GridGain – Java Grid Computing With AOP

Nikita Ivanov
www.gridgain.org

Emerging Technologies for the Enterprise
Philadelphia, March 2008

Agenda

- GridGain
  - What is Grid Computing and why?
  - What is GridGain?
  - Key Concepts
- Demos
  - Grid Application in 15 Minutes

What is Grid Computing?

- Compute Grids
  - Parallelize logic execution
- Data Grids
  - Parallelize data storage
- Grid Computing = Compute Grids + Data Grids
  - a.k.a. Data Partitioning + Affinity Map/Reduce

Why Grid Computing?

- Ask Google, Yahoo, eBay, Amazon
- Solves problems often unsolvable otherwise
  - Google has ~1,000,000 nodes in its grid
- Uniformed programming paradigm
  - Scales from garage to Google

What is GridGain?

- Grid computing framework
  - Compute + Data = Grid Computing
    - Innovative MapReduce
    - Integration with leading Data Grids
  - Ground-breaking simplicity
- Java centric
  - Built in Java and for Java
- Open source
  - LGPL/Apache license

Professional Open Source

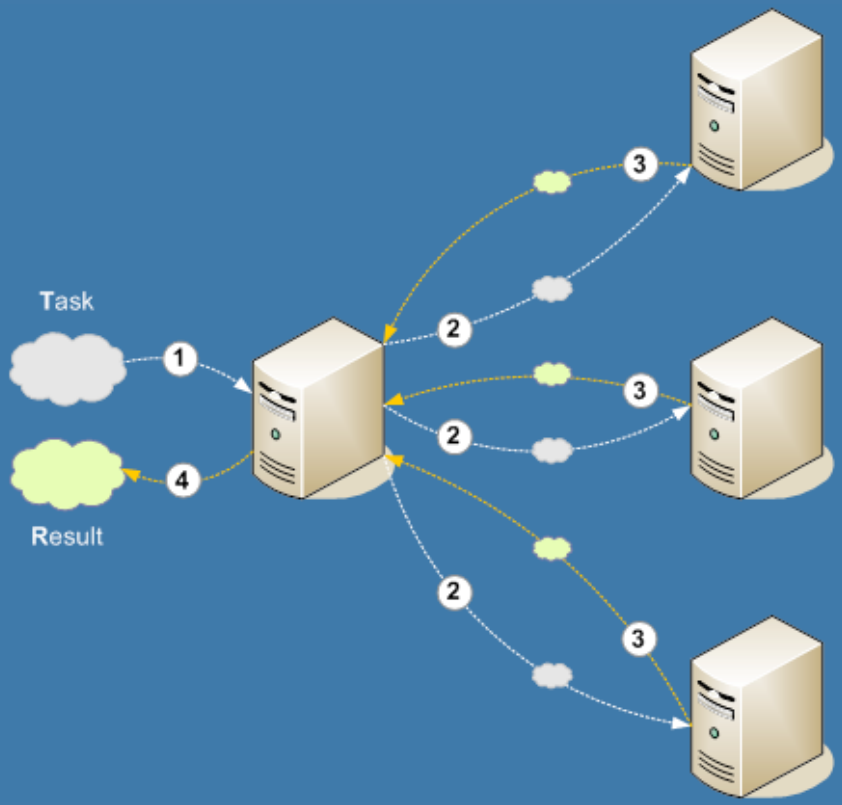
- GridGain - Professional Open Source
  - Open source => Free
    - Software
    - Community support
    - Source code
    - No gimmicks, no attributions, no strings attached
  - Plus commercial enterprise-level support and services
    - Indemnification
    - Custom SLAs
    - On-demand patches
    - Advanced management & monitoring
- Like JBoss, Spring Source, Mule Source…

Key Concepts

- MapReduce
- Zero Deployment
- On Demand Scalability
- Fault Tolerance
- Blend-In Integration
- Transparent Grid Enabling
- Data Grids Integration
- JMX-based Monitoring

MapReduce

1. Task execution request
2. Task splits into jobs
3. Result of job execution
4. Aggregation of job results



Zero Deployment

- Peer-to-Peer On-Demand Class Loading technology
  - No Ant scripts to run
  - No JARs to copy or FTP
  - No nodes to restart
- Develop in EXACTLY the same way as locally
  - Change->Compile->Run on the grid
- Start many grid nodes in
  - Single JVM – debug grid apps locally (!)
  - Single computer – run grid on your workstation

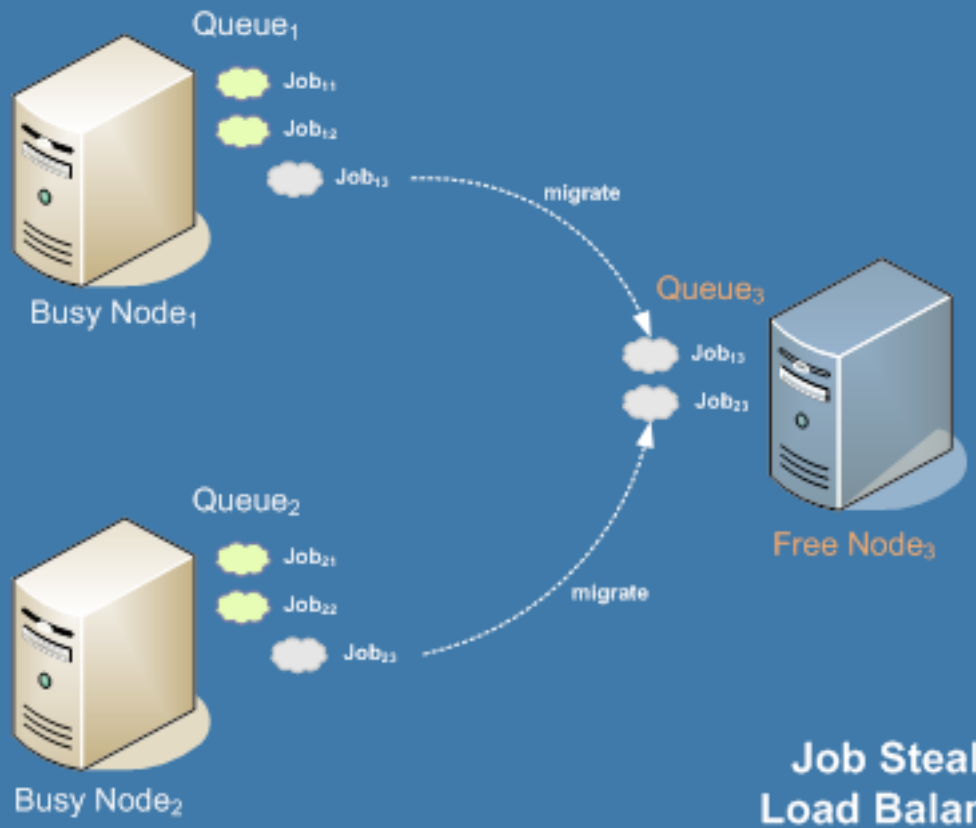
=> Single biggest developers' productivity boost

On Demand Scalability

- Early and late load balancing:
  - Optimal scalability for non-deterministic execution on the grid
- Load Balancing SPI
  - Early load balancing
- Collision (scheduling) SPI
  - Late load balancing

=> Most comprehensive scalability support

On Demand Scalability, cont.



Fault Tolerance

- Customizable failover resolution
  - Automatic failover
  - Fail-fast, fail-slow implementation
- Failure – is result too
- Redundant jobs
- Asynchronous results processing
  - Policy-based continuation
- Checkpoints for long-running tasks
  - “Smart” restart in case of failover

Blend-In Integration

- Service Provider Interface (SPI)-based architecture
  - Plug in and customize almost any aspect of grid computing framework
  - LEGO-like assembly of custom grid infrastructure
- Grid computing framework aspect that are fully pluggable:
  - Communication
  - Discovery
  - Tracing
  - Startup
  - Event storage
  - Marshalling
  - OnDemand
  - Checkpoints
  - Failover
  - Collision Resolution
  - Topology management
  - Load balancing
  - Deployment

Blend-In Integration, cont.

"Out-of-the-box" integration with:

Application Servers
- JBoss AS
- BEA Weblogic
- IBM Websphere
- Glassfish
- Tomcat

Data Grids
- JBoss Cache
- Coherence
- GigaSpaces

AOP
- JBoss AOP
- Spring AOP
- AspectJ

Messaging Middleware
- Mule
- JMS
  - ActiveMQ
  - SunMQ
- Jgroups
- Email
- TCP, IP-Multicast

Others
- Spring
- Junit
- JXInsight

Transparent Grid Enabling

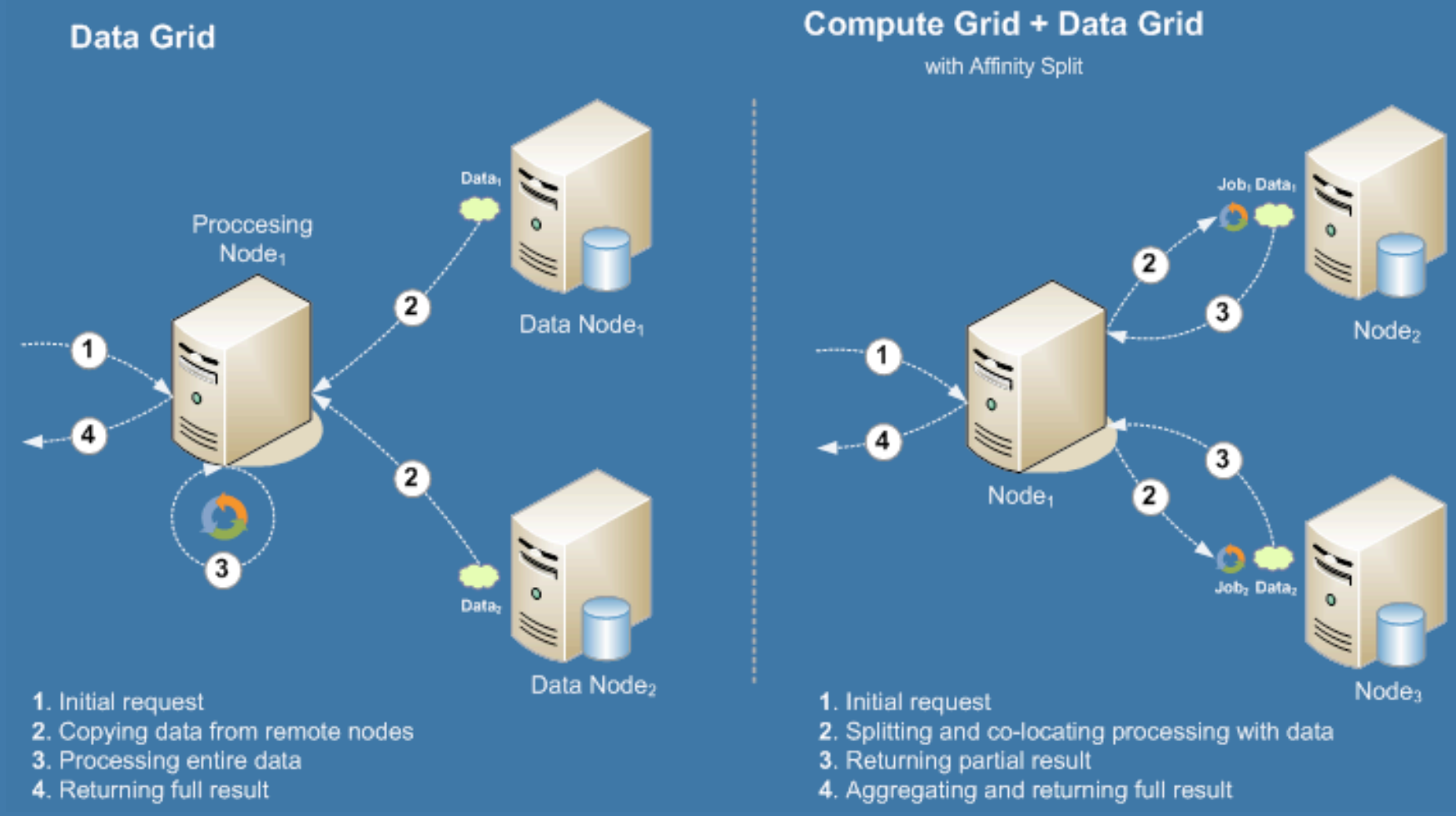
```
01 class BizLogic {
02   @Gridify(...)
03   public static Result process(String param) {
04     ...
05   }
06 }
07
08 class Caller {
09   public static void Main(String[] args) {
10     GridFactory.start();
11
12     try {
13       BizLogic.process(args[0]);
14     }
15     finally {
16       GridFactory.stop();
17     }
18   }
19 }
```

Execution of process() method will be performed on the grid

Data Grids Integration

- Compute + Data = Grid Computing
  - Out-of-the-box support:
    - JBoss Cache, Oracle Coherence
- Affinity Map/Reduce – ability to co-locate processing logic and the data
  - Minimizes “noise” traffic
  - Optimal grid load and performance

Data Grid Integration, cont.



JMX-Based Monitoring

- Full JMX instrumentation
  - Every SPI
  - Kernal
- Flexible access
  - Programmatic via JMX API
  - From GUI JMX console
    - Jboss Management
    - Hyperic
    - JConsole

Roadmap

- GridGain 1.5.0 out July 27th, 2007
- GridGain 2.0.0 out on February 12th, 2008
- GridGain 2.0.1 out on March 7th, 2008
- GridGain 2.0.2 out this weekend ☺
- GridGain 2.1 – Q308
  - Mobile Grid Computing: Google Android
  - Utility or on-demand computing: Amazon EC2
  - Web 2.0 Grid Computing: REST + JSON
  - Enhanced Management and Monitoring

Demos

- Java 5/Eclipse 3.2/Windows XP
- GridGain 2.0

Q & A

Thanks for your time!

Nikita Ivanov: nivanov@gridgain.com

GridGain: www.gridgain.org