



Measuring glibc

{Micro, System}benchmark measurement and
analysis

Siddhesh Poyarekar

9-Aug-2015

The Agenda

- Describe the current state of benchtests
- Using benchmarks
- Whole system benchmarks

Coverage

- Reasonably complete math tests
- Some malloc tests
 - Testing small workloads (up to 32K)
 - Only heap allocation testing
 - Multithreaded testing to test contention
 - Except that it may not be testing contention at all
 - Except, maybe in 32-bit
- String tests
 - Wait, don't start flaming yet!
- Some other tests (strcoll, pthread_once, sprintf, etc.)



Architecture

- Bench
 - A default template for output (min/max/mean)
 - Detailed run prints a measurement per input
 - Concise JSON output
 - Most tests (math, pthread, etc.)
- Benchset
 - Fully customized input/output
 - Usually uses test-skeleton.c
 - String/malloc tests
- compare_bench.py
 - Compare two bench.out files
 - Pretty graphs!



Whole system benchmarks

The hand-wavy bit

- Minimum framework (a systemtap script?)
 - Users measure their workloads and contribute data
 - We stuff it in a DB and run tools on it
 - Decide simulated workloads for benchtests
 - Show pretty graphs and write vague, important sounding analyses
- Is anybody interested in seeing this?
- Who wants to do the work?



Open questions

- What's missing and who wants to fill it in?
 - More comprehensive malloc testing
 - String benchmarks
 - Per-arch benchmarks?
- Using benchmark outputs
 - Performance regression testing
 - Visualizing function performance across releases

