FabSim

Derek Groen, Agastya Bhati, James Suter, James Hetherington, Stefan Zasada, Peter Coveney

Centre for Computational Science, UCL
Department of Computer Science, Brunel

Research Programming Seminar 11th Jan 2016

UCL

Computational research is tricky

- We want to do work of “excellent” quality.
- Chart out problems we’ve never tackled before.
- Do computations of unprecedented size and complexity.
- Using state-of-the-art, high quality research production codes.

The reality

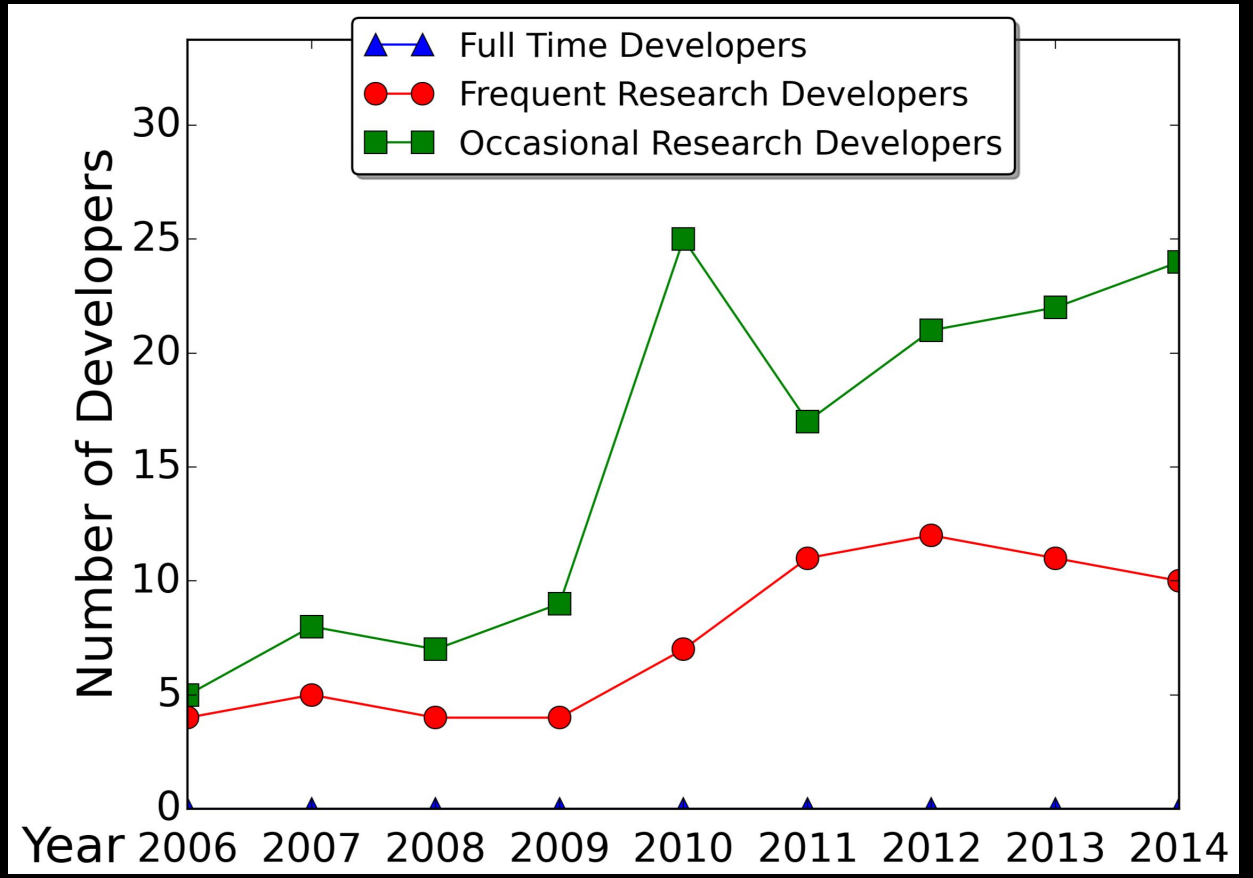
- Projects of limited size, limited duration.
- PhD-ware, *Titanic*-ware.
- Hofstadter’s law.
  - “It always takes longer than you expect, even when you take into account Hofstadter's Law.”
- Research self-gravitation.
  - Focus research on the relatively easy/few features that we *did* manage to get working.

Why do we as academics suck so much, while industry frequently does pull it off to create something solid?

The curious case of academic software

ESPResSo

(major CFD code developed in Germany)

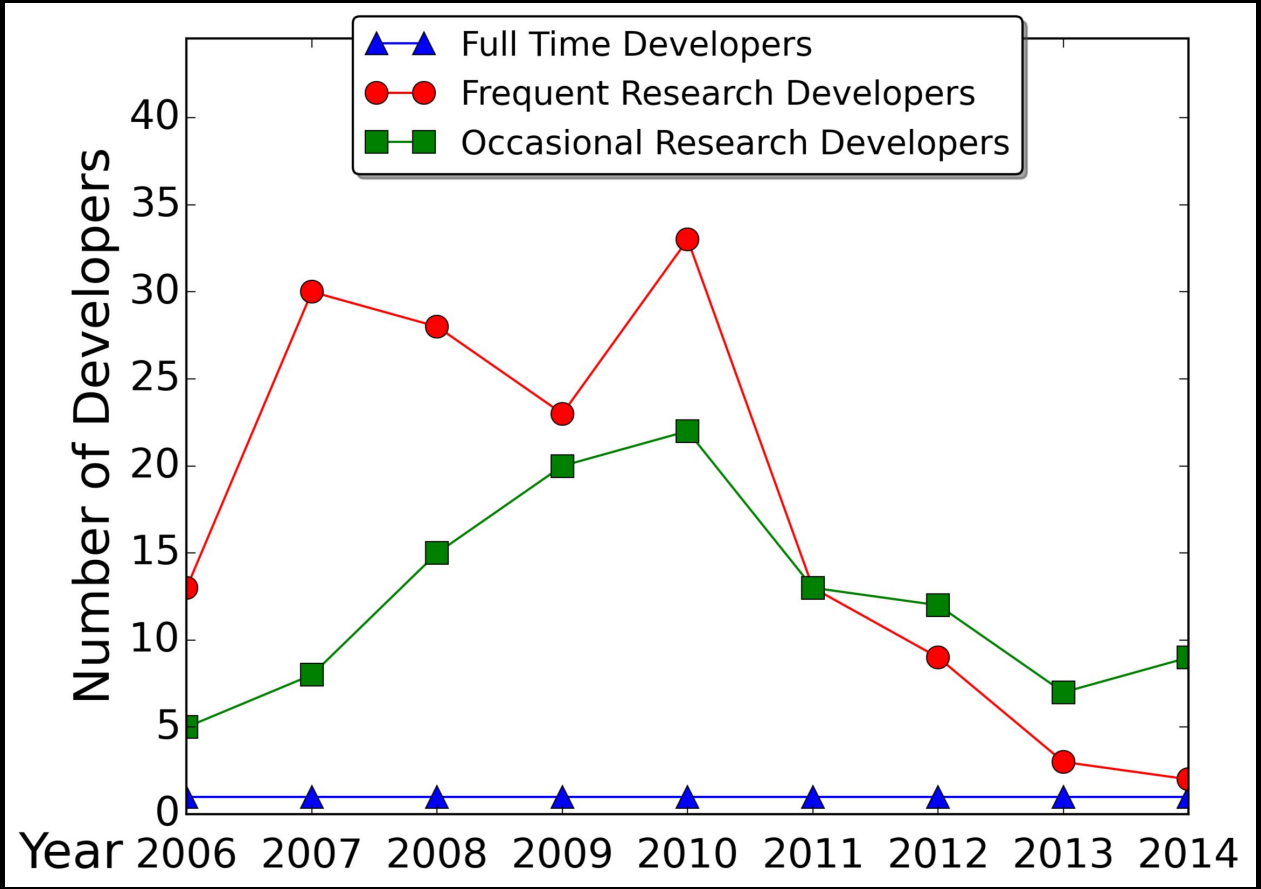


Groen et al., arXiv:1506.05272

The curious case of academic software

Fluidity

(major CFD code developed in the UK)



Multi-tasking researcher-developers: my perspective

- Developing software.
  - years...
- Coupling codes
- Finding correct and relevant simulation configurations.
  - sometimes years..

- Managing the execution of simulations.
- Analyzing the (big) data.
  - months, years?
- From code line #1 to publication #1.
  - 2+ years?

Where do we find the human effort to do all this?

~~Big data~~

~~Extreme scaling~~

~~End-to-end solutions~~

Small Effort

FabSim: aiming to mitigate Small Effort

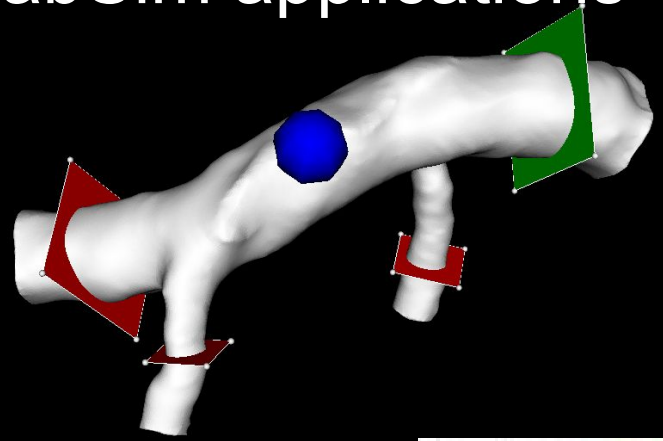
- Save time.
- Automate frequently used patterns in computational research.
- Provide quick one-liners to use patterns, or composite ones.
- Prevent double-work, by curating data automatically.
- Make software for the user-developer, not the end-user.

FabSim: an approach to mitigate Small Effort

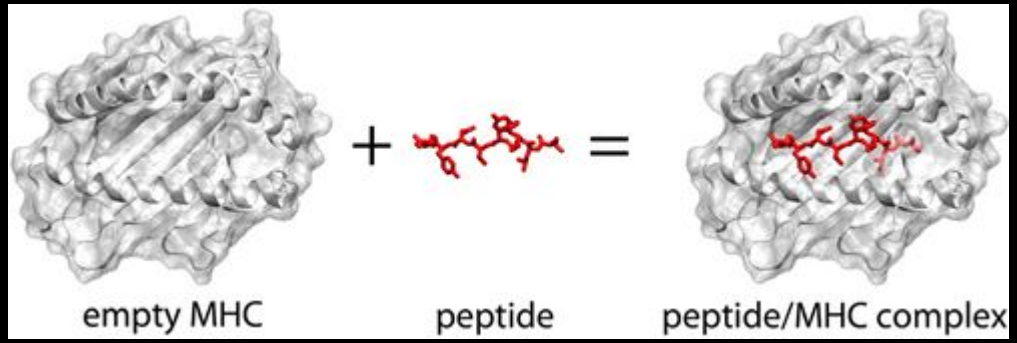
- Written in Python using YaML and Fabric (www.fabfile.org)
- Optimized to be highly customizable.
- General-purpose software, with special-purpose adaptations.

- www.github.com/UCL-CCS/FabSim

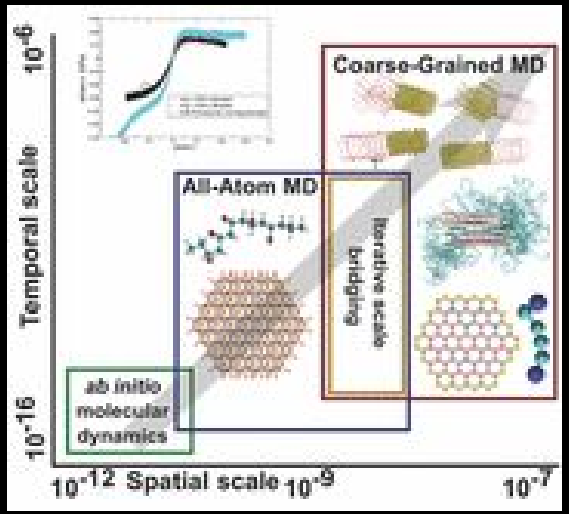
FabSim applications



FabHemeLB



FabBioMD



FabMD

Simple commands

```
fab archer cold
```

Compile the HemeLB code on ARCHER.

```
fab <machine> <code_name>:
<config_name>
fab <machine> fetch_results
```

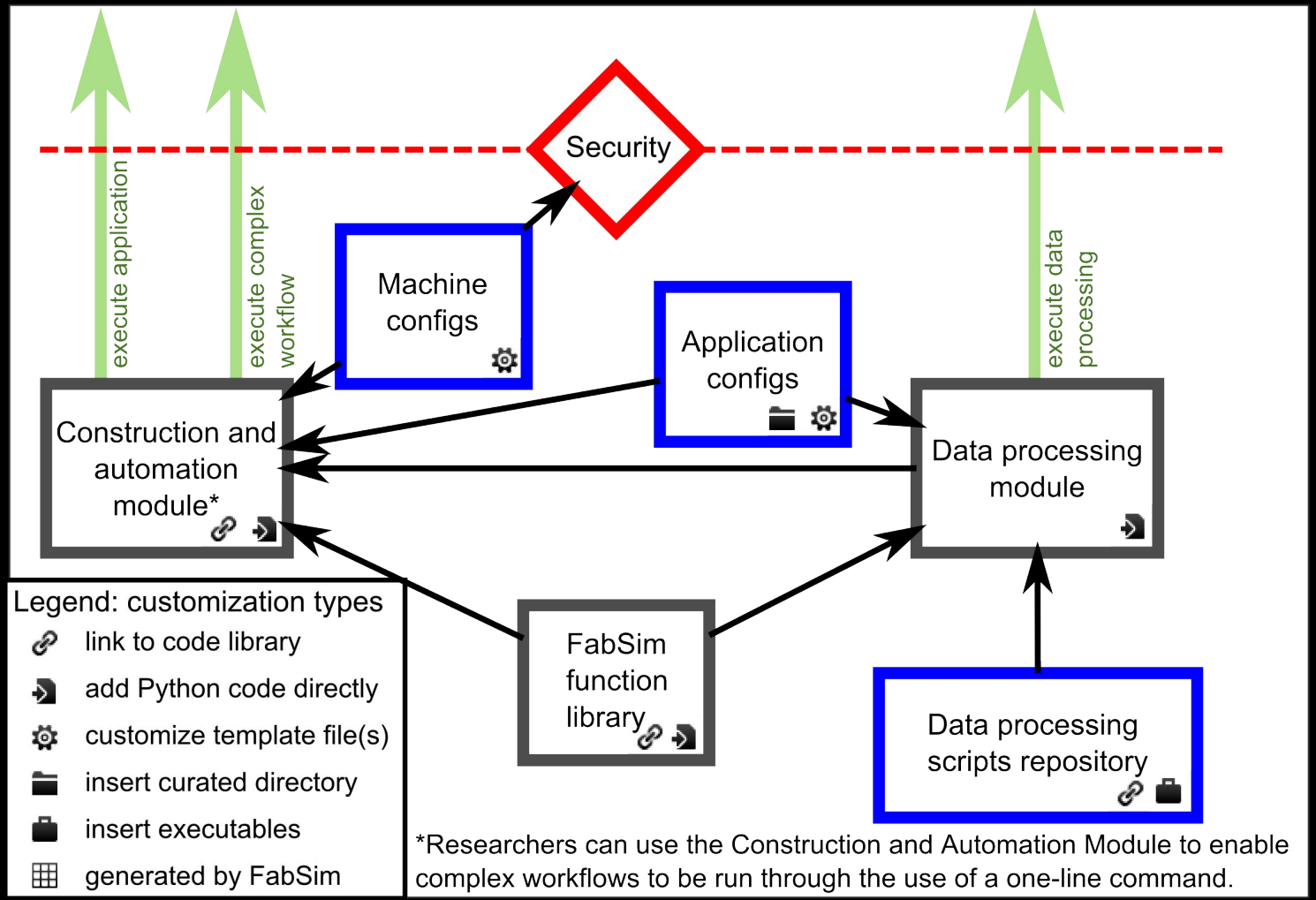
Run <code_name> with <config> on <machine_name>.
Fetch results from the runs from <machine_name>.

```
fab bluejoule lammps:h2osystem,\
cores=128,label=validation
```

Run LAMMPS on BlueJoule to simulate some atomistics system of water using 128 cores, and prefix all directories with the label “validation”.

FabSim

Your local user
Linux machine
(user-space)



FabSim example

Install HemeLB on a new machine, manually

1. Installs dependencies. (3h)
2. Installs HemeLB. (3h)

1. Asks user 1 for advice.
2. Installs dependencies. (1.5h)
3. Installs HemeLB. (1h)

1. Doesn't ask for advice.
2. Installs dependencies. (3h)
3. Installs HemeLB. (3h)

1. Novice user, doesn't ask for advice.
2. Installs dependencies. (12h)
3. Installs HemeLB. (6h)

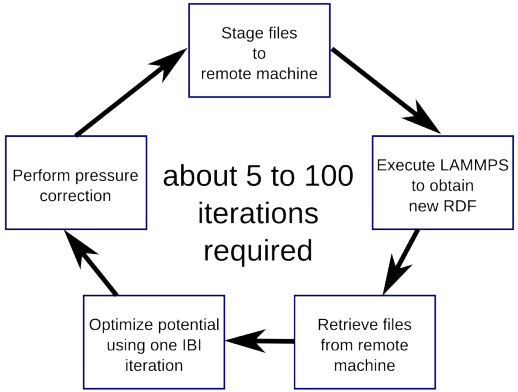
Install HemeLB on a new machine, using FabSim

1. Installs dependencies. (3h)
2. Installs HemeLB. (3h)
3. Records installation in FabSim. (10 min)

1. fab <machine> cold (1h)

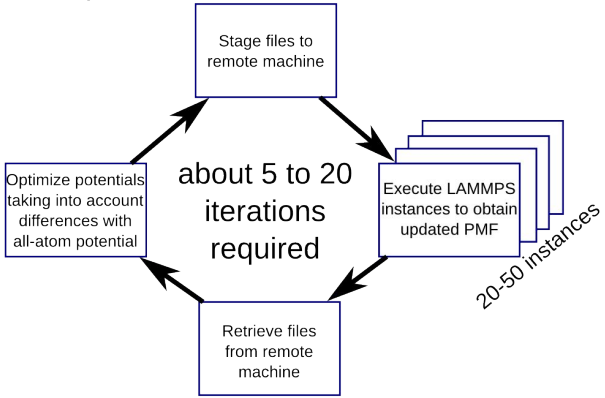
(a) Iterative Boltzmann Inversion

`fab <host> full_ibi_multi`

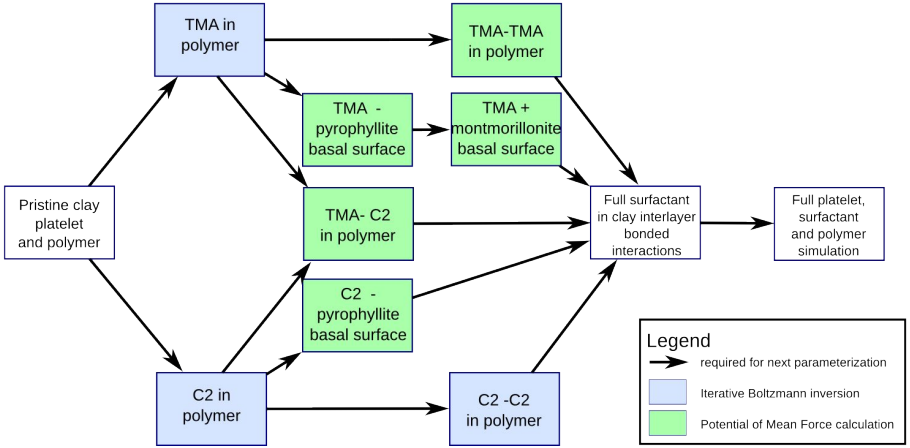


(b) Calculating the Potential of Mean Force

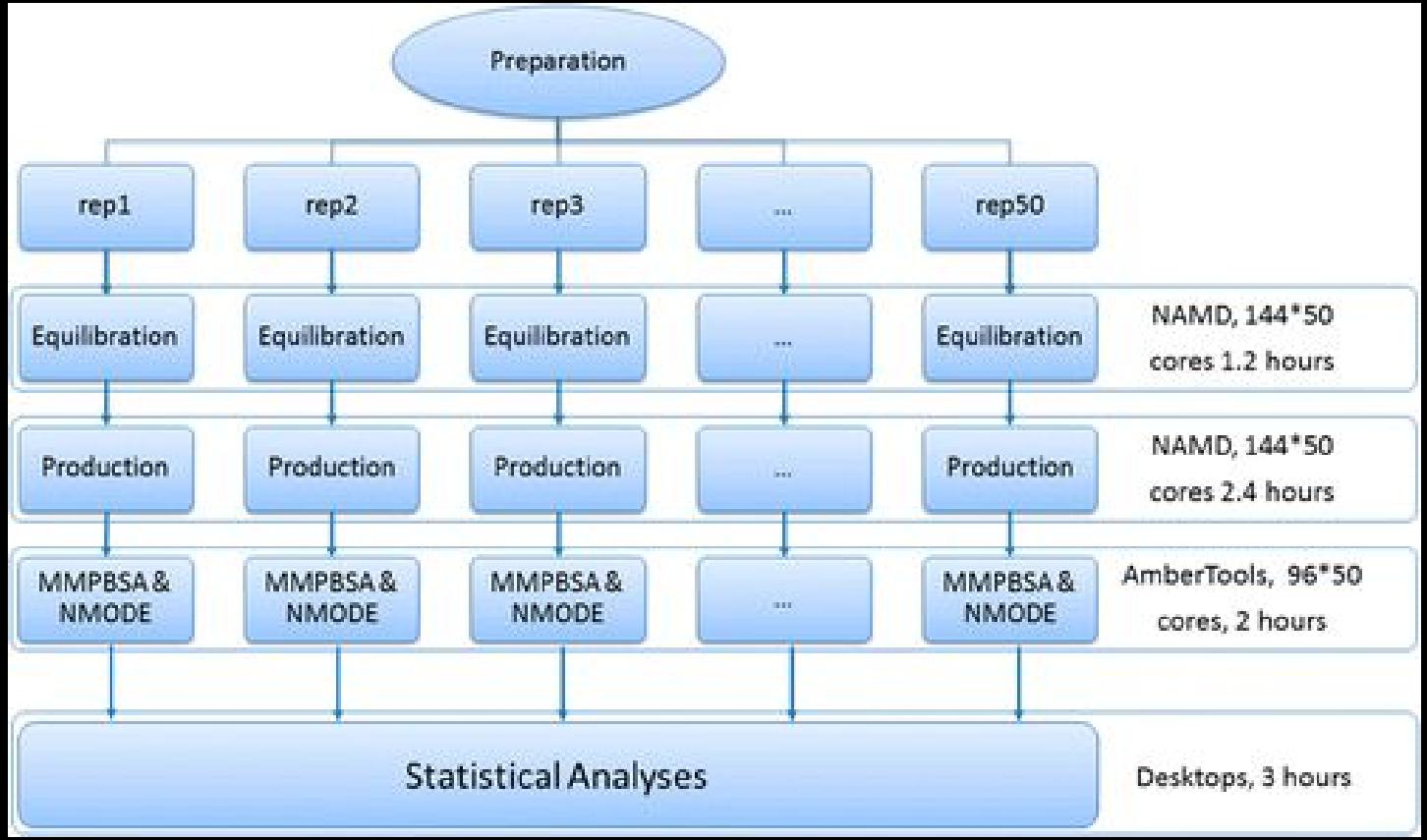
`fab <host> full_pmf_multi`



(c) Coarse-grained parameterization



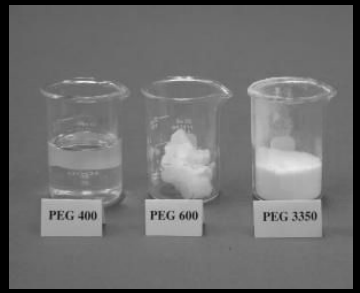
>1320 runs...
FabSim made this doable



Source: Wan et al., 2015.

Thank you for your time

This work was funded by the EU FP7 projects MAPPER and CRESTA, the EPSRC funded 2020 Science project and by the Qatar National Research Foundation.



Clay

Multiscale applications: Clay-polymer nanocomposites

Aim: Develop quantitative coarse-grained models of clay-polymer nanocomposites.
Uses:

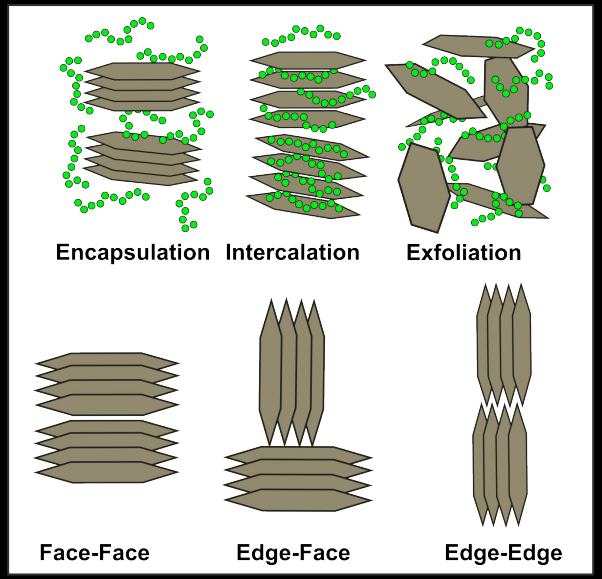
- Predict thermodynamically favourable states.
- Predict elasticity.

We require:

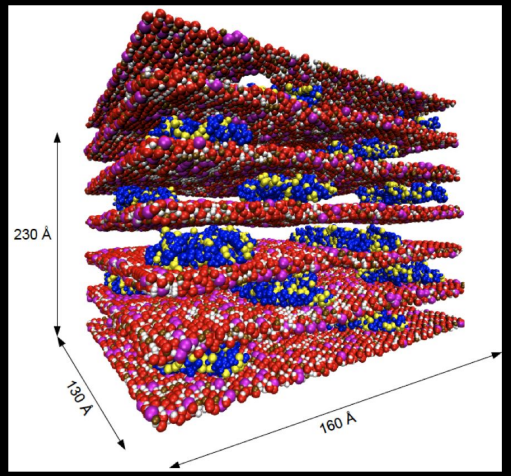
- Accurate potentials.
- Realistic structures.
- Representative time scales.

Suter, Groen and Coveney, Adv. Mat. 27 (6) 966-984, 2015.
Suter, Groen and Coveney, Nano Letters (ASAP), 2015.

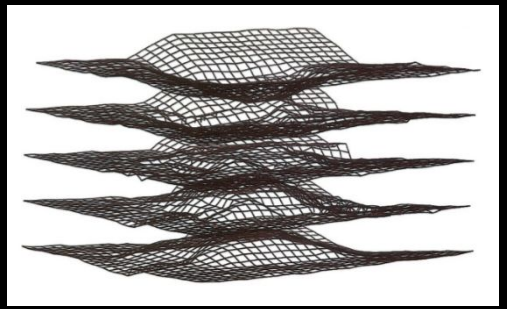
Nanocomposites



What we can do with single scale models

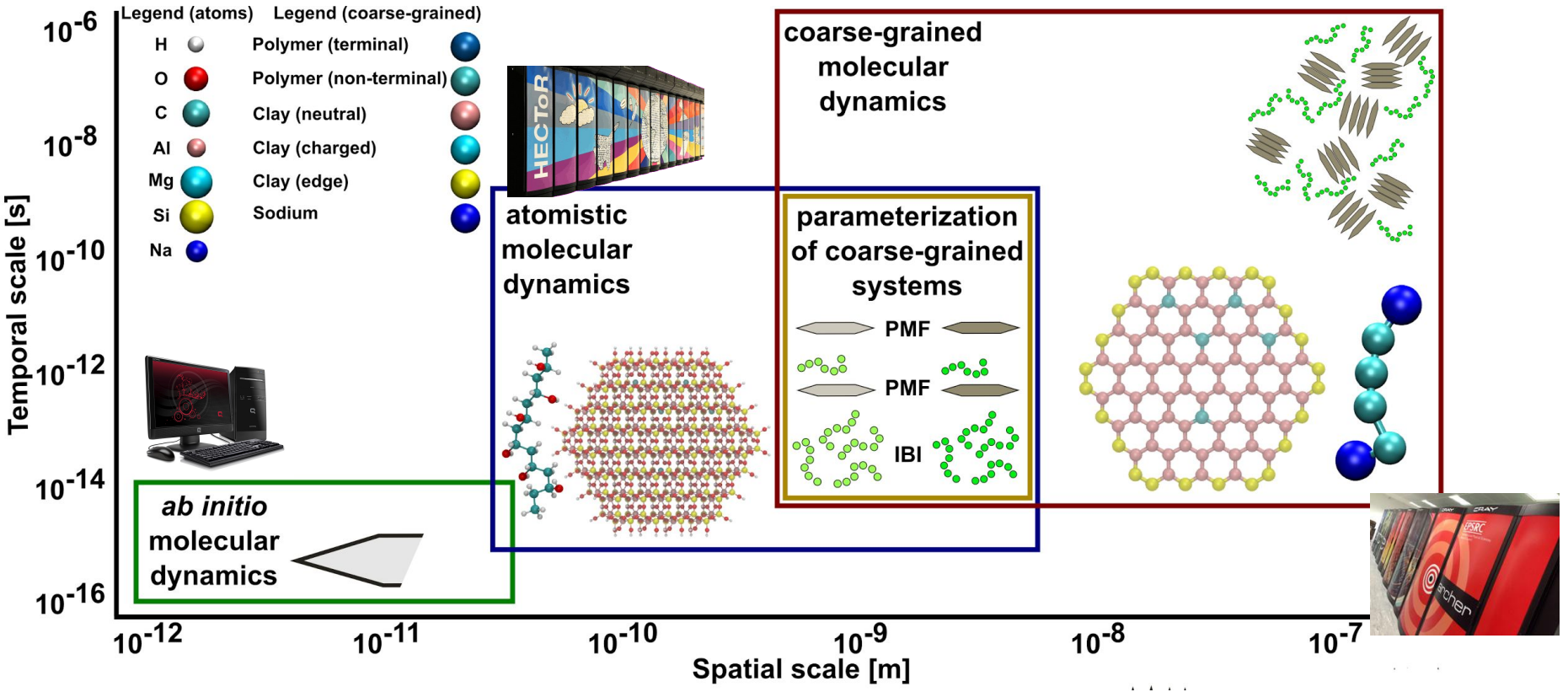


Suter et al., 2009



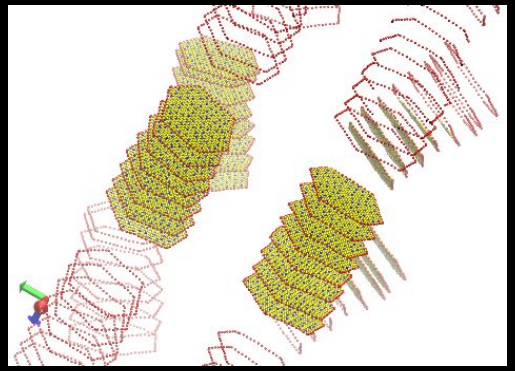
Sinsawat et al., 2003

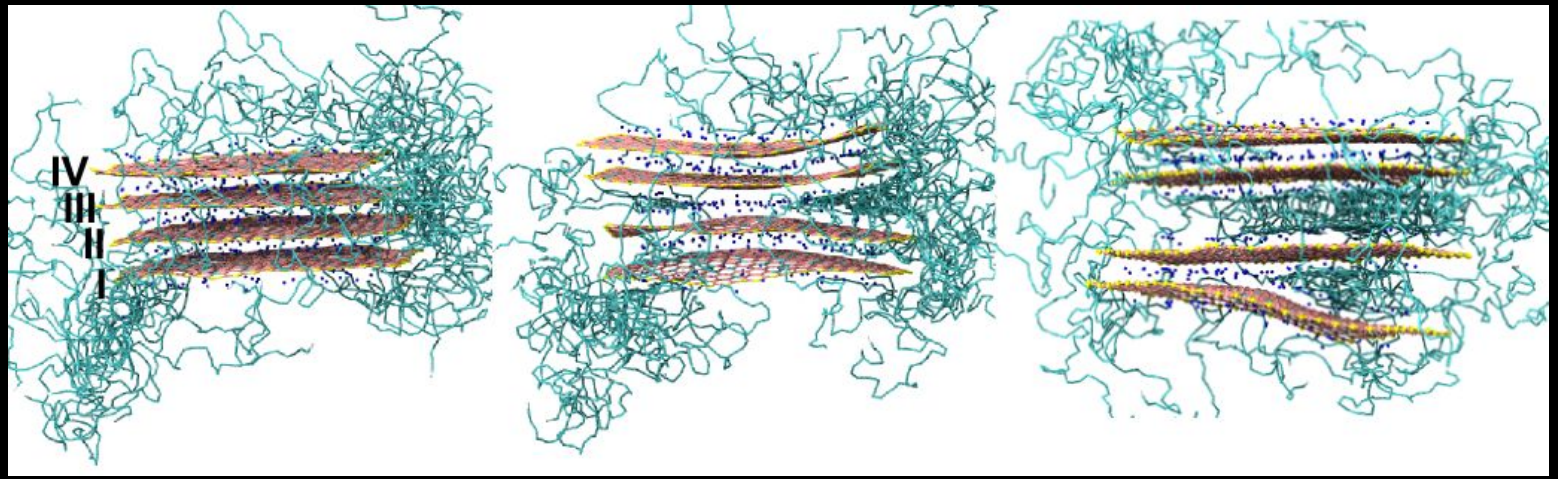
Our multiscale approach

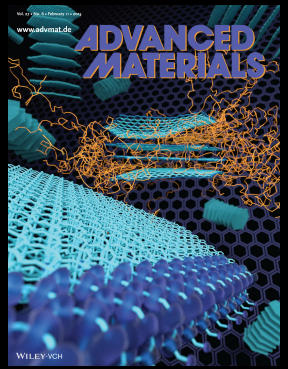


Sample of the Results

- Montmorillonite clay + Polyvinyl alcohol (PVA).
- Polymers have 30 monomers.
- Simulation box contains 8 tactoids, each with 4 platelets.

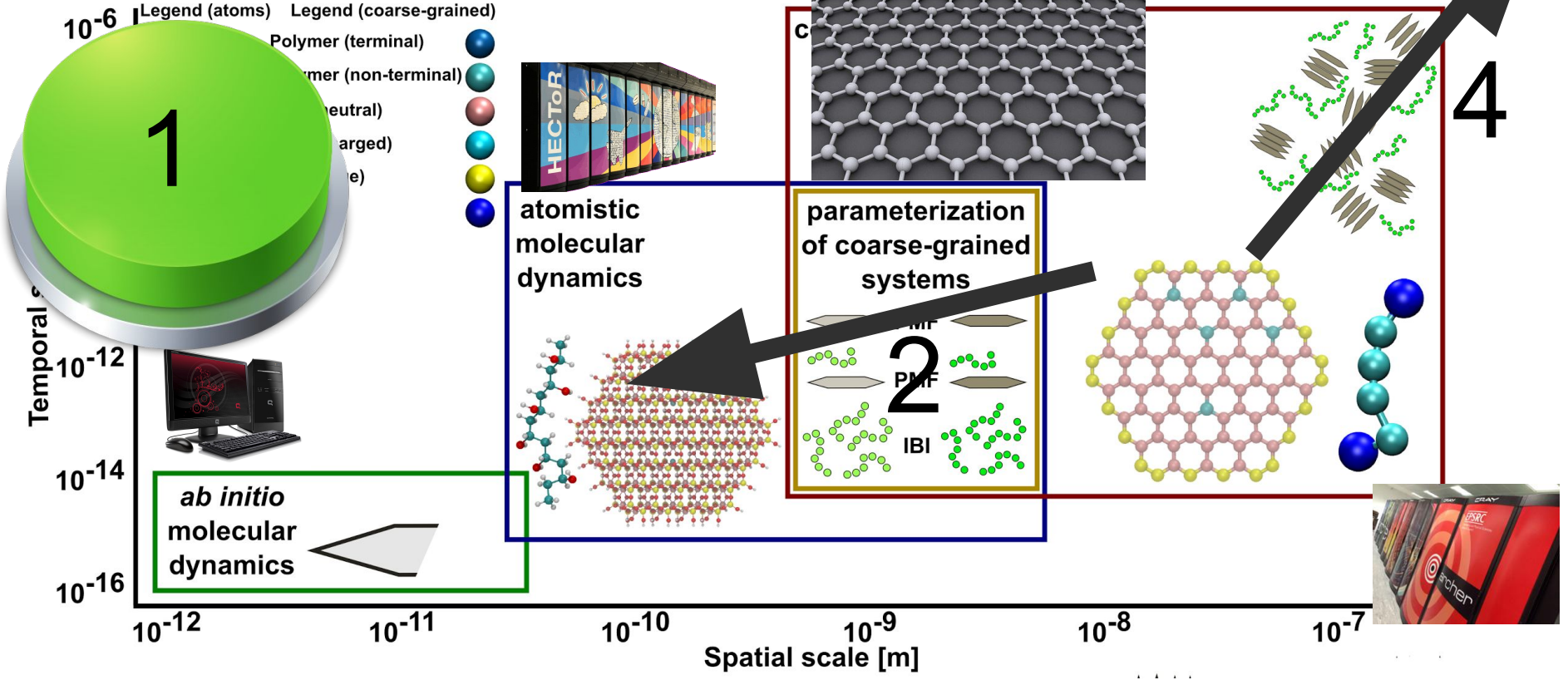






Suter, Groen and Coveney, Adv. Mat. 27 (6) 966-984, 2015.

Future Work


Legend (atoms)
Legend (coarse-grained)
Polymer (terminal)
mer (non-terminal)
eutral)
arged)
e)
1
HECToR
atomistic
molecular
dynamics
parameterization
of coarse-grained
systems
PMF
2
IBI
3
4
ab initio
molecular
dynamics
Temporal
10-6
10-12
10-14
10-16
10-12
10-11
10-10
10-9
10-8
10-7
Spatial scale [m]
EPSRC
archer