

Creating a **community of contributors** for scientific open-source projects

The **preCICE** case

Gerasimos Chourdakis, Technical University of Munich
www5.in.tum.de/~chourdak

Benjamin Uekermann, Eindhoven University of Technology

Context

preCICE is a free C++ library that couples your simulations

Usually in RSE

- Developers are also users
- Domain experts

Our case

- We are not the target users
- Software/methods specialists
- Munich, Stuttgart, Eindhoven

Why build a community?

Don't get users

- “It’s just a prototype”
- “They will steal my work”
- No plans to continue
- No time to support them

Get users

- Feedback & contributions
- Citations (strong delay)
- Stronger proposals
- Industry funding
- Recruiting
- Career opportunities

Community dream: Users help each other

It's motivating!

After the amount of support I received from this community, I am switching to opensource for every one of my needs.

[@nithinadidela on Discourse](#)

Get your first users

Don't do

- Develop unimportant features
- Convince developers of similar tools
- Go only to conferences in your field

Do

- Add missing components
- Ready-to-use packages
- Easy documentation
- Simple tutorials
- Go to domain conferences
- Jump on existing trains
- Interview users and adapt
- Find your audience & USPs
- Find the right language

Develop the right thing, communicate it right

Communication is crucial

Communication barrier

- Advertise channels to reach you
- Be active on these channels

Direct communication

Low barrier

Does not scale

Repeat for every similar case

Mailing list

Low barrier (?)

Reach inactive users

Fear of bothering

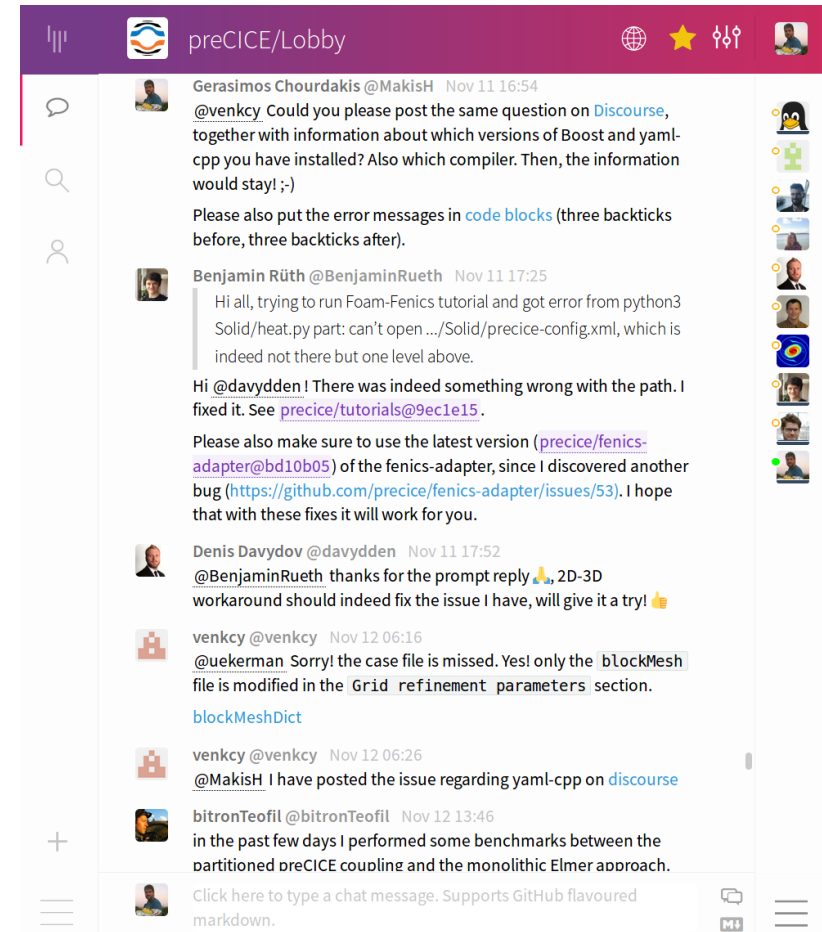
Non-searchable (?)

A thing of the past (?)

Chatroom



Low barrier
Some users stay
Chaotic
Non-searchable



Different kind of questions in chatroom: you need both!

Forum

Structured
 Searchable
 Also serves as FAQ
 Some users stay
 Higher barrier

Obtaining singular values of Jacobian

This is the first time ajaust has posted — let's welcome them to our community!

A ajaust Oct 13

Hi,

could you give me a hint how to dump out information about the Jacobian? I would like to get a better feeling for one of the problems that I try to solve. For that I would like to get a feeling how the singular values look like. The problem is a steady state problem that only depends on the boundary values. This means I do not have to give initial conditions or so. My idea was roughly the following.

1. Set up the simulation to have two "time steps" in preCICE, i.e. setting `<max-time value="2"/>` and `<timestep-length value="1" />` in the preCICE config file. The first preCICE "time step" is used to solve the problem. The second "time step" is purely used to mess around with preCICE.
2. Use the IMVJ postprocessing.
3. Solve the problem until reaching the steady state in the first "time step".
4. Set the options of the IMVJ postprocessing such that drops all singular values at the beginning of the "second" time step. Ideally, it would tell me what the values where.
5. Abort simulation after dumping the information such that "second time step" is never carried out.

Does this approach make sense/work? Can I find a bit more information about the settings for the IMVJ postprocessing than in the XML reference and the IMVJ wiki page?

Thank you in advance!
 Best,
 Alex

Solved by uekerman in post #2

Reply

created Oct 13 last reply Oct 18 3 replies 51 views 2 users 1 like 1 link

Oct 13
 1 / 4
 Oct 13

Oct 18

Disclaimer: We only recently started with this

From user to contributor

Make it easy

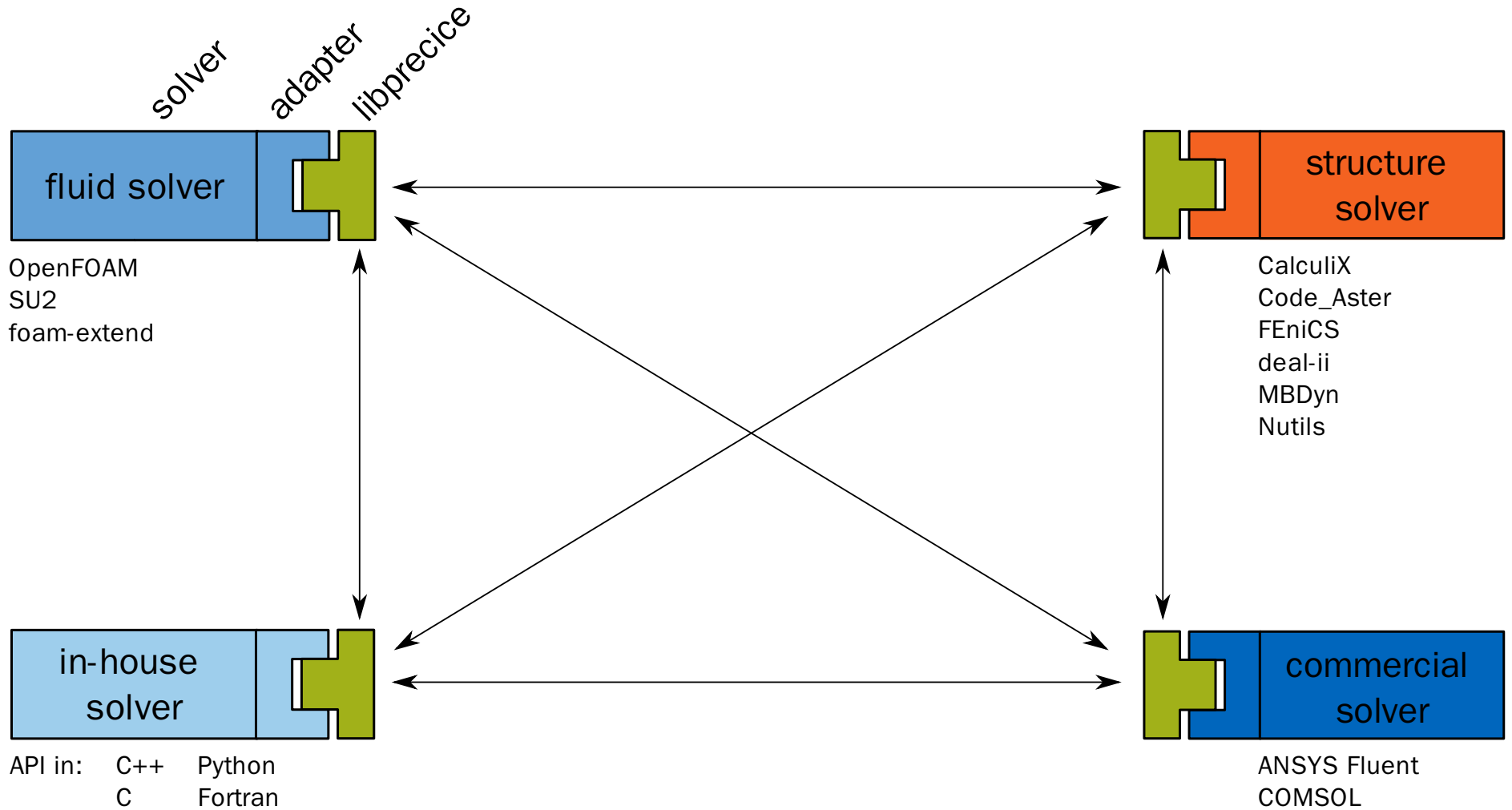
- Advertise where the development is done
 - Choose mainstream platforms
 - Don't require logging in to see the code
 - Don't "code available under request"
- Document architecture
- Setup CI / CD
- Stick to standards

Welcome the tiniest contribution!

Split components

- Easier to contribute to smaller codebase
- Isolated potential damage
- Might solve license issues

preCICE ecosystem

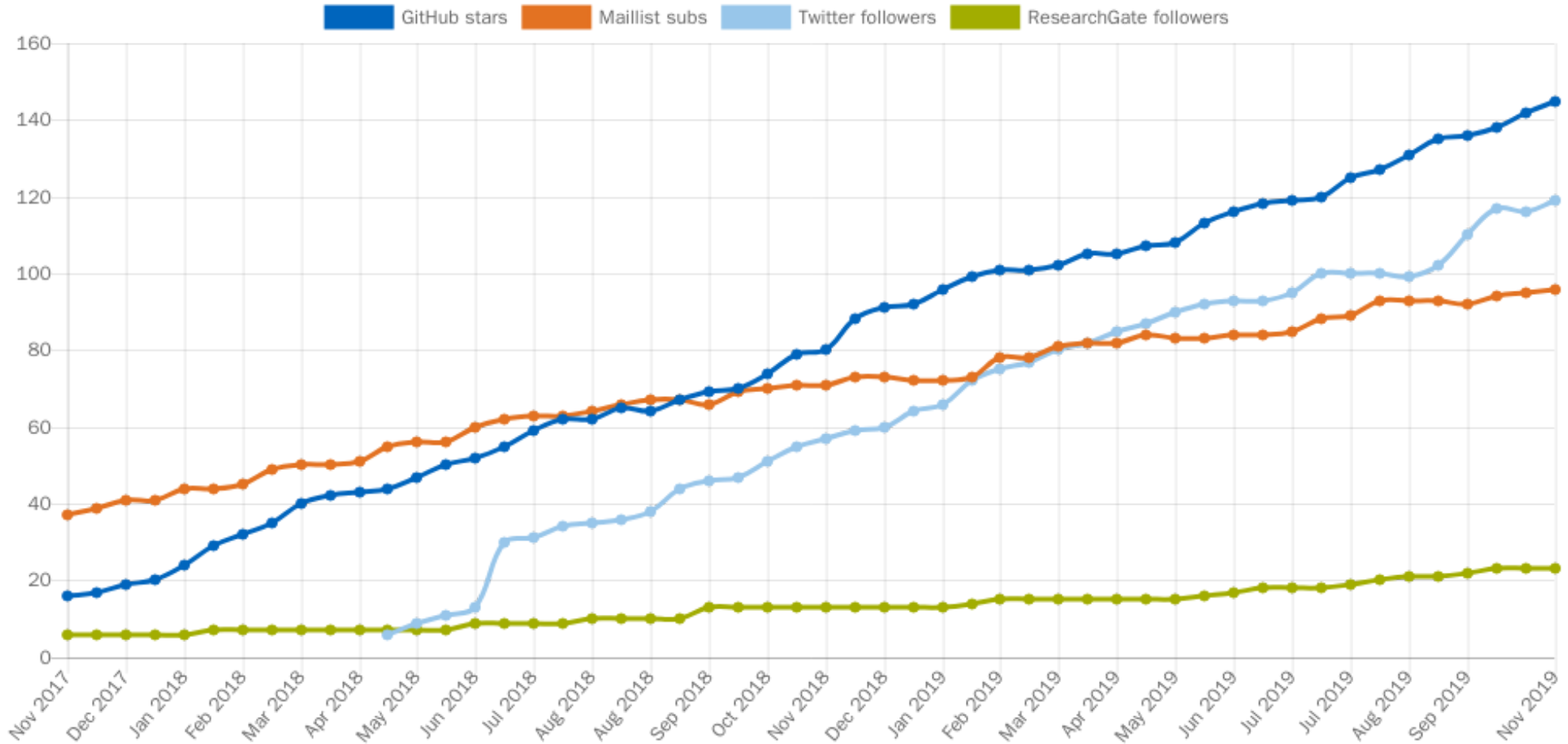


Give motives

- Visibility
- Shared ownership

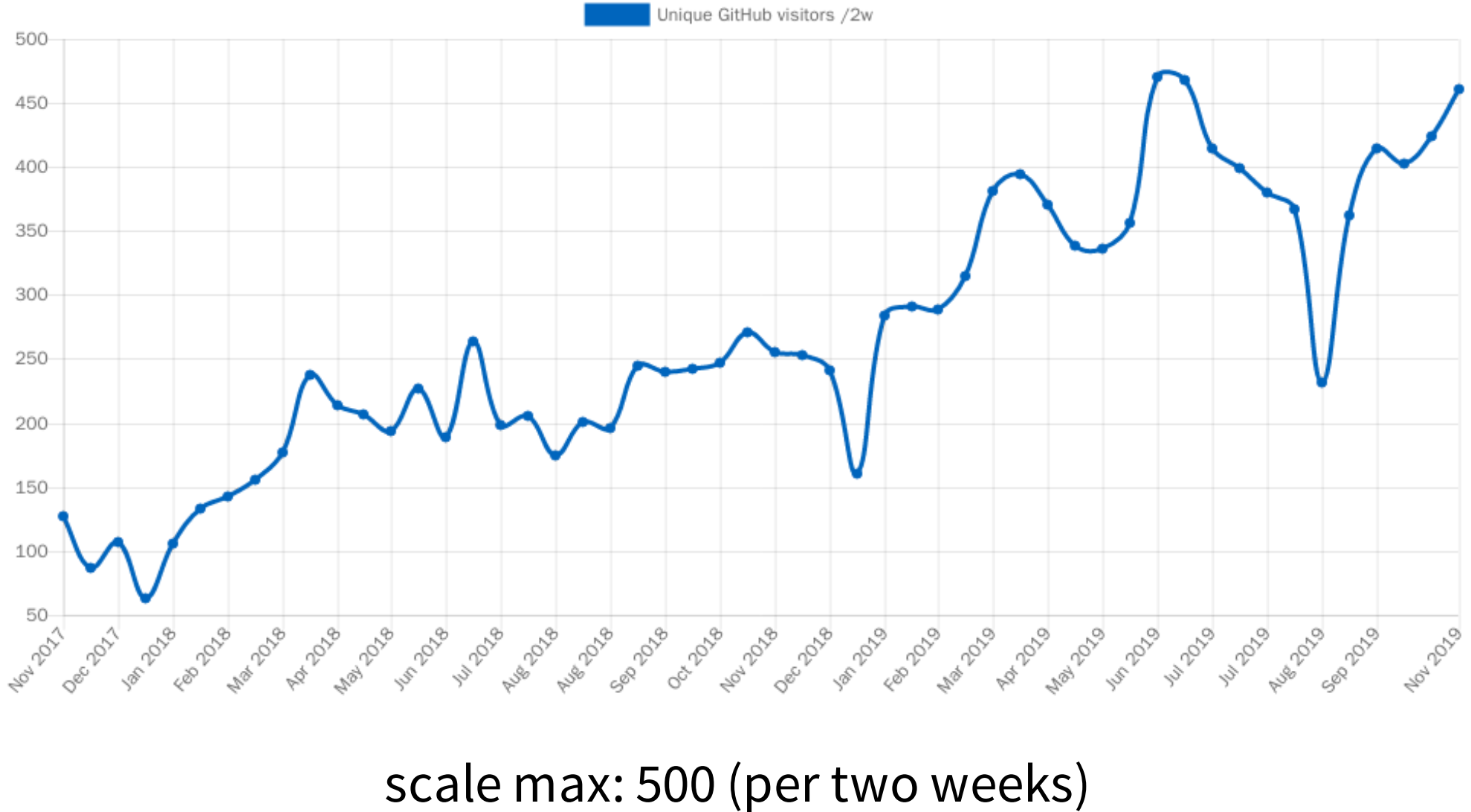
Good community metrics?

Followers: interested in preCICE

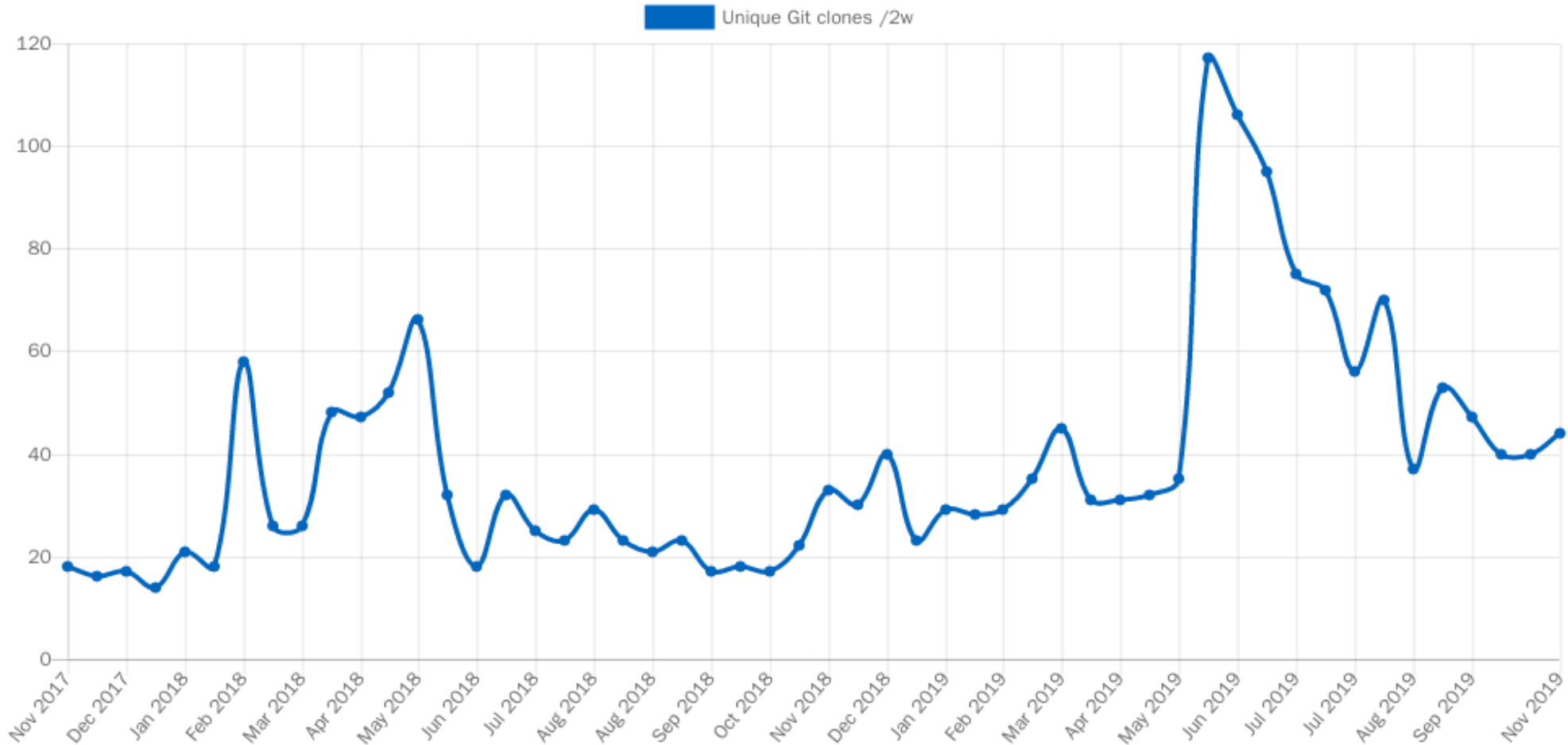


scale max: 160 (cumulative)

Traffic: discovered preCICE

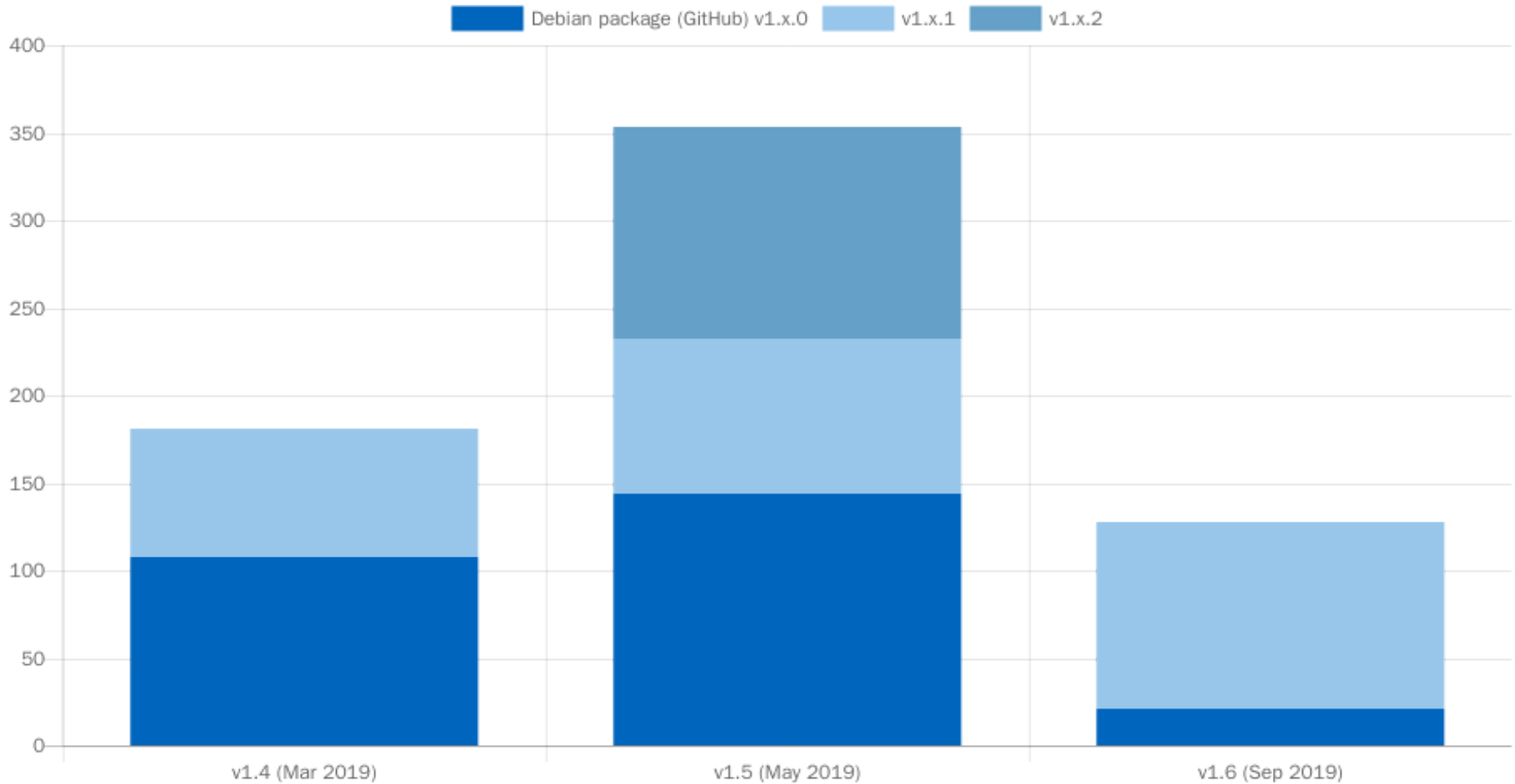


Git clones: trying preCICE from source



scale max: 120 (per two weeks, bots included)

Binary package downloads



scale max: 400 (cumulative)

Community engagement

Repository	Contrib.	External	★	
precice	24	5 (21%)	148	50
openfoam-adapter	6	2 (33%)	46	22
calculix-adapter	7	1 (14%)	7	23
mbdyn-adapter	3	2 (67%)	1	3

Too small numbers to conclude

Community engagement

Pull requests	PR	External	Issues	External
precice	259	10 (4%)	306	53 (17%)
openfoam-adapter	33	12 (36%)	67	18 (27%)
calculix-adapter	6	3 (50%)	15	5 (33%)
mbdyn-adapter	1	1 (100%)	0	0

Too small numbers to conclude

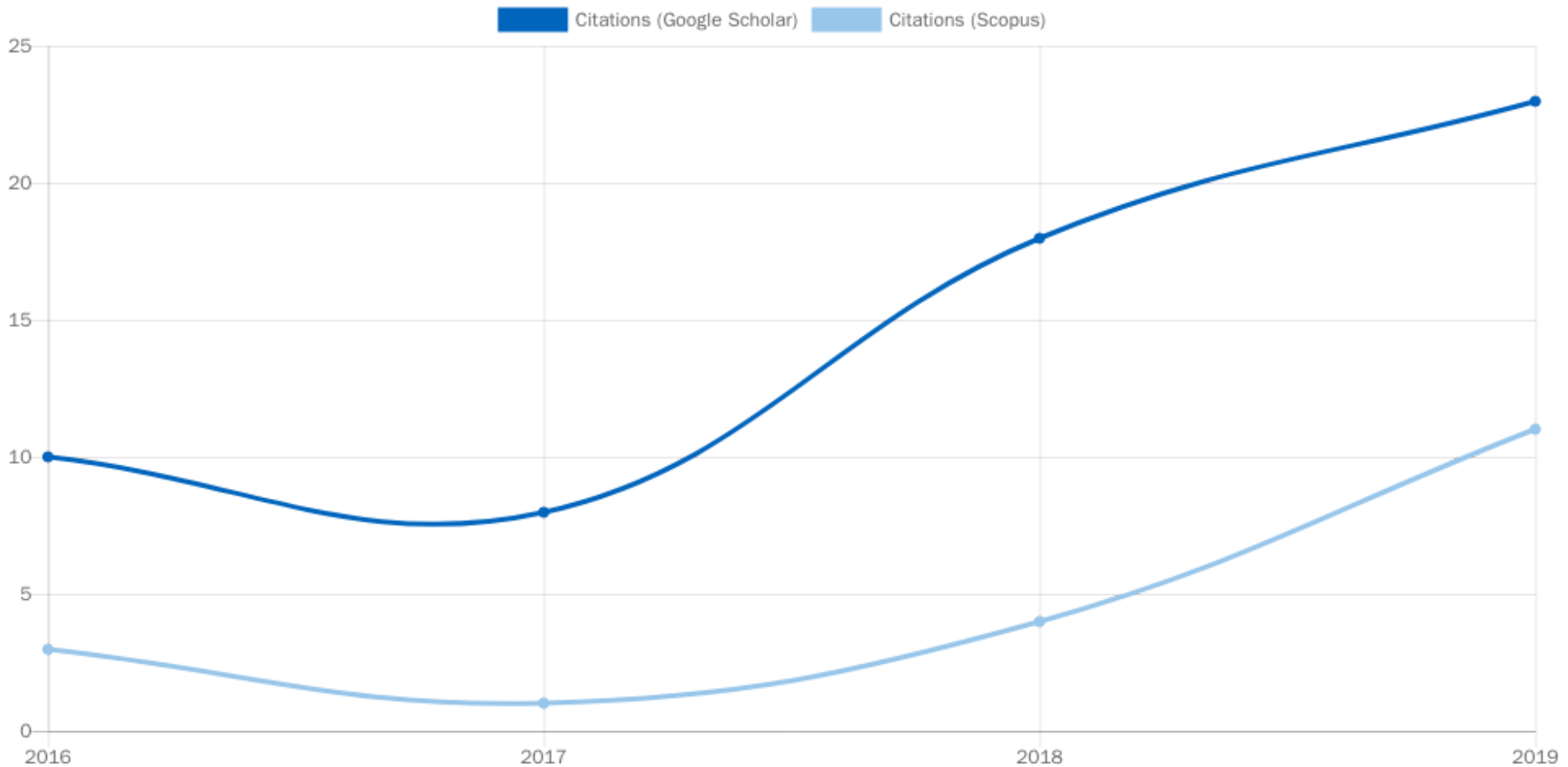
More metrics

- Activity on channels
- Citations
- Activity on channels

Activity on channels

- Gitter: 5 messages / day – 14 active users / month
- Discourse: 3 posts / day (just started)
- Mailing list: 1 email / 2 days

Citations



Main preCICE paper (2016): 61 total citations

Collaborations (which?)

- Known users: 24 academic & 9 industrial groups
- Testimonials: 11

Summary

Help the users help themselves!

preCICE website: www.precice.org - Twitter: [@preCICE_org](https://twitter.com/preCICE_org)

Gerasimos Chourdakis (TUM)
Benjamin Uekermann (TU/e)
+ many more



This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).
Get this presentation from [GitHub](https://github.com).

 Come to our poster (preCICE, sustainability, ...)