# Présentation en Comité des Projets

4 juillet 2016

Martin Quinson, professeur ENS Rennes, rattaché à l'équipe Myriads

- Depuis septembre 2015
- 2005-2015 : EPI AlGorille/VeriDis à Nancy + Télécom Nancy

**slides de candidature, mai 2015**

Projet de recherche **Computational Science of Computer Systems**

- Faire face à la complexification des systèmes informatiques

Projet d'enseignement **Massification de l'enseignement de l'informatique**

- Aider à l'introduction de l'informatique pour tous en France

# Présentation en Comité des Projets

4 juillet 2016

Martin Quinson, professeur ENS Rennes, rattaché à l'équipe Myriads

- Depuis septembre 2015
- 2005-2015 : EPI AlGorille/VeriDis à Nancy + Télécom Nancy

**slides de candidature, mai 2015**

Projet de recherche Computational Science of Computer Systems

- Faire face à la complexification des systèmes informatiques

Projet d'enseignement Massification de l'enseignement de l'informatique

- Aider à l'introduction de l'informatique pour tous en France

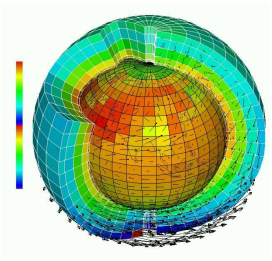
# Computational Science

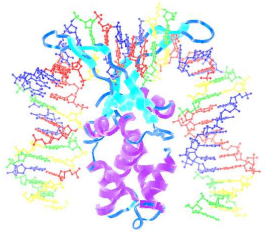
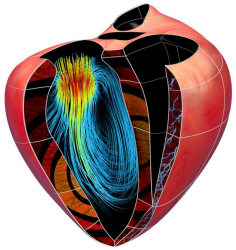
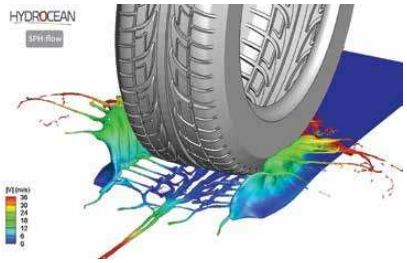
## Principe de base

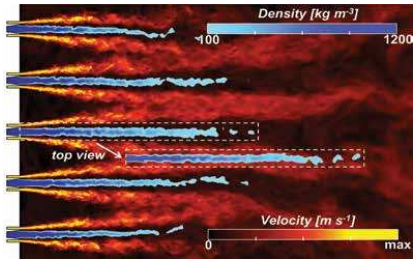
- Modèles mathématiques des phénomènes
- Simulation sur super-calculateurs
- *In*validation: prédictions vs. observations

  Évaluation expérimentale des théories
- Puis des résultats sans réaliser d'expérience

# Computational Science



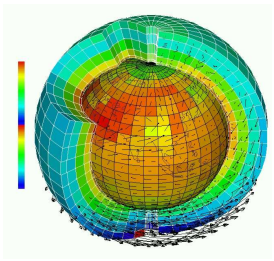


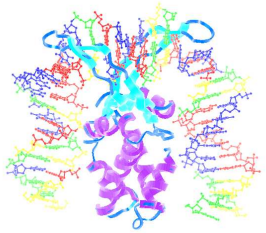
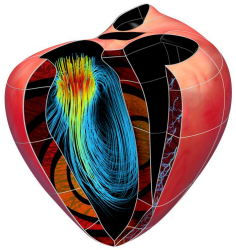
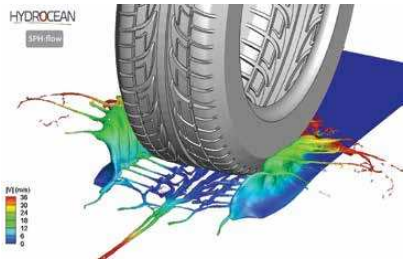

## Principe de base

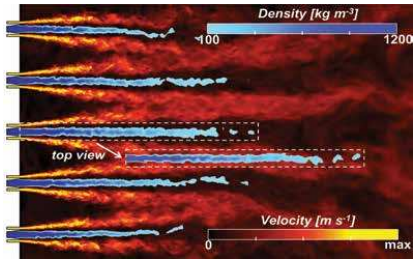
- Modèles mathématiques des phénomènes
- Simulation sur super-calculateurs
- *In*validation: prédictions vs. observations

  Évaluation expérimentale des théories
- Puis des résultats sans réaliser d'expérience

# Computational Science






## Principe de base

- Modèles mathématiques des phénomènes
- Simulation sur super-calculateurs
- *In*validation: prédictions vs. observations

  Évaluation expérimentale des théories
- Puis des résultats sans réaliser d'expérience


## Computational Science of Computer Systems

- Utiliser cette approche pour comprendre les systèmes informatiques modernes



# Gigantisme des centres de calcul modernes

## Calcul scientifique

- Énorme impact sociétal, systèmes énormes :

| | | | |
|---|---|---|---|
| 1. TaihuLight | 10 649 600 cœurs | 125 Pflops | 15MW |
| 2. Tianhe-2 | 3 120 000 cœurs | 55 Pflops | 18MW |
| 3. Titan | 560 640 cœurs | 27 Pflops | 8MW |
| 4. Sequoia | 1 572 864 cœurs | 20 Pflops | 8MW |
| 5. K Computer | 705 024 cœurs | 11 Pflops | 13MW |

- ExaScale ($10^{18}$ flop/s) prochainement

## Pas seulement pour le calcul scientifique

- Clouds : Google dissipe 300MW avec environ 1 000 000 serveurs
- Pair-à-Pair, Internet of Things, BotNets, ...

# Gigantisme des centres de calcul modernes

## Calcul scientifique

- Énorme impact sociétal, systèmes énormes :

| | | | |
|---|---|---|---|
| 1. TaihuLight | 10 649 600 cœurs | 125 Pflops | 15MW |
| 2. Tianhe-2 | 3 120 000 cœurs | 55 Pflops | 18MW |
| 3. Titan | 560 640 cœurs | 27 Pflops | 8MW |
| 4. Sequoia | 1 572 864 cœurs | 20 Pflops | 8MW |
| 5. K Computer | 705 024 cœurs | 11 Pflops | 13MW |

- ExaScale ($10^{18}$ flop/s) prochainement

## Pas seulement pour le calcul scientifique

- Clouds : Google dissipe 300MW avec environ 1 000 000 serveurs
- Pair-à-Pair, Internet of Things, BotNets, ...

## Mon champ de recherche : Méthodologies d'expérimentation

- But : Performances et correction de ces systèmes
- Approche fondamentalement expérimentale : tester ce qui est
- Contribution majeure : SimGrid, simulateur de systèmes distribués

# Simulation de systèmes parallèles et distribués

## Instruments scientifiques et simulation

- Réseaux : Quelques standards établis
- Grids — OptorSim, GridSim, ...
- P2P — P2Psim, PeerSim, OverSim, ...
- Volunteer — SimBA, SimBOINC, ...
- Clouds — CloudSim, DCSim, GroudSim, ..., iCanCloud, GreenCloud, CDOSim
- HPC/MPI — Dimemas, PSinS, BigSim, ..., LogGoPSim, XSim, SST

# Simulation de systèmes parallèles et distribués

## Instruments scientifiques et simulation

- Réseaux : Quelques standards établis
- Grids — OptorSim, GridSim, ...
- P2P — P2Psim, PeerSim, OverSim, ...
- Volunteer — SimBA, SimBOINC, ...
- Clouds — CloudSim, DCSim, GroudSim, ... iCanCloud, GreenCloud, CDOSim
- HPC/MPI — Dimemas, PSinS, BigSim, ... LogGoPSim, XSim, SST

**SimGrid:** Projet communautaire depuis 15 ans

- Versatile : Grid, P2P, Clouds, HPC, ...
- Collaboratif : ANR, CNRS, Universités, Inria
- Largement utilisé : Des centaines de publications

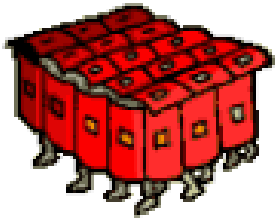
## Raisons de ce succès

- Modèles de performance validés par Open Science $\rightsquigarrow$ puissance prédictive
- Pensé comme un OS : efficacité, évaluation conjointe perfs et correction

# Étudier la correction dans SimGrid

Applications distribuées correctes : notoirement difficile

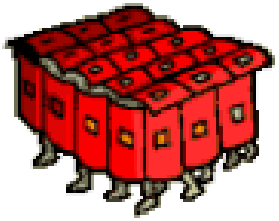
- Solution pessimiste : Moins d'exigences de performance
- Solution optimiste : *Eventually Consistant*
- Solution classique : Preuves d'algorithmes
- Solution HPC : Schémas de communication rigides

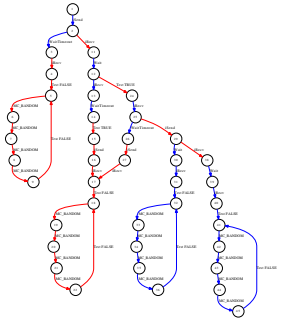


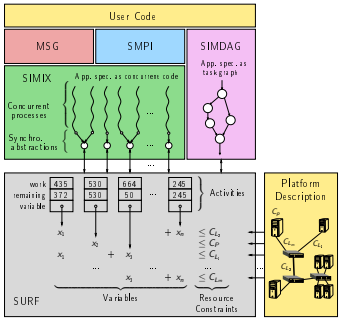
# Étudier la correction dans SimGrid

## Applications distribuées correctes : notoirement difficile

- Solution pessimiste : Moins d'exigences de performance
- Solution optimiste : *Eventually Consistant*
- Solution classique : Preuves d'algorithmes
- Solution HPC : Schémas de communication rigides








## Vérification Dynamique Formelle dans SimGrid

- Test exhaustif des évolutions d'une configuration du système
- Modèle d'applications : *Concurrent Sequential Processes*
- Point d'indécision : ordre des msgs ; Transition : code entre msgs (déterministe)
- Réutilisation de la virtualisation de SimGrid : codes MPI non modifiés

# Projet de recherche

## Computational Science **of** Computer Systems

- Modèles calculés par ordinateur pour comprendre les systèmes IT modernes
- Modèles de perfs et sémantique; simulation et méthodes formelles
- En pratique, je m'appuie sur SimGrid et sa communauté

### Axe 1 : Modélisation de systèmes distribués modernes — Myriads

- Simulation de la dissipation énergétique — *A.-C. Orgerie*
- Clouds et composition de services — *G. Pierre*

### Axe 2 : Virtualisation d'applications distribuées — Myriads

- Émulation par interception des appels systèmes — *C. Morin*

### Axe 3 : Étude formelle d'applications distribuées — Sumo

### Mes apports

- Approche cohérente et originale : de l'épistémologie à des outils pragmatiques
- Expertise en modélisation et simulation d'applications distribuées

# Choses engagées depuis mon recrutement

## Axe 1 : Modélisation de systèmes distribués modernes

- Réécriture de SimGrid en cours depuis juillet 2015
- Article soumis (Energy-Efficient Shutdown Techniques for Cloud DC)
- Stage L3 ENS sur la modélisation de workload cloud
- Collaboration démarrée avec le Satie, ENS Rennes: Smart Grids

## Axe 2 : Virtualisation d'applications distribuées

- Soumission d'un article TPDS sur la virtualisation de codes MPI
- Contributions d'un stagiaire Bull pour la virtualisation d'applications

## Axe 3 : Étude formelle d'applications distribuées

- IPL *Hac Specis* pilotée par A. Legrand $\rightsquigarrow$ thèse (co-dir T. Jéron)
- Article sur la vérification formelle de codes MPI (J. Algebraic Meths for Prog)

## Autres :

- Demandes : ADT (SimGrid valorisé et enseigné); Post-doc SAD région
- Collabs : SimGrid Workflow (UH, ISI), SimGrid OS (NEU), Communauté SG

# Présentation en Comité des Projets

4 juillet 2016

Martin Quinson, professeur ENS Rennes, rattaché à l'équipe Myriads

- Depuis septembre 2015
- 2005-2015 : EPI AlGorille/VeriDis à Nancy + Télécom Nancy

**slides de candidature, mai 2015**

Projet de recherche **Computational Science of Computer Systems**

- Faire face à la complexification des systèmes informatiques

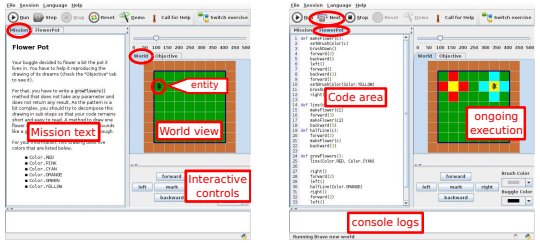
Projet d'enseignement **Massification de l'enseignement de l'informatique**

- Aider à l'introduction de l'informatique pour tous en France

# Activité d'enseignement

## Enseignements à Telecom Nancy

- Passé : Algorithmes distribués; Applications réparties
- 2015 : AlgoProg (Scala, Java, C); Prog Système
- PLM : Exerciseur de programmation




## Diffusion de la culture scientifique

- Activités débranchées en fête de la science
- Programmation créative en Scratch en MJC

## Enseignement massif de l'informatique en France

- Journées d'échanges pédagogiques
- Expérimentation en maths en collège (IREM Nancy)
- Groupes de travail : Programmes et manuel LAMAP
- Formation de formateurs, séminaires, interviews

# Projet d'enseignement à l'ENS Rennes

## Réenchanter la technique et le *low level*

- Partager ma passion du système : besoin sociétal, diversifier horizon élèves

## Pédagogie expérimentale avec la PLM

- Révolution *Learning Analytics* (Big Data $\rightsquigarrow$ détection d'élèves en difficulté)

## Massification de l'enseignement de l'informatique en France

- Décision politique imminente, mais nous ne sommes pas prêts

## Lien avec le programme de recherche

- Collaborations entre apprenants, co-constructions de savoirs et *Open Science*

# Choses engagées depuis mon recrutement

## Enseignement de la technique à l'ENS

- Cette année: Beaucoup de TP (Caml, Robotique, Système & Réseau) + Scala
- *2016-2017:* Système/Réseau, Aggrég Sciences Industrielles option Info

## Recherche pour l'enseignement de l'informatique

- Inria Learning Lab: suite du Mooc Lab plus orientée recherche
- Lancement GdT SIF «Informatique débranchée», Wikipédia
- Atelier aux journées Orphées : Big Data pour l'enseignement de la prog»
- PIA probablement accepté (Chambéry) $\rightsquigarrow$ Thèse learning analytics de la PLM
- Collabs : P. Hubweiser, Ch. Boisvert, *Tacit (psycho Rennes 2)*

## Massification de l'enseignement de l'informatique

- Cours de pédagogie: création de ressources en M2, visites de classe en L3
- ENS Rennes partenaire de Class'Code
- *2016-2017:* Graines de science avec LAMAP, Formation des EMF/ CPC du 35

# Résumé de mon activité ici

## Recherche : Faire face à la complexification des systèmes informatiques

- Motivation : Impact sociétal des systèmes informatiques
- Objectif : Permettre l'étude de ces systèmes
- Moyen : Vers une informatique computationnelle
- Outils variés : Modélisation, virtualisation et OS, méthodes formelles, ...

## Enseignement : Massification de l'enseignement de l'informatique

- Former des experts en systèmes, maniant théorie et pratique
- Préparer l'instruction de l'ensemble de la société
- Déconstruire la presti*digit*ation numérique des citoyens

## Intégration

- Équipe Myriads : expérimentation d'applications distribuées, OS distribués
- Équipe Sumo : méthodes formelles pour les systèmes distribués
- ENS Rennes: pédagogie de l'informatique (recherche et pratiques)
- Environnement idéal pour mener à bien mes différentes activités

Slides à questions

# Mes autres travaux dans SimGrid

## Modèles hybrides de performance du réseau

- Modèle hybride pour TCP, modèles des collectives MPI (soon IB)
- Défi : Reproductibilité des expériences. Solution : Open Science.

## SimGrid as an OS

- Défi : Tests exhaustifs, vérif. Solution : Séparation des processus par *simcalls*.
- Défi : Applications MPI C/Fortran. Solution : Médiation des globales.
- Défi : Simulation parallèle efficace. Solution : Parallélisation originale.

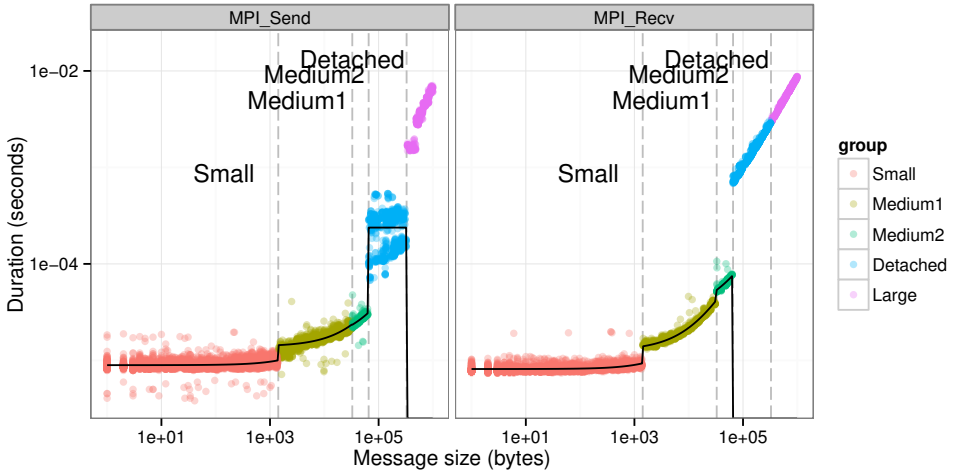
## Verification formelle d'applications distribuées

- Sûreté, Vivacité ou CTL, avec DPOR ou égalité d'états
- Utilisable sur de vrais codes MPI (C/Fortran)

On en reparle lors des questions ;)

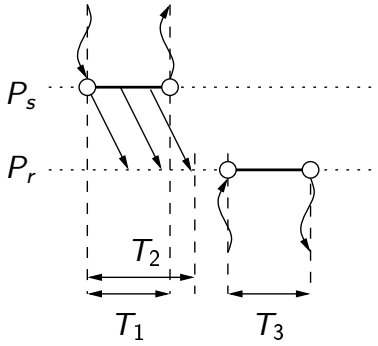
# Finding 1: Such Models are possible

## Measurements

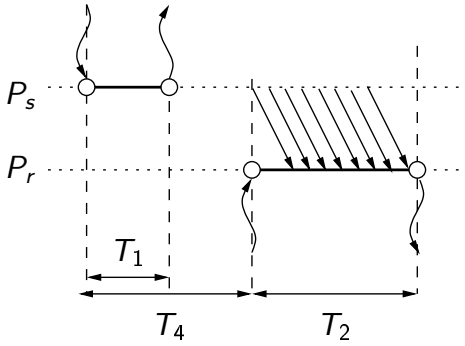



## Model hybridizing LogP...

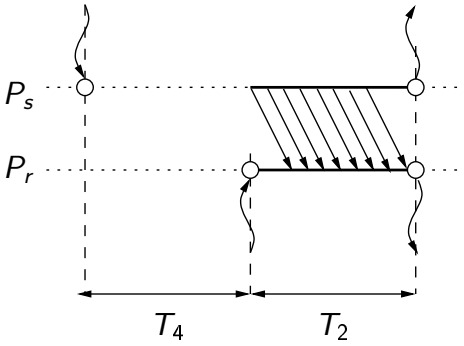
Asynchronous ($k \leq S_a$)



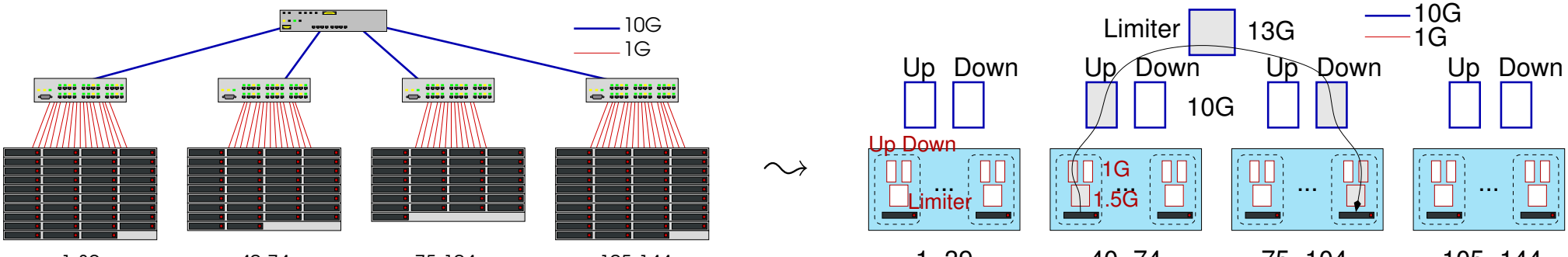

Detached ($S_a < k \leq S_d$)




Synchronous ($k > S_d$)



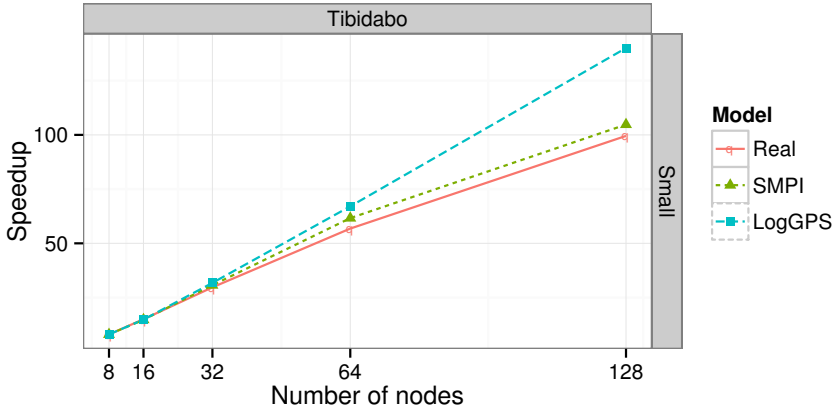

## ...and Fluid model: account for contention and network topology



# Finding 2: These Models are useful

## Sometimes, it work rather well

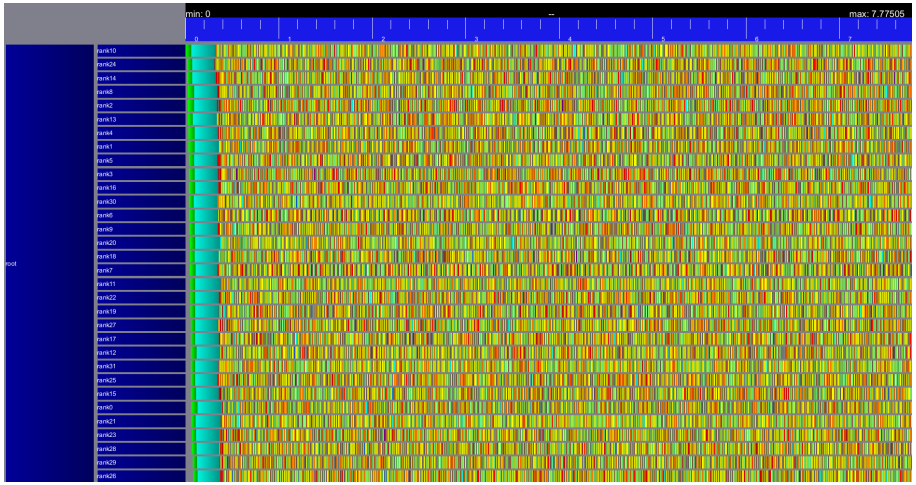
App: BigDFT (physics)
Host: Tibidabo (ARM + Ethernet 10G)

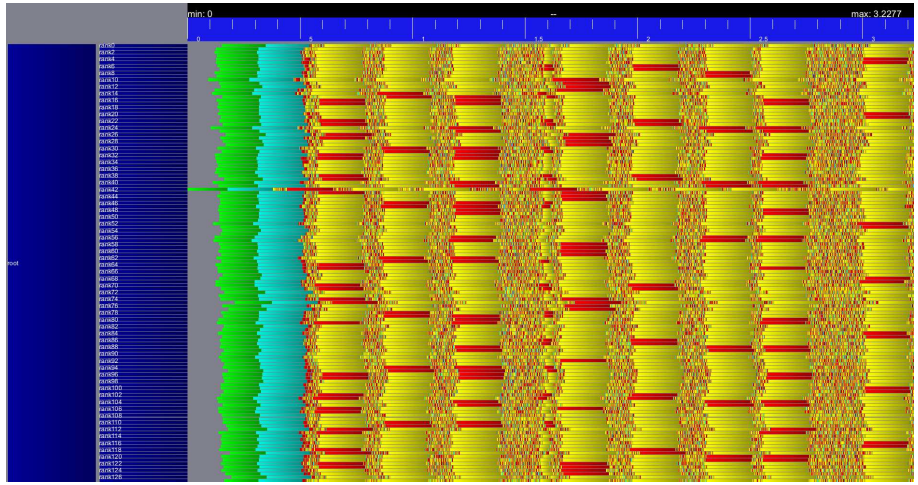



## Sometimes, Simulation sucks

- Model limits, Bad instanciation, Applicative model faulty

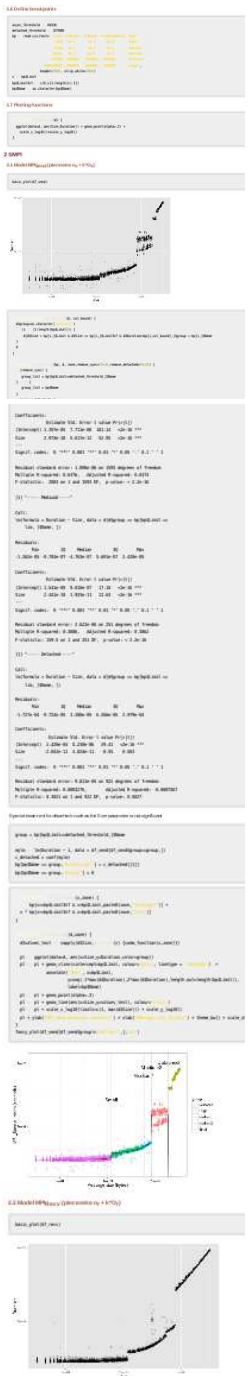
## Sometimes, Reality sucks





- NAS PB benchmark. Left: simulation; Right real execution
- Discrepancy: Reality experiences timeouts that are probably due to TCP RTO

# Finding 3: Open Science: new Scientific Eden



## Devil's in the Details *vs.* Grail of Reproducibility

- Describe Experience environ.+protocol hard: data deluge
- Experiences very sensible: impact macro of micro errors
- Statistical Data Processing become tedious

## But it remains possible!

- Grid'5000 is a precious hardware and community
- Our tools (YMMV): git + org-mode + R
  *Computational scientists* routinely use them

## Next Step: Convincing our Research Community ;)

- *I found the results section of this paper to be pretty weak.*
- *If less accurate models drive the user to the same conclusions as Fig. 8 indicates, why we need more complex models?*
  R: Well, for the experiment of Fig. 9 for example...

# SMPI

## What is it?

- Reimplementation of MPI on top of SimGrid
- Imagine a VM running real MPI applications on platform that does not exist
  - Horrible over-simplification, but you get the idea
- Computations run for real on your laptop, Communications are faked

## What is it good for?

- Performance Prediction ("what-if?" scenarios)
  - Platform dimensioning; Apps' parameter tuning
- Teaching parallel programming and HPC
  - Reduced technical burden
  - No need for real hardware, or hack your hardware

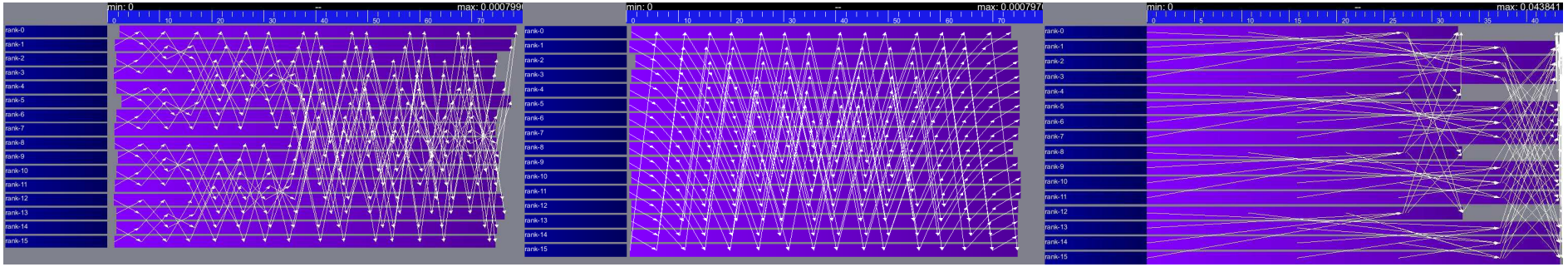
## Studies that you should NOT attempt with SMPI

- Predict the impact of L2 caches' size on your code
- Interactions of TCP Reno vs. TCP Vegas vs. UDP
- Claiming a simulation of 1000 billions nodes

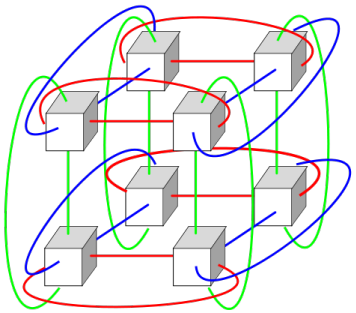
# SimGrid Modeling of MPI

## MPI Collectives

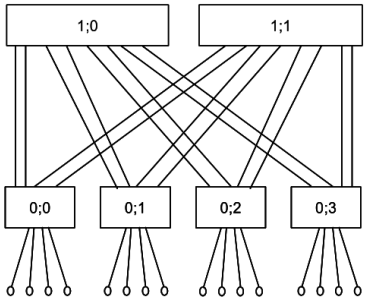
- SimGrid implements more than 120 algorithms for the 10 main MPI collectives
- Selection logic from OpenMPI, MPICH can be reproduced



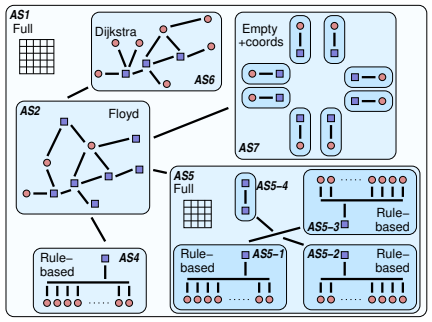
## HPC Topologies



Torus




Fat-trees




Hierarchies of ASes

## But also

- External load (availability changes), Host and link failures, Energy (DVFS)
- Virtual Machines, that can be migrated; Random platform generators

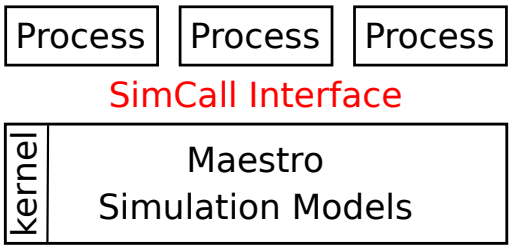
# Efficient Parallel Fine-Grained Simulation

## SimGrid is an Operating System

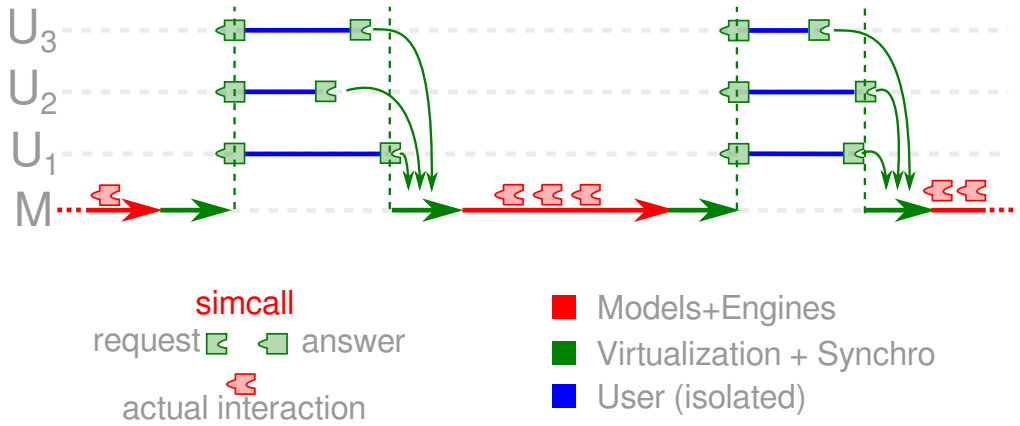
## Simcalls separate processes, alleviating locking issues

- Very similar to syscalls in an operating system

### Functional View

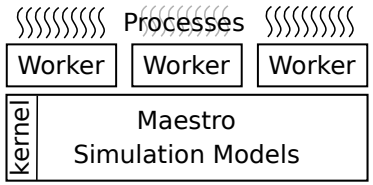



### Temporal View

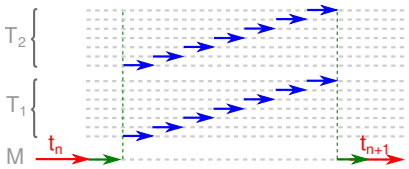



## Leveraging Multicores

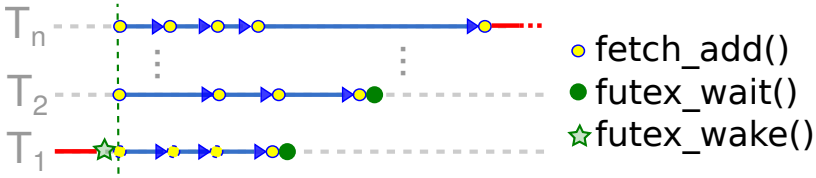
⇒ More processes than cores ⇝ Worker Threads (that execute co-routines ;)




Functional View



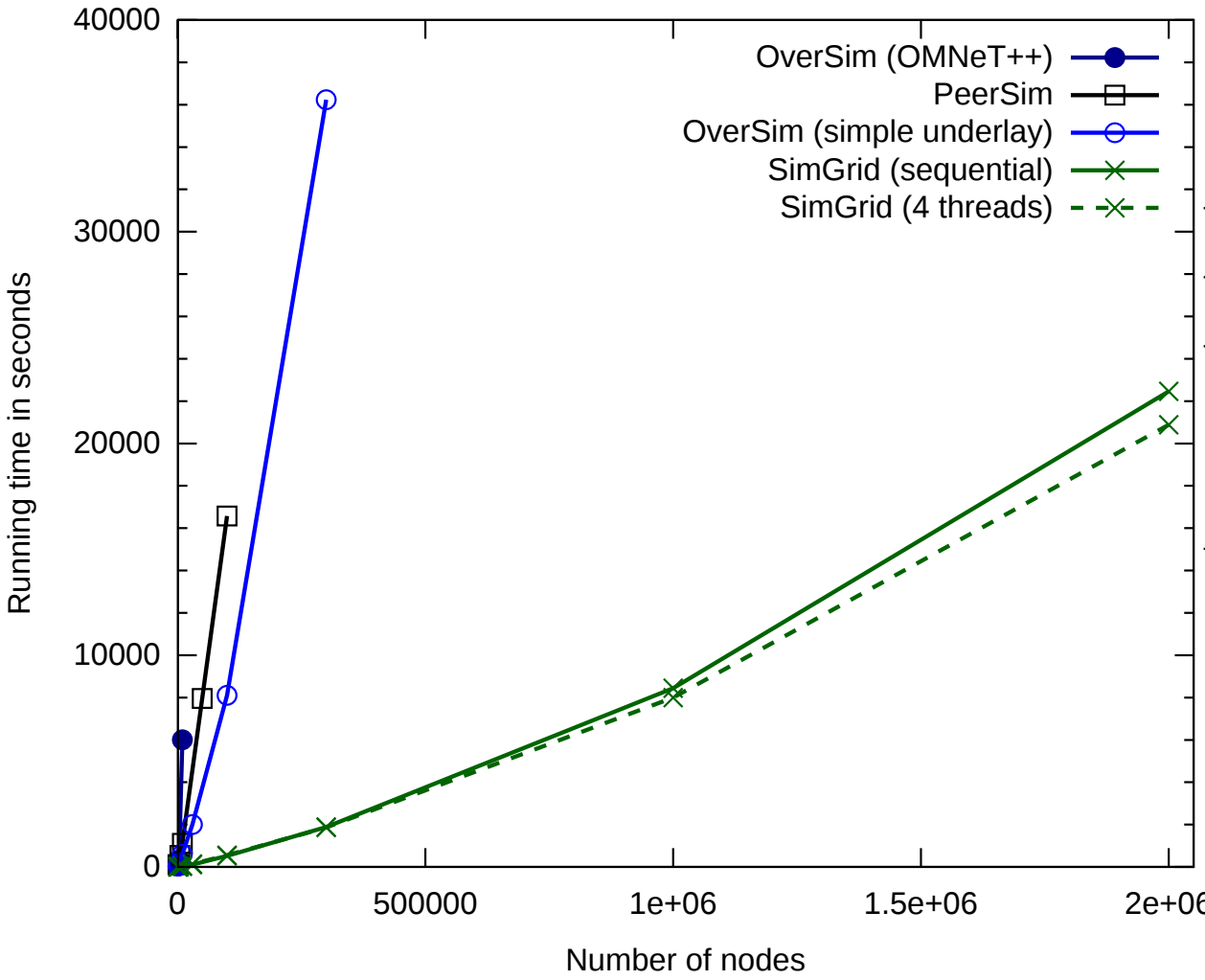

Temporal View




Ideal Algorithm

# Performance Results

- Scenario: Initialize Chord, and simulate 1000 seconds of protocol
- Arbitrary Time Limit: 12 hours (kill simulation afterward)




## Largest simulated scenario

| | Size | Time |
|---|---|---|
| Omnet++ | 10k | 1h40 |
| PeerSim | 100k | 4h36 |
| OverSim | 300k | 10h |
| SimGrid, seq | 10k | 32s |
| | 300k | 32mn |
| | 2M | 6h18 |
| SimGrid// | 10k | 130s |
| | 300k | 40mn |
| | 2M | 5h55 |

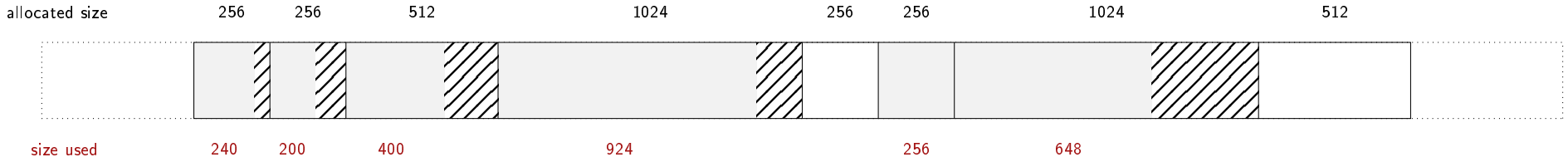
## Memory Usage

18kiB /process (stack: 12kiB)

First time that PDES is (a little) faster than DES [CCGrid'12], cited 17

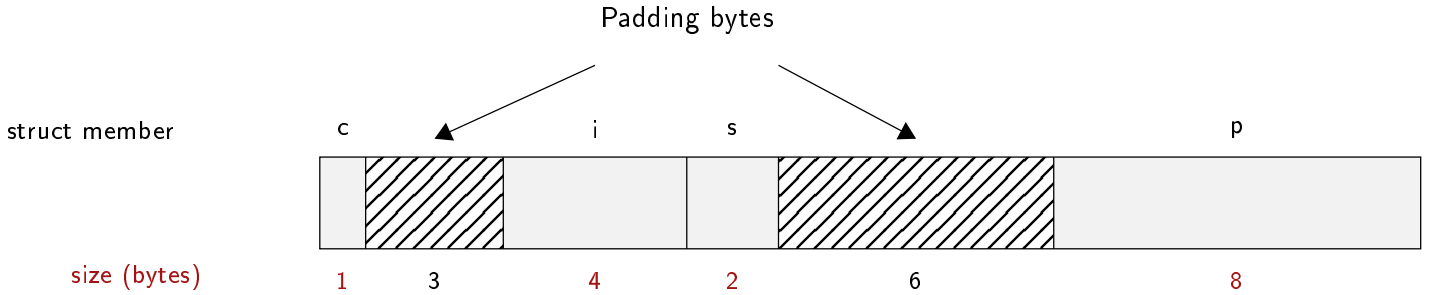
# OS-level Challenges of State Equality Detection

- Memory over-provisioning

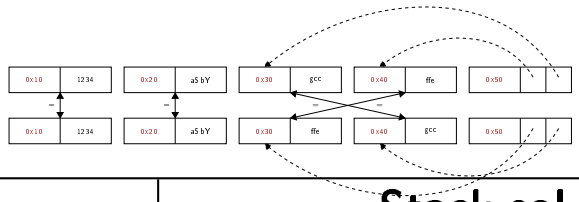



- Padding bytes: Data structure alignment

```
struct foo {
char c;
int i;
short s;
void *p;
}
```




- Irrelevant differences: system-level PID, fd, ...
- Syntactic differences / semantic equalities:



## Solutions

| Issue | Heap solution | Stack solution |
|---|---|---|
| Overprovisioning | `memset` 0 (customized `mmalloc`) | Stack pointer detection |
| Padding bytes | `memset` 0 (customized `mmalloc`) | DWARF + libunwind |
| Irrelevant differences | Ignore explicit areas | DWARF + libunwind + ignore |
| Syntactic differences | Heuristic for semantic comparison | N/A (sequential access) |

# Some Results

## Wild safety bug in our Chord implementation ($\approx$ 500 lines of C)

- Simulation: bug on large instances only; MC finds small trace (1s with DPOR)

## Mocked liveness bug

- Buggy centralized mutual exclusion: last client never obtains the CS
- About 100 lines – state snapshot size: 5Mib
- Verified with up to 7 processes (12,000 states, 9 minutes, 45Gb).

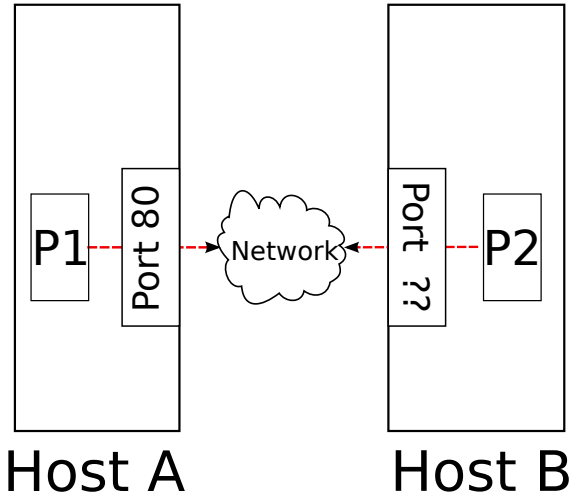
## Verifying MPICH3 complience tests

- Looking for assertion failures, deadlocks and non-progressive cycles
- 6 tests; $\approx$ 1300 LOCs (per test) – State snapshot size: $\approx$ 4MB
- With no reduction: no test concluded in a few hours
- With state equality: Exhaustive exploration up to 10 procs, but no error found
- With memory compaction: use only dozen of Gb in RAM, not hundreds
- We verified several MPI2 collectives too ☺ (but all good so far ☹)

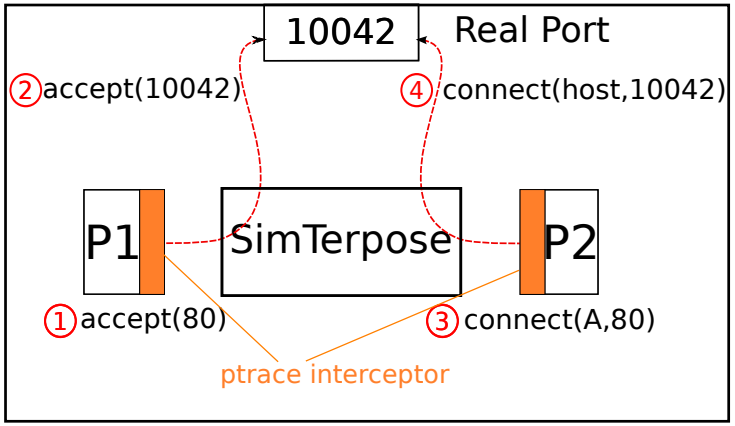
# SimTerpose Project

## Dream: Simulate any applications on top of SimGrid

### Simulated Setup

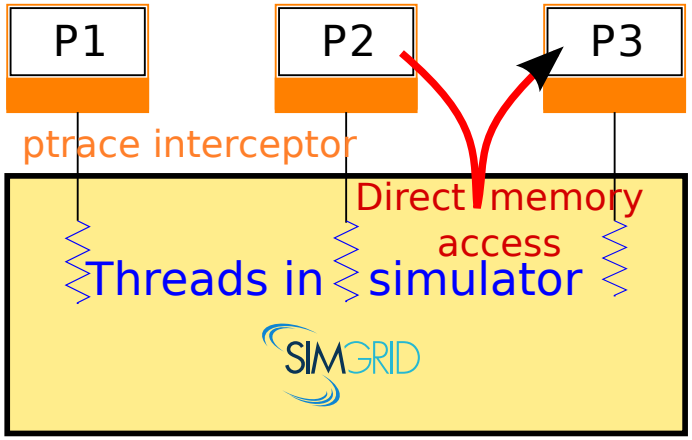



### Take 1: ptrace plumbering




Hosting Computer

### Take 2: Full Emulation



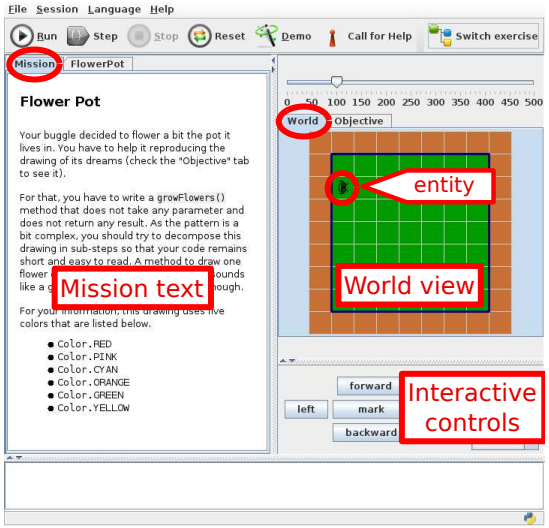

### Current State

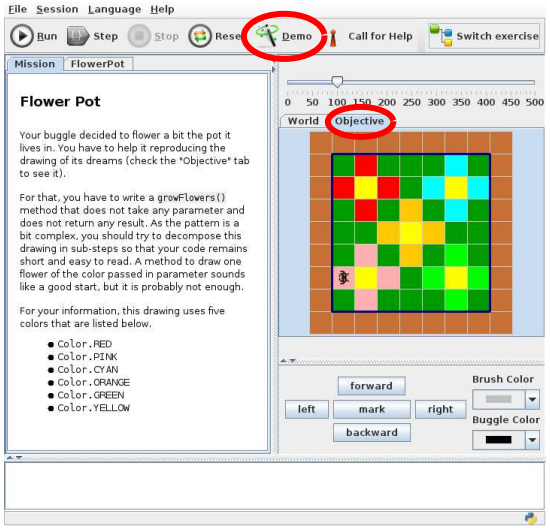
- Functional POC: send/recv exchange
- Need to handle the other 200 syscalls
  - Intercept, store metadata
  - Inform simulator, report effect on procs
- Time and DNS need love at link time
- We are redeveloping a libC! (in strange way ;)

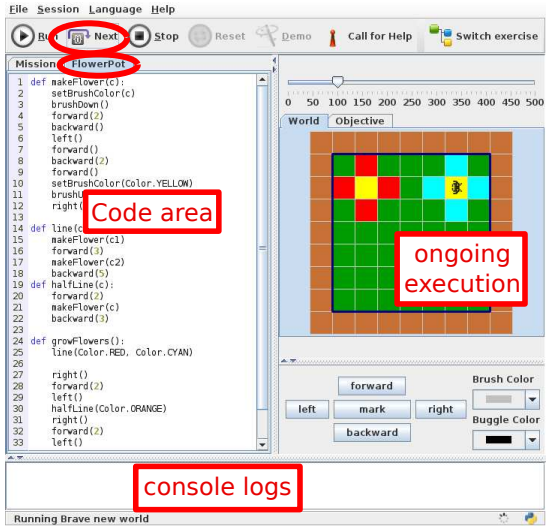
# La PLM (Programmer's Learning Machine)

## Exerciseur interactif dédié à la programmation

- Outil interactif et graphique pour apprendre à coder
- C'est en forgeant qu'on devient forgeron (et qu'on apprend à aimer ça)






## Usage classique

- On lit la mission à gauche, on compare à droite l'état initial et l'état désiré
- On tape le code, on clique sur un bouton, et *ça s'anime* à droite
- Boucle de *feedback* très courte (et motivante pour les élèves)

# Cod Cod Coding

## Activité hebdomadaire de programmation créative

- Une douzaine d'enfants pendant deux heures chaque semaine
- Un doctorant (Inria), un animateur éduc pop (MJC), +référent scientifique
- Objectif d'expérimentation et d'essaimage dès l'an prochain (+réseau Inria)
- `https://iww.inria.fr/codcodcoding/`


- On apprend beaucoup, on échange dans jecode et Inria
- On réfléchi à un workshop pour diffuser

# Sciences Manuelles du Numérique

## Activités proposées (et objectifs)

- Jeu de Nim: introduire le coté implacable des algorithmes
- Crépier psychorigide: tri linéaire pour que chacun découvre un algo
- Baseball multicolore: notion d'algorithme correct, travail du chercheur
- TSP ou pavage: notion de complexité et introduction de NP

## Menée de chaque activité

- Exploration individuelle, remise en commun en groupe, verbalisation d'algo
- Ensuite on joue sans les mains (verbaliser) puis sans les yeux (conceptualiser)
- 10 minutes interactives à plusieurs séances de recherche personnelle guidée

## Usages de ces activités

- Chaque fête de la science ou assimilée: des élèves Telecom Nancy avec moi
- Groupe IREM Nancy: redynamiser l'enseignement des maths au collège
  - Plus efficace pour la *Verbalisation d'énoncés formels* que la géométrie
- Démonstration *Qu'est ce que l'informatique* au Sénat Français le 11 février