Performance Analysis with TAU

Allen Malony
University of Oregon
Dept. of Computer and Information Science
Eugene, Oregon, USA
malony@cs.uoregon.edu

KOJAK ↔ TAU ↔ VAMPIR

KOJAK - OTF

TAU - TRACE

KOJAK

TAU - EPILOG

TAU - PROFILE

OTF / VTF3 trace

VAMPIR

TAU trace

VampirTrace <= 4.0

CUBE profile

CUBE Presenter

EPILOG trace

EXPERT Analyzer

gprof / mpiP profile

PerfDMF

PARAPROF

covered now

© 2006 Allen Malony, University of Oregon
TAU Analysis and Visualization

- Architecture
  - Bridges to other tools
- Parallel Profiling
- Parallel Tracing
- Parallel performance data management
- Parallel performance data mining
- Example
- Demonstration

TAU Analysis and Visualization Architecture
### Building Bridges to Other Tools

- **Analysis of parallel profile and trace measurement**
- **Parallel profile analysis**
  - Pprof: parallel profile analysis, ASCII presentation
  - ParaProf: parallel profile analysis, graphical presentation
  - ParaVis: parallel performance visualization package
  - Profile generation from trace data (tau2pprof)
- **Performance data management framework (PerfDMF)**
- **Parallel trace analysis**
  - Translation to VTF (V3.0), EPILOG, OTF formats
  - Integration with VNG (Technical University of Dresden)
- **Online parallel analysis and visualization**
- **Integration with CUBE browser (KOJAK, UTK, FZJ)**

---

**Performance Analysis and Visualization**

- **Analysis of parallel profile and trace measurement**
- **Parallel profile analysis**
  - Pprof: parallel profile analysis, ASCII presentation
  - ParaProf: parallel profile analysis, graphical presentation
  - ParaVis: parallel performance visualization package
  - Profile generation from trace data (tau2pprof)
- **Performance data management framework (PerfDMF)**
- **Parallel trace analysis**
  - Translation to VTF (V3.0), EPILOG, OTF formats
  - Integration with VNG (Technical University of Dresden)
- **Online parallel analysis and visualization**
- **Integration with CUBE browser (KOJAK, UTK, FZJ)**
Example Applications

- NAS parallel benchmarks (LU, BT)
  - Fortran, MPI
- Miranda
  - research hydrodynamics code, Fortran, MPI
- FLASH
  - physics simulation, Fortran, MPI
- sPPM
  - ASCI benchmark, Fortran, C, MPI, OpenMP or pthreads
- GYRO
  - tokomak turbulence simulation, Fortran, MPI
- WRF
  - weather research and forecasting, Fortran, MPI
- S3D
  - 3D combustion, Fortran, MPI
- Uintah
  - Large-grained dataflow system, component-based, C++

Pprof – Flat Profile (NAS PB LU)

- Intel Linux cluster
- F90 + MPICH
- Profile
  - Node
  - Context
  - Thread
- Events
  - Code
  - MPI
- Metric
  - time
- Text display
ParaProf – Graphical Parallel Profile Analysis

ParaProf – Flat Profile (Miranda)

8K processors!

Miranda
- hydrodynamics
- Fortran + MPI
- LLNL

Run to 64K

© 2006 Allen Malony, University of Oregon
ParaProf – Stacked View (Miranda)

ParaProf – Histogram View (Miranda)

MPI_Alltoall()

MPI_Barrier()

8k processors

16k processors
ParaProf – 3D Full Profile (Miranda)

16k processors

ParaProf – 3D Scatterplot (Miranda)

- Each point is a "thread" of execution
- A total of four metrics shown in relation
- ParaVis 3D profile visualization library
  - JOGL
ParaProf – Flat Profile (NAS BT)

Application routine names reflect phase semantics

How is MPI_Wait() distributed relative to solver direction?

ParaProf – Phase Profile (NAS BT)

Main phase shows nested phases and immediate events
ParaProf – Callpath Profile (Flash)

ParaProf – 3D Full Profile (Flash)

© 2006 Allen Malony, University of Oregon
ParaProf – Callgraph Zoomed (Flash)

ParaProf – Statistics Table (Uintah)
### Vampir - Trace Analysis (TAU-to-VTF3) (S3D)

<table>
<thead>
<tr>
<th>Process</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>1</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>2</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>3</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>4</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>5</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>6</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>7</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>8</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>9</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>10</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>11</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>12</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>13</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>14</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>15</td>
<td>VAMPIR</td>
</tr>
</tbody>
</table>

**3D combustion**

**Fortran + MPI**

**PSC**

---

### Vampir - Trace Zoomed (S3D)

<table>
<thead>
<tr>
<th>Process</th>
<th>Methodology</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>1</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>2</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>3</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>4</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>5</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>6</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>7</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>8</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>9</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>10</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>11</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>12</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>13</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>14</td>
<td>VAMPIR</td>
</tr>
<tr>
<td>15</td>
<td>VAMPIR</td>
</tr>
</tbody>
</table>

**S3D**

© 2006 Allen Malony, University of Oregon
Performance Data Management (PerfDMF)

TAU Performance System

- raw profiles
- gprof
- mpiP
- psrun
- HPXtoolkit
- ...

profile metadata

Query and Analysis Toolkit

- scalability analysis
- ParaProf analysis
- cluster analysis

Performance Analysis Programs

- Java PerfDMF API
- SQL (PostgreSQL, MySQL, DB2, Oracle)
- ...

Data Mining (Weka)

Statistics (R / Omega)

Performance Data Mining (Objectives)

- Conduct parallel performance analysis process
  - In a systematic, collaborative and reusable manner
  - Manage performance complexity
  - Discover performance relationship and properties
  - Automate process
- Multi-experiment performance analysis
- Large-scale performance data reduction
  - Summarize characteristics of large processor runs
- Implement extensible analysis framework
  - Abstraction / automation of data mining operations
  - Interface to existing analysis and data mining tools
Performance Data Mining (PerfExplorer)

- Performance knowledge discovery framework
  - Data mining analysis applied to parallel performance data
    - comparative, clustering, correlation, dimension reduction, ...
  - Use the existing TAU infrastructure
    - TAU performance profiles, PerfDMF
- Client-server based system architecture
- Technology integration
  - Java API and toolkit for portability
  - PerfDMF
  - R-project/Omegahat, Octave/Matlab statistical analysis
  - WEKA data mining package
  - JFreeChart for visualization, vector output (EPS, SVG)
PerfExplorer - Analysis Methods

- Data summaries, distributions, scatter plots
- Clustering
  - k-means
  - Hierarchical
- Correlation analysis
- Dimension reduction
  - PCA
  - Random linear projection
  - Thresholds
- Comparative analysis
- Data management views

PerfExplorer - Cluster Analysis

- Performance data represented as vectors - each dimension is the cumulative time for an event
- k-means: k random centers are selected and instances are grouped with the “closest” (Euclidean) center
- New centers are calculated and the process repeated until stabilization or max iterations
- Dimension reduction necessary for meaningful results
- Virtual topology, summaries constructed
PerfExplorer - Cluster Analysis (sPPM)

- Four significant events automatically selected (from 16K processors)
- Clusters and correlations are visible
**PerfExplorer - Correlation Analysis (Flash)**

- Describes strength and direction of a linear relationship between two variables (events) in the data

![PerfExplorer Correlation Analysis Flash]

---

**PerfExplorer - Comparative Analysis**

- Relative speedup, efficiency
  - total runtime, by event, one event, by phase
- Breakdown of total runtime
- Group fraction of total runtime
- Correlating events to total runtime
- Timesteps per second
- Performance Evaluation Research Center (PERC)
  - PERC tools study (led by ORNL, Pat Worley)
  - In-depth performance analysis of select applications
  - Evaluation performance analysis requirements
  - Test tool functionality and ease of use
PerfExplorer - Interface

Select experiments and trials of interest

Data organized in application, experiment, trial structure (will allow arbitrary in future)

Select analysis

Experiment metadata
Cray X1 is the fastest to solution
   □ In all 3 tests
FFT (nl2) improves time
   □ B3-gtc only
TeraGrid faster than p690
   □ For B1-std?
All plots generated automatically

PerfExplorer - Relative Efficiency (B1-std)

By experiment (B1-std)
   □ Total runtime (Cheetah (red))
By event for one experiment
   □ Coll_tr (blue) is significant
By experiment for one event
   □ Shows how Coll_tr behaves for all experiments

16 processor base case
TAU Performance System Status

- Computing platforms
  - IBM, SGI, Cray, HP, Sun, Hitachi, NEC, Linux clusters, Apple, Windows, ...
- Programming languages
  - C, C++, Fortran 90/95, UPC, HPF, Java, OpenMP, Python
- Thread libraries
  - pthreads, SGI sproc, Java, Windows, OpenMP
- Communications libraries
  - MPI-1/2, PVM, shmem, ...
- Compilers
  - IBM, Intel, PGI, GNU, Fujitsu, Sun, NAG, Microsoft, SGI, Cray, HP, NEC, Absoft, Lahey, PathScale, Open64, ...

Support Acknowledgements

- Department of Energy (DOE)
  - Office of Science
    - MICS, Argonne National Lab
  - ASC/NNSA
    - University of Utah ASC/NNSA Level 1
    - ASC/NNSA, Lawrence Livermore National Lab
- Department of Defense (DoD)
  - HPC Modernization Office (HPCMO)
  - Programming Environment and Training (PET)
- NSF Software and Tools for High-End Computing
- Research Centre Juelich
- Los Alamos National Laboratory
- ParaTools
TAU

http://www.cs.uoregon.edu/research/tau/
tau-bugs@cs.uoregon.edu