The GIRD Grid Job Management Framework
Middleware-Independent Grid Job Management

NSC’09 2009-10-13

Erik Elmroth and P-O Östberg
Department of Computing Science, Umeå University
There exists a need for generic Grid infrastructure components that provide
- multiple levels of job control & fault tolerance
- middleware independence
- flexible & customizable architectures
- flexibility in deployment and administration

Understanding structure of, and best practices for, decentralized Grid infrastructure components

Evaluation and utilization of Grid standards

Service composition-based development methodology
- small, standards-compliant components
- component dependencies well-defined and minimal
- short development cycles & software reuse
A Middleware-Independent Grid Job Management Architecture

- Composable Service-Oriented Architecture for middleware-independent Grid job management
- Provides multiple types of Grid job submission, control, and management functionality
- Decouples Grid applications from infrastructure
- Builds on Web Services organized in hierarchical layers of functionality
- Foundation layer abstracts middlewares and infrastructure
- Higher layer services aggregate lower level services to provide more advanced functionality sets
Middleware-Independent Grid Job Management

Architecture

Application Layer
- E.g. Client API, End User Tools, Application Portals

Advanced Job Submission Layer
- E.g. Workflow Execution Service

Reliable Job Submission Layer
- Task Group Management Service

Brokered Job Submission Layer
- Task Management Service

Middleware Abstraction Layer
- Brokering & Submission Service
- Job Control Service
- Resource Selection Service

Grid Middleware Layer
- Middleware(s)
Component Structure

- Local call optimization mechanisms allow components transparent distribution and highly efficient communication when codeployed.
- Transparency in service composition and invocation optimizations allows services to be deployed in a number of constellations:
  - utilizing the framework as a middleware-independent Grid job submission interface
  - employing a personal job management tool
  - deploying multiple instances of the framework to provide alternative job submission behaviors, policies, or partitioning / load balancing between Grids.
Middleware-Independent Grid Job Management

Service Structure

Service Container Process  |  Network  |  Client Process

- Service Resource
- Service Instance
- Service Back-End
- Service Resource Home

- Web Service Call
- SOAP

- Immutable Wrapper

- Local Call
- Immutable Wrapper

- Service Client
- Application

- Local Call Client
- Service Client Factory

The GIRD Grid Job Management Framework

Middleware-Independent Grid Job Management

Erik Elmroth and P-O Östberg

Department of Computing Science, Umeå University
Task Management Service (TMS)
The Grid Job Management Framework (GJMF)

- Built on Globus Toolkit 4
- Integrated with the Globus and ARC middlewares (experimental support for GT2, LCG/gLite, Condor)
- Customization points provide integration support for additional middlewares
- Components implemented as GT4 WSRF web services
- Current version a third generation software
- Production quality version developed at HPC2N with support from SNIC
The Grid Job Management Framework (GJMF)

Features

- Composable set of tools
  - each tool add value and (most) can be used individually
- Generic Grid middleware compatibility
  - customization points for new middlewares
    (concentrated to two components)
- (Transparent) local calls are utilized when possible
  - service interface bypassed by in-process local calls
- Support classes simplifies application development
  - Java client API abstract service interface functionality
  - command-line tools and reference clients provided
- User-level isolation of service capabilities
  - unique service instantiated for each user
Web Service Performance Issues

- Invocation latencies
  - message serialization and parsing is slow
  - (Java) class loading overhead impact invocations severely
  - security overhead will be added as invocation overhead
  - mediated by batch invocation modes (fewer calls)
  - mediated by service invocation optimizations (local calls)

- Memory loads
  - text-based formats are storage-wise not very efficient
  - XML serialization requires large amounts of memory
  - security contexts adds to message memory load
  - mediated by batch invocation modes (fewer messages)
  - mediated by invocation optimizations (less memory)
GJMF Overhead

- GJMF Job execution overhead consists of 3 parts
  - job submission overhead (GJMF contribution)
  - job processing overhead (GJMF contribution)
  - job execution time (includes middleware overhead)

- GJMF overhead contributions mediated
  - GJMF overhead masked by parallelization
  - GJMF mechanisms mediate job submission latencies
  - GJMF contribution typically small
The Grid Job Management Framework (GJMF)

Sequential Invocation Overhead

Infinite resources

Limited resources
The Grid Job Management Framework (GJMF)

Batch Invocation Overhead

Infinite resources

Limited resources
Job Submission Throughput

Ideal jobs, codeployed services, sequential & batch invocations
The Grid Job Management Framework (GJMF)

Job Submission Throughput

Ideal jobs, standalone services, sequential & batch invocations

The GIRD Grid Job Management Framework
Erik Elmroth and P-O Östberg

Department of Computing Science, Umeå University
The Grid Job Management Framework (GJMF)

Job Processing Throughput

Realistic jobs, codeployed services, sequential & batch invocations

The GIRD Grid Job Management Framework
Erik Elmroth and P-O Östberg
Department of Computing Science, Umeå University
The Grid Job Management Framework (GJMF)

Job Processing Throughput

Realistic jobs, standalone services, sequential & batch invocations
Invocation Throughput Comparison

LAS & JTS Invocation Throughput

The GRID Job Management Framework (GJMF)

Conclusion
Service Invocation Optimization Types

The Grid Job Management Framework (GJMF)

Service Container Process | Network | Client Process

SOAP Engine | SOAP Message | SOAP Engine

Web Service Instance

Axis Local Call | SOAP Message | Axis Call Utility

Web Service Instance

GT4 Local Invocation | Stub Type | GT4 Invocation Utility

Web Service Back-End

GJMF Local Call | Immutable Wrapper | Web Service Client
The Grid Job Management Framework (GJMF)

Service Invocation Optimization Benchmarks

Sequential & parallel invocations

![Graph showing service invocation optimization benchmarks](graph.png)

Conclusion

The GIRD Grid Job Management Framework

Erik Elmroth and P-O Östberg

Department of Computing Science, Umeå University
The GJMF provides convenient middleware-independent access to Grid resource capabilities

Client API abstracts web service development complexity

Builds on Grid standards, e.g., JSDL, OGSA BES & RSS

Transparent local calls enables multiple usage models

The GJMF local calls exhibit performance comparable to alternative service optimization mechanisms

Overhead contributions sufficiently small to motivate use of the framework, and can be mediated in multiple ways
Conclusion

Further Reading

- www.gird.se