Experiments in Computer Science

"The fundamental principle of science, the definition almost, is this: the sole test of the validity of any idea is experiment"
— Richard P. Feynman

• Tried and true experimental scientific methodology from Physics, Biology, Chemistry ...
  • Often not followed in Computer Science
  • Let's be better Computer Scientists!

Scientific Methodology
• Observe
  – (Devise solution)
• Hypothesize
• Design
• Experiment
• Analyze
• Report

Methodology: Observe and Understand
• Find Problem
  – Test: make of Linux kernel
  – Build: memory intensive programs
  – Read: Linux Hacker’s guide says …
• Understand Relationships
  – Hard page faults are expensive
  – Logical memory larger than physical

Methodology: Devise and Hypothesize
• Devise Solution (unless empirical)
  – Claypool Reliable Audio Protocol (CRAP)
  – Claypool buffering algorithm
• Make Hypothesis
  – Generalization about relationships
  – Soft page faults are common
  – Malloc does not cause page fault
  – Needs to be tested (not proven)

Methodology: Experiment
• Design Experiment
  – Variable: variable workload
  – Control: baseline workload
• Run Experiment
  “Whoa! That’s not what I expected!”
  – Bug in code
    + Back to “Run”
  – Uncontrolled event (system backup)
    + Back to “Design”
  – Insufficient understanding (Unix scheduling)
    + Back to “Understanding”
Methodology: Analyze

- Interpretation and Evaluation
  - Statistical significance
    + mean, confidence intervals, correlation, goodness of fit
  - Does data support or reject hypothesis?
  - Explanation of other phenomena
    + Better code reduces page faults, improves performance

Dirty Little Secrets

- Mini-experiments (no, “Pilot Tests”)
- Hypotheses after the fact
  - Running yields understanding
- Results here mean results there
- Controlled system still says meaningful things about the real world
- Observing a system will not change it

Graph: A Data Analysis Tool

- A picture is worth a thousand words
- Title, label axes (units!), legend