

Debugging and Observability

Robert Underwood

Clemson University

What is debugging?

How can I debug systematically?

Questions?

What is debugging?

Debugging is a Process

Doesn't this look familiar?

- **Question** – What problem am I trying to solve?
- **Hypothesis** – What do I think is happening?
- **Prediction** – What should I see if it is?
- **Testing** – Do I see what I expect?
- **Analysis** – How do I know?

Why be systematic about debugging?

- You'll do this everyday
- Your time is too valuable:
 - To “resolve” problems from yesterday
 - To get stuck in infinite loops

How can I debug systematically?

1. Question: What problem am I trying to solve?

- Is the system:
 - consistent? *Failing to check error codes*
 - inconsistent? *Problems with random seeds*
 - often problematic? *Parallel file IO*
 - hypothetical? *Modeling or sampling when you can't do a test*

2. Hypothesis: What do I think is happening?

- Consider the model:
 - What could be going on?
 - What cannot be going on?
 - could the effect be of this order of magnitude?
- Think in logical deductive arguments.
- **NEVER** eliminate a case without testing it first!

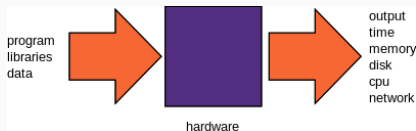


Figure 1: Computers as a Function

Aside: On making good hypotheses

- Start with ballpark tests (short, large scope)
- Consider building a prototype/minimum viable product
- “The simplest solution is often the right one”
- “Once you eliminate the impossible, whatever remains, no matter how improbable, must be the truth”

3. Prediction: What do I expect to see?

- What should you measure (goal, question, metric)?
- How should I measure it?
- What are the limits of my measurement?
- How will measurement itself impact results?

Aside: what tools exist?

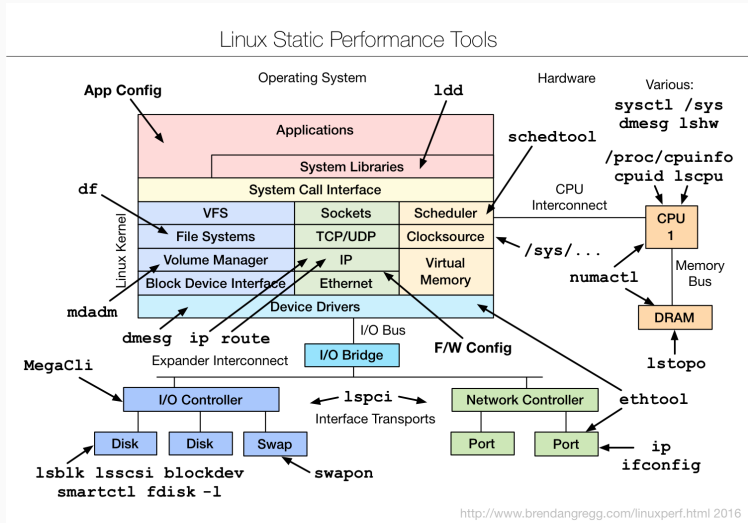


Figure 2: Linux Observability Tools, Brendon Gregg

Aside: on taking “enough” measurements

- Why take multiple measurements?
- How precise does your measurement need to be?
- How much error is in your measurement?
- How many measurements is enough?
 - Only for normally distributed errors
 - Means: $n = \left(\frac{2z\sigma}{E}\right)^2$ Proportions: $n = \frac{4z^2}{E^2 p(1-p)}$

4. Testing: What do I see?

- Use multiple identical systems “under test”
- Also use “control” machines

5. Analysis

- Was my prediction right?
- Did my assumptions hold?
- Does this matter?
- What should I ask next?

Questions?
