TraViz Status Update #2

- Matheus Stolet
- Vaastav Anand
BACKGROUND
What are Distributed Systems?

“A distributed system is one in which the failure of a computer you didn’t even know existed can render your own computer unusable.”

- Leslie Lamport
Distributed Systems are everywhere

- Distributed systems are widely deployed [1]
  - Graph processing
  - Stream processing
  - Distributed databases
  - Failure detectors
  - Cluster schedulers
  - Version control
  - ML frameworks
  - Blockchains
  - KV stores
  - ...

Need for Observability: Ability to answer questions

- Which nodes/services did the request go through?
- Where were the bottlenecks for the request?
- What happened at every node/service to process the request?
- Where did the errors happen?
- How different was the execution of 1 request?
- How do different groups of requests differ?
- Axes for differences
  - Structural
  - Performance
- Root cause analysis
Need for Observability: Ability to answer questions

- Which nodes/services did the request go through?
- Where were the bottlenecks for the request?
- What happened at every node/service to process the request?
- Where did the errors happen?

- How different was the execution of 1 request?
- How do different groups of requests differ?
- Axes for differences
  - Structural
  - Performance
- Root cause analysis

Distributed tracing can answer these questions
What is Distributed Tracing?

- Each trace represents path of 1 request through the system
- Trace collects and contains timing info, events across nodes, processes, and threads.
- Depending on verbosity, may also contain stack traces.

“Story of a request through a system”
DATA & TASKS
Datasets

- 2 Trace Datasets & respective source code
  - Hadoop: [https://gitlab.mpi-sws.org/cld/systems/hadoop](https://gitlab.mpi-sws.org/cld/systems/hadoop)
- DSB: 22390 traces
- Hadoop: 72030 traces
Data Abstraction
Want to support 3 different classes of tasks

- Overview Tasks
- Individual Trace Tasks
- Comparison Tasks
Overview Tasks

We want to provide general analytics on the workings of a distributed system

- **Overall stats**
  - Latency Distribution
  - Events Distribution
  - Distribution by Day

- **Src code integration**
  - Connect to source code
  - Identify hot spots in source code

- **Dependency Graph**
  - How do different services/apps depend on each other
Individual Trace Tasks

Allows users to have a detailed view of a trace.

- Visualization of the flow of the trace
  - Use existing viz
- Highlight critical path in visualization
  - Stretch Goal
Comparison Tasks

Want to support 3 comparison tasks

- One Trace vs One Trace
- One Trace vs Many Traces
  - Aggregate Many Traces to 1
- Many Traces vs Many Traces
  - Aggregate Many Traces to 1

Example comparisons

- Request type
- Day request was made
- Latency
Current Status
Things that are done

- Overview Dashboard
- Source Code Dashboard
- Source Code link to github
- Dependency Graph
- Single Trace Viz
- Trace Comparison (backend)
- Trace Aggregation (backend)
- Trace Selection (backend)
- CSS design (partial)
Things to be done

- Trace Aggregation (frontend)
- Trace Selection (frontend)
- Trace Comparison (frontend)