Open Grid Middleware Panel

Industrial Views on Existing and Future Grid Middlewares

Jean-Pierre Prost
Senior Technical Staff Member, Grid Computing
IBM Montpellier, France

© 2005 IBM Corporation

Current Situation in Grid Computing

- **Most commercial Grids today are application Grids or enterprise Grids**
- **Large scale Grids are mainly academic**
- **Some of the inhibitors for larger scale deployment of commercial Grids:**
  - Limited QoS (performance, security, availability) guarantees
    - No real deadline scheduling available
    - End to end security is still an issue
    - Resources may come and go at any time
  - Governance issues
  - No standards yet for QoS goal expression and negotiation
  - Standards in web/grid services are still in the making

© 2005 IBM Corporation

Limitations of Existing Grid Middleware

- Lack of interoperability between scheduling offerings
- No real meta-scheduler or peer scheduler capability
- Quality of Service is most of time best effort
- No standardized SLA based negotiation between requestor and provider
- Limited support of network resource management (actual bandwidth and latency measurement, bandwidth reservation, application network requirement characterization)
- Limited resource usage metering, accounting, and billing capability
- Limited integration between scheduling, workload management, and provisioning
- Issues related to commercial licensed software management
- Lack of global resource namespace
- Limited data aware scheduling capabilities
- Data requirements should be weighed in scheduling decisions at the same level as compute resource requirements
- Limited activity in job flow scheduling
- Limited fault tolerance
- Lack of a clearly defined grid programming model
- Limited application development and runtime tooling (no grid application construction wizard, limited application debugging tools)

© 2005 IBM Corporation

Wishlist for Future Grid Middleware

- **Full interoperability between scheduling offerings**
- **Meta- to sub- scheduler and peer to peer scheduler standardized interactions**
- **Support for QoS based scheduling (eg real deadline scheduling, business value driven scheduling)**
- **Standard based SLA negotiation between requestor and provider, with enforcement and auditing capabilities**
- **Dynamic configurability of network infrastructure to satisfy application expressed network requirements**
- **Standards for resource usage metering and end-to-end accounting**
- **Simple and flexible pricing models for capacity on demand services, software as a service offerings, licensed software grid deployment**
- **Full integration between scheduling, workload management, and provisioning allowing for optimized utilization of resources in compliance with established service level agreements**
- **Global resource namespace allowing for transparent activity/service instance migration and easy referenceability of resources**
- **Good data aware scheduling capabilities, allowing for resource allocation optimization with data access considerations**
- **Standard job flow scheduling expression and support for optimized flow execution**
- **Complete application development, deploiement, debugging, and performance analysis environment, based on a common grid programming model**

© 2005 IBM Corporation

Importance of Standards and Open Source Reference Implementations

- Standards are key for interoperability of implementations
- Open source reference implementations are key for standard validation and early adoption
- Proprietary implementations will complement open source implementations with higher quality of implementation and additional customer support capabilities
- What matters first is the interoperability between implementations (open source and proprietary) through their compliance to open published interfaces and resource models
- What matters next is robustness and performance of the implementations

© 2005 IBM Corporation