

## Introduzione ai Sistemi Grid

## Generalità

- Un sistema Grid permette all'utente di richiedere l'esecuzione di un servizio computazionale, che verrà offerto in modo trasparente da una o più risorse distribuite
- Esistono diverse definizioni, ciascuna che mette in evidenza aspetti diversi

## Definizione

- Per gli scopi del corso, una Grid è un insieme di unità computazionali **indipendenti, distribuite e cooperanti**, in grado di raggiungere uno specifico obiettivo computazionale all'interno di **organizzazioni virtuali dinamiche e multi-istituzionali (\*)**

–(\*) I. Foster, C. Kesselman, and S. Tuecke, "The anatomy of the Grid: Enabling scalable virtual organizations", *Int. J. High Perform. Comput. Appl.*, vol. 15, n° 3, pp. 200-222,

## OGSA

- OGSA = Open Grid Service Architecture
- Definisce un insieme di specifiche che documentano i requisiti delle componenti hw/sw di una Grid

## Funzionalità secondo OGSA

- OGSA identifica 6 principali funzionalità di una Grid:
  - **Execution Management Services (EMS)**;
  - Data Services;
  - Resource Management Services;
  - Security Services;
  - Self-Management Services;
  - Information Services
- Per gli scopi del corso è particolarmente interessante EMS

## EMS secondo OGSA

- EMS address the job management and execution capability of a grid system and it is concerned with the research of candidate locations for
  - execution
  - preparation for execution
  - initiating and managing the execution of jobs until the end

## Tipi di Job

- Sequenziali / Batch
- Paralleli
- CheckPoint
- Interattivi
- DAG (job con interdipendenze, che possono essere modellati da un Grafo Diretto Aciclico)
- Partizionabili

## Stati di un Job (1)

- **SUBMITTED:**
  - the user has submitted the job via UI
- **WAITING**
  - the WMS has received the job
- **READY**
  - A CE, which matches job requirements, has been selected, and the job is transferred to the JSS
- **SCHEDULED**
  - the JSS has sent the job to the CE



## Requisiti (1)

- **Req.1** A job can be submitted to the grid; if there is no job, the system remains in a inactivity state.
- **Req.2** The grid middleware checks the matchmaking between resources and job constraints before the execution; if there are problems (e.g. lack of memory or of devices, or slow CPU speed, etc.) the system reject the job.
- **Req.3** After accepting the job the system runs it.

## Requisiti (2)

- **Req.4** If there are no failures the job is completed; on the other case the job fails.
- **Req.5** A user can cancel a job every time.
- **Req.6** At the end of the computation (job completed or aborted for every reason) every resource is released.
- **Req.7** If software or hardware errors occur the job is aborted.
- **Req.8** At the end of every computation the result is communicated to the end user.