

Advanced Trouble-Shooting of Critical Real-Time Systems

Matthew Khouzam, Ericsson

About this Presentation



- Intro: Why am I here?
- Real-Time Problems
- Proposed Solutions
- Demos
- Q/A?

About Matthew



- Coder
- Worked on Trace Compass since before Trace Compass existed. ;)
- Works at Ericsson, with Efficios and École Polytechnique de Montréal.
- Not a Hurricane.

About ERICSSON



- Working in Open Source
- Over 100k Employees, open source saves a lot of money
- Speeds up developer time (reduces time to market)
- No vendor lock-in

Eclipse Trace Compass



“The single greatest piece of software ever”

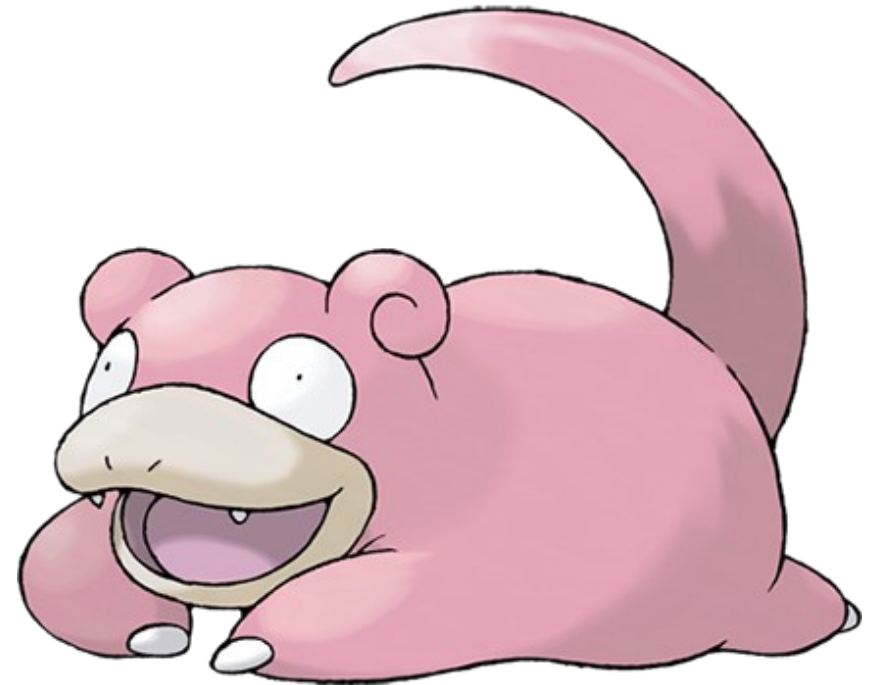
- Bribed Eclipse Trace Compass Developer

- Trace viewer
- Plays well with LTTng
- Can read many large traces and bring it all together

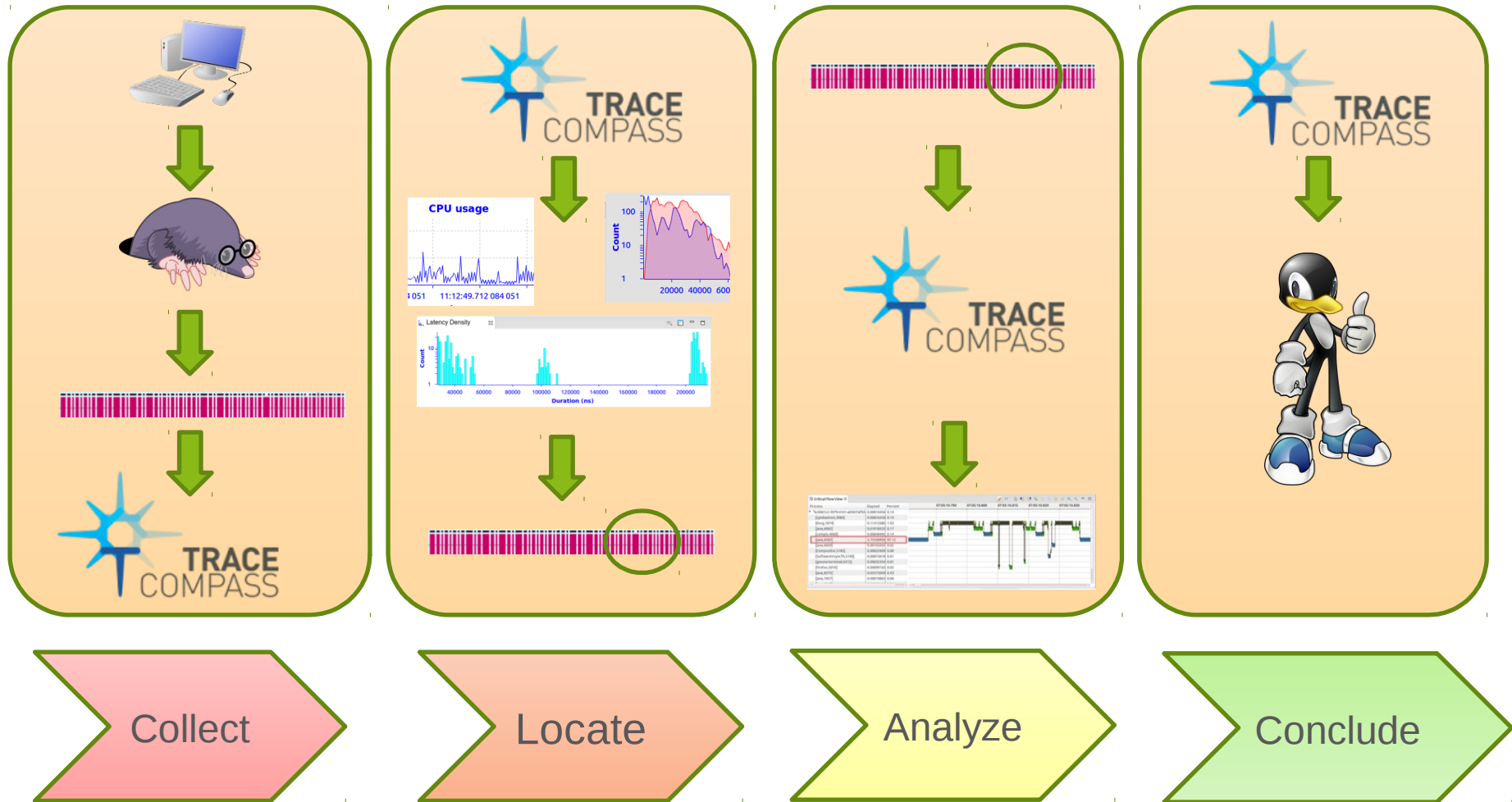
Real-Time Problem



- Late data is bad data
- Timing analysis finds and predicts problems
- 3 major problems:
 - Intermittent missed deadlines
 - Understanding in general
 - Regular missed deadlines



How to solve rt issues?

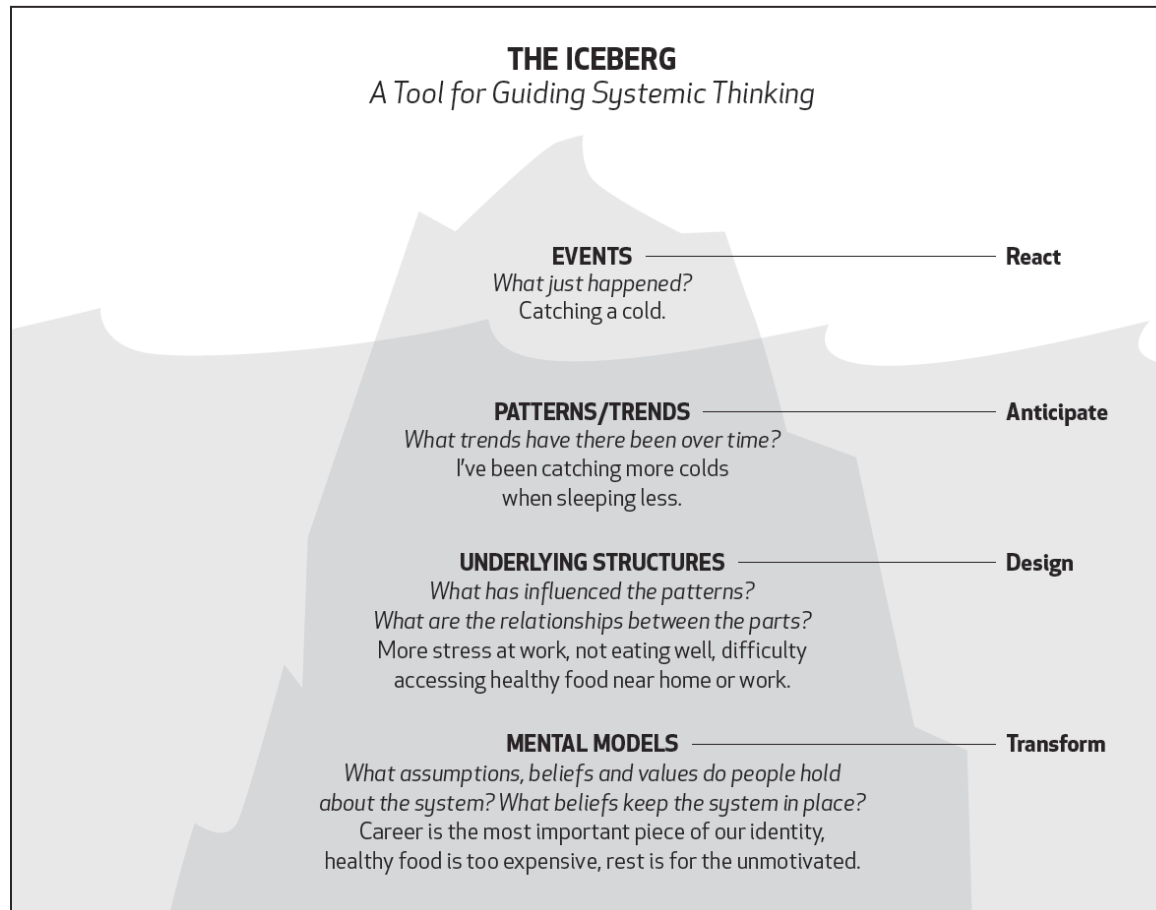


How we help



- Every event has a timestamp
- **Events** alone can find missed deadlines.
- **Event Patterns** can predict missed deadlines
- **Event Structures** can isolate the root cause of missed deadlines
- **Event Models** reflect why the bugs were written

Systemic Thinking





Examples

Example 1:



- Bug in RT System:
 - Cyclic test with 3ms period/deadline “high” priority
 - IRQs on threads with “medium” priority
- Let's see what happens when trigger an IRQ



Demo

Example 1 (cont)



- Reason: KsoftIrqD wakes up cyclic test, it is a lower priority than the IRQ.
- The IRQ has a long service provider for HDMI handshakes.
- The system was misconfigured.

Example 2:



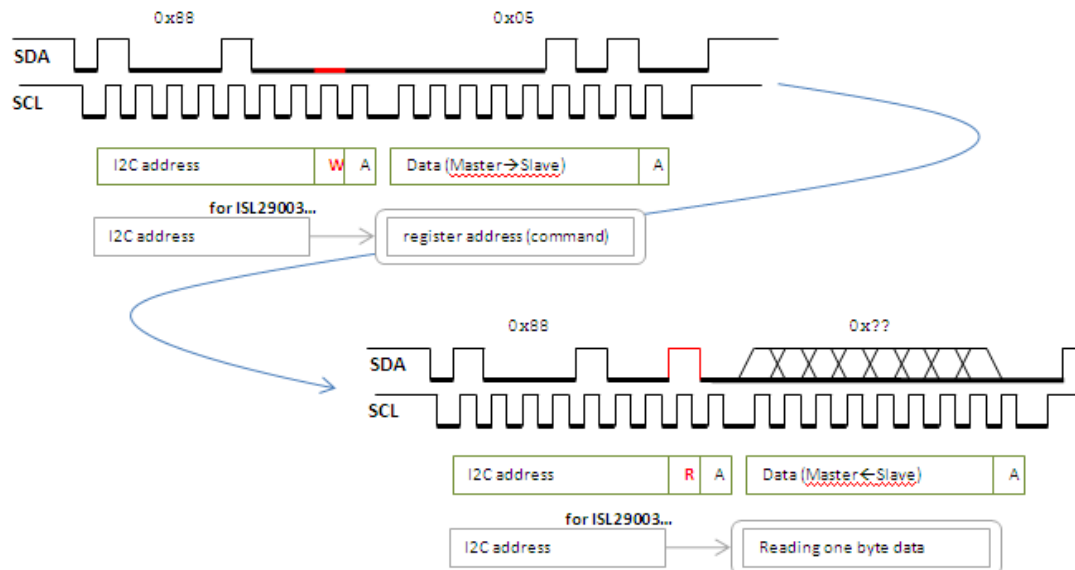
- Rover: A collaboration between Ericsson, École Polytechnique de Montréal, and the Canadian Space Agency
- An open source example:
 - To test robotic solutions for space at a reduced cost
 - To allow access to the full stack for a small fee
 - To have fun with tracing and debugging

Example 2 (cont)



- Objective: Understanding what's going on.
- I2C contention model

Simplified sample of I2C read
1 byte read from register (0x05)





Demo

Example 2 (cont)



- It is important to model shared resources in simulation
- More aggressive locking may be desirable for predictable performance at the cost of peak throughput

Problem 3: (no example)



- Regular missed deadlines are easier to spot due to their regularity
 - Observe CPU usage
 - Observe I/O usage
- Typically misconfiguration or CPU overload

Take Aways



- Eclipse Trace Compass is able to correlate many trace sources and process the data.
- Anything Eclipse Trace Compass can do, a simple command line viewer can do to, but it requires much more expertise.
- The rover allows Eclipse Trace Compass developers to explore user experience opportunities.
- The priorities view can be used for more than finding inversions.



ERICSSON