# Traditional HPC User: Special Quality-of-Service Computing

## Summary

Special quality of service (QOS) high performance computing and data jobs refer to a class of applications that require use of a specific HPC resource at a specific time, within a specific timeframe range, on a periodic recurring basis or on an urgent basis. The QOS types include high priority, advanced reservations and urgent computing. High priority QOS has no definite start time and should start as soon as possible. Advanced Reservation QOS has a definite start time and a reservation on the required resources and may include a periodic reservation. Urgent Computing QOS has an urgent requirement and needs specific resources as soon as possible (essentially immediately when the job(s) are ready). These applications need high priority or guaranteed access to the required resources or services within a specific timeframe due to deadlines, due to time dependent collection and/or processing requirements of data from specialized instruments, for time dependent simulations such as climate modeling or to satisfy other high priority time dependent requirements. Use of manual or automated advanced reservations is usually employed on the end resources to provide the capability for QOS jobs. These reservations are usually manual, but automated methods are wanted for some applications. Standing reservations are used for the periodic recurring QOS jobs. Application types for QOS jobs include HPC capability jobs, HPC mid range jobs, and HPC data intensive jobs.

## Customers

Beneficiaries include users with deadlines for work (perhaps for a paper, a journal, or a conference deadline) or with periodic model/simulation frequencies based on periodic data collection or periodic deadlines. QOS users come from a variety of different disciplines, including, but not limited to, physics, aerospace and mechanical engineering, meteorology, and climatology. They can be expert users either who develop their own large scale applications, have extensive experience modifying community applications in an HPC environment, are associated and have experience with a portal or gateway, or are familiar with direct batch job submission.

## Scenarios and applications

On Sept. 7, 2008, six days before Hurricane Ike, the third most destructive hurricane in U.S. history, crashed into the Texas coast, NOAA and NSF urgently contacted TACC to help answer several vital questions using computational analysis [2]. The multi-tiered effort used global models with twice the resolution of the best operational simulations, regional models six times as high-resolution as those now used, and added, for the first time, Doppler radar data streamed directly from NOAA’s planes to TACC Ranger. A partition of Ranger was cleared and dedicated to running ensembles of thirty 1.5 km resolution WRF model jobs to predict landfall. Also in dedicated mode on Ranger, the ADCIRC model was used to predict the storm surge, while another server at TACC was engaged in evacuation logistics coordination management.

## Involved resources and production Grid infrastructure

High priority or dedicated computational, data and archival resources available at a specific site.

The following functional requirements exist for this use case.

**Scheduling**: For computational resources, access to a high priority queue or high scheduling priority in the resource management (batch) system.

**Advanced Reservations**: Ability either manually or in an automated way to request and obtain an advanced reservation for computational resources. The reservation can either be a single advanced reservation or a recurring periodic reservation. For data or archival resources, this is usually a manually made advanced reservation of adequate storage space.

**Urgent Computing**: These high priority on-demand large-scale computations can't wait endlessly in a job queue for supercomputer resources to become available. These jobs must have access to resources on an as soon as possible or “now” basis. Service providers need to have the capability to schedule these jobs as a high priority or preempt (or checkpoint/restart if available) existing jobs to run these “now”.

## Security considerations

This use case typically requires normal user account access and local and remote batch queue access.

## Performance considerations

This use case requires high priority or reserved access to resources.

## Usecase situation analysis and PGI expectations

This use case relies heavily on the ability to make reservations of resources (be they compute, or data), and at the proper time, execute jobs on those resources. Further, for the urgent computing aspect of this use case, the ability to manage jobs already running (either to kill, suspend, checkpoint, etc.) is needed.

At the moment, the existing BES, JSDL, and JSDL extension specifications can be used to support the execution of, and management of jobs. Since most of the functionality for execution in this scenario is implementation and not interface (i.e., the interface to BES doesn’t have to change to take into account urgent computing needs), few if any extensions are necessary. However, extensions would be needed to support the reservation aspect of this use case. For this, a reservation port type (interface) would be needed that could then be folded in to an appropriate JSDL extension to support the reservation. The back end BES implementation would ofcourse need to be able to handle this JSDL extension and the reservation itself (though its not necessarily the case that the BES implementation has to manage the reservation, merely be aware of and work with it).

From a security perspective, existing OGSA security profiles and specifications cover most of the use case.

## References

1. From Call for Papers, Special Issue on Scientific Workflows, International Journal of Business Process Integration and Management, http://www.cs.wayne.edu/~shiyong/swf/ijbpim09.html, October 2009.
2. http://www.tacc.utexas.edu/RangerImpact/pdf/Science\_Center\_of\_Storm.pdf

## Author information

Andrew Grimshaw (grimshaw at virginia.edu), Genesis II Project, University of Virginia