

GridGain – Open Cloud Platform



www.gridgain.org

Agenda

- **GridGain**
 - **What is Grid/Cloud Computing**
 - **GridGain In a Glance**
 - **Key Concepts**
- **Demos**
 - **Grid Application in 15 Minutes**

What is Grid/Cloud Computing?

Grid Computing = Compute Grids + Data Grids

- Compute Grids: parallelize execution
- Data Grids: parallelize data storage

Cloud Computing = **Grid Computing** + Data Center Automation

Cloud computing is a new way to deploy
and run grid applications

Why Grid Computing?

- Ask Google, Yahoo, eBay, Amazon
 - Amazon: 100ms latency cost 1% of sales
 - Google: 500ms latency drops traffic 20%
 - Financial: \$4M/ms lose if 5ms behind
- Solves problems often unsolvable otherwise
 - Google has ~1,000,000 nodes in its grid
- Uniformed programming paradigm
 - Scales from garage and up

Why Cloud Computing?

- Ask Google, Yahoo, eBay, Amazon 😊
 - In-house, enterprise clouds
- Potential cost reduction of **10x** and **100x**
 - If you don't run 24/7
- Grid computing as it should have been
 - Final piece in a long
HPC -> Grid Computing -> [...] saga

GridGain In a Glance

Open Source Java-Based Open Cloud Platform

- Open Cloud Platform
 - Innovative Compute/Data Grid
 - Native Cloud Applications
- Java
 - Built in Java and for Java
- Open Source
 - LGPL and Apache 2.0

Elegant Simplicity with Powerful Features

Professional Open Source

- GridGain - **Professional Open Source**
 - Free and Open Source licenses: [LGPL](#) and [Apache 2.0](#)
 - Commercial support, training and consulting
- Best business model for software middleware
- Like JBoss, Spring Source, Mule Source...

GridGain Statistics

In **18 months** since the 1st release:

- Over 30,000 downloads
- Starts every 60 seconds around the globe
- One of the largest Amazon EC2 clouds – 512 nodes
- Over 2000 different individuals, projects and organizations

Fastest Growing **Java-Based Cloud Computing Platform**

Key Concepts

- Native Cloud Applications
- MapReduce
- Zero Deployment
- Comprehensive Load Balancing
- Fault Tolerance
- LEGO-like Integration
- Transparent Grid Enabling
- Data Grids Integration

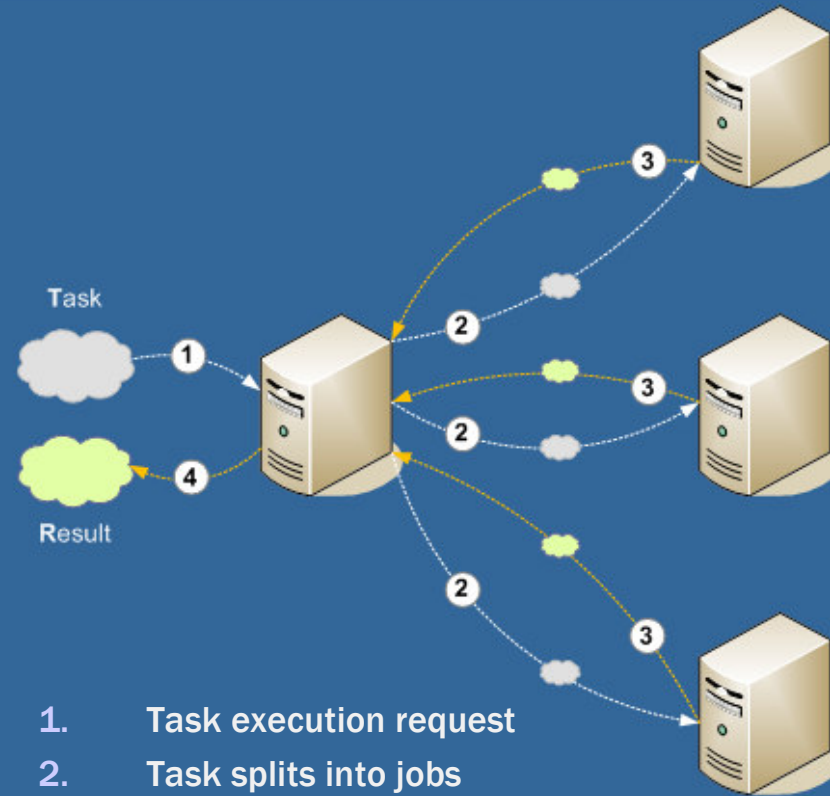
Native Cloud Applications

- **Massively parallel**
 - Ability to scale on fine grain level for extreme performance
- **Full advantage of native cloud services**
 - Develop with native services like S3 or SimpleDB
- **Cross-cloud paradigm**
 - Utilize full location transparency with virtualized cloud
 - Avoid cloud provider vendor lock-n

MapReduce

Features:

- Direct API support for MapReduce
- Pluggable failover resolution
- Pluggable topology resolution
- Distributed task session
- Annotation-based execution
- Asynchronous execution
- Redundant mapping
- Partial asynchronous reduction
- Adaptive split
- Checkpoints for long running tasks
- Early and late load balancing
- Affinity co-location with data grids



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

Zero Deployment

- Peer-to-Peer Grid Class Loading technology
 - No Ant scripts to run
 - No JARs to copy or FTP
 - No need 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
- ⇒ Biggest developer's productivity boost

Comprehensive Load Balancing

- **Early and late** load balancing:
 - Optimal scalability for non-deterministic execution on the grid
 - Load Balancing SPI
 - Early load balancing
 - Collision SPI
 - Late load balancing
- => Most comprehensive scalability support**

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
- => Most comprehensive fault tolerance functionality

LEGO-Like 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
 - Design approach enabling transparent usability for HPC, traditional grid computing and cloud computing
- 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

LEGO-like Integration

“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 }
```

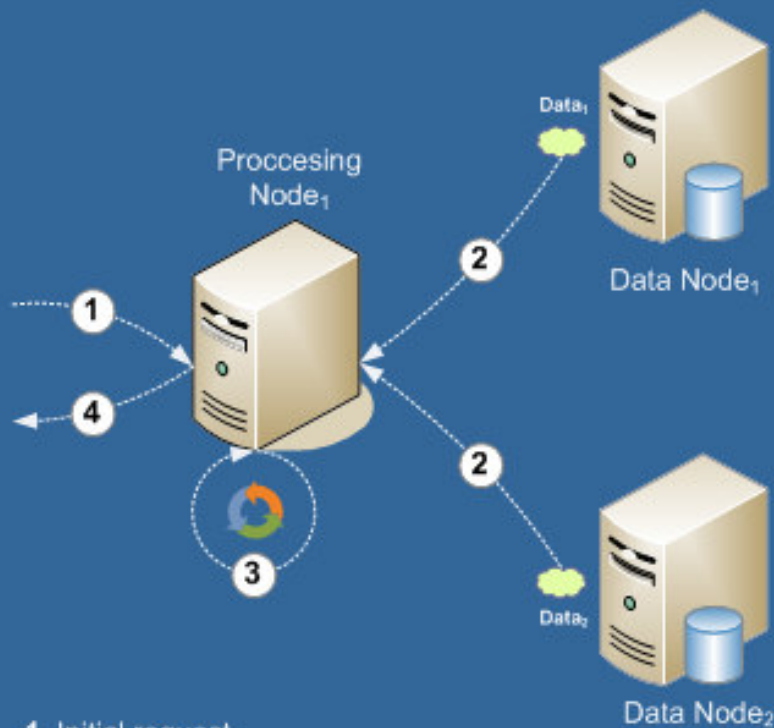
Execution of `process()` method will be performed on the grid

Data Grids Integration

- Integration with Data Grids – **key** to ultimate scalability
- Affinity MapReduce – ability to co-locate processing logic and the data
 - a.k.a. Data-aware routing
 - Minimizes “noise” traffic
 - Optimal grid load and performance
- Out-of-the-box support:
 - JBoss Cache
 - Oracle Coherence

Data Grid Integration

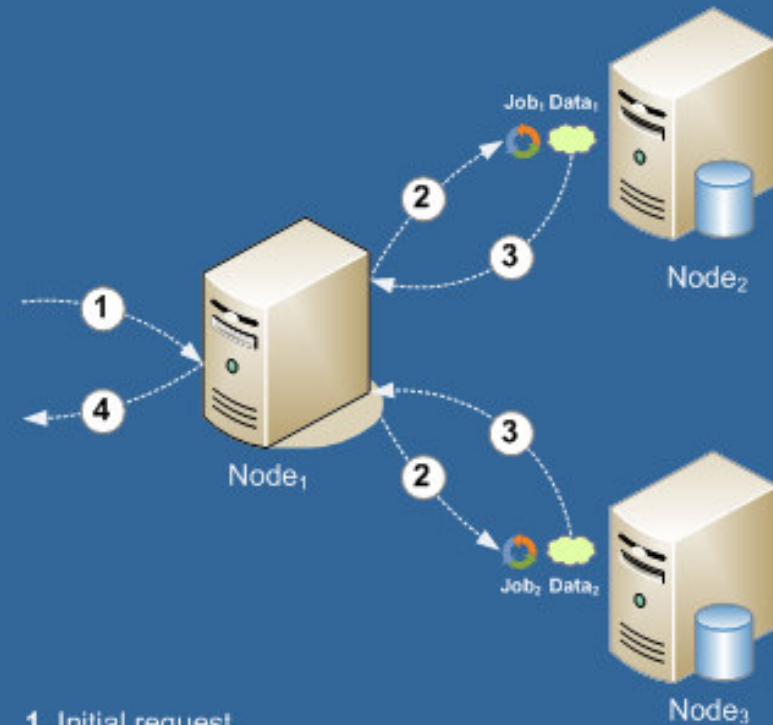
Data Grid



1. Initial request
2. Copying data from remote nodes
3. Processing entire data
4. Returning full result

Compute Grid + Data Grid

with Affinity Split



1. Initial request
2. Splitting and co-locating processing with data
3. Returning partial result
4. Aggregating and returning full result

Demo

- 10 minutes to get your 1st grid application
- GridGain 2.1
- Eclipse 3.3

Q & A

Thanks for your time!

GridGain: www.gridgain.org

Nikita Ivanov: nivanov@gridgain.com