

Parallel Software Performance Metrics as Continuous Functions of Event Times

S.D. Prestwich

Abstract

In software development, a metric is the measurement of some characteristic of a program's performance. When developing software for parallel architectures, metrics can play a very useful role in tuning properties such as task granularity and load balancing. A current approach to parallel software development is the use of analysis or visualisation tools to reveal important aspects of the parallel execution, often by post-execution analysis of trace files. For example, it may be useful to detect the creation of a large number of fine-grained tasks, which cause significant runtime overheads. A simple metric to estimate this danger is the number of parallel tasks created per second. This may be computed by dividing the trace into time slots, and counting the number of created tasks in each slot. Such metrics require parallel events like task creation and completion to be assigned time stamps. However, precise times are hard to obtain in asynchronous parallel systems. The difficulty is exacerbated by the well-known "probe effect" whereby the act of monitoring performance affects the performance itself. This inaccuracy may render a metric meaningless if it relies on the order in which events occur, or on the precise duration of tasks. Another danger is that the choice of time slot length may create artifacts, which becomes obvious if a visualisation tool allows the user to zoom in on parts of the trace file. The designer of parallel performance metrics must therefore take great care to ensure that they are meaningful.

This paper argues that small inaccuracies in event times should not produce large effects on metrics, and hence on numbers, graphs, pictures or animations produced by analysis or visualisation tools. This is not sufficient to guarantee good metrics but it is a useful necessary condition. The requirement that small changes have small effects is characteristic of continuous functions, and it is therefore proposed that metrics be defined as continuous functions of event times. To bridge the gap between discrete events and continuous functions, metrics can be defined as integrals (over time slots) of simpler functions called trace abstractions. A trace abstraction need not be continuous in time: it may be a simple step functions that changes value when an event occurs. Such functions can easily be integrated over time slots, and a trace abstraction obeying certain conditions yields a metric that is insensitive to event time inaccuracy. Practical sufficient conditions are provided for such trace abstractions, and rules provided for the composition of complex trace abstractions from simple ones. As a bonus, continuous metrics are shown to be insensitive to time slot length and hence to be well-behaved under changes in time scale.
