Open Source Grid Computing – Java Roundup

Nikita Ivanov

www.gridgain.org
Introduction

- Nikita Ivanov
  - Over 15 years of experience
  - Last 7 years developing grid computing and distributed middleware
  - JSR-107 “JCache” Expert Group member

- GridGain Founder, [www.gridgain.org](http://www.gridgain.org)
  - Computational grid
  - Java 5
  - LGPL professional open-source
Presentation Overview

- What is Grid Computing and why?
- Why Java?
- Why Open Source?
- Comparative Analysis
  - 9 categories of comparison
  - Unbiased as possible – but not completely
  - Based on experience of:
    - Using
    - Developing
Why Grid Computing?

- Ask Google, Yahoo, eBay, Amazon
- Solves problems often unsolvable otherwise
- Allows smaller businesses to compete with big corporations
  - Many examples in financial sector
- Ideal technology for Web 2.0, mash-ups, geo-processing, etc.
  - Start small – grow with the business
  - Uniform programming model throughout life-cycle of business
Why Java?

- Most of today grids are running C/C++
  - HPC/MPI legacy
  - Fortran/C/C++ libraries
- Yet Java has a number of intrinsic benefits:
  - Much more productive than C/C++
  - Performance gap is minimal
  - Easily call out to existing Fortran/C/C++ libraries
  - Extensive server-side software stack
    - Commercial
    - Professional open-source
  - Cross-platform (at least for server side systems)
    - Makes heterogeneous grids a simple reality
Why Open Source?

- Best way to **own** a middleware product
  - No per-CPU license penalties
  - Free to try and free to use
- Developers friendly
  - Feel like a part of the team
- LGPL Business friendly
  - JBoss, Spring, Mule, *etc.* successes
  - Selling services vs. licenses
Open Source Java Grid Computing

- **Globus**
  - [http://www.globus.org](http://www.globus.org)

- **Sun Grid Engine**
  - [http://gridengine.sunsource.net](http://gridengine.sunsource.net)

- **GridGain**
  - [http://www.gridgain.org](http://www.gridgain.org)
Globus

- **Description**
  - OGSA/OGSI-based general purpose grid toolkit
  - De-facto standard in academic grid computing
  - Java is well supported

- **History**
  - Work started in 1994-1996
  - First released in 1998
  - Univa corp. founded in 2004

- **Current state**
  - Current version is 4.0
Sun Grid Engine

- Description
  - Primarily a DRM grid platform with some Java support
  - Batch-queuing

- History
  - Acquired by SUN from Gridware in July 2000

- Current state
  - Current version is 6.1
GridGain

- Description
  - Java-centric computational grid computing platform
    - Java language centric
    - JEE paradigm centric

- History
  - Project started in June 2005
  - First release in January 2007

- Current state
  - Current version is 1.6
Analysis Criteria

- License
- Support
- Community
- Documentation
- Maturity
- Standards
- Developer’s resources
- Integration
- Key grid computing concepts
License

- **Globus**
  - Apache (4.0.1+)
  - Huge list of 3rd party licenses
  - Open source version is not limited

- **Sun Grid Engine**
  - SISSL – Sun Industry Standards Source License
    - FSF and OSI recognized
  - Open source version is limited

- **GridGain**
  - LGPL
  - 3rd party libraries GPL, MIT and Apache
  - Open source version is not limited
Support

- **Globus**
  - Large online community for free support
  - Univa corp. for commercial support
  - Even IRC :-)

- **Sun Grid Engine**
  - Small online community for free support
  - SUN for commercial support

- **GridGain**
  - Email, forums, wiki, JIRA for free support
  - GridGain Systems for commercial support
Documentation

- Globus
  - Extensive: http://www.globus.org/toolkit/docs/4.0/

- Sun Grid Engine
  - Extensive: http://gridengine.sunsorce.net

- GridGain
  - Extensive: www.gridgain.org
  - Wiki, JIRA and Forums
Maturity

- Globus
  - Most widely used
  - Core components are extremely stable and well tested

- Sun Grid Engine
  - Sparsely used
  - Mature and well tested

- GridGain
  - Young product
  - Less than 1 year in usage
Community

- **Globus**
  - Most used grid computing software
  - Thousands of active users

- **Sun Grid Engine**
  - Over 9000 downloads in first 9 months
  - Hundreds of active users

- **GridGain**
  - Over 2000 downloads for the last 4 months
  - Dozens of active users
Standards

- Globus
  - OGSA/OGSI
  - Heavily WS-*

- Sun Grid Engine
  - DRMAA

- GridGain
  - Java-centric
  - Non WS-*
  - Following JEE methodology:
    - IoC, resource injection, annotation-based AOP, etc.
Developer’s Resources

- Globus
  - Extensive online documentation
  - Number of book
  - Conferences, seminars and workshops
    - Univa corp.
    - Globus Alliance

- Sun Grid Engine
  - Extensive online documentation
  - Number of books
  - Training through SUN

- GridGain
  - Online documentation
  - Training through GridGain Systems
Integration

- Globus
  - Highly modular via design
  - Highly customizable
  - Unix, Linux, Mac OS X, limited Windows

- Sun Grid Engine
  - Limited integration
  - Limited Windows support
  - Unix, Linux, Mac OS X, limited Windows

- GridGain
  - Highly integrated and customizable via SPIs:
    - JBoss, Spring, Mule, Coherence, Weblogic, WebSphere, etc.
  - Unix, Linux, Windows, Mac OS X
Key Grid Computing Concepts

Globus
Key Grid Computing Concepts

Globus
Key Grid Computing Concepts

Sun Grid Engine

1. Execds report load information to qmaster
2. User submits job using qsub command
3. Qmaster notifies Schedd about new job
4. Schedd dispatches job to an Execd
5. Qmaster delivers job to Execd; Execd starts job using Shepherd
6. At job end Execd notifies Qmaster about job finish
7. Qmaster feeds jobs resource consumption in accounting database
Key Grid Computing Concepts

Sun Grid Engine

- Policy based allocation of distributed resources (CPU time, software licenses, etc.)
- Batch queuing & scheduling
- Support diverse server hardware, OS and architectures
- Load balancing & remote job execution
- Detailed job accounting statistics
- Fine-grained user specifiable resources
- Suspend/resume/migrate jobs
- Tools for reporting Job/Host/Cluster status
- Job Arrays
- Integration & control of parallel jobs
Key Grid Computing Concepts

GridGain

1. Initial (root) job execution request
2. Initial job split into sub-jobs
3. Result of sub-job execution
4. Aggregation of sub-results into final result

Node 1

Node 2

Node 3

Node 4
Key Grid Computing Concepts

GridGain

- Direct support for split & aggregate *a.k.a.* map/reduce
- Fully customizable:
  - Grid node discovery, communication
  - Failover, deployment
  - Checkpoints, collision resolution *a.k.a.* scheduling
  - Load balancing, topology management
- Integration with:
  - Jboss, Spring, Spring AOP
  - JBoss AOP, AspectJ, JGroups
  - Weblogic, Websphere, Oracle Coherence
  - Mule, JXInsight, GigaSpaces
Globus: Summary

- **Pros:**
  - Mature and well established
  - Ideal for academia or global grids
  - Plenty of 3rd party support
  - Largest ecosystem

- **Cons:**
  - Too complex (costly) to use
  - Heavy WS-*, OGSA/OGSI is not for everyone (anyone?)
  - Limited Java support
Sun Grid Engine: Summary

- Pros:
  - Official “interface” to Sun Grid
  - Well establish and stable

- Cons:
  - Limited in key grid computing features
    - More DRM than grid computing
  - Limited 3rd party support
  - Java support and open source are limited
  - Command line interface
GridGain: Summary

- **Pros:**
  - 100% Java centric
  - Clear focus on computational grid computing
  - Simple and productive to use

- **Cons:**
  - Young product
  - Developer’s resources are limited
  - Less stable than the other two
Q & A

Thanks for your time!

Nikita Ivanov: nivanov@gridgain.com
GridGain: www.gridgain.org