

Paper Review

“Capturing, Indexing, Clustering, and Retrieving System History”

1. Summary

This paper introduces a method to leverage the system metrics to diagnose problems. The classifier can distinguish a periodically happened problem from a one-time problem, and annotate possible solutions based on the previous solution of the problem.

This paper collects history metrics. First, it uses k-means (which is a distance from metric to signature) to compute the relationship between the happening of a problem (for example, a violation of service) to a metric (for example, cluster scale). Then, this paper uses the Tree-Augmented Naive Bayes models to compute the joint possibilities of signatures to system state, which can be later used in the discovery of retrieving signatures. It is a native “Stackoverflow”, which can use system metrics (such as high response time and low throughput) to discover the problem, and apply a usual solution. The author has discussed, like Stackoverflow, cannot find the root cause of the problem but only find the possible solution.

I focused on the evaluation of this paper, especially how the author proved the paper works. The author placed the application in real clusters and collect real data. 90% of the data and signatures can be separated between periods of violation and compliance, which is an acceptable precision for diagnosis. Two graphs (Figure 7 and 8) of Average Response Time shows that problems are related to metrics and the application can discover (and fix) the problem by the weight of metrics. With a fact example of SQL performance overhead, the author convinced me that the application works.

2. Advantages

- + This paper combined static methods with system metrics to identify recursive signatures for the violation of SLO problems.
- + The performance impact of the application is low.
- + The result was convincing combined with facts and statics.
- + In clusters, the difference of static can be more obvious since each node runs similar jobs. Anomalies are prominently characterized.

3. Disadvantages

- The application is not able to discover the root cause of the problem.
- Basically, this paper uses static methods to identify problems and their corresponding solutions. It can be brilliant in 2005. Facing more complicated situations, can it work fine now?
- It can identify existing or historical problems, facing a new problem, this paper didn't mention its solution to new problems. It is more like a classifier but not an online learner.

4. Brainstorming

What we want to do in this class, "Provide Deadline-Aware Scheduler for Multi-Scale Heterogeneous Systems by Throughput Predictor" can be divided into two parts, predictor and scheduler. This paper inspires me in the implementation of predictors since we need to know the relationship between metrics and program affinities. In the method of metric sampling, we can find a modern one like The Case for Task Sampling-based Learning for Cluster Job Scheduling, which provide more accuracy. However, the evaluation methodology still has its value today. This paper uses three sections to prove: the effectiveness of the signatures, the validity of clustering, and the quantity of retrieval. For us, we need to prove that our design of predictor is valid, our prediction is useful and our scheduler has better efficiency.