# TER
# Simulation d'application dynamiques pour plateformes de calculs hautes performances

Équipe MESCAL

Steven QUINITO MASNADA

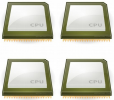
Grenoble, 8 Juin 2015

# Présentation générale

TER dans l'équipe MESCAL, encadré par Arnaud LEGRAND et Luka STANISIC

# Architecture et standard HPC

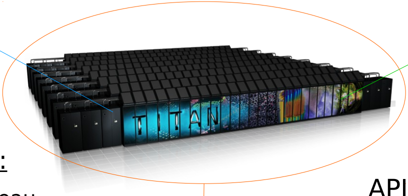
## Multicœurs



OpenMP

API multithread :

- De plus haut niveau que PThread,
- Permet d'exploiter les architectures multicœurs
- Facilite le découpage des traitements



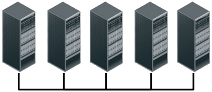

## Hybride

NVIDIA CUDA

API de caculs sur GPU :

- De plus que les sockets
- Mécanismes de comminucation supplémentaires

## Clusters



**MPI**

API de communication :

- De plus haut niveau que les sockets,
- Mécanismes de comminucation supplémentaires (exemple broadcast)

# Limites des approches classiques

- Utiliser plusieurs paradigmes $\rightsquigarrow$ programmation complexe
- Exemple pour exploiter efficacement un GPU sur un seul noeud:
  - transférer données du CPU au GPU,
  - lancer le calcul sur le GPU
  - gérer synchronisation pendant attente résultat
  - occuper CPU
  - récupérer résultat.
- Et avec plusieurs noeuds?
- Statique, système réglé comme un horloge $\rightsquigarrow$ pas portable.
- Solution: Dynamique mais presque impossible avec APIs classiques.

# Nouvelle approche: Paradigme de tâches

- Nouvelle abstraction: les tâches
  - Plus besoin de se soucier de la ressource sur laquelle le traitement est effectué.
  - Exprimer calcul en graphe de tâches $\rightsquigarrow$ système dynamique plus simple.

img/task_graph.jpg

- Librairie StarPU:
  - Système runtime
  - basé sur le paradigme de tâches $\rightsquigarrow$ graphe de dépendances.
  - Ordonnancement dynamique et opportuniste.
- Problèmatique : Performances difficiles à évaluer
  - Configuration runtinme, heuristique, politique ordonnancement.
  - Configuration application, découpage des tâches.

# Test sur systèmes réels

- Exécution réelle sur la plateforme cible $\rightsquigarrow$ coûteux
- Exéuction non déterministe nécessite de réaliser beaucoup d'expériences $\rightsquigarrow$ extrapolations difficiles.

# Simulation

## Généralités

- Utilisation de modèles pour prédire comportements.
- Permet s'affranchir de la plateforme $\rightsquigarrow$ peu coûteux.
- Contrôle paramètres $\rightsquigarrow$ systèmes déterministes.
- Extrapolation simplifiée.
- Exécution plus courte.

## Simulation par rejeu de trace

Exécution post-mortem: pas adapté ici car flot de contrôle non déterministe.

## Hybride simulation / émulation

- Simuler plateforme et OS.
- Emuler de l'application.

## Choix

# SimGrid

# StarPU MSG

# StarPU SMPI

# Project-Team Composition

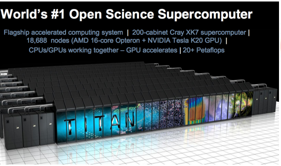
- *Natural* evolution of the MESCAL team.

| Name | Affiliation | Provenance | Expertise |
|---|---|---|---|
| V. Danjean | MdC UJF | MOAIS | HPC, Tracing, Experimental Methodology |
| N. Gast | CR2 Inria | MESCAL | Optimization, Stochastic Modeling |
| B. Gaujal | DR1 Inria | MESCAL | Modeling, Optimization, Game Theory |
| G. Huard | MdC UJF | MOAIS | HPC, Tracing, Visualization |
| A. Legrand | CR1 CNRS | MESCAL | HPC, Simulation, Visualization, Optimization |
| F. Perronnin | MdC UJF | MESCAL | Simulation, Stochastic and fluid models |
| P. Mertikopoulos | CR2 CNRS | MESCAL | Optimization, Game/Information Theory |
| J.M. Vincent | MdC UJF | MESCAL | HPC, Modeling, Simulation, Visualization |

- Inria field / theme:
  - Network, Systems and Services
  - Distributed Computing / Distributed and High Performance Computing
- Keywords: HPC/large distributed systems, performance analysis, distributed and stochastic optimization, . . .

# Context and Objectives

- Large distributed infrastructures
  - **HPC/cloud/...**
  - Wireless networks
  - Smart grids
  - Transportation systems




- Common questions scalability, resilience, adaptability, capacity planning, energy consumption, ...
- Common characteristics ever growing size, distributed, heterogeneous, user-centric $\rightsquigarrow$ stochastic nature

- This requires involved tools and new techniques that will be useful to the D&HPC community

# Scientific Foundations: POLARIS in a Nutshell

*Contribute to the understanding (from the observation, modeling and analysis to the optimization through adapted algorithms) of performances of very large scale distributed computing systems by applying original ideas from other research fields and application domains.*

**POLARIS = Team of people with the right spectrum of skills**

Experiment design measuring/monitoring/tracing tools, experimental methodology (design, control, reproducibility)

Modeling and Simulation discrete event simulation, emulation, Markov chains, perfect sampling, Monte Carlo methods, ...

Visualization and Statistical Analysis workload characterization (failures, parallel systems), visualization and analysis of parallel applications

Optimization stochastic approximations, mean field limits, game theory, mean field games, primal dual optimization, learning, information theory

# Research Methodology

A continuum of 5 research areas

- Measurement design of experiments, observation overhead control, reproducible research
- Visualization performance qualification and debugging, multi-scale visualization, trace comparison
- Simulation faithful simulation of HPC systems, sensibility/robustness, trajectory coupling
- Fluid Modeling local interactions, transient analysis
- Optimization learning algorithms in continuous nonlinear games, online and distributed optimization

# Research Methodology

A continuum of 5 research areas

- **Measurement** design of experiments, observation overhead control, reproducible research
- Visualization performance qualification and debugging, multi-scale visualization, trace comparison
- **Simulation** faithful simulation of HPC systems, sensibility/robustness, trajectory coupling
- Fluid Modeling local interactions, transient analysis
- Optimization learning algorithms in continuous nonlinear games, online and distributed optimization

# **Measurement:** Reproducible Experimental Methodology

Real experiments are costly, difficult to control and to reproduce

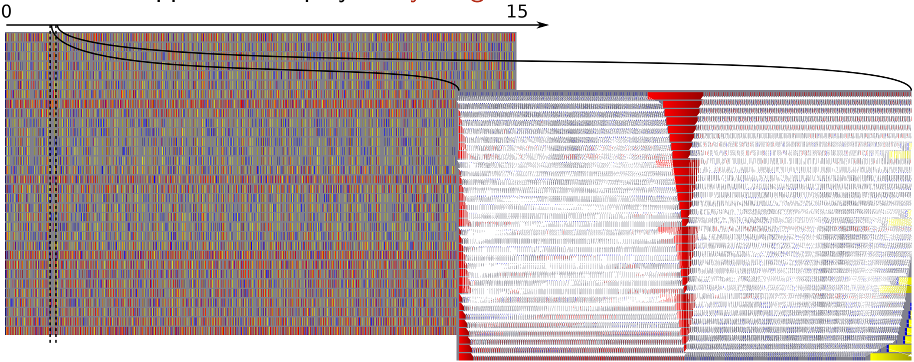
- Cannot be studied anymore like artificial systems. Need to inspire from other experimental fields

**Research directions**:

- Design of experiments: involved statistical technique widely used in all fields where experiments are expensive but CS
  - Bridge this gap and favor its adoption in the D&HPC theme
- Monitoring and tracing: need for multi-scale (application/space/time) observation where intrusiveness is controlled
  - Evaluate the observation/analysis quality trade-off
- Open science and reproducibility: complexity and rapid technological evolution = excuse for not taking care of results reproducibility
  - Monitor/document the whole process (design, execution, data gathering, filtering, analysis)
  - Investigate/design pragmatic workflows to alleviate this flaw

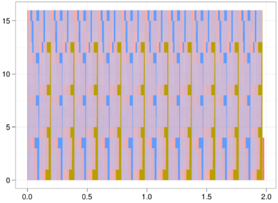
# **Visualization:** "Performance Driver" Identification

Traditional approach: display everything

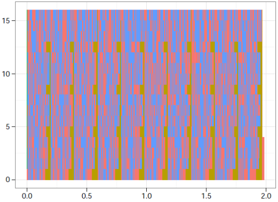


# **Visualization:** "Performance Driver" Identification

Traditional approach: display everything
$\rightsquigarrow$ harmful biases (*more information than what fits on your screen*)




Evince




Acroread

$\rightsquigarrow$ *overenthusiastic* use of *clustering*, pattern *mining*, *sequence alignment*

**Research Directions**:

- Performance qualification and debugging
  - Colleagues from D&HPC theme in deep need of new approaches/tools
- Multi-scale analysis (space/time/application) resilient to noise
  - Entropy-based Aggregation applied to embedded/HPC systems
- Trace comparison

# Simulation: Very Large Stochastic Systems

- Simulation circumvents some of the previous experimental issues
  - cost/screening, extrapolation, capacity planning, ...
- Traditional approach: simplistic models to study large-scale systems, developed by D&HPC experts who know little about simulation
  - Short-lived tools with no intent of predicting anything. At best grossly indicates trend but no more expectation

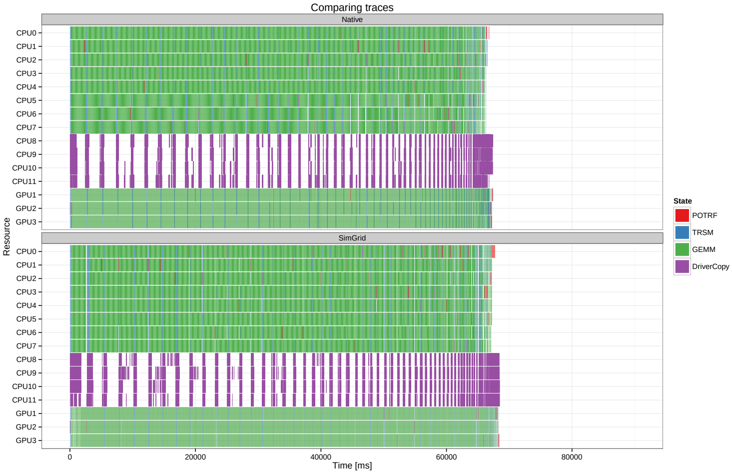
**Research directions**:

- Accurately reproduce the dynamic of real systems
  - SimGrid: Versatile simulation of large-scale distributed systems coarse-grain fluid models, mix emulation/simulation, invalidation
  - Used is RUNTIME/HIEPACS, ASCOLA, KERDATA, AVALON, ...
- Provide sensibility analysis and robustness indicators
- Trajectory coupling for discrete event simulations
  - $\text{PSI}^2$: Perfect sampling for Markovian systems

# Simulation: Very Large Stochastic Systems

- Simulation circumvents some of the previous experimental issues

## Simulation of Cholesky/StarPU on a hybrid platform



# **Analysis:** Local Interactions and Transient Analysis

Analysis of stochastic systems is particularly difficult but mean-field approximation is suited to large systems

- Key hypothesis: the dynamic solely depends on the entity state (not on their identity nor on their spatial location) and state space does not scale

**Research directions**:

- Locality is essential: possible approaches
  - pair approximation from statistical physics
  - fixed interaction graphs and a multi-scale approach
  - *never used for distributed computing systems and high potential*
- Transient behavior:
  - Finite horizon: OK (discrete system is uniformly close to its continuous limit)
  - Infinite time horizon when the continuous limit is globally stable: OK
  - Trajectory dependent stopping time: ???
  - *Could be used to analyze the complexity of distributed algorithms*

# **Optimization:** Game Theory, On-line Distributed Optimization

## Modeling interactions through game theory

Nash equilibrium often inefficient but efficient equilibrium can be learned

- Finite set of strategies = OK. **In**finite set = ???
  - Examples: routing packet flows, power control in wireless networks, . . .
  - Discretizing is not an viable option (state space explosion exponentially hard to analyze, mixed strategy space is irrelevant)
- **Goal**: Design learning algorithms in continuous nonlinear games that can be applied to realistic network scenarios

## Online and distributed optimization

- Common unsatisfactory use of greedy approaches based on offline heuristics
- Each agent is faced with an unknown and evolving loss function and seeks to minimize his cumulative loss via the use of past observations
- Regret minimization: notion at the interface of game theory, optimization, statistics and theoretical computer science
- **Goal**: Develop and apply such techniques to actual systems

Ensure that key practical properties are met (asynchronous operations, numerical stability, robustness to noisy or delayed inputs, low overhead)

# Within Inria and National

Distributed and H.P. Computing/Distributed Systems and Middleware

*Potential* or ongoing collaborations with: DATA-MOVE, (CORSE), AVALON, *ROMA*, STORM, HIEPACS, (*REALOPT*), *TADAAM*, *KERDATA*, MYRIADS, ASAP, REGAL

Other Inria themes

- Optimization of and control of dynamic systems: BIBOP, NECS
- Networks and Telecommunications: MAESTRO, DIOGENE, DIONYSOS, RAP, SOCRATE

Other groups Game theory (LSS/supelec, Ceremade/Dauphine, HEC), Stochastic optimization (Toulouse)

# International

International collaborations

- Inria JLESC (NCSA/UIUC, BSC, Jülich)
- Inria@SiliconValley/Berkeley (BOINC)
- LICIA (UFRGS)
- EPFL
- Univ. of Athens

Connexion with Grenoble industry through CIFRE contracts

- Bull/ATOS, STMicroelectronics, HP, Orange, CEA
- Alcatel, Huawei