

# Low Impact Profiling for Fun, Performance Modelling and Prophecy

J.A. Smith

## Abstract

The accuracy of predictions from performance modelling activities is governed by the efficacy of the performance analysis. The performance analysis, in turn, depends on the profiling tools used to perform the analysis. Current profiling tools vary in methodology, terms of data collected, runtime impact, accuracy and available analyses. In some cases, these concerns can be mitigated by averaging the results of a number of different runs. Such functionality is often built into profilers, but such duplicated runs are time consuming and, on chargeable systems, expensive. This situation is exacerbated when features from more than one profiler is required, further increasing the number of runs required.

And, despite the number of profilers available, few are targeted towards performance modelling, most are designed around the needs of application developers rather than the cross of debugger and profiler desirable for the activity of performance modelling. The issues with using a profiler intended for the application developer are well illustrated by an examination of the venerable Unix `gprof` profiler, itself an improvement on the Unix `profil()` function call. The performance modeller requires the ability to examine the behaviour of the application over time, whilst the requirements of the application developer are often satisfied by a summary of the behaviour.

The major problem when collecting data to allow analysis of behaviour over time is that recording a trace, rather than building a summary, can be expected to have a larger runtime impact, particularly as the amount of data collected is increased. Research is therefore focused on three problems; what data is required, how to collect that data and how to minimise the effect of recording that data. Problems to be addressed by the research are therefore; the technical issues of extracting the data efficiently, reducing the overheads of recording that data and on whether it is possible to record a more summarised version of a run and recover sufficient runtime behaviour from that.