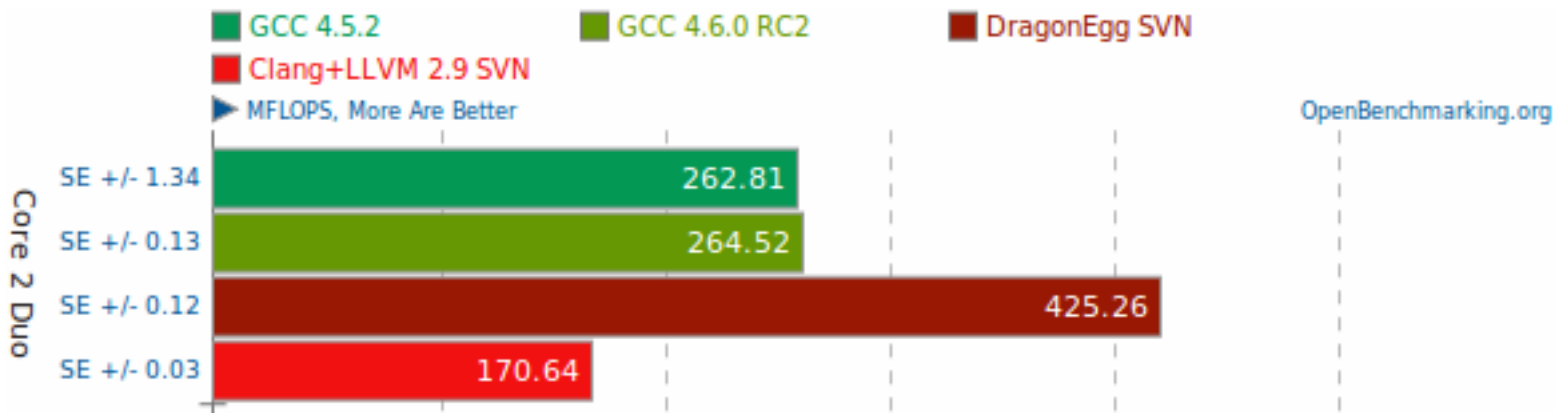
Benchmarking & Continuous Testing LLVM

By Michael Larabel & Matthew Tippett

Competitive Benchmarking LLVM

LLVM can perform well, or it can perform badly



Both workload dependent and version dependent.

LLVM based tools can be extremely interesting.

Observations from the Outside

- LLVM has a vibrant community
- LLVM is best of breed for some workloads under some scenarios.
- Managing performance across many workloads will help generalize and increase LLVM's value across the ecosystem.
- Continuous Integration & Test is a great way to monitor and maintain performance.

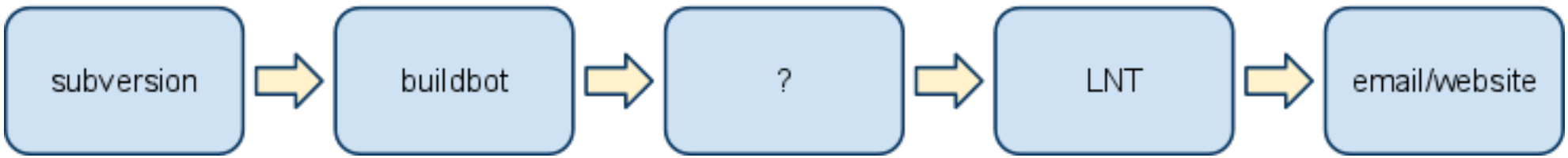
But...

- Setting up centralized CI infrastructure is actually quite hard and usually supports only a subset of projects and targets.

LLVM's CI workflow

Typical CI Workflow

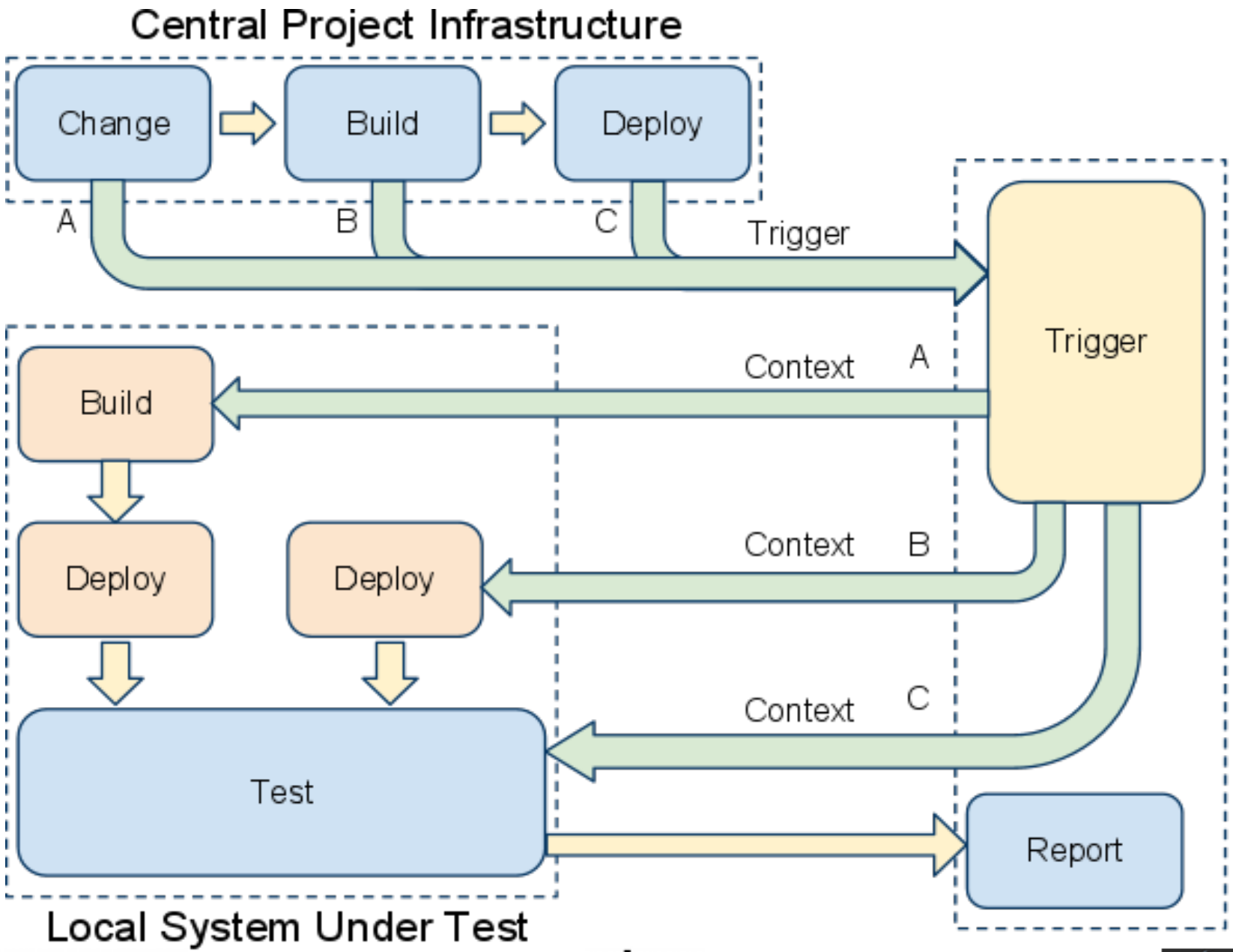
Current LLVM CI Workflow



LLVM has infrastructure for supporting building for a subset of projects, and then testing those builds on a few pieces of hardware

Unfortunately, this doesn't scale to 10's of projects with individual hardware for each project.

Phoromatic's CI workflow



How Phoromatic Works

(We are in beta, so bear with us)

1. Projects can Trigger the CI flow at the most convenient point in their process (VCS Change, Build, Deployment)
2. Phoromatic will the send the "Set Context" to the registered Agent Systems Under Test
3. The "Set Context" call completes the balance of the CI workflow on the local machine (Build, Deploy & Test)
4. Results are pushed up to a Central Server

Benefits are that you can start a CI workflow with a single local machine under your desk and a VCS.
Easily scales to multiple projects and multiple systems under test.

Setting it Up Yourself

1. Create & Publish Tests
   - Bash & XML
   - Local or uploaded to OpenBenchmarking.org
2. Use Phoronix Test Suite to log onto Phoromatic
3. Create a `set-context.sh` script
   - It receives the context and sets up the for the test
   - This may include a local build and deployment - depending on the point of triggering
4. Configure your VCS, build system or deployment server to hit Phoromatic with a context that `set-context.sh` understands.
5. Monitor your results.

Suggestions for the LLVM Community

- Choose the right trigger point
  - For projects using buildbot
    - trigger with the build output as a context
  - For projects without buildbot support
    - trigger based on VCS commits (like buildbot)
- Allow the project community to have ad-hoc CI infrastructures
  - Developers, users may have different interests and tests
  - Since cost of entry is low, let the community self-organize based on interest and focus
  - Optimize once established

OSS Projects Using Phoromatic/PTS for CI Testing

- Phoronix Media Sponsored (getting reworked to support openbenchmarking.org)
  - Ubuntu Daily Kernel
  - Daily Ubuntu
- Community
  - Wine (using PTS and custom scripts)
  - XBMC (under discussion)
  - Intel Graphics Driver (under discussion as of yesterday)
- Lots of random users
  - Doing their own daily testing for their own purposes

Next Steps...

- If your project wants to get some level of testing
  - Contact us, we've got lots on the move (and we really need to document it better :)
  - You really only need one or more machines that can be semi-dedicated

- We're in beta, so we can bring LLVM projects into the beta deployments
  - It's easier than you think, less than a couple of days of setting it up and running.

Questions?

Michael Larabel <michael@phoronix.com>
Matthew Tippett <matthew@phoronix.com>