



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



Middleware-Independent Grid Job Management

- ▶ 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



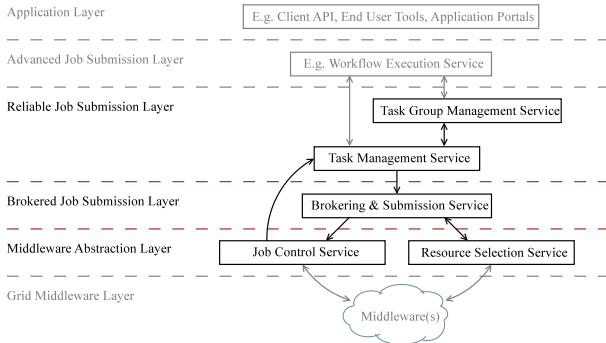


A Middleware-Independent Grid Job Management Architecture

- ▶ 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



Architecture



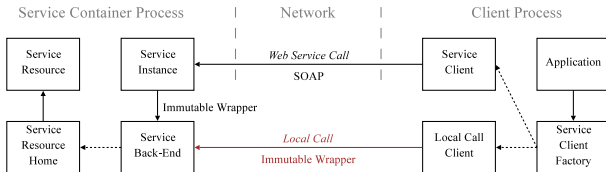


Component Structure

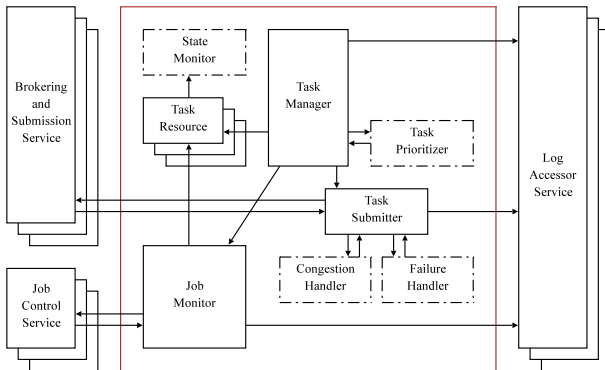
- ▶ Local call optimization mechanisms allow components transparent distribution and highly efficient communication when codeployed
- ▶ Transparency in service composition and invocation optimizations allows services to be deployed in a number of constellations
 - utilizing the framework as a middleware-independent 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



Service Structure



Task Management Service (TMS)





Middleware-
Independent Grid
Job Management

The Grid Job
Management
Framework (GJMF)

Conclusion

THE GRID JOB MANAGEMENT FRAMEWORK (GJMF)

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





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)





Middleware-
Independent Grid
Job Management

The Grid Job
Management
Framework (GJMF)

Conclusion

THE GRID JOB MANAGEMENT FRAMEWORK (GJMF)

GJMF Overhead

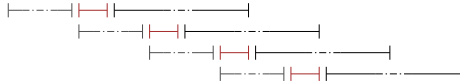
- ▶ GJMF Job execution overhead consists of 3 parts
 - job submission overhead (GJMF contribution)
 - job processing overhead (GJMF contribution)
 - job execution time (includes middleware overhead)
- ▶ GJMF overhead contributions mediated
 - GJMF overhead masked by parallelization
 - GJMF mechanisms mediate job submission latencies
 - GJMF contribution typically small





Sequential Invocation Overhead

Infinite resources



Limited resources





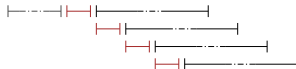
Batch Invocation Overhead

Middleware-Independent Grid Job Management

The Grid Job Management Framework (GJMF)

Conclusion

Infinite resources



Limited resources



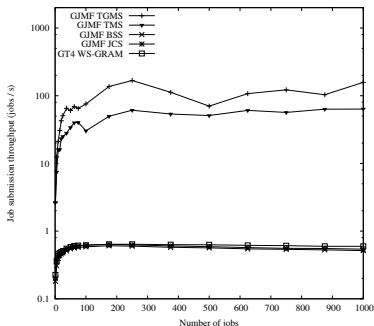
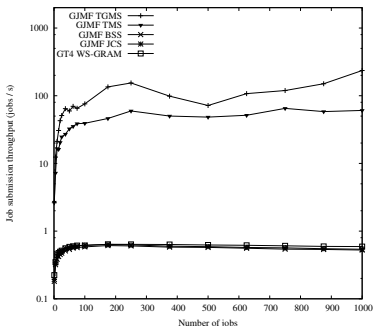
Job Submission Throughput

Ideal jobs, codeployed services, sequential & batch invocations

Middleware-Independent Grid Job Management

The Grid Job Management Framework (GJMF)

Conclusion



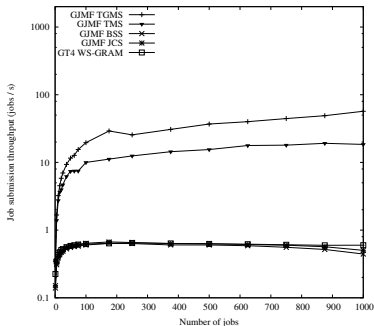
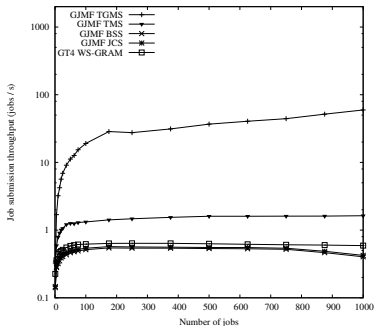
Job Submission Throughput

Middleware-Independent Grid Job Management

The Grid Job Management Framework (GJMF)

Conclusion

Ideal jobs, standalone services, sequential & batch invocations



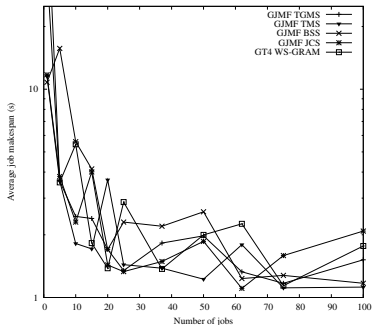
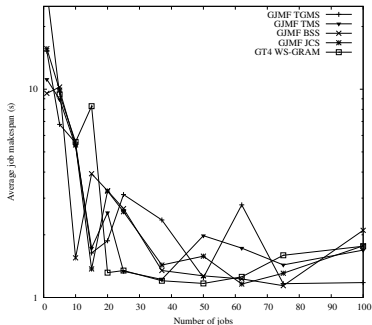
Job Processing Throughput

Realistic jobs, codeployed services, sequential & batch invocations

Middleware-Independent Grid Job Management

The Grid Job Management Framework (GJMF)

Conclusion



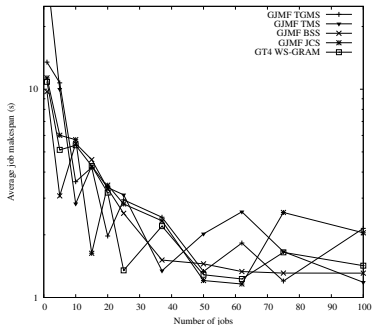
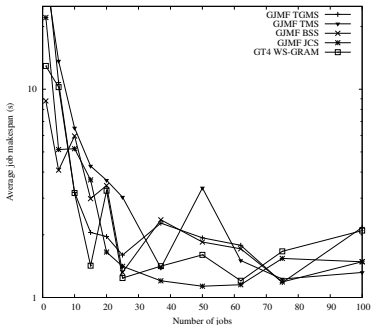
Job Processing Throughput

Middleware-Independent Grid Job Management

The Grid Job Management Framework (GJMF)

Conclusion

Realistic jobs, standalone services, sequential & batch invocations



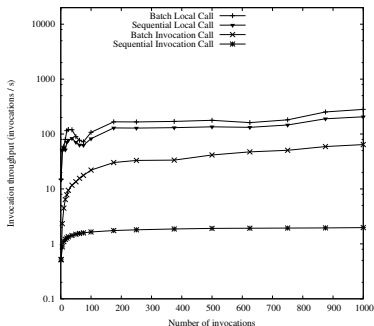
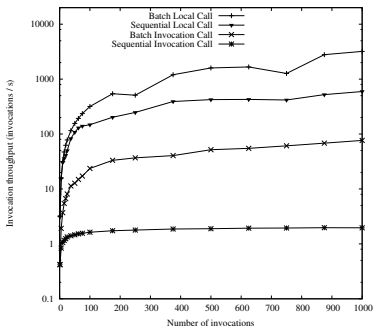
Invocation Throughput Comparison

Middleware-Independent Grid Job Management

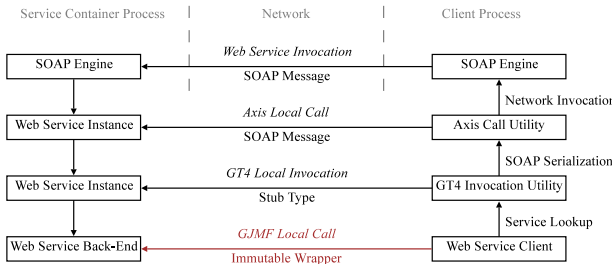
The Grid Job Management Framework (GJMF)

Conclusion

LAS & JTS Invocation Throughput



Service Invocation Optimization Types





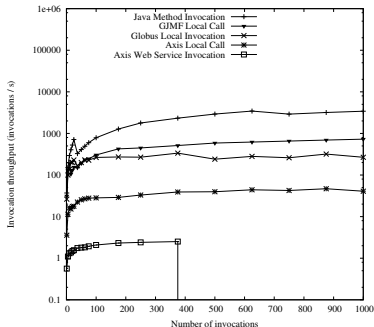
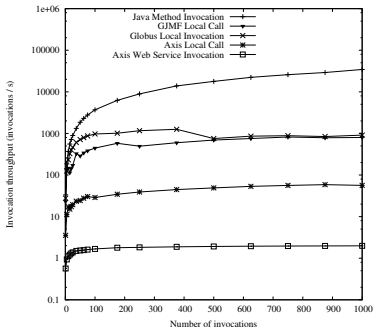
Service Invocation Optimization Benchmarks

Middleware-Independent Grid Job Management

The Grid Job Management Framework (GJMF)

Conclusion

Sequential & parallel invocations





Summary

- ▶ 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
- ▶ E. Elmroth and P-O. Östberg. A Composable Service-Oriented Architecture for Middleware-Independent and Interoperable Grid Job Management. *Submitted for journal publication*, 2009.
- ▶ P-O. Östberg. Architectures, Design Methodologies, and Service Composition Techniques for Grid Job and Resource Management. *ISSN 0348-0542 UMINF 09.15*, 2009.
- ▶ E. Elmroth, S. Holmgren, J. Lindemann, S. Toor, and P-O. Östberg. Empowering a Flexible Application Portal with a SOA-based Grid Job Management Framework. *The 9th International Workshop on State-of-the-Art in Scientific and Parallel Computing*, to appear, 2009.
- ▶ E. Elmroth and P-O. Östberg. Dynamic and Transparent Service Compositions Techniques for Service-Oriented Grid Architectures. In S. Gorlatch, P. Fragopoulou and T. Priol (Eds.), *Integrated Research in Grid Computing*, Crete University Press, pp. 323–334, 2008.
- ▶ E. Elmroth, F. Hernandez, J. Tordsson, and P-O. Östberg. Designing Service-Based Resource Management Tools for a Healthy Grid Ecosystem. *Parallel Processing and Applied Mathematics*, Springer-Verlag, Lecture Notes in Computer Science, vol. 4967, pp. 259-270, 2008.
- ▶ E. Elmroth, P. Gardfjäll, A. Norberg, J. Tordsson and P-O. Östberg. Designing general, composable, and middleware-independent Grid infrastructure tools for multi-tiered job management. In T. Priol and M. Vaneschi (Eds.) *Towards Next Generation Grids*. Springer Verlag, pp. 175 - 184, 2007.