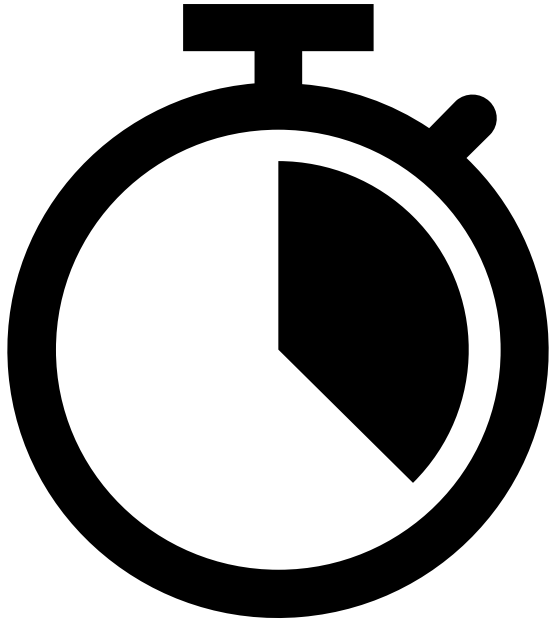
Trace Compass Update

With MVP Demo

Matthew Khouzam Ericsson 2020/01/22

Agenda



- Welcome to the new members
- About us
- About TC
- New Features
- Migration progress
- Demo
- Expected duration: 20 min

Welcome to the new members of the project!

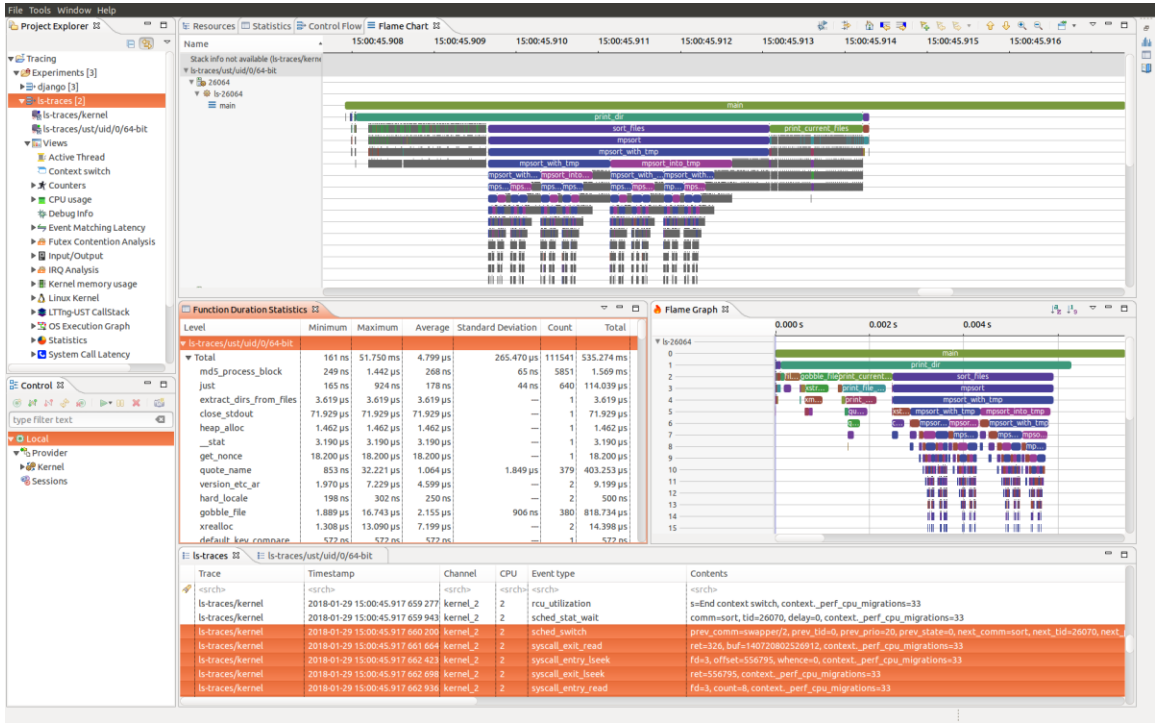
- New Project coming soon ™
- Brock University
- Many new Masters students
- Many new PhD students
- New research associates
- New Professor!

About Us/Trace Compass

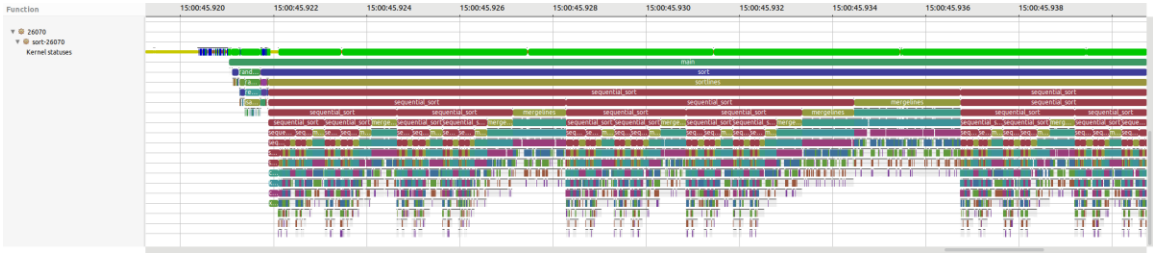
- ***Eclipse Trace Compass*** is an open-source application to solve performance and reliability issues by reading and analyzing traces and logs of a system. Its goal is to provide views, graphs, metrics, and more to help extract useful information from traces, in a way that is more user-friendly and informative than huge text dumps.
- Team of 3 developpers
- Supporting teams internal to Ericsson (they are here, Howdy!)
- Supporting open-source communities
  - Trace Compass (git)
  - Incubator (git)
  - TSP (git)
  - TSP Clients (typescript , python (soon))
  - Trace Extension (git)
  - Timeline chart (git)

Trace Compass

- Main improvement: Speed
- Cross view interactions improvements (e.g. focus on a thread)
- Preserving selection on trees when context changes
- Hide less relevant information
- Time Graph (Gantt chart) is much faster to draw (average of 2x acceleration)
- Density chart improvements
- Security fixes
- Migrated to EPLv2
- Cursors are identical for all views showing the potential action
- Released 6.0, 6.1 and 6.2



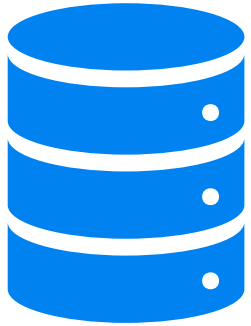
Trace Compass



- Descriptive statistics API
  - Several Internal implementations
  - Not just on categorical data, on continuous data too!
- Improved documentation
- Decoupled UI From XY Plots
- Improved Pie Charts
- Provided XY chart
- Better feature support for incubator (Coming up later)
- Performance improvments
- Markers on Control Flow/Resources view when an experiment has a kernel trace and another with TID info

Trace Compass Incubator Updates

Trace Server Development
(Ongoing)



Java 11 Support

Trace Server Protocol

Efficient way to communicate between server and front-end

Only sends data needed

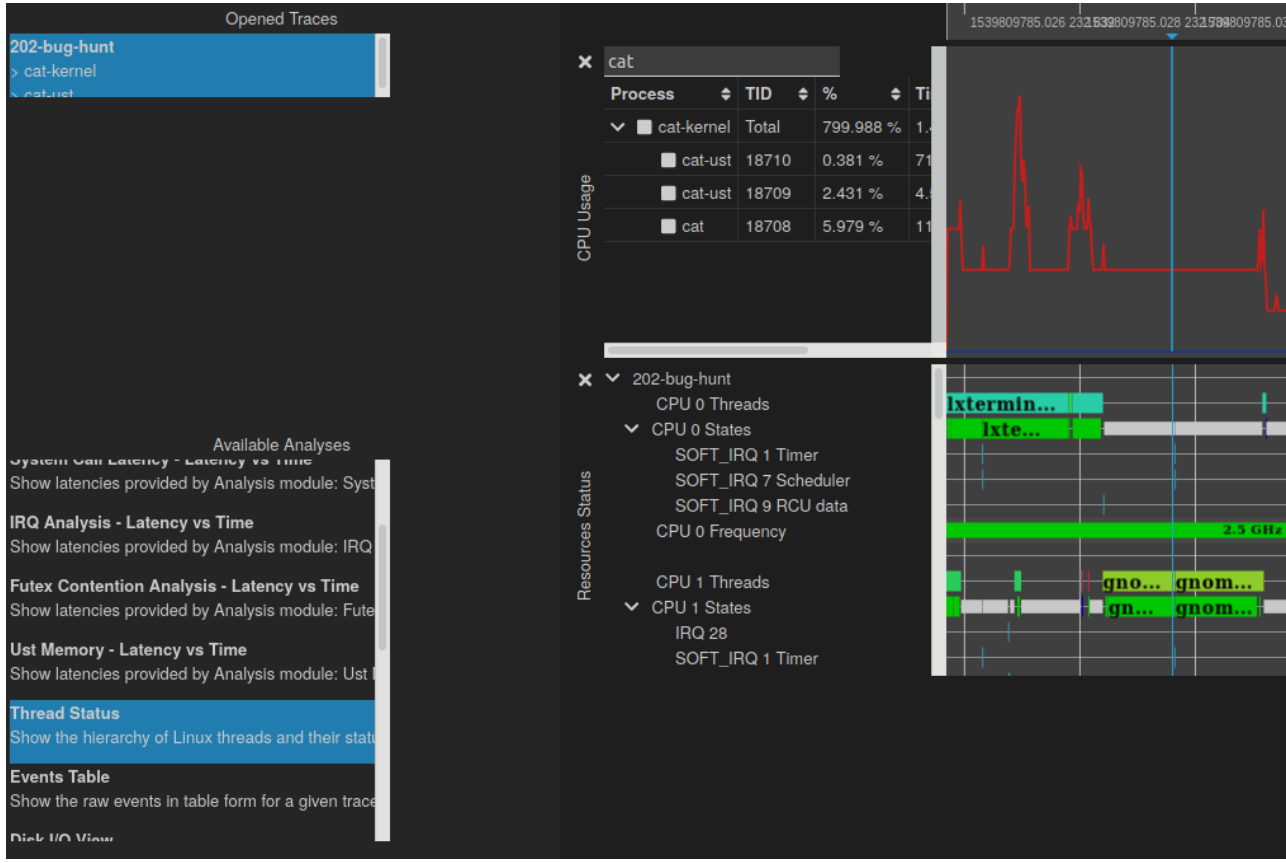
RESTful

In progress. Trying to keep it lean

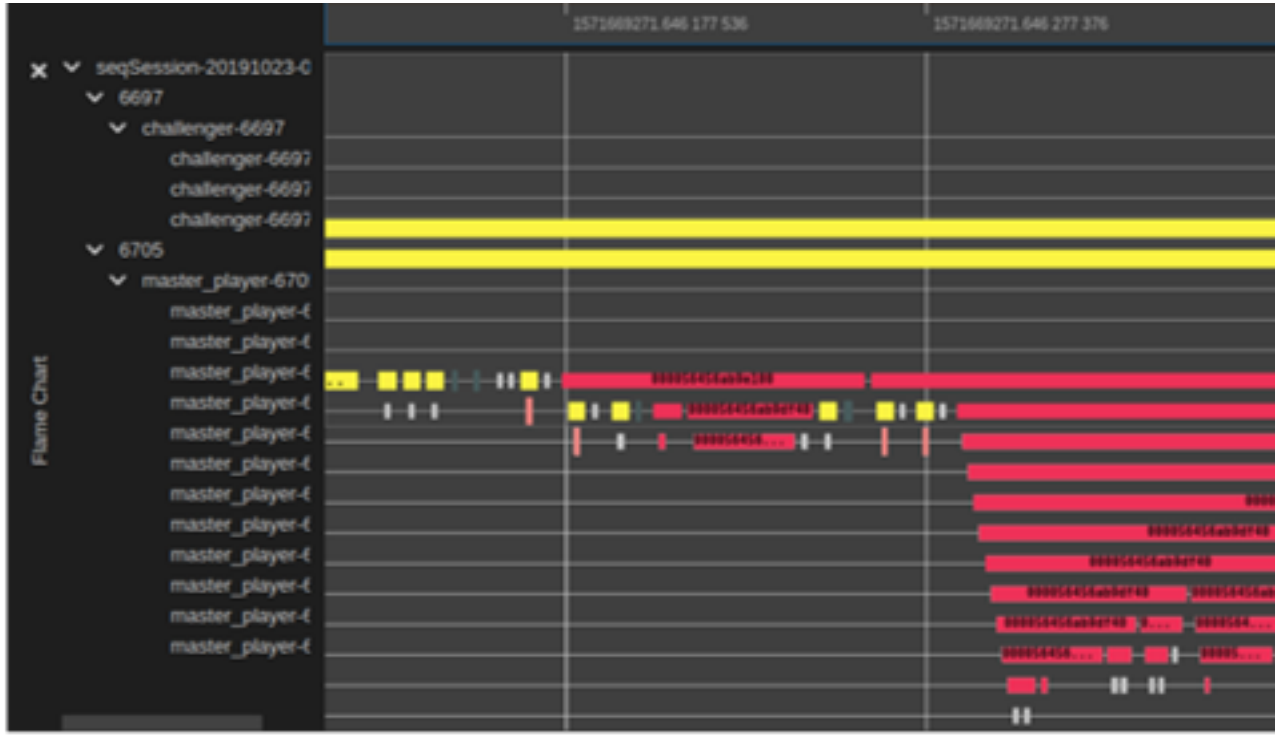
- Tree
- Discrete values
- Continuous values

Trace Extension

- Front end, reads TSP, no notions of trace analysis
- VSCode Migration (POC)
- What is lost by going VSCode right now? NOTHING
- To be deployed in Openvsx. Thank you Theia community (including Redhat) (https://open-vsx.org/)

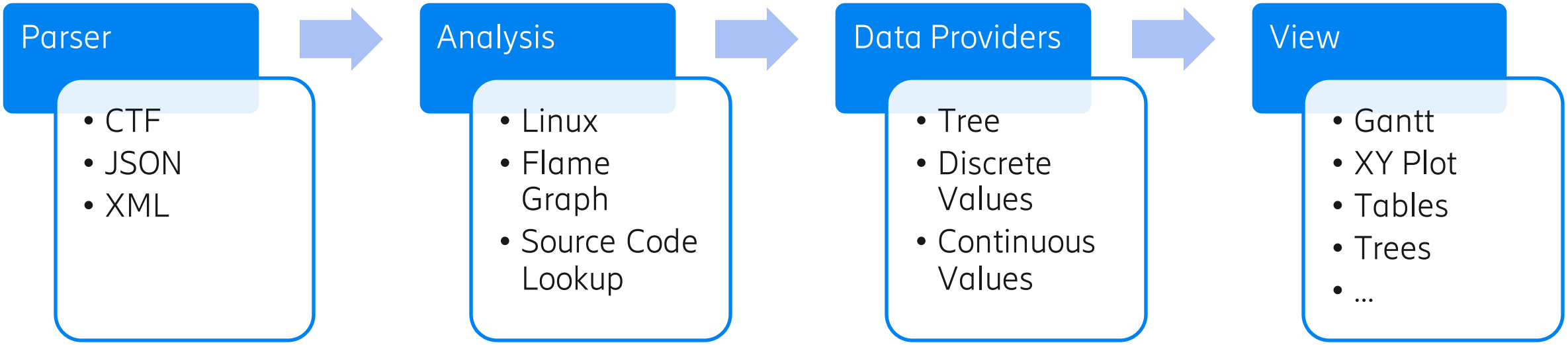


Timeline Chart



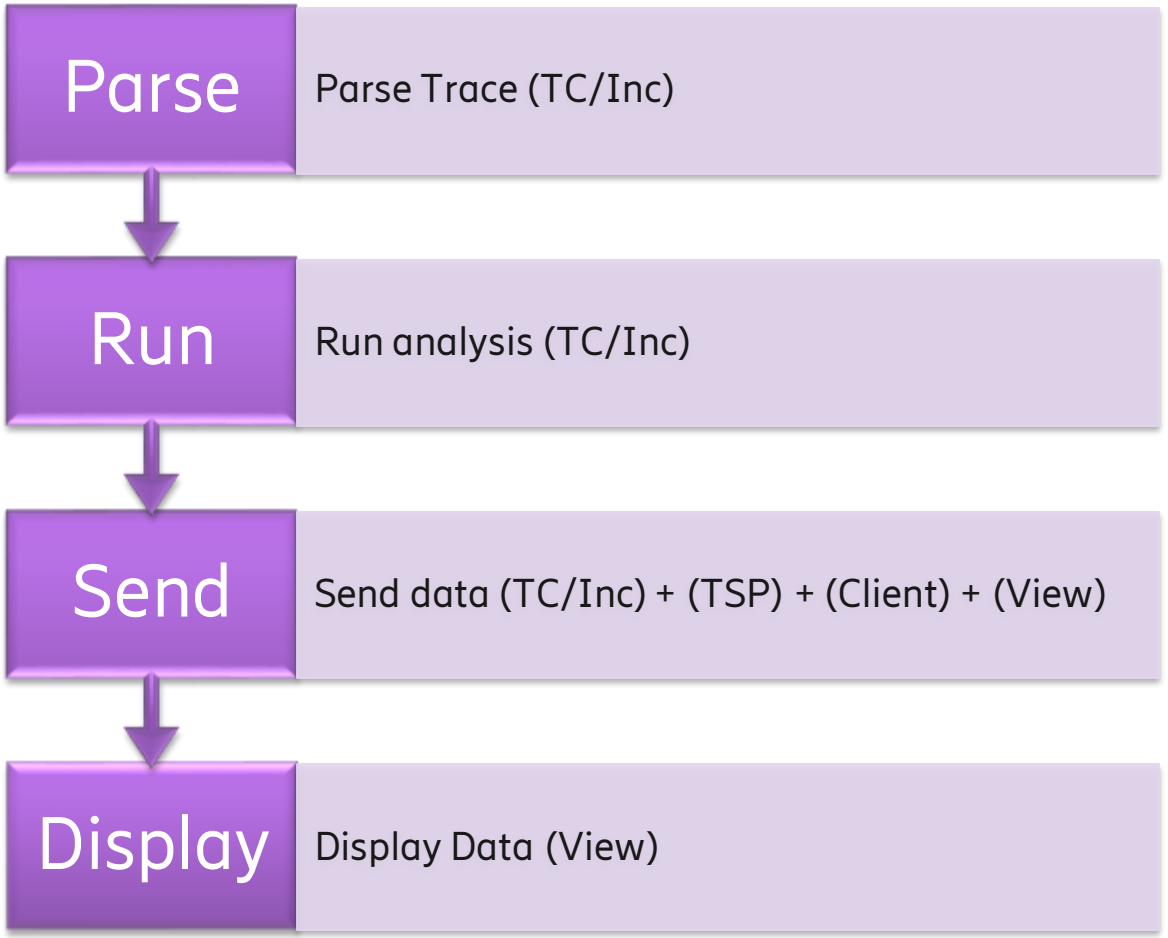
- Uses Pixi.js (Hardware accelerated)
- Structure similar to scene graph
- Anyone with video game experience, come and have fun.

How it all plays together



Migration process and how it affects YOU

- If you want a new way to represent data, it can come in 3 ways (see TSP):
  - Tree
  - Discrete
  - Continuous
- If you want to supply data, use data providers and it should work
- If you want to make a new analysis (data) and a new view, you need to do both
- Most of the bulk path is done. Configurations need to be handled.



Towards the future

- New Eclipse Project: Trace Compass Cloud
  - Working in Github (github way of working)
  - Many things TBD, including timeline :(
  - We will keep you in the loop

Demo

Closing thoughts

- Trace Compass is there to show what happened. In that sense it is a forensic reporting tool
- Trace Compass is highly extensible/scalable, and we will keep pushing for those core values
- New members, I highly recommend you solve an unsolved problem with tracing. This is the best way to learn!