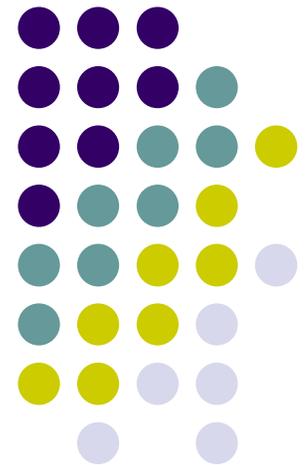


# Computação Paralela

**Gestão de *clusters* ligados em grelha (*grid*)**

**João Luís Ferreira Sobral  
Departamento de Informática  
Universidade do Minho**

**Novembro 2006**

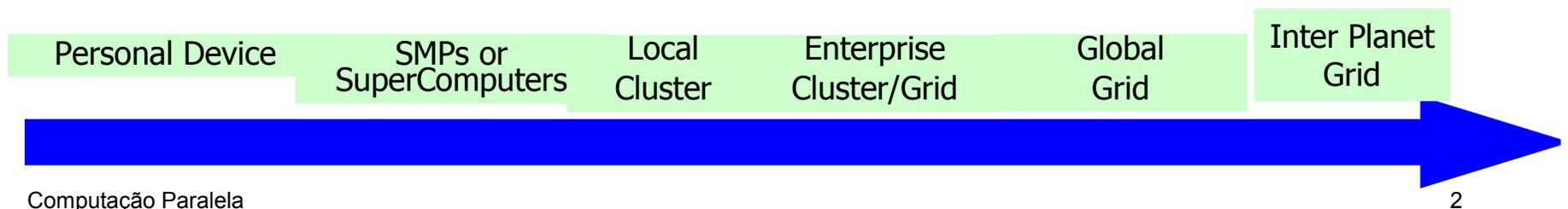




# Gestão de *clusters* em grelha

## Evolução da GRID

- Antigamente os supercomputadores eram recursos dedicados, apenas utilizados por um número reduzido de programadores
- A progressão da Internet levou uma mais fácil utilização de recursos computacionais remotos
- A partilha dos supercomputadores e de clusters de máquinas origina uma utilização mais eficiente dos recursos (em termos de grau de utilização das máquinas)
- Os recursos partilhados e heterogéneos exigem ferramentas que permitam efectuar a gestão das tarefas submetidas pelos vários utilizadores:
  - **Determinar em que instante um utilizador pode aceder ao recurso**
  - **Escolher as máquinas mais adequadas para a execução da tarefa requerida.**
  - **Efectuar a autenticação e autorização dos utilizadores em vários domínios organizacionais**





# Gestão de *clusters* em grelha

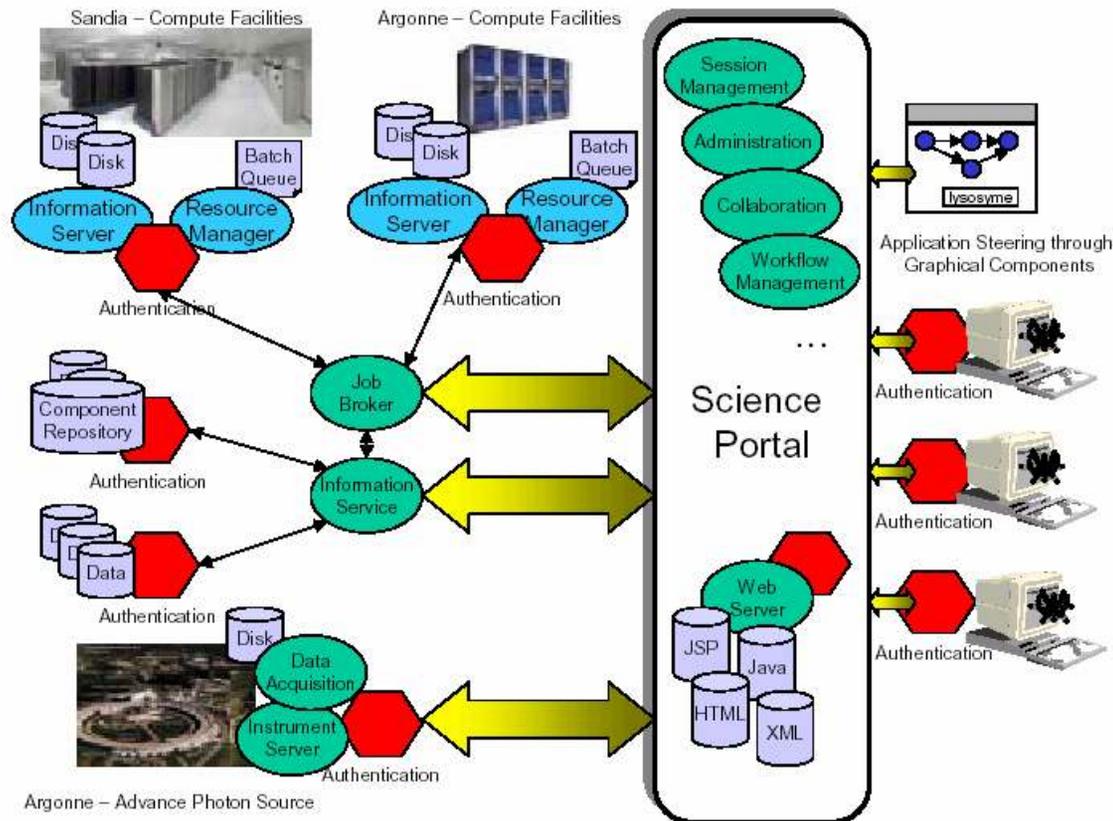
## Tipos de GRIDs

- GRID = agregação de recursos distribuídos e autónomos (e.g., distribuídos por várias organizações)
- Computacionais – partilha de capacidade de cálculo agregando clusters, supercomputadores ou estações de trabalho (SETI@home, LHC Grid)
- Dados – partilha de dados, distribuição de dados (base de dados do gnoma humano, sistemas peer-to-peer, gmail?), selecção/agregação de múltiplas fontes de dados
- eScience – partilha de instrumentos científicos (telescópios, etc..)
- Aplicaçionais – servidores de aplicações, licenças (netSolve, Google Map?)
- Interacção – eLearning, access Grid, Visualização remota
- Conhecimento
- Desafios:
  - **Segurança, acesso uniforme, descoberta de recursos, alocação e escalonamento de recursos, localização de dados, construção de aplicações, gestão da rede**



# Gestão de *clusters* em grelha

## Exemplo de uma GRID





# Gestão de *clusters*

## TORQUE (Tera-scale Open-source Resource and QUEue manager) (<http://www.clusterresources.com/products/torque>)

- Gere um conjunto de máquinas distribuídas (clusters), através da submissão de tarefas que são executadas quando os recursos requeridos estiverem disponíveis
- Comandos base:
  - **qsub script** – submeter um *job*
    - Os resultados da execução são gravados na directoria com o nome *script.odjob* para stdout e *script.didjob* para stderr
    - -l nodes=NN:ppn=PP - reserva NN nodos, cada nodo com PP processadores
    - -l walltime=00:10:00 - reserva do nodos durante 10 minutos
  - **pbsnodes -a | more** – ver o estado/informação sobre os nodos
  - **showq** – ver o estado dos jobs submetidos