TAU : 591C Report

Project Team

Sibin Mohan, Salil Pant, Harini Ramaprasad

{ smohan, smpant, hramapr }@unity.ncsu.edu

Our Project

Our project involves:

- ***Installation*** *(on the i32/Linux platform)*
- ***Demonstration*** *(using test programs like LU decomposition, RSA encryption)*
- ***Evaluation*** *(using the ASCI benchmark* **ParBenCCh***.)*

of ***TAU****.*

Tuning and Analysis Utilities(TAU)

(from: http://www.cs.uoregon.edu/research/paracomp/tau/tauprofile/docs/installation.html)

TAU (Tuning and Analysis Utilities) is a portable profiling and tracing toolkit for performance analysis of parallel programs written in C++, C, and Fortran. The model that TAU uses to profile parallel, multi-threaded programs, maintains performance data for each thread, context, and node in use by an application. The profiling instrumentation needed to implement the model captures data for C++ functions, methods, basic blocks, and statement execution at these levels. All C++ language features are supported in the TAU profiling instrumentation including templates and namespaces, which is available through an API at the library or application level. The API also provides selection of profiling groups for organizing and controlling instrumentation. The instrumentation can be inserted in the source code using an automatic instrumentation tool based on the Program Database Toolkit, or manually using the instrumentation API.

TAU's profile visualization tool, racy, provides graphical displays of all the performance analysis results, in aggregate and single node/context/thread forms. The user can quickly identify sources of performance bottlenecks in the application using the graphical interface. In addition, TAU can generate event traces that can be displayed with the ***Vampir*** *trace visualization tool.*

ParBenCCh

(from: http://www.llnl.gov/asci/purple/benchmarks/limited/parbencch/parbencch.readme.html)

The ParBenCCh suite is a collection of small C and C++ applications designed to characterize compiler optimization capabilities, language support, Object Oriented programming style overhead, and machine performance. A testing framework has been developed using a virtual base class that encapsulates the essential functionality of any benchmark. Each specific test derives from this base class and contains the code unique to that test. This common interface makes creating a new benchmark straightforward and gives us identically formatted output files containing timing information that can then be easily and automatically processed.
The tests currently available in the ParBenCCh suite are :

- *The* ***Haney*** *test.*
- *The* ***Stepanov*** *test.*
- *The* ***Blitz*** *test.*
- *The* ***OpenMP*** *test.*
- *The tests in the* ***IndirectAddressing*** *directory.*

Project Objectives

- *Installation of TAU on i32/Linux machines.*
- *Test the TAU toolkit using existing sample programs (such as LU Decomposition, RSA encryption, etc.)*
- *Analyse results from TAU and improve performance of sample programs.*
- *Install the ParBenCCh Benchmark suite.*
- *Analyse results from executing the ParBenCCh Suite, and see if any improvements are possible for TAU.*

References

[1] http://www.cs.uoregon.edu/research/paracomp/tau/

[2] http://www.llnl.gov/asci/purple/benchmarks/limited/parbencch/

[3] http://www.cs.uoregon.edu/research/paracomp/tau/taudemo/

[4] A. Malony and S. Shende, Performance Technology for Complex Parallel and Distributed Systems, Proc. Third Austrian-Hungarian Workshop on Distributed and Parallel Systems, DAPSYS 2000, "Distributed and Parallel Systems: From Concepts to Applications," (Eds. G. Kotsis and P. Kacsuk) Kluwer, Norwell, MA, pp. 37-46, 2000.

[4] S. Shende, and A. D. Malony, Integration and Application of the TAU Performance System in Parallel Java Environments Proceedings of the Joint ACM Java Grande - ISCOPE 2001 Conference, June 2001.

[5] T. Sheehan, A. Malony, S. Shende, A Runtime Monitoring Framework for the TAU Profiling System, Proceedings of the Third International Symposium on Computing in Object-Oriented Parallel Environments (ISCOPE'99), San Francisco, CA, December 1999.