The QOsCosGrid Project:

Quasi-Opportunistic Supercomputing for Complex Systems Simulations

Description of a general framework from different types of applications

Magali Charlot

Ibergrid 14-16 May

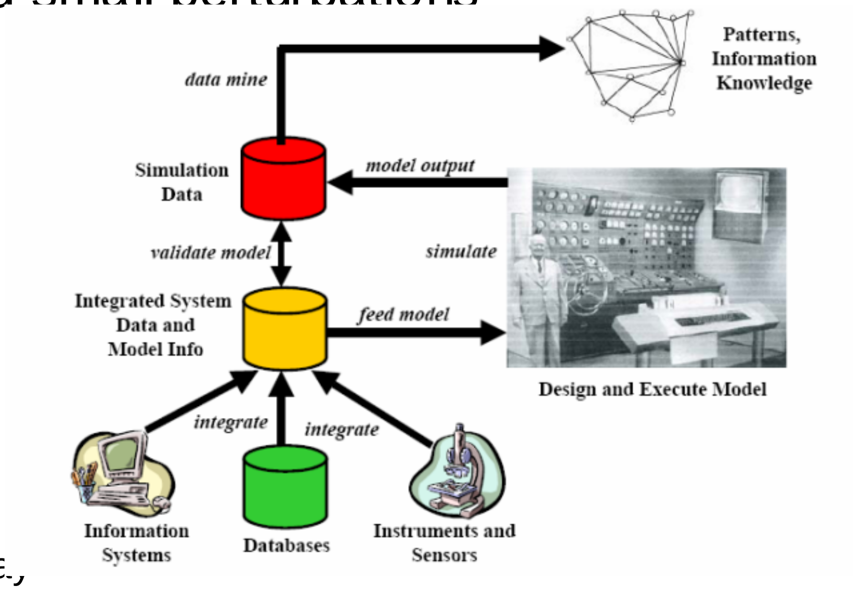
Plan

- Introduction about the complex systems
- The QosCosGrid project: partners and objectives
- The different use cases and their characteristics
- General framework for parallelization and communication
- Conclusion

Complex Systems

- Many real-world systems involve large numbers of highly interconnected heterogeneous elements.
- A system is complex if it is characterized by multiple interactions between many different components.
- Complex systems are Systems that constantly evolve in time according to a set of rules.
- It is characterized by evolution such that it may be highly sensitive to the initial conditions and small perturbations
- The rules are usually non linear making it difficult to understanding and verify

The methodologies used to understand their properties involve modelling & simulation that usually require supercomputers.



Overview of the qosCosGrid project
(FP6 European Commission project)

- Develop quasi-opportunistic supercomputing grid middleware and services that provide:
  - dynamic resource brokering giving the best QoS to any given CS simulation,
  - reservation of resources, communication, synchronization and routing
- Research the non-trivial parallelisation of the CS applications and to adapt the underlying algorithms to this environment.
- Proof-of-concept demonstrations exemplifying various types of CS applications.

QosCosGrid Goals

- Technology facilitating dynamic selection of the available grid resources suitable for CS simulation
- Functionalities such as reservation of resources, synchronization and routing communication
- Support of Quality of Service policies on which reservation could be based
- Grid architecture capable of tolerating system instabilities and of treatment of non-dedicated resources in the context of complex systems simulations
- Virtualisation of the resources and applications to hide the underlying complexity of the system and enabling an easy access and use
- Technology for data distribution suitable for CS applications
- Non trivial parallelisation of CS simulations and applications
- Adaptation of the algorithms used in the CS simulations to the quasi-opportunistic supercomputing specific constraints
- Business model and accounting for cost-optimisation of grid resources in a broad range of application domains

Partners

University of Ulster	United Kingdom
University of Queensland	Australia
Israel Institute of Technology	Israel
Cranfield University	United Kingdom
Universitat Pompeu Fabra	Spain
Collegium Budapest	Hungary
INRIA	France
University of Amsterdam	Netherlands
Platform Computing	France
AITIA	Hungary
Poznan	Poland

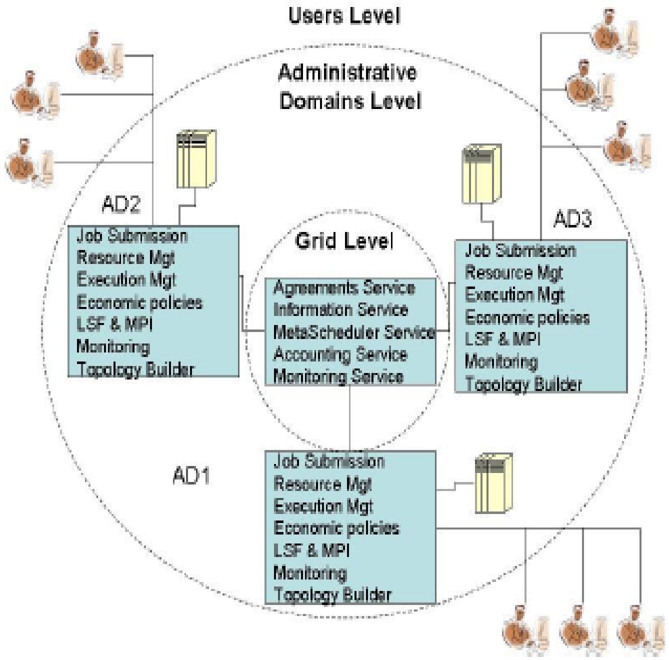
Complex systems partners

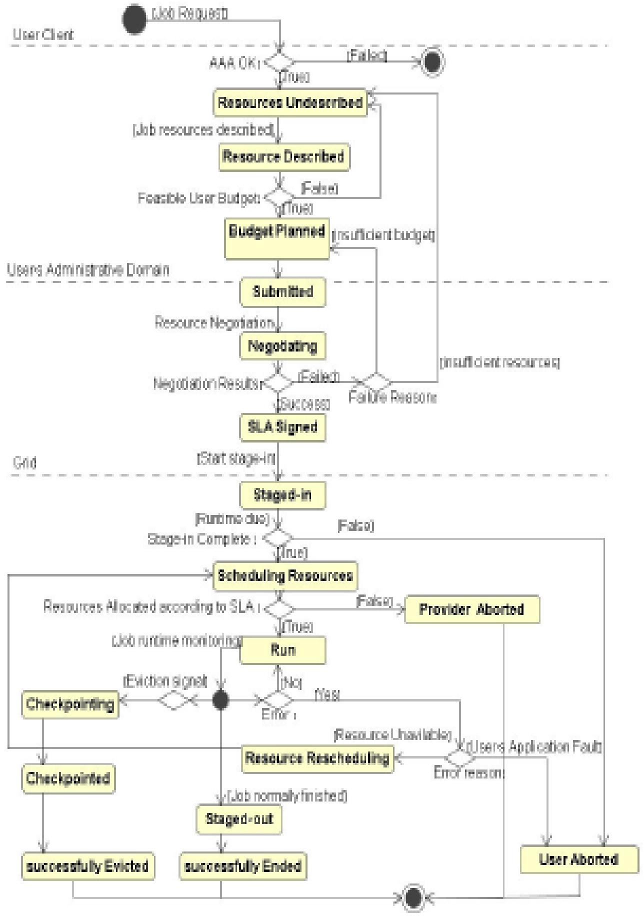
Partners

University of Ulster
University of Queensland
Cranfield University
Universitat Pompeu Fabra
Collegium Budapest
University of Amsterdam
AITIA

Solar system
Economy
Epidemics
Wheather systems
Ecology
Living things

Architecture of the QOsCosGrid system





Grid related Challenges

- Semantic and agent technologies for intelligent resource brokering and management
- Dynamic composition and orchestration of ubiquitous grid services
- Negotiation oriented resource provisioning
- Fault tolerant protocols for grid middleware aware communication
- Data-processing applications for Non-trivial parallelisation of the CS simulation
- Adaptation of the underlying CS algorithms to the quasi-opportunistic supercomputing environment

Complex Systems related Challenges

• Dynamic work-flow planning and dynamic resource scheduling, especially intelligent parameter sweeps

•Handling of data that is produced by the living CS simulation and maintaining its integrity

•Expansion and coupling of existing models

•Highly dynamic allocation of resources

•Translation of generic CS modelling environment into a grid friendly language and operating environment

Complex systems Classification

- Living Simulations: Systems where diferent computational paradigms are needed through a single computational run
  - Stellar Dynamics and Evolution
  - Protein Folding and conformational changes in PPI
  - Spatial aware gene regulatory networks using cellular automata

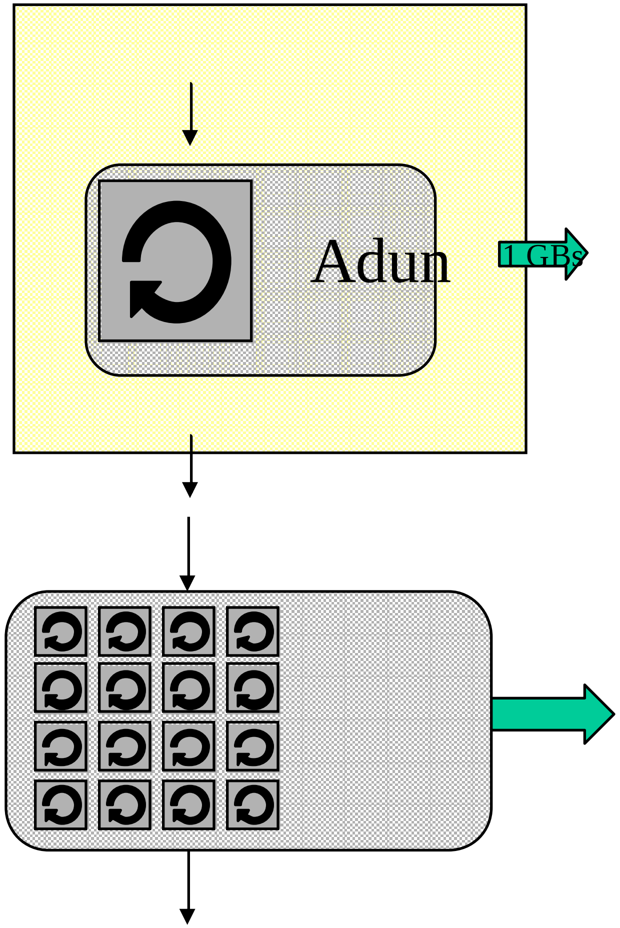
- Evolutionar Modelling: Biological evolution is mimickes to perform global Optimizations in complex functions
  - Topology of gene regulatory networks
  - Parameter Estimation in Systems of Highly Coupled Differential Equations
  - Evolutionary algorithms toolbox

- Coevolutionary Agent Based Modelling
  - Business supply chain evolutionary dynamics
  - Agen Based modelling of ecological evolution
  - Social influence and discrete choice

Characteristics of each use case

		Representation type					Approach			Challenge			
		Networks	Systems/Nonlinear Dynamics	Cellular Automata	Agent-Based	Statistical /Markov	Bottom up	Top down	Multi-scale & Embedded systems	Heterogeneous components	Missing or uncertain data	Changing state space	High Interconnection
Living simulations	Stellar Dynamics and Evolution		X						X	X			X
	Protein Folding and P-P Interactions		X						X				X
	Spatial Model Gene Regulation (CA)			X			X		X				
Evolutionary Computation	Gene Regulation Networks (ODE)	X				X		X		X	X		X
	Highly Coupled ODEs		X			X		X					X
	Genetic/Evolutionary Algorithms T.Box					X		X			X		
Agent Based Modelling	Business Supply Chain Dynamics				X		X			X	X	X	
	Ecological Evolution Models				X		X			X		X	
	Social Influence on Discrete Choices	X			X		X			X	X		

Example: Protein Folding and Protein-Protein interactions



PPI is a challenging problem:

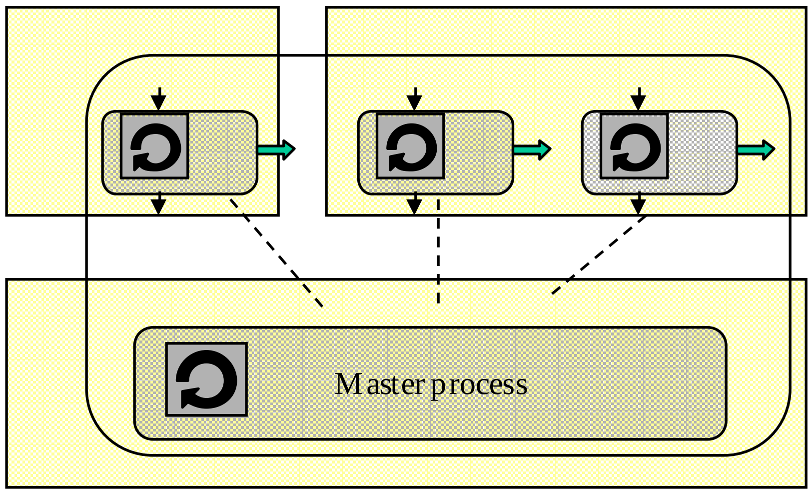
- Non-bonded interaction difficult to calculate
- Better description of the energy surface is needed: The exploration Of the comfomational space of the chain is limited and need to be Explored

➢ Switch between different computationnaly algorithms during The dynamic process

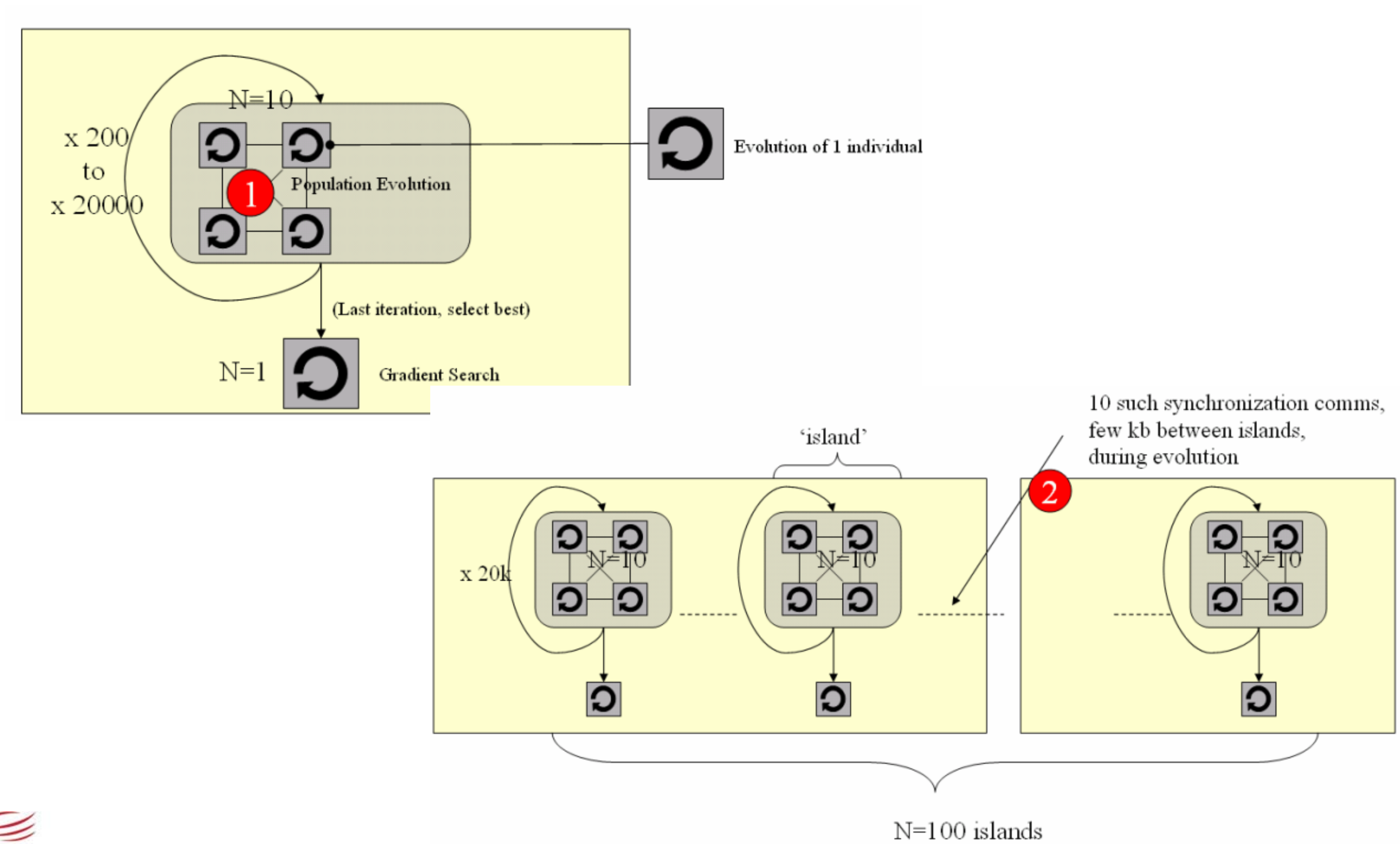
➢ Switch dynamically from one type of simulation to the other

Multiple clusters to obtain a statistical Information of the energy: Non communication

OR



Example: Evolutionary computation



General Framework of Complex systems simulations

A general view of CS is needed, motivated by te urge to speed up Simulations by parallelizing them and distributing the computation Load over a network of processors.

Common features and computing requirements:

Flexible and efficient communication environment
dynamic character of CSS requires the migration of several components from one execution to another
Interaction of the CSS during the simulation progress..
...

Specific traits:

- Parallelization
  - In-Run level: The simulation itself needs parallelization
  - Inter-run level: set of simulations needs to be distributed among processors
  - Inter-inter-run level: isolated sets of simulations needs to be distributed
- Communication: interactions between components are the key of the dynamics of complex systems

Communication patterns of CSS

It should reflect communicational dependencies within CSS and the inner structures of CSS

The communication-based classifiation comprises six templates:

- CSS execute repeatedly, with different initial conditions, to explore the parameter space
- The communication graph is static in time
- The communication graph is updated at a certain time scale
- The communication between components decreases as the spatial distance increases
- One function is enough to describe communication dependencies (cellular automata)
- No communication graph is given. There the relative proportions of the communication cost and the computational time of the single components have to be estimated.

Use cases selected to test the abilities of the grid tools

Major benefit of grid technology ↓	*Major* source of complexity →		
	Number of entities	*Structure of entities*	*Topology*
Inter-Model Parallelisation (Parameter space search, Model fitting, Sensitivity analysis)	Genetic / Evolutionary Algorithms Toolbox	Gene Regulation Networks	Parameter Estimation in Highly Coupled ODEs
Intra-Model Parallelisation (Spatial segregation, Multi-scale modeling, etc.)	Stellar Dynamics and Evolution	Protein Folding and P-P Interactions	Gene Regulation (spatial model)
Inter- and Intra-model Parallelisation (agent-based models)	Ecological Evolution Models	Business Supply Chain Dynamics	Social Influence on Discrete Choices

Summary for the different use-cases of the communication template of the identified parallelization/CSS levels

Use Case Name and Parallelization level		Communication Template	Details
Stellar Dynamics and Evolution (UvA)			
	In-Run	5, (3)	Components: stars Communication: gravity
	Inter-Run	1	Components: layers of the simulation
	“Inter-Inter-Run”	-	
Protein Folding and conformational changes in PPI (PFU)			
	In-Run	5	Components: atoms Communication: P to P interactions
	Inter-Run	1, (1)	Components: heterogeneous simulations
	“Inter-Inter-Run”	-	
Spatial aware gene regulatory networks using cellular automata			
	In-Run	4	Components: genetic regulatory networks
	Inter-Run	-	
	“Inter-Inter-Run”	-	

Conclusion

Based on requirements arising from CS applicatios, we have formulated an initial design of the QosCosGrid: structure maping, sophisticated resource management, management of changing execution topologies, preservaion and restoration of system state, advance reservation features

Next steps...

- Provision of a test bed on which each Use case can be demonstrated in a rudimentary form
- Job description interface
- Parallelization if the CSS

Thank You for your attention