

Scaling ColdFusion



Presenter Mike Collins, Sr. ColdFusion Consultant - SupportObjective

- Help you develop Strategy for Scaling
- Improve overall server stability
- Giving your End-users a better Experience



1. Deciding on a Platform
2. Understanding Key Metrics
3. Learning Key Metrics thru Load Testing
4. Scaling Strategies - VMs and Orchestration
5. Understanding ColdFusion Clustering
6. Tomcat Load Balancing and Status Worker
7. Session Management Strategies

Deciding on a Platform



Major Scaling Decisions Start Early – Picking Your Stack

- **ColdFusion Version** – Enterprise or Standard
 - *Enterprise Only Features*
 - *Multi-instance Support*
 - API Manager
 - Advanced PDF features, attachments...
 - Security Code Analyzer
 - CFThread without throttling
 - Oracle, Sybase and DB2 DataDirect Drivers
- **OS** - Windows, Linux
- **J2EE Application Server**
 - Default Tomcat, WebSphere, WildFly, WebLogic
- **ColdFusion Frameworks \ REST Frameworks**
 - ColdBox, FW\1
 - API Manager, Taffy

- **Platform**

- Locations, On-Premise or Hosted
- VM Architecture
- Cloud Architecture
- Containers and Orchestration



Once Platform is Built – Time for Tuning

Tuning your Platform

ColdFusion Settings and Code Tuning

- Review ColdFusion Admin Settings
- Review older code for performance issues
 - Example look for overuse of cflock, evaluate functions, expensive SQL
- Look for opportunities to use Caching features
 - Trusted Cache, Query Cache, EHCACHE, Redis

JVM Tuning

- Memory Allocation
- Garbage Collecting

Visibility into Platform - Ability to Monitor and Troubleshoot

- Make sure to have ongoing visibility into Servers Health
- Make sure you have alerts sending your Stack traces for troubleshooting servers
 - ColdFusion Server Manager Alerts
- CF 2018 Server Monitor Enhancements
- API Manager with Metrics
- Many products are out there:
 - FusionReactor by Intergral
 - SeeFusion by Webapper
 - FuseGuard Security Filters by Foundeo
 - Many other more Enterprise-wide Products too

Know what this buttons do before Starting them.

 STOP MONITORING  START PROFILING  START MEMORY TRACKING

Alert Configuration

Unresponsive Server | Slow Server | JVM Memory | Timeouts | Email Settings

If the number of threads specified by Hung Thread Count execute for longer than the time specified by Busy Thread Time, the server is considered to be unresponsive.

Enable

Notify Client

Event

Hung Thread Count:

Busy Thread Time: (secs)

Actions

Send E-mail.

Dump snapshot.

Kill Threads running longer than (secs)

Reject any new requests.

Processing CFC:

Great alert when the errors in the exception log are not giving you a direction.

** Dump Snapshot checkbox requires Monitoring to be Started

Visibility - Many Monitoring Products Available



Check Out ColdFusion 2018 Monitoring Session at 3pm

3:00pm-3:50pm

**Introducing the Brand New
Performance Monitoring
Toolkit in ColdFusion 2018**

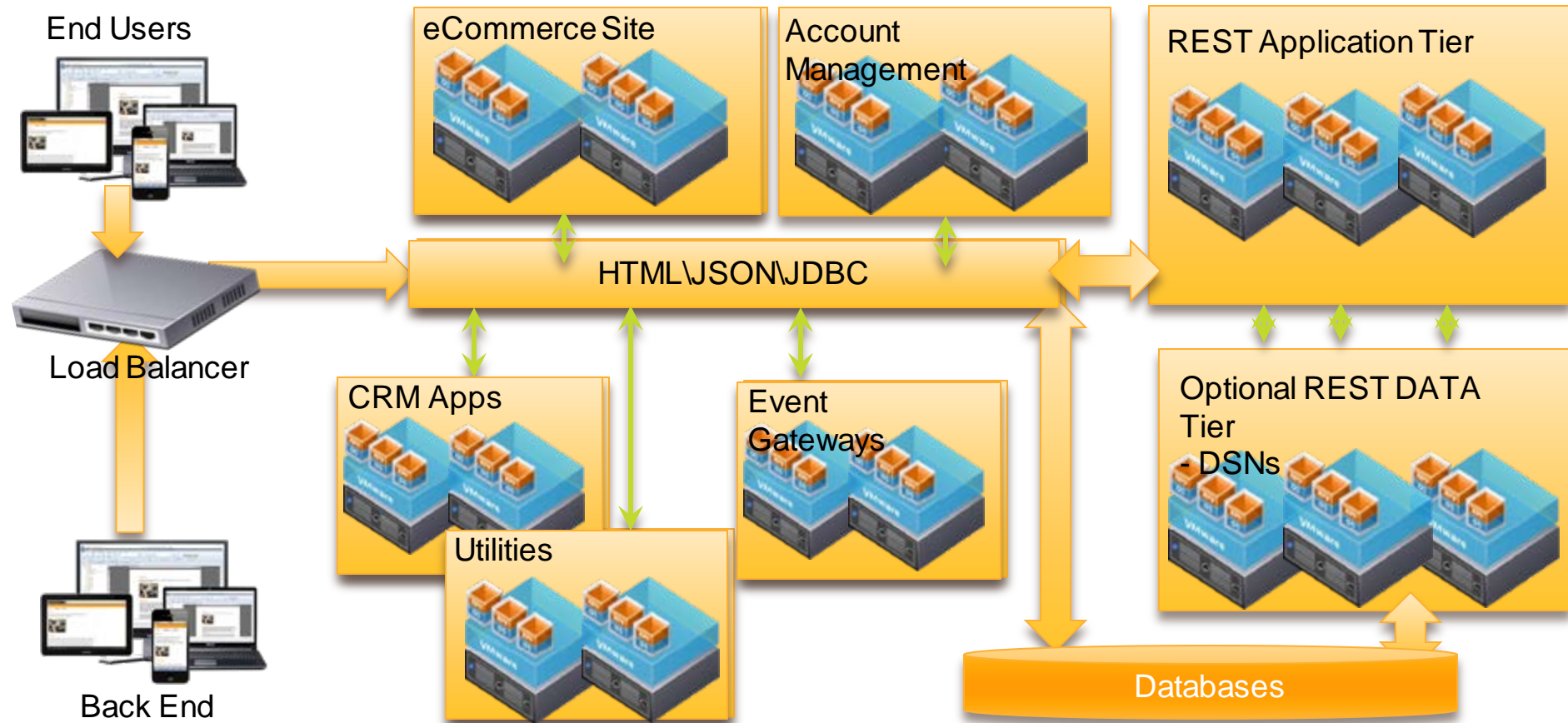
Rakshith Naresh

Many Use-Cases Exist for ColdFusion Applications

- Classic Applications - Html with CF code mixed in with some cfquery calls
- DB heavy
- LDAP queries
- CFImage conversions
- CFDocument PDF Creation heavy
- External web service or HTTP calls
- REST API based apps

Might look at isolating and tuning VMs separately for better scalability.

End Goal – Scalable Company Wide Topology



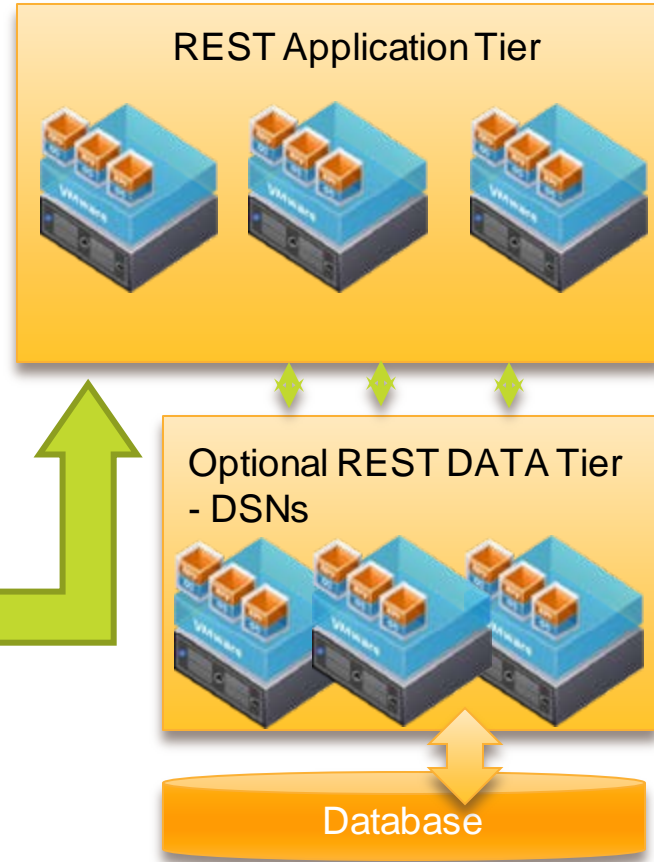
End Goal – Scalable Single Client App – API Based Topology

- Single Page Apps
- REST API base applications
- JavaScript Apps
- Mobile Apps
- API Access



Load
Balancer(s)

JSON/HTML



Key Metrics



Key Performance Metrics

- Thresholds
 - CPU Levels
 - Memory Levels
 - Requests per Second
- Have Target Performance Times
- Have a Target Throughput – how many request can you run and still have acceptable performance while keeping CPU and Memory within acceptable levels
- Goal is to not under or over allocate resources
 - Under and you suffer performance issues
 - Over and your paying more then you should



Learning Your Key Metrics - Load Testing

- **We can learn your thresholds by load testing your servers**
- **Develop several load test scripts:**
 - **Mixed or realistic load patterns**
 - **Targeted Scripts at one endpoint**
- **Run Scripts and watch how your Key Metrics perform**
- **Determine your comfort levels with CPU and memory**
- **Resulting in knowing a capacity for simultaneous requests**
- **Stress ColdFusion and Monitor CPU and Memory**



Learning Your Key Metrics Demo

Processes ■ 57% CPU Usage ■ 119% Maximum Frequency

| Image | PID | Description | Status | Threads | CPU | Average CPU |
|---|-------|---------------------------------------|---------|---------|-----|-------------|
| <input type="checkbox"/> Image | | | | | | |
| <input type="checkbox"/> coldfusion.exe | 4264 | Adobe ColdFusion Launcher Application | Running | 99 | 21 | 2.51 |
| <input type="checkbox"/> coldfusion.exe | 17796 | Adobe ColdFusion Launcher Application | Running | 100 | 12 | 1.58 |
| <input type="checkbox"/> w3wp.exe | 1676 | IIS Worker Process | Running | 22 | 4 | 2.75 |

Services ■ 0% CPU Usage

Associated Handles

Associated Modules

CFSummit2017.jmx (C:\VMeter33\bin\CFSummit2017.jmx) - Apache JMeter (3.3 r1808647)

File Edit Search Run Options Help

00:00:08 0 15 / 15

Test Plan

- threadsettings
 - GetAllFilms
 - GetAllFilms_querycached
 - Aggregate Report**
- WorkBench

Aggregate Report

Name: Aggregate Report

Comments:

Write results to file / Read from file

Filename Log/Display Only: Errors Successes

| Label | # Sam... | Avera... | Median | 90% L... | 95% ... | 99% L... | Min | Max | Error... | Thro... | Recei... | Sent ... |
|-------------------------|----------|----------|--------|----------|---------|----------|-----|-----|----------|---------|----------|----------|
| GetAllFilms | 1405 | 122 | 81 | 251 | 274 | 316 | 9 | 753 | 0.00% | 1.7/... | 1.21 | 0.01 |
| GetAllFilms_querycached | 1398 | 118 | 79 | 250 | 270 | 293 | 9 | 703 | 0.00% | 1.7/... | 1.21 | 0.01 |
| TOTAL | 2803 | 120 | 80 | 250 | 272 | 307 | 9 | 753 | 0.00% | 3.5/... | 2.42 | 0.01 |

CPU - Total 100%

60 Seconds

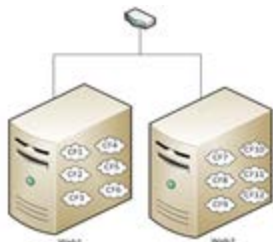
Service CPU Usage 0%

My laptop Dell XPS 9550 Intel i7-6700HQ CPU @ 2.60GHz, 4 Core(s), 8 Logical Processor(s) RAM: 32.0 GB

Scaling Concepts

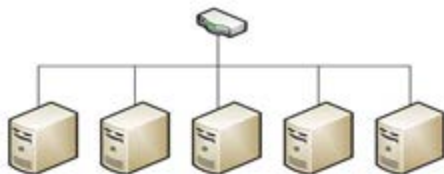
Scaling Techniques - VM or Physical Hardware

Vertical



- Many instances on each peer
- Greatest utilization of hardware
- Most cost-effective
- Need to consider hardware failures

Horizontal



- Peers are on other physical servers
- Protects against hardware failures
- Higher costs

Distributed



- Peers are located across physical servers
- Each server has multiple CF Instances
- Protects against hardware failures
- Higher utilization of available CPUs

VM Building Blocks

- **Establish a baseline building block VM for your Application**
 - OS, CF settings, Web server, JVM Settings
 - Use to Clone to other servers
 - Establish Vertical scalability
- **Scale Up Test each VM – CF Instances**
 - Establish how many CF Instances can you run on each VM
 - Determine how many vCPUs
 - Determine how much RAM to allocate to each VM
- **Scale Out Test - VMs**
 - How many VMs do you need to meet your Response Time SLAs without reaching 70%-80% saturation of CPU?
 - Create a testing environments where you can get accurate numbers



LB, Scaling and Failover by Category

Category One

Cost \$

- One server \ VM
- 2-3 Instances of CF
- Load Balanced with Tomcat Connector
- Shared code base
- Session might be lost on failover
- Email Alerts

Category Two

Cost \$\$

- Two servers \ VMs
- 2-3 Instances per server
- Load Balanced with Tomcat Connector or Hardware Load Balancer
- Code is synched or build updates both servers
- Smaller chance of lost sessions on failover
- Email Alerts

Category Three

Cost \$\$\$

- Three servers \ VMs
- 2-3 Instances per server
- Load Balance with Hardware LB
- Code is synched or build script pushes to servers
- Seamless session state on failover
- Cached Content
- Full monitoring

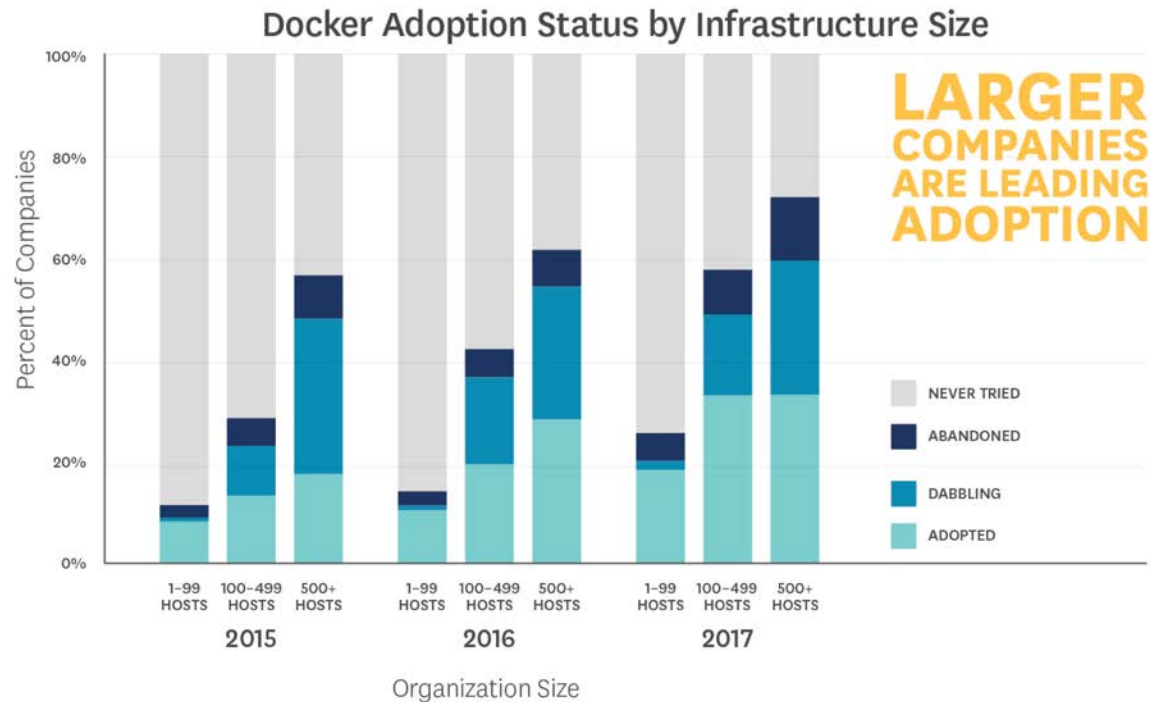
Category Four

Cost \$\$\$\$

- Six or more servers \ VMs
- 2-3 Instances per application
- Load Balance with Hardware LB
- Multi location
- VM replication for backup
- Code is synched
- Seamless session state on failover
- Dist Cached Content
- Full Monitoring

Looking forward - Docker Adoption

- Please share your story on the ColdFusion Portal



Source: Datadog

Looking Forward - Scaling with Container Orchestration



- Docker Swarm

- ECS



- Kontena

- Nomad



- Kubernetes

- Rancher

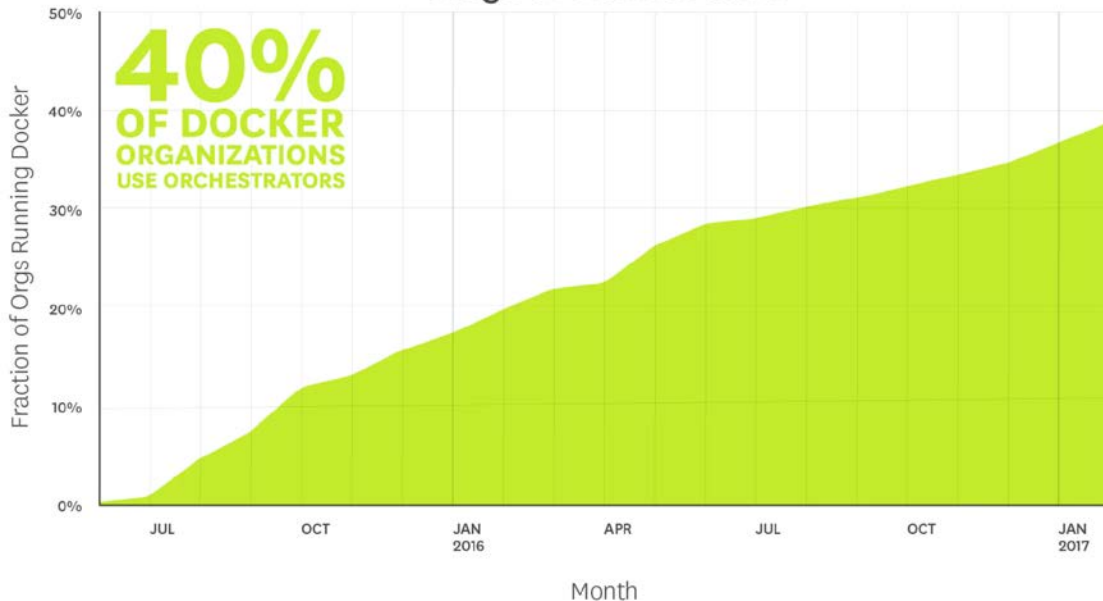


- OpenShift

- DC/OS

© 2016 Mesosphere, Inc. All Rights Reserved

Usage of Orchestrators



Source: Datadog

Choosing Container Orchestrator

- Many of the same concepts apply with knowing your thresholds and key metrics.

FUNCTIONAL CAPABILITIES

SCHEDULING

- Placement
- Replication/Scaling
- Resurrection
- Rescheduling
- Rolling Deployment
- Upgrades
- Downgrades
- Collocation

RESOURCE MANAGEMENT

- Memory
- CPU
- GPU
- Volumes
- Ports
- IPs

SERVICE MANAGEMENT

- Labels
- Groups/Namespaces
- Dependencies
- Load Balancing
- Readiness Checking

Understanding ColdFusion Clustering

Understanding Clustering – Creating Instances

- Uses the **cfusion** instance to seed the new instance
- Do not use **cfusion** inside a defined cluster
- Use **cfusion** as your baseline instance
 - Setup any default data sources and settings in cfusion
 - Settings are brought over
 - Make sure JVM arguments do not conflict such as port numbers or any custom settings
 - Review ports in {instancename}/runtime/config/server.xml
 - JVMRoute Property should be unique in server.xml

ColdFusion Enterprise lets you create and manage multiple ColdFusion servers



Add New ColdFusion Server

Server Name

Server Directory

Create Windows Service

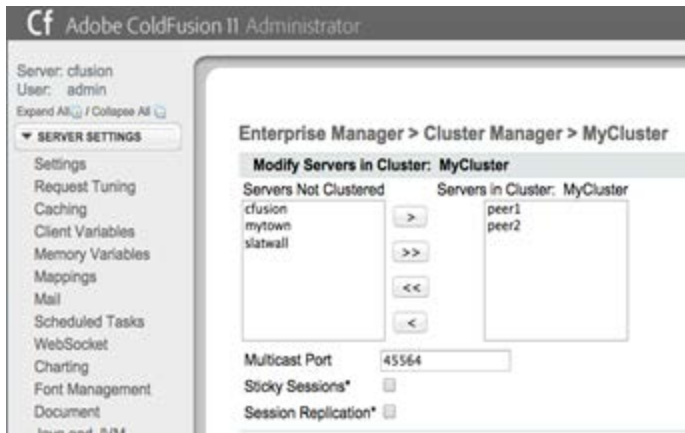
Submit Cancel

- Requires CF Enterprise

ColdFusion Clustering Basics

- **Basics**

- Allows for 2 or more peer servers to work together
- Uses Multicast to auto recognize when peers are up or down
- Peers can be local or remote



- **Defining a Cluster**

- Can be defined in the ColdFusion Admin
- Cluster network settings stored in the server.xml

- **Load Balancing Setup**

- Happens when you run wsconfig to setup a Connector
- Wsconfig.exe will use the clustering settings to build the load balancing rules
- Additional Properties and settings are located in these files
 - Cluster.xml
 - Server.xml

ColdFusion Clustering vs Load Balancing

Clustering

- Auto discovery of peers required to support in memory session replication
- Edits the Server.xml with multicast settings so peers can have a backchannel to communicate
- **Provides support for in memory session replication**
- Keep in mind in-memory replication is not 100% reliable

Load balancing

Happens when you run the wsconfig and choose a predefined CF Cluster

Do I even need to setup a cluster if I just want load balancing?

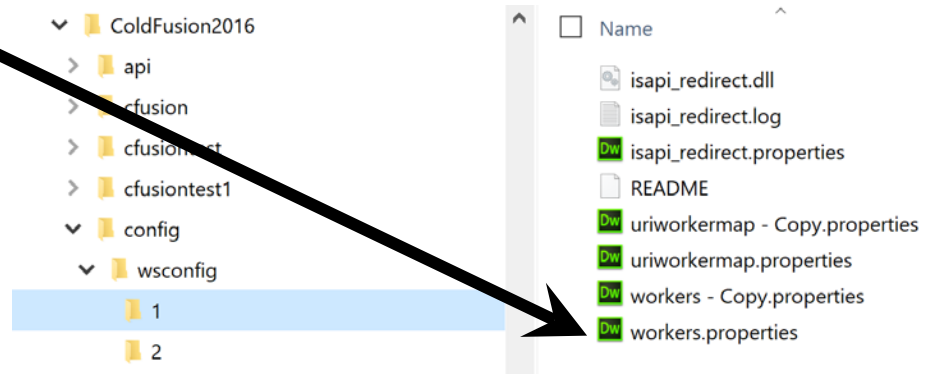
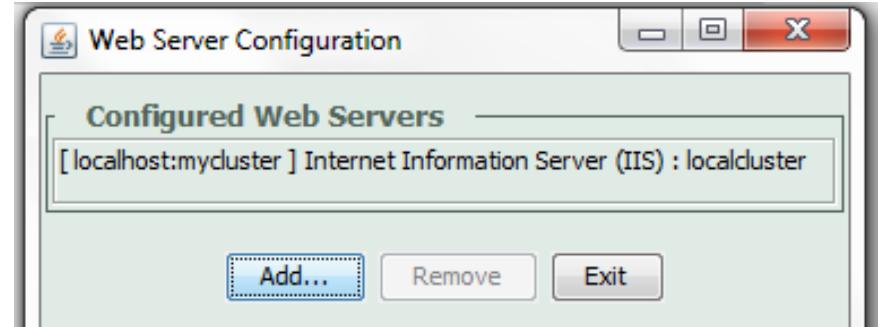
- No you don't
- However you might want to use it to initially setup your connector files to get started with load balancing
- Once your connector settings are setup to load balance you can delete your cluster in the CF Admin

Load balancing does not require CF Enterprise just make sure each CF instance has it's own Standard license.

Tomcat Load Balancing

ColdFusion Connector Setup for a Cluster

- Run wsconfig.exe command
- Once completed you will get a numbered directory with configuration
- Once the connector is complete your load balancing definition is now defined in the workers.properties file



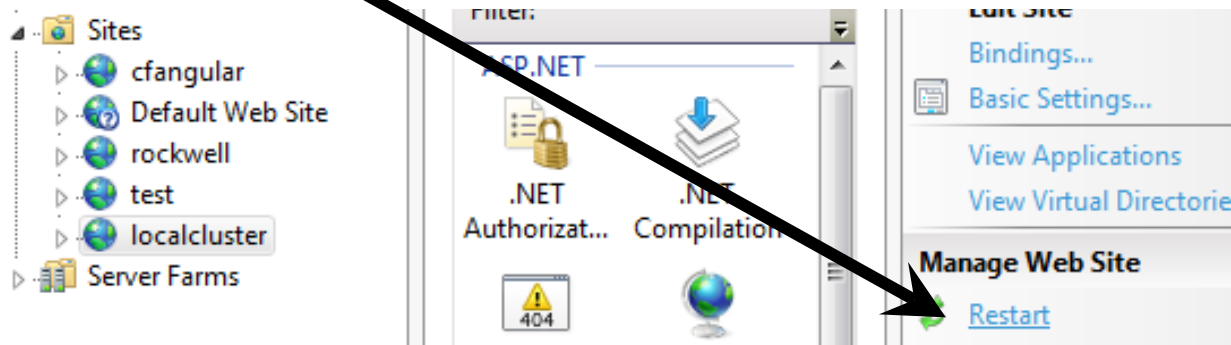
Connector Setup for Cluster – workers.properties

- Since we are using the Cluster ColdFusion will give you a worker.properties that looks something this →
- **Hosts**
 - Can be local or remote
 - The CF Server needs a Connector defined in the server.xml
- **Status Worker**
 - Allows you to monitor and edit peers
 - Mount to a protected url path
 - Restarting website in IIS will reload connector settings
 - Make sure to secure Status Worker
- **Post CF 11 Updater 4 – connector property files are backed up. Prior they were replaced.**

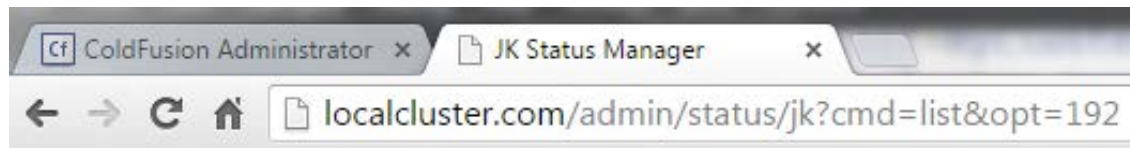
```
worker.list=mycluster,status
worker.mycluster.type=lb
worker.mycluster.balance_workers=peer1,peer2,peer3
worker.mycluster.sticky_session=false
worker.balancer.method=B
worker.peer1.type=ajp13
worker.peer1.host=localhost
worker.peer1.port=8013
worker.peer1.max_reuse_connections=250
worker.peer1.lbfactor=1
worker.peer1.route=peer1
worker.peer2.type=ajp13
worker.peer2.host=localhost
worker.peer2.port=8016
worker.peer2.max_reuse_connections=250
worker.peer2.lbfactor=1
worker.peer2.route=peer2
worker.status.type=status
worker.status.mount=/admin/status/jk
```

Changing the Workers.Properties

- You can edit worker properties anytime
- Add Peers, Disable Peers
- Change Load Balancer settings
- Simply Restart web site reload the connector which will bring in new settings.



Using the Connector Admin aka Status Worker



- Built into the Tomcat Connector

| Name | Type | Hostname | Address:Port | Connection Pool Timeout | Connect Timeout | Prepost Timeout | Reply Timeout | Retries | Recovery Options | Max Packet Size |
|-------|-------|-----------|----------------|-------------------------|-----------------|-----------------|---------------|---------|------------------|-----------------|
| peer1 | ajp13 | localhost | 127.0.0.1:8013 | 0 | 0 | 0 | 0 | 2 | 0 | 65536 |
| peer2 | ajp13 | localhost | 127.0.0.1:8016 | 0 | 0 | 0 | 0 | 2 | 0 | 65536 |
| peer3 | ajp13 | localhost | 127.0.0.1:8017 | 0 | 0 | 0 | 0 | 2 | 0 | 65536 |

| | Name | Act | State | D | F | M | V | Acc | Sess | Err | CE | RE | Wr | Rd | Busy | Max | Con | Route | RR | Cd | Rs | LR | |
|-----------|-------|-----|---------|---|---|---|---|-----|---------|-----|---------|----|----|----|---------|-----|---------|-------|----|----|-------|-----|----|
| [S][E][R] | peer1 | ACT | OK/IDLE | 0 | 1 | 1 | 0 | 0 | (0/sec) | 0 | (0/sec) | 0 | 0 | 0 | (0/sec) | 0 | (0/sec) | 0 | 0 | 0 | peer1 | 0/0 | 50 |
| [S][E][R] | peer2 | ACT | OK/IDLE | 0 | 1 | 1 | 0 | 0 | (0/sec) | 0 | (0/sec) | 0 | 0 | 0 | (0/sec) | 0 | (0/sec) | 0 | 0 | 0 | peer2 | 0/0 | 50 |
| [S][E][R] | peer3 | ACT | OK/IDLE | 0 | 1 | 1 | 0 | 0 | (0/sec) | 0 | (0/sec) | 0 | 0 | 0 | (0/sec) | 0 | (0/sec) | 0 | 0 | 0 | peer3 | 0/0 | 50 |

- Demo Status Worker

Session Management

Different Ways to Manage Sessions

Session Failover Options

1. Client Scope stored in Database
2. Tomcat Clustering with in-memory session Replication
3. Use Distributed Caching solution
 - Use CF2016 Redis Support
 - EHCACHE
 - New options with CF2018
4. Use Custom Solution

Server Goes Down

End user is sent to new server , Seamlessly continue with app

Routing Traffic

Sticky Session, Round Robin Requests

Support for Single Sign On

- NTLM
- REST API Key
- OAUTH, SiteMinder
- Active Directory \ LDAP

No Server Failover Options

- Sticky Session, with no session backup, Each session always returns to same server
- **Server goes Down** End User re-logs if using authenticated session

Using Client Scope Stored In a Database

Any variable stored in the client scope will be saved to the database.

All the servers would then share the database

Prior to each request ColdFusion initializes the client scope with a DB call

Description

Client Scope Stored in a Database

Create Client database tables

Enable this option only if this is the first time you are configuring the current data source for client variable storage. If the current data source has already been configured for client variable storage, do not enable this option.

Purge data for clients that remain unvisited for days

Enable this option if you want ColdFusion to periodically purge client data that has not been accessed in the specified number of days.


If this data source is being used by more than one ColdFusion server, as in the case of clustered servers, make sure that only one server in the cluster is configured to purge client data.

Disable global client variable updates

This option controls how ColdFusion updates global client variables, such as HITCOUNT and LASTVISIT. If updates are disabled, ColdFusion updates these variables only when they are set or modified. If updates are enabled, ColdFusion updates global client variables for each page request.

Select default storage mechanism for Client Sessions

| Actions | Storage Name | Description |
|--|--------------|-----------------------------------|
| <input type="radio"/> <input checked="" type="radio"/> <input type="radio"/> | cfartgallery | Client Scope Stored in a Database |

 Click arrow on left to return without submitting changes...

Submit Changes

Server Settings > Memory Variables

Application variables expire when you restart the ColdFusion server. Session variables expire when the user's session ends. Both types of variables also expire after a timeout period that you specify on this page or in the cfapplication tag.

Use J2EE session variables

Enable Application Variables **Enable Session Variables** (when unchecked, CSRF protection is disabled)

Session Storage Settings

The following settings control where ColdFusion server stores the session variables. These settings only apply when option to use J2EE session variables is not enabled.

| | |
|--|------------------------------------|
| Session Storage | <input type="text" value="Redis"/> |
| Redis Server | <input type="text"/> |
| Redis Server Port | <input type="text" value="0"/> |
| Password | <input type="text"/> |
| <input type="button" value="Verify Connection"/> | |

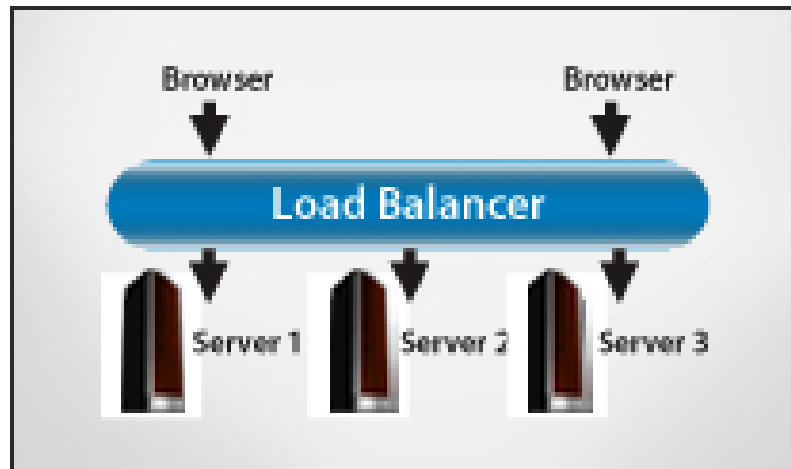
Hardware LB vs Software LB

- **Hardware LB**

- Most common on scalable architectures
- Network Probe determines outage
- Also supports HTTP probe which can call a CF page to determine database health etc
- F5 and BigIP are the most popular devices

- **Software LB**

- Offers low cost option to seamlessly take out peers from the pool
- Great for testing simulating failover scenarios outside production
- Tomcat Connector Load Balancing
- HAProxy



Summary

- No matter the platform
 - Measure and know your Key Performance Metrics
 - Know your thresholds for Requests per second, CPU and Memory
 - Only then will you know how many VMs you need or instructions to provide the Container Orchestrator ie Docker Swarm, Kubernetes
 - Learn your Throughput numbers using load testing
- Session Management
 - Choose a Session Management strategy to meets your needs
- Authentication
 - Choose an authentication strategy that will fit
- Use Tomcat Load Balancer with Status Worker for testing session failover
- Visibility
 - Make sure you plan to have visibility into your servers with monitoring and metrics logging



Adobe

Questions

ADOBE COLDFUSION SUMMIT EAST

WASHINGTON, D.C. | APRIL 25, 2018