**Pythia: A cross-layer just-in-time instrumentation framework for debugging distributed systems**

Lily Sturmann, Shuwen (Jethro) Sun, Ayse Coskun, Orran Krieger, Peter Portante, Raja Sambasivan

**Motivation**

Today’s distributed systems are often instrumented heavily; however, debugging is still difficult because it is hard to know:
* where to add more instrumentation,
* within what stack level to add instrumentation,
* at what granularity to add instrumentation,
* what information should be collected for future problems

Today’s debugging approach:
* Manual, iterative instrumentation (logging statements)
* Time consuming and inefficient
* 49% of developers’ time is spent debugging!

**Key Insights**

Localizing variation among work that is expected to perform similarly yields insight into where instrumentation is needed.
* Requests that are processed similarly within and among the nodes of a distributed application should perform similarly (exhibit low variation).

* Programatically adding instrumentation in areas of unusual variation should reveal the source of the variation.

**Pythia Overview**

Pythia is a framework for automatically enabling instrumentation where needed in the system in response to newly detected problems or performance slowdowns.

**Pythia Workflow**

1. Group “similar” workflows (processed similarly in system)
2. Analyze groups for high latency variance
3. Localize variation in groups
4. Enable more detailed instrumentation in relevant areas of system

**Current and Ongoing Work**

Our current prototype of the Pythia framework is comprised of 1. a tracing instrumentation library, and 2. a trace data reconstruction, analysis, and diagnosis pipeline

We aim to answer such research questions as:
* how to intelligently collect information across stack levels?
* what would be the right set of search strategies to identify problematic areas?

We started our exploration with OpenStack and Ceph. The prototype analyzes the trace data and groups requests based on their workflow and response time.

**Current and Ongoing Work**

The next steps will be to evaluate the feasibility of such a just-in-time instrumentation framework in real systems and test our hypotheses with in-production systems.

Future work on this project includes support incorporating information from different stack levels or performance counters and implementing focused search strategies to explore what additional instrumentation is needed to diagnose problems.