

GemFire Administration

Instructor-led Training

Version 9.0.4c

Barry Wood

bwood@rpymotion.com

Content ID: EDU-1086-GemFire-Adm ...

Pivotal

Copyright Notice

Copyright © 2017 Pivotal Software, Inc. All rights reserved. This manual and its accompanying materials are protected by U.S. and international copyright and intellectual property laws.

Pivotal products are covered by one or more patents listed at <http://www.pivotal.io/patents>.

Pivotal is a registered trademark or trademark of Pivotal Software, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies. The training material is provided “as is,” and all express or implied conditions, representations, and warranties, including any implied warranty of merchantability, fitness for a particular purpose or noninfringement, are disclaimed, even if Pivotal Software, Inc., has been advised of the possibility of such claims. This training material is designed to support an instructor-led training course and is intended to be used for reference purposes in conjunction with the instructor-led training course. The training material is not a standalone training tool. Use of the training material for self-study without class attendance is not recommended.

These materials and the computer programs to which it relates are the property of, and embody trade secrets and confidential information proprietary to, Pivotal Software, Inc., and may not be reproduced, copied, disclosed, transferred, adapted or modified without the express written approval of Pivotal Software, Inc.

GemFire Administrator

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Course Objectives

After this class, you should be able to do the following:

- Understand basic installation requirements and perform basic installation and configuration tasks
- Understand the Client/Server architecture
- Perform GemFire administrative tasks using gfsh command line tool
- Configure a distributed system and its members
- Configure and use logging and statistic tools such as JConsole, Pulse, and Visual Statistics Display to monitor and gather data about the distributed system
- Troubleshoot problems associated with the distributed system and its members

Course Introduction

- Course Duration: 3 days
- Course is designed for Windows, Linux, and Mac
 - 50% Theory
 - 50% Labs
- Students should have the following:
 - Student Handout
 - Student Lab Manual
 - Lab files (contains GemFire installation file)

Course Prerequisite

This is a GemFire Administration class:

- Knowledge of basic networking
- Basic command line knowledge
- Understanding of how to manage Java Applications
- Use of text editor of your choice

This is *not* a GemFire Programming class:

- Knowledge of Core Java not required
- Knowledge of Object modeling not required
- Use of IDE not required

Course Agenda: Day 1

- GemFire Terminology Review
- Basic Installation/Configuration
- Client/Server Configuration
- Cluster Configuration Service

ContentName : EDU-100-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



Course Agenda: Day 2

- Replicated and Partitioned Regions
- JVM Options and Off-heap
- Administration/Management Basics
- Monitoring Tools
- Network Architecture

ContentName : EDU-1086-Gemfire-Admin ...
Barry Wood
bwood@rpymotion.com



Course Agenda: Day 3

- Disk Stores
- Backup and Restore
- Security Overview
- Logging, Statistics and Troubleshooting
- Multi-Site Configuration

ContentName : EDU-100_GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



Gemfire Administration



Let's Get Started!

ContentName : EDU-386-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



Pivotal™

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

GemFire Terminology, Topologies & Use Cases

ContentName : EDU-106-GemFire-Admin ...
Barry Wood
bwood@rpymotion.com

Objectives

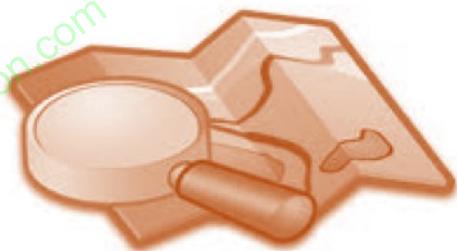
After completing this lesson, you should be able to:

- Define GemFire's purpose in the enterprise system
- Describe GemFire's contributions to the enterprise system
- Understand GemFire-specific terminology
- Describe common GemFire topologies
- Describe common use cases for GemFire

Lesson Road Map

- **GemFire**
- GemFire Terminology
- GemFire Common Topologies
- GemFire Use Cases

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

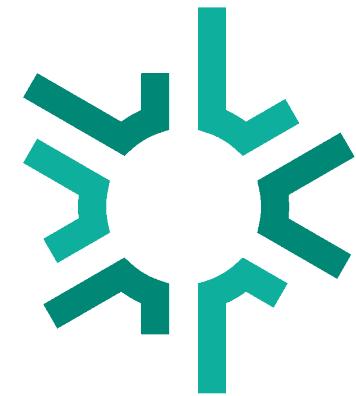


Pivotal GemFire

In-Memory Data Grid Powered by Apache Geode

Predictable low latency for highly concurrent read/write access in a horizontally scalable and reliable way for cloud and multi-site deployments.”

- In-memory data grid built for predictable low-latency at scale
- Cluster based system running on “commodity” hardware & OS
- Cloud ready
- Multi-site capable
- 10+ years of R&D investment
- Core of GemFire recently contributed to Apache Geode project



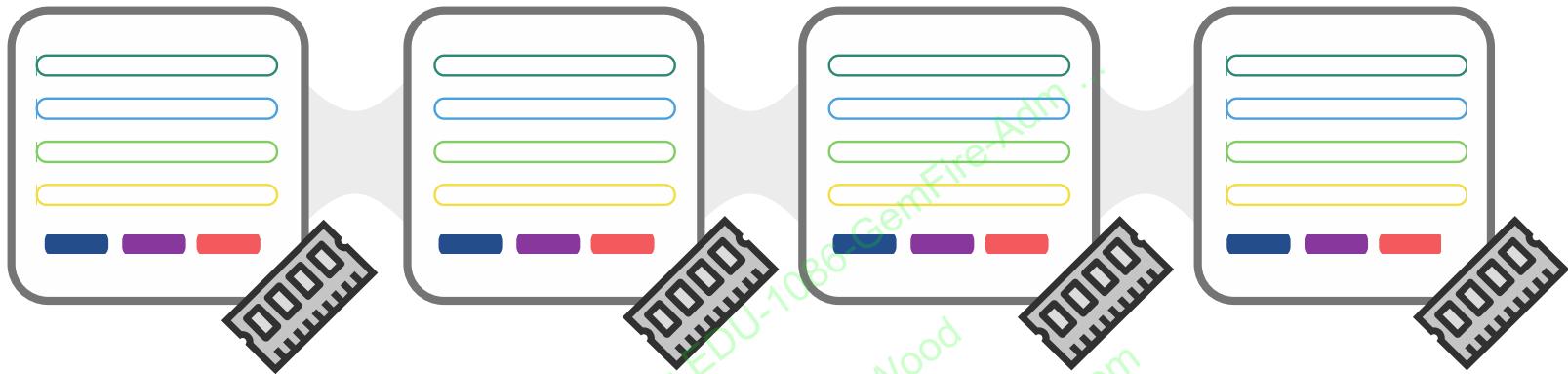
GemFire - The Enterprise Data Grid

- A distributed memory based data management platform
- Gartner - *In Memory Data Grid (IMDG)*
- Provides continuous availability, high performance, and linear scalability for data intensive applications
- Allows for configurable data consistency
- Event driven data architecture

Content ID: ETL-108
Author: Bryan Wood
Email: bwood@rpymotion.com



Key Features



- Key/Value Object Store
- Horizontally Scalable and Elastic
- Memory Oriented
- Active Everywhere – Multi-Site

Lesson Road Map

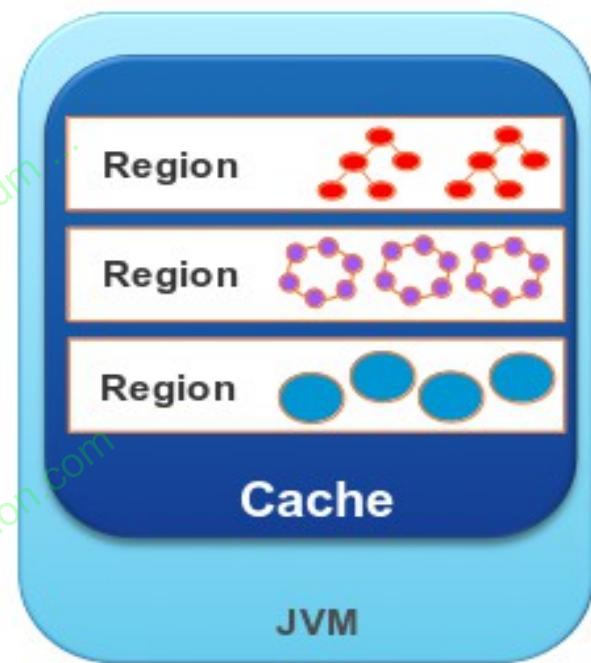
- GemFire IMDG
- **GemFire Terminology**
- GemFire Common Topologies
- GemFire Use Cases

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



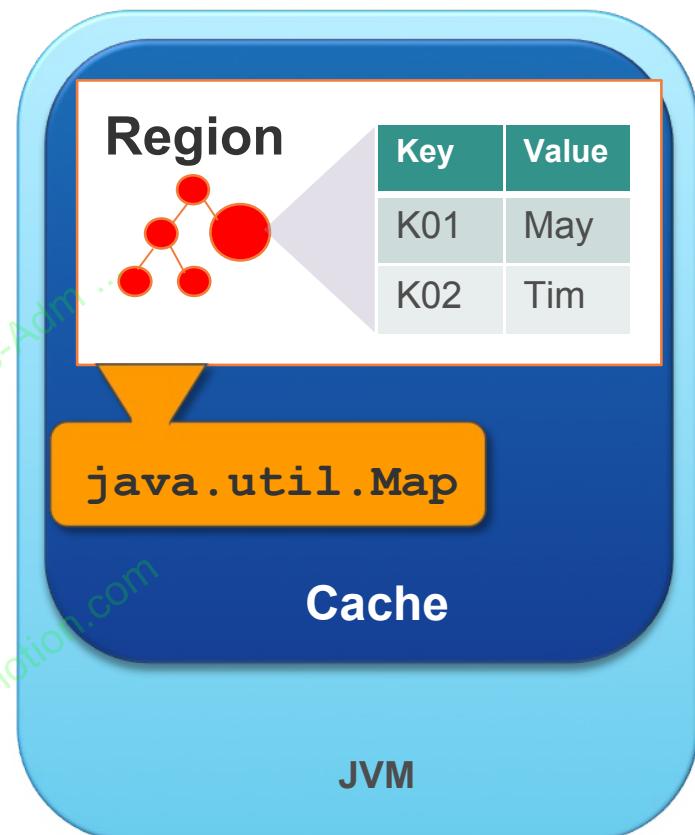
Cache

- Cache provides in-memory data storage and management
- Cache configuration
 - gfsh
 - XML declaration
 - Java API
- Consist of Regions (distinct named data sets)



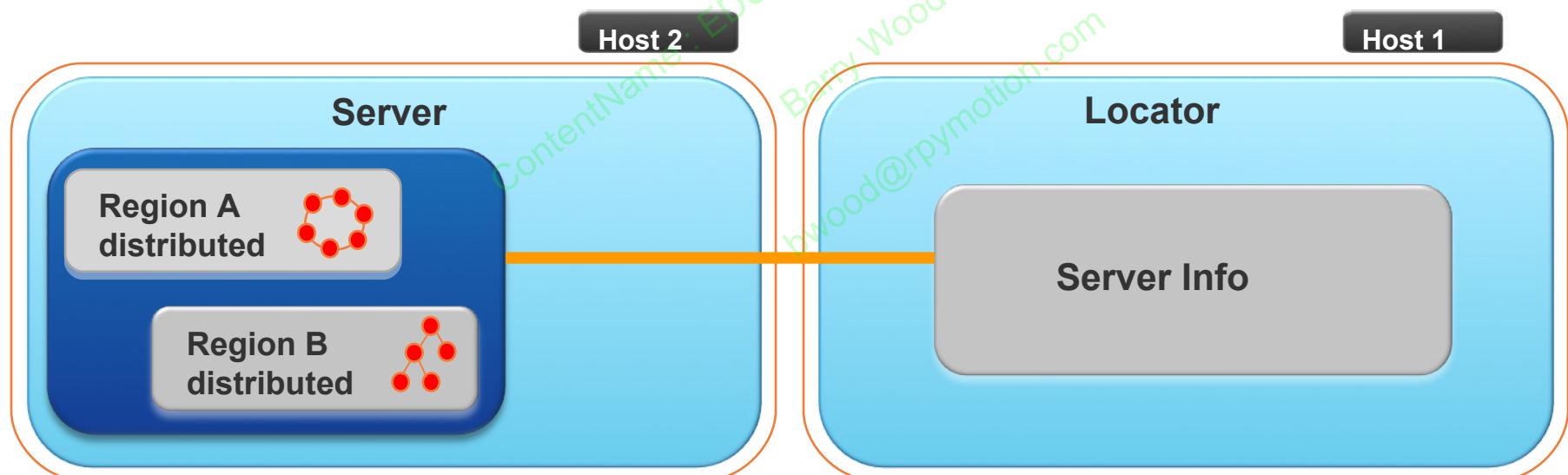
Region

- Region groups cached objects into hierarchical namespaces
- Data objects are stored in a Region as key/value pairs
- Keys must be unique within a given region
- Manages a set of data via configuration



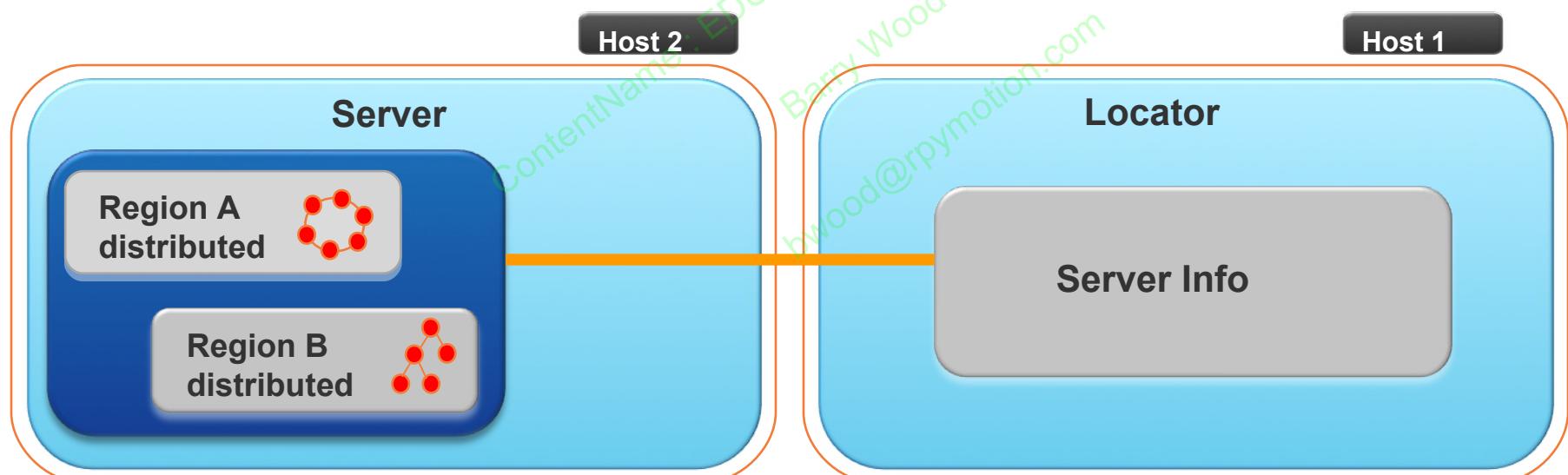
Server

- Server host cache
- Cache made up of zero or more regions
- Process that service client requests
- Communicates status information to Locator
- Also called *Cache Server*



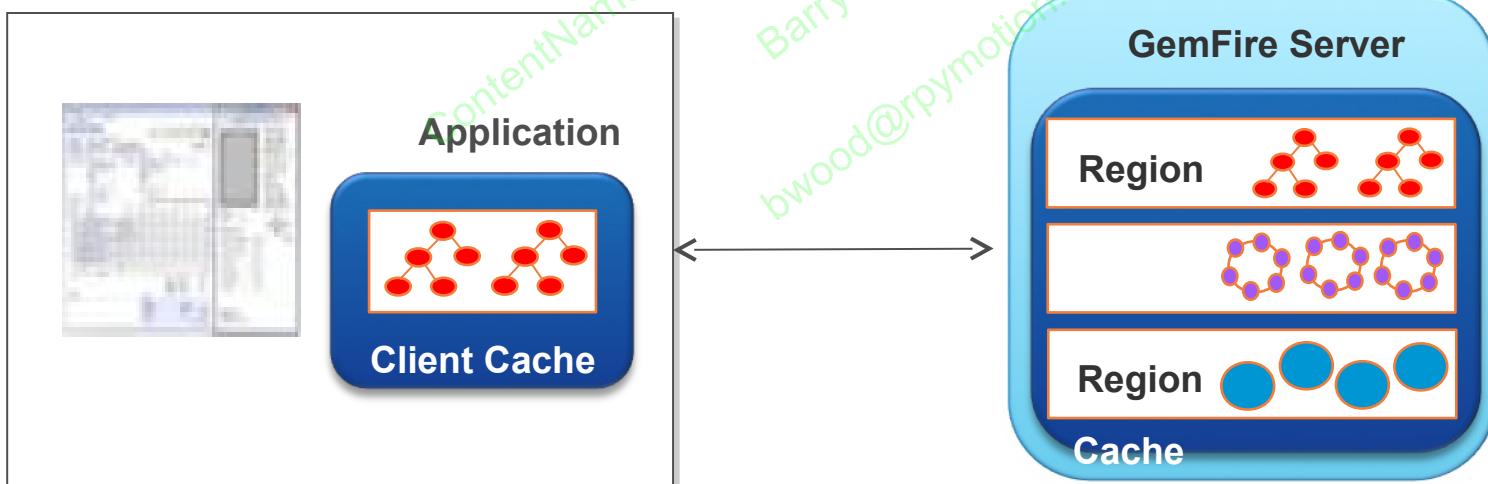
Locator

- Are used for peer discovery by the servers
- Are used for server discovery by the clients
- Are specified in a Pool instance in the client applications
- Give dynamic server information to clients
- Provide server connection load balancing & fault tolerance



Client Cache

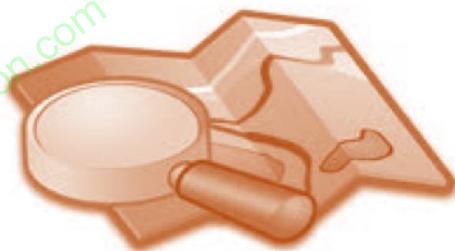
- Client gets the list of servers to connect to from the locator
- Clients can access data on servers AND can optionally cache a local copy
- If the client registers for changes, the server notifies the client on change



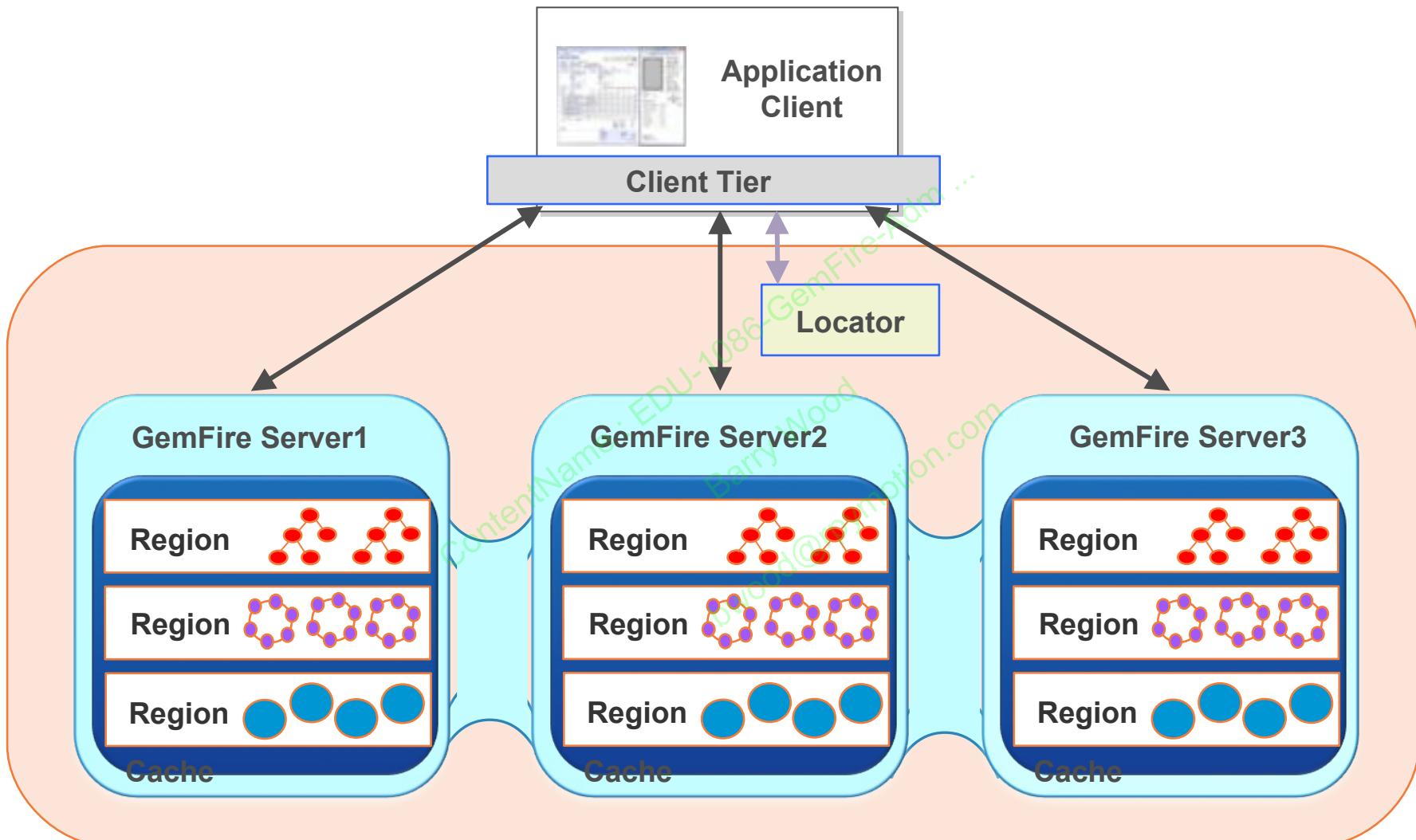
Lesson Road Map

- GemFire IMDG
- GemFire Terminology
- **GemFire Common Topologies**
- GemFire Use Cases

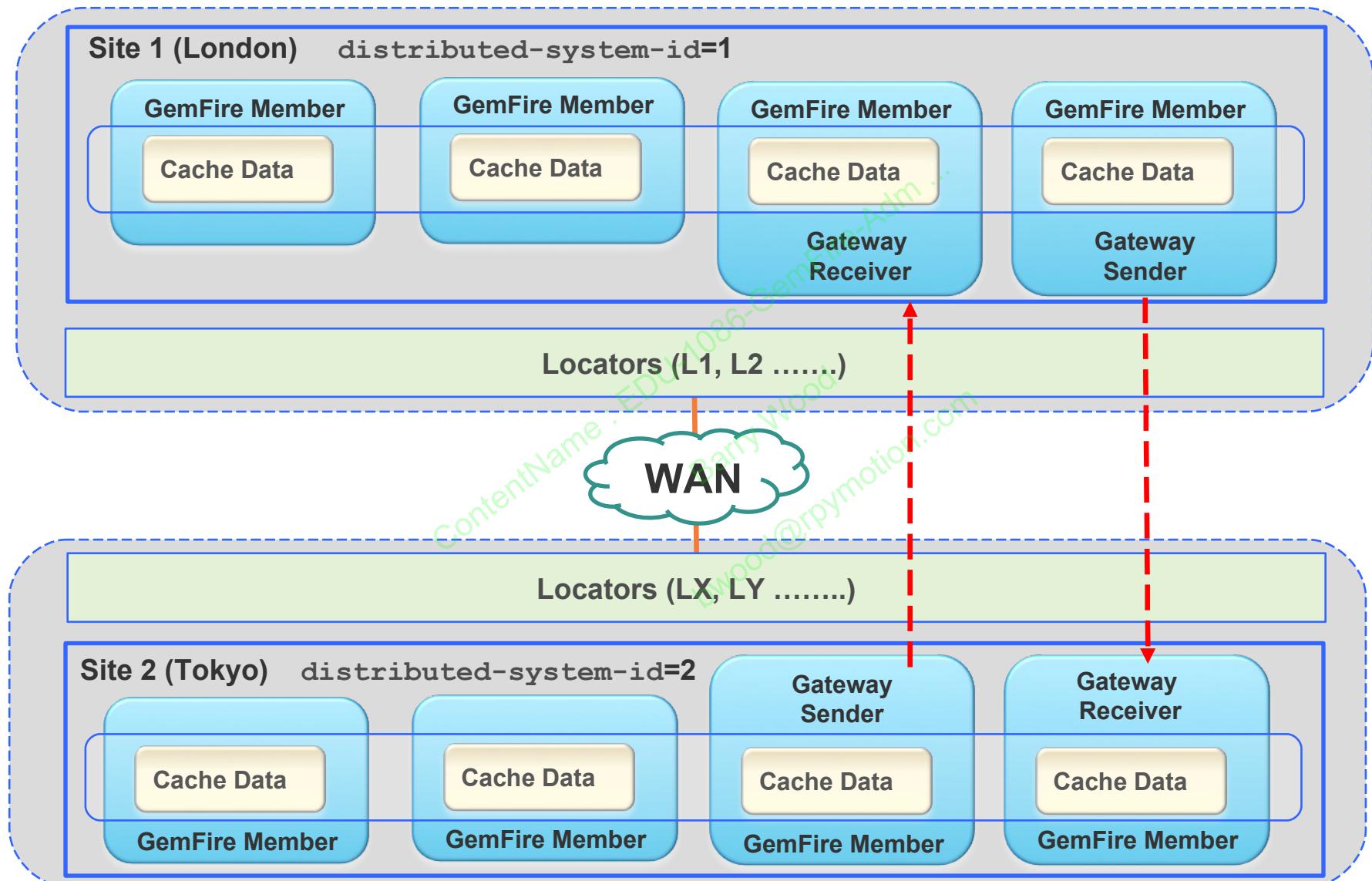
ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



Client-Server Topology



Multi-site (WAN) System



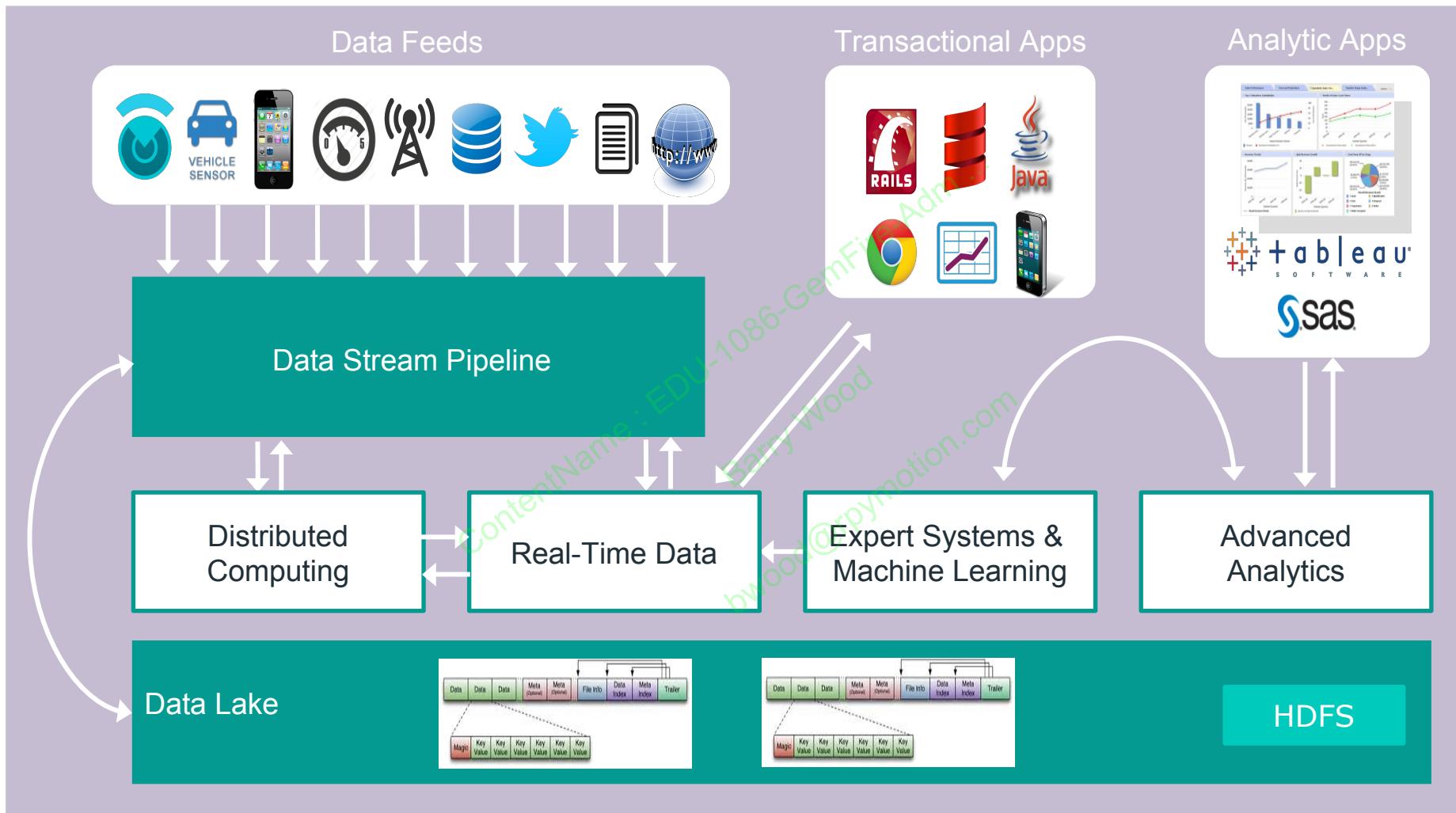
Lesson Road Map

- GemFire IMDG
- GemFire Terminology
- GemFire Common Topologies
- **GemFire Use Cases**

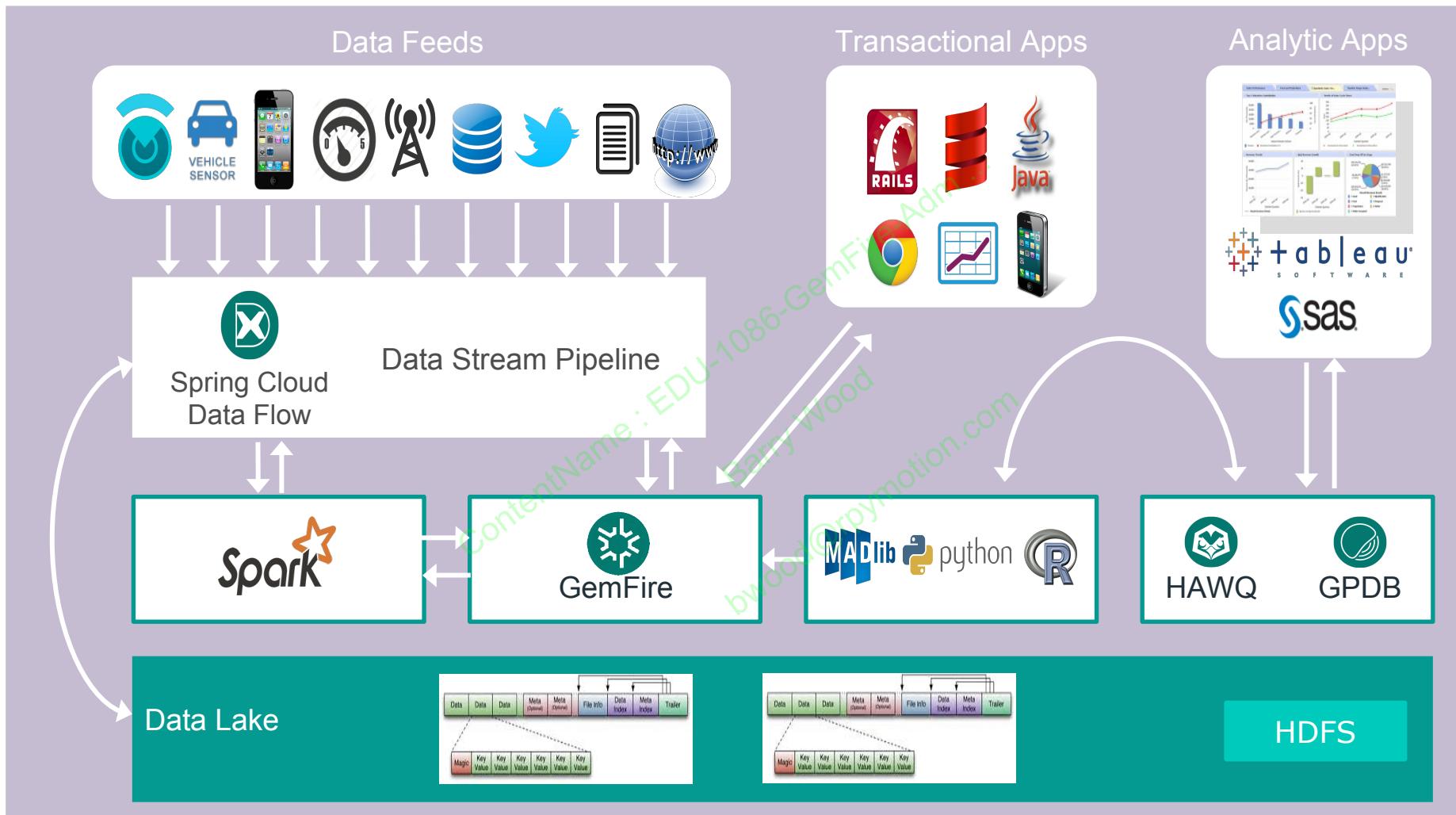
ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



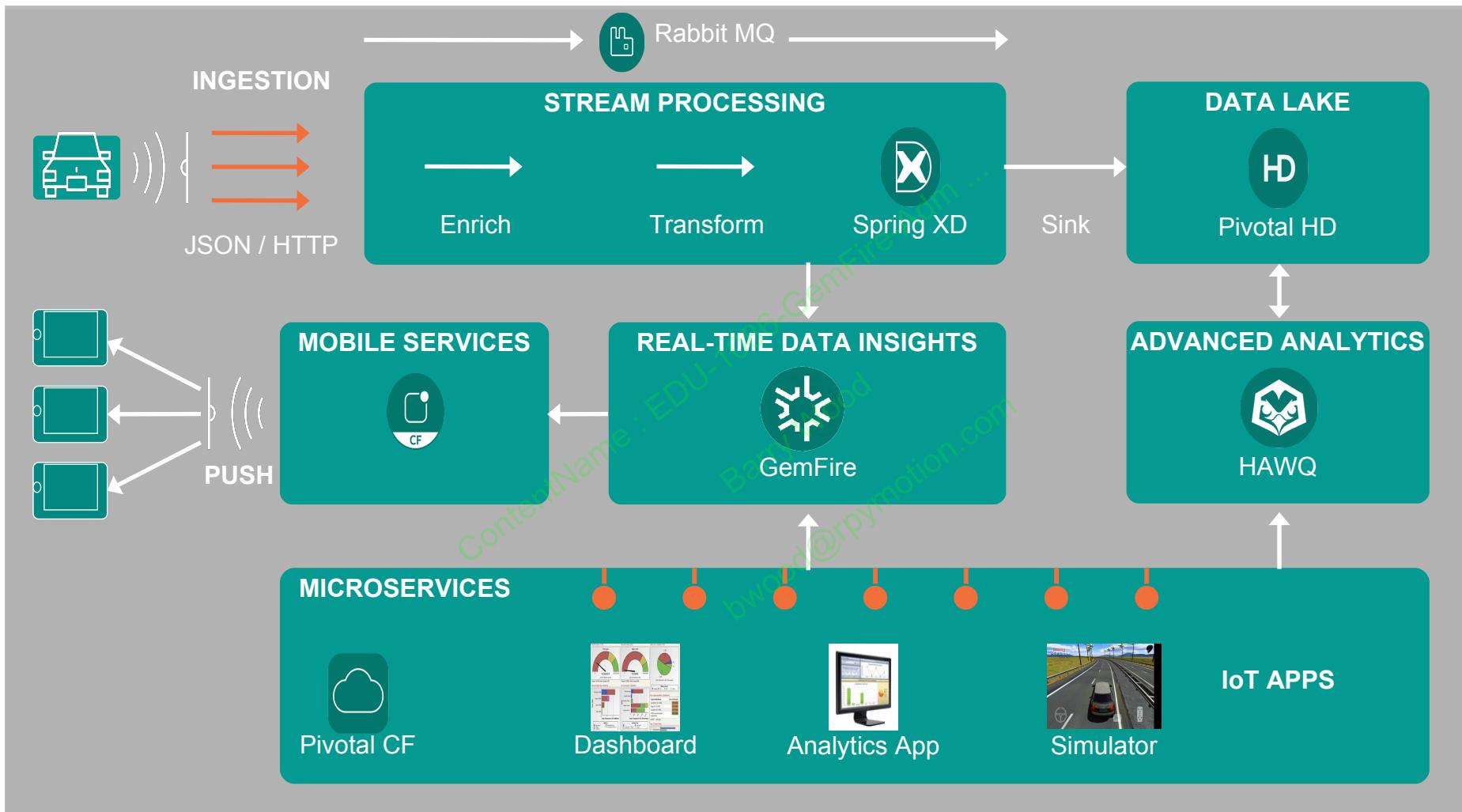
Data Streaming Reference Architecture



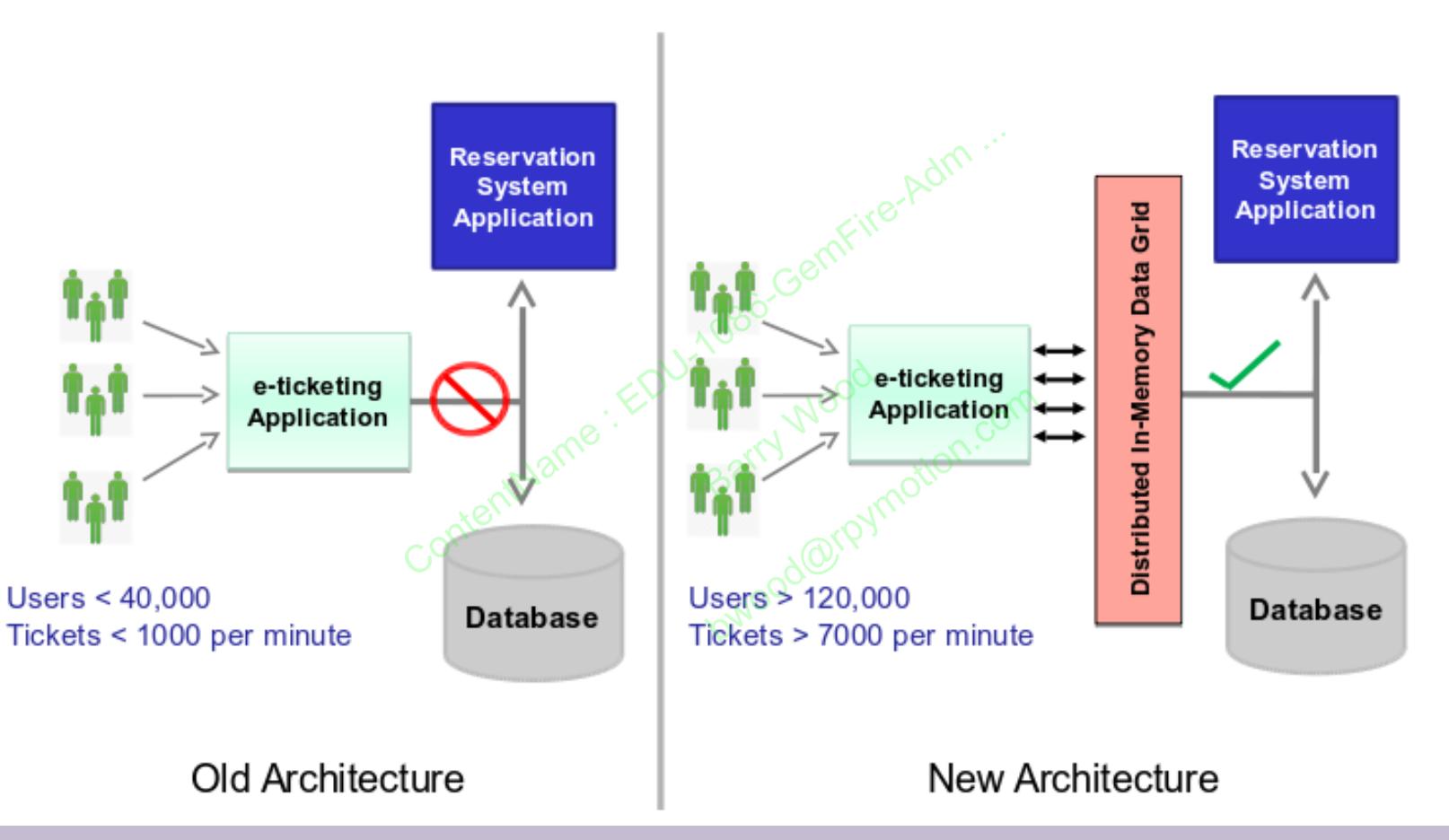
Data Streaming Reference Architecture



The Connected Car Architecture



Use Case – Online Reservation System



Use Case - Foreign Exchange Trading System

- Low-latency trade insertion
- Permanent archival of every trade
- Rapid, Event-based financial position calculation
- Distribution of updates globally
- Consistent global views of positions
- High Availability
- Disaster Recovery



Version 9.0.4c

Review of Objectives

You should be able to do the following:

- Define how GemFire functions as a IMDG
- Describe the relationship between Caches and Regions
- Describe function of Cache Server and Locator as members of a Distributed System
- Describe GemFire Client-Server and Multi-Site WAN deployment topologies
- Describe common use cases for GemFire IMDG

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

GemFire Installation & Configuration

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Objectives

After completing this lesson, you should be able to:

- Determine if a platform is supported by GemFire
- Understand which GemFire packages are required for a specific topology and platform
- Understand system requirements to run GemFire
- Describe procedure to install GemFire
- Describe GemFire configuration hierarchy
- Understand the process for performing rolling upgrades

Lesson Road Map

- Platform support and GemFire packages
- System requirements and installation
- Configuration options
- Rolling Upgrades

ContentName : EDU-1086-GemFire-Admin
Barry Wood
bwood@rpymotion.com

GemFire Supported Platforms (Tested)

- Linux/Unix
 - Red Hat RHEL 7 / CentOS 7
 - SUSE Linux Enterprise Server - SLES 11
 - Ubuntu 14.04, 16.04
 - Solaris 11
- Windows
 - Windows 2012 Server R2
 - *Windows 10 (Development)*
- Mac OS/X
 - *Mac OS 10.12 (Development)*

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

http://gemfire.docs.pivotal.io/docs-gemfire/getting_started/system

GemFire Packages

- Pivotal GemFire
 - Pivotal GemFire Zip (Windows)
 - Pivotal GemFire tar.gz (Linux & OS/X)
- Pivotal GemFire Native Client
 - Linux 64 bit
 - Windows 64 bit
 - Solaris 64 bit

ContentName : EDU-1080-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

GemFire Version Compatibility

- Peer and Cache Servers
 - Support for minor version differences
 - Example version 9.0.0, 9.0.1 and 9.1.0
- Client – Server
 - Server must be 8.2.x or equal or newer version than client
 - Example Client version 8.2.3 and Server 9.0.1
- Native Client
 - Native client 9.x is compatible with 8.2.x and 9.x server
- Multi-Site (WAN)
 - Can run major version differences
 - Example London version 8.2.3 and Dallas 9.0.1



Pivotal
GemFire

Version 9.0.4c

Lesson Road Map

- Platform support and GemFire packages
- **System requirements and installation**
- Configuration options
- Rolling Upgrades

ContentName : EDU-1086-GemFire-Administration
Barry Wood
bwood@rpymotion.com

Installation - System Requirements

- Oracle Java SE 1.8 update 92+

```
java -version
```

- JDK 1.8 supported as of 8.2.0
- Adequate file handles – on Unix/Linux, set via **ulimit**
- System clock synchronization using NTP
 - Important for useful log files
 - Accurate statistics information gathered
- Support for IPv4 and IPv6, but not in combination
- Detailed requirements see:

<http://gemfire.docs.pivotal.io/docs-gemfire/latest/getting-started/system-requirements.html>

System Preparation

- If using Zip installation, designate parent directory for GemFire Home
- Define non-root user to own GemFire home & subdirectories
- Enable SSH between servers WITHOUT password prompt
 - Cluster-wide operational scripts for start/stop/check, etc require this
- Modify OS & System level tuning
 - Open files limit (`ulimit -n`)
 - Max user process (`ulimit -u`)
 - Socket buffer size
- Ensure reverse DNS is configured

Network and Firewall Configuration

- There should be NO firewalls running between members
- For client to server firewalls, two main port settings are important
 - Locator port: the port the locator listens on
 - Configured when starting locator
 - *Example: gfsh start locator --port=41111*
 - Server Port: the port the cache server listens on
 - Configured when starting server OR in cache.xml file
 - *Example: gfsh start server --server-port=41444*



Other Firewall Considerations

Default Ports

Port Name	Default Port
HTTP (for Pulse)	7070 (as of GemFire 8.0)
Cache Server	40404
Locator	10334
Multicast	No default port (unless <code>--mcast-port</code> specified)
JMX Manager	1099
Membership Port Range	1024 to 65535



GemFire Installation

Install GemFire from a Zip File

1. Make sure a supported Java JDK is installed and `JAVA_HOME` is set
2. Create a non-root account that will run the GemFire processes. This account should own the GemFire directories
3. Unzip the archive file into the desired GemFire installation directory
4. Configure the environment for GemFire
 - Set `GEMFIRE` environment variable to point to the top level GemFire installation directory
 - Set `PATH` to include `$GEMFIRE/bin`

GemFire Installation (2)

- Unzip the file to target folder
- Set the following environment variables

- **JAVA_HOME**
- **GEMFIRE**
- **PATH**
- **CLASSPATH**: defined in lab

XX - Version number

ZZZ - Platform

```
unzip Pivotal_GemFire_XX_bXX_ZZZ.zip
```

```
GEMFIRE=<GemFire root installation directory>
```

```
JAVA_HOME=<Java JDK root installation directory>
```

```
PATH=$JAVA_HOME/bin:$GEMFIRE/bin:$PATH
```

Lesson Road Map

- Platform support and GemFire packages
- System requirements and installation
- **Configuration options**
- Rolling Upgrades

ContentName : EDU-1086-GemFire-Admin
Barry Wood
bwood@rpymotion.com

Distributed System Configuration

- Configuration of system member:
 - Connection and communication behavior
 - Logging, statistics, and security settings
- Three ways to set properties with priority:
 - 1) Via command line arguments from `gfsh` (highest)
 - 2) Via `gemfire.properties` file
 - 3) Through Java API (lowest)
- See Reference section of GemFire User's Guide for detailed list of properties

Setting system startup parameters

gemfire.properties

```
log-level=config  
statistic-sampling-enabled=true  
statistic-sample-rate=1000  
archive-file-size-limit=100  
statistic-archive-file=/Users/myMac/myApp/stats1007.gfs  
locators=localhost[41111],localhost[41112]...  
mcast-port=0  
cache-xml-file=myCache.xml
```

Use **gemfire.properties** for:

- Configuring individual system member communications
- Interaction with the distributed system

Cache Configuration for Servers

- Configuration of cache for cache servers
- Three ways to define cache configuration with priority:
 - 1) Via command line arguments from `gfsh` (highest)
 - Create regions
 - Destroy Regions
 - 2) Via `cache.xml` file - Initial cache configuration
 - 3) Through Java API - Dynamic configuration (lowest)

Configure cache in the cache.xml file

- Note the switch to Schemas and rebasing on Geode

```
<?xml version="1.0" encoding="UTF-8" ?>
<cache
    xmlns="http://geode.apache.org/schema/cache"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://geode.apache.org/schema/cache
                        http://geode.apache.org/schema/cache/cache-1.0.xsd"
    version="1.0">

    <region name="Customer" refid="PARTITION"/>

    <region name="Inventory">
        <region-attributes refid="REPLICATE"/>
    </region>
</cache>
```

Cache Configuration for Clients

- Configuration of client local cache
- Two ways to define cache configuration with priority:
 - 1) Via `clientCache.xml` file - Initial cache configuration
 - 2) Through Java API - Dynamic configuration

ContentName : EDU-108C GemFire Admin
Barry Wood
bwood@rpymotion.com

Configure Client using `clientCache.xml`

- Client cache can also use either older DTD or Schemas

```
<client-cache>
  <pool name="locatorPool">
    <locator host="host3" port="41111"/>
  </pool>
  <region name="Inventory">
    <region-attributes refid="PROXY" pool-name="locatorPool"/>
  </region>
  <region name="clientsPrivateR">
    <region-attributes refid="LOCAL"/>
  </region>
</client-cache>
```

Lesson Road Map

- Platform support and GemFire packages
- System requirements and installation
- Configuration options
- **Rolling Upgrades – Not Available for 8.x -> 9.x**

ContentName : EDU-103-GemFire-Admin ...
Barry Wood
bwood@rpymotion.com

Rolling Upgrades – Overview

- Better support for rolling upgrades between major versions added in GemFire 8.0
 - Multiple versions allowed between members
 - *Example:* server1 = 8.0, server2 & server3 = 8.1
- Applies to peer members within a single distributed system
 - Other rules apply between client & server and Multi-Site configurations
 - Recall Installation section about version compatibilities



Version 9.0.4c

Preparing for Upgrade

- Ensure all members meet minimum version
- Ensure partitioned regions have redundancy enabled
 - Check configuration
 - Also check no buckets without redundancy (more later)
- Back up data prior to upgrade
 - Can perform full backup if region is persistent
 - Snapshots can be used during slower periods
- Be sure all pre-requisites verified and pre-steps performed
 - Once started, you can't go back
 - Take the extra time to verify
 - Make sure you have multiple locators running beforehand

Performing the Upgrade

- Pick a low activity time
- Start with locators
 - Do one at a time
 - All must be done before moving to servers
 - Once completed, no older members may be re-started or added
- Once begun avoid region operations
 - Includes rebalance, region creation or destruction
 - Also do not alter region attributes via gfsh until complete
- Rebalance if **start-recovery-delay=-1** (disabled)
 - Required for each member once member re-started
 - If **start-recovery-delay** is low, may need to wait for recovery of redundancy



Locator Upgrade Procedure – Part 1

1. First install new version in its own directory (both versions must be available for now)
2. Open 2 terminal windows – one will be for the older version and the other for the newer version
3. Export existing locator's configuration
 - Connect to existing locator's JMX Manager
 - Export configuration

```
gfsh> export config --member=locatorName --dir=target_dir
```

 - Exports locator's config files to a folder for upgrade
4. Stop the locator



Locator Upgrade Procedure – Part 2

5. Set **GEMFIRE** path to new GemFire installation in terminal 2
6. Start **gfsh** and verify it's the new version
7. Restart locator with exported configuration

```
gfsh> start locator --name=locator_name --dir=locator_export_dir
```

8. Verify correct startup
 - View log file
 - Ensure gfsh is connected to JMX Manager
 - Ensure start using same ports as the older version
9. Repeat for other locators
 - Do NOT begin servers until all locators are upgraded
 - Verify locators are correctly running via logs and gfsh

Server Upgrade Procedure

- Similarly, use two terminals
 - Since startup must be performed locally, open on candidate machine
 - Since each cache server may reside on a different machine, installation process will need to be performed on each machine
- A similar export process must be performed for each member
 - Saves member's cache.xml and properties files
 - May not be necessary if Cluster Configuration Service used
 - You may want to capture a snapshot of a region (more later)
- Basic process is the same
 - Complete one member at a time
 - Verify successful upgrade from log file

After the Upgrade

- If **recovery-delay** was set, you may need to perform region rebalance one final time
- Verify all members are up and running
- Verify for partitioned regions there are 0 buckets without redundancy

```
gfsh> show metrics --region=/Customer --member=server1
```

- Must be done for each member hosting partitioned region(s)
- Look for attribute: **numBucketsWithoutRedundancy**



Lab – Installation Configuration

In this lab, you will:

1. Install GemFire and the lab environment
2. Finalize lab environment configuration

ContentName : EDU-1086-GemFire-Adm
Barry Wood
bwood@rpymotion.com

Review of Objectives

You should be able to do the following:

- Define GemFire packages required for Client-Server and Multi-site WAN deployments
- Describe GemFire installation procedure
- Describe purpose of GemFire configuration files
- Successfully perform rolling upgrades

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

Client/Server Configuration

The GemFire Distributed System

Content source: EDU-1086-GemFire-Admin
Author: Bryan Wood
Email: bwood@rpymotion.com

Objectives

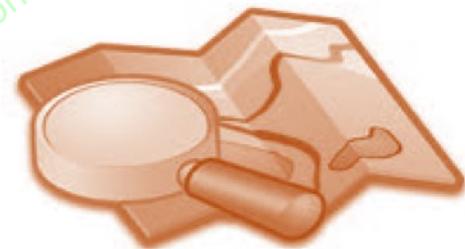
After this lesson, you should be able to do the following:

- Describe the components of a GemFire distributed system
- Describe the GemFire client-server architecture
- Explain logical grouping of servers/members
- Describe client-server data flow
- Describe basic usage of gfsh

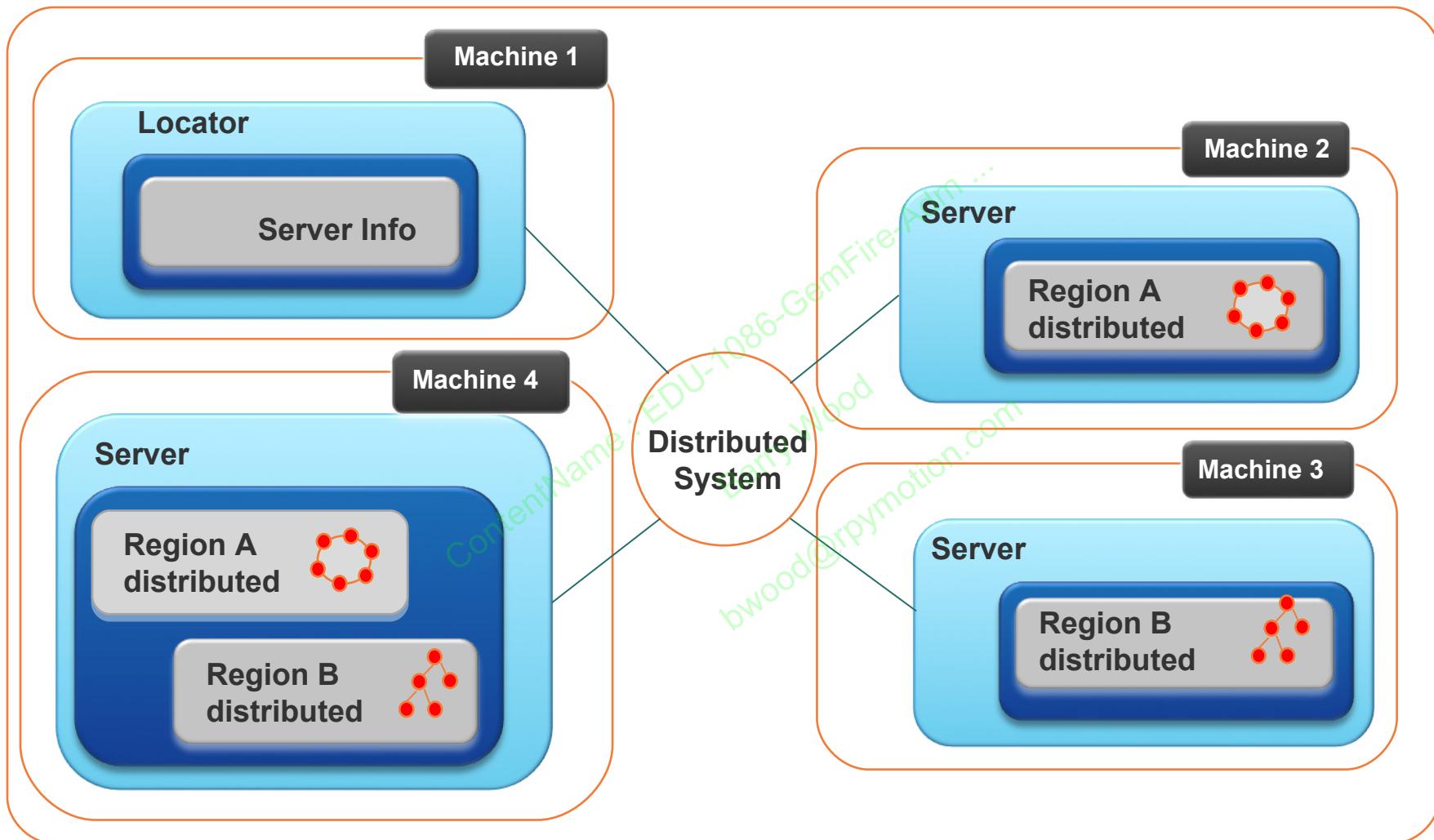
Lesson Road Map

- Client-Server
- Configuring the Server
- Configuring the Client
- Introduction to gfsh

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

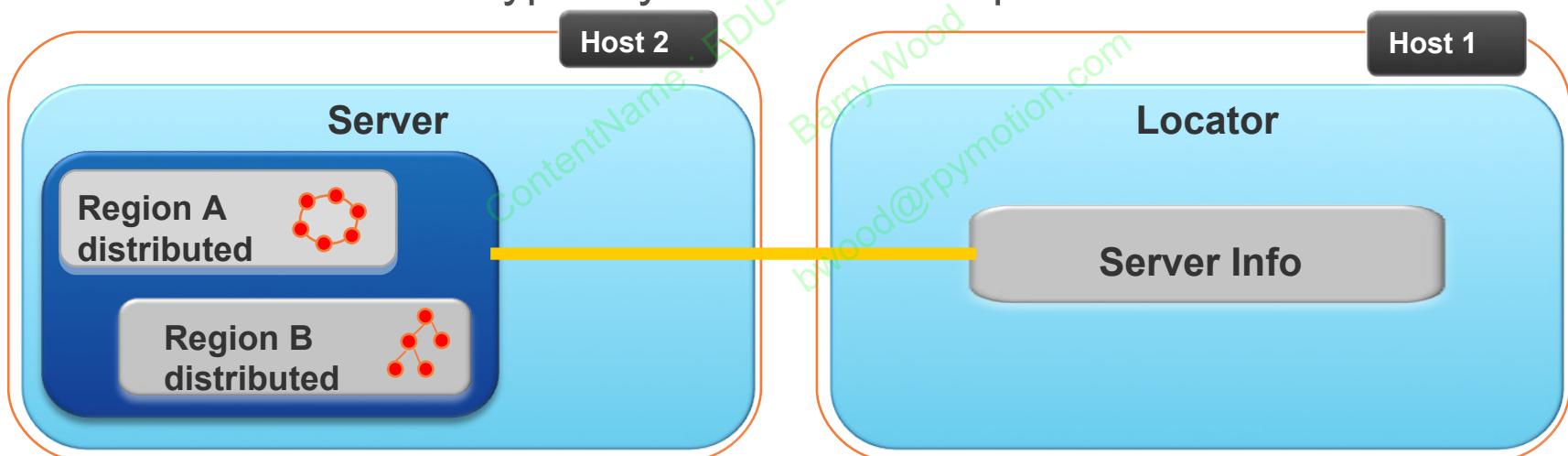


GemFire Distributed System



Distributed System: Locators

- Peer discovery by the servers
- Server discovery by the clients
- Specified in a Pool instance in the client applications
- Give dynamic server information to clients
- Provide server connection load rebalancing and fault tolerance
- NOTE: Should be typically 3 locators in a production cluster



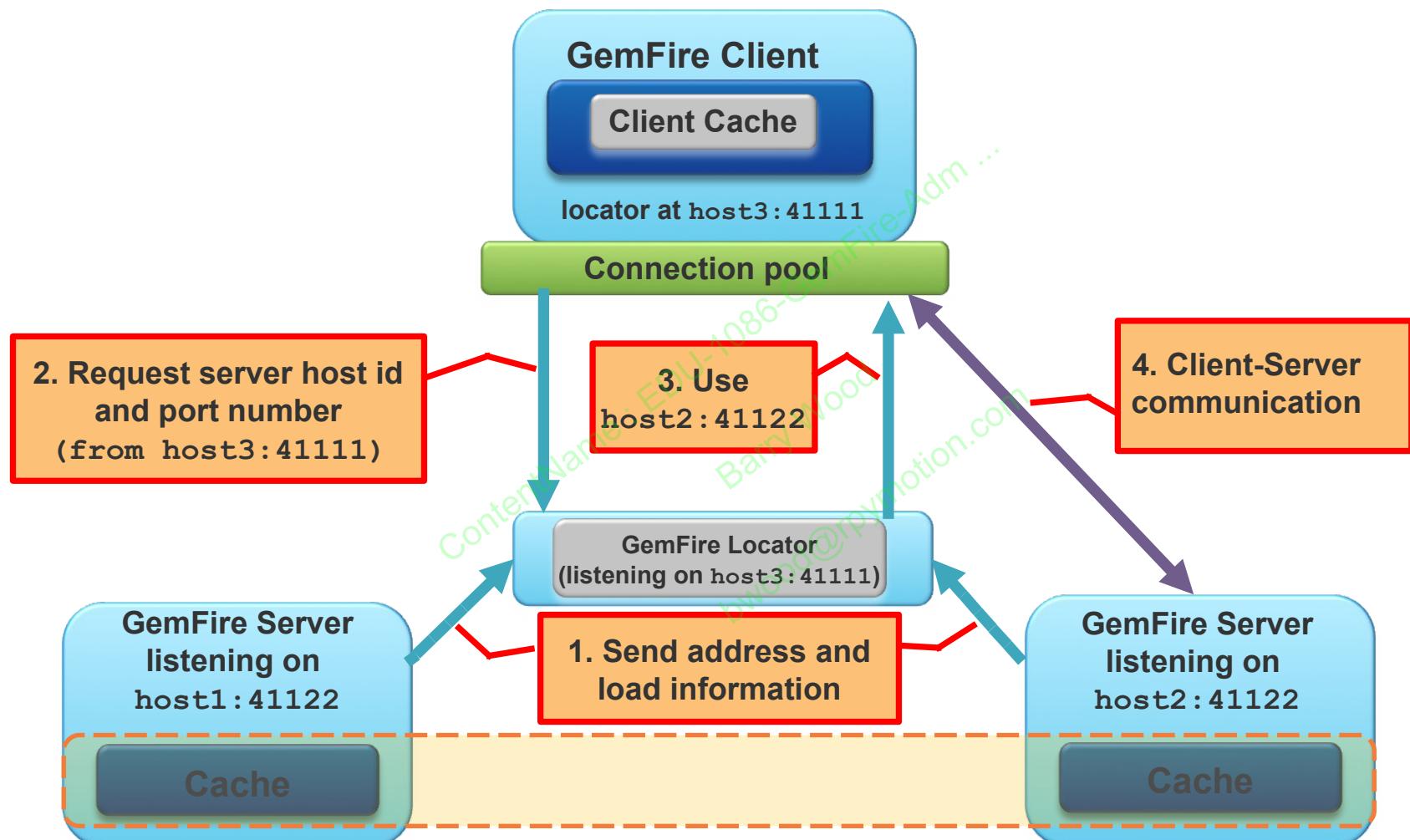
Distributed System: Member Discovery

The discovery process:

- Is performed by the locator and server processes
- Uses TCP for discovery (lookup)
- Updates membership dynamically

ContentName : EDU-100
Barry Wood
bwood@rpymotion.com

Client-Server Deployment Example



Lesson Road Map

- Client-Server
- **Configuring the Server**
- Configuring the Client
- Introduction to gfsh

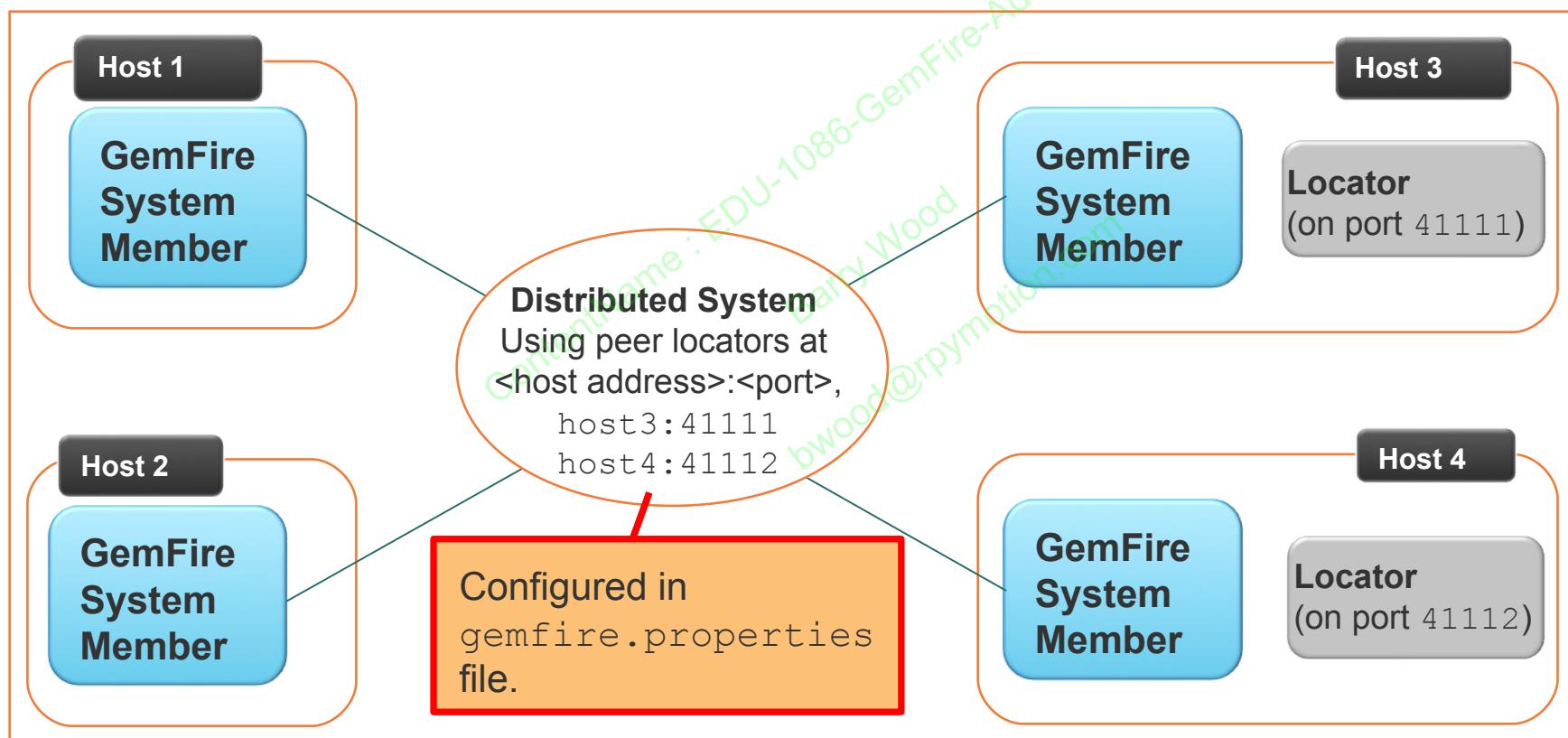
ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



Distributed System: Locators for Peers

Members (Locators and Servers) can be configured in three key ways:

- Cluster configuration (via gfsh commands)
- `gemfire.properties` file (via `locators` property)
- `gfsh` command line (via `--locators` argument)



Configuration Options

Multiple ways to configure GemFire Locators and Servers

- Via gfsh & Cluster Configuration Service.
 - Newest and most preferred method
 - Commands issued via gfsh stored centrally
- Via gemfire.properties file
 - Common properties can be set
 - Can override cluster configuration
- Via Cache XML file
 - Legacy way to configure cache features (creating regions)
 - Can be used to override cluster configuration

Basic Configuration

More on properties files and cache.xml file

- Configuring common settings in a gemfire.properties configuration file:

```
locators=host3[41111]  
log-level=warning
```

- Cache features such as regions configured in cache.xml file

```
<cache>  
  <region name="Inventory"/>  
</cache>
```

Member Groups

Organizing Peers into logical member groups enables:

- Splitting data management between members
- Altering a subset of configuration properties for a specific member or members
- Deploying or undeploying JAR application files on all members in a group
- Executing functions on all members of a specific group
- Defining a subset of members a client may connect to



Configuring Member Groups

- Set in the member's `gemfire.properties`:

```
groups=gold,chicago
```

```
...
```

[gemfire.properties](#)

- Or using `gfsh`:

```
gfsh> start server --name=server1 --group=gold
```

or

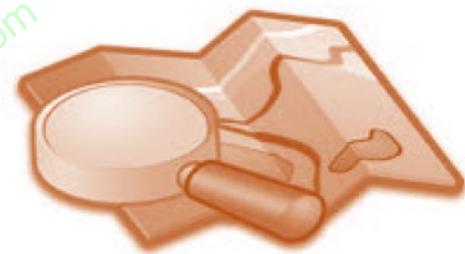
```
gfsh> start server --name=server1 --properties-file=...
```

```
gfsh> deploy --group=gold --jar=./myJar.jar
```

Lesson Road Map

- Client-Server
- Configuring the Server
- **Configuring the Client**
- Introduction to gfsh

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

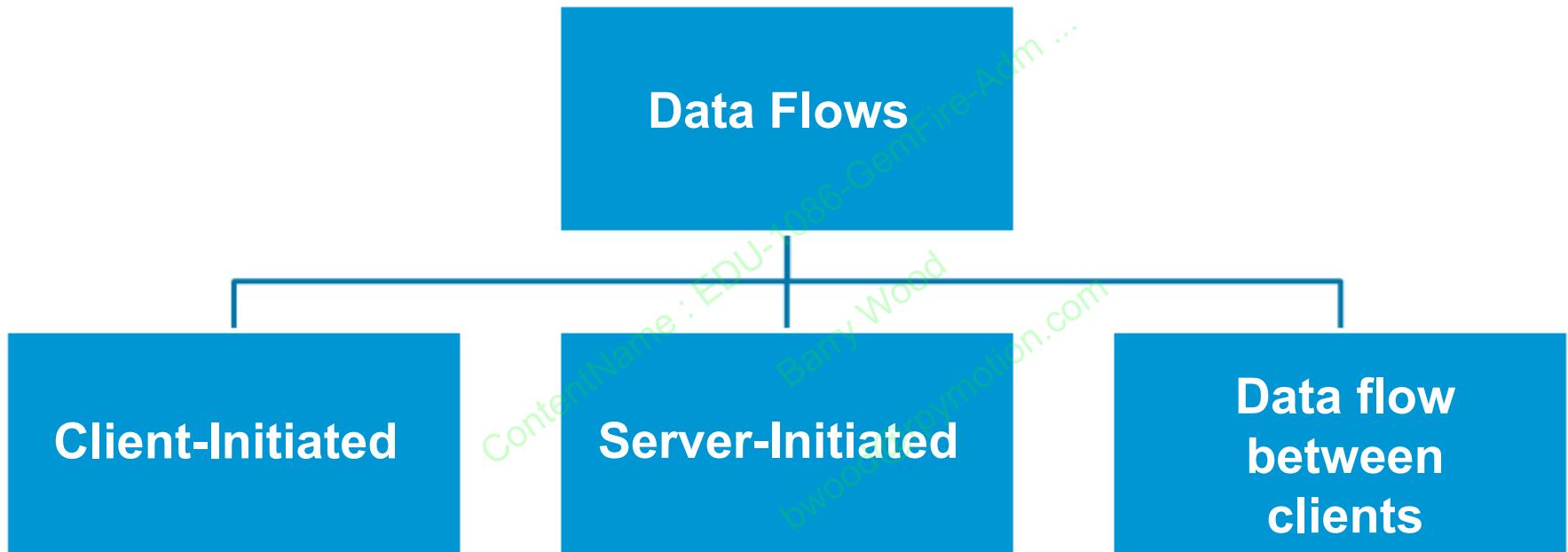


Basic Configuration

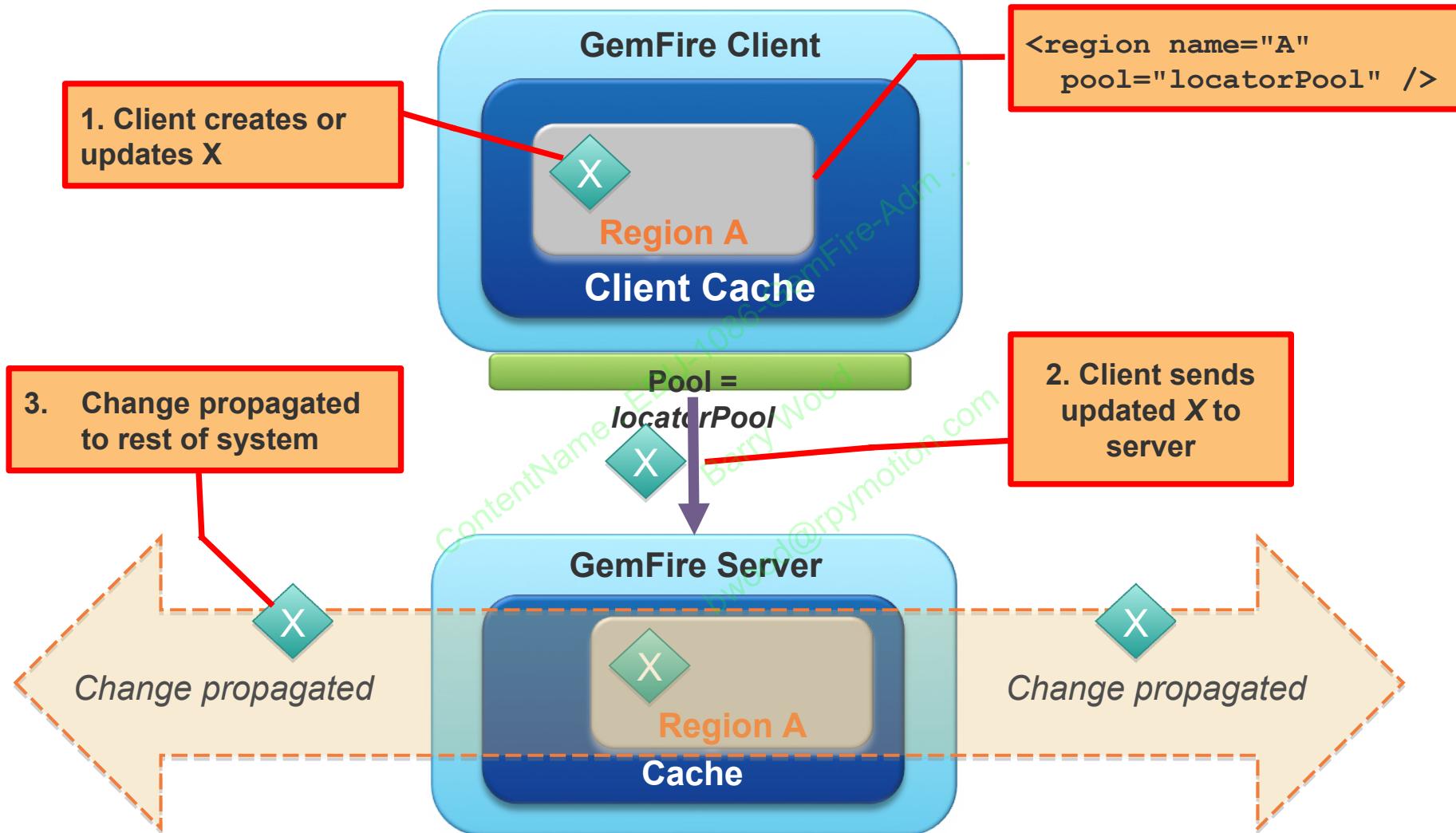
- Client's cache.xml file contains:
 - connection pool configuration
 - region-attributes configuration

```
<client-cache>
    <pool name="locatorPool">
        <locator host="host3" port="41111"/>
    </pool>
    <region name="Inventory">
        <region-attributes refid="PROXY" pool-name="locatorPool"/>
    </region>
</client-cache>
```

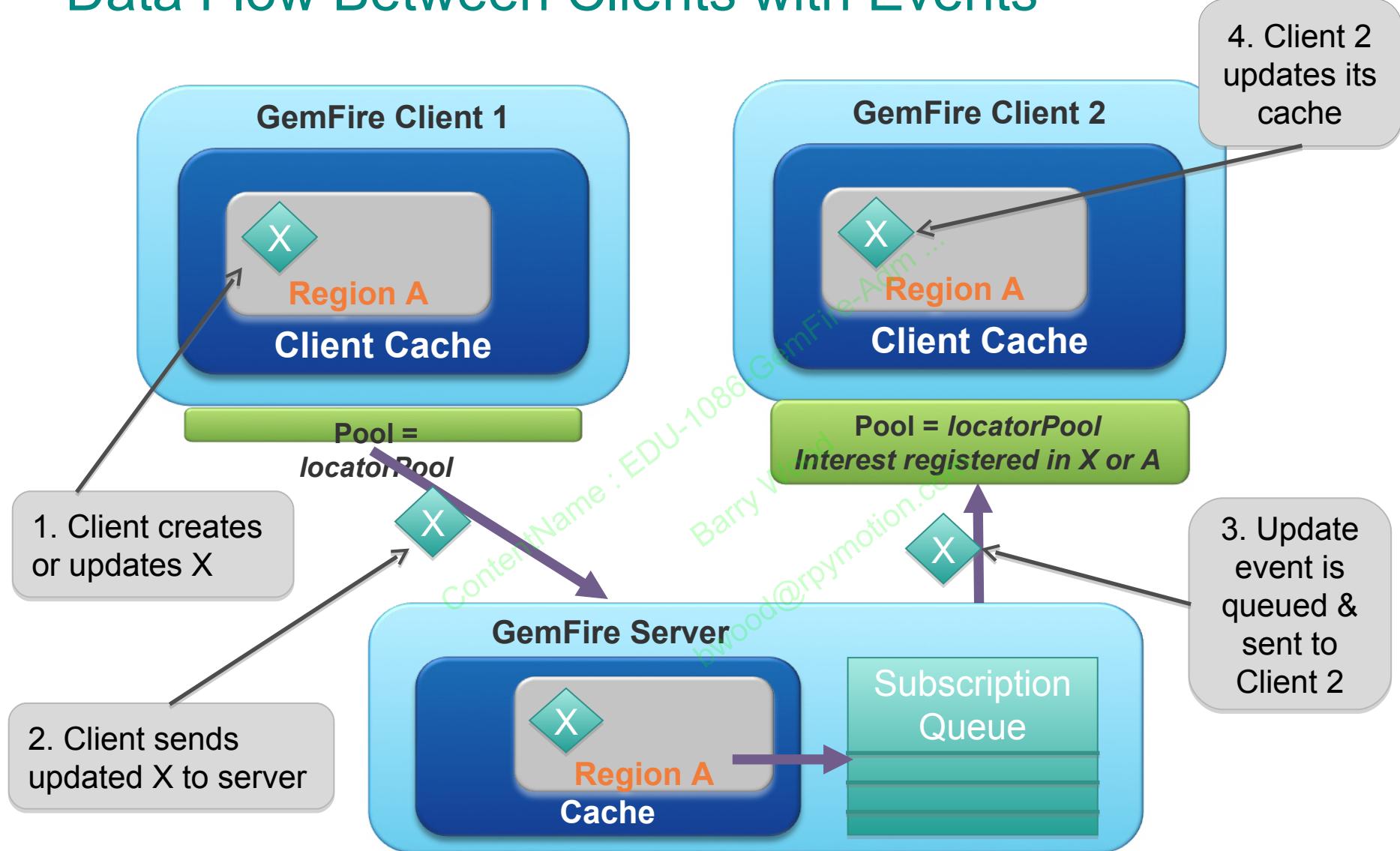
Client-Server Data Flow



Client-Server Data Flow



Data Flow Between Clients with Events



Lesson Road Map

- Client-Server
- Configuring the Server
- Configuring the Client
- **Introduction to gfsh**

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



gfsh – GemFire Shell

- Command line tool that ships with GemFire
- Unix-like shell command interface
- Browse and edit data in GemFire
- Start/stop distributed system members
- Say “gee-fish”

ContentName: EDU-100G GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

gfsh: Starting / Stopping Locators

- Start locators using the gfsh command-line utility:

```
gfsh> start locator  
  --name=locator1  
  --port=41111  
  --dir=locator  
  --properties-file=gemfire.properties  
  --initial-heap=1024m  
  --max-heap=1024m
```

- Stopping a locator:

```
gfsh> stop locator --name=locator1
```

gfsh: Starting / Stopping Servers

- Start servers through the gfsh command-line utility:

```
gfsh> start server  
      --name=server1  
      --locators=host3[41111]  
      --server-port=41122  
      --dir=server1  
      --properties-file=gemfire.properties  
      --initial-heap=2048m  
      --max-heap=2048m
```

Set to zero, locator will choose port dynamically for the server

- Stopping a server:

```
gfsh> stop server --name=server1
```

gfsh – Misc Commands

- Connection via a locator (must use if re-launching **gfsh**):

```
gfsh> connect --locator=localhost[41111]
```

- List all regions:

```
gfsh> list regions
```

- List the partition and bucket information:

```
gfsh> show metrics --region=Customer  
          --member=server1  
          --categories=partition
```

gfsh commands are explained in detail in the labs!

Lab – Configure Client Server

In this lab, you will:

1. Configure & start locator
2. Configure & start servers
3. Configure client
4. Execute test client application
5. Shutdown servers and locator

Content home: <http://www.gemfire-administrator.com>
bwood@rpymotion.com
Copyright 2017 by Barry Wood

Review of Objectives

You should be able to do the following:

- Describe the components of a GemFire distributed system
- Describe the GemFire client-server architecture
- Explain reason for grouping of servers members
- Describe client-server data flow
- Describe basic usage of gfsh

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

Cluster Configuration Service

Managing a GemFire cluster using *gfsh*

Content from: EDU-1086-GemFire-Admin ...
Darryn Wood
bwood@rpymc.com

Objectives

After completing this lesson, you should be able to:

- Understand Cluster Configuration Service and how it's used
- Use gfsh to manage cluster configuration
- Export and import cluster definitions
- Be aware of where configuration data is stored

Lesson Roadmap

- **Cluster Configuration Overview**
- gfsh and Cluster Configuration
- File storage and troubleshooting

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Cluster Configuration Service Overview

- New feature added in GemFire 8.0
- Allows for centralized storage and management of server configuration
- Allows gfh to become the primary mechanism to manage the distributed system
- No more per-server cache.xml files!
- *Note:* Does NOT enable centralized server startup (yet)

Cluster Configuration Service Features

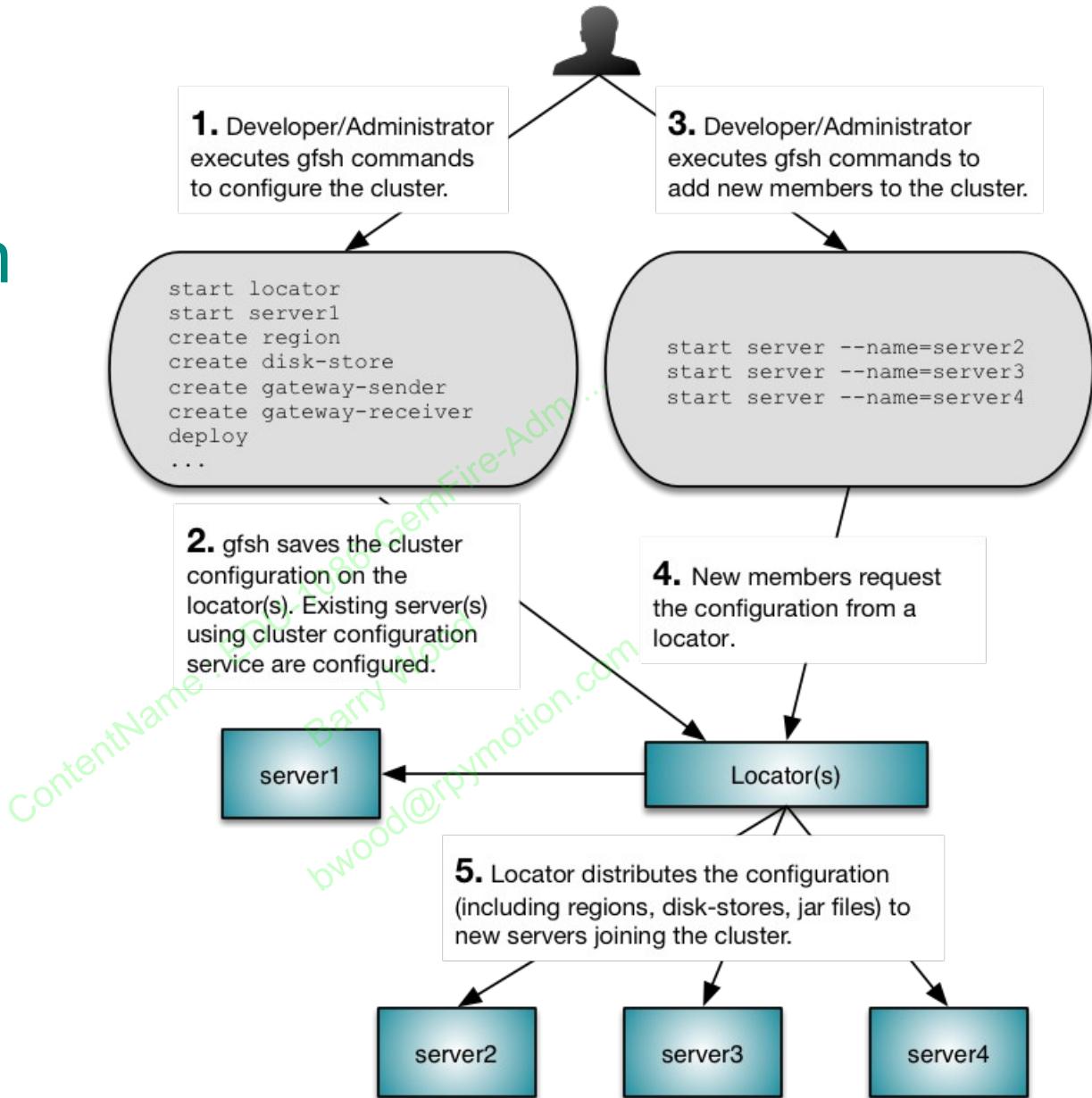
- Save the configuration for an entire Pivotal GemFire cluster
- Restart members using a previously-saved configuration
- Export a configuration from a development environment and migrate that configuration to create a testing or production system
- Start additional servers without having to configure each server separately
- Target configuration to certain servers via member groups



Enabling Cluster Configuration Service

- Locator process MUST be run as a separate process
 - Locator cannot be run co-located in the same process as a cache server
 - Can run on the same machine but must be a separate process
- Cluster configuration is enabled by default
 - Disable using `--enable-cluster-configuration=false`
 - Servers also have this enabled by default but can have it disabled using the same option
- Existing configuration files can be used
 - Must place files in locator's working directory under `cluster_config/cluster` directory
 - Follows a naming convention (`cache.xml` → `cluster.xml`)
 - Use `--load-cluster-configuration-from-dir=true`

How Cluster Configuration Works



Lesson Roadmap

- Cluster Configuration Overview
- **gfsh and Cluster Configuration**
- File storage and troubleshooting

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Using gfsh with Cluster Configuration Service

- Many **gfsh** commands now affect the cluster's configuration via the service
 - **configure pdx**
 - **create/destroy region**
 - **alter region/runtime**
 - **create disk-store**
 - **deploy/undeploy jar**
- Not everything can be fully configured via cluster config
 - Registering functions with parameters
 - Registering event handlers with parameters
 - Configuring eviction
 - More



Example: Managing Regions

```
gfsh> start locator --name=locator1 --port=411111  
  
gfsh> start server --name=server1 --server-port=44211 ...  
  
gfsh> start server --name=server2 --server-port=44212 ...  
  
gfsh>create region --name=Customer --type=REPLICATE  
Member | Status  
----- | -----  
server2 | Region "/Customer" created on "server2"  
server1 | Region "/Customer" created on "server1"  
  
gfsh>create region --name=BookMaster --type=REPLICATE  
Member | Status  
----- | -----  
server1 | Region "/BookMaster" created on "server1"  
server2 | Region "/BookMaster" created on "server2"
```



Deploying JAR files

- Deploying JAR files is not specifically a Cluster Configuration feature

```
gfsh> deploy -jar=myFunctions.jar
```

- With Cluster Configuration Service, JAR files are stored with other files in Cluster Configuration repository
- When new servers start up, JAR files are deployed to new members along with configuration info



Exporting Cluster Configuration

- Used for copying cluster configuration from one distributed system to another
 - *Example:* from dev → production or visa versa
- Entire configuration packaged in zip file and copied to specified directory

```
gfsh> export cluster-configuration  
      --zip-file-name=myConfig.zip  
      --dir=/home/gemfire/configs
```

- Only configuration specified using **gfsh** will be managed and copied



Importing Cluster Configuration

- Locator(s) must be running and gfsh connected to JMX Manager
- All cache servers must be stopped

```
gfsh> import cluster-configuration  
      --zip-file-name=myConfig.zip  
      --dir=/home/gemfire/configs
```

- Once import is complete, any servers started receive the imported configuration



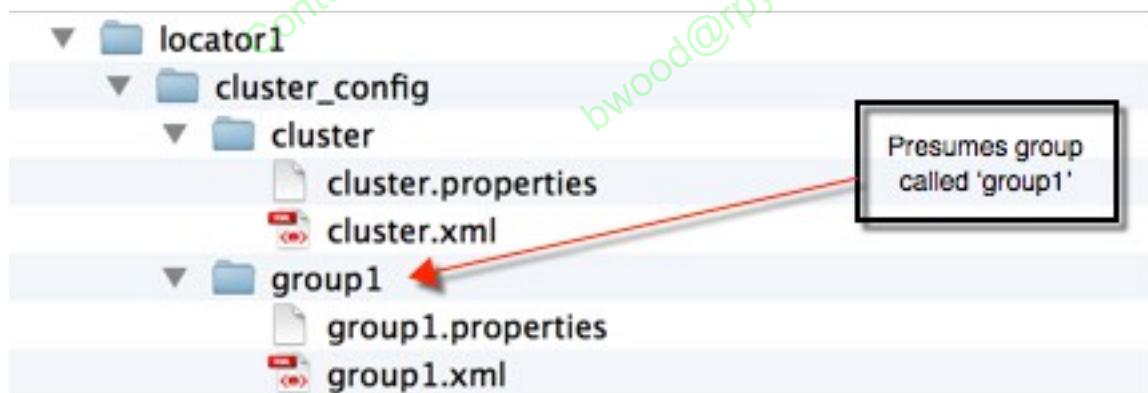
Lesson Roadmap

- Cluster Configuration Overview
- gfsh and Cluster Configuration
- **File storage and troubleshooting**

ContentName : EDU-1086-GemFire-Admin ...
Barry Wood
bwood@rpymotion.com

Configuration File Storage

- All files are stored by default in locator directory under folder **cluster_config**
 - Cluster-wide configuration stored under folder ‘**cluster**’
 - Group-specific configuration stored under folder with name of group
- Name of folder and base name for xml and properties files are the same



Altering Storage Configuration

- Change location of cluster configuration directory

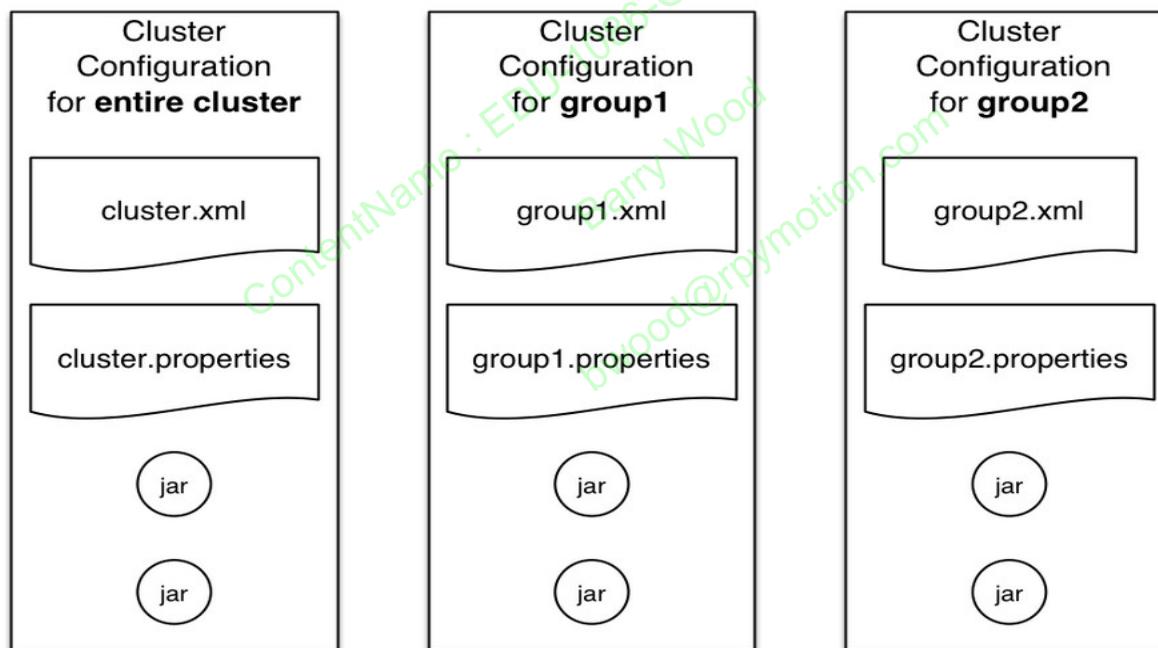
```
gfsh> start locator --name=locator1  
      --cluster-config-dir=myCustomDir
```

- Force load of configuration from directory
 - By default, configuration is read from Locator's internal persistent region
 - Override by adding
`--load-from-cluster-configuration-dir=true`
- Disabling use of cluster configuration service

```
gfsh> start locator --name=locator1  
      --enable-cluster-configuration=false
```

How Configuration Files are Applied

- Order of priority in the how configuration is applied
 - Cluster: Applied to all servers that are started
 - Group: Applied to servers that members of group
 - Server: Any server-specific configurations applied



Troubleshooting Locator & Cluster Configuration Service

- Verify service is running on locator start

```
gfsh> start locator --name=locator1 --port=41111  
... general output ...  
Cluster configuration service is up and running.
```

- If you suspect cluster configuration service may not be running

```
gfsh> status cluster-config-service  
Status of shared configuration on locators  
  
Name | Status  
----- | -----  
locator1 | RUNNING
```

- You may see ‘WAITING’ instead
- Try ‘status locator --name=locator1’ for more detail

Troubleshooting Servers

- You may see an exception:

ClusterConfigurationNotAvailableException

- May mean service wasn't started in locator
 - Troubleshoot locator and cluster configuration service

- Other server troubleshooting

- Ensure logging is set to config or fine
 - Captures origin of configuration information
 - Server fails to start
 - Could be due to clashes in local cache.xml vs cluster configuration
 - Configuration of server incorrect

Lab: Cluster Configuration Service

In this lab, you will:

1. Create cluster-wide region definitions
2. Manage servers using cluster configuration
3. Export cluster configuration
4. Examine file storage

ContentName : EDU-1086-GemFire-Admin ...
Barry Wood
bwood@rpymotion.com

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

Partitioned & Replicated Regions

How Gemfire Manages Distributed Data

Content from: EDU-HPS-GemFire-Admin ...
By Bryan Wood
bwood@rpymotion.com

Objectives

After completing this lesson, you should be able to:

- Describe and configure partitioned regions
- In partitioned regions, describe how data entries are placed in buckets based on hashing
- Describe data co-location between regions
- Describe and configure replicated regions

Lesson Road Map

- **Partitioned Regions**
- Hardware Failure Scenarios
- Redundancy Zones and Virtualization
- Data Co-located between Regions
- Replicated Regions
- Region Compression
- Quick Java Overview

Content Name : EDU-103_GemFire_Advanced
Barry Wood
bwood@rpymotion.com

Partitioned Regions

The dataset is partitioned across multiple members (nodes)

- Each member contains a subset of the data
- Each member may also contain *redundant* copies of data
- Accommodates data sets that are **too large** to hold on a single member



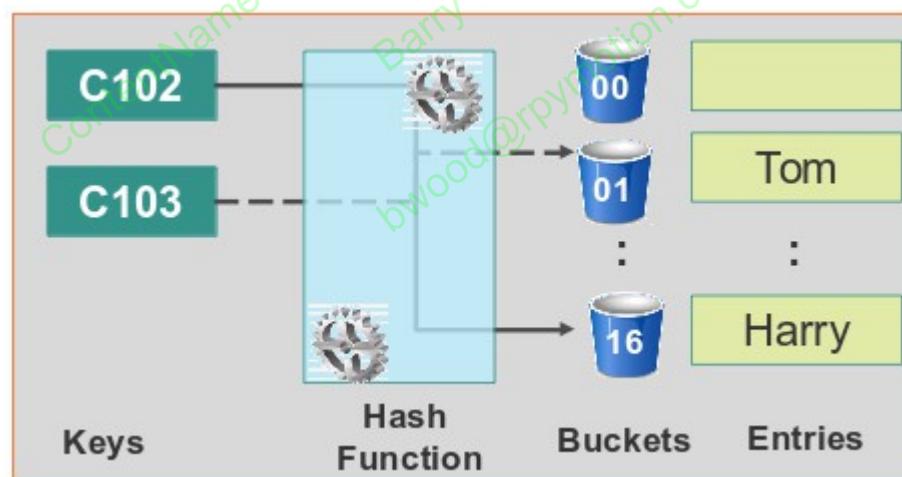
Version 9.0.4c

When do I need partitioning?

- **Large Data Sets**
 - Store data sets that are too large to fit in a single Java process or single cache member
 - Suitable for 10's of TB
- **High availability**
 - Data in partitioned regions can also be replicated across nodes
 - Redundancy prevents data loss on cache member failure
- **Scalability**
 - Add more nodes, at runtime, to expand data capacity
 - Can give near linear scalability

Hashing and Buckets

- Objects stored as Key/Value pairs in region
- Data entries are stored in “buckets”
 - Based on the hashing of the entry key
 - `key.hashCode() modulo numberOfBuckets -> bucket`
 - Use *prime number* as the number of buckets for optimal distribution



Partitioning

- Uses a function or algorithm to map object data to some representative integer value
 - A *hash value* or *hash code*
 - Used to narrow down our search when looking for an item in a Map
- Multiple values are expected to have the same “hash”
 - Entries within a bucket are traversed in order to find item
 - A good hash code should provide *uniform* distribution to reduce number of keys in a given bucket
 - *Even* distribution across the platform is the goal
- Think “index”

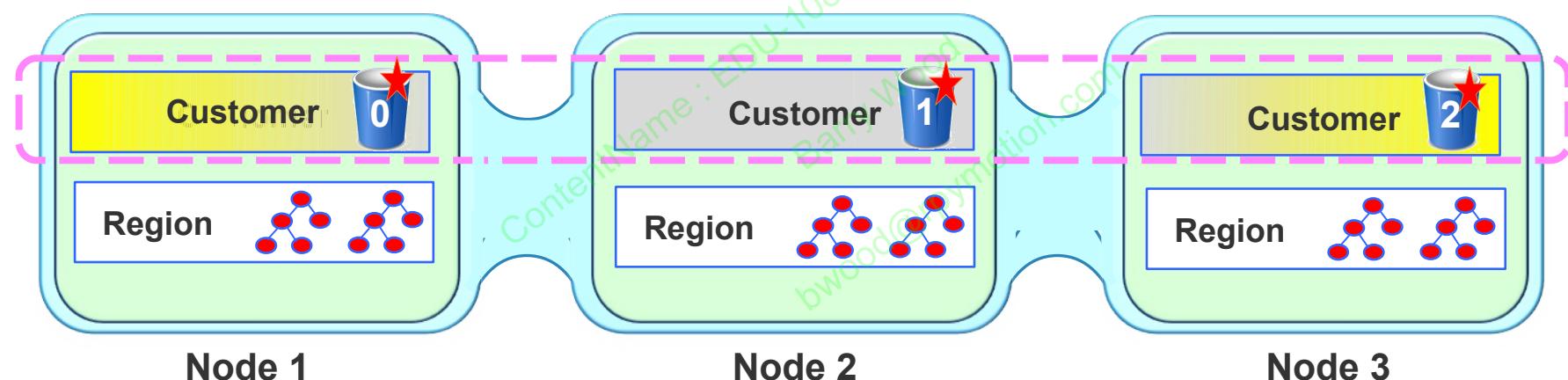
Partitioned Regions and Buckets

- Use hashing to sort keys into buckets
- Buckets are distributed across nodes
- Default number of buckets is 113 (prime)



Basic Partitioning

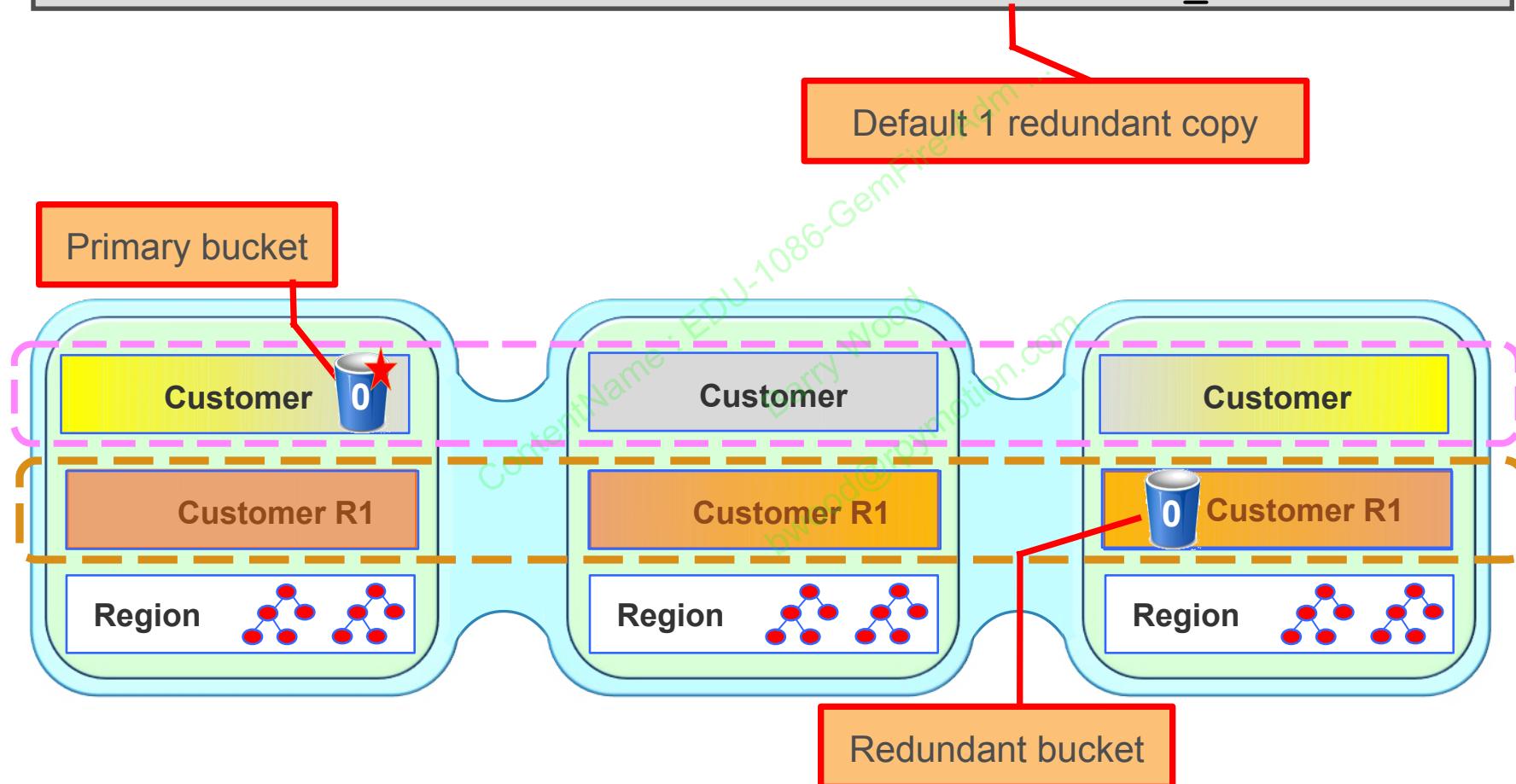
```
gfsh> create region --name=Customer --type=PARTITION
```



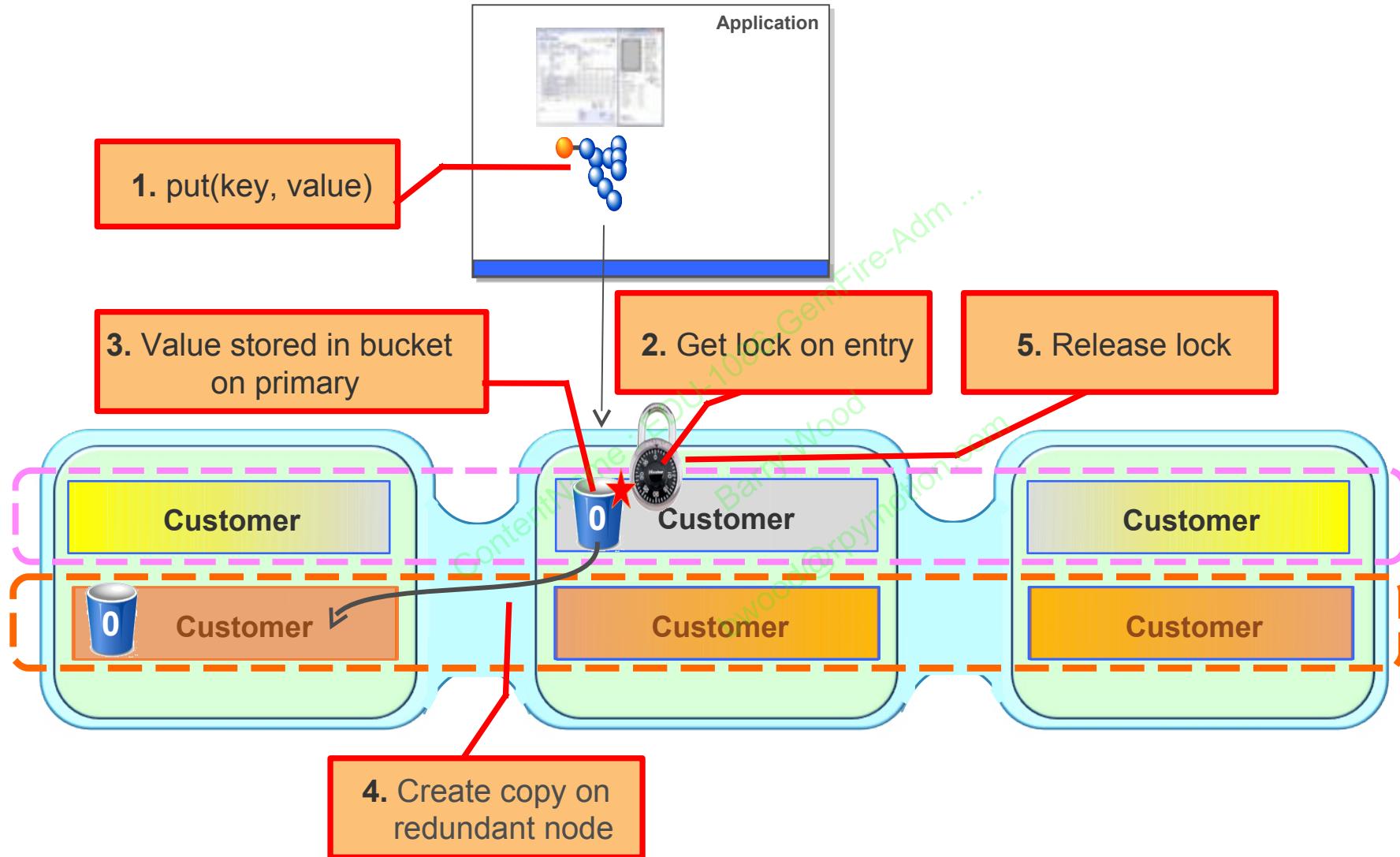
All updates are synchronous, and consistency is guaranteed

Partition Redundancy

```
gfsh> create region --name=Customer --type=PARTITION_REDUNDANT
```



Putting a New Entry in a Partitioned Region



Partitioned Region Shortcuts

- Example: Partitioned Region with 0 redundant copies

```
gfsh> create region --name=Customer --type=PARTITION
```

- Example: Partitioned Region with 1 redundant copy

```
gfsh> create region --name=Customer --type=PARTITION_REDUNDANT
```

- Example: Partitioned Region with 2 redundant copies

```
gfsh> create region --name=Customer --type=PARTITION  
--redundant-copies=2
```

No shortcut for > 1 redundant copy

Caveats of Partitioned Regions

- Transaction data must be co-located on the same server
- Synchronous redundant copy updates, maintain data consistency
- The larger the ratio of buckets to data entries, the more evenly the load can be spread across the members
- Number of buckets $20-50 * \text{number of servers}$ you expect to have for the region
- Guidelines:
 - More entries than buckets
 - Use a prime number
 - Look for even distribution – important to handle member failures

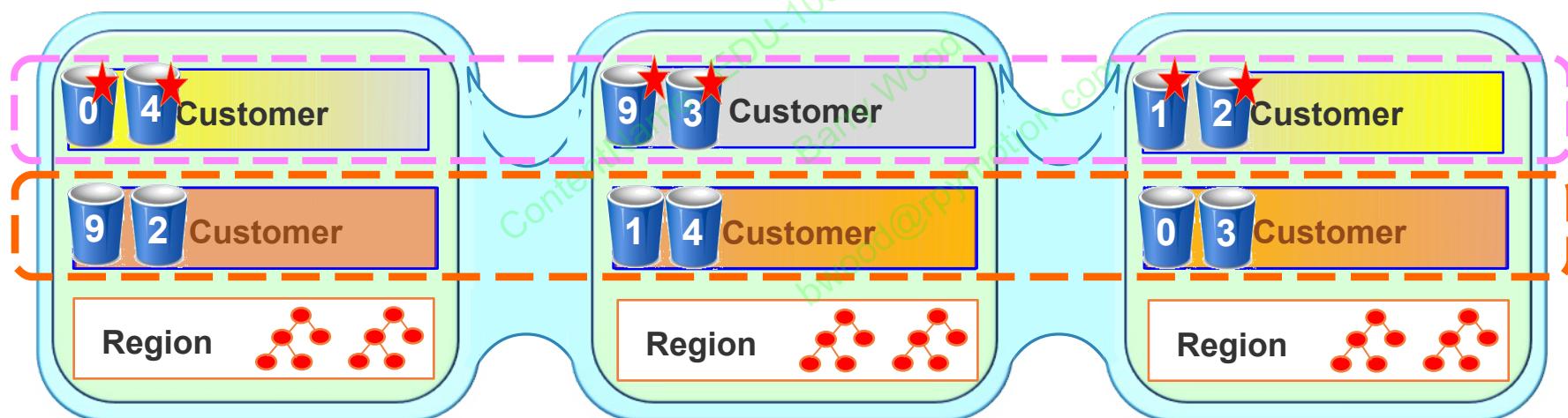


Lesson Road Map

- Partitioned Regions
- **Hardware Failure Scenarios**
- Redundancy Zones and Virtualization
- Data Co-located between Regions
- Replicated Regions
- Region Compression
- Quick Java Overview

Bucket Redundant Distribution

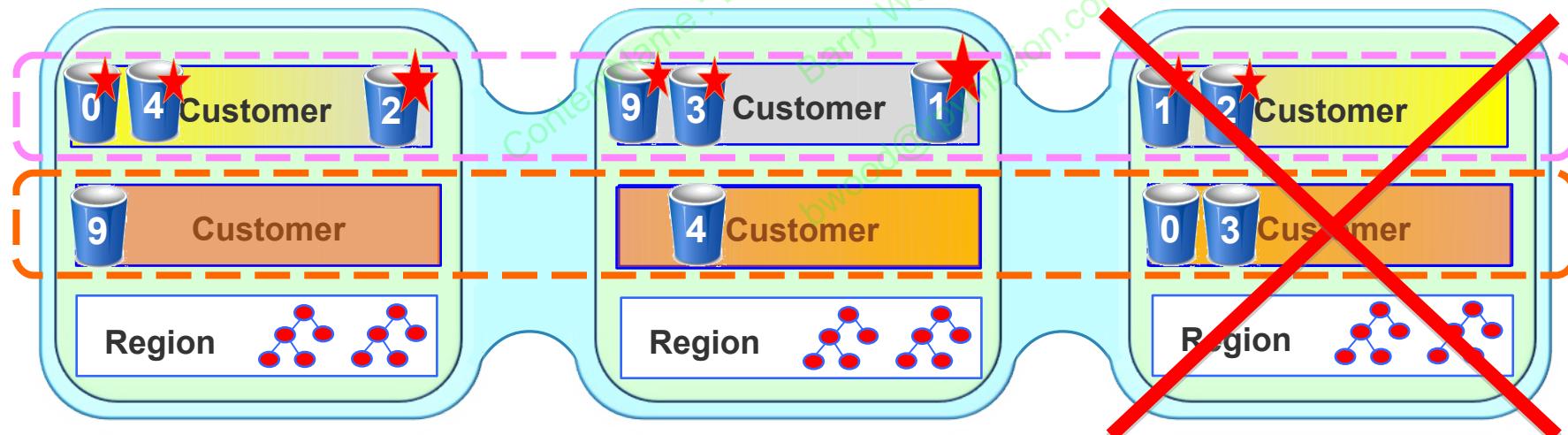
- Buckets are made redundant across the nodes in the distributed system
- Buckets are allocated across nodes to ensure balance



Bucket Promotion on Failure

If a node fails, redundant copies of buckets will be instantly promoted to primary copies

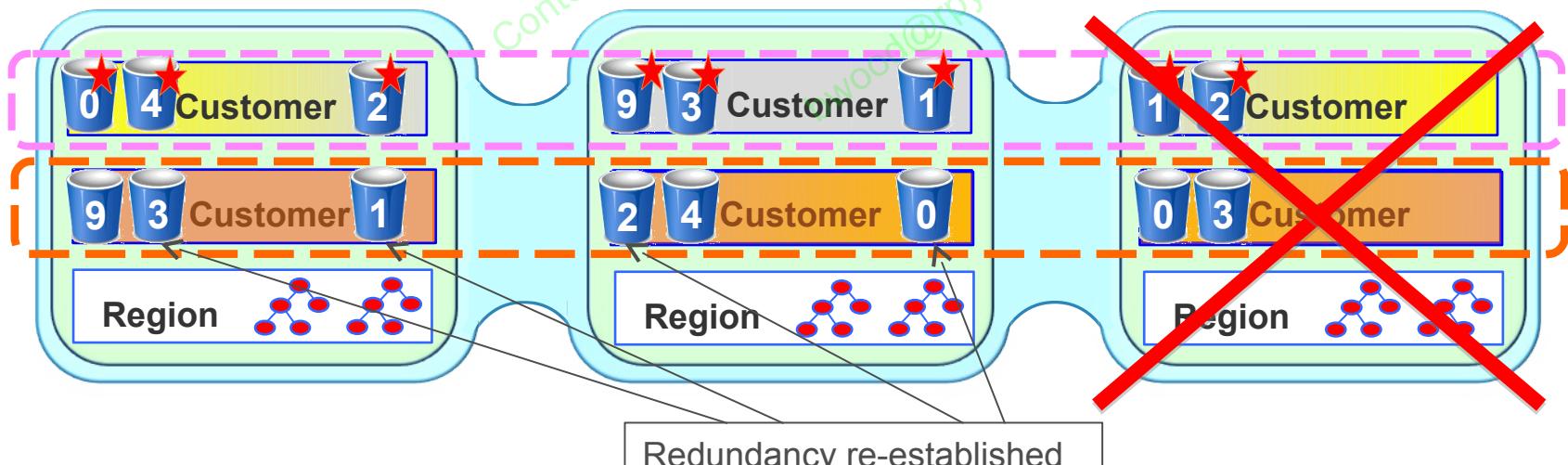
```
gfsh> create region --name=Customer --type=PARTITION_REDUNDANT
```



Bucket Redundancy after Recovery Delay

- Auto-reconnect provides an efficient re-joining mechanism
 - By default, redundancy is not re-established
 - Network latency can cause a node to appear to have failed
- Use **recovery-delay** to allow time to restart the failed node, without incurring the work of the recovery

```
gfsh> create region --name=Customer --type=PARTITION_REDUNDANT  
--recovery-delay=50000
```



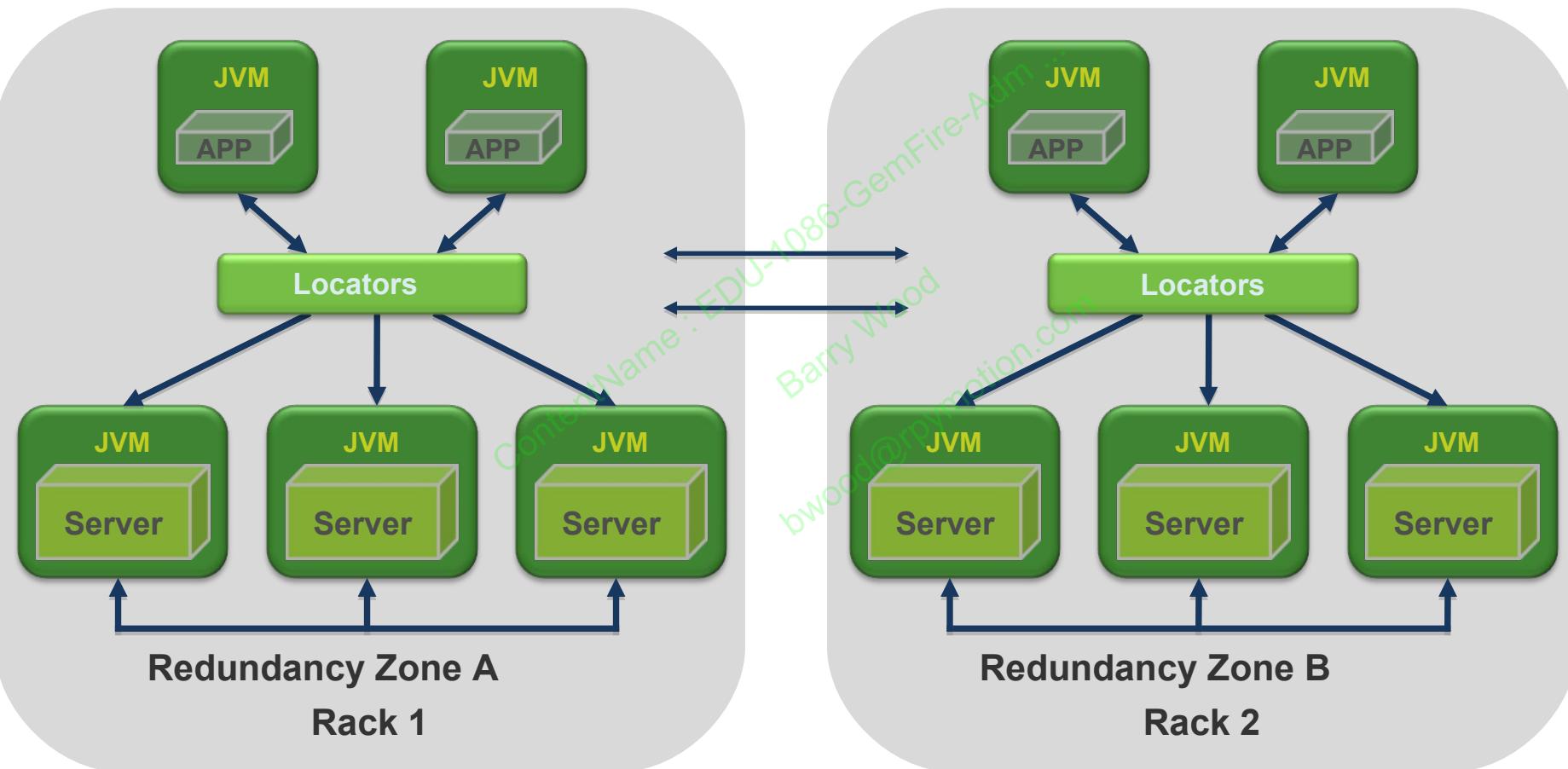
Lesson Road Map

- Partitioned Regions
- Hardware Failure Scenarios
- **Redundancy Zones and Virtualization**
- Data Co-located between Regions
- Replicated Regions
- Region Compression
- Quick Java Overview

Content Name : EDU-101_GemFire-Admin ...
Barry Wood
bwood@rpymotion.com

Redundancy Zones

Controlling location of redundant copies with Redundancy Zones



Redundancy Zones

- Provides an extra measure of high availability
 - Each server assigned a redundancy zone (you should have 3 minimum)
 - Redundant copies of partition data placed into member on a different redundancy zone
 - There should be at least as many redundancy zones as redundant copies for partitioned regions
- Configuration is easy

```
gemfire.properties
```

```
...
```

```
redundancy-zone=rack1
```

Virtualization and High Availability

- In a virtualized environment, concept of ‘machine’ is different
 - Different ‘machine’ may simply be a different VM running on the *same* physical hardware
 - Goal is to have redundant copies of different physical machines
- Enabling is equally as easy
 - Tells GemFire to identify physical machines and store redundant copies on unique physical machine

```
gemfire.properties  
...  
enforce-unique-host=true
```

Lesson Road Map

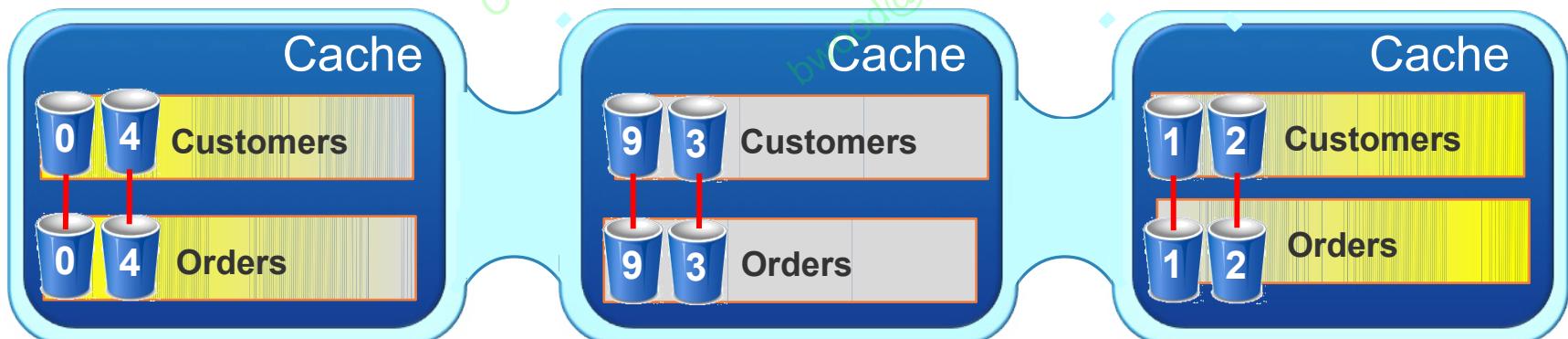
- Partitioned Regions
- Hardware Failure Scenarios
- Redundancy Zones and Virtualization
- **Data Co-located between Regions**
- Replicated Regions
- Region Compression
- Quick Java Overview

Content Name : EDU-103 GemFire Admin ...
Barry Wood
bwood@rpymotion.com

Colocated Regions

- Related entries in multiple data regions, stored in single member
- GemFire does this by storing all of the regions buckets with the same ID together in the same member
- During rebalancing operations, GemFire moves these bucket groups together or not at all

```
gfsh> create region --name=Customers" --type=PARTITION_REDUNDANT  
gfsh> create region --name=Orders --type=PARTITION_REDUNDANT  
--colocated-with=Customers
```



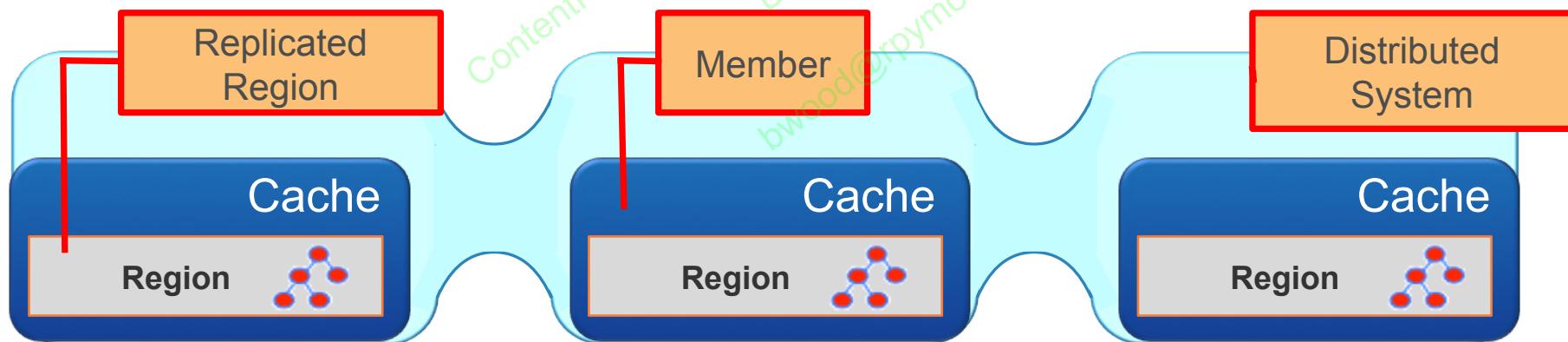
Lesson Road Map

- Partitioned Regions
- Hardware Failure Scenarios
- Redundancy Zones and Virtualization
- Data Co-located between Regions
- **Replicated Regions**
- Region Compression
- Quick Java Overview

Content Name : EDU-103_GemFire_Advanced
Barry Wood
bwood@rpymotion.com

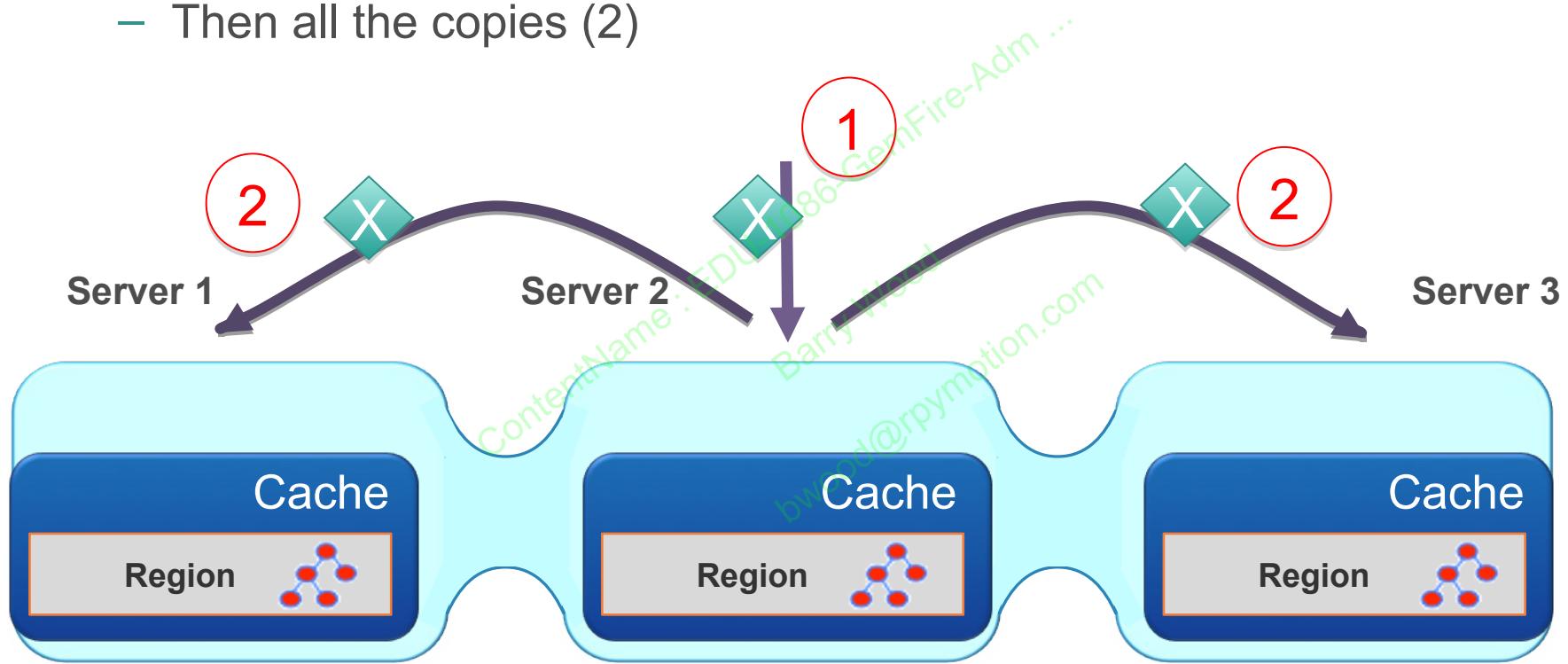
What is a Replicated Region?

- Entire region replicated across GemFire nodes
 - Each member contains *all* data entries
- Best suited for:
 - Read heavy, small data set (reference or lookup data)
 - Data sets that can be entirely contained in a single member
 - Data accessed quickly (each member holds a local copy)



Replicated Region: Updates

- Data on client's server updated first (1)
 - Then all the copies (2)



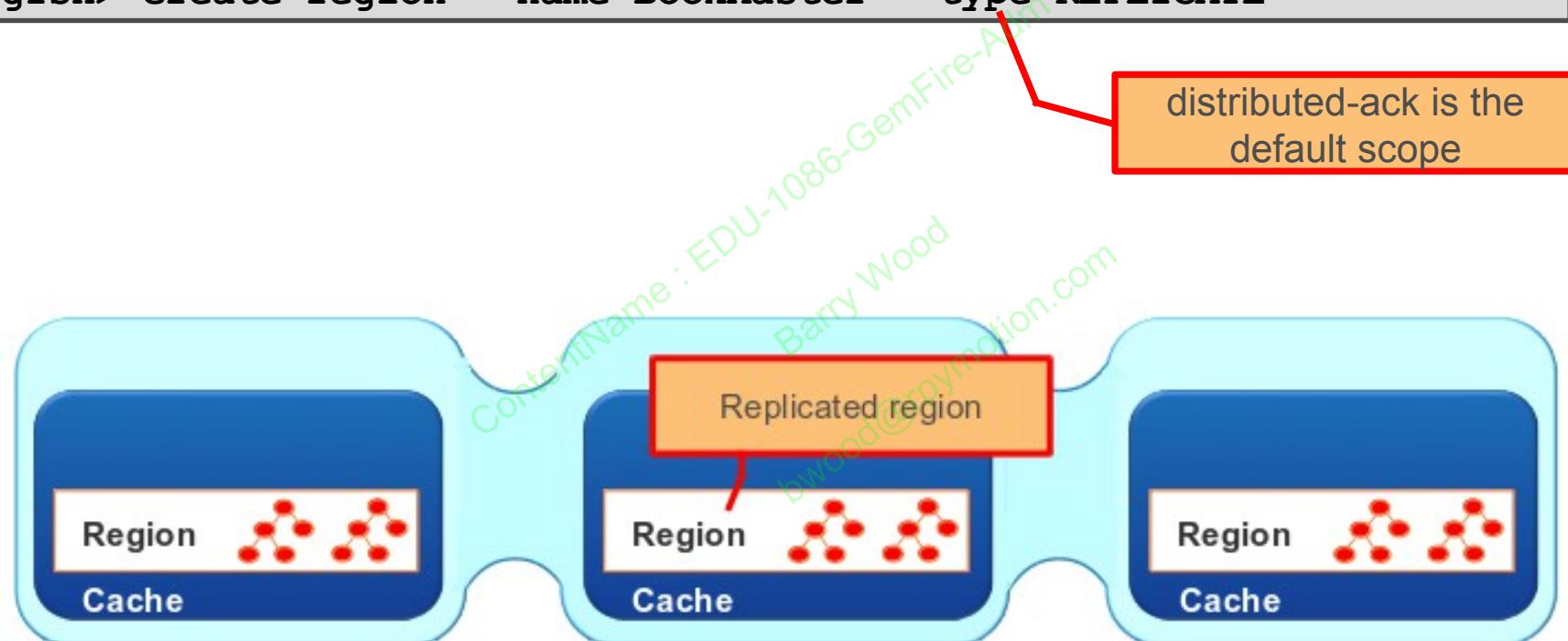
The Automatic Distribution of Entry Value Modification

Relevant Replicated Region Scopes

Scope	Semantics
<code>distributed-no-ack</code>	<ul style="list-style-type: none">Updates are sent to all members, asynchronously, without acknowledgement, without locking.Best performance, but updates can pass each other, resulting in inconsistent data.Network disruptions can cause problems.
<code>distributed-ack</code> (default)	<ul style="list-style-type: none">Updates are sent synchronously, in parallel, to all members, and acknowledged back to sender.Concurrent updates on the same key may pass each other.No locking.
<code>global</code>	<ul style="list-style-type: none">All load, create, put, invalidate, and destroy operations on the region are performed with a distributed lock on the key.Consistency is guaranteed across the distributed system.Slowest performance.

Creating a Replicated Region

```
gfsh> create region --name=BookMaster --type=REPLICATE
```



Lesson Road Map

- Partitioned Regions
- Hardware Failure Scenarios
- Redundancy Zones and Virtualization
- Data Co-located between Regions
- Replicated Regions
- **Region Compression**
- Quick Java Overview

Content home: EDU-103 GemFire Admin ...
Barry Wood
bwood@rpymotion.com

Region Compression Overview

- Compression can reduce memory consumed by entries in a region
- Entries are compressed by registering a compression codec with the region
 - Compression enabled by registering a codec
 - Default codec provided is the Snappy compressor
 - You can write and register your own
- Keys and indexes are NOT compressed



Compression Considerations

- Can be effective when JVM memory usage is high
 - Evaluate free memory relative to Max memory
 - If free memory < 20-25% compression may be beneficial
- Some entries overall benefit more
 - Entries are first serialized into byte array before compression
 - Duplicate bytes can be compressed more effectively (across entire entry, not just a given field)
- Different types benefit more from compression
 - Strings & text values may compress better than numeric
 - Given they are more likely to have repeating bytes when serialized
- Querying on entries
 - Indexing becomes more important
 - Requires decompressing to evaluate query if non-indexed

Configuring Region Compression

- Registered on per-region basis

```
gfsh create region --name=Customer --type=PARTITION_REDUNDANT  
--compressor=org.apache.geode.compression.SnappyCompressor
```

ContentName : EDU-1086-GemFire Admin ...
Barry Wood
bwood@rpymotion.com

Lesson Road Map

- Partitioned Regions
- Hardware Failure Scenarios
- Redundancy Zones and Virtualization
- Data Co-located between Regions
- Replicated Regions
- Region Compression
- **Quick Java Overview**

Content Name : EDU-103_GemFire_Advanced
Barry Wood
bwood@rpymotion.com

About Java Classes

- Java classes are programmatic containers for information and functionality

Customer.java

```
class Customer {  
    // There are usually setters and getters for these  
    private Integer customerNumber;  
    private String firstName;  
  
    public String toString() {  
        System.out.println("Name:" + firstName +  
                           ", CustomerNumber: " + customerNumber);  
    }  
}
```

- A Java source file is compiled into a platform neutral format called ‘bytecode’ with a .class extension

```
$ javac Customer.java
```

About Packages

- Packages are a naming mechanism to avoid confusion
- How do you tell my Customer class from your Customer class?

```
package com.bookshop.domain;

class Customer {  
    ...  
}
```

Fully qualified class name is:
com.bookshop.domain.Customer

- When compiled with 'javac -d', the result will be a directory structure reflecting the package



About JAR Files

- JAR = Java Archive
 - Like a zip file for packaging a bunch of class files
 - Can contain a manifest with information on contents
- Create a JAR file
- Examine JAR contents

```
$ jar cf bookShop.jar com
```

```
$ jar tf bookShop.jar
META-INF/
META-INF/MANIFEST.MF
com/
com/bookshop/
com/bookshop/domain/
com/bookshop/domain/Customer.class
```

About CLASSPATH

- CLASSPATH is similar to the concept of PATH environment variable
 - Multiple segments contain platform specific separator (ex ‘:’ for Unix, ‘;’ for Windows)
 - Represents all the paths the Java runtime will search for the specified class
- CLASSPATH can contain JAR files
- CLASSPATH can contain directories with .class files

```
$ export CLASSPATH=$CLASSPATH:./bookShop.jar
```

Add bookShop.jar to the current CLASSPATH

```
$ export CLASSPATH=$CLASSPATH:.
```

Add current directory to the current CLASSPATH

Putting It All Together

- You added bookShop.jar to your CLASSPATH variable
- bookShop.jar contains a set of classes for this exercise
- One of the classes is BookLoader in the com.bookshop.buslogic package
- You can now execute the BookLoader class with the following command

```
$ java com.bookshop.buslogic.BookLoader
```

- The java command searches the CLASSPATH for the above class in the specified package
- Note we don't include the '.class' extension



Pivotal
GemFire

Version 9.0.4c

Lab – Replicated & Partitioned Regions

In this lab, you will:

1. Configure a replicated region across multiple servers
2. Configure a partitioned region with redundancy across multiple servers
3. Use gfsh describe and list commands to examine members and regions
4. Use gfsh to read and query region entries
5. Use gfsh to add and update region entries

Review of Objectives

You should be able to do the following:

- Describe and configure partitioned regions
- In partitioned regions, describe how data entries are placed in buckets based on hashing
- Describe data co-location and how to configure data co-located between regions
- Describe and configure replicated regions

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

JVM Configuration and Off-Heap Storage

ContentName : EDU-RP-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Objectives

After completing this lesson, you should be able to:

- Understand basic JVM Heap configuration
- Understand benefits of off-heap storage
- Understand how to configure off-heap storage

ContentName : EDU-105-GemFire-Admin
Barry Wood
bwood@rpymotion.com

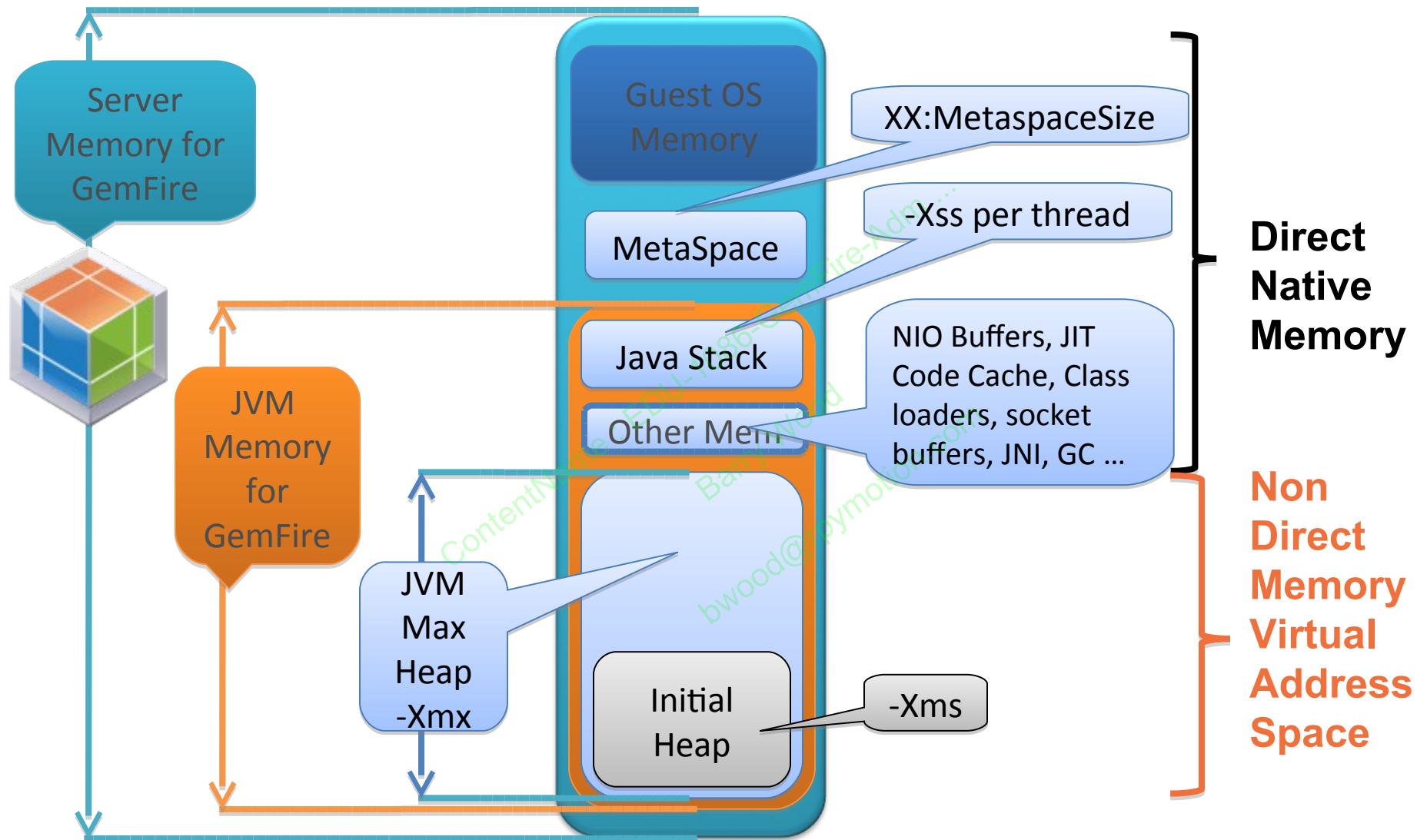
Lesson Road Map

- **JVM Configuration**
- Off Heap Storage and Configuration



ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

JVM Memory Segments



JVM Memory Segments

- GemFire Server Memory consists of
 - Guest OS Memory
 - Approx 0.5-1GB depending on OS & other processes)
 - JVM Metaspace Size (**-XX:MaxMetaspaceSize**)
 - JVM Maximum Heap Size (**-Xmx**)
 - Memory for each concurrent thread (**-Xss**)
 - “*Other Memory*” – additional memory required for:
 - NIO buffers and socket buffers (receive/send)
 - JIT code cache, Class-loaders, JNI, GC internal info



Total Memory Usage

- Calculated as follows

GemFire Server Memory =

Guest OS Memory +

JVM Metaspace Size (-XX:MaxMetaspaceSize) +

JVM Memory for GemFire

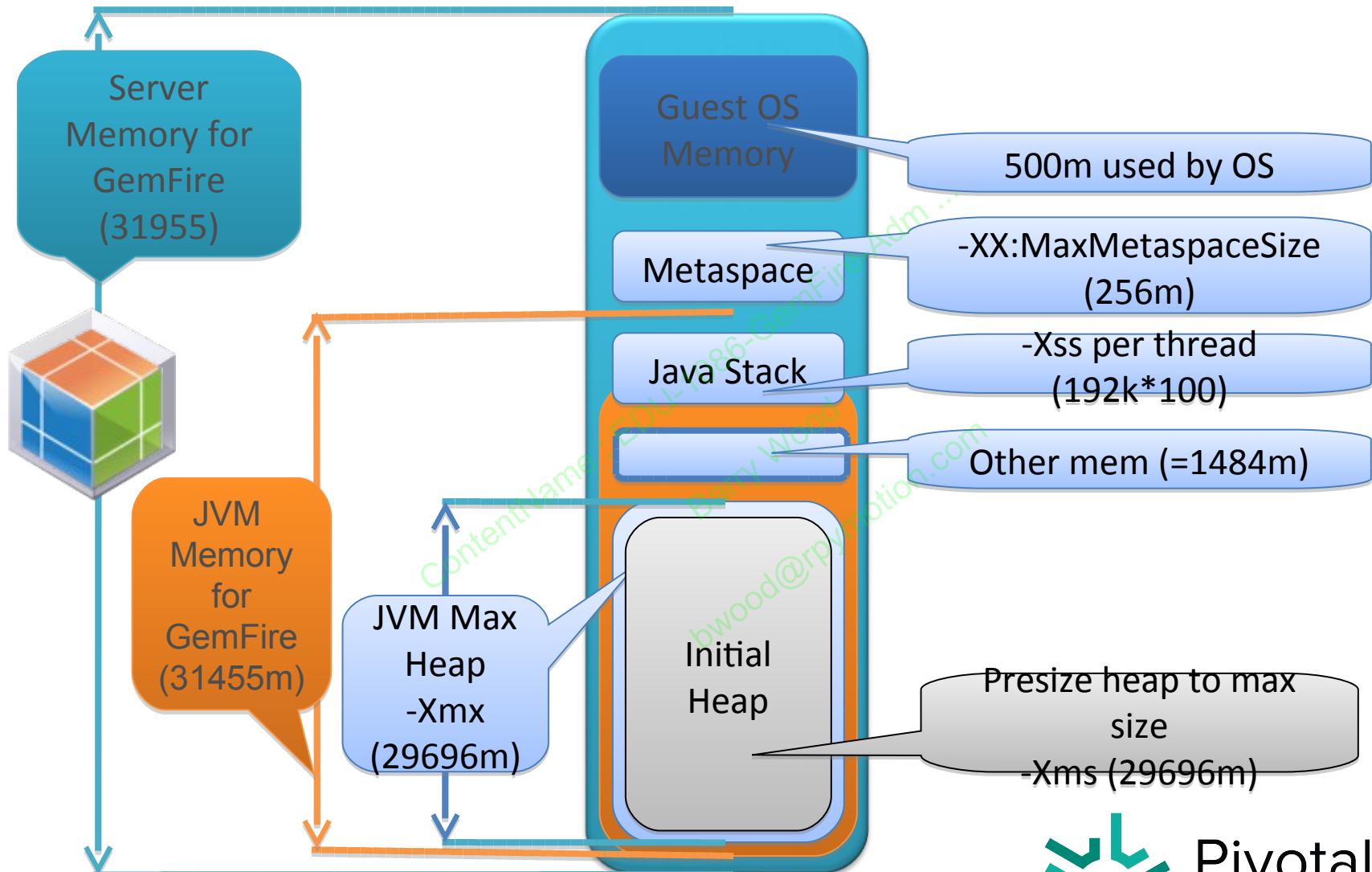
JVM Memory for GemFire =

JVM Max Heap (-Xmx value) +

NumberOfConcurrentThreads^{opt}* (-Xss) +

Other Memory

Example JVM Memory Configuration



JVM Options

VM Flag	What it does
-XX:+UseParNewGC	Instructs the VM to use parallel GC in the “New” (Young) heap space
-XX:+UseConcMarkSweepGC	Instructs the VM to use Concurrent Mark and Sweep in the “Old” (Tenured) heap space
-XX:+DisableExplicitGC	Disables calls to “System.gc()”
-XX:CMSInitiatingOccupancyFraction	Tells the CMS collector when to start. Coordinates with Resource Manager
-Xms same as -Xmx	Set the initial and maximum heap to be same size
-XX:NewSize same as -XX:MaxNewSize	Sets the initial and maximum Young Generation size to be same size
-Xmn	Sets the Young Generation heap – recommended to be 33% of max heap
-XX:MaxMetaspaceSize	Set the Metaspace size

Specifying JVM Options via gfsh

- Options can be specified when starting locators or servers

```
gfsh> start server --name=<server-name> \
    --J=-XX:+UseParNewGC \
    --J=-XX:CMSInitiatingOccupancyFraction=60
```

Translates to:
java -XX:+UseParNewGC -XX:CMSInitiatingOccupancyFraction=60 ...
(other startup arguments)

Lesson Road Map

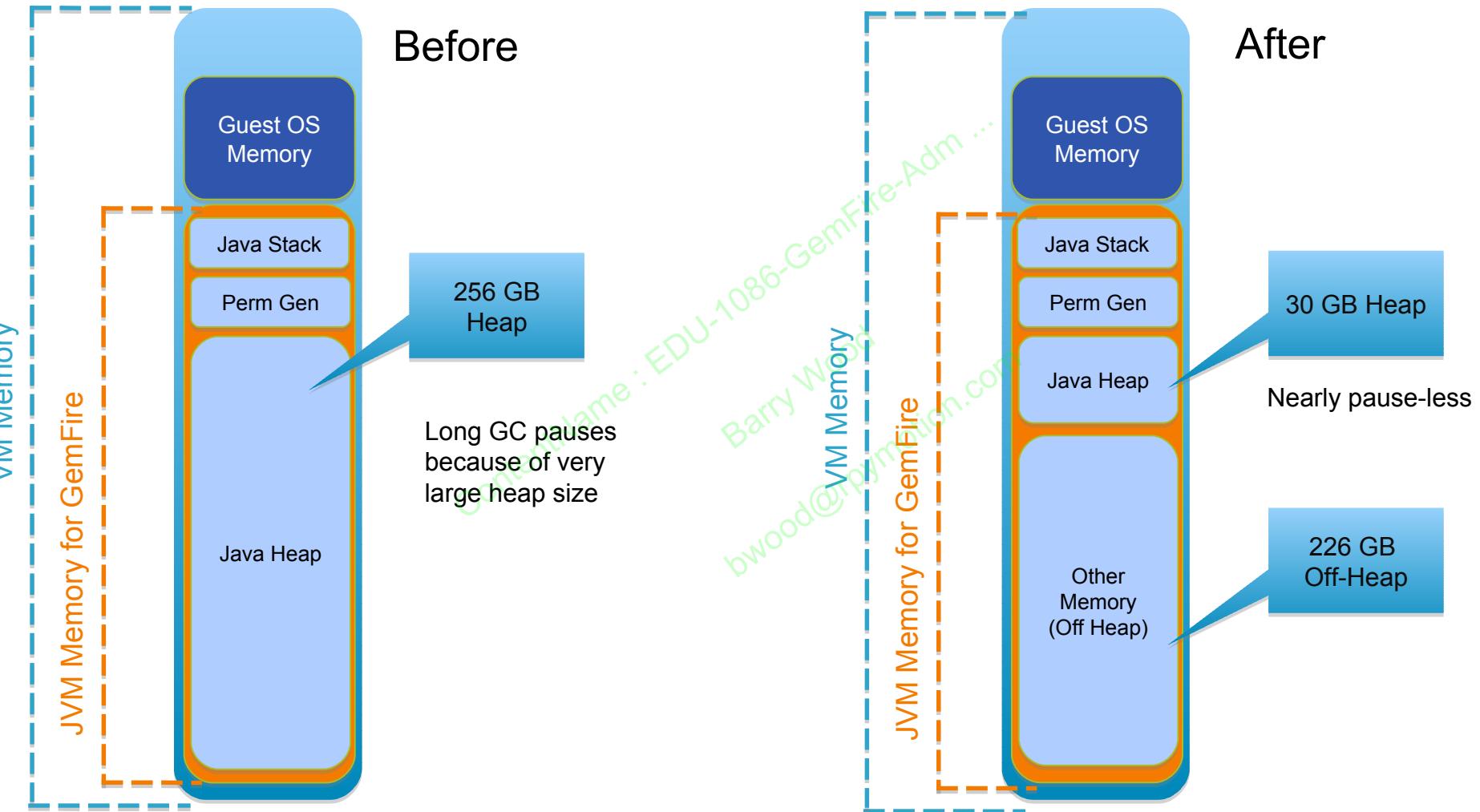
- JVM Configuration
- **Off Heap Storage and Configuration**



ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Off-Heap Memory - New In GemFire 9.0!

Using Memory That is Separate From the Java Heap



Why Off-Heap Memory?

- Dramatically increases the volume of user data that can be managed
 - 128+ GB user data in one JVM
 - 10s of TB user data in one cluster
- Usable out-of-box without extensive GC tuning of JVM
- Maintain existing throughput performance
- Reduces the need for extensive tuning related to garbage collection
- Implements its own memory manager (including handling deleted objects)

How do I use it?

- Set the off-heap memory size for the process
 - Using the new property: **off-heap-memory-size**
- Mark regions whose entry values should be stored off-heap
 - Using the new region attribute: **off-heap (false | true)**
- Adjust the JVM heap memory size down accordingly
 - The smaller the better; at least try to keep it below 32G

Start-up options

- **--off-heap-memory-size**
 - specifies amount of off-heap memory to allocate
- **--lock-memory**
 - specifies whether to lock the off-heap memory
- ***Example:***

```
gfsh> start server --name=server1 \
    --initial-heap=10g --max-heap=10g \
    --off-heap-memory-size=200g --lock-memory=true
gfsh> create region --name=Customer --type=REPLICATE --off-heap
```

Off-heap interaction with other features

- Deltas: values have to be copied (the clone option is ON, and cannot be turned off)
- EntryEvents
 - Limited availability of oldValue, newValue
- Indexes
 - Functional range indexes not supported (too expensive)
- Region operations that compare the value
 - Do not call object equals
 - Either PDXInstance or serialized bytes in order to do equality check without having to deserialize data

More expensive with off-heap

- PDX-instances
 - Values currently copied from off-heap to create a PDXInstance
- Deltas: to apply a delta you have to serialize/deserialize every time
- Compression: there is an extra copy on decompression
- Querying: deserialize on every query
 - Use Indexes

gfsh Support for Off-heap Memory

- ***alter disk-store***: new option "--off-heap" for setting off-heap for each region in the disk-store
- ***create region***: new option "--off-heap" for setting off-heap
- ***describe member***: now displays the off-heap size
- ***describe offline-disk-store***: now shows if a region is off-heap
- ***describe region***: now displays the off-heap region attribute
- ***show metrics***: Now has an offheap category. The offheap metrics are: maxMemory, freeMemory, usedMemory, objects, fragmentation, and compactionTime
- ***start server***: added --lock-memory, --off-heap-memory-size, --critical-off-heap-percentage, and --eviction-off-heap-percentage

What is Stored On-heap vs. Off-heap

Always Stored On-heap	Stored Off-heap
Region Meta-Data	Values
Entry Meta-Data	Reference Counts
- Off-Heap Addresses	Lists of Free Memory Blocks
Keys	WAN Queue Elements
Indexes	
Subscription Queue Elements	

Heap vs. Off-Heap Comparison

	Java Heap	Off-Heap
creates/sec	30,000	45,000
updates/sec	17,000 (std dev: 2130)	51,000 (std dev: 737)
Java RSS size	256 GB	32 GB
CPU load	70% (load avg 10 cpus)	32% (load avg 5 cpus)
JVM GC	ConcurrentMarkSweep	ConcurrentMarkSweep
GC ms/sec	777 ms	24 ms
GC marks (GC pauses)	1 per 30 sec	never

Note: Based on one preliminary benchmark

Review of Learner Objectives

You should be able to do the following:

- Understand basic JVM Heap configuration
- Understand benefits of off-heap storage
- Understand how to configure off-heap storage

ContentName : EDU-105-GemFire-Admin
Barry Wood
bwood@rpymotion.com

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

Management Basics

Overview of common *gfsh* commands

Content frame : EDU-1086-GemFire-Adm ...
Deny Wood
bwood@rpymode.com

Objectives

After completing this lesson, you should be able to:

- Understand usage of `gfsh`
- Describe how to start/stop Locators and Servers
- Understand how to use scripts for operational tasks
- Describe how and when to rebalance a system
- Describe how to create regions and deploy JAR files
- Understand how to operationalize GemFire using scripts

Lesson Road Map

- **Using gfsh**
- Starting/Stopping members
- Rebalancing
- Adding regions & deploying JAR files
- Operationalizing

ContentName : EDU-101-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Gfsh “Gee-fish”

Overview

- GemFire **gfsh** (pronounced "gee-fish")
- Command-line interface - launch, manage, and monitor GemFire processes, data, and applications
- Unix-like shell commands
- Features command completion, context-sensitive help
- Uses JMX/RMI to communicate with GemFire processes



What can I do with gfsh?

Start/Stop commands

- connect
- disconnect
- start (locator, server)
- start (gateway-receiver/sender)
- start (pulse, jconsole, etc)
- stop (locator, server)
- stop (gateway-receiver/sender)

Informational commands

- debug
- describe (member, region,...)
- echo
- help / hint / history
- list (members, regions, ...)
- netstat
- show (metrics, log, ...)
- status
- version

General Maintenance commands

- alter (disk-store, region, etc)
- change loglevel
- clear defined indexes
- close
- configure
- create (index, region,...)
- define index
- deploy
- destroy
- encrypt password
- export
- gc
- pause/resume gateway-sender
- import
- pdx rename
- run
- set variable
- shutdown
- sleep
- undeploy

Region/Entry commands

- execute function
- get
- locate entry
- put
- query
- rebalance
- remove

DiskStore Operations

- backup disk-store
- compact
- revoke missing-disk-store
- upgrade offline-disk-store
- validate offline-disk-store

gfsh - help, hint, history

- **gfsh help**
 - Syntax & usage information for all available commands
 - Typing help without command lists all commands
 - Provides extensive *context-sensitive* tab-completion assistance
 - Takes advantage of JMX Manager info
- **gfsh hint** – to display commands related to a topic
- **gfsh history** – shows command history
 - **--file** defines text file to which history is saved

```
gfsh> help [command]
gfsh> hint [topic]
gfsh> history [--file=<history text file>]
```

Lesson Road Map

- Using gfsh
- **Starting/Stopping members**
- Rebalancing
- Adding regions & deploying JAR files
- Operationalizing

ContentName : EDU-101-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Starting and Stopping - Locators

- Locators:
 - Started individually through the gfsh command-line utility
 - Uses own **gemfire.properties** files for configuration
 - Starting locator

```
gfsh start locator --name=locator1  
    --dir=/gflogs/locator1  
    --port=41111  
    --group=Group1  
    --initial-heap=1024m  
    --locators=LOC001[10777],LOC002[10777]  
    --max-heap=1024m  
    --properties-file=./gemfire.properties
```

- Stopping locator

```
gfsh> stop locator --name=locator1
```

Starting and Stopping – Cache Servers

- GemFire Cache Servers:
 - Started individually through the gfsh command-line utility
 - Uses own **gemfire.properties** files for configuration
 - Starting server

```
gfsh start server --name=server1  
    --classpath=./clientServer.jar  
    --dir=/gflogs/server1  
    --group=Group1  
    --locators=host1[41111]  
    --properties-file=gemfire.properties  
    --server-port=41223
```

- Stopping server

```
gfsh> stop server --name=server1
```

Attributes for Starting Locators & Servers

--dir attribute

- Defines directory Locator or Server will run in
- Log files are written to this directory
 - If dir attribute is defined, then the directory must exist, otherwise "*The working directory for the Locator could not be found*"
 - If dir attribute not defined, creates <CWD>/memberName
 - CWD = *Current Working Directory* of gfsh when start command is issued

--classpath

- Path on which servers should search for classes, value appended to the current CLASSPATH



Start Up Your System

- Start locators first
- Then start the cache servers
- If you have processes that persist data to disk, start them as close to the same time as possible
 - Don't start one and wait before starting next
 - We'll cover in more detail in Disk Store section

Shut Down Your System

- Shut down using **gfsh shutdown** command
 - Must be connected to JMX Manager

```
gfsh> connect --locator=localhost[41111]
```
 - Shut down only servers

```
gfsh> shutdown
```
 - Shut down all members

```
gfsh> shutdown --include-locators=true
```
- Shut down individual servers using **gfsh stop server**
 - While connected to JMX Manager

```
gfsh> stop server --name=serverName
```
 - Not connected to JMX Manager

```
gfsh> stop server --dir=serverDir
```

gfsh – Status of Locators & Servers

- **gfsh> status locator** command
 - display status of locator
- **gfsh> status server** command
 - display status of server

```
status locator [--name=locator_name]
                [--host=ip_address_or_hostname]
                [--port=locator_port]
                [--pid=locator_pid]
                [--dir=locator_working_directory]

status server [--name=server_name] [--pid=server_pid]
                [--dir=server_working_directory]
```

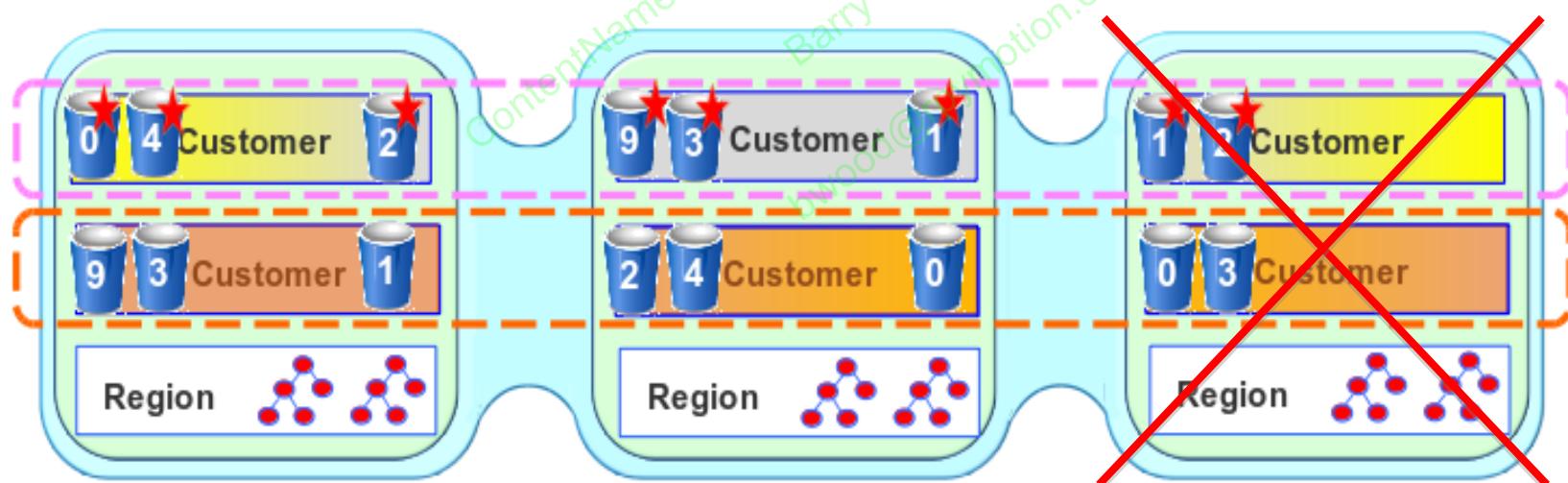
Lesson Road Map

- Using gfsh
- Starting/Stopping members
- **Rebalancing**
- Adding regions & deploying JAR files
- Operationalizing

ContentName : EDU-101-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

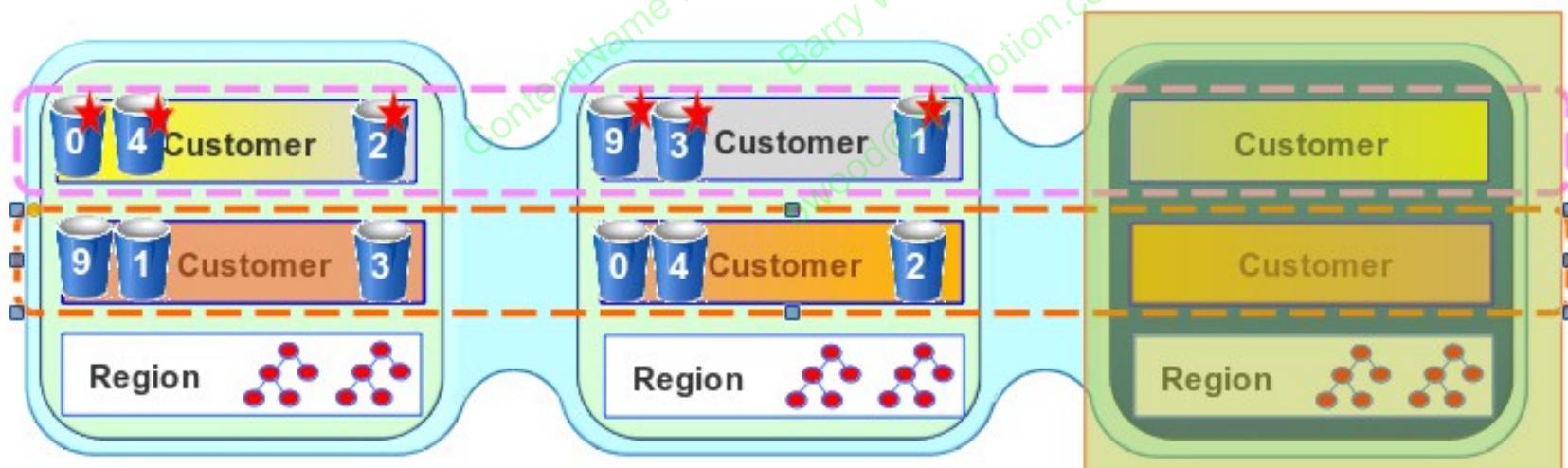
Bucket Promotion

- Use **recovery-delay** to allow time to restart the failed node, without incurring the work of the recovery
 - But: network latency can cause a node to appear to have failed
- If a node has failed in the eyes of the membership coordinator, it will *not* be allowed to bring its data back in
 - It will be loaded with a fresh copy from remaining nodes



Rebalancing using gfsh – Command line

```
gfsh> rebalance [<attribute name> = <attribute Value>]*  
  
gfsh> rebalance [--include-region=partition_region_path_or_name  
                  (,partition_region_path_or_name)*]  
                  [--exclude-region=partition_region_path_or_name  
                  (,partition_region_path_or_name)*]  
                  [--time-out=time_to_wait_in_seconds]  
                  [--simulate(=true_or_false)?]
```



Lesson Road Map

- Using gfsh
- Starting/Stopping members
- Rebalancing
- **Adding regions & deploying JAR files**
- Operationalizing

ContentName : EDU-100-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Create Region – From XML

- Region defined in cache.xml file

```
gfsh> start server --cache-xml-file=cache.xml
```

```
<cache>
  <region name="myRegion" refid="PARTITION_REDUNDANT">
    <region-attributes>
      <partition-attributes local-max-memory="512" />
    </region-attributes>
  </region>
</cache>
```

cache.xml

Create Region – Without Restarting Server

gfsh command to create a region:

- Must specify either a **--type** or a **--template-region** for initial configuration when creating a region
- Definitions added to Cluster Configuration Service

```
create region --name=region12 --template-region=/region1  
  
create region --name=region2 --type=REPLICATE \  
--cachelistener=com.gemstone.gemfire.examples.MyCacheListener1, \  
com.gemstone.gemfire.examples.MyCacheListener2 \  
--group=Group1,Group2
```

Deploying JAR Files

- JAR files copied across the network
- JAR files automatically added to server classpath
- Automatic registration of functions in JAR files
- All managed from gfsh
 - Deploying to all servers

```
gfsh> deploy --jar=myFunctions.jar
```

- Deploying to server group

```
gfsh> deploy --jar=myFunctions.jar --group=myGroup
```

- Listing deployed JAR files

```
gfsh> list deployed [--group=myGroup]
```

- Un-deploying

```
gfsh> undeploy --jar=myFunctions.jar [--group=myGroup]
```

JAR Deployment

- By default, JAR files written to each member's working directory
 - Override with **deploy-working-dir** property

```
#gemfire.properties  
deploy-working-dir=/usr/local/gemfire/deploy
```

- Deploying a JAR file to server also adds to the server **CLASSPATH**
- Un-deploy will remove JAR from **CLASSPATH** but will *not* remove classes loaded in server – restart server
- When deploying a JAR containing a GemFire Function, it is automatically registered

JAR Deployment and Event Handlers

- Other than Functions, deploying a JAR file only adds classes to server CLASSPATH
- You must register event handlers after deploying JAR
- Cache Listener

```
gfsh> alter region --name=/Customer  
      --cache-listener=com.bookshop.MyListener
```

- Cache Loader

```
gfsh> alter region --name=/Customer  
      --cache-loader=com.bookshop.MyLoader
```

- Cache Writer

```
gfsh> alter region --name=/Customer  
      --cache-writer=com.bookshop.MyWriter
```

Lesson Road Map

- Using gfsh
- Starting/Stopping members
- Rebalancing
- Adding regions & deploying JAR files
- **Operationalizing**

ContentName : EDU-101-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Operationalizing GemFire Tasks

- Once key parameters and processes determined, consider how to automate
- gfsh** commands can be organized into script files
 - Script can be run interactively from **gfsh** prompt or from command line non-interactively
 - Limited support for environment and other variables
- Commands can be issued from Linux script or from Windows batch file
 - Allows for parameterization of values using environment variables
 - gfsh** invoked in non-interactive mode



Using gfsh Script Files

- Multiple **gfsh** commands can be executed from single script file
- Order of commands is important

```
start locator  
--name=locator1 --properties-file=./gemfire.properties  
  
start server  
--name=server1 --properties-file=./gemfire.properties
```

mycommands.gfsh (script)

- You can run scripts that contain gfsh commands by using the gfsh run command:

```
gfsh> run --file=mycommands.gfsh
```

- Or, from shell/batch window

```
$ gfsh run --file=mycommands.gfsh
```

Using gfsh Variables

- **gfsh** provides a set of built-in variables for use in scripting
 - To see a list of currently defined variables and values:

```
gfsh> echo --string=$*
```

- Reference variables using the `$(variable)` syntax
- Set your own variables using `set variable`

```
gfsh> set variable --name=LOCATORS --value="localhost[41111]"  
Value for variable LOCATORS is now: localhost[41111]  
gfsh> start server --name=server1 --locators=${LOCATORS} ...
```

- Note: Not all environment variables are available in gfsh

gfsh Scripting Pros and Cons

- Advantages
 - Contents of scripts are platform neutral
 - Can be run both from inside **gfsh** and from command line
- Disadvantages
 - Limited platform-specific environment variable support
 - Scripting language limited (i.e. no branching support, etc)



Scripting the Startup

- You can also script starting and stopping of services from your native command prompt

```
$ gfsh start locator --name=$NAME  
    --dir=$GEMFIRE_LOCATOR_LOGDIR  
    --port=$PORT  
    --group=Group1  
    --initial-heap=$JVM_INITIAL_HEAP  
    --locators=LOC001[10777],LOC002[10777]  
    --max-heap=$JVM_MAX_HEAP  
    --properties-file=$GEMFIRE_PROPERTIES_FILE
```

- You can encapsulate in a native script (Linux bash or Windows command)

Linux Bash example

```
#!/bin/bash

# Defines all the variables used below
source setenv.sh

# Start the locator using variables
gfsh start locator
--name=$LOCATOR_NAME          --port=$PORT
--dir=$GEMFIRE_LOCATOR_LOGDIR --group=Group1
--initial-heap=$JVM_INITIAL_HEAP --locators=LOC001[10777]
--max-heap=$JVM_MAX_HEAP
--properties-file=$GEMFIRE_PROPERTIES_FILE

# Start server using variables
gfsh start server --name=$SERVER_NAME --dir=$SERVER_LOGDIR ...
```



Windows example

```
echo off
rem Defines all the variables used below
setEnv.bat

rem Start the locator using variables defined in setEnv.bat
gfsh start locator
--name=%LOCATOR_NAME% --port=%PORT%
--dir=%GEMFIRE_LOCATOR_LOGDIR% --group=Group1
--initial-heap=%JVM_INITIAL_HEAP% --locators=LOC001[10777]
--max-heap=%JVM_MAX_HEAP%
--properties-file=%GEMFIRE_PROPERTIES_FILE%

rem Start server using variables
gfsh start server --name=%SERVER_NAME% --dir=%SERVER_LOGDIR% ...
```



Scripting Pros and Cons

- Advantages
 - Rich syntax for scripting (i.e. branching, conditional execution, etc)
 - Full access to environment variables and variable manipulation
- Disadvantages
 - Very platform-specific – if you need to support multiple platforms, you need multiple scripts

Lab – Management Basics

In this lab, you will:

1. Create script to start locator and servers
2. Create script to shutdown servers and locator
3. Export data from regions
4. Import data into regions on startup using script

Review of Objectives

You should be able to do the following:

- Use gfsh to perform standard system admin task
- Describe how to start/stop Locators and Servers
- Be able to write basic operational scripts
- Describe how and when to rebalance a system
- Describe how to create regions and deploy JAR files
- Understand how to operationalize GemFire using scripts

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

Additional Management Tools

JMX Manager, Pulse, jConsole

Contentame: EDU-116-GemFire-Adm ...
Davy Wood
bwood@rpymotion.com

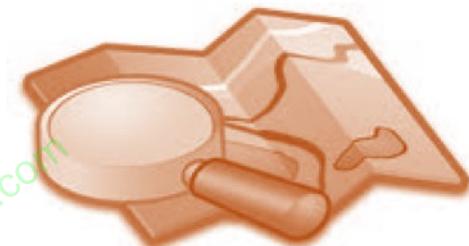
Objectives

After completing this lesson, you should be able to:

- Understand how JMX supports viewing and monitoring of all members in the distributed system
- Understand the differences between a Managed Node and a JMX Manager Node
- Understand Pulse monitoring views
- Understand monitoring functionality provided by JConsole

Lesson Road Map

- JMX Manager
- Pulse
- JConsole



ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Distributed System Monitoring

- JMX – Federated Open MBean strategy
 - Beginning GemFire 7.0
 - Manage and monitor all members of the distributed system
 - Single client to **view** and **manage** all system members
 - GemFire integrates with third party JMX compliant tools
- Each member of the distributed system is a Managed Node
- JMX Manager Node – configured Managed Node, 1+
 - Proxy to Managed Nodes
 - Manage all other system nodes
 - Access all system nodes through JMX Manager Node



Monitoring – JMX Architecture

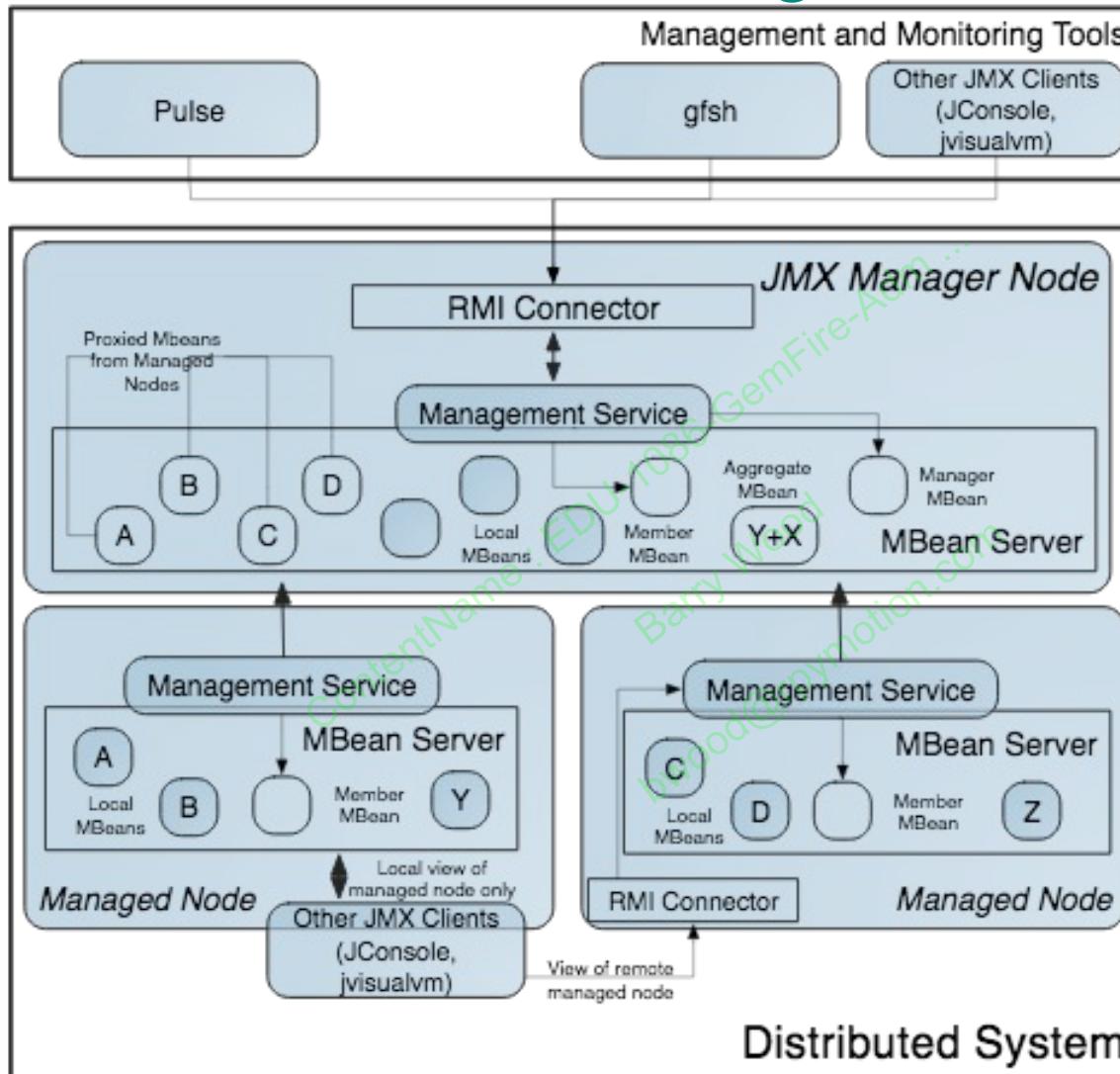
Managed Node

- RMI connector allowing remote JMX client to connect directly to node and access all local MBeans.
- Local MBeans that represent the locally monitored components on the node
- Built-in platform MBeans

JMX Manager Node

- RMI connector allowing JMX clients to connect to and access all MBeans in the distributed system
- Local MBeans that represent the locally monitored components on this node
- Aggregate MBeans
- Proxy to MBeans on managed nodes
- Built-in platform MBeans

JMX – GemFire Monitoring Architecture



JMX - gfsh

- GemFire shell (gfsh) only connects to:
 - JMX Manager Node
 - Locator
 - Has information on JMX Manager so a connection to the JMX Manager can be established
 - If no JMX Manager exists, then the Locator becomes the JMX Manager
- Being a JMX Manager requires additional memory
 - Locators are typically JMX Managers



Starting JMX Manager

- By default when started, a Locator becomes a JMX Manager if no other exist, In the gemfire.properties file:

```
jmx-manager=true //default
```

- By default a Server starts as a Managed Node, jmx-manager=false. To start a server as a JMX Manager, then in gemfire.properties file:

```
jmx-manager=true //become JMX-Manager, passively  
jmx-manager-start=true //on start-up, 1+
```

- Start Server as a JMX Manager from gfsh command line:

```
gfsh> start server -name=<server-name> \  
      --J=-Dgemfire.jmx.manager=true \  
      --J=-Dgemfire.jmx-manager-start=true
```

Stopping JMX Manager

- To stop a JMX Manager, simply stop or shutdown the locator or server hosting the JMX Manager:

```
gfsh>stop locator --name=locator1
Stopping Locator running in /Applications/GemFire-Admin-
8.1.0a.RELEASE/runLab/locator1 on 192.168.0.60[41111] as
locator1...
Process ID: 84164
Log File: /Applications/GemFire-Admin-
8.1.0a.RELEASE/runLab/locator1/locator1.log
...
No longer connected to 192.168.0.60[1099].
```

- JMX Manager
 - Only one is required within a single distributed system
 - Multiple within a single distributed system are supported

Working with Remote JMX Manager

- In some cases, systems may have internal address and external address
 - JMX Manager known to Locator by internal address
 - gfsh connects to locator via external address – locator tries to refer to JMX Manager & fails
- On locator system

```
gfsh start locator --name=server1 locator --port=41111 \
--J=-Djava.rmi.server.hostname=164.55.80.204
```

Ensures the JMX Manager is known by the external IP address

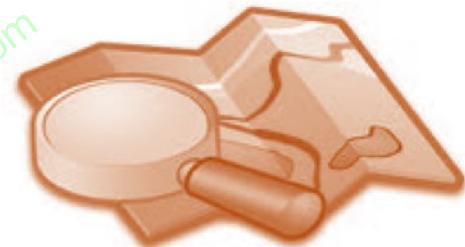
- Then on remote client

```
gfsh>connect --jmx-manager=164.55.80.204[12501]
Connecting to Manager at [host=164.55.80.204, port=12501] ..
Successfully connected to: [host=164.55.80.204, port=12501]
```

Lesson Road Map

- JMX Manager
- **Pulse**
- JConsole

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



GemFire Pulse

Supported Browsers (Tested)

- Internet Explorer (9.0.8112.16421)
- Safari (5.1.7 or greater)
- Google Chrome (22.0.1229.79)
- Mozilla Firefox (16.0.1)

ContentName : EDU-1006-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

GemFire Pulse Is:

- Web Application providing a graphical dashboard for:
 - monitoring vital, real-time health
 - performance of GemFire clusters, members, regions
- JMX client that connects to a GemFire JMX manager to:
 - Retrieve & present statistics about the GemFire system
- Application that runs in a:
 - Tomcat server container embedded in JMX manager node
 - Optionally deploy Pulse to a Web application server of your choice, tool runs independently of GemFire clusters
 - Hosting Pulse on an application server also enables you to use SSL for accessing the application



GemFire Pulse Provides:

- **Cluster view**
 - High-level overview of the GemFire distributed system
 - Displayed immediately after you log into Pulse
 - Information displayed around perimeter cluster view:
 - Statistics - memory usage, JVM pauses, throughput
- **Member view**
 - Provides member-specific information:
 - Regions available on that member
 - Configured listen ports and WAN configuration
- **Data view**
 - Provides a comprehensive overview:
 - All regions in the GemFire distributed system

Starting Pulse (Embedded)

- Connect to the locator:

```
gfsh> connect --locator=host3[41111]
```

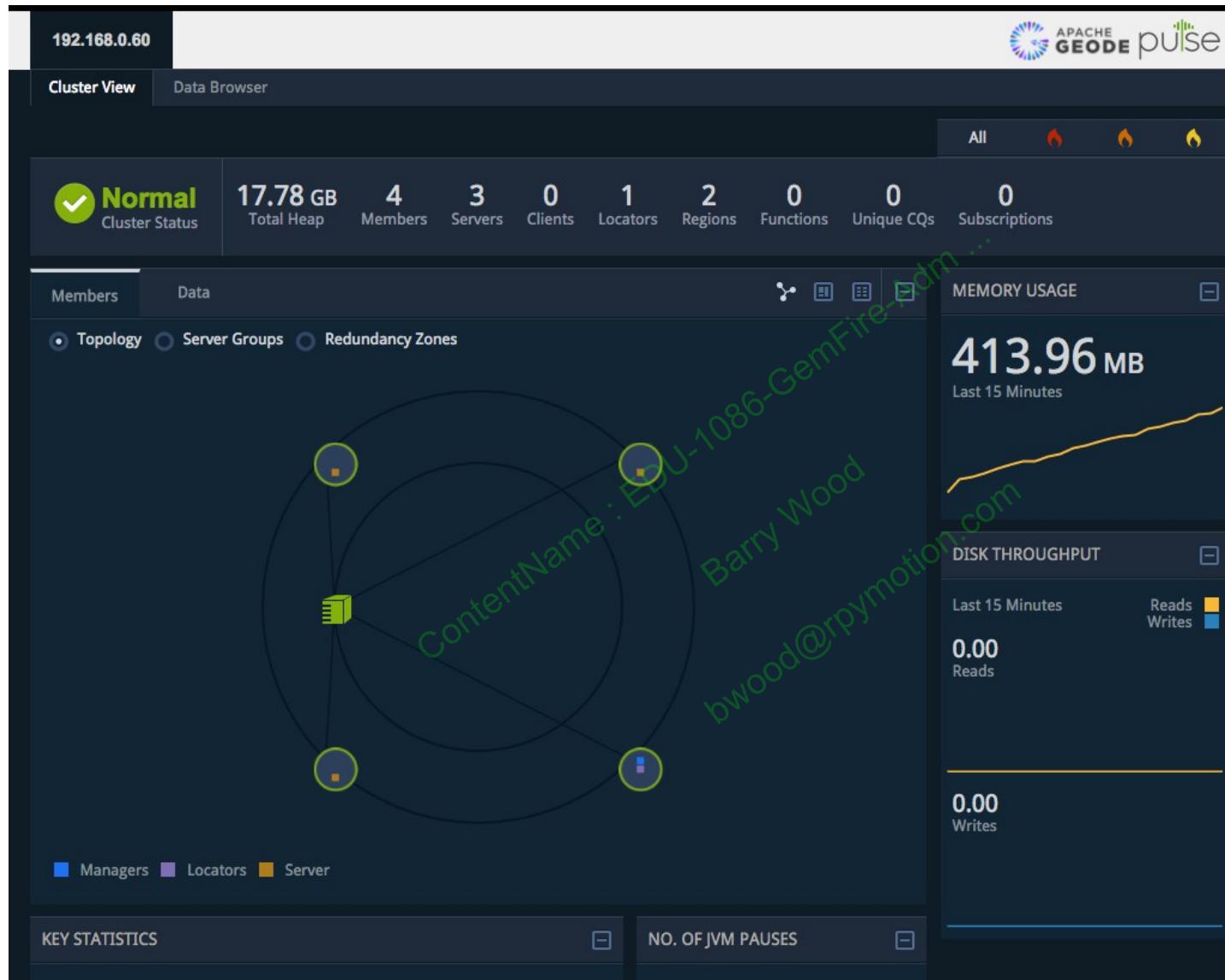
- Launch GemFire Pulse:

```
gfsh> start pulse
```

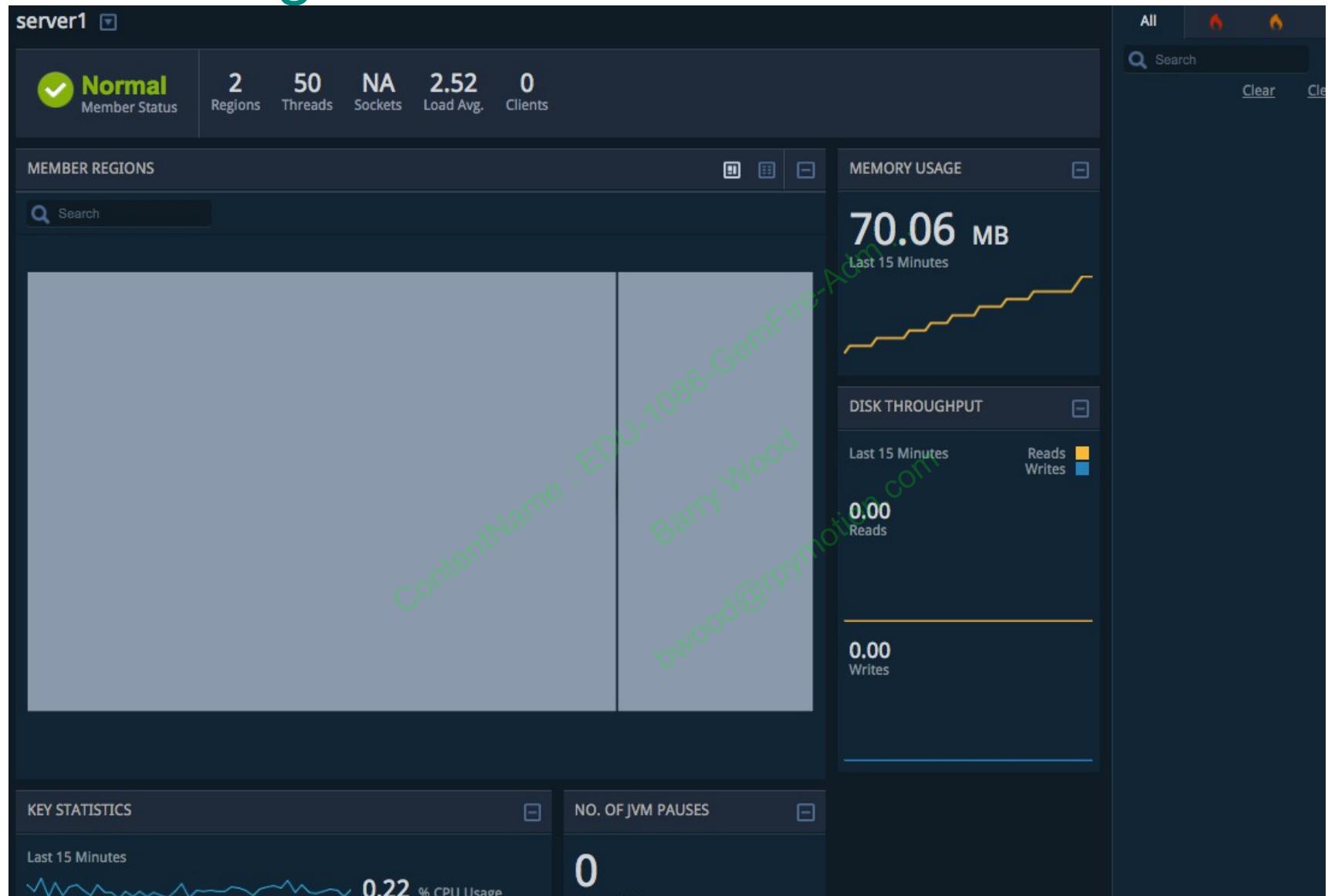
- Default HTTP port is 7070 unless set by http-service-port property (changed from 8080 in 7.0.x)
- Pulse Default Login:
 - Username = admin
 - Password = admin



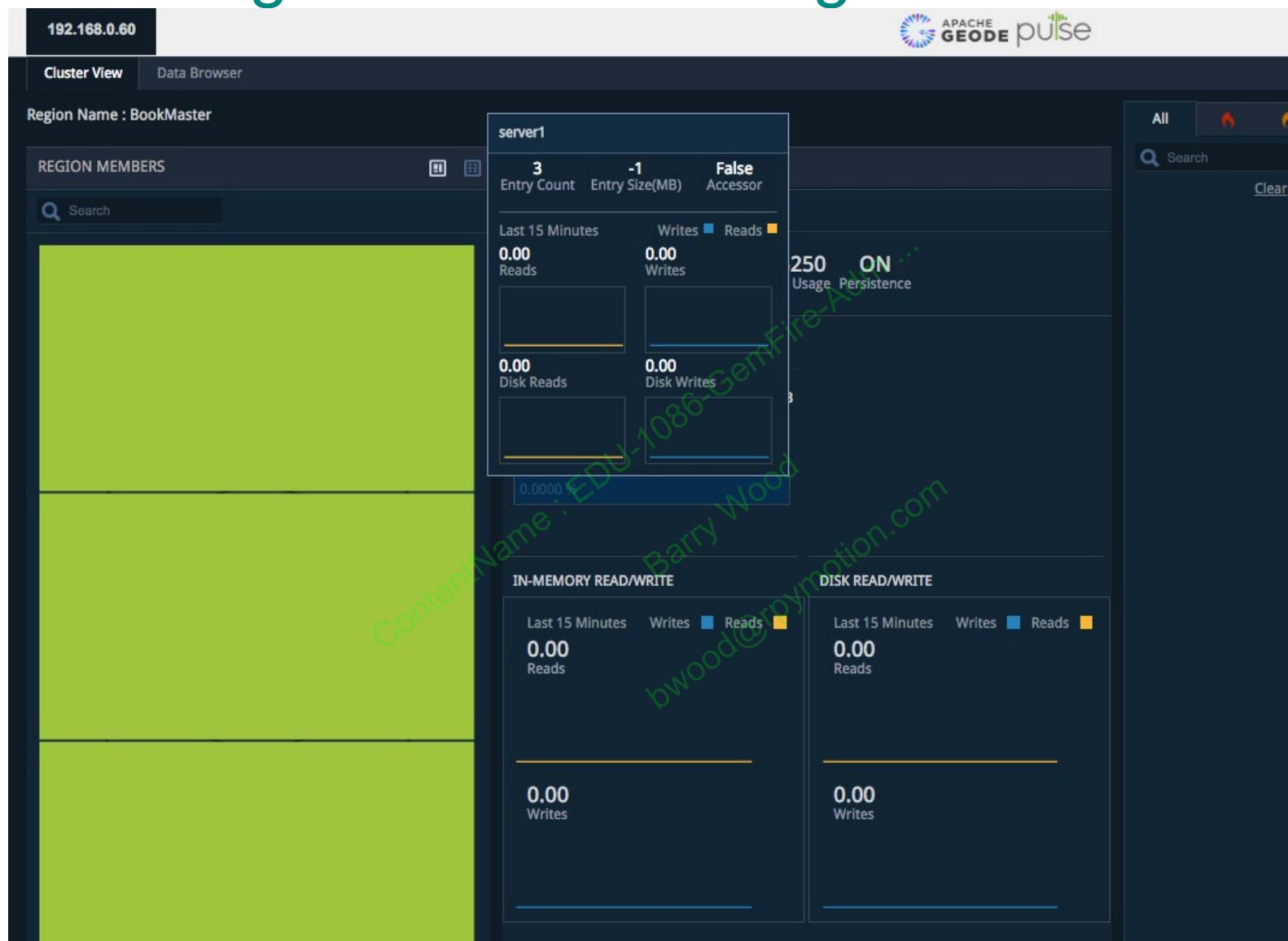
Monitoring – Pulse Cluster View



Monitoring – Pulse Member View



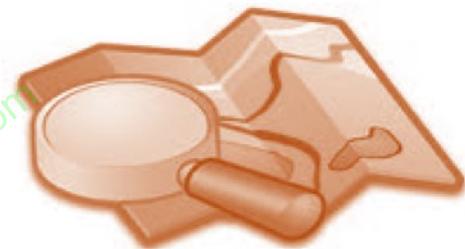
Monitoring – Pulse Data/Region View



Lesson Road Map

- JMX Manager
- Pulse
- **JConsole**

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



JConsole – Connection

- JConsole connection to JMX Manager Node:
 - RMI based connection
 - Can access MBeans of all system members
- JConsole connection to a Managed Node:
 - Only access local MBean of this Managed Node
 - Cannot access the MBeans of other cluster members



Starting JConsole

- 1) First connect to the JMX Manager:

```
gfsh> connect --locator=host3[41111]
```

- or -

```
gfsh> connect --jmx-manager=host3[1099]
```

- 2) Start Jconsole

```
gfsh> start jconsole
```

Message Running JDK Console will appear while the JConsole application is launching

JConsole – Cluster View

Java Monitoring & Management Console

Connection Window Help

service:jmx:rmi://server1/jndi/rmi://server1:1099/jmxrmi

Overview Memory Threads Classes VM Summary MBeans

Attribute values

Name	Value
ActiveCQCount	0
AlertLevel	severe
AverageReads	3.5
AverageWrites	1.0
DiskFlushAvgLatency	0
DiskReadsRate	0.0
DiskWritesRate	0.0
DistributedSystemId	-1
GarbageCollectionCount	592
JVMPauses	6
LocatorCount	2
ManagerObjectName	GemFire:service=Manager,type=Member,member=locator1
MemberCount	6
MemberObjectName	GemFire:type=Member,member=locator1
NumClients	1
NumInitialImagesInProgress	0
NumRunningFunctions	0
NumSubscriptions	0
QueryRequestRate	0.0
RegisteredCQCount	0
SystemDiskStoreCount	0
TotalBackupInProgress	0
TotalDiskUsage	0
TotalHeapSize	6570
TotalHitCount	0
TotalMissCount	56688
TotalRegionCount	1
TotalRegionEntryCount	0
UsedHeapSize	604

Refresh

service:jmx:rmi://server1/jndi/rmi://server1:1099/jmxrmi

JConsole – Cache Server View

The screenshot shows the Java Monitoring & Management Console (JConsole) interface. The title bar reads "Java Monitoring & Management Console". The menu bar includes "Connection", "Window", and "Help". The URL in the address bar is "service:jmx:rmi://server1/jndi/rmi://server1:1099/jmjaxrmi". The tabs at the top are "Overview", "Memory", "Threads", "Classes", "VM Summary", and "MBeans", with "MBeans" being the active tab.

The left pane displays a tree view of MBeans:

- GemFire
 - Distributed
 - Member
 - CacheServer
 - 40401
 - server1
 - Attributes
 - Operations
 - Notifications
 - 40402
 - 40403
 - 40404
 - Locator
 - Manager
 - Region
 - locator1
 - locator2
 - server1
 - Attributes
 - Operations
 - Notifications
 - server3
 - server4
 - server2
 - Attributes
 - Operations
 - Notifications
 - server3
 - server4
 - JMImplementation
 - Tomcat
 - com.sun.management
 - java.lang
 - java.nio
 - java.util.logging

The "server1" node under "40401" is selected, and its "Attributes" tab is active. The right pane shows a table of attribute values:

Name	Value
ActiveCQCount	0
BindAddress	
Capacity	1
ClientConnectionCount	125
ClientNotificationAvgLatency	0
ClientNotificationRate	0.0
ConnectionLoad	0.0
ConnectionThreads	0
ContinuousQueryList	java.lang.String[0]
CurrentClients	1
DiskStoreName	
EvictionPolicy	none
GetRequestAvgLatency	0
GetRequestRate	97.5
HostNameForClients	
IndexCount	0
IndexList	java.lang.String[0]
LoadPerConnection	0.0
LoadPerQueue	1.0
LoadPollInterval	5000
MaxConnections	1000
MaxThreads	0
MaximumMessageCount	500000
MaximumTimeBetweenPings	0
MessageTimeToLive	180
NumClientNotificationRequests	0
NumSubscriptions	0
Port	40401
PutRequestAvgLatency	0
PutRequestRate	48.5
QueryRequestRate	0.0
QueueLoad	0.0

A "Refresh" button is located at the bottom right of the table area.

JConsole – Member View

The screenshot shows the Java Monitoring & Management Console (JConsole) interface. The title bar reads "Java Monitoring & Management Console". The connection URL is "service:jmx:rmi://server1/jndi/rmi://server1:1099/jmxrmi". The menu bar includes "Connection", "Window", and "Help". The tab bar shows "Overview", "Memory", "Threads", "Classes", "VM Summary", and "MBeans", with "MBeans" being the active tab.

The left pane displays a tree view of MBeans:

- GemFire
 - Distributed
 - Region
 - System
 - Member
 - CacheServer
 - Locator
 - Manager
 - Region
 - locator1
 - locator2
 - server1
 - Attributes (selected)
 - Operations
 - Notifications
 - server2
 - server3
 - server4
 - JMImplementation
 - Tomcat
 - com.sun.management
 - java.lang
 - java.nio
 - java.util.logging

The right pane shows the "Attribute values" table:

Name	Value
LoadAverage	0.02
LoadsAverageLatency	0
Locator	false
LockLease	120
LockRequestQueues	0
LockTimeout	60
LockWaitsInProgress	0
LruDestroyRate	0.0
LruEvictionRate	0.0
Manager	false
ManagerCreated	false
MaximumHeapSize	1519
Member	server1
MemberUpTime	34154
Name	server1
NetLoadsAverageLatency	0
NetSearchAverageLatency	0
NumRunningFunctions	0
NumRunningFunctionsHavingResults	0
NumThreads	66
PDXDeserializationAvgLatency	0
PDXDeserializationRate	0.0
PartitionRegionCount	1
ProcessId	2914
PutAllAvgLatency	0
PutAllRate	0.0
PutsAvgLatency	0
PutsRate	0.0
RootRegionNames	java.lang.String[1]
SerializationAvgLatency	2
SerializationRate	0

A green watermark "BarnsbyName : EDU-108-GemFire-Adm ..." is overlaid across the center of the screen.

JConsole – Region View

The screenshot shows the Java Monitoring & Management Console (JConsole) interface. The title bar reads "Java Monitoring & Management Console". The menu bar includes "Connection", "Window", and "Help". The toolbar has icons for Overview, Memory, Threads, Classes, VM Summary, and MBeans. The "MBeans" tab is selected.

The left pane displays a tree view of MBeans:

- GemFire
 - Distributed
 - Region
 - /searchResults
 - Attributes
 - Member
 - CacheServer
 - Locator
 - Manager
 - Region
 - locator1
 - locator2
 - server1
 - Attributes
 - Operations
 - Notifications
 - server2
 - server3
 - server4
 - JMImplementation
 - Tomcat
 - com.sun.management
 - java.lang
 - java.nio
 - java.util.logging

Attribute values

Name	Value
AverageReads	0.0
AverageWrites	0.0
AvgBucketSize	0
BucketCount	0
CacheListenerCallsAvgLatency	0
CacheWriterCallsAvgLatency	0
CreatesRate	0.0
DestroyRate	0.0
DiskReadsRate	0.0
DiskTaskWaiting	0
DiskUsage	0
DiskWritesRate	0.0
EmptyNodes	0
EntrySize	0
FullPath	/searchResults
GatewayEnabled	false
GetsRate	0.0
HitCount	-1
HitRatio	-1.0
LastAccessedTime	-1
LastModifiedTime	-1
LruDestroyRate	0.0
LruEvictionRate	0.0
MaxBucketSize	0
MemberCount	4
Members	java.lang.String[4]
MinBucketSize	0
MissCount	-1
Name	searchResults
NumBucketsWithoutRedundancy	0
ParentRegion	searchResults

Refresh

service:jmx:rmi://server1/jndi/rmi://server1:1099/jmxrmi

Troubleshooting With JConsole - 1

How can I tell if all the members in the system are up and running?

- Check: GemFire/Distributed/System
 - MemberCount
 - LocatorCount
- MemberCount = number locators + number cache servers
- So number cache servers running = MemberCount – LocatorCount

Troubleshooting With JConsole - 2

How can I tell if all the regions are up and the grid is processing traffic?

- Check: GemFire/Distributed/System
 - TotalRegionCount
 - NumClients
 - TotalRegionEntryCount
- TotalRegionCount = number of regions defined in system
- If traffic is moving through GemFire:
 - TotalRegionEntryCount increases as data is stored
 - NumClients will be greater than 0

Troubleshooting With JConsole - 3

How can I tell if a region's redundancy is setup correctly and is working?

- Check: GemFire/Distributed/Region/<region name>
 - BucketCount
 - PrimaryBucketCount
 - NumBucketsWithoutRedundancy
- BucketCount = total number of buckets defined for the region + number of buckets needed to satisfy redundancy
- PrimaryBucketCount = number of buckets for region and the NumBucketsWithoutRedundancy should be 0
- If NumBucketsWithoutRedundancy is > 0 :
 - There are some primary buckets without secondary copies
 - Rebalance should be performed to rebuild secondary copies

Troubleshooting With JConsole - 4

How can I tell if a server is approaching its limit for client connections?

- Check: `GemFire/Member/CacheServer/<port>/<server name>`
 - `ClientConnectionCount`
 - `MaxConnections`
- `ClientConnectionCount < MaxConnections`
- Once the cache server reaches its connection limit, it will start refusing client connections

Troubleshooting With JConsole - 5

How can I tell if a server is approaching its File Descriptor limit?

- Check: GemFire/Member/<server name>
 - TotalFileDescriptorOpen
 - FileDescriptorLimit
- If TotalFileDescriptorOpen ~ FileDescriptorLimit:
 - Server is approaching its file descriptor limit
- If exceeding FileDescriptorLimit
 - Hamper communications
 - Slow down the system
 - Conserve-sockets setting will be ignored

Note: File Descriptor limit should be set around 32000 or more

Interesting Off-heap Metrics

Metrics found on the Member bean attributes

Metric	Description
offHeapFragmentation	Provides the percentage of off-heap fragmentation
offHeapFreeMemory	Provides the size of available off-heap memory (in bytes)
offHeapObjects	Provides the number of off-heap objects
offHeapUsedMemory	Provides size of utilized off-heap memory
offHeapMaxMemory	Provides the total allocated off-heap space
offHeapCompactionTime	Returns the total time spent compacting in milliseconds

Lab – Management Tools

In this lab, you will:

1. Become familiar with Pulse monitor tool
2. Become familiar with JConsole to monitor members

ContentName : EDU-1086-GemFire-Admin
Barry Wood
bwood@rpymotion.com

Review of Objectives

You should be able to do the following:

- Define how JMX supports viewing and monitoring of all members in a distributed system
- Describe differences between a Managed Node and a JMX Manager Node
- Describe views of distributed system provided by Pulse monitoring tool
- Describe how JConsole can be used monitor the distributed system



Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

Network Architecture

Membership, Partitioning, Split-Brain

Content source: EDU-1086-GemFire-Adm ...
By Brian Wood
bwood@rpymotion.com

Objectives

After completing this lesson, you should be able to:

- Understand how membership in distributed system is obtained
- Understand GemFire Failure Detection
- Understand member weighting, quorum and loss percentage
- Understand how network partitioning occurs
- Understand configuration options to reduce network partitioning

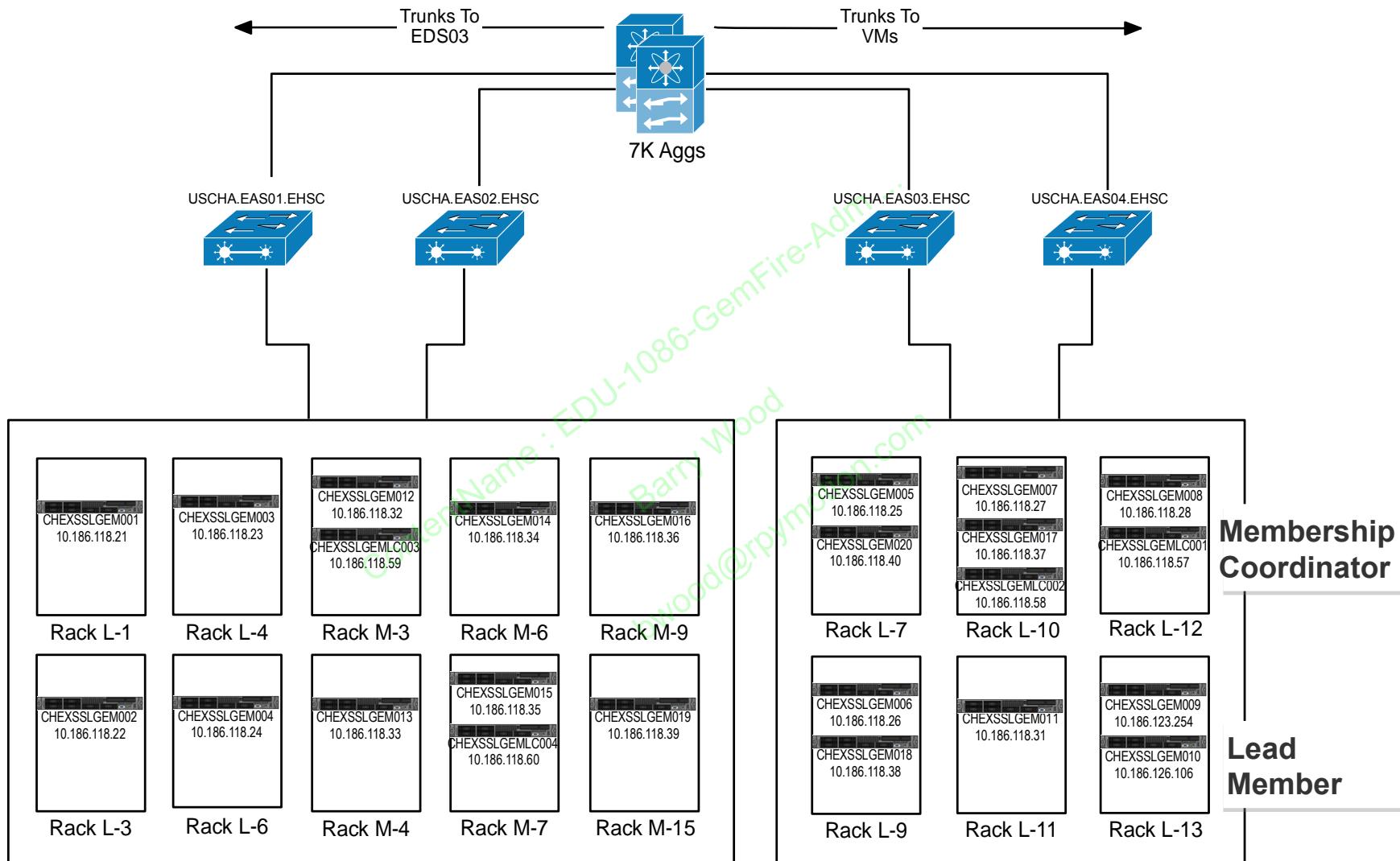


Lesson Road Map

- **Membership Process**
- Network Partitioning
- System Members & Member Weighting
- Configuration Guidelines

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Network Architecture Example



Membership Process - 1

- Distributed system typical startup sequence:
 - First the locators
 - Then cache servers
 - Finally other applications or processes
- Oldest member assumes the role of membership coordinator:
 - Managing entry/exit of members of the distributed system
 - As members startup, they are added to the membership discovery list by the locators
- Before members can join the distributed system:
 - Authenticated by the locator
 - Added to the membership view
 - Membership view distributed to all members of the system

Membership Process - 2

- Each time a member joins or leaves the system:
 - Membership coordinator is alerted
 - New membership view is generated and distributed
- Each time membership coordinator sends membership view:
 - Calculates a current system weight value
 - Compares it to the previous system weight value
- If GemFire detects membership weight drop below 51% (loss of quorum)
 - GemFire will declare a network partition event if network partition detection is enabled
 - membership coordinator sends a network-partition-detected UDP message to all system members
 - closes the distributed system with ForcedDisconnectException

Membership View

The format of each membership view is:

- **address (processId) <vXX>:mem-port/dist-port**
 - membership-port: port used to send membership messages
 - distribution-port: port used for cache messaging

```
[info 2012/01/06 11:44:08.164 PST bridgegemfire1 <UDP Incoming Message  
Handler> tid=0x1f]  
Membership: received new view [ent(5767)<v0>:8700|16]  
[ent(5767)<v0>:8700/44876,  
ent(5829)<v1>:48034/55334, ent(5875)<v2>:4738/54595,  
ent(5822)<v5>:49380/39564,  
ent(8788)<v7>:24136/53525]
```

- **[ent(5767)<v0>:8700|16]** membership coordinator
is always first member in the list

Membership – Failure Detection

- Each member watches the member to its left for failure detection:
 - On peer failure detection it broadcasts a **VERIFY_SUSPECT** to all members
 - Each member responds with **ARE_YOU_DEAD** to the suspect process
 - If the process does not respond once with **I_AM_NOT_DEAD** it is kicked out of the membership list
- It is at this point when the partition detection algorithm is applied and executed

Lesson Road Map

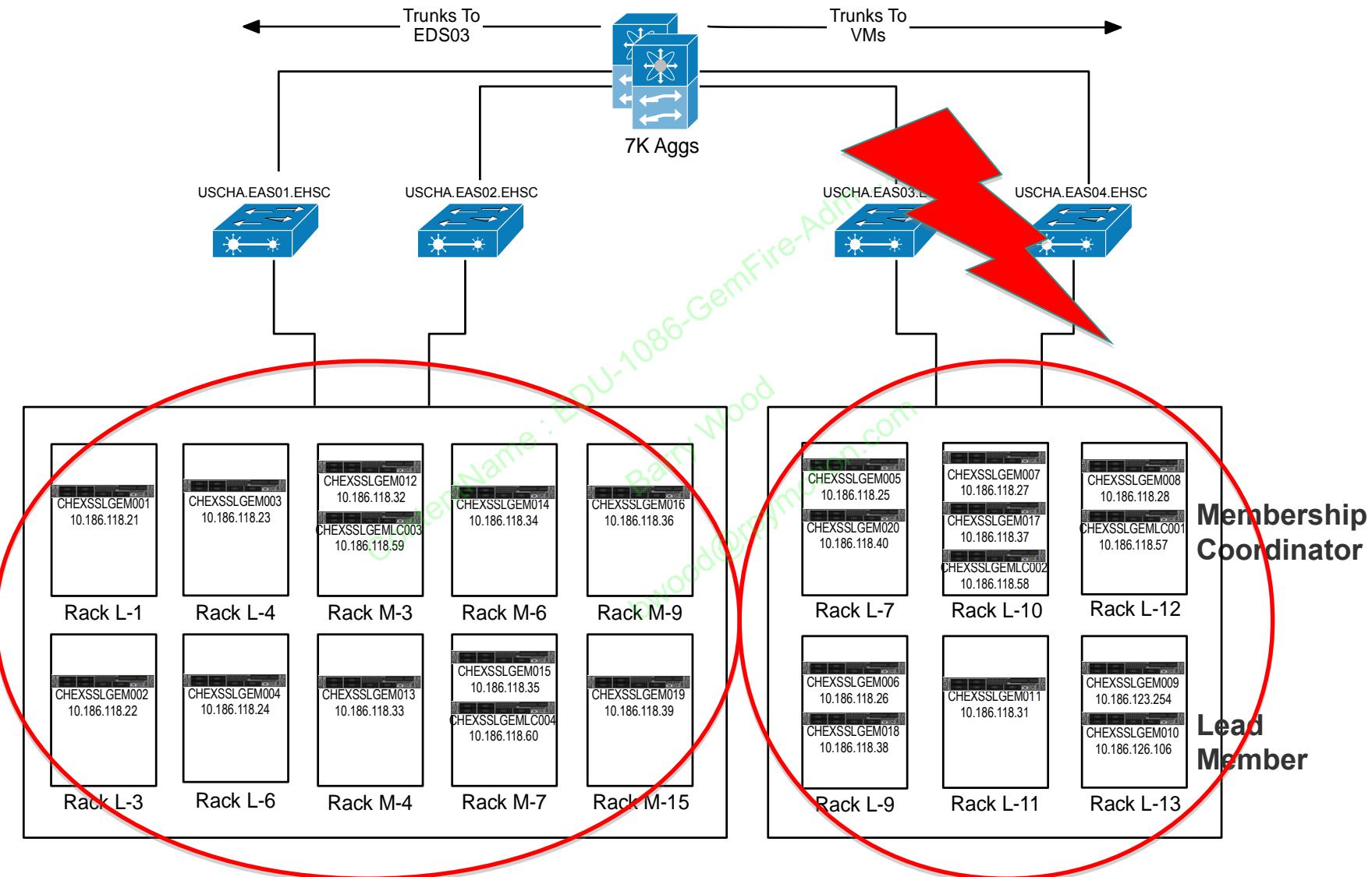
- Membership Process
- **Network Partitioning**
- System Members & Member Weighting
- Configuration Guidelines

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Network Partitioning Management

- Distributed system split into separate running systems:
 - members lose the ability to communicate with each other
 - typically because of network failure
- GemFire uses a failure detection system:
 - based on system member weighting to detect member loss (non-responsive)
 - determine the overall health of the system:
 - Weighted quorum based algorithm used to determine health of the distributed system
- GemFire isolates:
 - failed, unhealthy segments of the system
 - healthy portion of distributed system remains operational

Network Architecture Example



Handling a Network Partition

- Upon a network-partition event:
 - Surviving members try to form a quorum
- If quorum is obtained:
 - New membership coordinator is elected

ContentName : EDU-108
GemFire-Admin
Barry Wood
bwood@rpymotion.com

Network Partitioning

- Split Brain:
 - A distributed system can be split when members lose the ability to see each other
 - Is the P in CAP theorem
 - Consistency, Availability, Partition-Tolerance
 - Members lose the ability to communicate to each other
- Partitioning Causes:
 - Physical network failure
 - Network bandwidth saturation due to load

Lesson Road Map

- Membership Process
- Network Partitioning
- **System Members & Member Weighting**
- Configuration Guidelines

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Membership Coordinator

- Manages entry/exit of members from the distributed system
- Generates membership view
- Calculates total system weight:
 - Based on the health of the system (responsive members)
- Manages failure detection by comparing current and previous membership weight totals
- Signals system members of failure detection events:
 - Issuing network-partition-detected UDP messages
- Membership Coordinator is typically the first Locator to startup, but can be other system members



Lead Member

- Chosen by Membership Coordinator
- Cannot be a Locator
- Member must have network-partitioning-detection enabled
- Typically longest living member that is not a Locator
- Purpose of Lead Member is to provide extra weight to a portion of the system
- Lead Member performs no special functionality

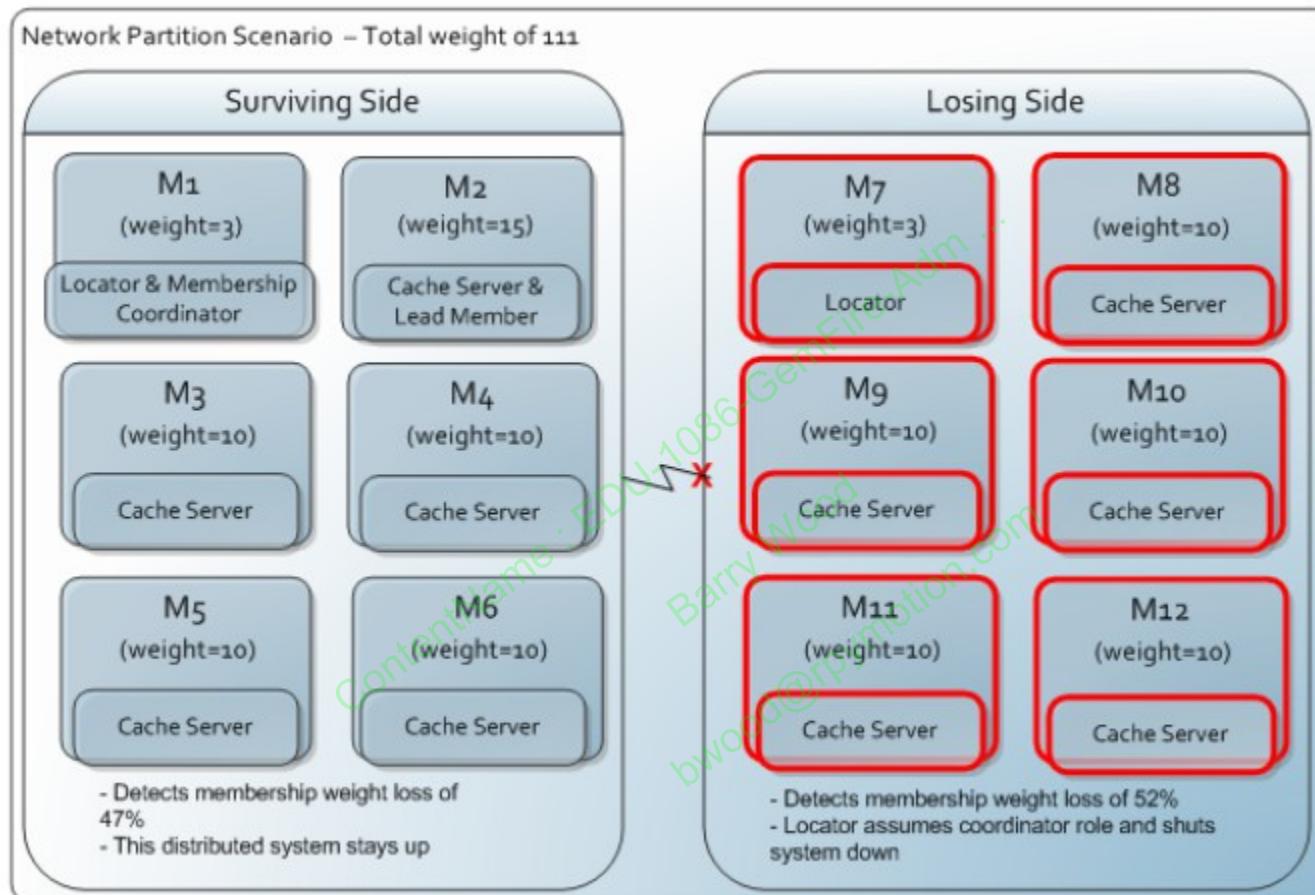
Member Weighting

- Locator has default weight of 3
- Lead Member has default weight of 15.
- All other members have default weight of 10
- Modify a specific member weight in
 - `gemfire.properties`
 - Specify as command line argument
 `-Dgemfire.member-weight`
- Applied at startup

Quorum & Loss Percentage

- Quorum is determined by comparing:
 - The total weight of currently responsive members
 - Previous total weight of responsive members
- Loss percentage threshold is set at 51%
- If Membership Coordinator calculates a 51% loss percentage, then a network-partition-detection event is generated by the Membership Coordinator

Network Partitioning - Example



Network connection between M1-M6 to M7-M12 lose.

Network Partitioning - Example

- Initial membership weighting
 - Member M1-M6 weight = 58
 - Member M7-M12 weight = 53
 - Total system weight (connected) = 111
- After loss of network connection:
 - Membership loss from perspective of M1-M6 = $[(111-58)/111]*100 = 48\%$
 - Membership loss from perspective of M7-M12 = $[(111-53)/111]*100 = 52\%$
- M7-M12 determine too much has been lost (i.e $\geq 51\%$ threshold):
 - A network-partition-detection UDP message is sent to these members
 - Coordinator closes distributed system with ForcedDisconnectException
- M1-M6 are the surviving (operational) members.
 - M1-M6 elect a new Membership Coordinator



Pivotal
GemFire

Version 9.0.4c

Lesson Road Map

- Membership Process
- Network Partitioning
- System Members & Member Weighting
- **Configuration Guidelines**

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Network Partitioning – Configuration Tips

- Use multiple locators for member discovery
- Run locators as standalone & separate from other locators
- Identify & have multiple lead member candidates
- Identify long lived members
- Lead members must not be locators
- Enable partition detection



Network Partition – Configuring Redundancy

- Use locators for membership discovery
- Enable partition detection using the following property
enable-network-partition-detection=true
- Replicate regions should be configured to
DISTRIBUTED_ACK or **GLOBAL** scope
- Assign member-weight a higher value for members that are a critical service
- Consider using Redundancy zones topology to protect from rack level outage



Review of Objectives

You should be able to do the following:

- Describe how membership in the distributed system is obtained
- Describe GemFire Failure Detection
- Describe member weighting, quorum and loss percentage
- Describe how network partitioning occurs
- Describe configuration options to reduce network partitioning

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

Disk Stores and Cache Management

ContentName : EDU-MCS-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Objectives

After completing this lesson, you should be able to:

- Understand the use case for disk store
- Describe how to configure a persistent region
- Understand compaction
- Understand OpLog rolling
- Define meaning of data expiration and data eviction
- Understand use case for Resource Manager

Lesson Road Map

- **Disk Store overview**
- Disk Store usage and configuration
- OpLog rolling and compaction
- Cache Management - expiration, eviction, overflow

ContentName : EDU-101-GemFire-Admin ...
Barry Wood
bwood@rpymotion.com



What is a GemFire Disk Store

- Optional file directory that holds region and event information on disk
 - Members *may not* share directories
 - Use local attached storage for best performance
- Use Cases:
 - Store data to disk to be retrieved after a shutdown/startup cycle
 - Manage memory by “overflowing” data when a region grows too large in memory (much like virtual memory on disk)
 - Client event queues
 - WAN Gateway queues



Region Data – what is written to disk?

- System member status:
 - Running/offline
 - Time stamps
- List of Regions in the disk store, for each Region:
 - Region configuration, load the data quickly on startup
 - Region data operations
 - Region key/value pairs
- Indexes not written to disk

Creating Disk Stores - XML

Default is file size 1GB, default directory is current working directory

```
gfsh> create disk-store --name=persistentStore  
          --dir=/export/persistence/one  
          --compaction-threshold=50 --auto-compact=true  
          --allow-force-compaction=false --max-oplog-size=1024  
          --queue-size=0 --time-interval=100  
          --write-buffer-size=32768
```

Default values



/export/GemFire/persistence/one	-rw-rw-r-- 1 gemfire users	1924 Mar 22 13:57	BACKUPpersistentStore.if
	-rw-rw-r-- 1 gemfire users	0 Mar 22 13:56	DRLK_IFpersistentStore.lk
	-rw-rw-r-- 1 gemfire users	943718 Mar 22 13:57	BACKUPpersistentStore_1.crf
	-rw-rw-r-- 1 gemfire users	104857 Mar 22 13:57	BACKUPpersistentStore_1.drf
	-rw-rw-r-- 1 gemfire users	0 Mar 22 13:57	BACKUPpersistentStore_1.krf

Files in the Persistent Store

Extension	Semantics
.if	Metadata for the disk store – tends to be quite small
.lk	Access control. Only one process can read a disk store at a time. If .lk file is present, that means another process is using it. If shutdown is not graceful, this file may need to be manually deleted.
.crf	Operational log: create, update, and invalidate operations. Pre-allocated to 90% of the max size
.drf	Operational log: delete operations. Pre-allocated to 10% of the max size. Speeds startup
.krf	Key and .crf offset information. Created after the max-oplog-size is reached. For improved performance at startup, but not required.

When the files are closed, they are compacted to their actual size.

Disk Store Considerations

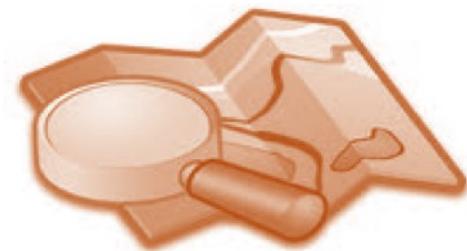
- Ensure you understand where disk stores will be created on all systems
- Will create the same directory(s) on each server's host machine in the same path if absolute paths specified
- Disk store file size (total size of disk data in directory) can be set on the directory

```
gfsh> create disk-store --name=persistentStore  
          --dir=/export/persistence/one#2048
```

Specify storage contents
size to be 2GB

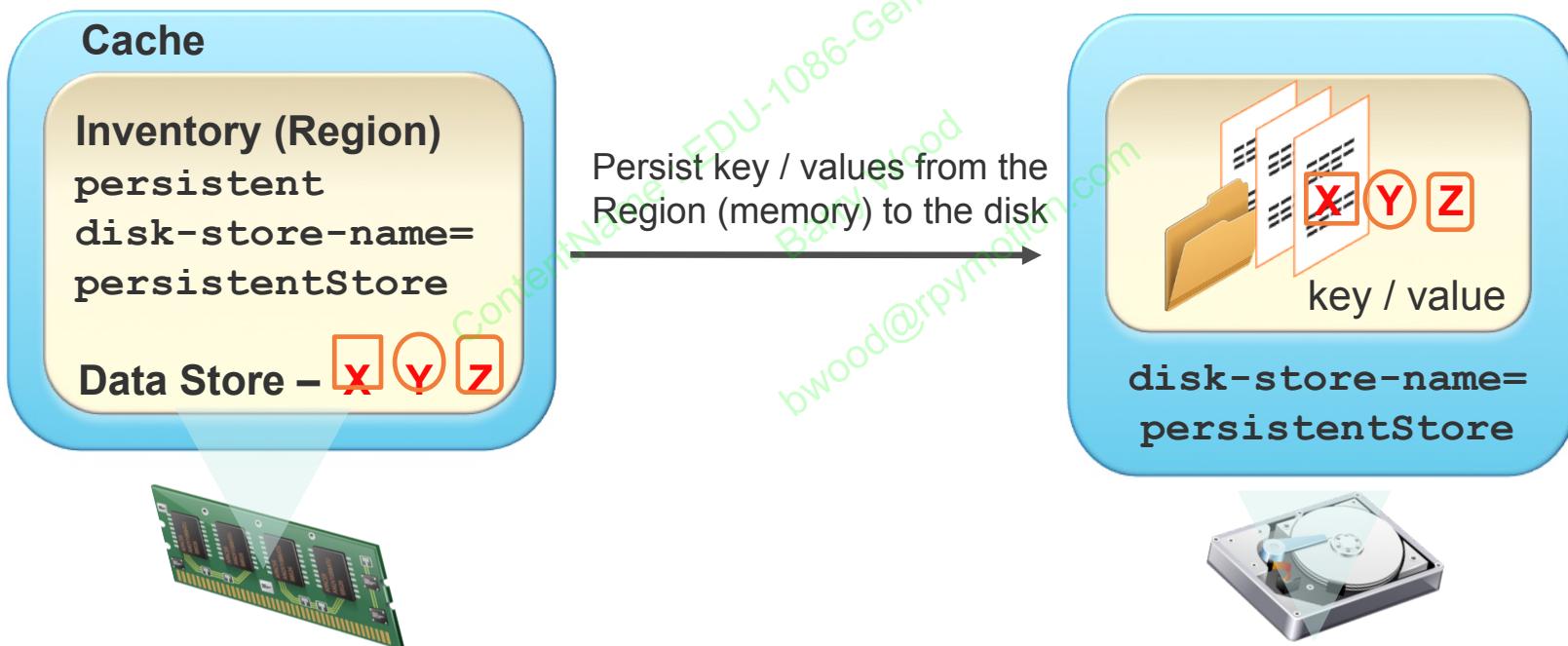
Lesson Road Map

- Disk Store overview
- **Disk Store usage and configuration**
- OpLog rolling and compaction
- Cache Management - expiration, eviction, overflow



Region Persistence

- Provides a disk copy of region entry data
- The keys and values of all entries are saved to disk



Persistent Regions – gfsh Configuration

- Two steps
 - Define one or more disk-stores
 - Specify a disk store for each persistent region
 - *Region must be explicitly configured with a disk-store*

```
gfsh> create disk-store --name=persistentStore...
```

```
gfsh> create region --name=Inventory  
          --type=REPLICATE_PERSISTENT  
          --disk-store=persistentStore  
          --enable-synchronous-disk=false
```

Define disk-store

Don't forget to
use it!

Persistent Regions – XML configuration

- Process is very similar
 - Define one or more disk-stores
 - Specify a disk store for each persistent region

```
<cache>
    <disk-store name="persistentStore">
        ...
    </disk-store>

    <region name="Inventory">
        <region-attributes refid="REPLICATE_PERSISTENT"
            disk-store-name="persistentStore"
            disk-synchronous=false>
        </region-attributes>

    </region>
</cache>
```

Define disk-store
(Can be defined via gfsh)

Don't forget to
use it!

Persistent Regions – Details

- Persistence is about recovery, if member stops:
 - Persistent region data on disk
 - Recovered at next startup
- Synchronous writes: **disk-synchronous=true (xml)**
enable-synchronous-disk=true (gfsh)
 - Blocks on writing disk
 - Not guaranteed anyway due to OS buffering
 - Worst performance
- Not the *only* source of recovery data
 - Other members also hold replicated data

Disk Store Characteristics

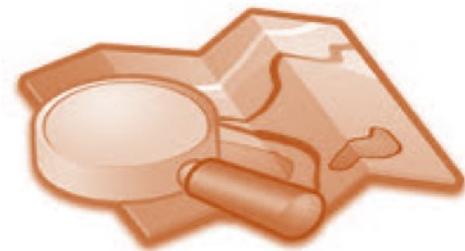
- Persistent
 - Local disk storage for each node, accessible to persist data
 - Graceful shutdowns - ensure data persisted
- Applicable to:
 - Replicated regions
 - Partitioned regions
- To ensure files are check-pointed and written:

```
gfsh> shutdown
```



Lesson Road Map

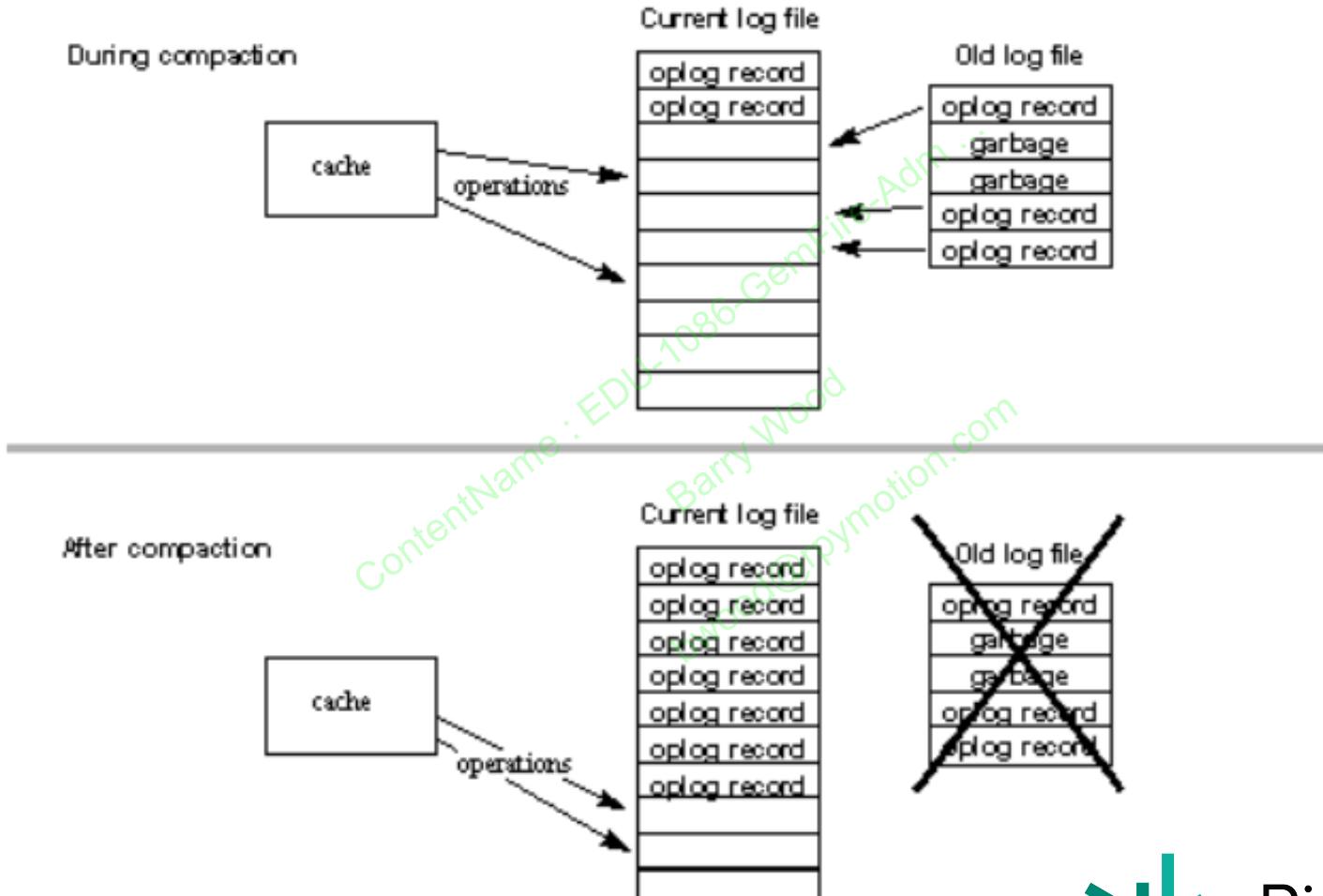
- Disk Store overview
- Disk Store usage and configuration
- **OpLog rolling and compaction**
- Cache Management - expiration, eviction, overflow



Disk Stores - Compaction

- Compaction
 - Process of cleaning up data in the OpLogs
 - Active objects copied into new set of OpLogs
 - Trigger - percentage garbage threshold
 - Background process, consumes resources
- OpLog roll required before compaction
- Compaction forces OpLog roll
- Auto-compact in production not recommended

Disk Stores - Compaction



Online Compaction – Automatic Compaction

- Trigger: *garbage > compaction-threshold*
- Non-garbage records are added to the current OpLog
- Does not block any disk store operations
- Not recommended for production
- Compaction-threshold determines OpLogs compacted

```
gfsh> create disk-store --name=persistentStore  
      --dir=/export/GemFire/persistence/one
```

- Defaults:
 - Auto compact on
 - Allow force compact off
 - Compaction threshold set to 50%

Online Compaction – Manual Compaction

- Manual compaction:
 - Run from `gfsh` command line or via Java API
 - Must also set `allow-force-compaction="true"`
- Compaction only performed if *garbage % > compaction-threshold*

```
gfsh> create disk-store --name=persistentStore  
      --compaction-threshold=50  
      --dir= ...  
      --auto-compact=false --allow-force-compaction=true
```

```
gfsh> compact disk-store --name=persistentStore
```

OpLog Roll

- OpLog Roll trigger:
 - OpLog becomes full, it is closed automatically
 - Automatic or manual compaction
 - Through Java API call
- OpLog Roll:
 - New set files created with the next sequence number
 - Sequence number higher than any existing number
 - Does not remove old file
- Compaction:
 - Moves active objects from closed files to current file
 - Removes closed files

Disk Stores - Rollover

```
gfsh> create disk-store --name=persistentStore  
      --dir=/export/GemFire/persistence/one  
      --dir=/export/GemFire/persistence/two
```



/export/GemFire/persistence/one

```
-rw-rw-r-- 1 gemfire users 1924 Mar 22 13:57 BACKUPpersistentStore.if  
-rw-rw-r-- 1 gemfire users 0 Mar 22 13:56 DRLK_IFpersistentStore.lk  
-rw-rw-r-- 1 gemfire users 939083 Mar 22 13:57 BACKUPpersistentStore_1.crf  
-rw-rw-r-- 1 gemfire users 48952 Mar 22 13:57 BACKUPpersistentStore_1.drf  
-rw-rw-r-- 1 gemfire users 20566 Mar 22 13:57 BACKUPpersistentStore_1.krf
```



/export/GemFire/persistence/two

```
-rw-rw-r-- 1 gemfire users 943718 Mar 22 13:57 BACKUPpersistentStore_2.crf  
-rw-rw-r-- 1 gemfire users 104857 Mar 22 13:57 BACKUPpersistentStore_2.drf
```

Files close with true size when they reach max-size

New files are pre-allocated

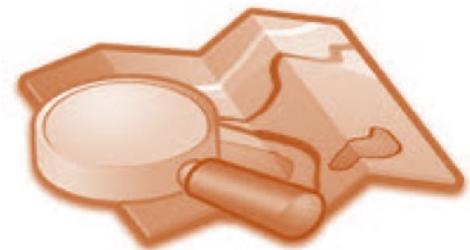
File number increments

Options for Persistence

Option	Semantics
allow-force-compaction	Default is false. Determines whether manual compactions are allowed.
auto-compact	Default is true. Allows auto compaction when the logs are filled with a defined amount of garbage (compaction-threshold)
compaction-threshold	Default is 50. The percentage of garbage in log files before auto compaction is done.
max-oplog-size	Default is 1024. Largest size, in megabytes, an operation log can become before automatically rolling to new file.
name	Name of this disk store.
queue-size	Default is zero. The maximum number of operations to allow into the write queue before automatically flushing the queue.
time-interval	Default is 1000. The number of milliseconds that can elapse before data is flushed to disk.
write-buffer-size	Default is 32768. Size of the buffer used to write to disk.

Lesson Road Map

- Disk Store overview
- Disk Store usage and configuration
- OpLog rolling and compaction
- **Cache Management - expiration, eviction, overflow**



Cache Management Overview

- Cache Management is about ways to manage the memory occupied by a region's entries.
- Two key mechanisms provided by GemFire
 - Expiration – *proactive* management
 - Expire based on time to live
 - Expire based on idle time
 - Eviction – *reactive* management
 - Eviction makes room by removing *Least Recently Used* (LRU) entries
 - Eviction can be based on memory or entry count thresholds
- Both result in objects removed from Cache memory



Data Expiration

- Expiration removes old entries
 - Destroy or Invalidate entries
 - Optimizes memory consumed by Region
- Expiration activities applicable to:
 - Distributed regions - server
 - Local cache - client
- Expiration types:
 - Time to live (TTL)
 - Idle timeout (not used for this period)

Configuring Expiration

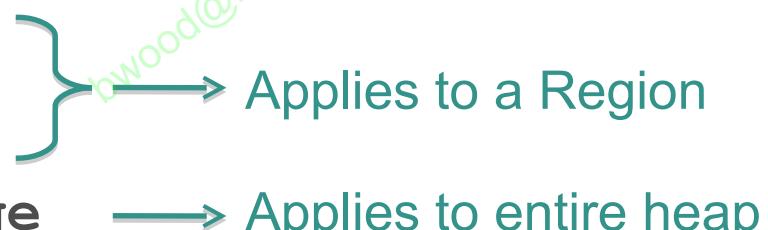
Set the region's enable-statistics attribute to true.

```
gfsh> create region --name=Customer --type=REPLICATE  
      --enable-statistics=true  
      --entry-idle-time-expiration=60  
      --entry-idle-time-expiration-action=invalidate
```

```
<cache>  
  <region name="Customer">  
    <region-attributes statistics-enabled="true">  
      <entry-idle-time>  
        <expiration-attributes timeout="60" action="invalidate"/>  
  
      </entry-idle-time>  
    </region-attributes>  
  </region>  
</cache>
```

Set the expiration attributes by expiration type, with the max times and expiration actions

Data Eviction Overview

- Use eviction to control how much heap your regions use
 - Controls data region size by removing *least recently used* (LRU) entries to make way for new data
 - Evicted entries are either destroyed or paged to disk
 - `local-destroy`
 - `overflow-to-disk`
 - Eviction algorithms:
 - `lru-entry-count`
 - `lru-memory-size`
 - `lru-heap-percentage`
- 
- Applies to a Region
- Applies to entire heap

Data Eviction – How it Works

- **lru-entry-count, lru-memory-size**
 - Eviction happens when data region reaches a specified entry count or size
 - *Implication:* Total sum of all regions may still cause an **OutOfMemory** exception
- **lru-heap-percentage**
 - Causes all regions to simultaneously evict to create space
 - Can ensure an **OutOfMemory** exception never happens
 - *Implication:* clients may be unable to create data
 - Requires a *Resource Manager* to co-ordinate

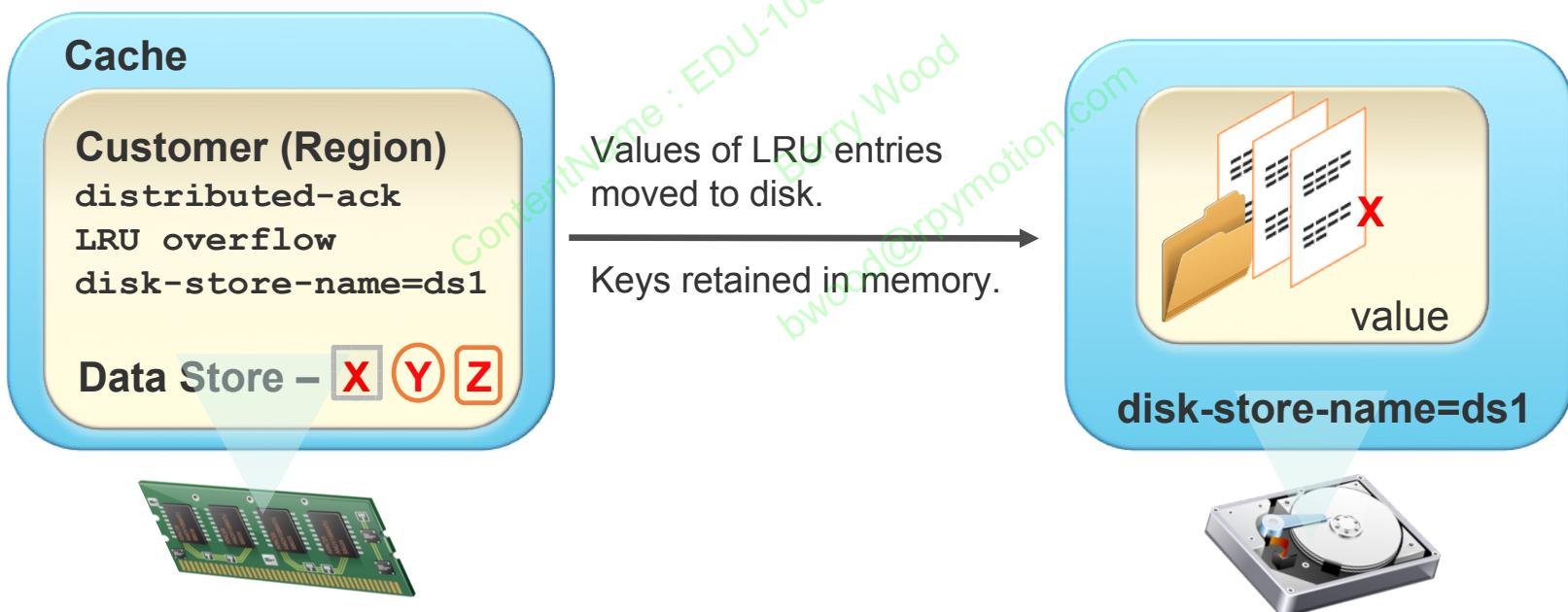
Note: eviction can't currently be configured from gfsh

Partitioned Regions and Data Eviction

- Region Size or Count Eviction
 - Oldest entry in the bucket evicted where the new entry operation is being performed
 - LRU is maintained on a bucket-by-bucket basis
 - Not maintained across the partitioned regions
 - Performance would be impacted
- Heap Eviction:
 - Each bucket treated as a separate region
 - Eviction on the bucket, not the region as a whole

Region Overflow

- Overflow limits region memory size
 - Move LRU entries to disk (invalidating in memory)
 - Persisted entries, key/value are on disk for recovery and can be queried



Configuring Overflow to Disk

- Eviction occurs when total size of all objects in region exceeds maximum
 - Gemfire makes “best-guess” as to object sizes

```
<region name="Customer">
  <region-attributes disk-store-name="ds1" ... >
    <eviction-attributes>
      <lru-memory-size maximum="512" - Maximum in Mbytes
        action="overflow-to-disk"/>
    </eviction-attributes>
  </region-attributes>
</region>
```

cache.xml



Configuring Eviction by Entry Count

```
<cache>
  <region name="Customer">
    <region-attributes>
      <eviction-attributes>
        <lru-entry-count maximum="1000"
          action="local-destroy"/>
      </eviction-attributes>
    </region-attributes>
  </region>
</cache>
```

cache.xml

You specify the number of entries to keep in the region. If you try to put more in then the oldest entries are evicted.



Pivotal
GemFire

Resource Manager

- Manages overall memory in a cache server, not just a Region
- Use in applications at risk of running out of memory
- **Eviction threshold**
 - Evictions for all regions with eviction-attributes set to Iru-heap-percentage
 - JVM garbage collector removes evicted data
 - Evictions continue until heap usage is below the threshold
 - Operation that add to heap, for each new entry : an entry of equal size must be evicted
- **Critical threshold** – set above the eviction threshold
 - Operations that add to heap refused
 - Allow the garbage collector to catch up
 - Clients may receive a **LowMemoryException** - retry

Heap LRU Eviction with Resource Manager

```
<cache>
  <region name="Customer">
    <region-attributes refid="REPLICATE_HEAP_LRU">
      <eviction-attributes>
        <lru-heap-percentage action="local-destroy" />
      </eviction-attributes>
    </region-attributes>
  </region>

  <resource-manager critical-heap-percentage="85"
                     eviction-heap-percentage="70" />
</cache>
```

Note: Resource manager options can also be configured from gfsh using options when starting the server

cache.xml



Version 9.0.4c

Configuring the Resource Manager

- Configure GemFire for LRU heap management
 - Set **critical-heap-percentage** threshold
 - As near to 100 as possible *without* getting OutOfMemory
 - Default is zero – no threshold
 - Set **eviction-heap-percentage**
 - Lower than critical, 50% is a good starting point
 - Region(s) eviction *must* be set to **lru-heap-percentage**
 - Set JVM GC tuning parameters (next slide)
- Monitor, Tune, Test!!!!

Garbage Collector and Resource Manager

- Enables control of JVM heap usage
 - Protects VM from hangs and crashes due to memory overload by implementing garbage collections (GC) to evict old data
- Uses **EvictionHeapPercentage** and **CriticalEvictionHeapPercentage** threshold
- Monitor performance via VSD

Configures Java VM for heap LRU eviction

```
java -Xms512M -Xmx512M -XX:+UseConcMarkSweepGC  
      -XX:CMSInitiatingOccupancyFraction=N
```

where **N < EvictionHeapPercentage**

Resource Manager for Off-heap

- CriticalOffHeapPercentage
 - triggers LowMemoryException for puts into **off-heap** regions
 - critical member informs other members that it is critical
- EvictionOffHeapPercentage
 - triggers eviction of entries in **off-heap** regions configured with LRU_HEAP
- Semantics the same as with the equivalent heap thresholds

Startup Options for Off-Heap Resource Manager

- Java Heap:
 - critical-heap-percentage
 - eviction-heap-percentage
- Off-Heap
 - critical-off-heap-percentage
 - eviction-off-heap-percentage
- Example

```
start server --initial-heap=10G --max-heap=10G  
          --off-heap-memory-size=200G --lock-memory=true  
          --critical-off-heap-percentage=99
```

CorbettName.EDU-1086-GemFire-Adm ...
BarWood
bwood@rpymotion.com

Lab – Persistent Cache Management

In this lab, you will:

1. Create a disk store
2. Configure a region to use disk store for persistence
3. Test startup of a persistent region
4. Configure cache eviction

ContentName : EDU-103 GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Review Objectives

You should be able to do the following:

- Describe use case for using disk store..
- Describe how to configure a persistent region
- Describe how compaction works
- Describe Oplog rolling
- Define meaning of data expiration and data eviction
- Describe use case for using Resource Manager



Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Darryl Wood
bwood@rpxmotion.com

Backup and Restore

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Objectives

After completing this lesson, you should be able to:

- Understand differences between full and incremental backup
- Understand how to perform an online full and incremental backup, along with how to restore a backup
- Understand offline operations such as validating a disk store, compaction of a disk store, synchronizing disk store & cache, and handle missing disk stores
- Understand purpose of cache and region snapshots, along with how to export/import a region snapshot using gfsh command line

Lesson Road Map

- **Full backup vs Incremental backup**
- Online backup and restore
- Offline backup and restore
- Snapshots overview

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Full Backup

- For each persisted member, a full backup includes:
- Disk store files for all persisted region data
- Files and directories configured in `cache.xml` file to be backed up using `<backup>` element, recommended files:
 - Deployed JAR files
 - `gemfire.properties`
 - `cache.xml`
 - Restore scripts

Disk stores can be created via gfsh and persistent regions can be defined via gfsh

```
<cache>
  ...
  <backup>./systemConfig/gf.jar</backup>
  <backup>user/document/user.doc</backup>
</cache>
```

cache.xml

Incremental Backup

- Saves differences between last backup and current data
- Saves data only from operation logs
 - Deployed JAR and configuration files not saved
- Files cannot be compacted
 - Requires file sequence for restoring backup incremental files
- Consumes less system resources than a full backup

Lesson Road Map

- Full backup vs Incremental backup
- **Online backup and restore**
- Offline backup and restore
- Snapshots overview

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Online Backup – Overview

- Online backup creates a backup of all online disk stores
- Each member with persistent data creates a backup of its own configuration and disk stores
- Online backup does not block system or disk activities
 - Backup does consume system resources
- Online backup must be invoked through gfsh command line
 - Use of operating system file commands to perform a backup will result in incomplete and non-usuable files



Preparation for Online Backup

- Optional Compaction operation
 - Reduces amount of data backed up from disk store
- Configure **<backup>** elements to backup:
 - Configuration files, **cache.xml**, **gemfire.properties**
 - Application JAR files
 - Script files
- Backup to SAN or directory that all members can access
 - Ensure directory permissions allow access
- **gemfire.properties** file must be in directory from which gemfire backup commands are run
- Make sure all members to be backed up are running
 - Offline members are not backed up



Full Online Backup Procedure

- To run manual compaction:

```
gfsh> compact disk-store --name=MyDiskStore
```

- Run gfsh backup command, providing backup directory:

```
gfsh> backup disk-store --dir=/home/customer/backup
```

- Backup tool reports status of backup:

```
The following disk stores were backed up:  
DiskStore at hosta.myProject /home/customer  
The backup may be incomplete. The following disk stores  
are not online:  
DiskStore at hostb.myProject /home/inventory
```

- Backup files can be validated offline:

```
gfsh> validate offline-disk-store --name=MyDiskStore \  
      --disk-dirs=/home/customer/backup
```

Incremental Online Backup Procedure

- Run gfsh backup command:
 - dir : backup directory
 - baseline-dir: directory containing backup to compare current disk store
 - If baseline data is not found, a full backup is performed

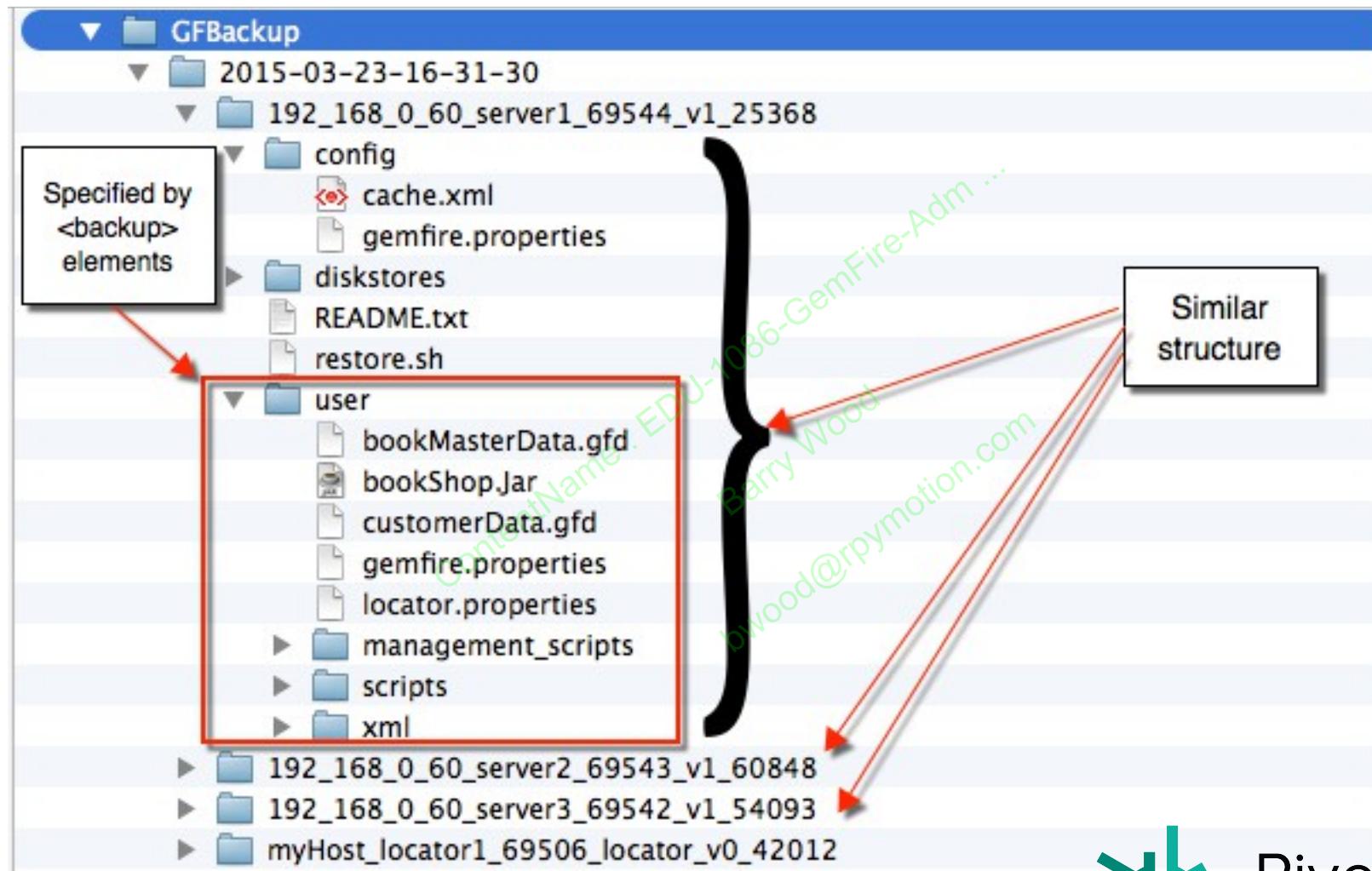
```
gfsh> backup disk-store --dir=/home/customer/backup  
--baselineDir=/home/customer/backup/yyyy-mm-dd-hh-mm-ss
```

- Backup tool reports status of backup:

```
The following disk stores were backed up:  
DiskStore at hosta.myProject /home/customer  
The backup may be incomplete. The following disk stores  
are not online:  
DiskStore at hostb.myProject /home/inventory
```

- Validation of backup file only for full backups

Directory Structure for Disk Store Backup



Pivotal
GemFire

Configuring Files for Backup

```
<cache>
  <disk-store name="ServerPersistence">
    <disk-dirs>
      <disk-dir>/export/GemFire/persistence/one</disk-dir>
    </disk-dirs>
  </disk-store>
  <region name="Customer" ...>
  <region name="BookMaster" ...>

  <backup>../bookMasterData.gfd</backup>
  <backup>../bookShop.jar</backup>
  <backup>../customerData.gfd</backup>
  <backup>../gemfire.properties</backup>
  <backup>../locator.properties</backup>
  <backup>../management_scripts</backup>
  <backup>../scripts</backup>
  <backup>../xml</backup>
</cache>
```

Restore Online Backup

- No `gfsh` commands for restoring backup files
- Backup generates a restore script that will restore files
 - Uses file system commands for restoration
 - Members must be offline
- Restore should copy disk store backup files to stores with persistence data
 - Be careful copying `gemfire.properties` and `cache.xml` files due to possible configuration changes
- Restore should copy any file defined in `cache.xml`
`<backup>`
- Incremental backups restored oldest to newest

Lesson Road Map

- Full backup vs Incremental backup
- Online backup and restore
- **Offline backup and restore**
- Snapshots overview

ContentName : EDU-1056-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Offline - Validating a Disk Store

- Run against offline disk store
 - Applicable to full backup files only
 - Verifies the health of your offline disk-store
 - Region information:
 - Total entries
 - Number of records that could be removed by compaction
- Use case:
 - Before compaction, determine amount reduction
 - Before restoring or modifying a disk store

```
gfsh> validate offline-disk-store --name=MyDiskStore  
      --disk-dirs=/export/GemFire/persistence/backupDir
```

Offline Compaction

- Compaction can be performed offline as a manual process
 - All OpLog files are compacted, regardless of file garbage level
 - New OpLog files are created
 - Must specify directories to compact: **--disk-dirs**

```
gfsh> compact offline-disk-store --name=MyDiskStore  
      --disk-dirs=/export/GemFire/persistence
```



Offline - Synchronizing Disk-Store & Cache

- Offline synchronization, **alter disk-store** command
 - Update disk store to match cache.xml configuration
 - Reduce memory usage
 - Reduce startup time
 - Disk-store synchronizes with cache.xml at startup
- List disk-store modifiable settings and current values
 - Use the alter command with no action specified

```
gfsh> alter disk-store --name=MyDiskStore --region=Customers  
      --disk-dirs=/export/GemFire/persistence/backup
```

- Remove region Customers from disk-store MyDiskStore

```
gfsh> alter disk-store --name=MyDiskStore --region=Customers  
      --disk-dirs=/export/GemFire/persistence/backup --remove
```

Handling Missing Disk Stores

- Online gfsh command to show missing disk stores
 - Lists all disk stores being waited on by other members
- Replicated regions
 - List regions preventing other members from starting
- Partitioned region
 - List offline disk store
 - Prevent system meeting redundancy requirements
- List disk stores being waited on by other members:

```
gfsh> show missing-disk-stores
```

- Revoke disk store, find latest available data and start:

```
gfsh> revoke missing-disk-store  
      --id=id_from_show_missing-disk-store
```

Restore Offline Backup

- No gfsh commands to restore backup files
- Use file system commands to perform restoration of backup files
- Members must be offline to restore backup files
- Restore scripts are typically used – but you must write
 - Copy backup files to disk-stores with persistence data
 - Copy any additional files to appropriate locations
- Restore incremental backup files from oldest to newest

Lesson Road Map

- Full backup vs Incremental backup
- Online backup and restore
- Offline backup and restore
- **Snapshots overview**

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Cache and Region Snapshots

- Cache snapshot can only be performed via Java API
 - Saves entire cache data for reload
- Region snapshot can be performed via gfsh or Java API
 - Saves region data for reload
- Snapshot data use cases
 - Populating QA or Development system
 - System troubleshooting
- Does not guarantee data consistency
 - Concurrent operations may cause snapshot data inconsistency
- Export of snapshot requires additional 10MB heap memory overhead per member



Exporting Region Snapshot via gfsh

- Export/save region snapshot data – option details
 - region** Region to be saved
 - file** File name export must end in ‘.gfd’
 - member** Name of member hosting region data to be exported – will also act as host for export

```
gfsh> export data --region=regionName  
          --file=fileName.gfd  
          --member=memberName
```



Importing Region Snapshot via gfsh

- Import/load region snapshot data – option details
 - region** Region to be loaded
 - file** File to import from
 - member** Member hosting region data – also acts as host for import

```
gfsh> import data --region=regionName  
          --file=fileName.gfd  
          --member=memberName
```

Lab – Backup and Restore

In this lab, you will:

1. Configure persisted members to perform full backup
2. Use gfsh to run manual compaction
3. Use gfsh to perform online backup
4. Take members offline
5. Restore online backup files
6. Use gfsh to verify region entries

Review of Objectives

You should be able to do the following:

- Describe differences between full and incremental backup
- Describe procedure to perform both full and incremental backup
- Describe procedure to restore backup files
- Describe use case for offline operations such as disk store validation, compaction of disk store, synchronization of disk store & cache, and how to handle missing disk stores
- Describe how to export/import region snapshots

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

GemFire Security

ContentName : EDU986-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Objectives

After completing this lesson, you should be able to:

- Explain security model in GemFire
- Configure authentication and authorization for the cluster including
 - gfsh / JMX Manager
 - Servers
 - Clients
- Configure a security post-processor
- Configure SSL communication

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
BWood@rpymotion.com

GemFire Security - Motivations

- Data stored in regions may need to be secure
 - no unauthorized access
 - Some or all data may be sensitive
- Some operations should be restricted
 - Only authorized users may change regions or region data
 - Only authorized members may join the cluster
 - Only authorized users may perform certain administrative operations
- Data transported across the network needs to be secure

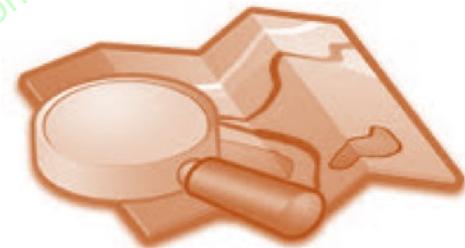
GemFire Security Overview

- Authentication
 - Resolve identity – represented as a ‘Principal’
- Authorization
 - Resolve roles and permissions
- Post Processing
 - Resolve final value presented back to user
- SSL Communication
 - Enable secure transportation of data

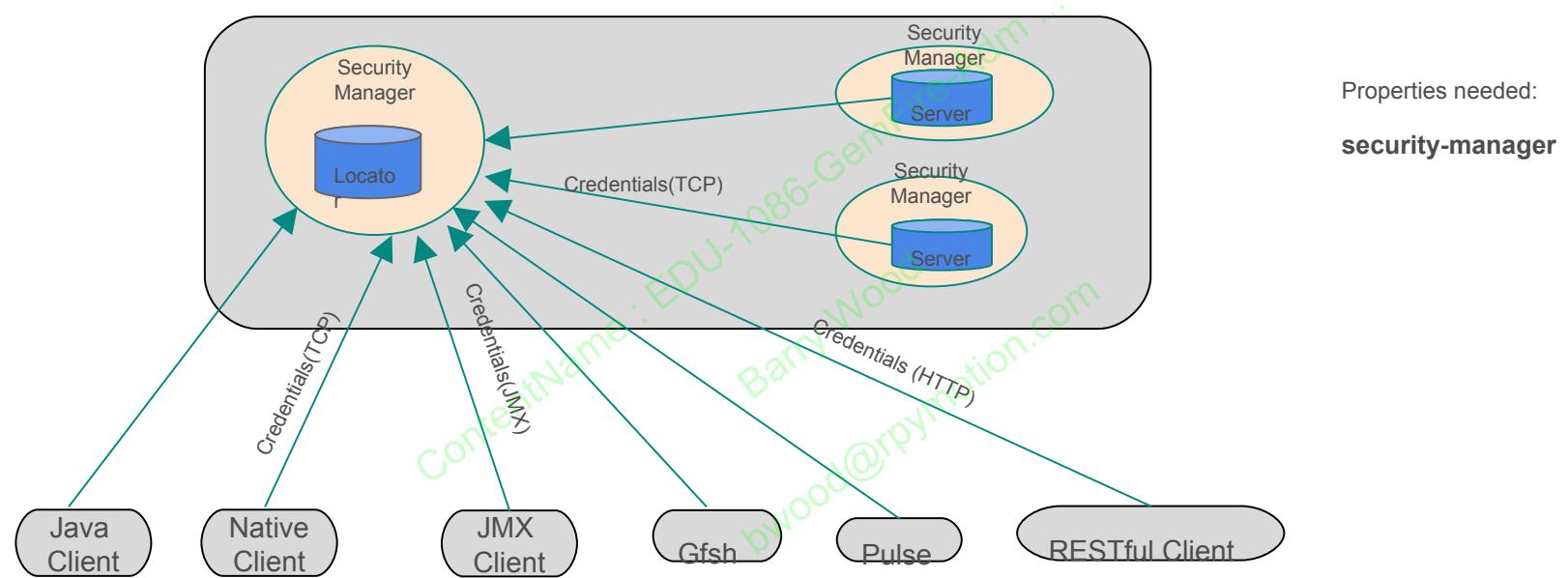
Lesson Road Map

- **Configuring Authentication & Authorization**
- Post Processing
- Configuring SSL

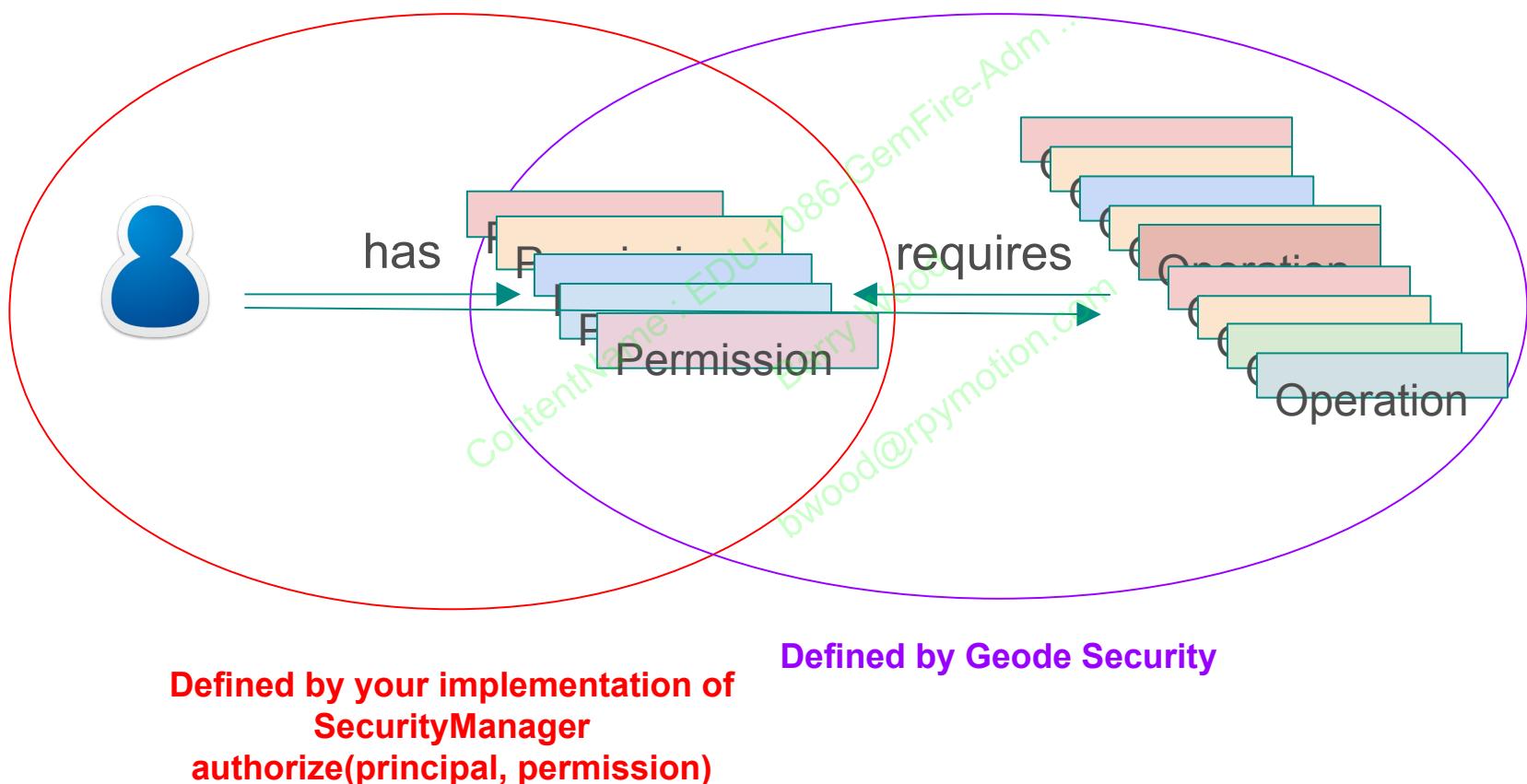
ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



GemFire Security (as of 9.0)



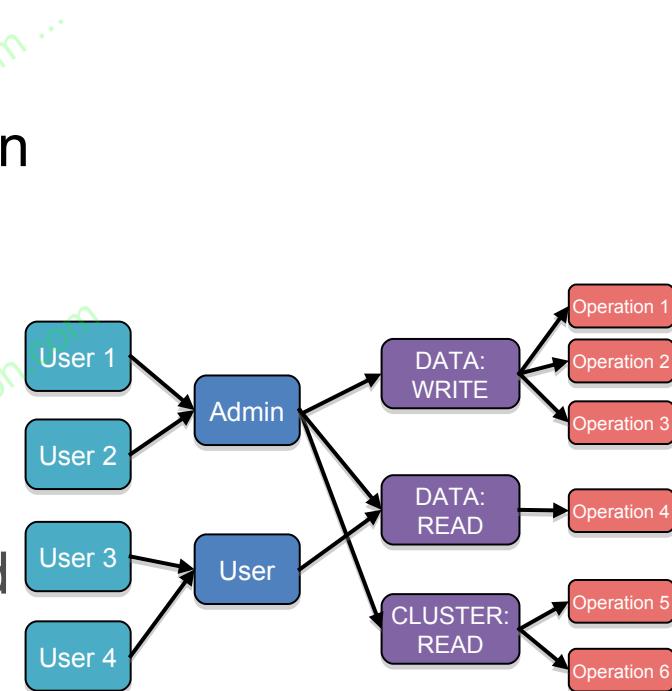
Users, Permissions and Operations



Integrated Security

Consistent Mechanism for Authentication and Authorization

- A single interface simplifies configuring security across all client/server access points
- Role-based, programmable, authorization for administrative activities
- Uniformity across clients that perform cache/region operations and those that perform management operations
- Authorization of data and cluster operations can be individually configured
 - Some users can read/write data
 - Others can stop/start servers
 - Still others can perform backups, etc



Integrated Security - New Callbacks

- One (Simpler) interface for securing everything
 - GFSH
 - JMX
 - Clients
 - Developer REST API
- Only one Interface for Both
 - Authentication
 - Authorization
- One out-of-the-box implementation
 - ExampleSecurityManager

ContentName : D:\1086-GemFire-Adm ...
Barry Wood

```
public interface SecurityManager {  
  
    Principal authenticate(Properties props)  
        throws AuthenticationFailedException;  
  
    default boolean authorize(Principal principal,  
        GeodePermission permission) {  
        return true;  
    }  
}
```

Integrated Security - New Callbacks

- Define using new gemfire property security-manager

```
security-manager = com.mycompany.MySecurityManager
```

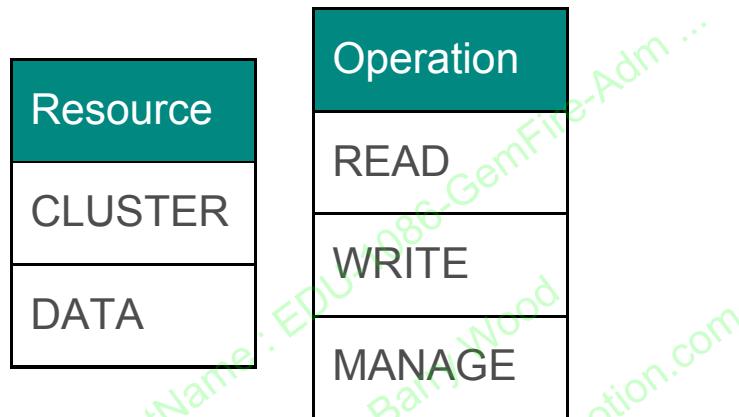
- Applies to everything (gfsh, JMX, client-server, peer, REST)

ContentName : EDU-1086-GemFire-ADM ...
Barry Wood
bwood@rpymotion.com

Integrated Security

- New Permissions

- Permission is defined as RESOURCE:OPERATION



- Each operation, no matter where it originates, has a corresponding ResourcePermission
 - Start server: CLUSTER:MANAGE
 - List member: CLUSTER:READ
 - Create region: DATA:MANAGE
 - Region get: DATA:READ:regionName:key
 - Region write: DATA:WRITE:regionName:key

ExampleSecurityManager

- Sample Implementation built-in

```
security-manager = org.apache.geode.examples.security.ExampleSecurityManager
```

- Create a “security.json” file and add it to classpath

- Define Roles

```
{  
  "roles": [  
    {  
      "name": "clusterMember",  
      "operationsAllowed": [  
        "CLUSTER:MANAGE",  
        "CLUSTER:WRITE",  
        "CLUSTER:READ"  
      ]  
    },
```

```
    {  
      "name": "clusterAdmin",  
      "operationsAllowed": [  
        "DATA:READ",  
        "CLUSTER:MANAGE",  
        "CLUSTER:WRITE",  
        "CLUSTER:READ"  
      ]  
    },  
  ],
```

ExampleSecurityManager - Users

- Define users for the Roles

```
"users": [  
  {  
    "name": "admin",  
    "password": "1234567",  
    "roles": ["clusterAdmin"]  
  },  
  {  
    "name": "member",  
    "password": "abc123",  
    "roles": ["cluserMember"]  
  }  
]
```

ContentName : EDU-1086-GemFire-Adaptive-Scaling
Barry Wood
bwood@rpymotion.com

Starting a Secure Cluster

1. Set the security-manager property in properties file

```
security-manager = org.apache.geode.examples.security.ExampleSecurityManager
```

2. Start locator, referencing properties file

```
gfsh> start locator --name=locator --properties-file=locator.properties
```

```
...
```

```
Authentication required to connect to the Manager.
```

3. Connecting to JMX Manager

```
gfsh> connect
```

```
Connecting to Locator at [host=localhost, port=10334] ..
```

```
Connecting to Manger at [host=192.168.0.60, port 1099] ..
```

```
user: admin
```

```
password: *****
```

```
Successfully connected to: [host=192.168.0.60, port 1099]
```

Starting a Secure Cluster - 2

4. Start Server(s) - no credentials provided

```
gfsh> start server --name=server1 --server-port=0 --locators=localhost[10334]
Exception in thread "main" org.apache.geode.security.AuthenticationRequiredException
user/password required. Please start your server with --user and --password. Failed to find
credentials from [192.168.0.60(server1:2768):1025]
```

5. Start Server(s) with credentials

```
gfsh> start server --name=server1 --server-port=0 --locators=localhost[10334]
--user=member --password=abc123 --classpath=..//security
Starting a Geode Server in /Applications/GemFire-Admin-9.0.2a.RELEASE/runLab/server1...
.....
Server in /Applications/GemFire-Admin-9.0.2a.RELEASE/runLab/server1 on
192.168.0.60[50949] as server1 is currently online.
```

Configuring Client Security

- Requires implementing `AuthInitialize` interface and registering
 - Developer team should implement and provide JAR
 - Developer team should also provide properties file with appropriate configuration items

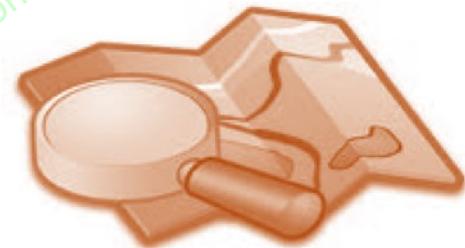
```
security-client-auth-init= com.mycompany.MyAuthInitializer
```

- Up to client implementation to determine how to obtain username and password

Lesson Road Map

- Configuring Authentication & Authorization
- **Post Processing**
- Configuring SSL

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



Integrated Security - New Callbacks

- New Interface for Post Processing
 - Provides ability to filter values before returning
- Examples include:
- Returning a subset of the entry data
 - Modifying or providing alternative values based on user & role

```
public interface PostProcessor {  
    Object processRegionValue(Principal principal,  
        String regionName, Object key, Object value);  
}
```

What You Control with Post Processor

- Allows you to intercept and modify results from the following operations
 - get (also getAll) - what is returned from a region
 - Query and Continuous Query results
 - Cache Listener (after update) coupled with Register Interest on client
- Motivations
 - Some entries may have protected information that shouldn't be shared based on requestor role
 - Some roles might require summarized or formatted result
- Caveat
 - Can incur additional processing overhead (esp. with gets)

Configuring the Post Processor

- Configure the Post Processor

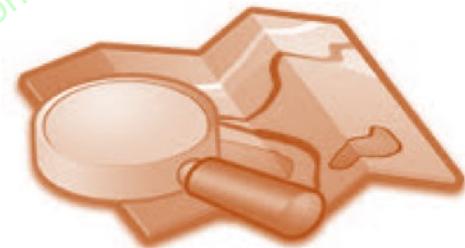
```
security-post-processor= org.apache.geode.examples.security.ExamplePostProcessor
```

- One out-of-the-box sample implementation
 - Basically concatenates principle info + region name + key + value
- Post Processor will be triggered if security has been enabled

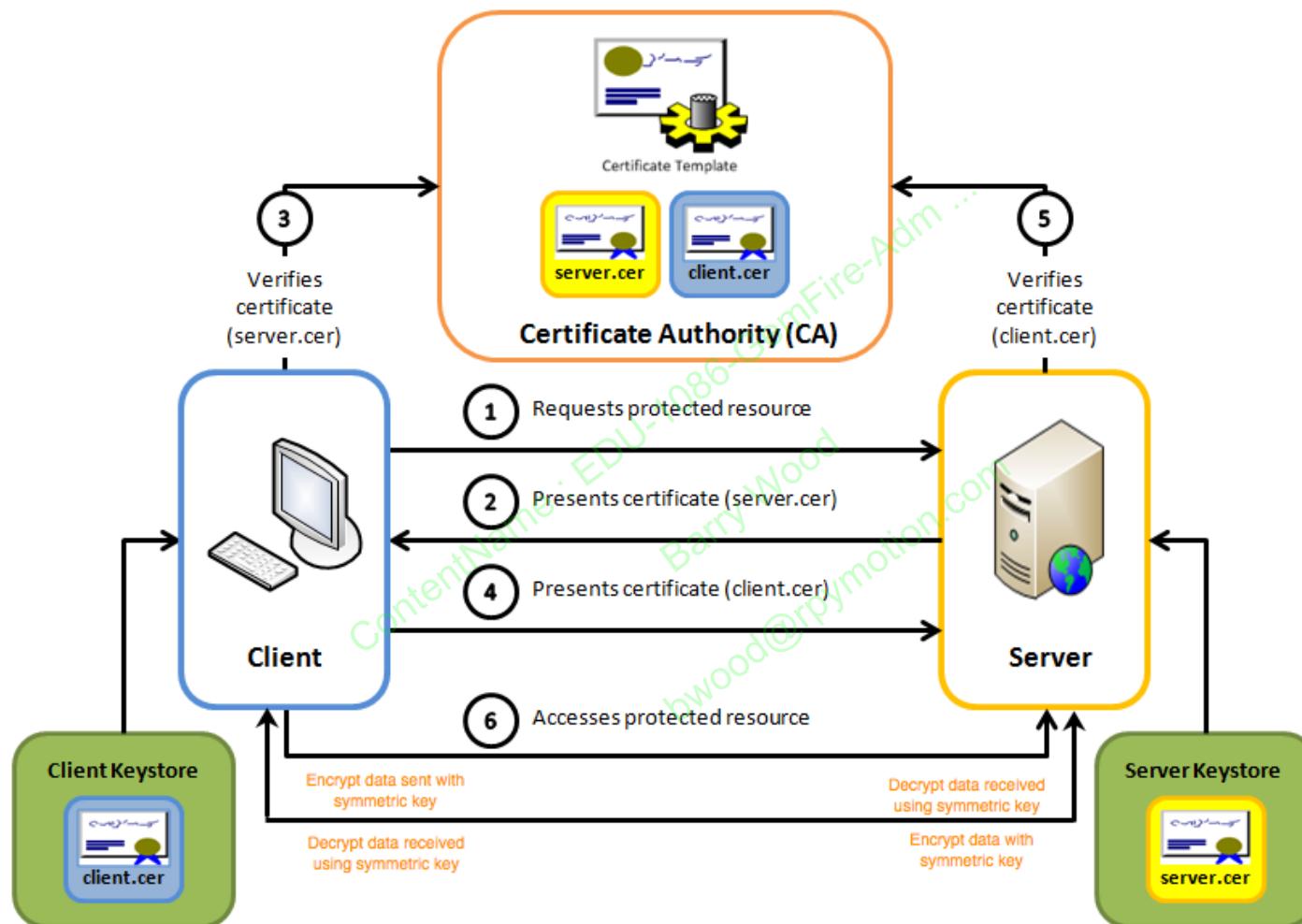
Lesson Road Map

- Configuring Authentication & Authorization
- Post Processing
- **Configuring SSL**

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com



SSL Overview



Mutual SSL authentication / Certificate based mutual authentication

SSL Terminology

- SSL Key Store
 - The store for private keys and public certificates that will be sent to remote
- SSL Trust Store
 - The store for public certs (CA certificates) the application is willing to trust
- Often will use the same store for both

GemFire SSL Configuration

- Add to gemfire.properties

```
# enable ssl only for client/server communication  
ssl-enabled-components=server
```

- Add to protected gfsecurities.properties

```
ssl-keystore=path-to-keystore  
ssl-keystore-password=password  
ssl-truststore=path-to-truststore  
ssl-truststore-password=password
```

Important SSL Properties

- **ssl-enabled-components**
Which components of architecture will use SSL (all or comma separated list)
- **ssl-require-authentication**
Whether two-way authentication is required (default: true)
- **ssl-protocols**
'all' or comma separated list (ex: TLSv1.2)
- **ssl-ciphers**
'all' or comma separated list (ex: SSL_RSA_WITH_NULL_MD5)

Full details at:

http://gemfire.docs.pivotal.io/geode/managing/security/ssl_overview.html

Lab – GemFire Security Configuration

In this lab, you will:

1. Configure a Security Manager
2. Configure authentication and authorization for:
 - a. gfsh administration
 - b. Members (servers) joining distributed system
 - c. Clients accessing data



Version 9.0.4c

Review Objectives

You should be able to do the following:

- Describe the GemFire security model
- Configure Authentication and Authorization
- Configure a security post-processor
- Configure SSL communication

ContentName : DLU-101-GemFire-Admin ...
Barry Wood
bwood@rpymotion.com

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Darryl Wood
bwood@rpxmotion.com

Logging, Stats and Troubleshooting

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

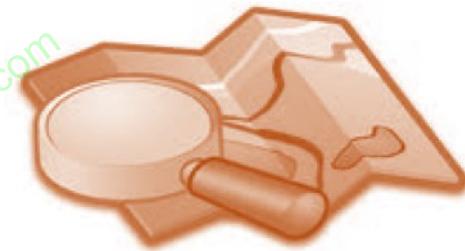
Objectives

After completing this lesson, you should be able to:

- Understand how to configure system to support logging
- Understand statistics data gathering options
- Understand how to configure system to collect statistics data
- Understand how to chart statistics data using Visual Statistics Display tool
- Review configuration to support logging and statistics data collection

Lesson Road Map

- **Logging**
- Statistics
- Visual Statistics Display
- Troubleshooting Configuration



ContentName : DLU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Logging Categories

- GemFire provides comprehensive logging messages to:
 - Confirm system configuration
 - Aid in debugging problems
- Logging categories:
 - **Startup information:**
 - Running system and configuration information
 - Versioning – Java and GemFire native version
 - Host system - current working directory and environment settings
 - **Logging management:**
 - Messages pertaining to the maintenance of the log files



Logging Categories

- **Connections and system membership:**
 - Report arrival and departure of distributed system members
 - Connection activities or failures occurrences
- **Distribution:**
 - Distribution of data between system members
 - Region configuration
 - Entry creation/modification and invalidation
- **Cache, region and entry management:**
 - Cache and region initialization
 - Listener activity
 - Locking/unlocking
 - Entry updates

Log Message Structure

Log Messages consist of two elements:

1) Message header (in square brackets) consisting of:

- Message level
- Time the message was logged
- ID of the connection and thread that logged the message, may be main program or a system management process

2) Message body consisting of:

- Message string
 - and/or-
- Exception with the stack trace

```
[config 2005/11/08 15:46:08.710 PST PushConsumer main nid=0x1]
Cache initialized using "file:/Samples/quickstart/xml/PushConsumer.xml".
```

Log File Configuration

- Log file configuration in `gemfire.properties` file

```
log-level=config  
log-file=          # Default based on CWD and member name  
log-file-size-limit=0  
log-disk-space-limit=0
```

These are default values

- `log-file-size-limit`

- Maximum size of single log file before roll to next log file
- Best practice: log file size limit = 100MB
- 0 Indicates no limit

- `log-disk-space-limit`

- Maximum size of all log files
- 0 indicates no limit

Configuration via gfsh

- Setting log levels

```
gfsg> change log-level --log-level=fine  
  
gfsh> alter runtime --log-level=fine  
      --log-file-size-limit=0 --log-disk-space-limit=0
```

- Why use gfsh?
 - Changes can be made without re-starting servers
- Limitations of gfsh
 - Can't specify log file location

Logging Message Levels

Level	Description
severe	Highest severity level, indicating a serious failure that usually prevents normal program execution. You might have to shut down or restart part of your cluster to correct the problem.
error	The operation indicated in the error message has failed. The server or locator should continue to run.
warning	Potential problem with the system. In general, warning messages describe events that are of interest to end users or system managers.
info	This is for informational messages, typically geared to end users and system administrators.
config	Informational messages for end users and system administrators.
fine	Tracing information that is generally of interest only to application developers. This logging level may generate "noise" that might not indicate a problem in your application.
finer/finest/all	Internal use only. DO NOT use unless instructed to do so by support.

Managing Log Files

- Log files can be configured:
 - **log-file-size-limit**: roll when they reach a specified size limit
 - When the limit is reached:
 - Current log file is closed
 - New file is created
- GemFire can be configured:
 - **log-disk-space-limit**: limit the amount of disk space used for logging
 - When the limit is reached:
 - GemFire starts deleting log files to stay under the limit starting with the oldest log file deleted first
- GemFire nodes will not function if they are unable to log messages



Log File Summary

- Prefer to use gfsh to configure logging
- Logging design
 - Define a local disk location (not NAS or SAN)
 - Manage log files via log-file-size-limit and log-disk-space-limit
- Beware - fine logging level will generate lots of detail

Content ID : EDU-100 GemFire Admin ...
By Bill Wood
bwood@rpymotor.com

GC Logs

GC logging is enabled with startup parameters added to the JVM. The following parameters should be added when enabling GC logging for GemFire GC tuning issues:

```
gfsh start server --name=server2...
```

```
...
```

```
--J=-XX:+UseParNewGC
```

Activates “simple” GC logging mode, prints basic information for every young generation GC and every full GC

```
--J=-XX:+PrintGC
```

Activates the “detailed” GC logging mode, which prints elaborated GC information.

```
--J=-XX:+PrintGCDetails
```

Print timestamps for each GC event (seconds count from start of JVM)

```
--J=-Xloggc:C:/gc.log
```

Redirects GC output to a file instead of console

GC Logs – The Log File

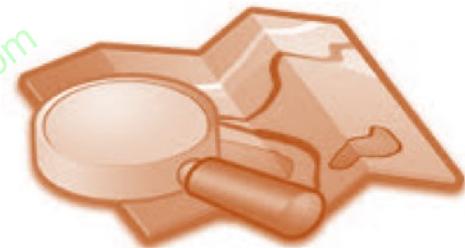
```
0.535: [GC@0.536: [ParNew: 17472K->2176K(19648K), 0.0140882 secs]
20472K->4122K(63360K), 0.0143121 secs] [Times: user=0.02 sys=0.00,
real=0.01 secs]
```

```
...
```

- **0.535** – Represents the number of seconds the JVM is launched (generated by PrintGCTimeStamps flag)
- **GC** – Type of GC occurred, minor or full
- **ParNew** – GC for young generation
- **17472k** – Represents the occupancy of young generation prior to GC
- **2176k** - Represents the occupancy of young generation after GC, represents the Survivor space
- **19648k** – Size of Eden space and Survivor space
- **20472K** – Total size of the heap (young space + old space) before GC
- **4122K** – Total size of heap after GC
- **63360K** - The committed size of the heap, the amount of space usable for Java objects without requesting more memory from the operating system
- **0.0143121** – Time in seconds to complete the GC
- **[Times: user=0.02 sys=0.00, real=0.01 secs]** – CPU utilization statistics

Lesson Road Map

- Logging
- **Statistics**
- Visual Statistics Display
- Troubleshooting Configuration



ContentName : DLU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Statistics - Overview

- Each application or cache server can collect and archive statistic data for analyzing system performance
- Statistic data collection and archiving is defined on a individual server basis, using `gemfire.properties` file
- GemFire supports:
 - Sampling based statistic data collection archiving
 - Does not affect system performance
 - Time based statistic data collection and archiving
 - Does affect system performance
 - Require that sampling based be enabled



Statistics - Overview

- GemFire provides a Java API for support of custom statistic data collection
- Visual Statistics Display (VSD) tool:
 - View and analyze historical statistic data
 - Chart sample and time based statistic data
 - But, not transient region or entry statistic data
- **gfsh show metrics**
 - Display or export metrics for the entire distributed system, member, or a region



Statistics Configuration – Sampling

gemfire.properties

```
statistic-sampling-enabled=true  
statistic-sample-rate=1000  
enable-time-statistics=true
```

- **statistic-sampling-enabled**
 - Enables non-time based statistics for the distributed system
 - Does not include transient or entry statistic data
- **statistic-sampling-rate**
 - Define frequency of sampling in milliseconds
- **enable-time-statistics**
 - Enables time-based statistics for the distributed system
 - Affects system performance

Statistics Configuration – Statistics Files

gemfire.properties

```
statistic-archive-file=/Users/myMac/myApp/stats1007.gfs.gz  
archive-file-size-limit=100  
archive-disk-space-limit=1000
```

- **statistic-archive-file**
 - Define directory and filename of statistics archive file
 - Suffix .gz to compress file
- **archive-file-size-limit**
 - Maximum size of single archive file in MB
- **archive-disk-space-limit**
 - Maximum size of all archive files in MB

Transient Region and Entry Statistics

- Transient Statistics include:
 - Hit and miss counts
 - Last access time
 - Last modified time
- Transient Statistics defined on per-region basis

cache.xml

```
<region name="myRegion" refid="REPLICATE" >  
    <region-attributes statistics-enabled="true" />  
</region>
```



Configuration via gfsh

- Most options can be configured and altered via gfsh

```
gfsh> alter runtime --enable-statistics=true  
      --statistic-sample-rate=1000  
      --statistic-archive-file=/Users/myMac/myApp/stats1007.gfs.gz  
      --archive-file-size-limit=100 --archive-disk-space-limit=1000
```

- Note:
 - ‘enable-time-statistics’ can be specified when starting a server but not via ‘alter runtime’
 - ‘enable-statistics’ can be specified when creating a region but not via ‘alter region’

Lesson Road Map

- Logging
- Statistics
- **Visual Statistics Display**
- Troubleshooting Configuration



ContentName : DLU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

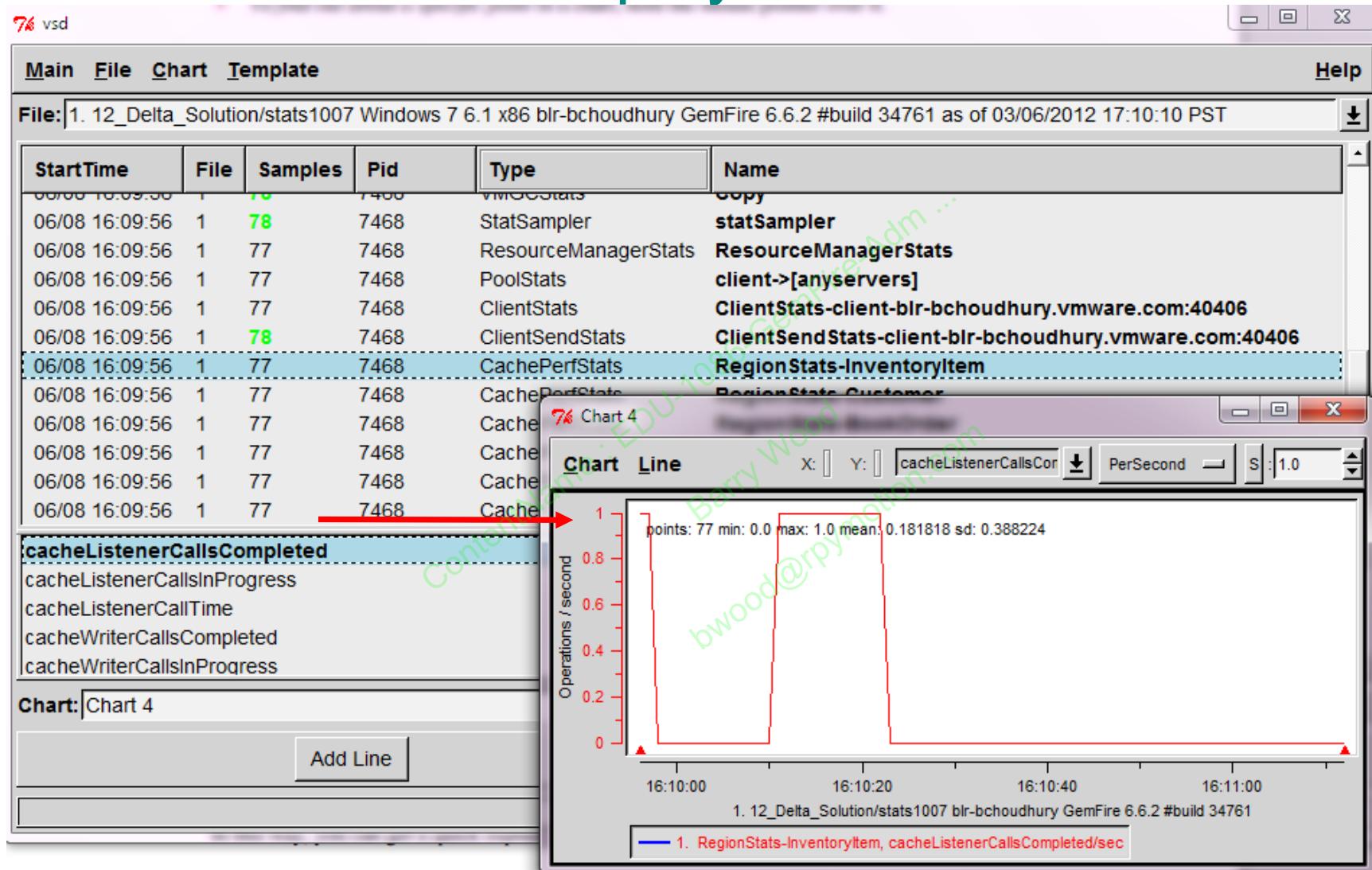
Visual Statistics Display Tool

- Visual rendering tool for statistics generated by GemFire
- Visual display allows for complex manipulation of statistics across several files:
 - Ability to diagnose problems across distributed systems
- Files generated for VSD are what Pivotal GemFire Support will usually ask for when you are having performance problems

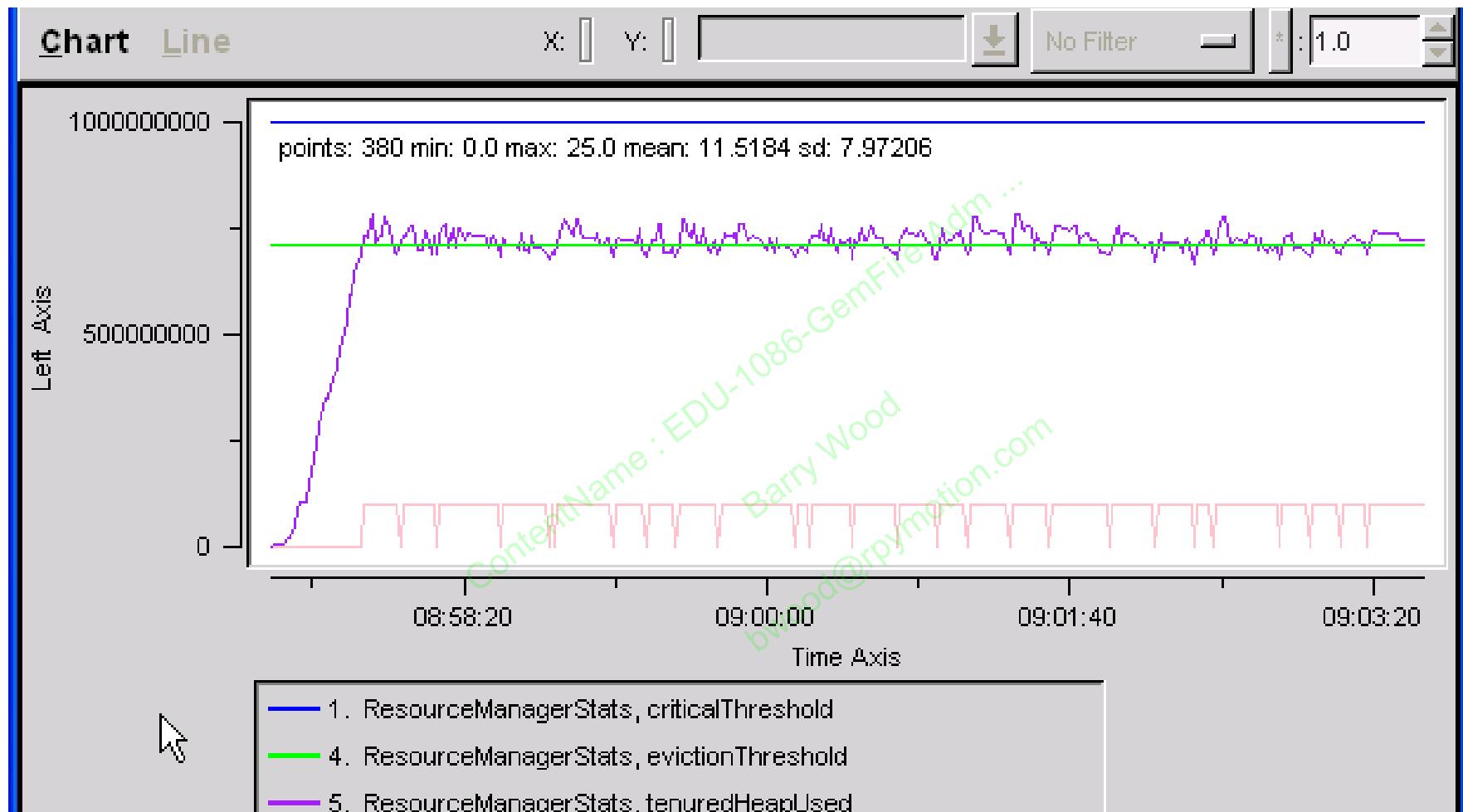


Version 9.0.4c

Visual Statistics Display Tool



Visual Statistics Display Tool – Chart



Visual Statistics Display Tool

- VSD distributed along with GemFire but must be downloaded separately (as of GemFire 9.x)
- Reads statistics files (such as **filename.gfs**)
- Started two ways
 - Run executable directly `$GEMFIRE/bin/vsd`
 - Using **gfsh** utility `gfsh> start vsd`
- Cumulative statistics are best charted (for readability)
 - Per second
 - Per sample
- Absolute value best charted with No Filter
- VSD templates can be used to aid charting
- Statistic level determines which statistics show up in list of available statistics to display

Lesson Road Map

- Logging
- Statistics
- Visual Statistics Display
- **Troubleshooting Configuration**



ContentName : HOU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Troubleshooting Configuration

- In GemFire production environment for:
 - System health verification
 - Performance analysis
 - Troubleshooting
- Enable the following:
 - Logging functionality
 - Statistics functionality
- Synchronize host clocks using Network Time Protocol(NTP)

CourseName : EDU-1086-GemFire-Admin ...
Barry Wood
bwg@rpymotion.com

Troubleshooting Configuration

- Set logging level via gfsh (or gemfire.properties file)

```
gfsh> change log-level --log-level=config
```

- Enable statistics via gfsh (or gemfire.properties file)

```
gfsh> alter runtime --enable-statistics=true  
      --statistic-archive-file=statisticsArchiveFile.gfs
```

- Use gfsh to export stack traces for analysis

```
gfsh> export stack-traces \  
      --file=applStackTrace.txt \  
      --member=server1
```



Troubleshooting Data Files

The following files are helpful for troubleshooting

- Log files
- Statistics archive files
- Core files
 - For Linux, you can use `gdb` to extract a stack from a core file
- Crash dumps
- Thread dumps are useful for hung, stuck or slow processes
 - Use Java `jstack` to generate dump
- Heap dumps/histograms are useful for memory issues
 - Use Java `jmap` command to generate dump or histogram



Review of Learner Objectives

You should be able to do the following:

- Describe how to configure system to support logging
- Describe differences between sampling, time-based, and transient statistic data
- Describe how to configure system to collect statistic data
- Describe use of VSD

Content ID : EJU-1086-GemFire-Admin
Barry Wood
bwood@rpymotion.com

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

Multi-site Configuration

Integrating Gemfire Installations at Different Sites

Using Gemfire's WAN Gateway

Content ID: EDU-1086-GemFire-Adm ...
By: Bryan Wood
Email: bwood@rpymotion.com

Objectives

After completing this lesson, you should be able to:

- Understand communication channels used in the multi-site topology
- Understand function of gateway senders and receivers
- Describe how to configure gateway senders and receivers
- Describe how to configure a persistent queue
- Understand queue conflation

Lesson Road Map

- **Multi-site overview**
- Configuration topologies
- Configuring persistent queues

ContentName : EDU-105 GemFire-Admin ...
Barry Wood
bwood@rpymotion.com



Multi-Site Systems

- Connects disparate distributed systems:
 - Region data is shared
 - Systems act as one when they are coupled
 - Act independently when communication fails
 - Tolerant of poor networks
 - Ordering of events is preserved
- Use cases:
 - WAN replication
 - Write behind
 - Disaster recovery
 - Running upgrades
 - Geographic Regional Isolation/Locality



Pivotal
GemFire

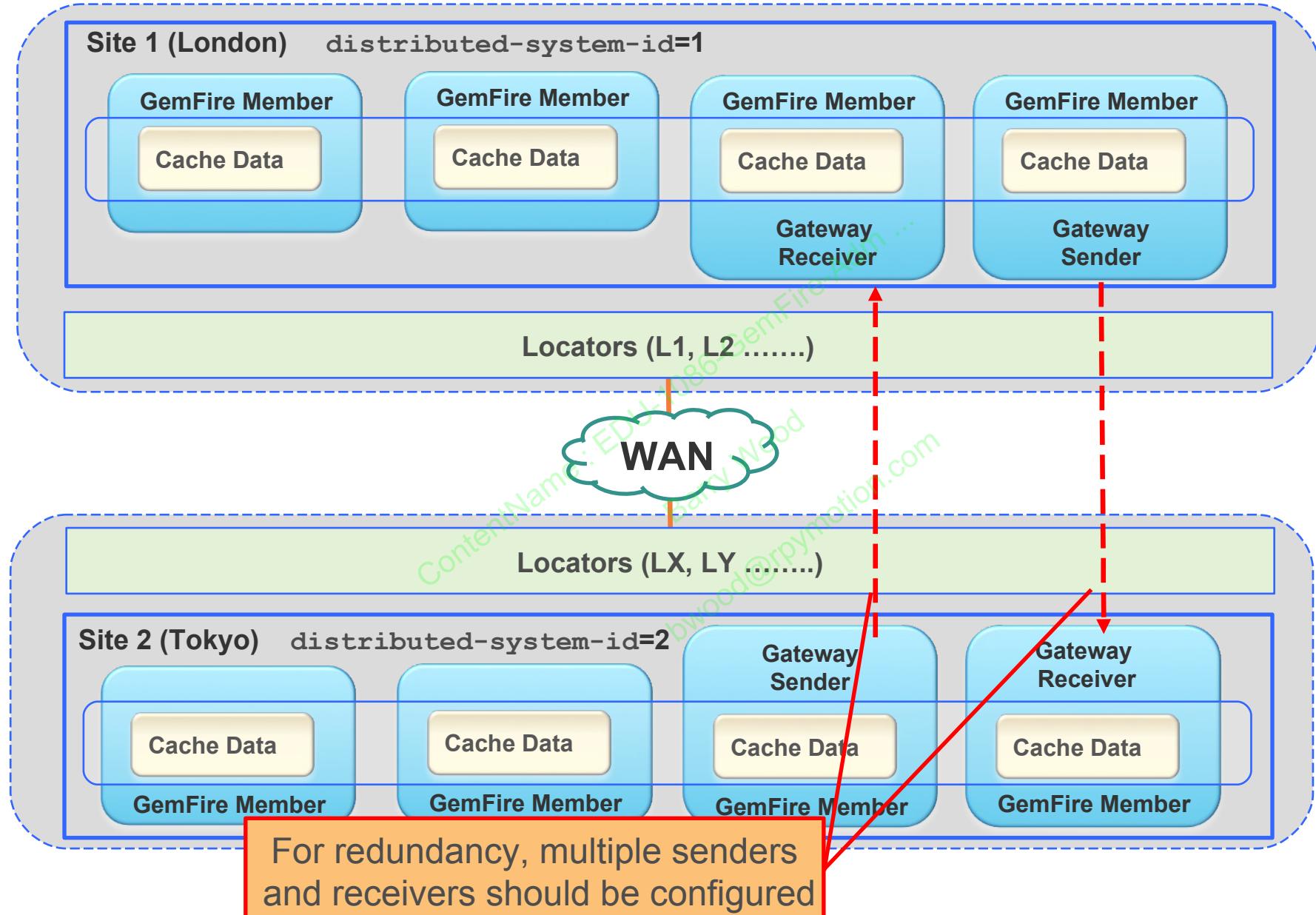
Version 9.0.4c

Multi-Site Systems

- Asynchronous
 - If updates happen on the same object at the same time, no guarantee of consistency
 - If you have multiple different data-sets you could set up one site as the primary for one data-set and the other site as the primary for the other data-set and they can serve as back-up for each other
 - In some cases you can use the inventory allocation pattern to host part of your inventory in different venues or locations
 - There are other patterns for doing “follow-the-sun” or “pass-the-book” types of scenarios

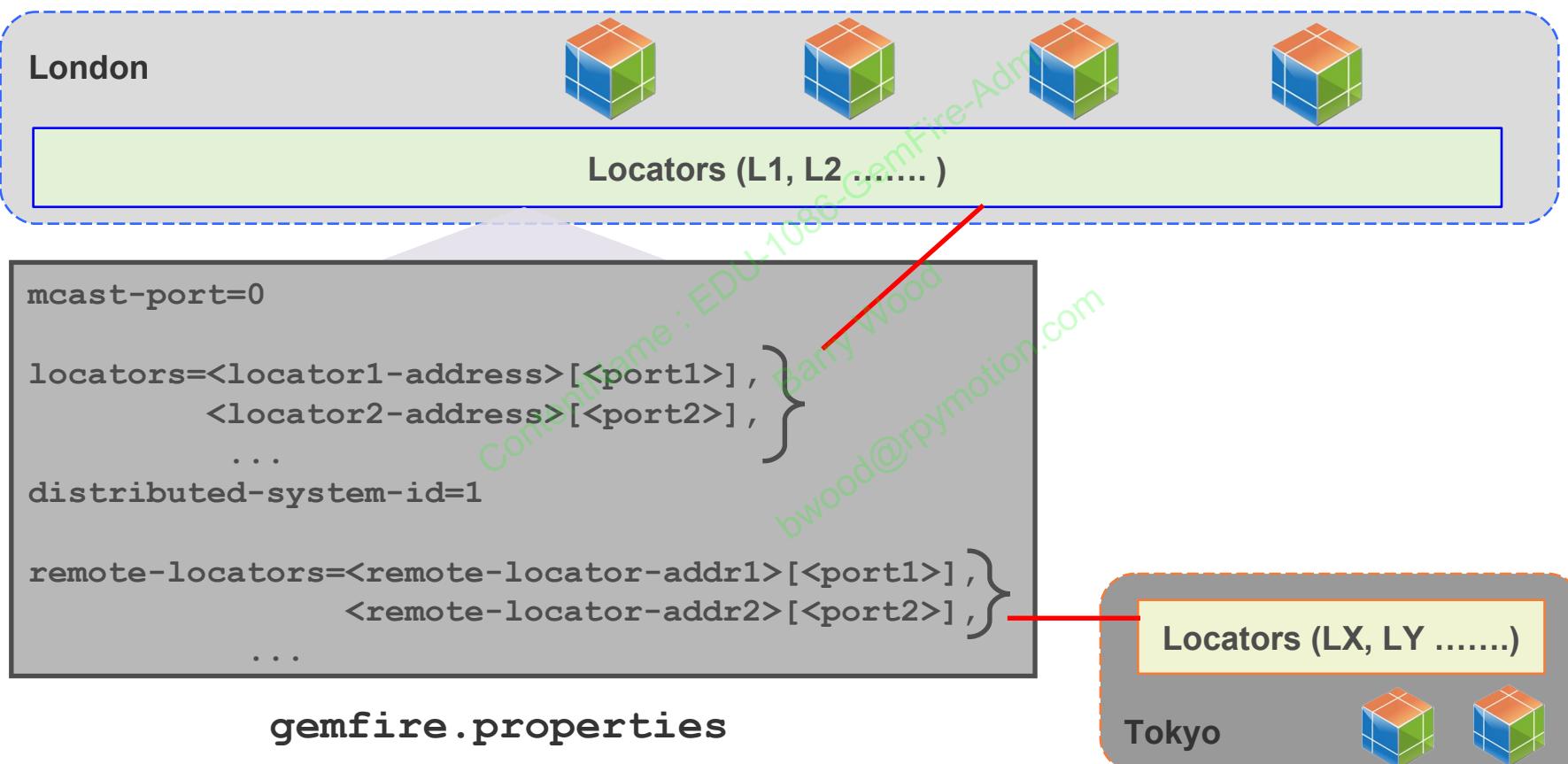


Multi-site Caching: Overview



Multi-site Caching: Overview

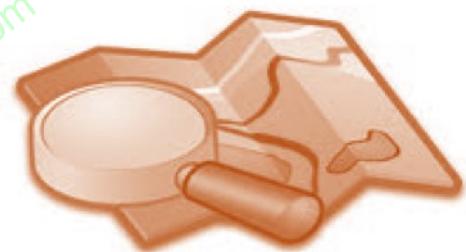
To use locators in a WAN configuration, specify a unique distributed system ID (integer value) for the local cluster, as well as the locator(s) available in remote clusters to which you will connect



Lesson Road Map

- Multi-site overview
- Configuration topologies
- Configuring persistent queues

ContentName : EDU-105 GemFire-Admin ...
Barry Wood
bwood@rpymotion.com



Configuring Multi-Site Systems

1

Plan the topology

2

Configure distributed system id and remote locator information for each cluster

3

Define and configure gateway senders and gateway receivers for communication between systems

4

Enable gateway communication in the data regions that will participate



Pivotal
GemFire

Version 9.0.4c

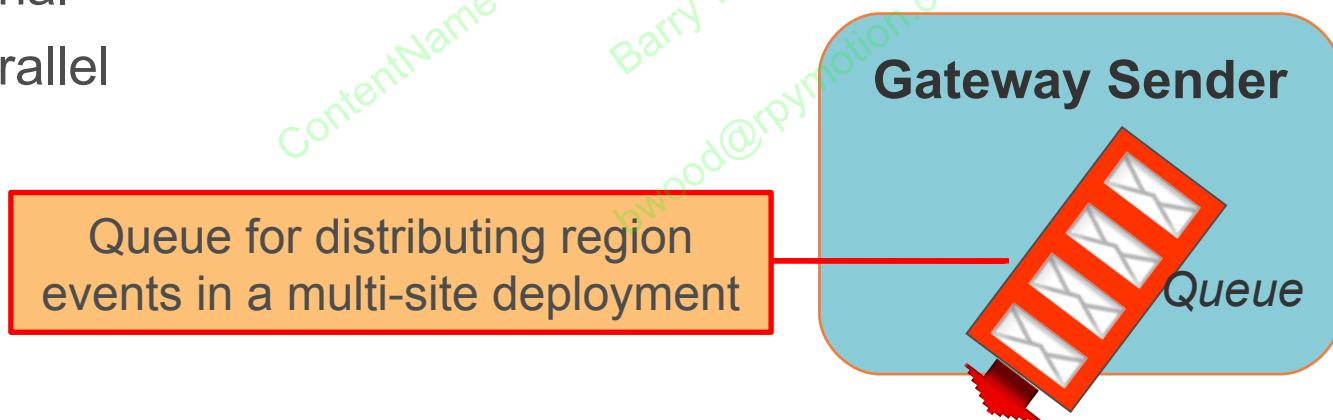
Gateway Senders

In a GemFire cluster:

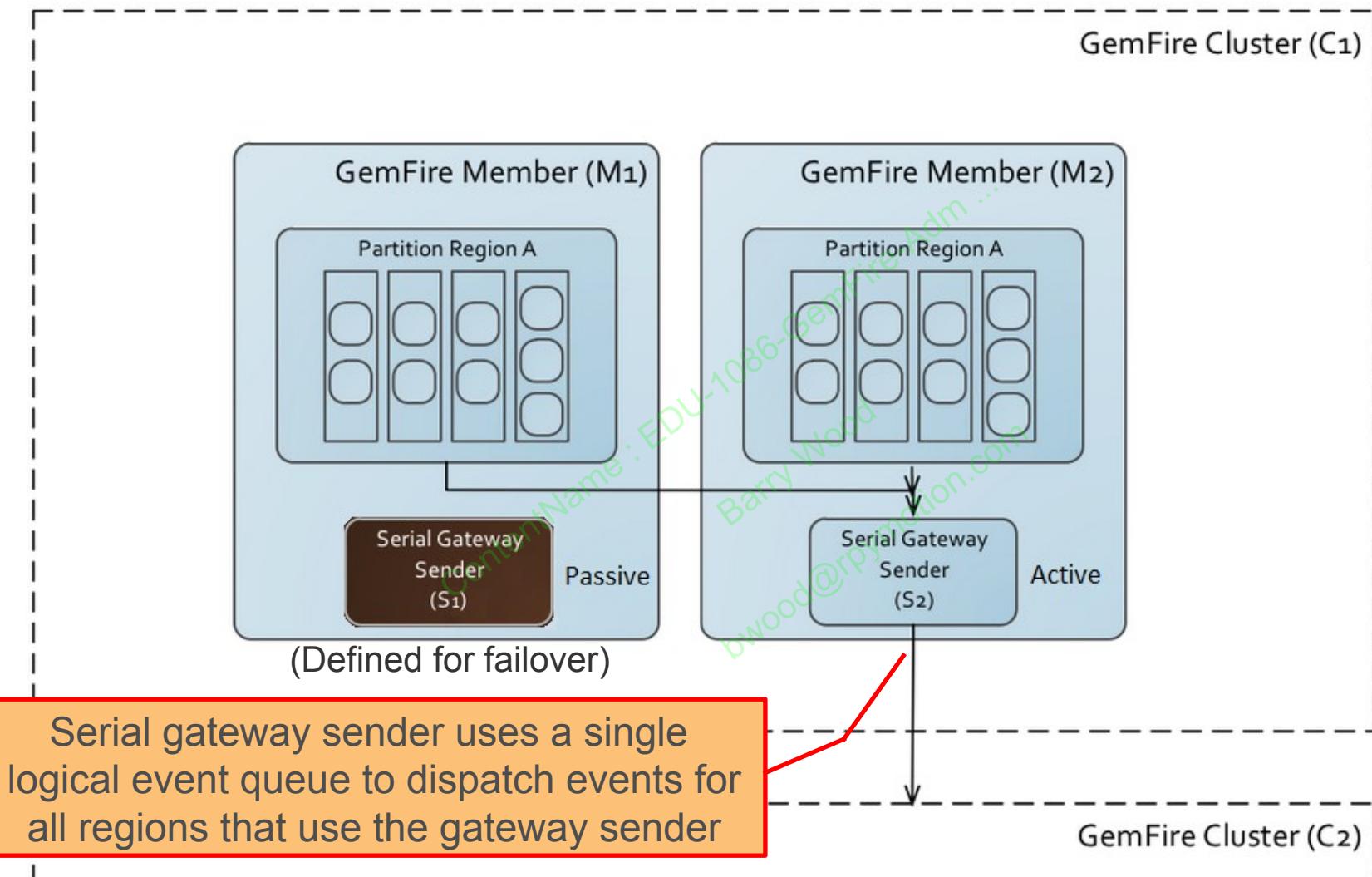
- Use gateway senders to distribute region events to another remote GemFire cluster
- Configure multiple gateway sender to distribute region events

Types of gateway sender configurations:

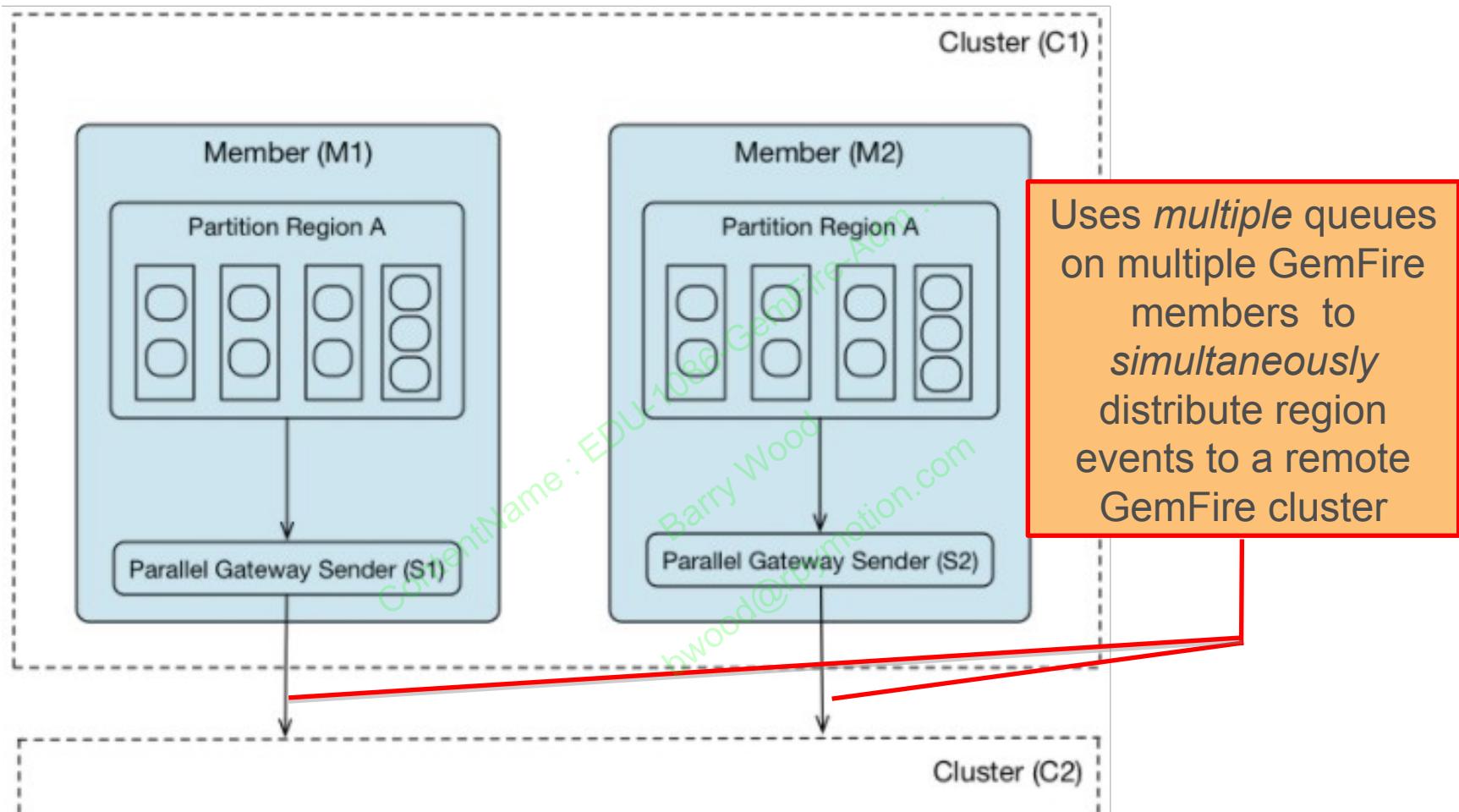
- Serial
- Parallel



Serial Gateway Sender



Parallel Gateway Sender



Replicated regions *cannot* use a parallel gateway sender

Configure Gateway Senders

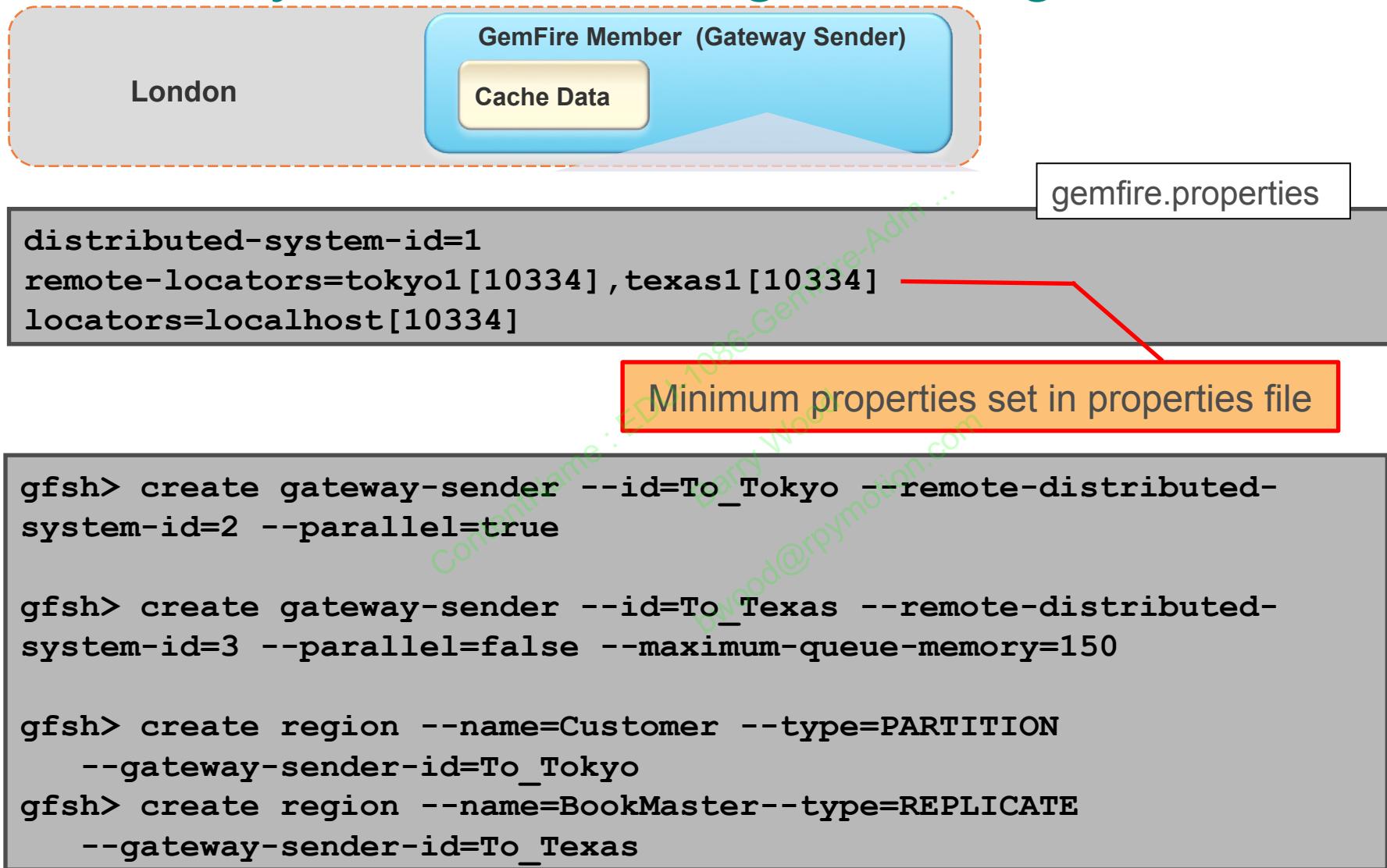
Each gateway sender configuration defines:

- Unique ID for the gateway sender configuration
- Distributed system ID of the remote site to which the sender propagates region events
- Property that specifies whether the gateway sender is a:
 - Serial gateway sender
 - Parallel gateway sender

ContentName : EDU1086-GemFire Admin
Barry Wood
bwood@rpymotion.com



Gateway Senders Configuration: gfsh



Gateway Senders Configuration: Cache.xml

London

GemFire Member (Gateway Sender)

Cache Data

cache.xml

```
<cache>
    <gateway-sender id="To_Tokyo" remote-distributed-system-id="2"
                     parallel="true" />

    <gateway-sender id="To_Texas" remote-distributed-system-id="3"
                     parallel="false" maximum-queue-memory="150"/>

    <region name="Customer">
        <region-attributes refid="PARTITION"
                           gateway-sender-ids="To_Tokyo"/>
    </region>

    <region name="BookMaster">
        <region-attributes refid="REPLICATE"
                           gateway-sender-ids="To_Texas"/>
    </region>
</cache>
```

Gateway Receiver

- In a GemFire cluster, a gateway receiver:
 - Configures a physical connection for receiving region events from gateway senders in one or more remote GemFire clusters
 - Incorporates each region event to the same region or partition that is hosted in the local GemFire member
- You can deploy gateway receiver configurations:
 - To multiple GemFire members
 - High availability
 - Load balancing
- If a gateway receiver receives an event for a region that the local member has not defined, GemFire throws an exception



Gateway Receivers Configuration: gfsh

Tokyo

GemFire Member (Gateway Receiver)

Cache Data

Presumes properties set
in gemfire.properties file

```
gfsh> start server --name=server1 ...  
  
gfsh> create gateway-receiver --member=server1 --manual-start=true  
      --start-port=1530 --end-port=1551  
  
gfsh> start gateway-receiver --member=server1
```

Specifies a *range* of port numbers on which to listen. The GemFire server picks an unused port number from the specified range to use for the receiver process.

Gateway Receivers Configuration: Cache.xml

Tokyo

GemFire Member (Gateway Receiver)

Cache Data

cache.xml

```
<cache>

<gateway-receiver start-port="1530" end-port="1551"/>

<region name="Customer">
    <region-attributes refid="PARTITION"/>
</region>

<region name="BookMaster">
    <region-attributes refid="REPLICATE"/>
</region>

</cache>
```

Specifies a *range* of port numbers on which to listen. The GemFire server picks an unused port number from the specified range to use for the receiver process.

Determining Gateway Status

- Gateway status

```
Cluster-1 gfsh>status gateway-sender --id=To_Tokyo
```

Member	Type	Runtime Policy	Status
192.168.0.60 (server1:77330)<v1>:1025	Serial	Primary	Running

- Obtaining more detail on gateways

```
Cluster-1 gfsh>list gateways
```

Gateways

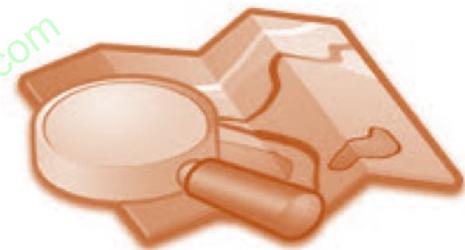
GatewaySender

GatewaySender Id	Member		Remote Cluster Id	
Type	Status	Queued Events	Receiver Location	
To_Tokyo1	192.168.0.60 (server1:77330)<v1>:1025	0	2	Serial

Lesson Road Map

- Multi-site overview
- Configuration topologies
- **Configuring persistent queues**

ContentName : EDU-105-GemFire-Admin ...
Barry Wood
bwood@rpymotion.com



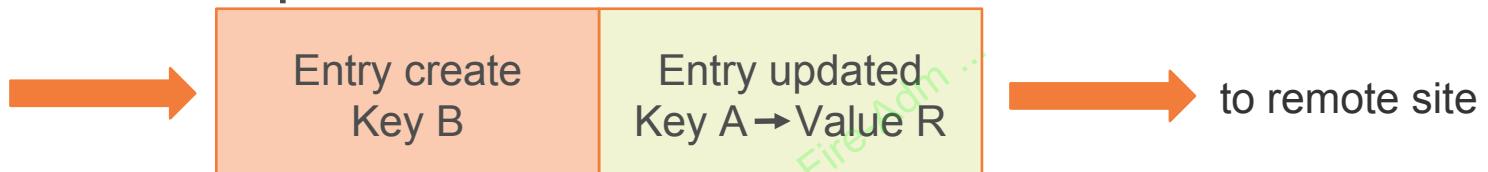
Gateway Queue Persistence

```
<cache>
  <disk-store name="ds1" >
    ...
  </disk-store >
  <gateway-sender id="To_Tokyo" parallel="true"
    remote-distributed-system-id="2"
    is-persistent="true"
    disk-store-name="ds1"
    maximum-queue-memory="100" />
  ...
</cache>
```

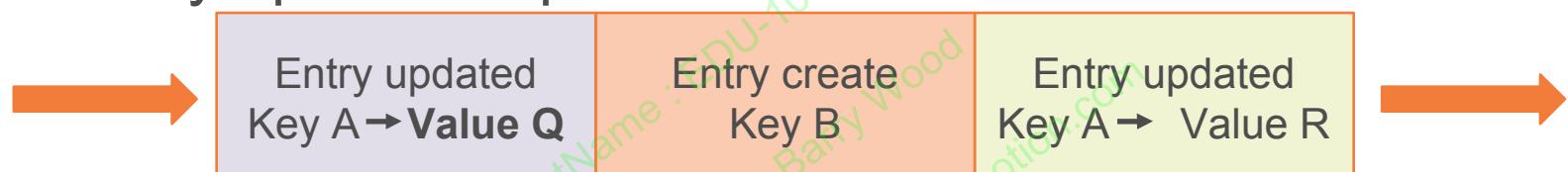


Site-to-Site Queue Conflation

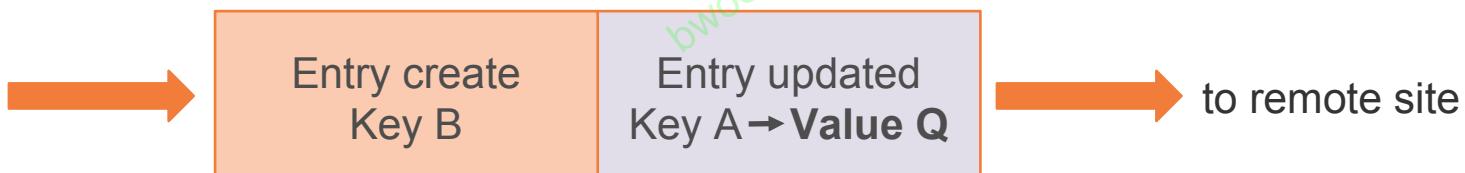
Initial local hub queue with conflation:



Add entry update to queue:



Queue after conflation:



Entry updates between remote distributed systems in a multi-site installation can be optionally *conflated* so only the latest value is sent for a particular key.

Gateway Queue Persistence and Conflation

```
<cache>  
  <gateway-sender id="To_Tokyo" parallel="true"  
    remote-distributed-system-id="2"  
    is-persistent="true"  
    disk-store-name="ds1"  
    maximum-queue-memory="100"  
    enable-batch-conflation="true"/>  
  ...  
</cache>
```

Review of Objectives

You should be able to do the following:

- Describe communication channels used in multi-site topology
- Describe function of gateway senders and receivers
- Describe configuration of gateway senders and receivers
- Describe how to configure a persistent queue
- Describe queue conflation

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com

Finishing Up

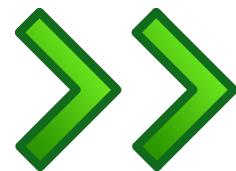
Course Completed

What's Next?

ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

What's Next

- Congratulations, you've finished this course
- What do do next?
 - Certification
 - Other courses
 - Resources
 - Evaluation
- Check out additional sections on (optional) ...



ContentName : EDU-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Certification



- Computer-based exam
 - 48 multiple-choice questions
 - 90 minutes
 - Passing score: 76% (36 questions answered successfully)
- Preparation
 - Review all the slides
 - Redo the labs
 - Dig deeper with documentation

ContentName : EDU-101-GemFire-Admin ...
Barry Wood
bwood@rpymotion.com

Certification: Questions

Typical question

- Statements
 - a. GemFire is an In Memory Distributed Database
 - b. GemFire offers High Latency processing
 - c. A GemFire cache is a container for one or more regions
- Pick the correct response
 - 1. Only a. is correct
 - 2. Both a. and c. are correct
 - 3. All are correct
 - 4. None are correct

Certification: Logistics

- Where?
 - Online at PSI (Innovative Exams)
 - <https://www.examslocal.com>
- How?
 - Purchase certification from <http://academy.pivotal.io>
 - You should receive a certification voucher by email
 - Register/sign-in and book an exam using the voucher
 - <http://it.psionline.com/exam-faqs/pivotal-faq>
 - Take the test from any location
- For any further inquiry, you can write to
 - education@pivotal.io

Voucher is valid for 3 months from purchase
- *do it soon*

Other courses



- Many courses available
 - Spring Cloud Data Flow
 - GemFire Developer
 - Greenplum Developer
 - Greenplum Administration
- More details here:
 - <http://www.pivotal.io/training>

ContentName : EPL-1086-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

GemFire Developer

- 4-day workshop
- Understand how to develop GemFire clients and server-side components
 - GemFire Client/Server applications
 - Event processing
 - Querying
 - Function creation, registration and execution
- Pivotal Certified GemFire Developer

Pivotal Support Offerings

- Global organization provides 24x7 support
 - How to Register: <http://tinyurl.com/piv-support>
- Premium and Developer support offerings:
 - <http://www.pivotal.io/support/offers>
 - <http://www.pivotal.io/support/oss>
 - Both Pivotal App Suite and Open Source products
- Support Portal: <https://support.pivotal.io>
 - Community forums, Knowledge Base, Product documents



Pivotal Consulting

- Custom consulting engagement?
 - Contact us to arrange it
 - <http://www.pivotal.io/contact/spring-support>
 - Even if you don't have a support contract!
- Pivotal Labs
 - Agile development experts
 - Assist with design, development and product management
 - <http://www.pivotal.io/agile>
 - <http://pivotallabs.com>



Resources

- The GemFire reference documentation
 - <http://gemfire.docs.pivotal.io>
 - Already 1000+ pages!
- The GemFire Knowledge Base
 - <https://discuss.zendesk.com/hc/en-us/categories/200072740>
- GemFire Forums
 - <https://discuss.zendesk.com/hc/en-us/community/topics/2034922700000>

Thank You!



- We hope you enjoyed the course
- Please fill out the evaluation form
 - Americas: <http://tinyurl.com/usa-eval>
 - EMEA: <http://tinyurl.com/emea-eval>
 - Asia-Pac: <http://tinyurl.com/apj-eval>
- Once you've done, login to *Pivotal Academy*
 - You can download your Attendance Certificate

Content ID : EU-100 Gemfire-Adm ...
Version : 9.0.4c
Author : rpyan@pivotal.com
Last Update : 2017-01-10 10:45:00

Thank You.

ContentName : EDU-1006-GemFire-Adm ...
Barry Wood
bwood@rpymotion.com

Pivotal

A NEW PLATFORM FOR A NEW ERA

Content ID: EDU-1086-GemFireAdm ...
Bryan Wood
bwood@rpxmotion.com