

Running ColdFusion 10 in Distributed Mode

Posted At : October 20, 2014 5:22 PM | Posted By : Mark Kruger

Related Categories: ColdFusion

A recent discussion on [CF-Talk](#) had a member trying to find instructions for setting up ColdFusion in "Distributed mode" on a CF 11 server. If you have never heard of "distributed mode" you are not alone among CF developers. It's not a common setup in my experience. Here's a quick 101. Many processes and daemons on a given server communicate using the TCP stack. TCP provides a predictable, high capacity, mature protocol for piping requests and data in and out of server processes. In this case, IIS or Apache communicate with CF through the local loopback (localhost or 127.0.0.1) IP address and a port - usually 8012 (or 8013 or 8014) chosen at installation. Naturally you can alter the IP address to which you are connecting, changing it from the loopback to... well any IP that's listening on 8012. That means you can set up your ColdFusion servers separately from your web servers. After all it's just IP networking. Why would you do that? The Muse will let his guest handle that question.

Meanwhile, to preempt (or perhaps spur) discussion, the Muse will note that this process is similar to something called "Reverse proxy" which functions in much the same way. The difference being with CF distributed, IIS on the front end handles all the "http stuff" and passes the request to CF just like it was a local engine, whereas with reverse proxy the HTTP request is simply redirected to the alternate server. That means the alternate server needs to be a full webserver plus application server. While that increases the overhead a bit, it has some advantages - but that's a topic for a different post.

Back to our CF-Talk question, it quickly became apparent that not many folks actually knew how to accomplish this task on a CF 10 (or 11) server due to the underlying platform switch to Tomcat. After some back and forth [Byron Mann](#) chimed in with some very specific instructions on how to get this done. Byron is a lead engineer at [HostMySite](#) - which makes the Muse feel better about their ColdFusion support. :) Here's his tutorial.

Byron Mann writes...

We actually run our internal CF sites in distributed mode. ColdFusion 10 or higher (on Tomcat) works better than ColdFusion 9 or below. Previously it seems to work very slowly and the clustering (at least in our implementation) was never up to snuff. Note: you can accomplish pretty much the same thing with reverse proxy but the ColdFusion distributed mode is still pretty useful. CF can still peg CPUs from time to time, and if you have other non-ColdFusion sites on the web server (as we usually do), it is nice to have those continue to run unconcerned while the lone CF site dies a horrible death on different servers.

My best suggestion for ease of configuration would be to go ahead and install IIS on the ColdFusion server (even though you are not going to use it in production) and then run the wsconfig.exe in the c:\ColdFusion10\cfusion\runtime\bin folder. This will create the configuration files as well as the handler mappings and virtual directory for the isapi_redirect.dll. Copy the c:\ColdFusion10\config to your web server from the CF server. Then recreate the IIS handler mappings and jakarta virtual directory to mimic the CF server's IIS configuration. Don't remove the files from the CF server - you can use them as a reference for setting up your web servers the same way each time. Also,

you will need to add the isapi_redirect.dll as an allowed executable. That permission can be found in IIS under the server root ISAPI instructions. Make sure the website application pool user has write permission to the 'config' directory.

Connector vs. Reverse Proxy

As an aside, here's my take on why the connector is a little more useful than reverse proxy. If you have more than one CF server on the backend, it can act as a kind of load balancer. We have 12 CF instances backing our applications. You can manually change the isapi_redirect worker files to create a pseudo cluster. You can also do things like redirect specific sites or sub-folders to a particular CF instances. One example I have is a CF service that routinely timeout and kills the server (as it calls another SOAP service off-site). I have this particular service request redirected to it's own CF instance to minimize it killing others and effecting other sites. I'm sure you can do such things with a reverse proxy as well, but the connector seems to be a more simple solution. As you might imagine there are 20 ways to catch a fish. :)

Meanwhile, the documentation for the Tomcat connector can be found at [this link](#), but don't follow the instructions for setting it up with IIS, it has unnecessary information about registry keys etc.. I just use it as a reference for the properties in the config files.

Here is a snippet from the workers.properties file for our cluster. Keep in mind that when I say cluster, I'm not speaking of a "ColdFusion cluster" (where sessions are replicated across instances). Instead I'm speaking of balancing. The connector acts like a simple load balancer between your CF instances. Your ports may differ than 8013. I think 8009 is the default. You can see this in your CF instance in the web.xml file for tomcat under the c:\ColdFusion10\cfusion\runtime\conf\server.xml folder. Look for something like the xml node below. It may be commented out. If so you will need to un-comment it and restart CF.

```
<Connector port="8013" redirectPort="8443" protocol="AJP/1.3"
tomcatAuthentication="false"/>
```

```
# workers.properties
worker.list=cluster-1

worker.cluster-1.type=lb
worker.cluster-1.balance_workers=cfusion-3,cfusion-5
worker.cluster-1.sticky_session=true

worker.cfusion-3.type=ajp13
worker.cfusion-3.host=10.78.4.57
worker.cfusion-3.port=8013
worker.cfusion-3.max_reuse_connections=250
worker.cfusion-3.lbfactor=1
worker.cfusion-3.route=cfusion-3

worker.cfusion-5.type=ajp13
worker.cfusion-5.host=10.78.4.58
worker.cfusion-5.port=8013
worker.cfusion-5.max_reuse_connections=250
```

```
worker.cfusion-5.lbfactor=1  
worker.cfusion-5.route=cfusion-5
```

Then your uriworkermap.properties would look something like this (Note: I removed some of the default stuff):

```
# uriworkermap.properties  
  
/rest/* = cluster-1  
/*.cfml/* = cluster-1  
/*.mxml = cluster-1  
/*.cfm = cluster-1  
/*.cfm/* = cluster-1  
/*.cfml = cluster-1  
/*.cfc = cluster-1  
/*.cfc/* = cluster-1
```

Finally it's a good idea to mimic the same directory structure for your web apps between the ColdFusion server and the web server. We use DFRS to replicate all the files, but other methods will work as well.

Muse Summary

Now I know there's some assumed knowledge left out of this post, but this can get you started - particularly if you migrating from CF 9 or lower to CF 10 and you are already handy with distributed mode. Meanwhile, I welcome constructive comments that can add to our joint knowledge.