Globus Toolkit 4 Execution Management

Alexandra Jimborean
International School of Informatics
Hagenberg, 2009

Agenda of the day

- Introduction to Globus Toolkit and GRAM
- Zoom In – WS GRAM
  - Usage Guide
  - Architecture
  - Overview – Administrating Servers
- Summary
- References

Globus Toolkit

- GT 4 - open source toolkit for building computing grids

- Enables sharing of
  - computing power
  - databases
  - other tools,

  securely, online across corporate, institutional, and geographic boundaries without sacrificing local autonomy

Execution Management

- initiation, monitoring, management, scheduling, and/or coordination of remote computations
  - Handles placement, provisioning and lifetime management of jobs

- GT4 supports the Grid Resource Allocation and Management (GRAM) interface as a basic mechanism for these purposes

GRAM (I)

- GRAM is a unifying remote interface to Resource Managers

- GRAM addresses a range of jobs where
  - reliable operation
  - asynchronous monitoring and control
  - credential management
  - file staging

  are important

GRAM (II)

- GRAM is not meant to serve as a "remote procedure call" (RPC) interface for applications not requiring many of the previously mentioned features

- its interface model and implementation may be too costly for such uses

Grid Job Management Goals

Provide a service to securely:

- Create an environment for a job
- Stage files to/from environment
- Cause execution of job process(es)
  - Via various local resource managers
- Monitor execution
- Signal important state changes to client
- Enable client access to output files
  - Streaming access during execution

WS GRAM Insight

File Management

- data and file staging via RFT and GridFTP
- a failed file staging operation can be retried by the GRAM file staging service—the reliable file transfer (RFT)
- RFT's retry policy – default or overridden
- concurrent transfers - limit number, no overload
- load balanced service work

Job submission

- supports single and multiple job submission
- able to check for and possibly resubmit jobs [transient communication]
- reliable job submission, without risk of running more than one copy of the job
- mechanism for cancellation of jobs

Job Submission Options

- Optional file staging
  - Transfer files “in” before job execution
  - Transfer files “out” after job execution
- Optional file streaming
  - Monitor files during job execution
- Optional credential delegation
  - Create, refresh, and terminate delegations
  - For use by job process
  - For use by GRAM to do optional file staging

Example – Submitting Jobs with options

- $ globusrun-ws -submit -c /bin/hostname
  - WS oriented job submission command
  - $ globusrun-ws -submit -s -c /bin/hostname
  - Streaming the output
  - $ globusrun-ws -submit -s -so out.txt -c /bin/hostname
  - Output go to file
  - $ globusrun-ws -submit -s -F ng2.vpac.org /bin/hostname
  - Send job to remote host
  - $ globusrun-ws -submit -s -F ng2.vpac.org -Ft PBS –c /bin/hostname
  - Send to the batch system

Common / Useful options

- globusrun-ws -J
  - Perform delegation as necessary for job
- globusrun-ws -S
  - Perform delegation as necessary for job's file staging
- globusrun-ws -s
  - Stream stdout/err during job execution to the terminal
- globusrun-ws -self
  - Useful for testing, when you have started the service using your credentials instead of host credentials

Submitting jobs in C

- 1. Loading the job description
- 2. Setting the security attributes
- 3. Creating the factory client handle
- 4. Querying for factory resource properties
- 5. Creating the notification consumer
- 6. Creating the job resource
- 7. Subscribing for job state notifications
- 8. Releasing any state holds (if necessary)
- 9. Destroying resources
- 10. Building a client

Submitting jobs in Java

- 1. Class imports
- 2. Loading the job description
- 3. Creating the factory service stub
- 4. Loading a proxy from a file
- 5. Setting stub security parameters
- 6. Querying for factory resource properties
- 7. Delegating credentials (if needed)
- 8. Creating the job resource
- 9. Creating the job service stub
- 10. Subscribing for job state notifications
- 11. Releasing any state holds (if necessary)
- 12. Destroying subscription resources

States of a Job

- submitted→ **pending**: resources not yet allocated
- resources → **active**: application running
- terminate job → **failed state**
  - explicit termination
  - an error in the format of the request
  - a failure in the resource management system
  - a denial of access to the resource
- all processes in the job finished → **done state**: deallocate resources

Example

Submit program echo with argument hello to default local host.

- **% globusrun-ws -submit -c /bin/echo hello**
- **Submitting job...Done.**
- **Job ID: uuid:d23a7be0-f87c-11d9-a53b-0011115aae1f**
- **Termination time: 07/20/2005 17:44 GMT**
- **Current job state: Active**
- **Current job state: CleanUp**
- **Current job state: Done**
- **Destroying job...Done.**

Command Line Tools

- globusrun-ws –submit –F *JobFactoryURL* –Ft *FactoryType* –c *command*
  - Interactive job submission
- globusrun-ws –submit –batch –F *JobFactoryURL* –Ft *FactoryType* –o *EPRfile* –c *command*
  - Batch job submission
- globusrun-ws –status –j *EPRfile*
  - Checking job status
- globusrun-ws –kill –j *EPRfile*
  - Kill a job

Security features (I)

- **Privilege limiting model**: *Service with sudo privileges*

- In a typical deployment, the GRAM server must be able to start jobs submitted by remote users under different user ids, and thus must be able to execute some code as "root."

- It is generally viewed as preferable to limit the amount of such "privileged" code. The GRAM service does not itself require privileges. Instead, it uses "sudo" to invoke operations for which privileges are required.

Security features (II)

- **Authentication**: *Message-level security*

- *A* client can authenticate with a GRAM service using a variety of protocols:
  - the standard message-level WS-Security
  - channel- level WS-SecureConversation
  - the choice of protocol is supported by a particular GRAM4 deployment specified in the service configuration

Security features (III)

- **Credential delegation and Credential Refresh**
- *A* job submitted to a GRAM service may require a delegated credential if it is to stage files or perform other remote operations for which authentication is required
- Clients can delegate a credential only when required and can share a delegated credential among multiple jobs
- If credentials expire, the same service can provide credentials refresh

GRAM Server and Delegation

- delegation of proxy credentials
- computation monitoring and management

in an integrated manner

- share credentials among jobs

- **Delegation** (specific to grid computing)
  - Process of giving authority to another identity (usually a computer/process) to act on your behalf

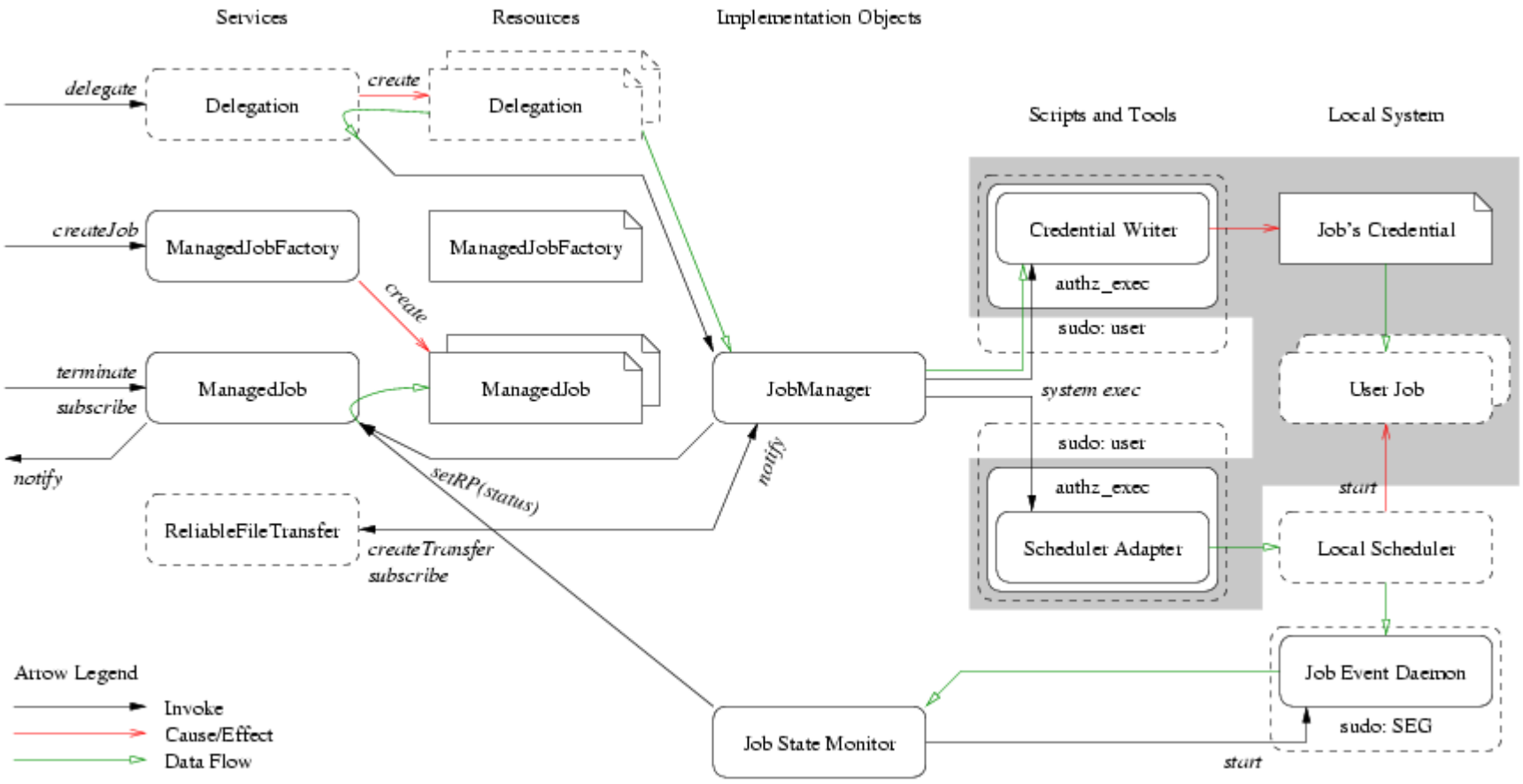
Command Line Tools

- globusrun-ws
  - Submit and monitor gram jobs
- grid-proxy-init
  - Creates client-side user proxy
- wsrf-query
  - Query a services resource properties
- globus-url-copy
  - Transfer files to remote hosts
- globus-credential-delegate
- globus-credential-refresh
  - Credential management to remote hosts

Components of GT used by GRAM

- ReliableFileTransfer
  - support for retry, restart, and fine-grained control of file staging before and after job computations
- GridFTP
  - manage transfers of data between remote storage elements and the compute element file systems
  - efficiently and reliably check the status of files and incrementally fetch new content (“streaming”)
- Delegation
  - delegate credentials into the correct hosting environment

Architecture



Performance

- GRAM should add little to no overhead compared to an underlying batch system
  - Throughput
    - Number of jobs (/bin/date) GRAM can process per minute
    - 100
  - Max concurrency
    - Total jobs a GRAM service can manage at one time without failure
    - 32,000
  - Job burst
    - Many simultaneous job submissions
  - Service uptime
  - Job recovery

Overview – Administrating Servers

- Consult quick guide first for basic GT setup
  - Setting up first machine
  - Setting up second machine
  - Setting up a compute cluster - PBS
  - www.globus.org/toolkit/docs/4.0/admin/docbook/quickstart.html
- Then consult GRAM admin guide for additional details
  - www.globus.org/toolkit/docs/4.0/admin/docbook/ch11.html

Summary

- Execution Management Overview
- WS-GRAM Model
  - File management
  - Job submission
    - Command Line Examples
  - Security Delegates
  - Software Architecture
- Administrating Servers – main steps

References

- *GT 4 WS_GRAM: User Guide, Developer Guide, Admin Guide*
- *GT4 GRAM: A Functionality and Performance Study*- Martin Feller, Ian Foster, Stuart Martin, TERAGRID 2007 CONFERENCE, MADISON
- *Virtual Workspaces: Achieving Quality of Service and Quality of Life in the Grid* - K. Keahey, I. Foster, T. Freeman, and X. Zhang, *Scientific Programming Journal*, 2006