Simulating clouds with SimGrid

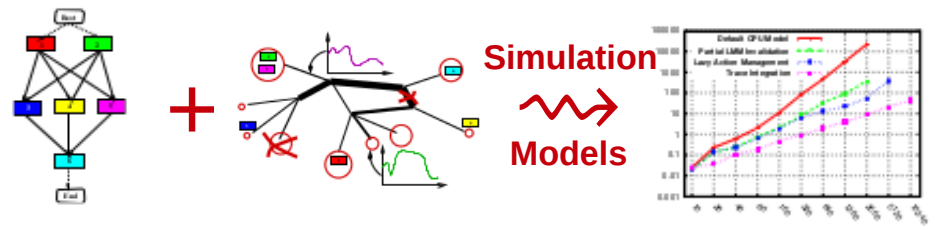
Martin Quinson (ENS Rennes / IRISA) and the SimGrid team

November 24., 2020

Simulating Distributed Systems

Simulation: Fastest Path from Idea to Data

- Test your scientific idea with a fast and confortable scientific instrument
- Experimental campaign with thousands of runs within the week



- Centralized and reproducible setup. Don't waste resources to debug and test
- No Heisenbug, full Clairevoyance, High Reproducibility, *What if* studies

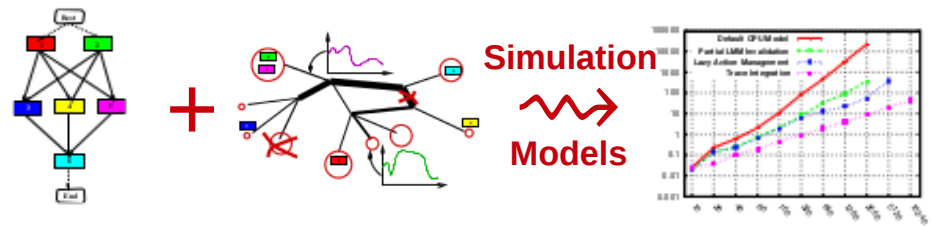
Simulator Expected Qualities

- Capability: Capture right concepts
- Software Accuracy: Q&A, Testing
- Result Accuracy: Validated Results
- Usability: Don't fool the users, doc

Simulating Distributed Systems

Simulation: Fastest Path from Idea to Data

- Test your scientific idea with a fast and confortable scientific instrument
- Experimental campaign with thousands of runs within the week



- Centralized and reproducible setup. Don't waste resources to debug and test
- No Heisenbug, full Clairevoyance, High Reproducibility, *What if* studies

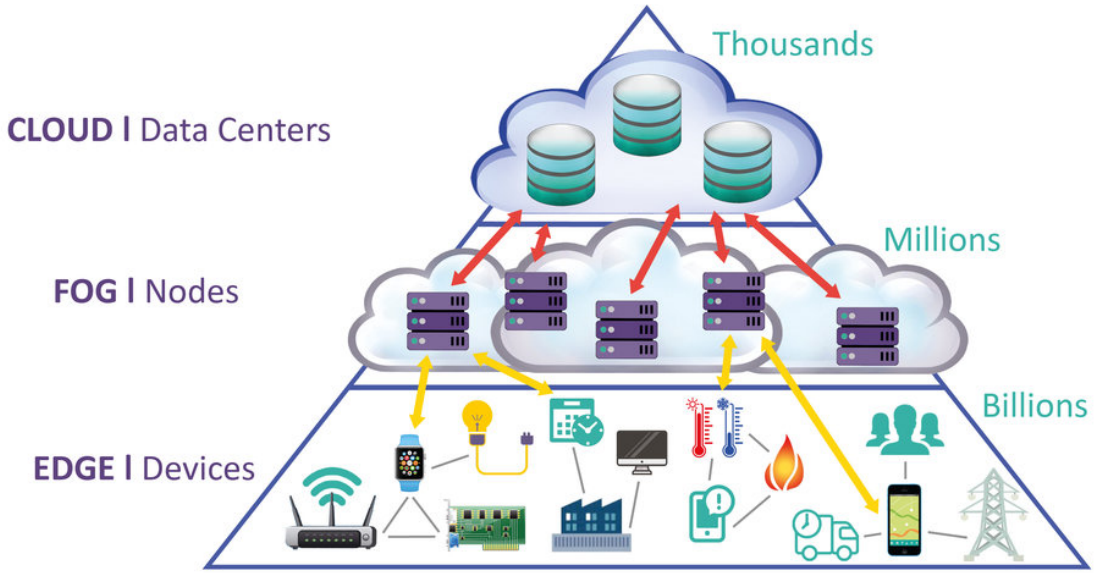
Simulator Expected Qualities

- Capability: Capture right concepts
- Software Accuracy: Q&A, Testing
- Result Accuracy: Validated Results
- Usability: Don't fool the users, doc

Usual tradeoffs and challenges

- Specific tools are narrow-minded
- High-level vs. Result Accuracy?
- Scalability vs. Precision?
- Soft. quality vs. research instrument

Cloud/Fog/Edge/IoT simulators



Credit: Marc Tesch

Simulating these systems?

- Many simulators focus on capability: CloudSim, DartCSim+, iFogSim, etc.
  - Great to quickly simulate a given system/algorithm. But poor or no accuracy
  - Narrow focus $\rightsquigarrow$ short-lived tools, with improvable software quality
- SimGrid focuses on accuracy and usability
  - Validated models (time, energy) of CPU, networks (TCP, wifi), disks
  - Historically, more HPC with limited support for Cloud/Fog/Edge/IoT systems

SimGrid: Versatile Simulator of Distributed Apps

Install a Scientific Instrument on your Laptop

- Joint Project since 1998, mostly from French institutions
- Open Project: users around the world (many academic and some industrial)
  - Almost 2000 publications only cite it, 60+ extend it, 400+ use it
  - Grid, Cluster, P2P, Volunteer computing, Hadoop, HPC, Cloud, Fog ...
- Several projects and extensions: Wrench, BatSim; StarPU, BigDFT; ProxyApp

Key Strengths

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

SimGrid-based cloud/fog/edge/IoT simulators

Features included in SimGrid

- VMs containing executions and can be migrated around
- Simple yet realistic CPU sharing model accounting for VMs:
  - sharing at PM level (full VMs limited by their CPU count) and then at VM level
  - $([o]_1 \ [o]_1 \ )_2 \rightsquigarrow 100\% + 100\%$ $\quad$ $([oo]_1 \ o)_2 \rightsquigarrow (50\% + 50\%) + 100\%$
  - $([o]_1 \ \ o \ )_2 \rightsquigarrow 100\% + 100\%$ $\quad$ $([oo]_2 \ o)_2 \rightsquigarrow (66\% + 66\%) + 66\%$
- Simple yet validated Wifi networks: time fairly shared, perf depends on SNR
  - Validated against ns-3 in some simple situations
- Full featured ns-3 networks usable directly from SimGrid (wired and wifi)

SimGrid-based cloud/fog/edge/IoT simulators

Features included in SimGrid

- VMs containing executions and can be migrated around
- Simple yet realistic CPU sharing model accounting for VMs:
  - sharing at PM level (full VMs limited by their CPU count) and then at VM level
  - $([o]_1 \; [o]_1 \;)_2 \rightsquigarrow 100\% + 100\%$ $\quad$ $([oo]_1 \; o)_2 \rightsquigarrow (50\% + 50\%) + 100\%$
  - $([o]_1 \;\; o \;)_2 \;\; \rightsquigarrow 100\% + 100\%$ $\quad$ $([oo]_2 \; o)_2 \rightsquigarrow (66\% + 66\%) + 66\%$
- Simple yet validated Wifi networks: time fairly shared, perf depends on SNR
  - Validated against ns-3 in some simple situations
- Full featured ns-3 networks usable directly from SimGrid (wired and wifi)

Several external cloud projects

- Schiaas: IaaS simulator (Strasbourg)
- Schlouder: Schiass extension to evaluate Cloud schedulers (Strasbourg)
- VMplaces: VM migration and consolidation (Nantes)

These projects leverage SimGrid 3

- Java programs based on the 15 years-old MSG interface for scheduling
- Writing these extensions was painful enough to motivate SimGrid 4

SimGrid 4: the S4U interface

Modern interface

- Generic and consistent API
- C++14 core (instead of plain C), with Python bindings (instead of Java)

Simple concepts

- Actors: run user code ($\approx$ thread or processes)
- Activities: explicit communication, computation, disk usage, synchro
- Resources: Host, Links, Disks. Shared through performance models.
- Scalable platforms: Hierarchy of network zones, with user-defined routing

Extensible through plugins

- Attach code to signals fired automatically by the simulation kernel
  - Actor: `creation`, `suspend`, `resume`, `sleep`, `wakeup`, `migration`, `termination`
  - Comm, Exec, IO, Synchro: `creation`, `start`, `state_change`, `completion`
  - Host, Link, Disk: `creation`, `state_change`, `speed_change`, `destruction`
  - Engine: `platform_creation`, `time_advance`, `simulation_end`, `deadlock`
  - VM: `start`, `suspend`, `resume`, `migration`, `termination`, NetZone
- Extend simulation objects with your data: Activities, Resources, NetZones

Existing plugins in SimGrid 4

Simulation capability

- VM migration: live in 2 phases, with dirty page tracking (default: immediate)
- File system: file creation/rm; disk capacity (default: only read/write activities)

Simulation models

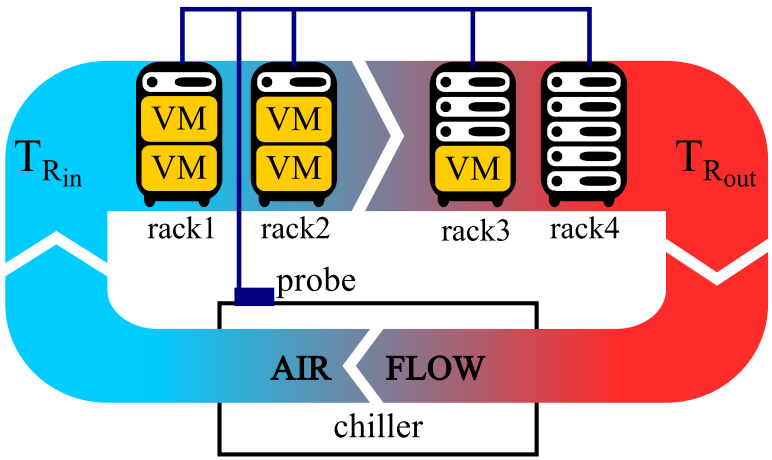
- VM sharing, DVFS: Implemented through plugins
- Resource load tracking: adapt this plugin to your scheduling algorithm
- CPU and link energy consumption: 200 lines each

Planned/possible models

- Other VM migration model
- Power capping: reducing computing performance when heating
- ...

Co-simulation of cyber-physical systems

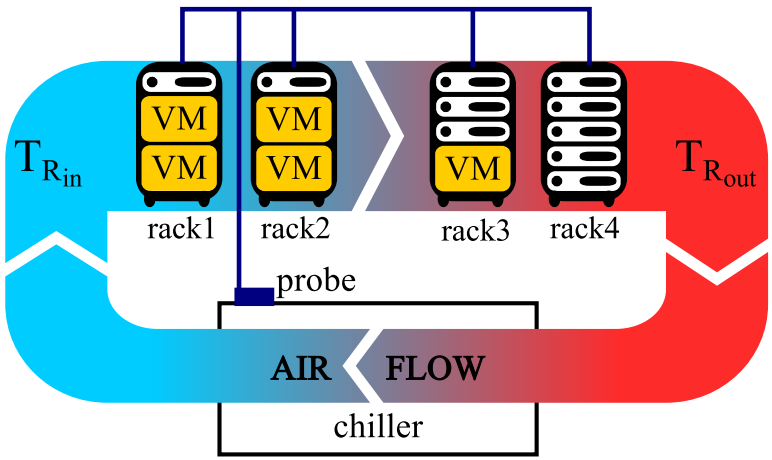
Motivating Example: A Data Center with a chilling facility



- Servers host VMs, which load heats the air
- The chiller cools the air flow, up to a point
- Above a given load threshold, the chiller stops
- Above a given temp., the DC shuts down
- Q: How to migrate the VMs before shutdown?

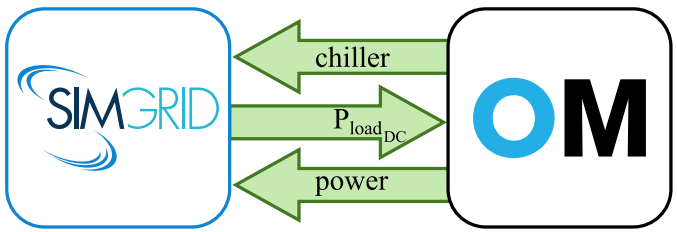
Co-simulation of cyber-physical systems

Motivating Example: A Data Center with a chilling facility



- Servers host VMs, which load heats the air
- The chiller cools the air flow, up to a point
- Above a given load threshold, the chiller stops
- Above a given temp., the DC shuts down
- Q: How to migrate the VMs before shutdown?

- Server heating + Cooling in OpenModelica
- IT system as SimGrid actors
- Server load: SG→ OM; Temp prob: OM→ SG

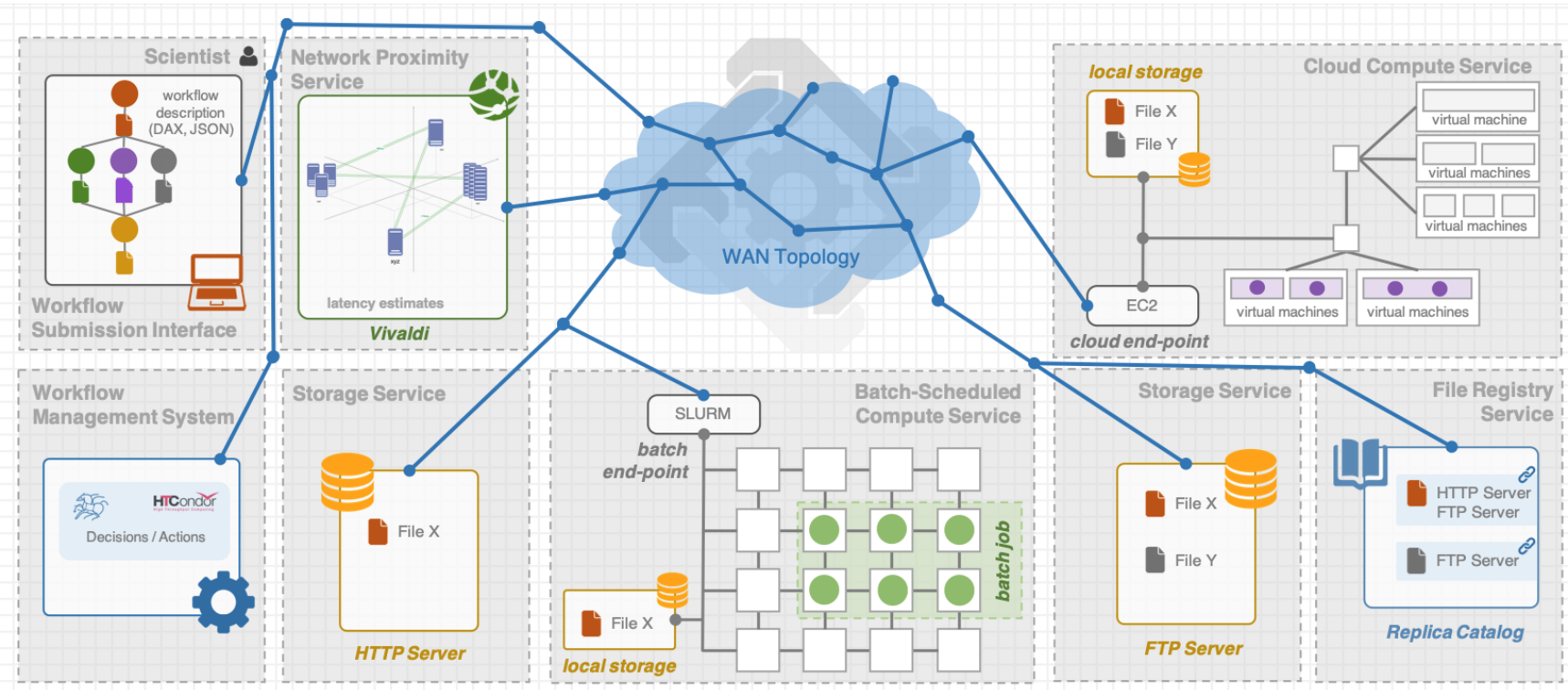


SimGrid-FMU: generic co-simulation (import 2.0 FMUs)

- Implemented as a plugin on top of SimGrid, released separately
- Used internally to interface with Power Factory and Power Panda

Wrench: Capable simulator on top of SimGrid

- High-level building blocks for developing custom simulators
  - Computation (Cloud EC2-like, Batch, Rack), Storage (FTP, HTTP, P2P)
  - Monitoring (Vivaldi, Sonar), Replica catalogs, etc
- Simulated production systems: WMS (Pegasus, Moteur, Makeflow), Hadoop
- Simulation-driven pedagogic modules for parallel and distributed computing
- Based on SimGrid models for accuracy, scalability and usability



Conclusion

SimGrid: accurate and scalable low-level building blocks

- Efficient core implemented in C++14, with Python bindings (among other)
- Performance models (time, energy) for CPU, network (wired, wifi) and disk
  - Validity = 20 years adventure; co-simulation with ns-3 and FMU
- Extensive testing work; now with a documentation and several tutorials
- `https://simgrid.org` (CNRS, Inria, Universities, ENS Rennes – ANR)

Wrench: high-level blocks to simulate Cluster/Grid/Cloud systems

- Declare a full platform in few lines, get comprehensive simulation vizualizations
- Full pedagogical curriculum on parallel and distributed computing
- `https://wrench-project.org` (ISI/USC and UH Manoa – NSF)

Build your own research on SimGrid

- Your plugin to accurately model other phenomena (live migration)
- Your algorithm on top of Cloud/Fog/Edge/IoT platforms
- (add your idea here)

SimGrid: Versatile Simulator of Distributed Apps

Install a Scientific Instrument on your Laptop

- Joint Project since 1998, mostly from French institutions
- Open Project: users around the world (many academic and some industrial)
  - Almost 2000 publications only cite it, 60+ extend it, 400+ use it
  - Grid, Cluster, P2P, Volunteer computing, Hadoop, HPC, Cloud, Fog ...
- Several projects and extensions: Wrench, BatSim; StarPU, BigDFT; ProxyApp

Key Strengths

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