



gem5 Developers' Meeting

April 2024

Agenda

Current Status of gem5 Development

Upcoming v24.0 release

gem5 Stats, “PyStats”, and what we want from the stats output

Other items (Open for any gem5 Development discussions to be had)

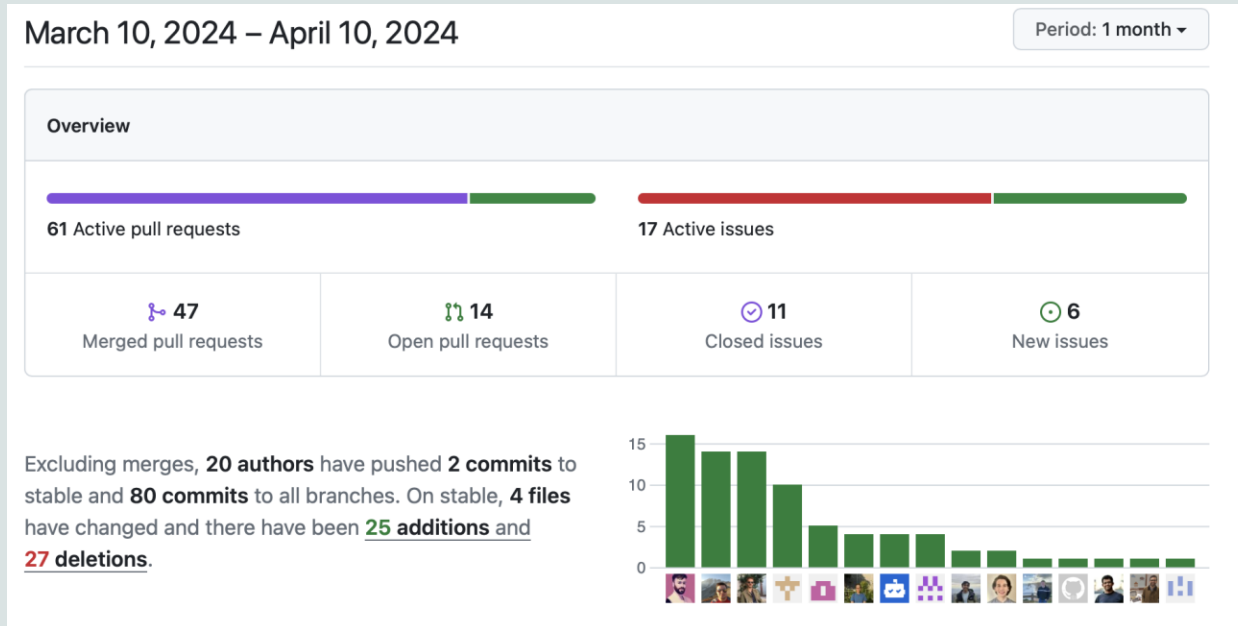


Current Status of gem5 Development

Daily Tests **passing**



Weekly Tests **passing**





Compiler Tests **passing**



Current Status of gem5 Development

misc: Add Clang format to pre-commit #362

 Open trollet91 wants to merge 9 commits into [gem5:develop](#) from [trollet91:clang_format](#) 

 Conversation 85  Commits 9  Checks 34  Files changed 2,106



Upcoming v24.0 release

Release*: June 1st

Staging Branch creation: May 14th

What we, at UC are focusing on:

- Improving statistics.
- Objects returned as exit events.
- Better support for multiprocessing.
- More disk images (somewhat related to exit event improvements)

Test support for Ubuntu 24.04 #909

Open

ivanaamit opened this issue on Mar 4 · 1 comment

24.04 is released April 25th: Should be supported for v24.0.

We had some issues with 22.04 so best to check how we're doing sooner rather than later.

* Motivated by mid July BootCamp (need 1.5 months to comfortably develop materials on this release)



Upcoming v24.0 release

Who else is working on something with the aim of incorporating in v24.0?



PyStats

In v21.0, I introduced By stats. It allowed for the gem5 statistics to be output as a JSON file instead of the text statistics.

It was not widely used as far as I know.

I've been working on improving it.

```
{
  "creation_datetime": "2019-02-20 15:00:00",
  "system": {
    "type": "SimObject",
    "name": "system",
    "cpu": [
      {
        "type": "SimObject",
        "name": "cpu0",
        "num_inst_executed": {
          "value": 101.0,
          "type": "Scalar",
          "description": "The number of Instructions Executed.",
          "unit": "Count",
          "datatype": "f64"
        }
      },
      {
        "type": "SimObject",
        "name": "cpu1",
        "num_inst_executed": {
          "value": 204.0,
          "type": "Scalar",
          "description": "The number of Instructions Executed.",
          "unit": "Count",
          "datatype": "f64"
        }
      }
    ]
  }
}
```

PyStats

```
simstat = get_simstat(root)

# Access in a script following the same structure as the JSON.
cpu0_inst_executed = simstat.system.cpu[0].num_inst_executed.value
cpu1_inst_executed = simstat.system.cpu[1].num_inst_executed.value

# This format also works.
cpu0_inst_executed = simstat['system']['cpu'][0]['num_inst_executed']['value']
cpu1_inst_executed = simstat['system']['cpu'][1]['num_inst_executed']['value']

# Allows for better iteration.
total_insts = sum(c.num_inst_executed.value for c in simstat.system.cpu)
```

```
{
  "creation_datetime": "2019-02-20 15:00:00",
  "system": {
    "type": "SimObject",
    "name": "system",
    "cpu": [
      {
        "type": "SimObject",
        "name": "cpu0",
        "num_inst_executed": {
          "value": 101.0,
          "type": "Scalar",
          "description": "The number of Instructions Executed.",
          "unit": "Count",
          "datatype": "f64"
        }
      },
      {
        "type": "SimObject",
        "name": "cpu1",
        "num_inst_executed": {
          "value": 204.0,
          "type": "Scalar",
          "description": "The number of Instructions Executed.",
          "unit": "Count",
          "datatype": "f64"
        }
      }
    ]
  }
}
```

stdlib: Improve gem5 PyStats #996

 Draft

BobbyRBruce wants to merge 13 commits into [gem5:develop](#)



PyStats

My Question: What do you need from statistics which isn't available?

What would make parsing and using the stats output, or the stats in the scripts easier?



Other items to discuss?

