Scalasca Tracing Tools
Scalable performance analysis of large-scale parallel applications

Features

- Localization of wait states & their root causes on large processor configurations
- Identification of the critical path
- Support for MPI, OpenMP, and hybrid MPI/OpenMP
- Based on the community-driven instrumentation & measurement infrastructure Score-P
- Uses open data formats OTF2 and CUBE4

Supported Platforms

- Cray XT/XE/XK/XC
- IBM Blue Gene
- IBM SP & Blade clusters
- Linux-based clusters (x86, Power, ARM)
- Tianhe-1A & 2
- SGI Altix (incl. ICE + UV)
- Fujitsu FX / K computer
- Intel Xeon Phi (native mode only)

www.scalasca.org
Scalasca Measurement & Analysis Workflow

1. **Run Score-P instrumented target application to produce runtime summary**
   - Provides initial insight into the application's run-time behavior
   - Allows optimizing the configuration for subsequent measurements (e.g., filtering of uncritical code regions, estimation of trace buffer requirements, etc.)

2. **Generate targeted event traces of critical code regions for closer investigation of concurrent behavior**
   - Automatic event trace analysis at the end of measurement searching for inefficiency patterns, wait states, and the critical path (using a parallel analysis tool to achieve scalability, executed as part of the same batch job)

3. **Examine trace analysis results using an intuitive graphical user interface (Cube)**

Scalable Automatic Wait-state & Root-cause Analysis

- **Replay-based trace analysis searches for wait states**
- **Attribution of short-term and long-term costs to identify delays as root causes of wait states**
- **Classification of wait states as propagating or terminal to assess inter-wait state influences**

Scalable Critical-path Analysis

- **Determines the critical path of the application in a scalable fashion, to help identify**
  - Program activities for which optimization will prove worthwhile
  - Parallelization bottlenecks such as load imbalance and serial execution

**Diagram**

1. Score-P measurement library
2. Summary report
3. Optimized measurement configuration
4. Local event traces
5. Parallel trace analysis
6. Trace analysis report
7. Cube report browser

**Scalasca Measurement & Analysis Workflow Diagram**

- Instr. target application
- Score-P measurement library
- Local event traces
- Parallel trace analysis
- Trace analysis report
- Summary report
- Cube report browser

**Scalasca Measurement & Analysis Workflow**

1. Run Score-P instrumented target application to produce runtime summary
   - Provides initial insight into the application's run-time behavior
   - Allows optimizing the configuration for subsequent measurements
   (e.g., filtering of uncritical code regions, estimation of trace buffer requirements, etc.)

2. Generate targeted event traces of critical code regions for closer investigation of concurrent behavior
   - Automatic event trace analysis at the end of measurement searching for inefficiency patterns, wait states, and the critical path (using a parallel analysis tool to achieve scalability, executed as part of the same batch job)

3. Examine trace analysis results using an intuitive graphical user interface (Cube)

**Scalable Automatic Wait-state & Root-cause Analysis**

- **Replay-based trace analysis searches for wait states**
- **Attribution of short-term and long-term costs to identify delays as root causes of wait states**
- **Classification of wait states as propagating or terminal to assess inter-wait state influences**

**Scalable Critical-path Analysis**

- **Determines the critical path of the application in a scalable fashion, to help identify**
  - Program activities for which optimization will prove worthwhile
  - Parallelization bottlenecks such as load imbalance and serial execution