Overview

Today’s emergence of Petascale architecture and the evolution of both research and production grids increase a lot the number of potential resources. However, existing infrastructures and access rules do not allow to fully take advantage of these resources. One key idea of the SPADES project is to propose a non-intrusive but highly dynamic environment able to take advantage of available resources without disturbing their native use.

Goals:
- Support platforms at a very large scale (Petascale environments)

Means:
- Distributed Schedulers based on the Desktop Computing paradigm.

Co-scheduling
The scheduling entities (Scheduler) have to cooperate with both Service Discovery Agents (SDA) and Batch Schedulers (BS).

Objectives:
- Determining the Best mapping of the jobs on resources.
- Designing reactive Scheduling heuristics.

Approaches:
- Adapting strategies coming from Desktop Computing
- Replication strategies
- Cooperation between several batch schedulers
- On-demand job migration

P2P Service Discovery in a large scale and distributed context.

Constraints:
- Platforms heterogeneity
- System dynamicity

Objectives:
- Reliability of infrastructure
- Efficiency of Fault-tolerance mechanisms:
  - Self-stabilisation
  - Replication.

Petascale Runtimes:
Runtimes hitherto proposed do not fit any more with the 100.000 resources

Problems:
- Deployment and launching step duration
- Scaling data structures
- Failure detection broadcasting
- Failure recovery

Issues:
- Unifying resources
- Efficient communication

Means:
- P2P approach
- Overlay networks

Expected results:
- Numerous and frequent failures

Applications

Target applications:
- Cosmic ray simulator
- Coupled climate model.

Test platform:
- IBM Blue Gene
- CRAY XT of PRACE
- DEISA grids

Benefits:
- Increase model resolution
- Reduce model time step
- Make replications

Expected results: Observe low scale Phenomena as hurricanes.

Batch Scheduler Interactions
To allow reactive Petascale architectures, fine interactions between Batch Schedulers (BS) and upper layers is a key issue.

Approaches:
- Using standards API (OGSA-BES, DRMAA) hiding heterogeneity.
- Customizing a BS and executing it in parallel with other BS.