UNIVERSITE DE REIMS
CHAMPAGNE-ARDENNE

Contributions à la Gestion de l'Hétérogénéité dans les Environnements Distribués et Pervasifs

Luiz Angelo STEFFENEL

Habilitation à Diriger des Recherches
Soutenance publique
Reims, 8 décembre 2017

CReSTIC

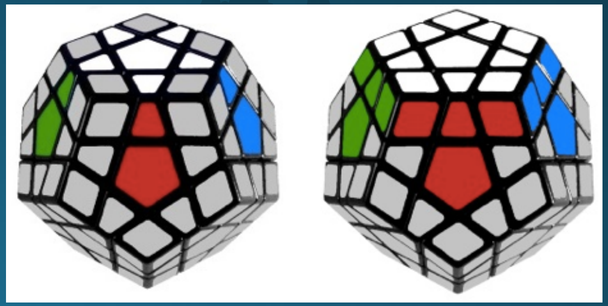
What is a Distributed System?

- Most definitions consider at least these properties
  - Independent elements, loosely coupled (nodes, machines)
  - Coordination is made by message exchange
  - Resources state and capabilities may vary over time

Heterogeneity

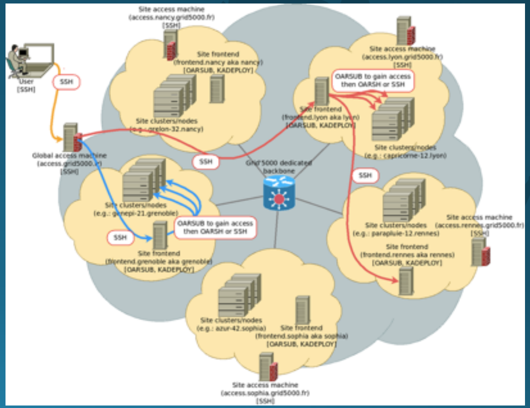
What is Heterogeneity?

- *Noun* - the quality or state of consisting of dissimilar or diverse elements
- There is not only one kind of heterogeneity
  - Diversity in a group
  - Diversity of groups
- Dealing with heterogeneity can be compared to a puzzle-solving problem



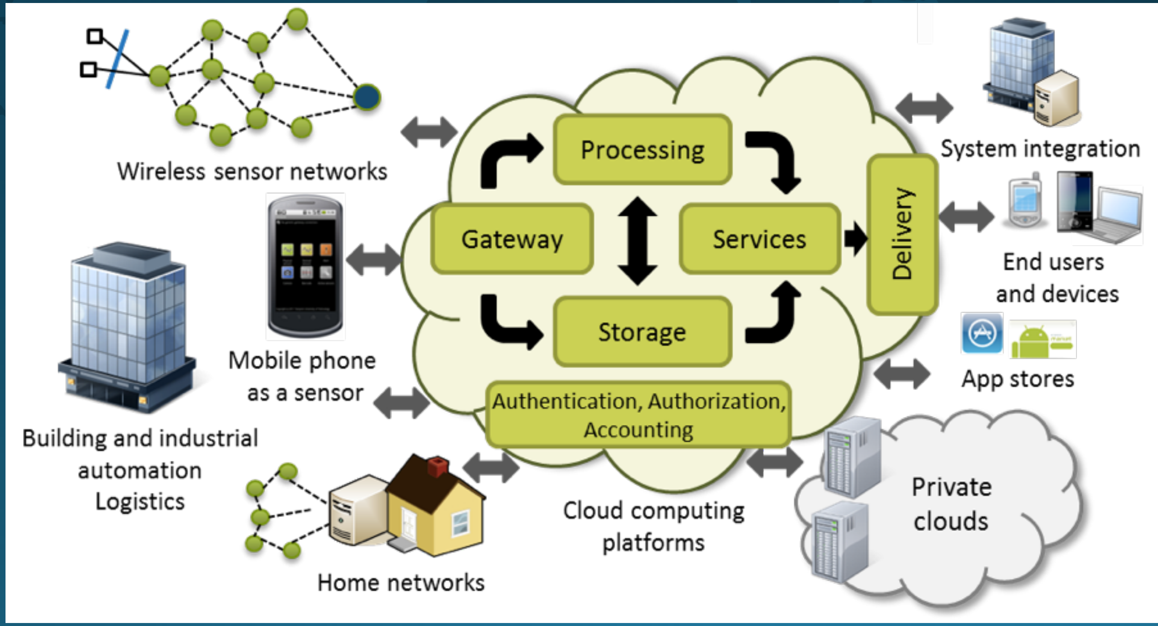
Heterogeneity in Distributed Systems

- Different views and classifications can be used, for example:
  - Material heterogeneity
  - Communication heterogeneity
  - Task heterogeneity
  - Data access heterogeneity
  - Variations through time
- The complexity tends to augment as the systems grow



Towards Pervasive Computing

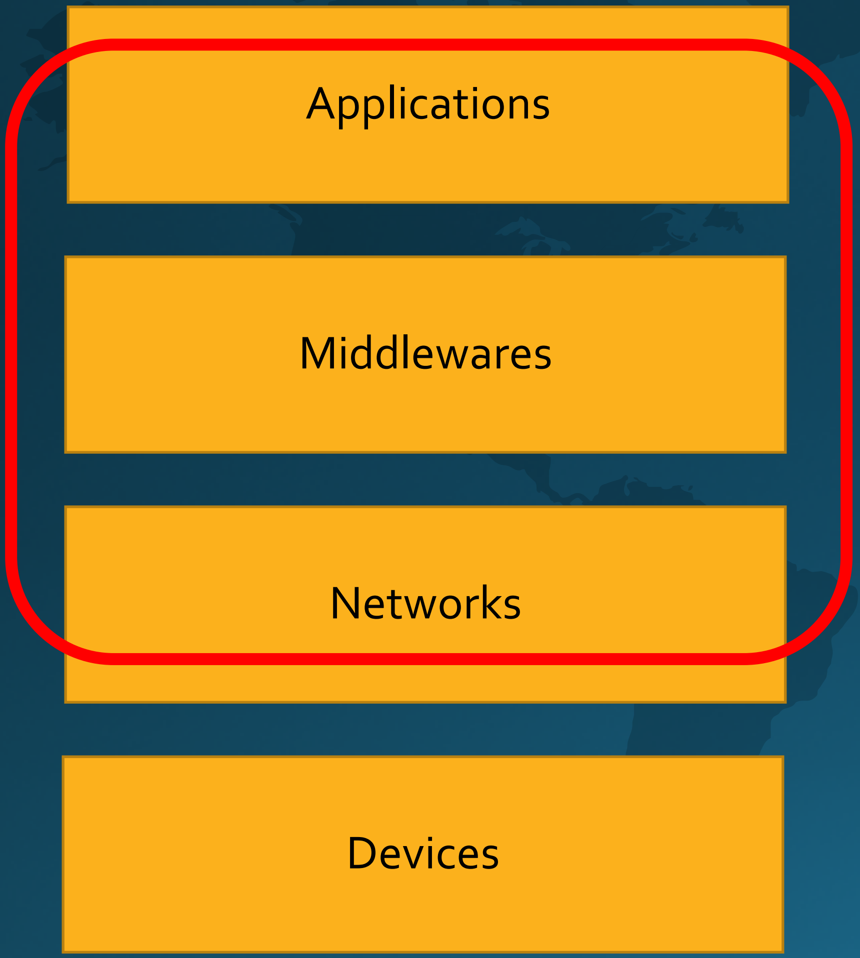
- Cloud, Mobile computing and IoT induce even more heterogeneity
  - Distant resources that are "somewhere" in the cloud
  - Mobile devices that move through the networks
  - Material diversity and capacity constraints



My Motivation

- To investigate and to develop (elegant) solutions for problems caused by heterogeneity
  - Sub-optimal performances
  - Lack of adaption, lack of context-awareness
  - Uncertainty about the resources (volatility, discovery, etc.)
  - Bad scheduling or placement of tasks and services

Contributions to Heterogeneity Management



- Modeling and optimizing communication heterogeneity for grids
- Dealing with task heterogeneity when parallelizing a biology application
- Context-awareness techniques for resources dynamicity
- Middleware development experiences

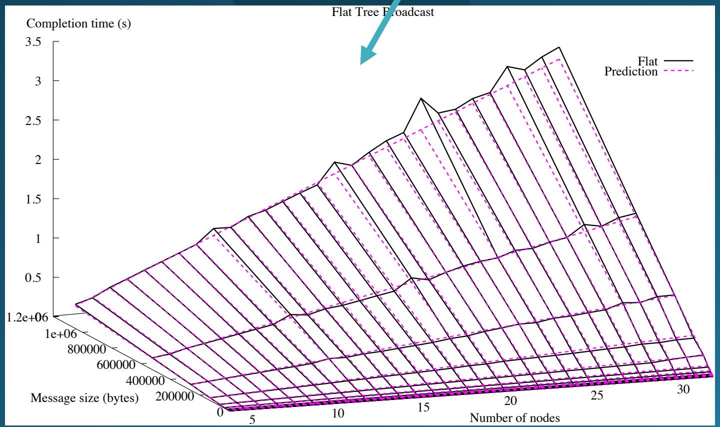
Communication Heterogeneity

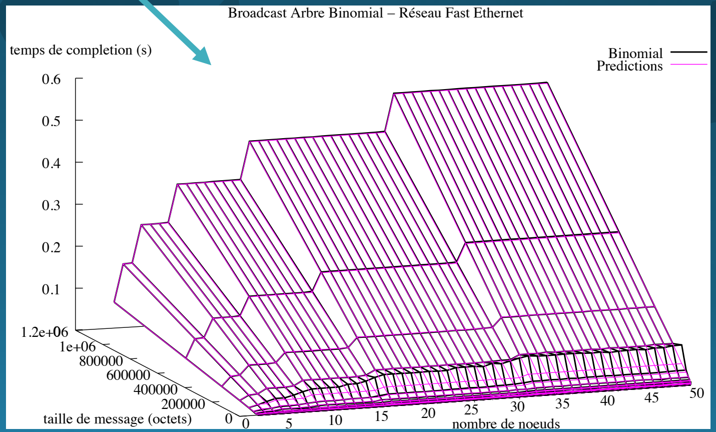
Modeling and Optimizing Collective Communications

Modeling communications

- Communication heterogeneity impacts the coordination of nodes
- In local networks we can approximate these costs
  - Just need a good model and good parameters
  - For example, the Broadcast (one-to-many) communication pattern

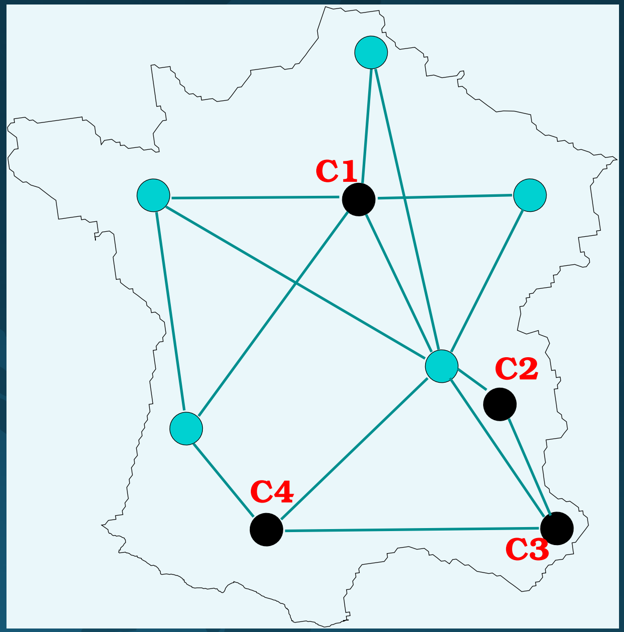
Strategy	Communication Model
Linear (Flat Tree)	$L + (P-1) \times g(m)$
Pipeline (Segmented Chain)	$(P-1) \times (g(s) + L) + (g(s) \times (k-1))$
Binary Tree	$\leq \lceil log_2 P \rceil \times (2 \times g(m) + L)$
Binomial Tree	$\lceil log_2 P \rceil \times L + \lfloor log_2 P \rfloor \times g(m)$





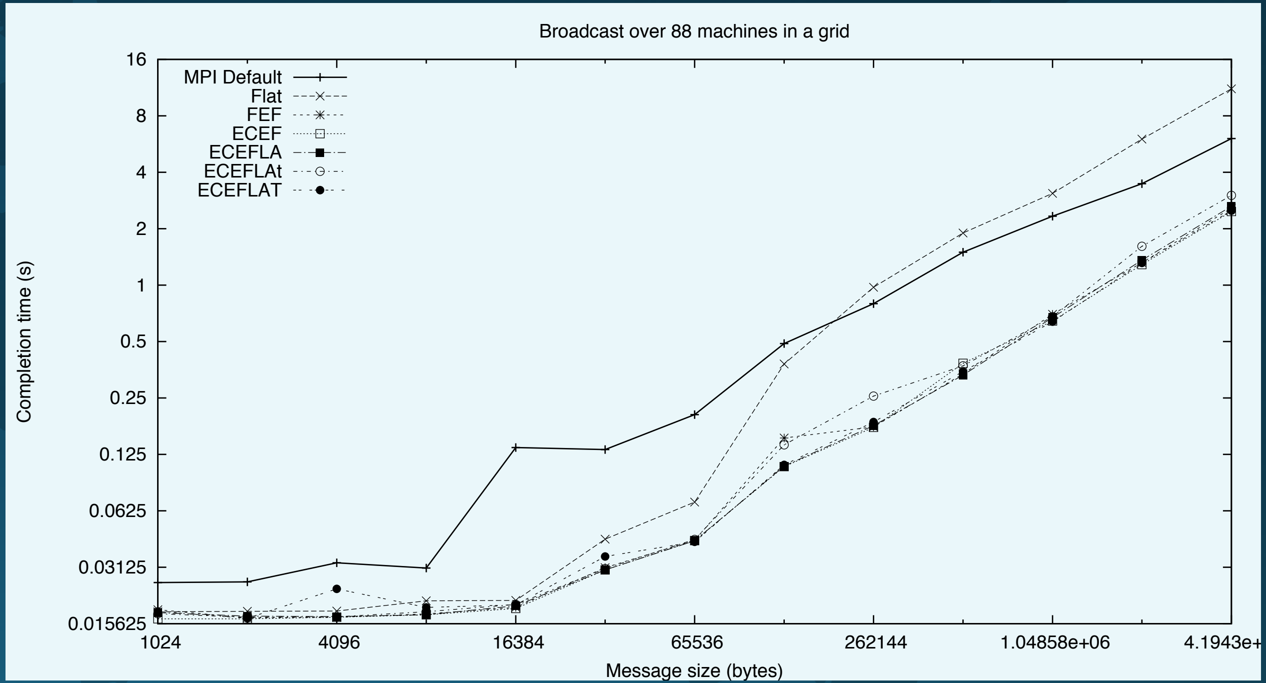
Broadcasts in grids

- In a grid, we have different network performances
  - Between clusters
  - Inside clusters
- One single model cannot represent all variations
- For simplicity, we can structure in 2 levels
  - **Internal** – chose the best algorithm from a set of existing models/implementations
  - **Inter-cluster** – specially tailored broadcast tree that interconnects "cluster heads"



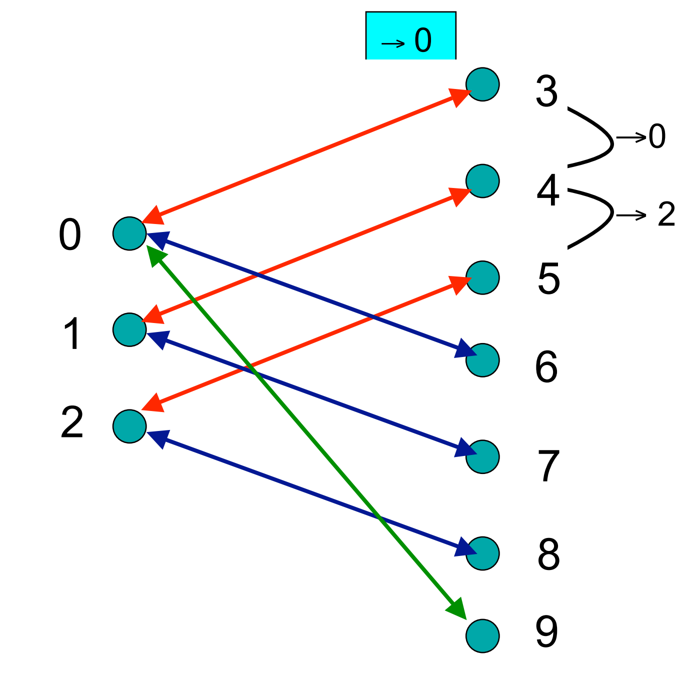
Broadcasts in grids - heuristics

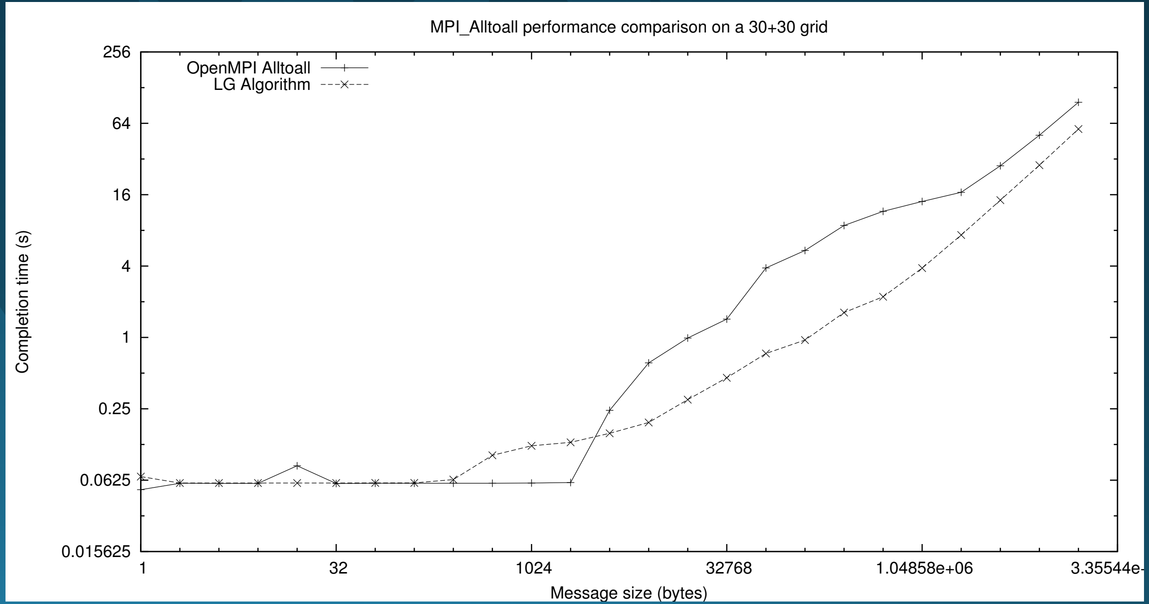
- We need to construct an ad-hoc broadcast tree
- Select the "next cluster head" according to different optimization rules
  - Faster nodes
  - Minimize intra-cluster comm
  - Maximize source nodes
  - ...



What about All-to-All?

- All-to-All is a more complex pattern
  - n(n-1) different data to send through the network
    - 2n messages over the inter-cluster link (slow)
  - **Highly impacted by network contention**
- Why not regroup all msgs for a destination **before** sending them?
- LG algorithm
  - Defines "corresponding" peers
  - Minimizes contention



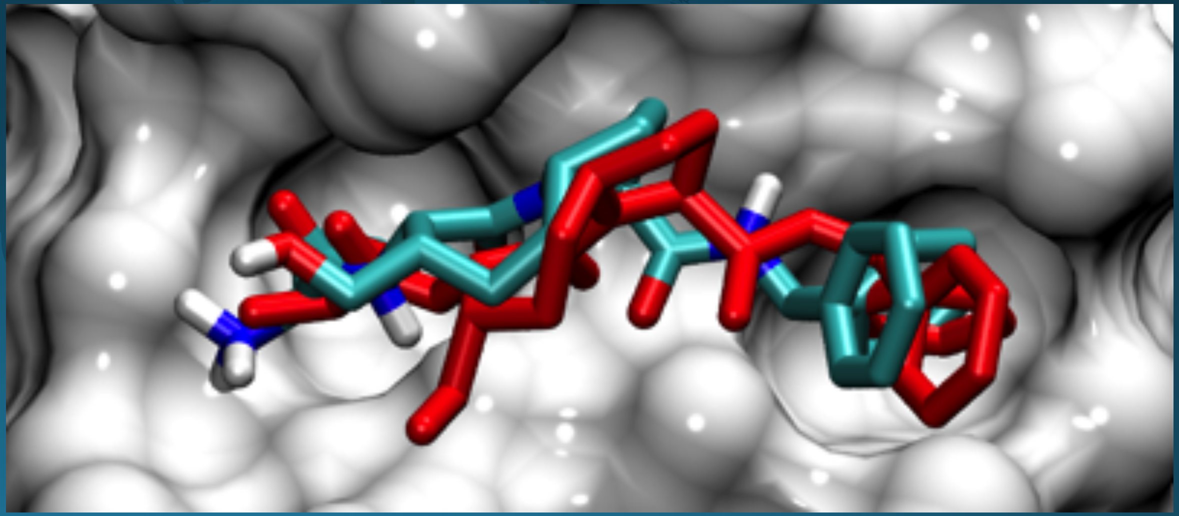


Task Heterogeneity

Parallelizing a bioinformatics application

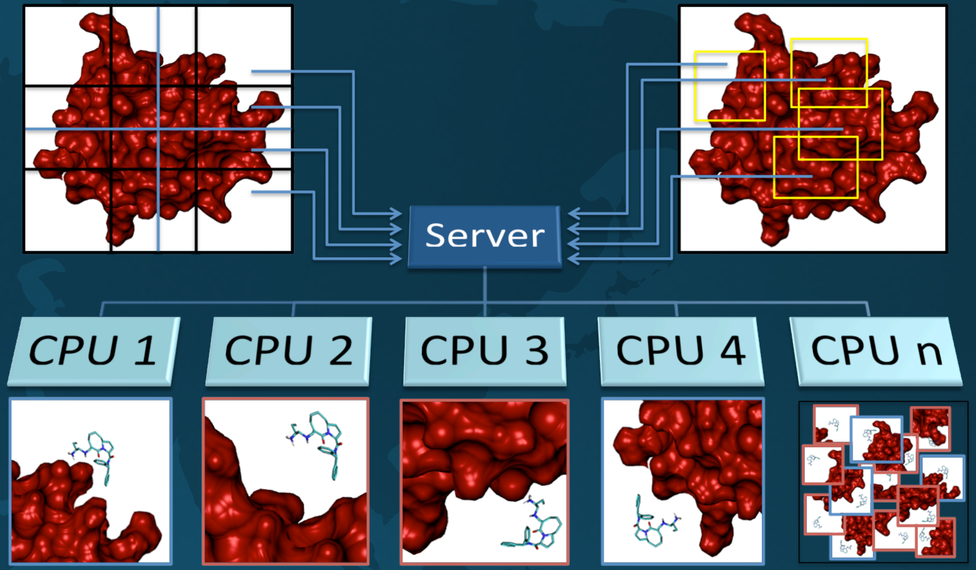
Motivation

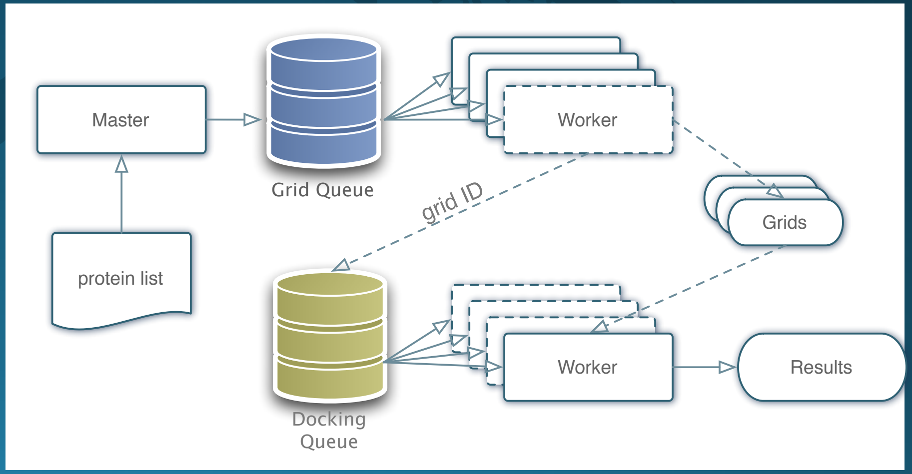
- Subject of a PhD thesis in Bioinformatics
  - Romain Vasseur – Inverse Docking
  - Used in drugs research by the pharmaceutic industry
- In this case, the "reference" application is a "monolithic" simulator
- We seek to explore databases with hundreds of proteins
  - One protein-peptide docking exploration may take hours
- How to parallelize it without touching the simulator code?



Parallelizing the Molecular Docking

- Solution: divide the data… But how?
  - Geometric decomposition
  - With superposition
- Development of AMIDE, a framework for data decomposition and task scheduling
  - Workflow for generating subgrids and deploying tasks
  - Can be used both in an ad-hoc environment or through a resource manager (like Slurm)





Context Adaption

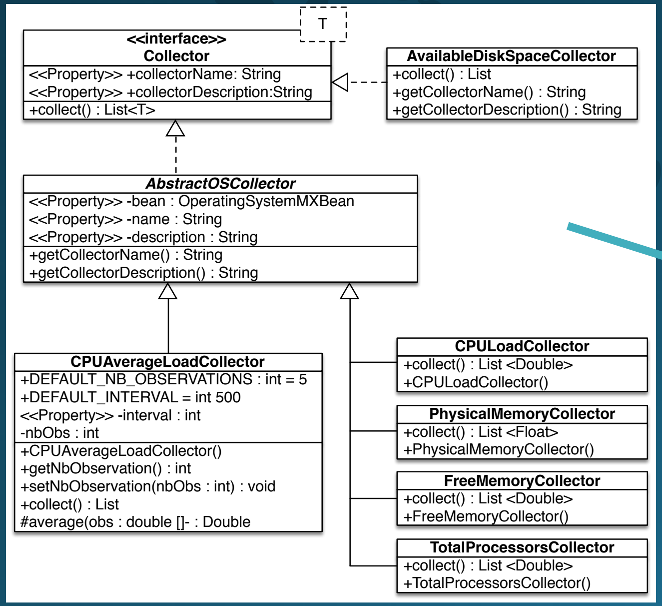
Improving Apache Hadoop

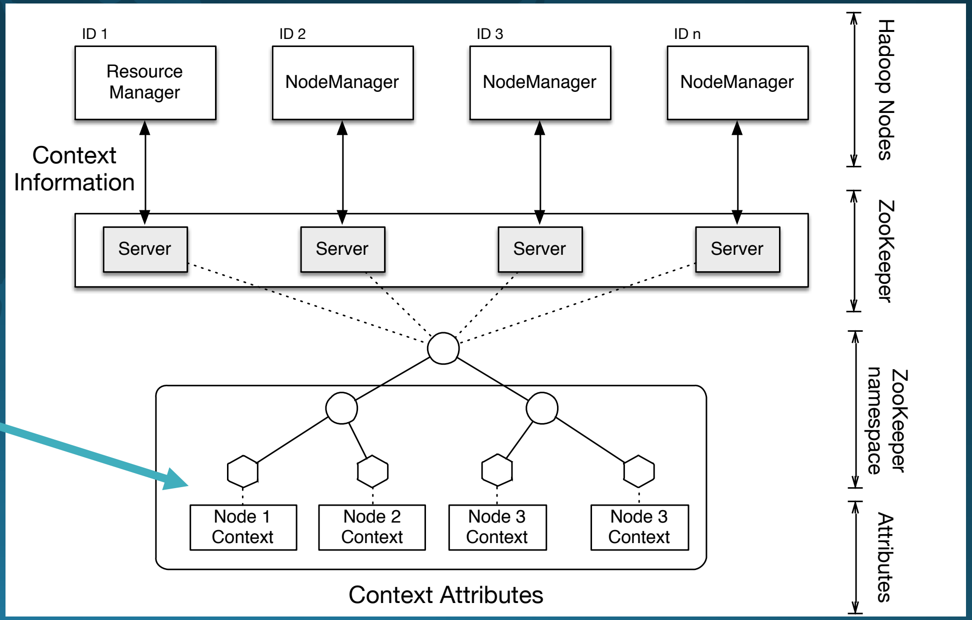
Improving Apache Hadoop

- Hadoop was designed to operate in clusters
  - Static and stable configuration (XML files)
- Does not react to changes in the resources capabilities
  - There is no "live" adaption
- Project PER-MARE aimed at deploying Hadoop over pervasive networks
  - Heterogeneous nodes (SoC, desktop grids, etc.)
  - Volatile nodes
- We need context information

Acquiring and Injecting Context Information

- The *ResourceManager* is the source of information for the schedulers
  - Its data comes from XML config files
- We created a context collector that updates the information through Zookeeper





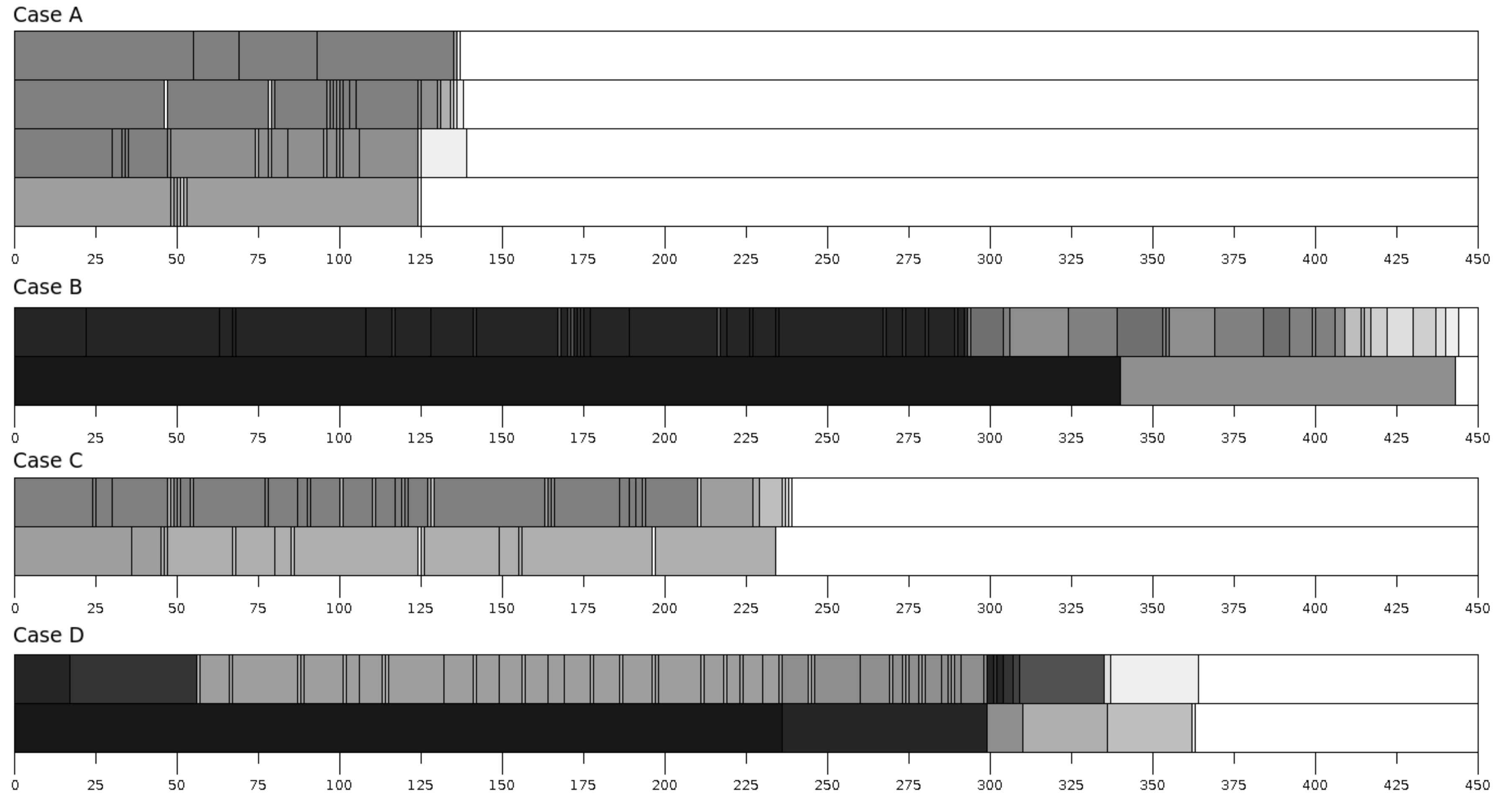
Impact of Context Change

4 nodes ok

2 nodes with the configuration of 4 nodes

2 nodes
Context updated from beginning

2 nodes
Context updated after the execution started



Middleware Development

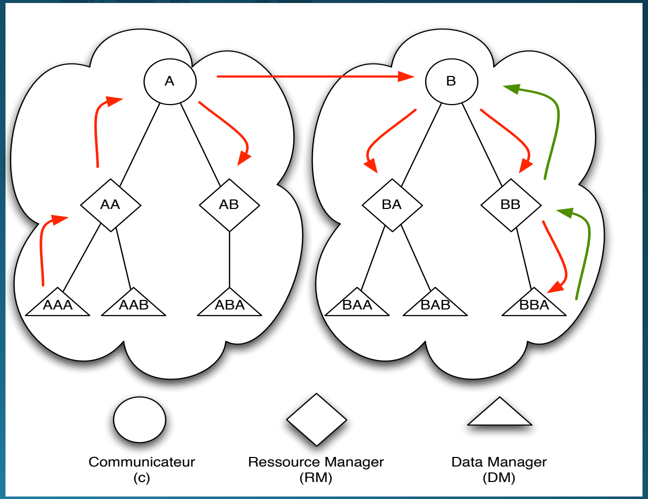
Learning steps

The Experience of GRAPP&S

- Subject of a co-tutelle PhD
  - Originally aimed to extend CONFIIT
- Tentative to specify an overlay for transparent data access

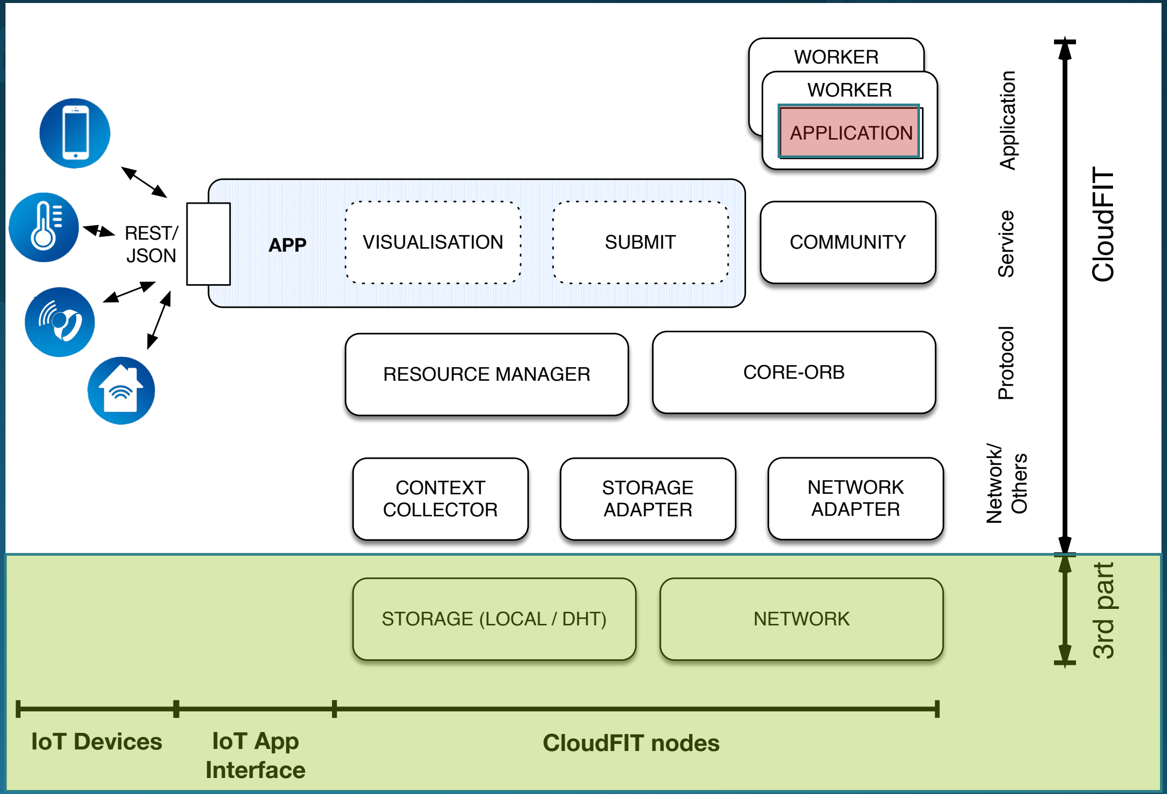
- The work never advanced beyond the specification phase
  - Over time, other APIs filled the gap (Apache Jena, etc.)

- Many possibilities that were never explored
  - Data relocation (like a "dynamic" Amazon Glacier)
  - Development and test of proxies
    - format identification standards (MIME, RDF, etc.)



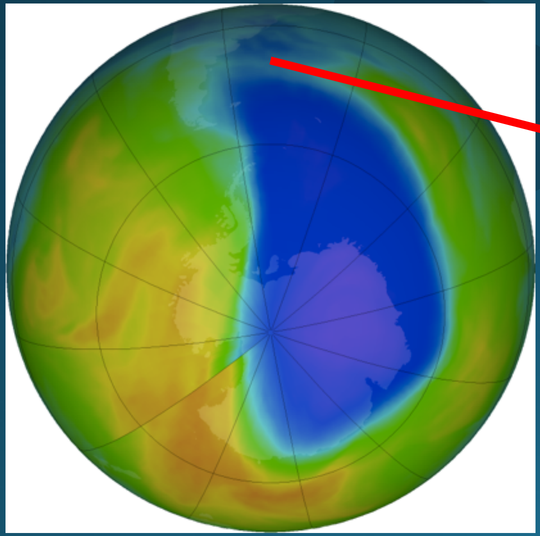
The CloudFIT framework

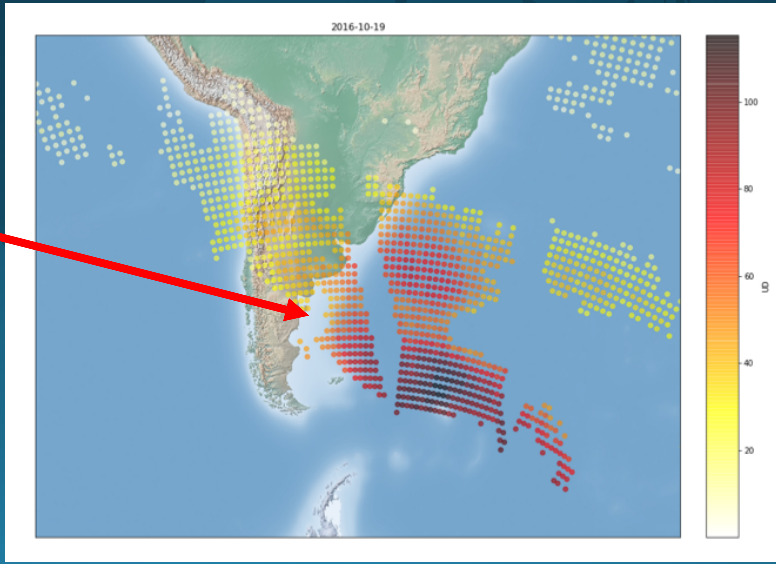
- PaaS computing framework based on P2P overlays
- Designed for computing on pervasive environments (PER-MARE project)
  - Fault tolerance
  - Decentralized scheduling
  - DHT storage with replication
- Modular and Extensible
  - Different P2P overlays
  - Context, scheduling, IoT
- FIT API
  1. How many tasks?
  2. What a task must do?
     - Which data to access, which actions
     - Optional task dependency (DAGs)



CloudFIT at work

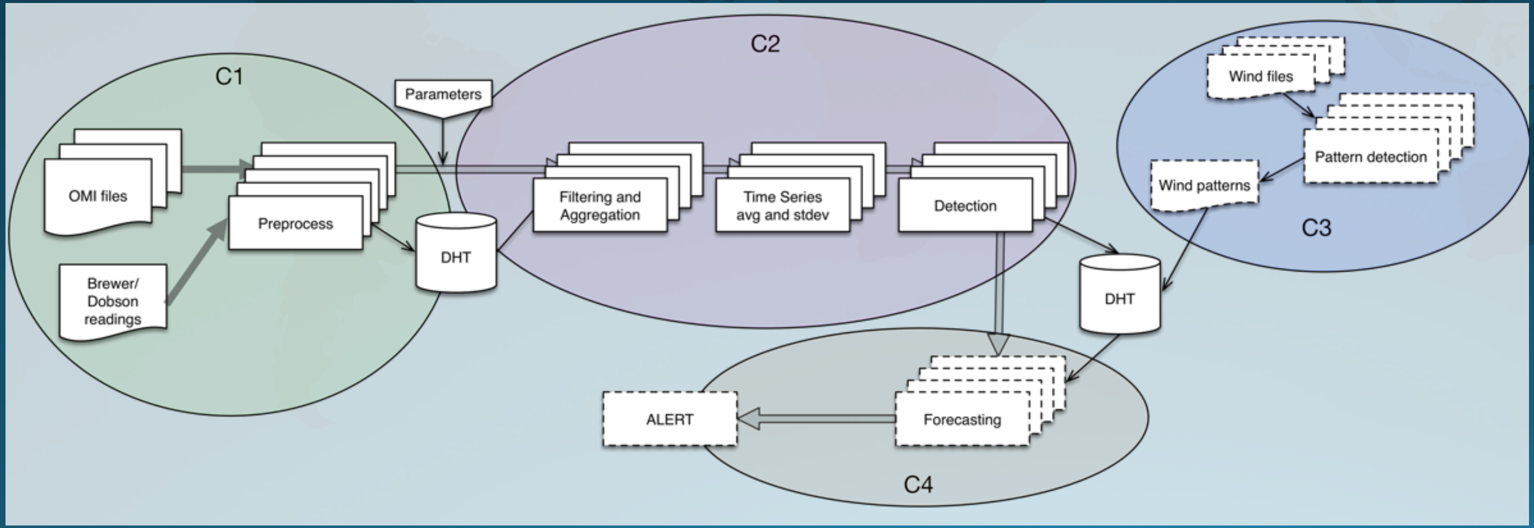
- Map-Reduce applications (benchmark against Hadoop)
- Ozone Secondary Events (OSE) detection
  - **Drastic reductions** on the ozone column that may reach medium latitudes and cause high UV exposition to the population
  - Caused by air masses that detach from the polar vertex
  - We created an OSE detection workflow





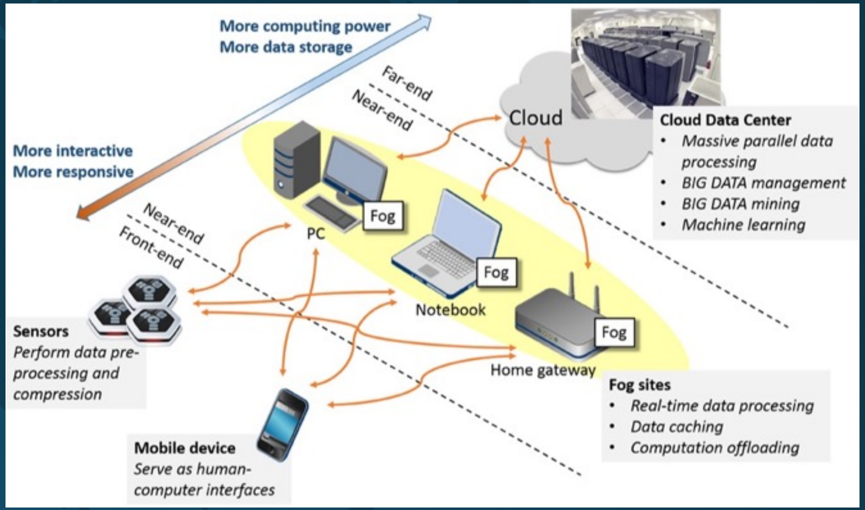
CloudFIT at work

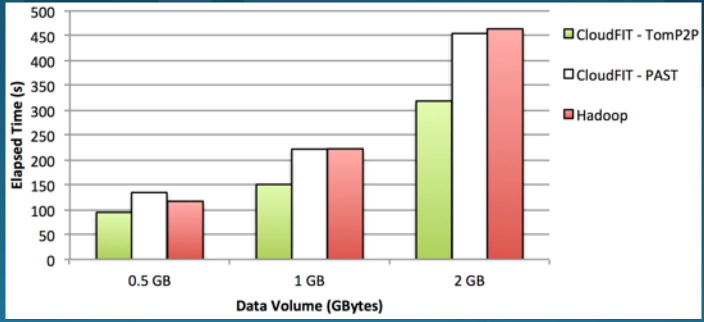
- Map-Reduce applications (benchmark against Hadoop)
- Ozone Secondary Events (OSE) detection
  - **Drastic reductions** on the ozone column that may reach medium latitudes and cause high UV exposition to the population
  - Caused by air masses that detach from the polar vertex
  - We created an OSE detection workflow

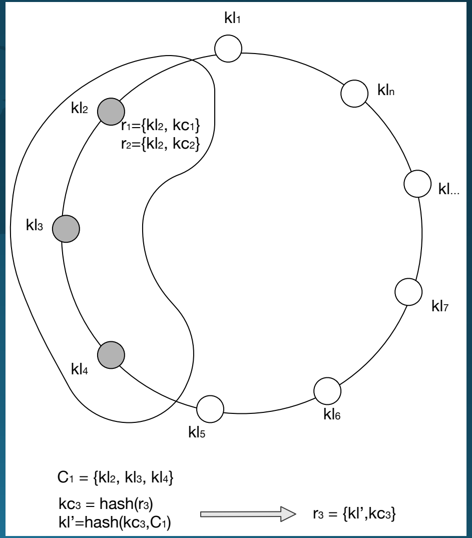


CloudFIT Perspectives

- Focus on fog/edge computing
  - Development of multi-scale applications
  - Context-awareness and automatic clustering to help task placement
- Data locality for big data applications
  - DHTs often spread data, losing locality
  - Access times may impact Big Data applications
  - Improve data locality through
    - Locality reinforcement (DHT trick)
    - Check if data is local (scheduling)

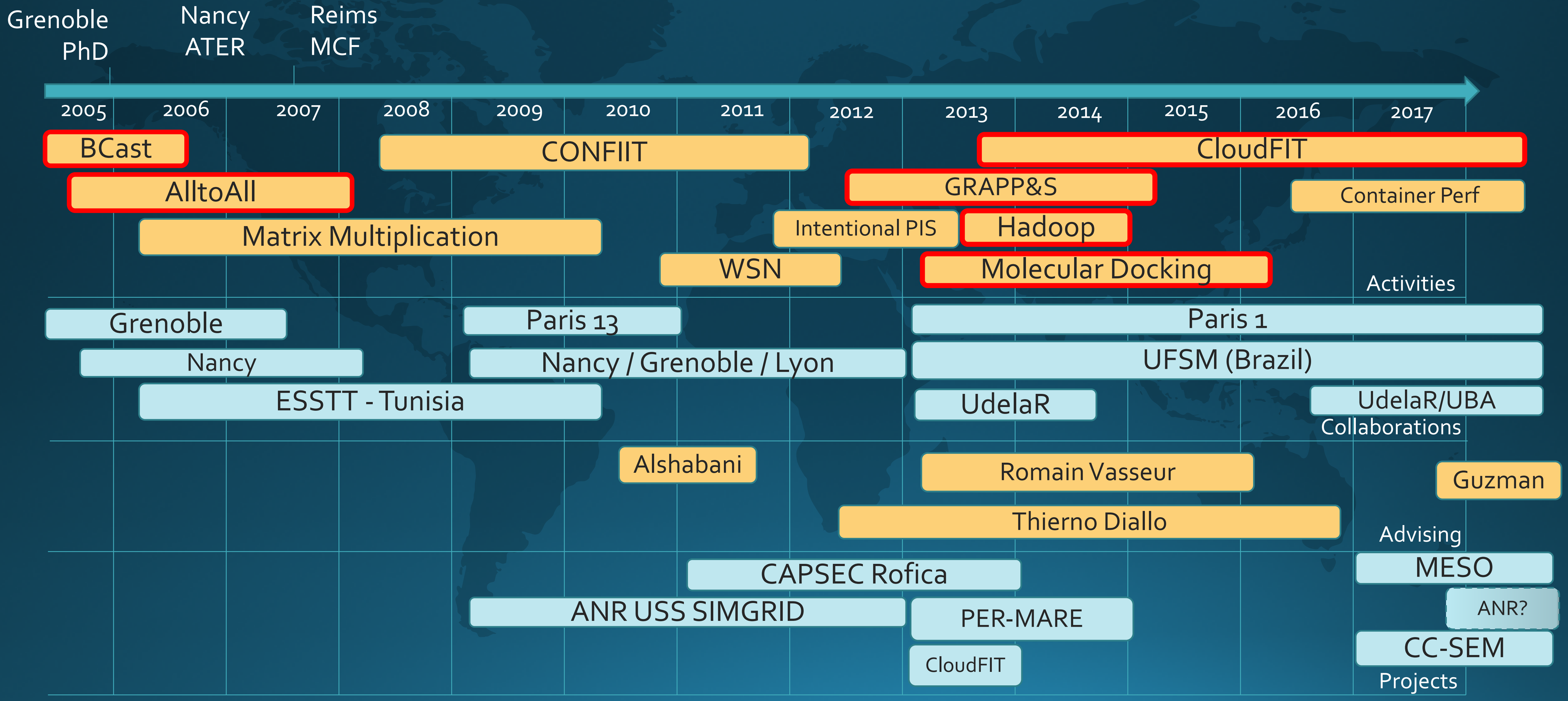






Conclusions and Perspectives

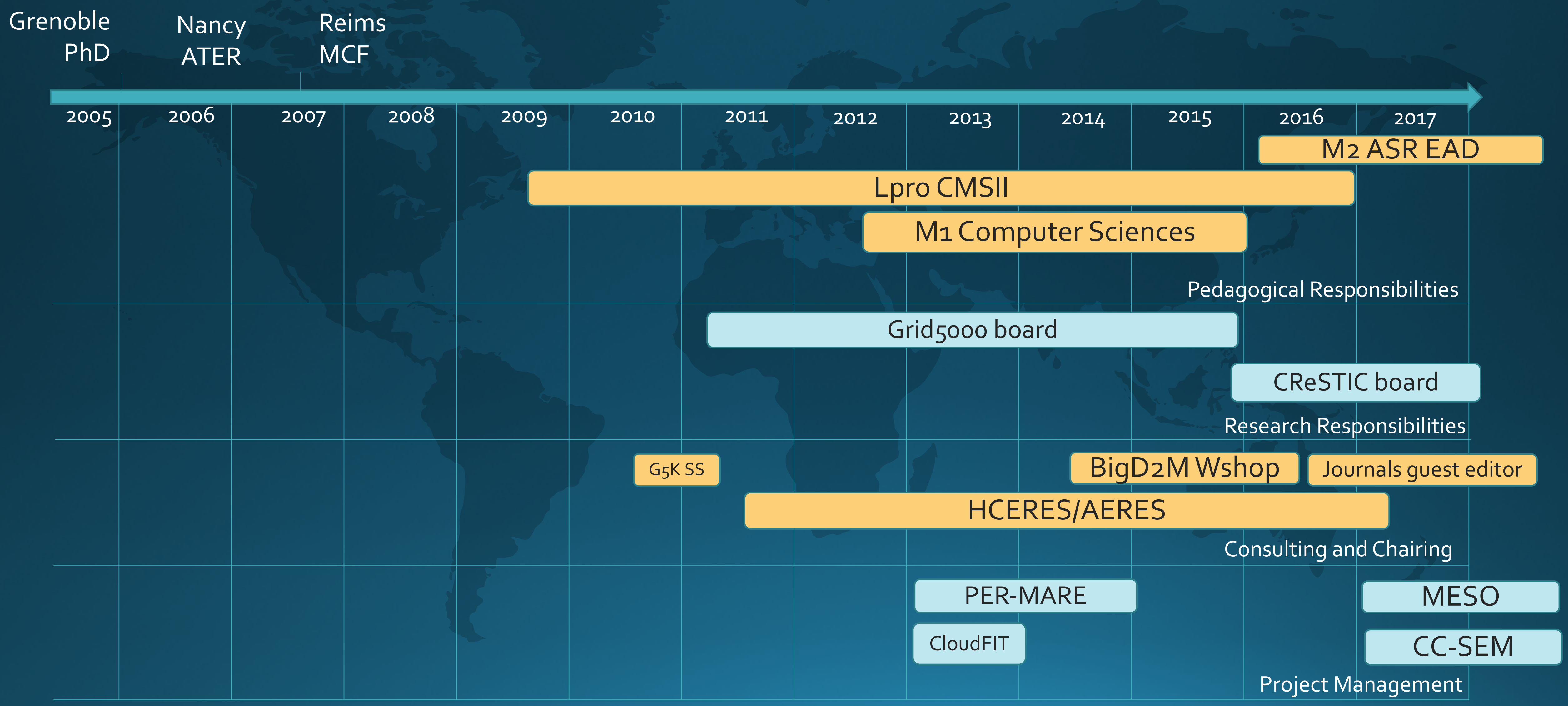
Research Summary



Research Domains

- Fog Computing and Pervasive environments
  - Context adaption, virtualization and microservices
  - Integration with SDN
- Internet of Things and Smart Cities
  - Smart Agriculture
  - Energy consumption monitoring
  - Green IT
- Distributed Computing and Big Data
  - Integration with fog computing applications and IoT
  - Exploration of new tools (Apache Storm, ...)
  - Data handling and analysis

Administration Responsibilities



UNIVERSITE DE REIMS
CHAMPAGNE-ARDENNE

Thank you for your attention

Contributions à la Gestion de l’Hétérogénéité
dans les Environnements Distribués et Pervasifs

Luiz Angelo Steffenel

Academic titles

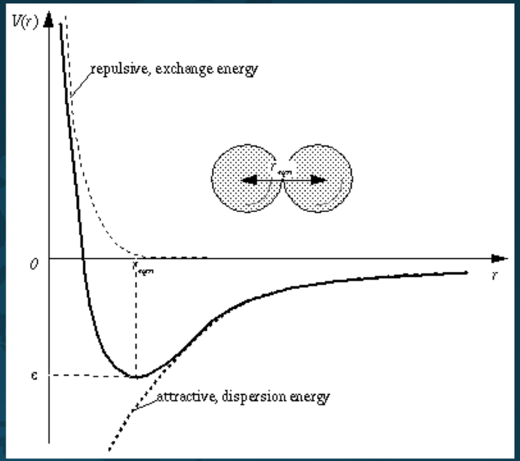
- PhD in Computer Science
  - Institut National Polytechnique de Grenoble, France - 2005
  - Subject: Modeling and Optimization of Collective Communications in a Grid
  - Supervisors: Denis Trystram and Grégory Mounié
- Doctoral School in Communication Systems - 2002
  - EPFL Lausanne, Switzerland
  - Subject: Algorithms for Distribute Membership (Fault Tolerance)
  - Supervisor: André Schiper
- MSc in Computer Sciences - 2001
  - Universidade Federal do Rio Grande do Sul, Brazil
  - Subject: Algorithms for the Consensus agreement problem (Fault Tolerance)
  - Supervisor: Ingrid Jasch Porto

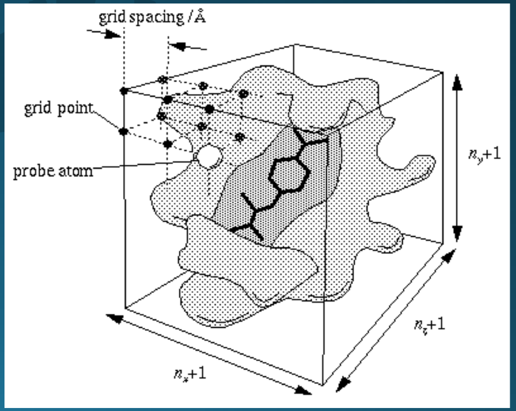
CReSTIC

Molecular Docking 101

- Molecular docking is a simulation technique to identify good binding conformances between two molecules
  - In our case, protein-peptide
- A "blind docking" consists in trying multiple arrangements and comparing the energy score
- Good coverage need hundreds of tentatives
- The data provided has two 3D grids:
  - The location (and energy) of each protein atom
  - The location (and energy) of each peptide atom

$$\begin{aligned}\Delta G = C_{vdw} \cdot \sum_{i,j} \left(\frac{A_{ij}}{r_{ij}^{12}} - \frac{B_{ij}}{r_{ij}^{6}} \right) \\ + C_{hbond} \cdot \sum_{i,j} E(t) \left(\frac{C_{ij}}{r_{ij}^{12}} - \frac{D_{ij}}{r_{ij}^{10}} + E_{hbond} \right) \\ + C_{elec} \cdot \sum_{i,j} \frac{Q_i Q_j}{\epsilon(r_{ij}) r_{ij}} \\ + C_{tor} \cdot N_{tor} \\ + C_{sol} \cdot \sum_{i,j} S_i V_j e^{(r_{ij}^2/2\sigma^2)}\end{aligned}$$





The Design of CloudFIT