NATIONAL FUSION COLLABORATORY

The NFC project defines a virtual organization devoted to fusion research and provides the "codes" developed by this community to the end users (researchers). Earlier, this "code" software was installed in the end user 's machine. This became a complex and unmanageable process of software management, distribution, versioning, and upgrade. Due to this change management and configuration problem the fusion community decided to adopt the ASP model, known as "network services model," where the "code" is maintained by the service provider and made accessible to the remote clients . This eliminates the burden on the end user but adds some QoS requirements on the service provider, including executing the "code" as efficiently as possible, executing within a certain time frame, and producing the results with accuracy. As you can imagine, this is the best-case usage model for a computational grid. Now, we can drill down into the usage scenarios of this grid and derive the functional requirements on Grid Computing architecture.

Customers (Actors)

Scientists. They are customers of the fusion code provided by the fusion service provider. Some of the customer requirements are:

The ability to run the "code" in remote resources on the condition of end-to-end quality of service with a guarantee of time-bound execution.

Availability of the resource (code execution) in the computational grid.

A policy-based management of resources; including who can run the code, how many hardware resources are available, etc.

Ability to use community services by getting accredited with the community rather than an individual service provider. This is a form of "dynamic account" creation and usage.

Scenarios

A remote client (scientist at an NFC facility) can run code on a remote site within a time frame. The service provider downloads the necessary data and executes a workflow script.

A monitoring agent starts and watches the submitted job for service-level agreement (SLA) validation. This helps the service provider to provision more resources or recover from failure conditions, etc.

Integrate with external applications and resources for data and/or code execution and flexible delegation of rights.

Functional Requirements on OGSA

After a thorough and careful examination of the static and dynamic behavior present in this use case, the following functional requirements of the grid architecture can be identified:

Discovery of available resources

Workflow management for job distribution across resources

Scheduling of service tasks

Enabling the facilities for disaster recovery in case of outages

Provisioning of resources based on the need

Resource brokering services to better utilize and use the resources and to achieve the level of quality requirements

Load balancing to manage workloads

Network transport management

Integration with legacy applications and their management

Handling application and network-level firewalls

Service-level agreement and agreement-based interaction

Providing end-to-end security and security authorization and use policies

Next we discuss an online media and entertainment project with some highly interactive content and data sharing among participants . This is an on-demand media and entertainment system, which can be a classic representation of the next generation of on-demand applications.