

UNIVERSITÀ DELLA SVIZZERA ITALIANA

Data Analysis for Performance Annotations

PROJECT PLAN

Irene Jacob

irene.jacob@usi.ch

SUPERVISED BY:

Professor Antonio Carzaniga

Daniele Rogora

March 2, 2017

1. Motivation

Performance problems in large software systems can be difficult to resolve. In particular, identifying a performance issue in the source code of a large system can be an arduous process, with little guarantee of success. The idea of performance annotations is to provide valuable information regarding the performance characteristics of a piece of code. For example, a performance annotation might relate the running time or space complexity of a function to, say, the size of an input data structure. This information can then be used to optimize a system. With proper analysis, the data received from these annotations could reveal the parts of the system causing performance problems.

2. Goal

There are a number of potential uses of performance annotations. Namely:

1. They can be used as performance assertions, to ensure that a particular function follows its performance requirements. In particular, an annotation may serve as an oracle for tests to detect performance problems.
2. Annotations may also provide useful information in debugging performance problems. For example, a performance annotation may provide information about the memory usage of a function. This information could reveal a possible memory leak, which may otherwise have been difficult to identify by simply examining the code.
3. They can also be seen as a design tool, where systems are designed and implemented to meet certain performance requirements. The annotations of a method, combined with the structure of the code of the system in which this method is used, might allow the developer to deduce the overall performance of the system.

The goal of this project is to derive performance annotations from the execution of a system. In essence, this amounts to recording and analyzing relevant data logged by instrumenting a subject system.

3. Project Description

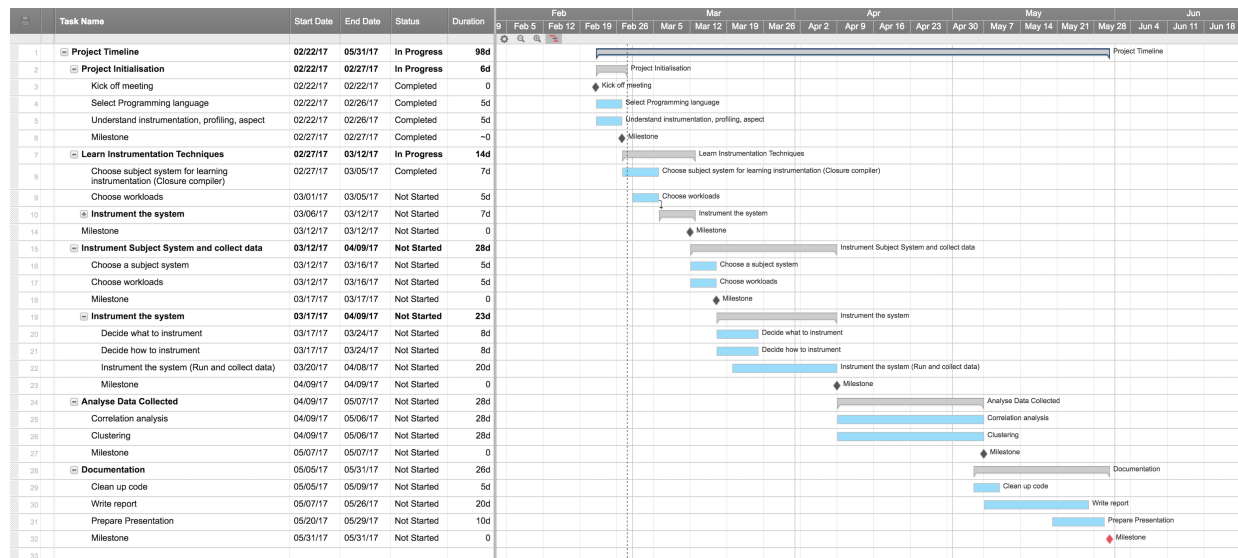
In this project we will instrument a subject system, based on some workload or application scenarios. For example, we could use profiling as an instrument to track the runtime of some function in the system. The performance results received would be recorded and later analyzed using various statistical techniques, and perhaps machine learning. In particular, we will attempt to find a correlation between the performance results and some independent variables (for example, the size of the thread pool). The correlations found would then help derive performance annotations for the subject system (for example an optimal thread pool size).

4. Plan

4.1. Milestones and Tasks

Milestone	Task
M1	Kick-off meeting
M2	Select Programming language Understand what is instrumentation and profiling
M3	Learning how to instrument a system Choose subject system for learning instrumentation (Closure compiler) Choose workloads Instrument the system
M4	Choose a system Choose workloads/ application scenarios (input)
M5	Instrument the system - Decide what to instrument - Decide how to instrument Choose/implement some independent variable to record with instrument result Run
M6	Correlation analysis Clustering
M7	Clean up code Write report Prepare Presentation

4.2. Timeline



4.3. Project Deliverables:

1. Project Report
2. Poster
3. Source code for instrumenting the system
4. Data collected
5. Analysis of Data