



# GRAM Overview

Steve Tuecke

November 2004



## What is GRAM?

- Protocol and software to for locate, submit, monitoring, and cancel jobs on computing resources.
- Addresses a range of jobs where reliable operation, stateful monitoring, credential management, and file staging are important.
- *Not* a job scheduler, but rather a set of services and clients for communicating with a range of different local job schedulers and meta-schedulers using a common protocol.

# GRAM Implementations

- **Pre-WS GRAM**
  - ◆ First implementation of GRAM
  - ◆ GT2 - Globus-specific protocol
  - ◆ Gatekeeper/jobmanager services
- **WS GRAM**
  - ◆ Web Service based implementations of GRAM
  - ◆ GT3 OGSI based implementation
  - ◆ GT4 WSRF based implementation

## GT3.x GRAM

- GT3.0 introduced a WS-based GRAM with performance inferior to that of pre-WS GRAM
- GT3.2 WS-based GRAM has performance comparable to that of pre-WS GRAM
  - ◆ Tuning and performance testing
  - ◆ Improvements in GT core
  - ◆ CERN evaluation effort a big help (thanks!!)
- Job submission rates: 4/sec → 16/sec
- Total jobs: 100 → ~1000
- Clearly not good enough yet!!



## GT4.0 GRAM Requirements

- Significant improvements in performance, scalability and stability
- Reduce load on head node & support larger numbers of jobs & job bursts
  - ◆ Enable a single head node to deal with many thousands of jobs
- Decrease client latency, server throughput
  - ◆ Hundreds of job submissions per minute
- Better resiliency
  - ◆ GRAM should never “fail” under any conditions



the globus alliance

[www.globus.org](http://www.globus.org)

## GT4.0: GRAM Redesign

- Re-architect to reduce overheads
  - ◆ Reduce memory footprint, optimize interfaces
- Move optional features out of critical path
  - ◆ E.g., code for streaming output
- Control cost of security
  - ◆ E.g., pay for delegation only when needed
  - ◆ Reduce from 6 to 1 roundtrips per job!!
- Continued tuning to improve performance
  - ◆ Better response to server load
  - ◆ Improved client implementations
  - ◆ Better server management and concurrency



## GT4.0 WS GRAM Major Changes

- Single hosting environment
  - ◆ No more dynamic server creation
- Per-user, not per-job credentials
  - ◆ Accelerate submission of multiple jobs
- Flexible credential management
  - ◆ Introduce credential cache service
  - ◆ Delegation used only when required
- Replace GASS file management with GridFTP and RFT
  - ◆ Removes overhead when not used
- Replace RIPS with optimized state tracking



## New WS GRAM GT 4.0 Features

- Managed Multi Job Service (MMJS)
  - ◆ Submits jobs to MEJS, given a multi-job RSL
  - ◆ Aggregates job states for all MEJS jobs
- Ability for jobs to self-organize (rendezvous)
  - ◆ Important for MPICH-G4
- Exit code of a job available to clients
- *User-mode* GridFTP for staging
  - ◆ Eliminate the complexity and inefficiencies of GASS cache, file streaming, URL staging



## GT 4.0 Release Timeline

- GT 3.9.3 (Alpha Quality)
  - ◆ Available - Oct 30, 2004
    - <http://www-unix.globus.org/toolkit/docs/development/3.9.3>
- GT 3.9.4 (Beta Quality)
  - ◆ Expected - Dec 2004
- GT 4.0 (Stable Release)
  - ◆ Expected - Q1 2004



## GRAM: Future Work

- Continue to improve performance, throughput, scalability, robustness
- Ability to target specific cluster nodes
  - ◆ Nodes connected to display devices
  - ◆ Nodes with high speed networking
- Job Lifetime Management (if we decide that this will not be included in the 4.0 release)
  - ◆ Allow admins to set policy on job manager lifetimes
- Service load management?
  - ◆ Allow admins to throttle impact of service on service node?
- Advanced reservation?