Masters Project Proposal

Clover: A root cause analysis system for distributed systems

Tyler Ruppert

Approvals

Approved By:

_____ Date: _____

Advisor: Dr. Yanyan Zhuang

_____ Date: _____

Committee member: Dr. Albert Glock

_____ Date: _____

Committee member: Dr. Sang-Yoon Chang

Background Information

In distributed systems, tracing the root cause of errors can be especially difficult. The metrics we monitor for performance often do not give much insight into what is actually wrong with the system. In web systems, for instance, the page load time is a very closely watched metric. When this grows or spikes, there is no easy way to determine the cause. There are simply too many metrics for a human being to monitor, and the one that actually alerts the user is not the real problem. Small, seemingly insignificant metrics could potentially be the cause of catastrophic failure in a system.

There is already software that solves parts of this problem, but not the entire problem. Two notable examples are Prometheus (https://prometheus.io) and Etsy Skyline (https://github.com/etsy/skyline).

Prometheus is a monitoring tool that will track metrics and alert when there is an issue. This is as far as it goes. Prometheus does not link anomalies between metrics to determine the likely source.

Etsy also built a tool for anomaly detection, called Skyline . Skyline targets monitoring a large number of metrics and automatically determining what it means for a metric to be anomalous without manual configuration. Like Prometheus, Skyline also does not link anomalies between metrics to determine the likely source.

There is a need for software that monitors every piece of a distributed system and can assert the root cause of errors. Clover satisfies this need by watching the most important metrics to users. In the case of this project, Clover will be monitoring response time for a web application. When the response time becomes anomalous, Clover builds a timeline of other metrics that likely led to the problem and alerts the user.

Plan of attack

The work that must be done is nicely split up into six parts. The first three are pre-work that needs to be done to create data for processing through Clover. The last three are actually the services of Clover and work closely together.

The six pieces will be completed in order, and are as follows:

1. Create an example system
2. Set up metric extractors to gather information on the example system
3. Set up a failure injector to cause failures in the example system
4. Build the metric processing service
5. Build the report building service
6. Build the alerting service

Tentative Schedule

Before start of the semester

- Write project proposal and design document.
- Build example system and set up metric extractors.
- Set up traditional monitoring tools that will be used to visually verify metrics are as expected while testing various pieces of Clover.

Jan 16 - 28 (~2 weeks)

- Set up and test ability to inject anomalies into the example system

Jan 29 - Mar 11 (6 weeks)

- Build the metric processing service

Mar 12 - Apr 8 (4 weeks)

- Build the report building service

Apr 9 - Apr 22 (2 weeks)

- Build the alerting service

Apr 23 - May 10 (~3 weeks)

- Write the thesis report

Deliverables

- The root cause analysis system, Clover.
- The thesis report for the design and implementation of Clover.