Linaro connect
San Francisco 2017

Benchmarking @TCWG

Maxim Kuvyrkov
Christophe Lyon

Linaro
connect
San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

DEMO!

Compiler battle: GCC vs LLVM

- AArch64: gcc-aarch64 vs llvm-aarch64
- ARMv7: gcc-armv7l vs llvm-armv7l

OR

Linaro
connect
San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

DEMO!

Compiler battle: GCC vs LLVM

- AArch64: gcc-aarch64 vs llvm-aarch64
- ARMv7: gcc-armv7l vs llvm-armv7l

OR

ISA/ABI battle: AArch64 vs ILP32 vs ARMv7

- GCC
  - gcc-aarch64 vs gcc-aarch64_ilp32
  - gcc-aarch64_ilp32 vs gcc-armv7l
  - gcc-aarch64 vs gcc-armv7l
- LLVM
  - llvm-aarch64 vs llvm-armv7l

What do we benchmark?

- We are ToolChain WG, so we benchmark toolchains
- We compare toolchains, to check performance changes:
  - From release to release
  - Our patches
- On the same hardware
- Using SPEC CPU2006
- Planning to add SPEC CPU2017, CoreMark, EEMBC

What we don’t do

- We do NOT benchmark hardware
- We do NOT compare absolute performance results

Hardware configuration

- Cortex-A57: AArch64 and AArch32
- Cortex-A15: ARMv7
- Several identical boards in parallel to reduce latency
- 4 cores per board
  - 1 core running benchmark, 3 cores idle / system
  - 3 cores running benchmarks, 1 core idle / system
- Up to 3 benchmarks in parallel with low noise
- Up to 9 benchmarks in parallel with high noise

Jenkins

- Provides UI to start benchmark jobs
- Schedules benchmark runs
- [optionally] reboots benchmarking board
- Calls harness scripts
- https://git.linaro.org/toolchain/jenkins-scripts.git/tree/tcwg-benchmark.sh

- Jenkins has no direct connection to benchmarking boards
  - Stub jenkins nodes used for scheduling to avoid Jenkins clients on the boards
- 1-1 mapping between stub nodes and benchmarking boards
- Stub node connects to benchmarking board via ssh

Linaro
connect
San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Harness

- Runs on the benchmarking board
- Installs benchmark
- Configures benchmark
- Builds benchmark [in parallel]
  - Remote and cross-compilation is supported
- Configures board
- Runs benchmark [in parallel]
- Collects results
- Uploads results to dev server for later analysis

https://git.linaro.org/toolchain/bmk-scripts.git

Reducing noise

- Out-of-order cores imply a fairly high level of noise in the results
- Several iterations
  - Keep either mean or peak performance
- Reboot boards before running the benchmarks
- At least 1 CPU reserved for system / harness tasks
- 1-3 CPUs used for benchmarking
- Benchmarking on only 1 CPU gives better results
- Fix CPU frequency at 90% of the max on the benchmarking cores
- [If possible] Use minimal CPU frequency for idle cores

Results

- Benchmarks run under “perf record”
  - perf process runs on the system core
- Perf.data files are raw results
- 10 perf cycle samples == 1 second
  - Use perf’s sample period setting, do NOT use perf’s sample frequency setting
- Record everything happening on the core
  - Benchmark executable
  - Dynamic libraries
  - Kernel
- Perf.data files from AArch64 / ARMv7l can be analyzed using x86_64 perf.

Linaro
connect
San Francisco 2017

ENGINEERS
AND DEVICES
WORKING
TOGETHER

Back to DEMO!

Linaro connect
San Francisco 2017