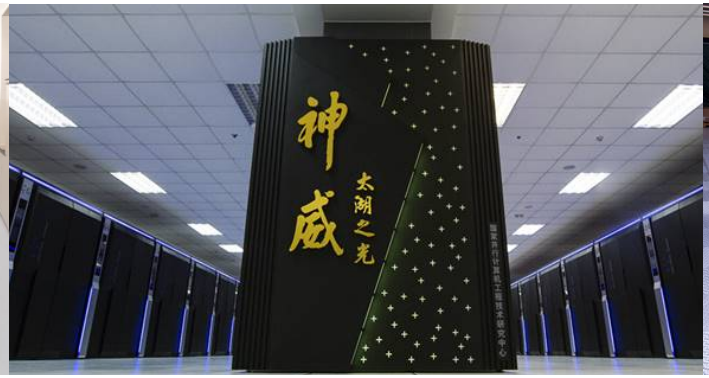
SAAP: SimGrid As A Platform

Seminaire ingénieurs ADT

October 18., 2018

Modern IT Systems

Huge Heterogenous Systems

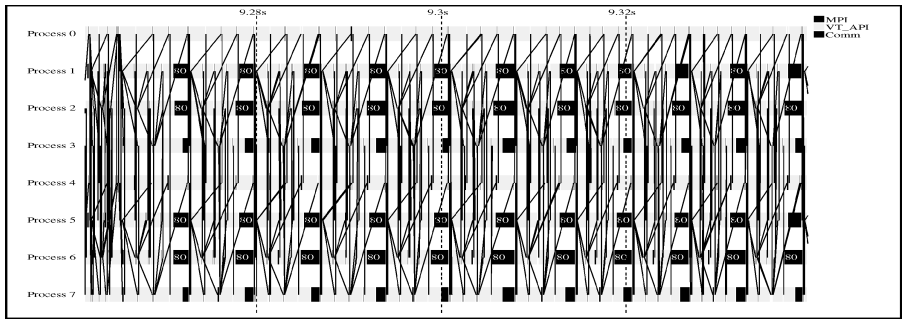


#1 Summit
2,282,544 cores
4608×(2×22-cores + 6GPU)
122 Tflops, 9MW

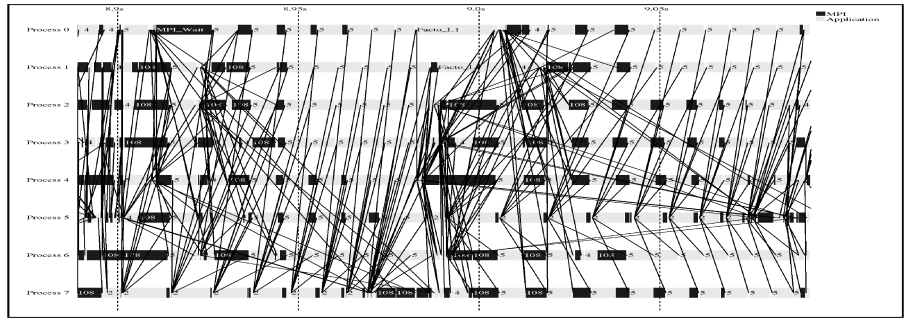
#2 Taihu Light
10,649,600 cores
40 960× 260-cores RISCs
93 Tflops, 15MW

#3 Sierra
1,572,480 cores
4300×(2×22-cores + 4GPU)
71 Tflops, 12MW

Complex Dynamic Applications



Rigid, Regular, Hand-tuned Comm Patterns



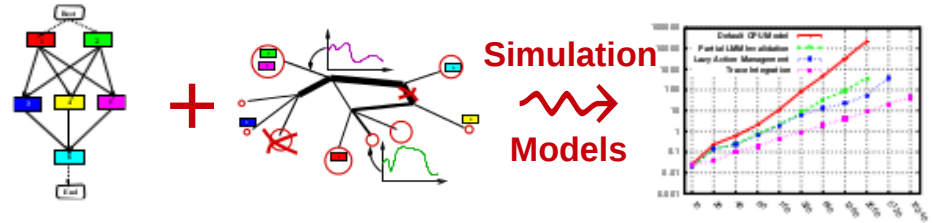
Dynamic, Irregular (task-based?)

How do we study these beasts?

Simulating Distributed Systems

Simulation: Fastest Path from Idea to Data

- Test your scientific idea with a fast and confortable scientific instrument



Idea or MPI code

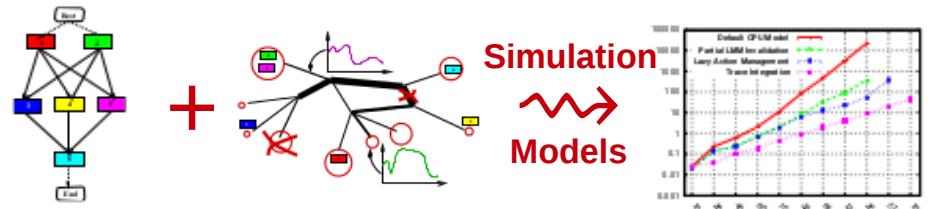
Experimental Setup

Scientific Results

Simulating Distributed Systems

Simulation: Fastest Path from Idea to Data

- Test your scientific idea with a fast and confortable scientific instrument



Idea or MPI code | Experimental Setup | Scientific Results

Simulation: Easiest Way to Study Real Distributed Systems

- Centralized, Reproducible, Clairevoyance, *What if* studies, No Heisenbug
- Hard/soft co-design and capacity planning

SimGrid: Versatile Simulator of Distributed Apps

Install a Scientific Instrument on your Laptop

Computational Science of Computer Science

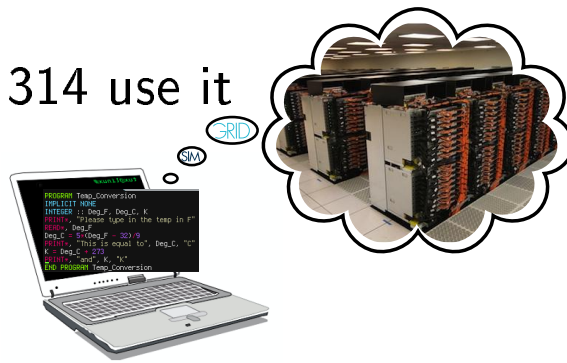
- Joint Project since 1998, mostly from French institutions
- Open Project, contributors in the USA (UHawaii, ISI, NEU), UK, Austria, Cern

Key Strengths

- Performance Models validated with Open Science $\rightsquigarrow$ Predictive Power
- Architectured as an OS $\rightsquigarrow$ Efficiency; Performance & Correction co-evaluation
- Usability: Fast, Reliable, User-oriented APIs, Visualization
- Versatility: Advances in HPC modeling reused by Cloud users

Community

- Scientists: 500+ publications only cite it, 58 extend it, 314 use it
- Apps/Model co-dev : StarPU, BigDFT, TomP2P
- Some industrial users on internal projects (Intel, Bull)
- Open Source: external Power Users (fixes & models)



ADT SimGrid As A Platform (SAAP)

How to ensure the Software Sustainability

- Beyond scientific projects (ANR, IPL): best transfer strategy = open access
- Engineering tasks currently handled by scientists (IJD not enough)

  Cultivate our garden: simplify everything to grow further

- Time consuming but rewarding: huge competitive advantage in science

ADT SimGrid As A Platform (SAAP)

How to ensure the Software Sustainability

- Beyond scientific projects (ANR, IPL): best transfer strategy = open access
- Engineering tasks currently handled by scientists (IJD not enough)

  Cultivate our garden: simplify everything to grow further

- Time consuming but rewarding: huge competitive advantage in science

Announced Workplan

- Next Generation API (SimGrid 4) $\rightsquigarrow$ Build Your Own Simulator
- Add callbacks for plugins, rework modularity for power users
- Improve examples and documentation for newcomers
- Provide compatibility layers to other simulators (PeerSim, DCSim)

ADT SimGrid As A Platform (SAAP)

How to ensure the Software Sustainability

- Beyond scientific projects (ANR, IPL): best transfer strategy = open access
- Engineering tasks currently handled by scientists (IJD not enough)

  Cultivate our garden: simplify everything to grow further

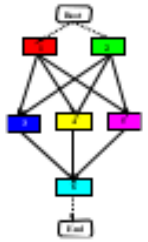
- Time consuming but rewarding: huge competitive advantage in science

Announced Workplan

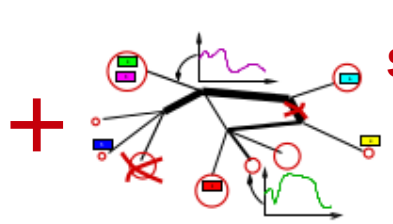
- Next Generation API (SimGrid 4) $\rightsquigarrow$ Build Your Own Simulator
- Add callbacks for plugins, rework modularity for power users
- Improve examples and documentation for newcomers
- Provide compatibility layers to other simulators (PeerSim, DCSim)

Increase Trust to Increase Community

Methodological Challenges raised



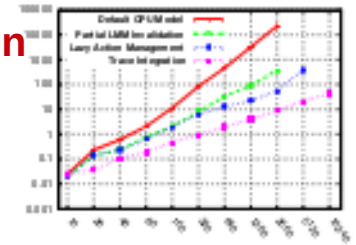
Idea or
MPI code



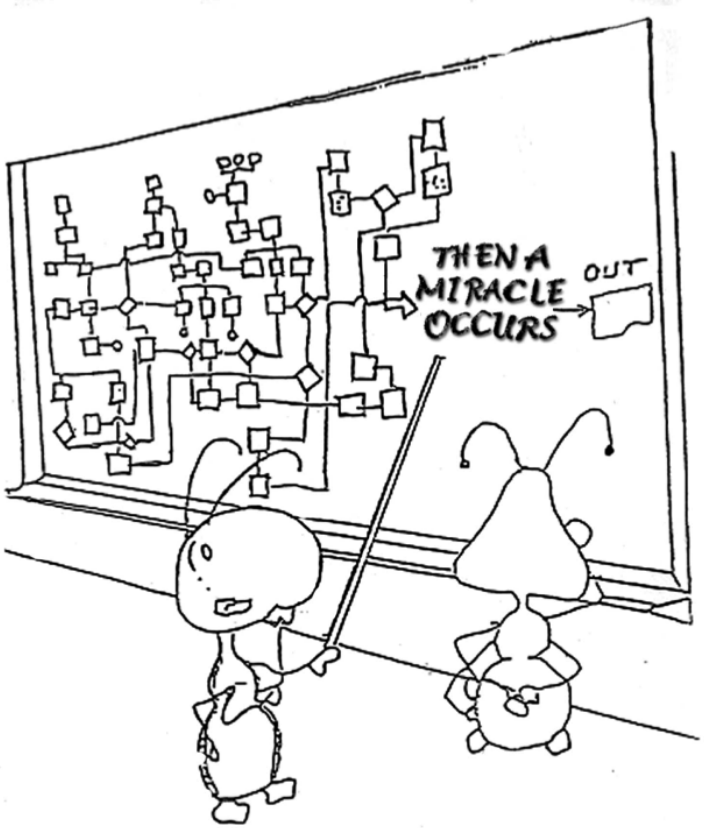
Experimental
Setup

Simulation

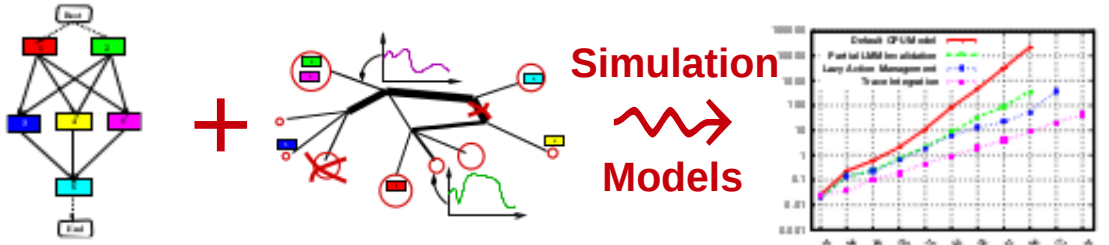
Models



Scientific Results



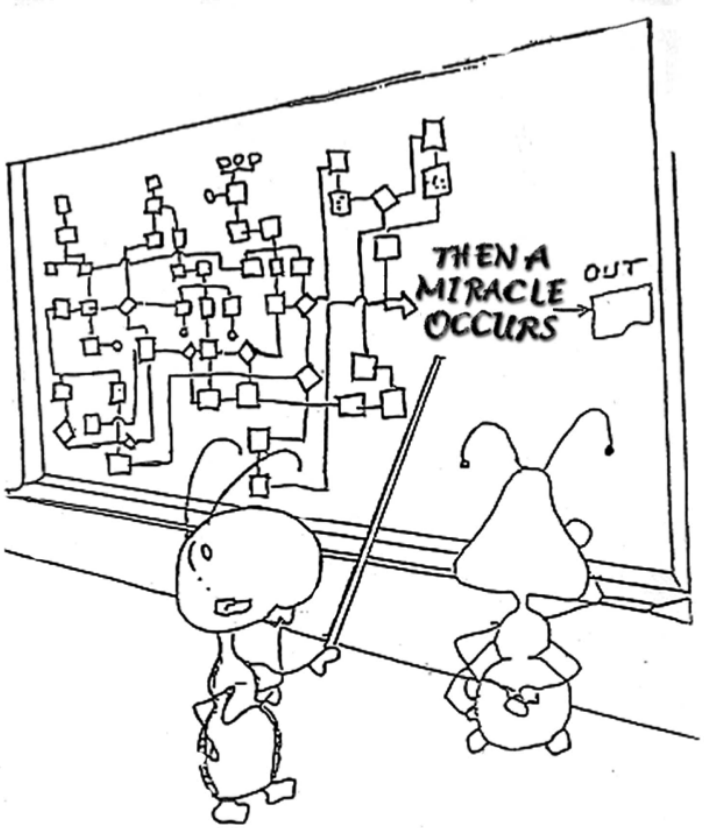
Methodological Challenges raised



Idea or MPI code

Experimental Setup

Scientific Results



Simulator Expected Qualities

- Capability: Abstraction purpose
- Software Accuracy: Q&A, Testing
- Data Accuracy: Valid Platform Models
- Result Accuracy: Validated Results
- Usability: Don't fool the users, doc

(from *Evaluating Simulation Credibility*, US Defense Technical Information Center, 2001)

Why should you Trust Simulation Results?

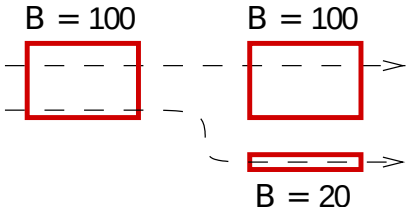
Why should you Trust Simulation Results?

You should not.

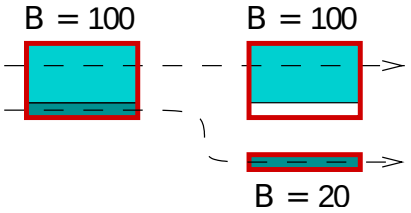
Why should you Trust Simulation Results?

You should not.

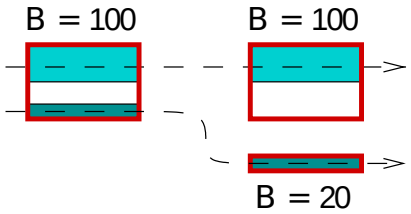
Model Limit: Heterogeneity (Narses, OptorSim, GroudSim)



Setting



Expected Result

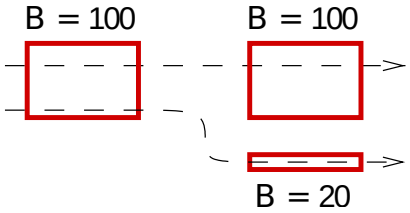


Actual Result

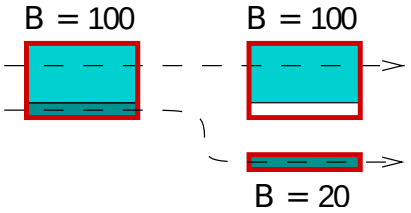
Why should you Trust Simulation Results?

You should not.

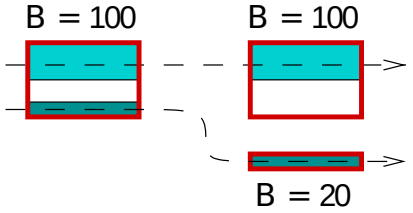
Model Limit: Heterogeneity (Narses, OptorSim, GroudSim)



Setting

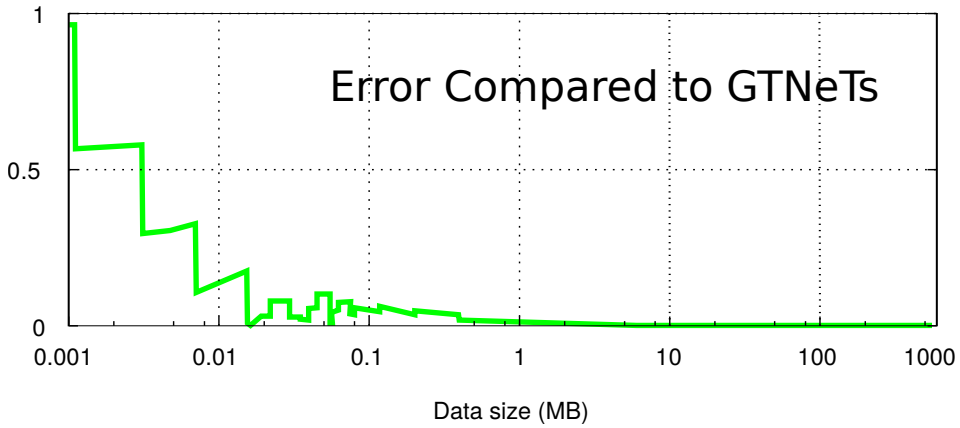


Expected Result



Actual Result

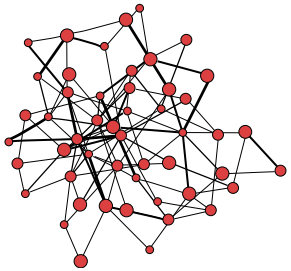
Model Limit: Slow Start (SimGrid without SMPI)

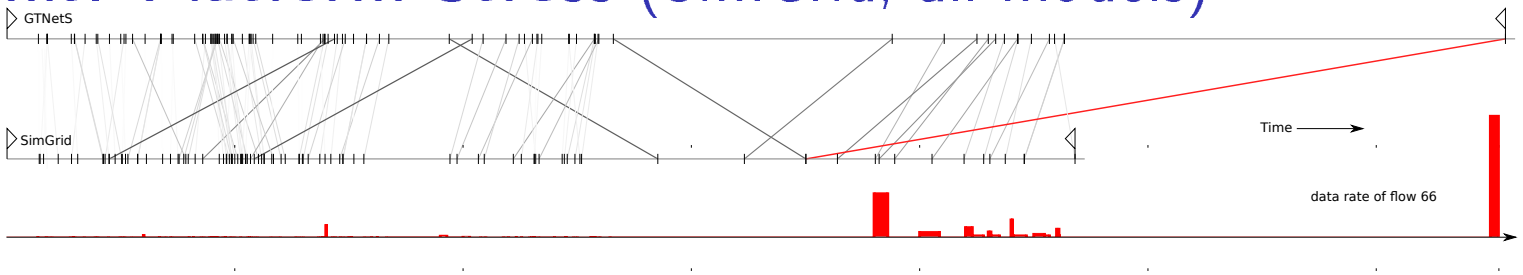


S	$\overline{\lvert\varepsilon\rvert}$	$\lvert\varepsilon_{max}\rvert$
$S < 100KB$	$\approx 12\%$	$\approx 162\%$
$S > 100KB$	$\approx 1\%$	$\approx 6\%$

More Crucial Experiments

Model Limit: Platform Stress (SimGrid, all models)

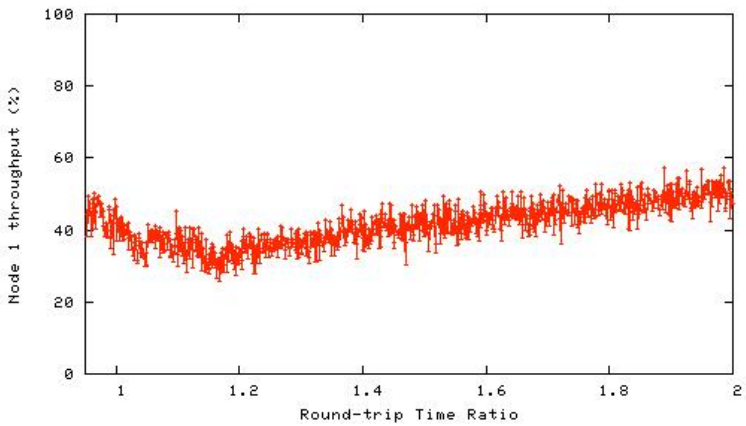


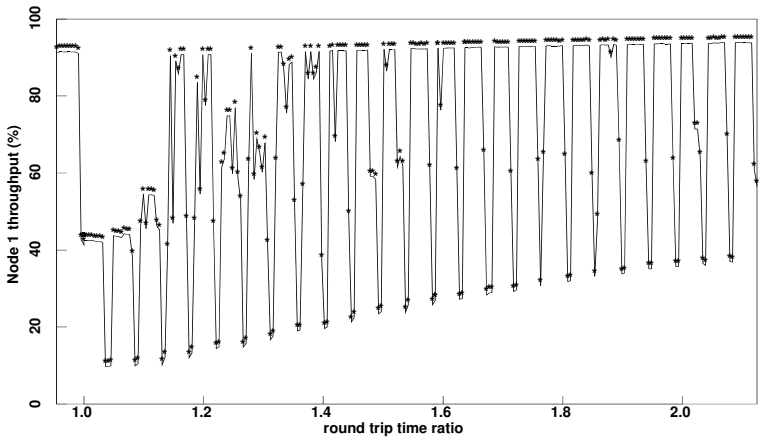


- Flow 66 terminates too early in SimGrid; seems stuck until timeout on GTNetS

Model Limit: Phase effect (packet-level tools: NS2, NS3)

- Two long-lived flows, real (left) and simulated (right)

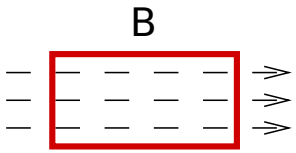




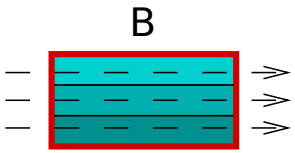
- Periodic, deterministic traffic ⇒ May resonate [Floyd and Jacobson 1991]

So, what can you expect from SimGrid??

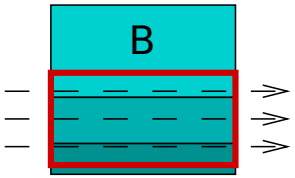
Implementation Limit: Bugs (GridSim)



Setting



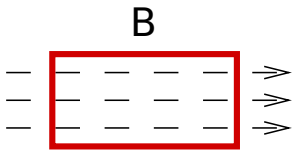
Expected Result



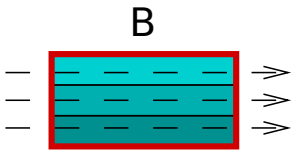
Actual Result

So, what can you expect from SimGrid??

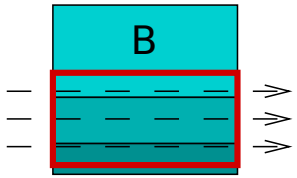
Implementation Limit: Bugs (GridSim)



Setting



Expected Result



Actual Result

- Issue reported since ages, but no answer from authors
- If you (really) want to use CloudSim, prefer CloudSimPlus (better quality)

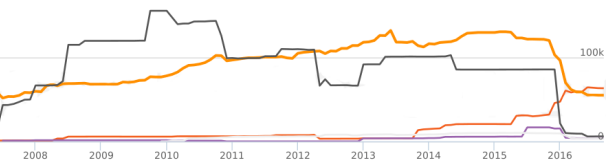
SimGrid is well Tested

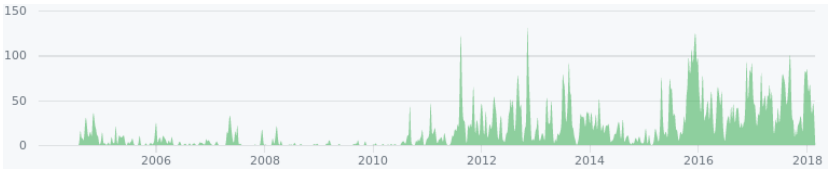
- 740 integration tests, 10k units (coverage: 80%)
- Each commit: 22 configurations (4 OS, 3 compilers, 2 archs; 3 providers)
- Nightly: 2 dynamic + 2 static analyzers; StarPU, BigDFT and Proxy Apps
- Still expect bugs, but our community strive to fix them if you provide a MWE

Technical Considerations

Complex and Dynamic Code Base

- Only 100k sloc, but complex due to versatile efficiency + formal verification
- Implemented in C++/C (+ assembly); Bindings: Java, Lua and Fortran
- Active project: commits every day by $\approx$ 6 commiters, 4 releases a year
- Ongoing full rewrite in C++ along with *Release soon, Release often*

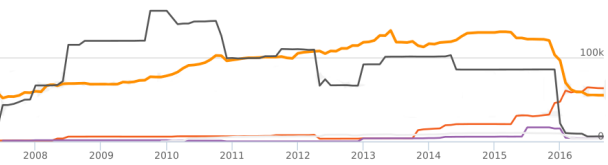


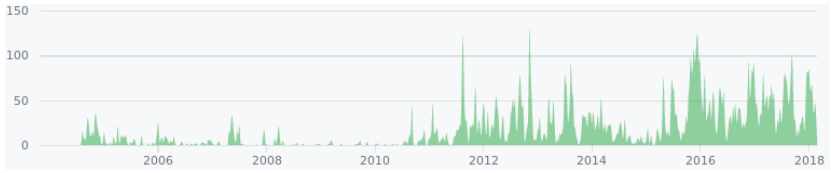


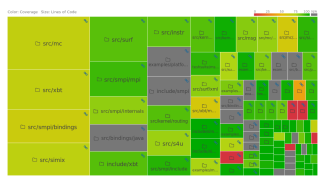
Technical Considerations

Complex and Dynamic Code Base

- Only 100k sloc, but complex due to versatile efficiency + formal verification
- Implemented in C++/C (+ assembly); Bindings: Java, Lua and Fortran
- Active project: commits every day by ≈ 6 commiters, 4 releases a year
- Ongoing full rewrite in C++ along with *Release soon, Release often*







Well Tested

- 740 integration tests, 10k units (coverage: 80%)
- Each commit: 22 configurations (4 OS, 3 compilers, 2 archs; 5 providers)
- Nightly: 2 dynamic + 2 static analyzers; StarPU, BigDFT and Proxy Apps
- Each Release: In Debian & Ubuntu (10+ architectures, 3 kernels)

- Software Q&A taken seriously

SaaP Work

Initial Difficulties: IC $\rightsquigarrow$ IE on a complex project

Capability Improvement

- Exascale Proxy Apps: Simplified code exhibiting classical characteristics
- Perfect test case for SimGrid. Toufik's work:
  - Automated Testing Infrastructure
  - Port each of these apps to SMPI (SimGrid MPI)
  - Fix the glitches found in SimGrid
  - Implement the missing pieces (in collab with Augustin Degomme)
- Conclusion: 51 Apps working, $\approx$ 30 not working (often: missing OpenMP)
- Ongoing: reproduce a paper from MeteoFrance

Usability Improvement

- New tutorial on S4U; Started tutorial on SMPI; Documentation Overhaul

Conclusion

- Successful ADT (despite difficulties), but unfinished
- The project progressed, but I now assume most of the technical work again
- Slows down the associated IPL Hac Specis (Formal Verification of HPC soft)

SimGrid: Versatile Simulator of Distributed Apps

Install a Scientific Instrument on your Laptop

Computational Science of Computer Science

- Joint Project since 1998, mostly from French institutions
- Open Project, contributors in the USA (UHawaii, ISI, NEU), UK, Austria, Cern

Key Strengths

- Performance Models validated with Open Science $\rightsquigarrow$ Predictive Power
- Architectured as an OS $\rightsquigarrow$ Efficiency; Performance & Correction co-evaluation
- Usability: Fast, Reliable, User-oriented APIs, Visualization
- Versatility: Advances in HPC modeling reused by Cloud users

Community

- Scientists: 500+ publications only cite it, 58 extend it, 314 use it
- Apps/Model co-dev : StarPU, BigDFT, TomP2P
- Some industrial users on internal projects (Intel, Bull)
- Open Source: external Power Users (fixes & models)