Abstract

Parallel computing is rapidly evolving to include heterogeneous collections of distributed and parallel systems. Concurrently, applications are becoming increasingly multidisciplinary with code libraries implemented using diverse programming models. To optimize the behavior of complex applications on heterogeneous systems performance analysis software must also evolve, replacing post-mortem analysis with real-time adaptive optimization tightly integrating compile-time analysis with performance measurement and prediction and supporting intuitive visualization and software manipulation.

Presently, there are few techniques for predicting application performance from first principles. Instead, they must exploit experimental techniques, making performance analysis subject to the same constraints as other experimental sciences. Furthermore, performance tools must be simple and intuitive. Unless compelled by circumstances, most users are unwilling to invest great time and effort to learn the syntax and semantics of new performance tools; they often view performance optimization as an unavoidable evil. Hence, portability and ease of use are critical to the acceptance of new performance tools. Simply put, the goal of experimental performance analysis is to provide insight into application behavior and performance bottlenecks by efficiently capturing and intuitively presenting performance data.

This paper presents PET (Program Execution Tree), a framework for analyzing the executions of parallel programs and intuitively visualizing their behavior on large parallel systems, built on top of a new instrumentation technique.