Performance Analysis and Tuning with TAU

Kevin James Edwards

kjedward@ncsu.edu

CSC 548 – Parallel Systems

North Carolina State University

Dr. Frank Mueller

1. Introduction

TAU is a performance analysis tool that can be used to for assisting in creating more efficient code. One can use the tools provided with TAU to help determine where inefficient portions of code may lie so that they can be analyzed and strengthened. TAU also has the capability to analyze parallel applications that use OpenMP. It also is able to be extended to work with programs using the MPI standard.

Since parallel applications must be especially focused on efficiency, TAU and similar tools can be extremely important for identifying portions of applications that are negatively affecting runtime. Parallel applications in particular can be difficult to assess and determine where weaknesses lie. So with the help of TAU and other tools, one can determine if particular techniques can be used to minimize runtime.

2. Description

I plan on using TAU to first determine which inefficiencies are present in some benchmarks commonly used to test parallel architectures. I will focus on benchmarks that use OpenMP for threading among processors. I may also use other tools for analysis such as mpiP which will help me find any areas for improvement in regards to MPI benchmarks. After finding potential problem areas, I will attempt to alleviate or improve these weaknesses. The goal is to improve overall runtime of one or two particular benchmarks by determining weak points in these benchmarks using analysis tools and attacking these faults with techniques for efficiency improvement.

3. Approach

I will first need to install TAU and link it with some potential benchmarks to work out any kinks in getting the tool working correctly on our architecture. I must then focus on finding potential problem areas in benchmarks. Luckily some previous research has been done in this area and inefficiencies have already been found in some benchmarks. Therefore I must use TAU on these benchmarks to make sure that I am seeing these inefficiencies myself. Then the problem becomes alleviating some of these inefficiencies using techniques we have learned in class. Again, there has been some work done in this area so I will likely be able to use others' work to assist in this portion of the project.

4. Metrics

The most important metric that I will be optimizing for will be overall runtime. This will be the focus of my research, reducing runtime. However, decreasing runtime will be in all cases a function of some other metric. Many times it is items such as communication or cache usage that are the biggest factors in runtime. TAU will be able to assist me in determining which of these factors is influencing runtime the most. So if for a particular benchmark, there exists a lot of cache misses, my goal will then turn to reducing these misses in order to increase efficiency and reducing runtime.

5. Conclusion

This project will allow me to gain more experience with performance analysis tools in reference to parallel applications. I will be using TAU and mpiP to determine areas of interest in parallel benchmarks and attempt to optimize them. Through optimizing these benchmarks, I will gain important experience and potentially gain the knowledge of how important efficiency can be.

6. References

Tuning and Analysis Utilities (TAU), http://www.cs.uoregon.edu/research/tau/home.php, 2006

Samit Jain ,Matthew Crocker, Nasim Mahmood and James C. Browne. Productivity and Performance Through Components: The ASCI Sweep3D Application

Jaydeep Marathe, Anita Nagarajan, Frank Mueller. Detailed Cache Coherence Characterization for OpenMP Benchmarks. ACM. 2004.

ASCI Purple Benchmarks. http://www.llnl.gov/asci/purple/benchmarks/limited/code_list.html, 2006