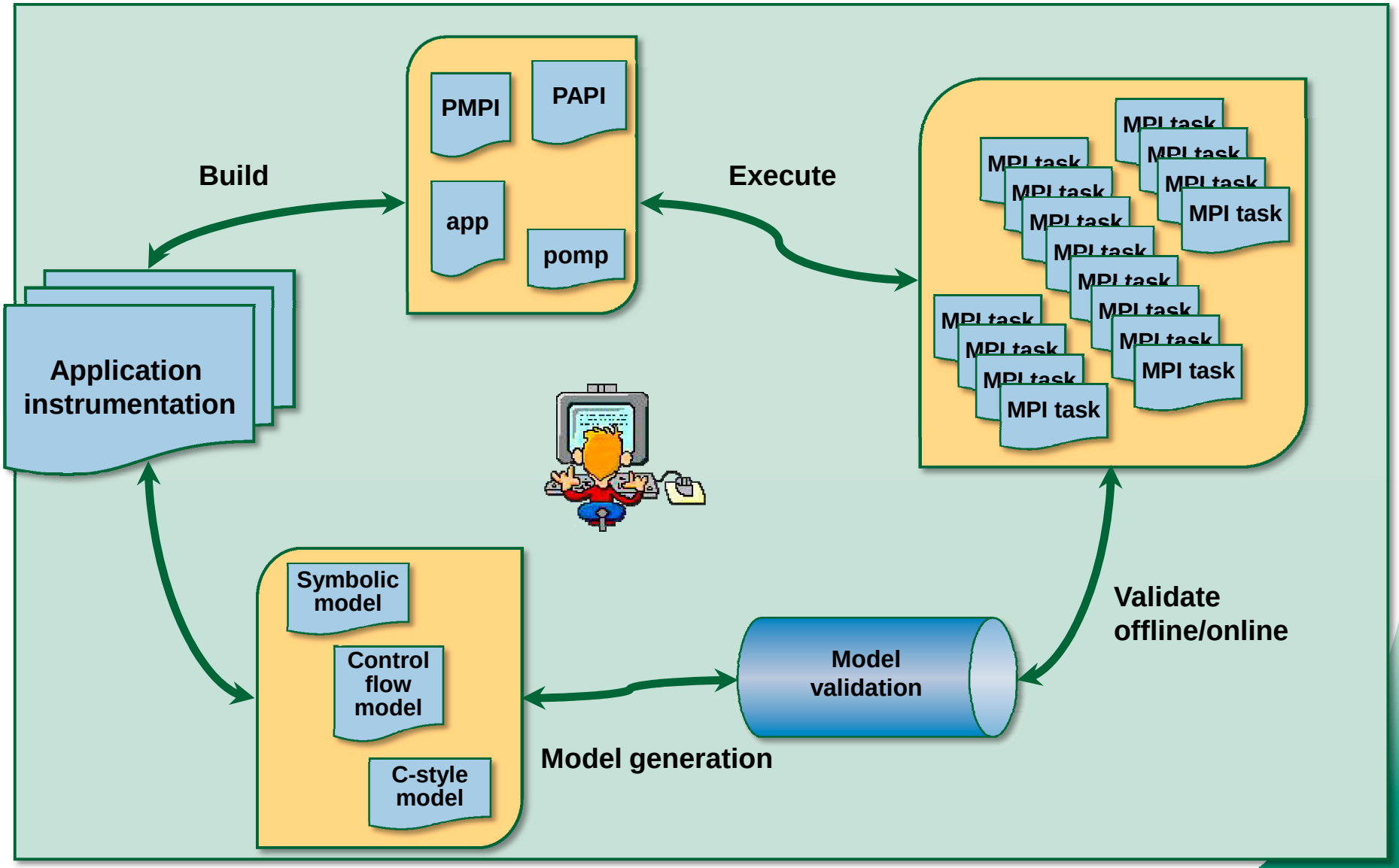
- **MA POST-processor: Models**
  - Model generation in alternate formats, for instance, in control-flow and symbolic notation
  - MA models to project future workload requirements
    - To generate synthetic traces
    - To perform sensitivity analysis

MA framework

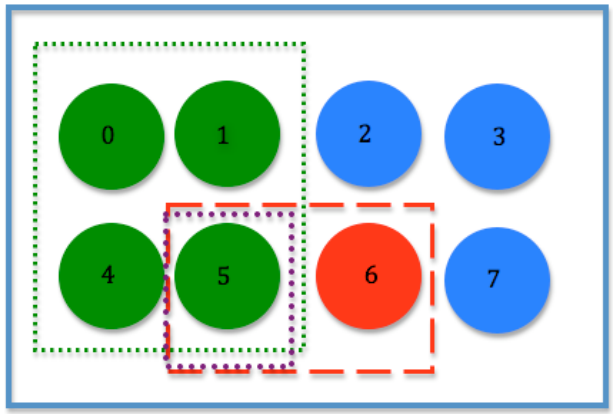
- **Main components**
  - **API**
  - **Model development tools**
  - **Model analysis tools**
- **Main features**
  - **Online validation, analysis, and prediction**
  - **Pragma-based instrumentation**
  - **Realtime, efficient error reporting mechanisms**



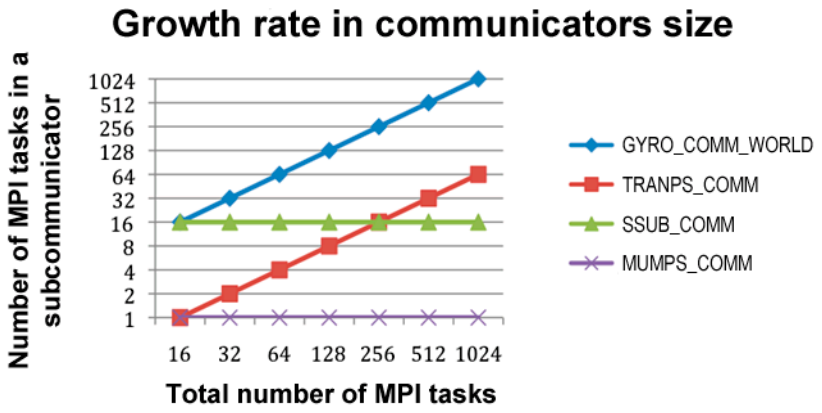
Modeling workflow using MA

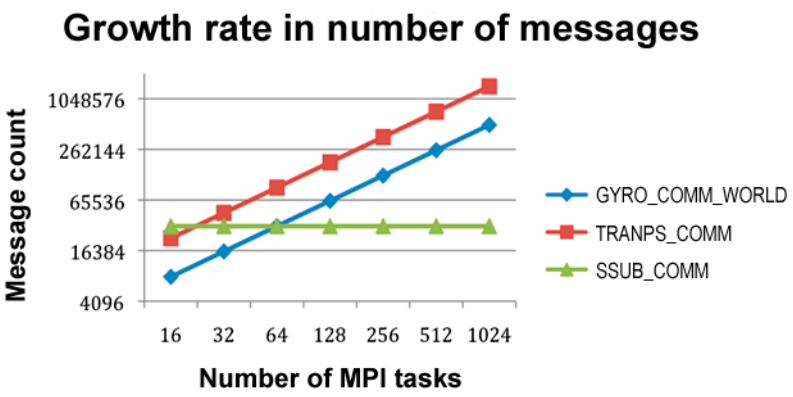


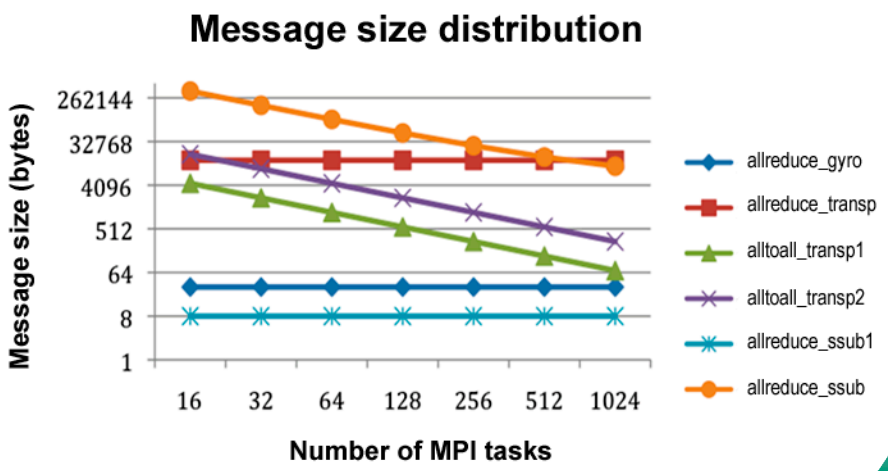
Hierarchical modeling of MPI communication for multi-core systems



GYRO sub-communicator structure (3 communicator sizes)

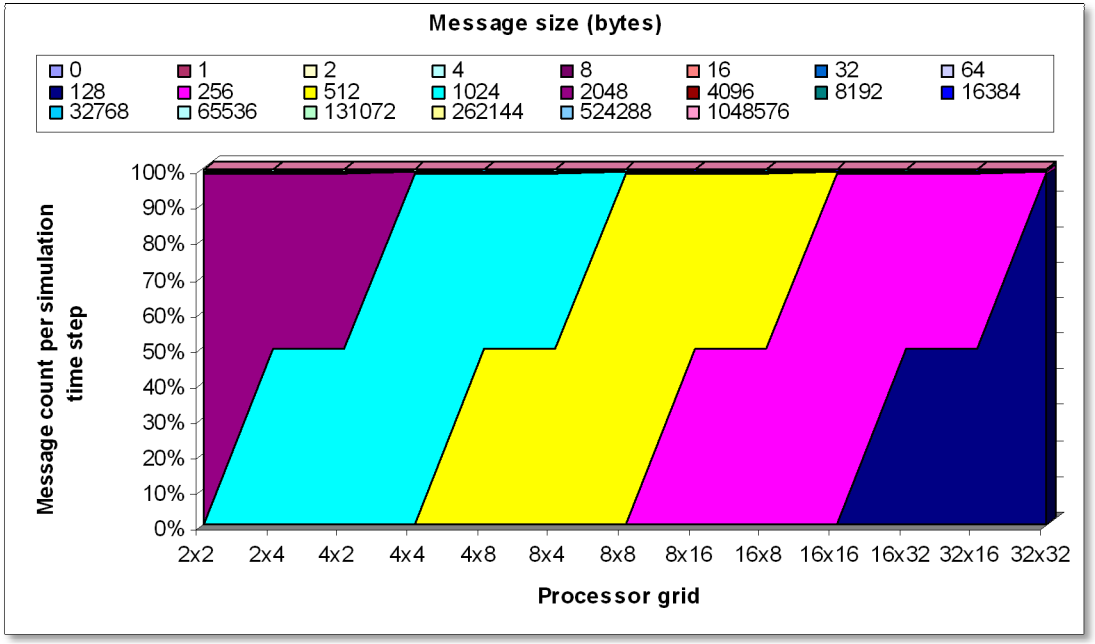






Sensitivity analysis and workload growth rate projections

Goal: Identify workload sensitivity at scale before system and application deployment



Message size distribution in the key calculation phases as the MPI grid topology is scaled

Performance and scaling bottlenecks due to

- **Application parameters**
- **System parameters**

Synthetic trace file generation

Goal: Generate input traces for future problem and system configurations that do not exist

- **Prototype for future network design at scale infeasible**
  - **Issues**
    - **Realistic and representative workload for petascale and exascale systems**
    - **Problem configurations; for example, input decks for petascale and exascale problems do not exist**
- **Solution**
  - **Generate parameterized network models and traces to drive network simulators**
  - **Support common MPI trace file formats like OTF**
  - **Identify workload scaling behavior and patterns**
  - **Investigate alternate algorithms and implementation**

Contacts

Jeffrey S. Vetter

Principle Investigator
Future Technologies Group
Computer Science and Mathematics Division
(865) 356-1649
vetter@ornl.gov

Sadaf R. Alam

865) 241-1533
alamsr@ornl.gov

Heike Jagode

(865) 974-8057
jagode@eecs.utk.edu