

Sustainable ModSim Software & Tools

A Case Study in gem5

Jason Lowe-Power
jlowepower@ucdavis.edu

Outline

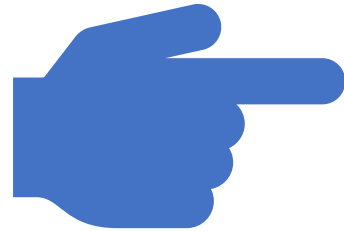
Start positive (what we did right)



Go a bit negative (what we have done wrong)



What to do next



About me

Architecture researcher focused on memory/caches

“Project management committee chair” of gem5

Involved in gem5 community for > 13 years



Interest in open-source and research software for > 20 years

Beginning to be involved in open-source sustainability research



Outline

Start positive (what we did right)

Go a bit negative (what we have done wrong)

Lessons learned





Sustainability successes

Software architecture

- Modular design

- Software engineering best practices

- Dogfooding

- Documentation

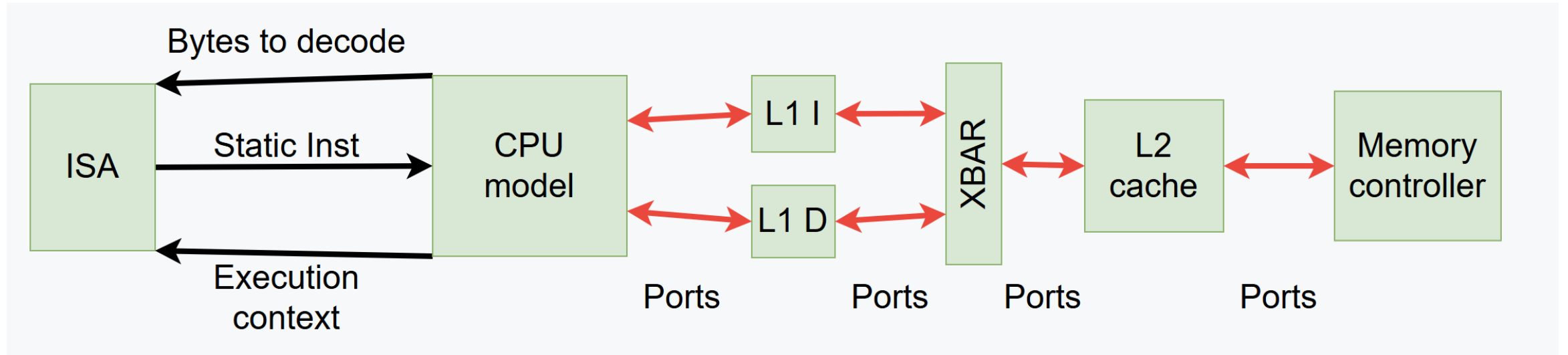
Community of contributors

Open-source license



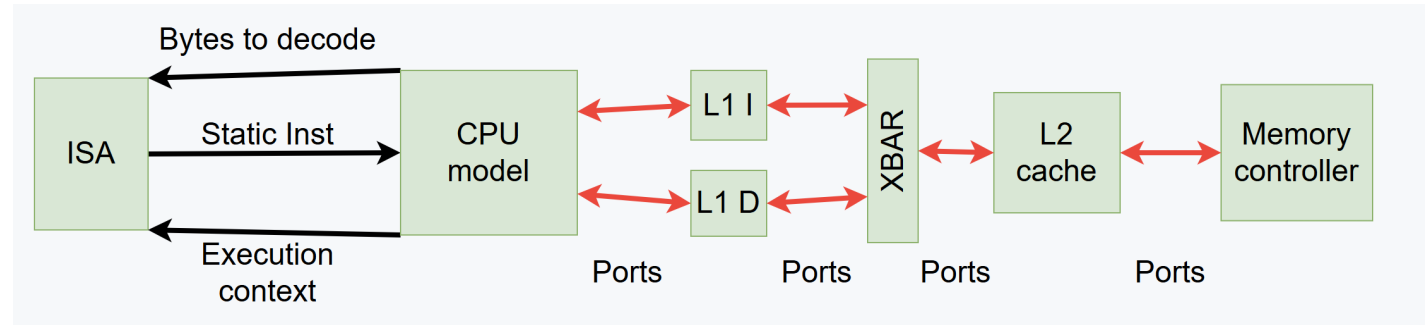
gem5's modular software architecture

Well-defined interfaces

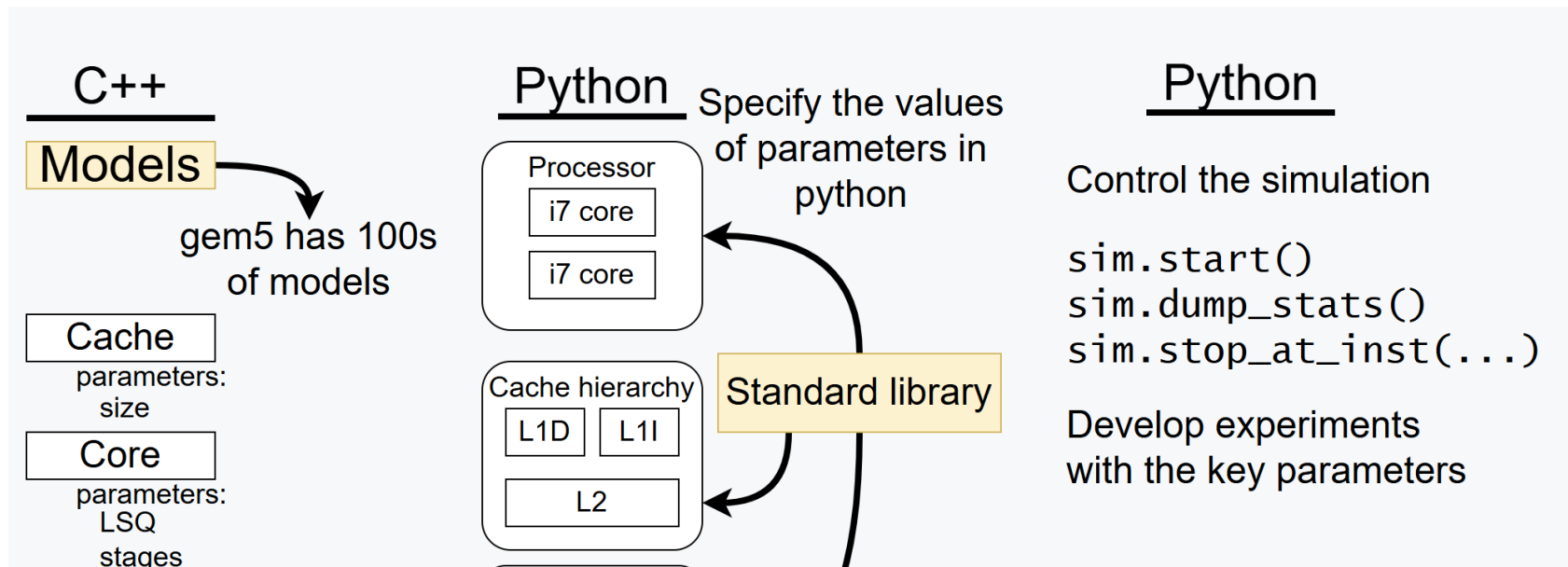


gem5's modular software architecture

Well-defined interfaces



Separation of concerns



Software engineering best-practices

Version control used... forever

Initial commit in 2003!

Strict code review

1955 comments on PRs in 1 year

Many PRs have 30+ comments

Average of >2 (including draft and closed)

```
commit a06eab767e186d309512950183176a57cb1be9d0
Author: Steve Raasch <sraasch@umich.edu>
Date:   Mon Oct 6 17:06:23 2003 -0400

    Initial repository create

--HG--
extra : convert_revision : 441385e70e7f61dccba3cb898cc6a6045a8205ec

commit 469b6e8a850a063d6c81d234b5e9cc19bd74c2c6
Author: Steve Raasch <sraasch@umich.edu>
Date:   Mon Oct 6 17:05:58 2003 -0400



    BitKeeper file /m5/Bitkeeper/m5/ChangeSet

--HG--
extra : convert_revision : ed2344d5584d178cb0a2f02e3dd23ff8c504b097
lines 317903-317931/317931 (END)
```

Lots of CI testing

Over 3000 runs in last year

12 hour dailies, 1.5 day weeklies

 gpu-compute,tests: Move GPU tests to testlib CI Tests #3225: Pull request #1270 synchronize by BobbyRBruce	📅 19 hours ago 🕒 7h 19m 25s	...
 gpu-compute,tests: Move GPU tests to testlib CI Tests #3224: Pull request #1270 synchronize by BobbyRBruce	📅 19 hours ago 🕒 1m 23s	...
 docs,misc: RELEASE-NOTES.md updates for v24.1 CI Tests #3223: Pull request #1460 opened by BobbyRBruce	📅 20 hours ago 🕒 11h 15m 46s	...

Dogfooding



gem5 has always been built
by researchers for
their research

New features are developed
by users

Code accepted upstream only
when generally useful



Provide documentation

In-person tutorials and bootcamp

- 1 week bootcamp in 2022 and 2024

- Over 1100 slides and 500 example files

- 1-2 tutorials per year at major conferences

- 200-300 people taught

Extensive online documentation

- Over 10,000 lines of markdown text

3,000 views, 230 watch-hours each month on YouTube



gem5's community

Over 400 contributors in log

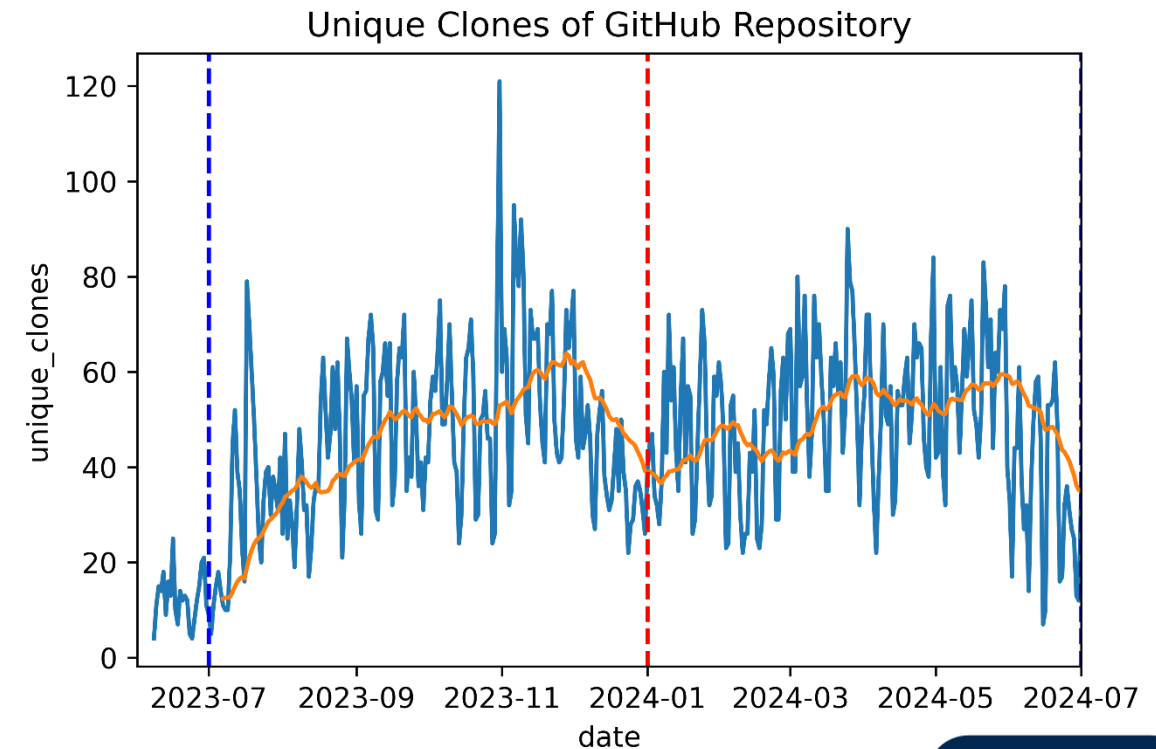
Approx 100 unique contributors last year

Significant diversity in industry, academia, startups, etc.

User community much larger

Approx. 6,500 people on website

Average about 50 clones per day



Open-source license

gem5 is licensed under BSD 3-clause

You can mostly do anything you want with the code

Industry use and contributions have been strong

Over 1000 forks on GitHub



Outline

Start positive (what we did right)

Go a bit negative (what we have done wrong)

Lessons learned



gem5's software architecture

No “official” APIs

Rebasing/updating a time sink

Everything in one binary

1.5 GiB binary... cool.

No distribution of application

Except via source code

```
> sudo apt install gem5  
E: Unable to locate package gem5
```



gem5's software architecture

Use esoteric C++ constructs

Chose a unique build system

What is SCons?

Most of gem5 code in
“abandoned” state

CI tests are Sisyphean

```
// A recursive template which defines the ABI specific implementation
// interface defined above.
//
// The types in Types are consumed one by one, and by
// the time we get down to the base case we'd have lost track of the concrete
// set we need to know what interface to inherit. The Base parameter keeps
// track of that through the recursion.
template <typename ABI, typename Base, typename ...Types>
class VarArgsImpl;

template <typename ABI, typename Base, typename First, typename ...Types>
class VarArgsImpl<ABI, Base, First, Types...> :
    public VarArgsImpl<ABI, Base, Types...>
{
protected:
    // Bring forward the base class constructor.
    using VarArgsImpl<ABI, Base, Types...>::VarArgsImpl;
    // Make sure base class _getImpl-es don't get hidden by ours.
    using VarArgsImpl<ABI, Base, Types...>::_getImpl;

    // Implement a version of _getImpl, using the ABI specialized version of
    // the Argument class.
    void
    _getImpl(First &first) override
    {
        first = Argument<ABI, First>::get(this->tc, this->state);
    }
};
```

“Code” sustainability

Developing the infrastructure is “boring”
And receives little credit

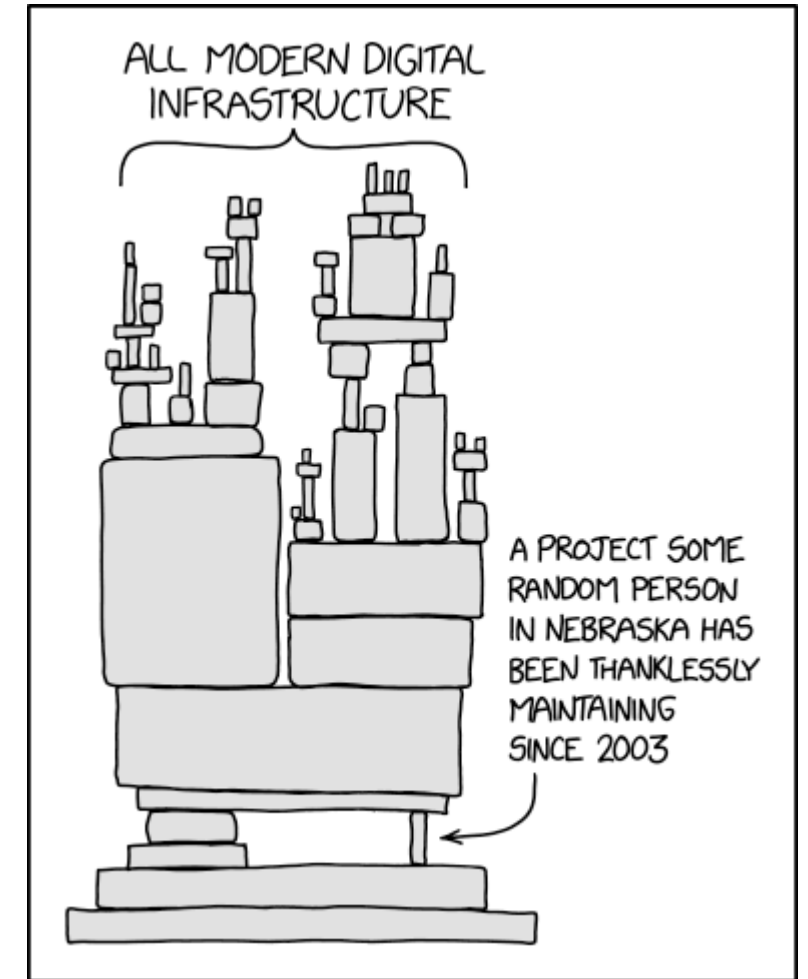
Corporate contributors

Goal: build a thing, add new features

Research contributors

Goal: Discover new thing, add new features

Who has the responsibility to *sustain*?



gem5's community

Lack of diversity of core developers

Very small number of main contributors

Most people contribute once and leave

While users have grown substantially

Developers have not



Open-source license

Oops... It's not “**O**pen **S**ource”

OSI does not approve of our license

Practical effect:

Really hard to join an organization



Outline

Start positive (what we did right)

Go a bit negative (what we have done wrong)

Lessons learned



How to develop an impactful tool

Do share your tool: **Let others use and develop**

Do share your tool as widely and easily as possible!

Do make your tool open source

Market your tool anywhere and everywhere

Websites, tutorials, books, videos, etc.

Do follow software best practices: **Make it easy for others to use your tool**

Do use git, good design practices, ...

Do use code review, CI testing, etc...

Do use the most popular tools for your tool

Do support the tool: **Help others use your tool**

Do provide documentation and support

Do continue development after initial release



DOs



DON'Ts



Make the tool Capital Open Source

Include a LICENSE file with all distributions

Use an OSI approved license (opensource.org)

Industry prefers more permissive licenses

Apache v2, BSD are good choices

Use creative commons for documentation / teaching

As the project grows, the leadership should mature

Governance document defining how to make decisions

Committee for management

Think about the exit strategy

Without an exit strategy the project will languish

Moving under an umbrella

Startup, nonprofit, etc.



open source
initiative[®]



Create a community around a tool

Foster a community so that others can give back

Answer questions when they come up

Mailing list, github issues, slack, etc.

Provide answers to questions *before* they come up

Documentation is hard, but *very* important

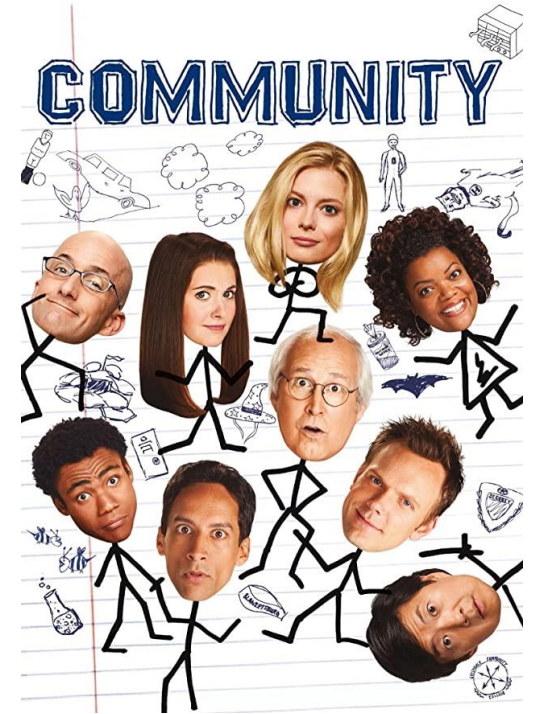
Document for both users and developers

readthedocs.org is a great tool

Include a CONTRIBUTING guide and a CODE-OF-CONDUCT

Make the community *inclusive* and *accepting*

The broader the community the more impact the tool will have





Path forward

Pooling resources

Avoid the tragedy of the commons?

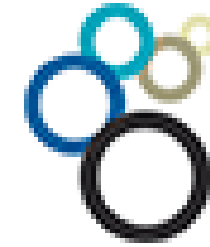
Foundations

Chips Alliance

FOSSi Foundation

OSHWA

OpenHW Group



OPENHW^{GROUP}
— PROVEN PROCESSOR IP —

Not all projects fit with a foundation

Still a lack of investment in infrastructure



Incentivizing sustainability

Give ***credit*** for ***impact***

Many improvements have big impact, but are not credited

- Adding tests

- Improving modularity

- Writing documentation

We have control over this in academia

Less control for profit-driven companies



Conclusions

20+ years of gem5

A sustainability success story!

Good decisions early

Software engineering, community,
documentation, and open source

Need to actively work to create

Sustainable community

Sustainable infrastructure

Incentives are not there

Let's change it 😊



Panel questions

What do you see as the successes and failures of your tool?

From industry, labs, and academia, developer, funder, user perspective

Should we form a ModSim Software Foundation (MSF)?

How can we incentivize people to work together?

Avoid reinventing the wheel

Encourage improving infrastructure

Secret bonus question

