Review 2

1/23/12

CS6V81.502

Seungtack Baek

**Predicting Faults from Cached History**

To predict which entity will likely to produce fault in the future, Kim et al. utilized the concept of cache. They chose open source projects and took their commit history as their sample. Using the localities of the bugs, such as new-entity, changed-entity, temporal and spatial locality, they predicted future bugs by simply tracing commit history in a chronological order.

To determine future faults, they used two types of caches, Bugcache and Fixcache. Whereas bugcache is their theoretical model that traces all traceable bugs, fixcache practically traces bugs by looking for bugfix and the corresponding changes that introduced the bug. However, they were not sure of the optimal size of the cache, block size and replacement policy, so they simply tried the all possible combination of those, which they found to be varying among different projects.

The authors assert that utilizing cache is better than other method for their method is dynamic, as opposed to be static. Since fault distributions are random and rather dynamic, using cache results in higher accuracy.

I found this article hard to read. It was mainly because they used the word “cache” without specifying that it is not physical cache, but just the concept of it. Also, I could not find how this article could be related to current computing fields. In my opinion, predicting 70% of the bug is little low, since for 30% of the time, cache will be loaded/unloaded for no reason (false positive?).

**Question:**

1. I found it hard to express what they had done in once sentence…. How would you summarize it in one sentence?
2. Also, they came up with rule of thumb for cache size… how??

\*NOTE: Plus, I found typo in their report. In the fugure 4, under Eclipse, it should be (333/331/0/BUG), not (333/331/0.BUG)