November developer meeting

Agenda

- Current status
  - Testing infrastructure
- Ongoing work
- gem5 development discussions
  - Memory leak in Ruby (#518, #514, #508)
  - Failing weekly and compiler tests
  - X86 performance issues (#398)
  - Branch predictor / decoupled front-end updates
- Plan for v23.1
- Other items

Community Code of Conduct

https://github.com/gem5/gem5/blob/stable/CODE-OF-CONDUCT.md

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

Code reviewing: **Everyone is welcome!** Experts not required

- Be kind!
- Give ***specific and actionable*** comments
- Don't let perfection be the enemy of progress
- Does it do what the commit message says? Does it improve gem5 for someone? Does it meet the style guide? Does it increase the maintenance burden?

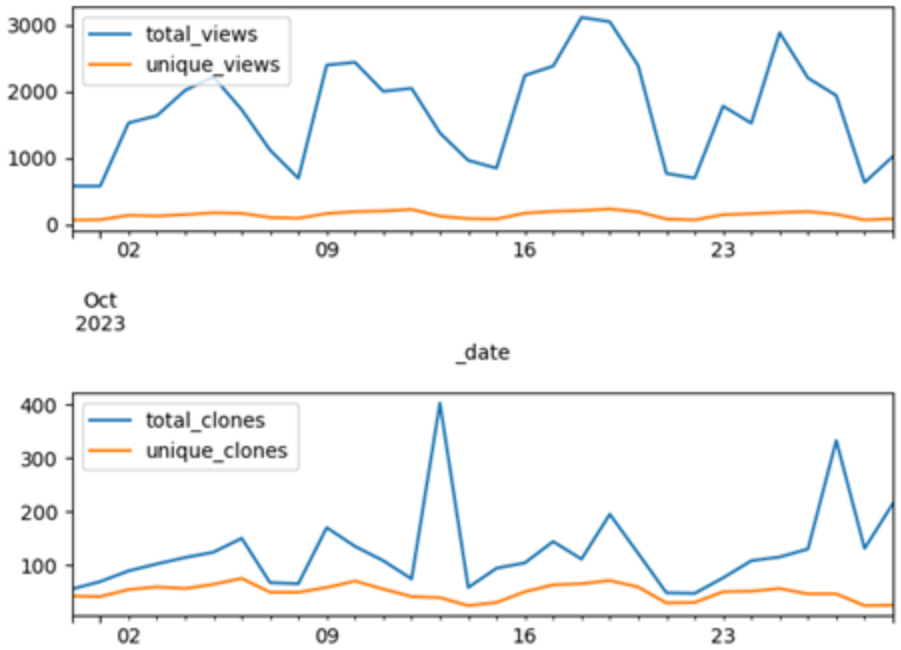
October Highlights

Issues: https://github.com/gem5/gem5/issues

- Issues Created in October: 24 (24 in Sept)
- Issues Closed in October: 12 (17 in Sept)
- Issues Open (total): 41 (31 in last meeting)

Pull Requests: https://github.com/gem5/gem5/pulls

- Pull Requests Created in October: 65 (73 in Sept)
- Pull Requests Merged in October: 54 (46 in Sept)
- Pull Requests Open (total): 43 (44 in last meeting)



18 Unique authors committed 88 commits (excluding merge commits)

Average of 52 unique clones a day (50 in Sept), 3624 clones total (2290 in Sept).

CI testing infrastructure

- Apologies for flakiness
- GitHub Actions with "self-hosted" runners
- Types of runs:
  - CI tests (2-3 hours)
  - Daily tests (24+ hours)
  - Compiler tests (15+ hours)
  - Weekly tests (2 days)
  - Weekly tests (GPU) ???
- Running in VMs for security
  - Vagrant not very user-friendly
  - Tradeoff between cost, control, and responsibility

Ongoing work

- Classic prefetchers in Ruby (#502)
  - Mostly ready to go. Waiting on confirmation of passing replacement policy tests.
- Improvements to workloads/resources (#532, #510)
  - In progress
  - Suites, workloads added to resources
  - Updating resources schema
  - "tooling" to make it easier to manage database
- Kconfig (#69)
  - Separate PRs for Daily and weekly
  - @Bobby needs to merge this
- Improvements to branch predictors/front-end (more later)

Discussion 1: Memory leak(#518, #514, #508)

Atomic log deletion missing in GPU coalescer (#508, #518)

execution of atomicPartial was populating the atomic log queue without ever clearing it. (#514)

Are these related? Is there a single solution?

Discussion 2: failing tests

Compiler tests failing because GCC 9 runs out of memory

- Increase memory of VM (16+GiB): Not ideal
- ???

Weekly tests failing

- Possibly just a configuration problem (not building the right version of gem5)

Discussion 3: x86 performance (#398)

Bhargav Godala (bgodala)

IPC limited in x86 due to dependencies on condition code register

One options is "atomic" sub registers

Another option could be masks for each registers

Discussion 4: Front-end improvements

We need a single place for discussion/tracking progress

Done:

- Restructure BTB (#412)
- Restructure RAS (#428)
- Refactor indirect predictor (#429)
- General improvements to BP (stats, clean up code, "target provider" class) (#455)
- New probe listener with lambda (#356)
- Get instruction size dynamically (#357)

In progress:

- Decoupled front end (#359)
- Require BTB hit to detect branches (#493)
  - Requiring a BTB hit will change performance for SPEC2017 (there will be a small IPC decrease). We are 'cheating' as there are some places where a BTB hit should have been required but it is assumed that the instructions have been decoded when they are passed to the BPU. Requiring the BTB hit makes the model more realistic but we have to think out where we require the hit.
  - Main problem is that we are not querying the BP on non-branches
- BPU support for decoupled front-end (#499)

gem5 v23.1 (staging branch by Dec 1)

Must haves

- Suites & workloads
- Classic prefetchers in Ruby
- Others?

Would like to have

- Improved exit event support (#474)
- Tested disk images in resources

Won't do

- Support for multiple Ruby protocols

Kconfig

Improved front-end