

Sick Server Troubleshooting Part 2 - Things to Try

Posted At : August 11, 2008 9:08 PM | Posted By : Mark Kruger

Related Categories: Coldfusion Troubleshooting

In step 1 we discussed gathering information. Without getting good information you are shooting in the dark. Make sure you take the time to know the system well enough to make educated guesses about what to try. That's our next step - trying stuff. I started out making a priority list... as in first try A, then B, but it soon became obvious that it wouldn't do to dictate the order in which you would attempt any of these changes.

In fact, some of these changes fall more under the auspices of "best practice" for configuration and you should probably do them regardless of whether they fix your problem or not. Still, if you are tracking a particular issue then you might need to try something, test, and wait for a result. So here are my "things to try" in no particular order. Your experience will have to help you figure out where to go first.

Things To Try

Traffic Patterns

Whoa... wait a minute. Before we jump the gun and try to "fix" anything, have you made sure that your issue isn't simply capacity? It is at least possible that you have reached a threshold for your current machine and you need to figure out what the next step is in capacity planning. If that's the case, you should probably still go through the check list below, but you need to start thinking more clearly about what comes next. Do you need a new server? Does the DB need upgrading? Do you need a cluster? Do you need to move to 64bit computing?

Don't forget traffic spikes either. We have a few sites that are way under capacity 90% of the time. But they just so happen to run TV commercials that point people to the web site. During the commercial (and immediately afterward) the site runs at or over capacity as impulsive couch potatoes log in to see what's up. Our planning has to take such spikes into account and this is where your web logs and baselines will help you make good choices.

JVM Settings and Tuning

If you installed your Coldfusion Server and left the default settings alone, shame on you! The default settings are simply not appropriate for a production server. At the very least the heap size should have both a minimum and maximum amount and it should certainly be more than the maximum of 512 megs that ships with the CF engine. There are a great number of posts out there on this topic including [this one](#), and there are many Java arguments that can help in specific situations.

Simultaneous Requests

This is an important item that you should visit the first time you log into the ColdFusion administrator after an install. The Adobe docs say this setting should be 4 to 5 times the number of processors on your machine. In my experience using that as a rule can be way too low. See [this post](#) for more information on a case for a higher setting of this variable. On dual proc dual or quad cores we have seen very high requests setting perform very successfully.

Database Tuning and Performance

Focus some attention on the database. A poorly performing database will show up as a pegged JRUN process - yet ColdFusion somehow gets the blame. I've seen 2 server systems where the web server is a big beefy machine and the db server is a poor step sister. That is the reverse of what it should be. In my opinion the database server is nearly always the single most important server in a system. It should be the best server you have. And don't forget indexing. Your tables should be appropriately indexed and the indexes should be regularly repaired.

Client Variables

I wish they had never invented the "registry" method of storing client variables. If you are using client vars take the time to move them to a database on a database server (Hint - Access is not a database server). What can happen is that the client vars build up in the registry until the number of keys gets quite large. Periodically (every 1 hour and 7 minutes by default) Coldfusion will try to "purge" this information store. If there a great many entries this can bring your server to a halt. Whereas, if you are using a DB things like concurrency will be handled by the DB server instead. This issue can "seem" random unless you manage to figure out that it seems to happen every hour.

One more note on the topic of client vars. It is possible to specify the storage mechanism for your client vars as an attribute of your cfapplication tag. You can specify the *default* location in the cf admin and you can add other data stores to the list of possible client storage, but merely moving the default to a DB on a DB server will not work if the clientstorage attribute is set to registry or a different database. Do a quick search for the cfapplication tag with the optional clientstorage attribute set:

```
<cfapplication
  clientmanagement="Yes"
  clientstorage="*registry or DSN*">
```

Then, either remove the attribute or set it to your new DSN.

Networking

I won't detail this here, but to summarize, your NIC card and the downstream port settings should be "hard coded" to a static value for speed and duplex and not set to "auto discovery". Allowing the NIC to try to negotiate a better speed will result in phantom threads (calls to the DB usually) that are not readily visible anywhere but are still using up resources. Read more about this ticklish issue [here](#).

Log File Sizes

I have 2 tips here. First, if you are using the wildcard config on IIS 6 (Win2k3) and you have turned on verbose logging you might end up with ginormous log files in the /runtime/lib/wsconfig/logfiles directory. Here's an [explanation](#) that might help you resolve that issue. Secondly, the *-out.log in the /runtime/logs directory can grow quite large and will not roll over like the other files. If it gets too large ColdFusion might be working a bit too hard on disk I/O just appending to this file. You can safely delete or rename but you will have to stop ColdFusion to do it. See this post on [handling Jrun Log files](#). Also note, large log files are sometimes an indicator of poor error trapping. Errors cause excessive information to be written to logs, so keeping your code trim is a good way to avoid this issue.

Debugging Off

As a rule, debugging should always be *disabled* on a production server. Not only is it unnecessary overhead, but the amount of information provided by verbose errors in ColdFusion is a buffet for hackers. Appropriate error handling is also relevant to security (less so to performance). Of course you might think that everything is okey dokey because only the loopback IP address (127.0.0.1) is specified in the list of debugging IP addresses. But actually, ColdFusion develops debug information behind the scenes whether it displays it or not, so you are still adding to the load. Also keep in mind that scheduled tasks (if you have any) present the loopback address as their own `cgi.remote_addr`. This means that these schedule tasks receive the full buffer of debug info each time they run. Since scheduled tasks often do heavy lifting tasks (import or clean up routines with dozens of queries for example) this means an undo burden and lengthening of these tasks - and it will make you scratch your head wondering why they take 3 times as long as when you run them manually in a browser.

Database Connections

In your data source settings do not use IP addresses. Doing so causes JDBC to do a reverse lookup when making the first connection. I know I know it probably doesn't matter - but I'm throwing it in here anyway. If, for some reason, the DNS server is down or slow this can cause connection issues. Of course using a FQLDN will have the same issues since a DNS query is involved there as well. You can use a host file but make sure and document it and don't forget how that name is resolved. Please note, this is a trivial setting and probably not germane to your troubleshooting.

Miscellaneous Stuff

These all fall into the general category of "stuff to check" that may or may not mean anything to your specific problem.

- **Look for System Errors** - I/O errors aren't ColdFusion specific, but they can affect things like mail, logging, and write operations.
- **Check Processes and services** - is there a correlation between the outage and a backup process or a scheduled task for example. Perhaps the virus scanner is scanning your ColdFusion directory and keeps locking log or class files as they change. Eliminate un-needed processes (scan your files periodically but unless users are allowed to upload files don't scan continuously). Shut down MS Indexing (do you really need shell access to be able to find files super fast?). De-install unneeded software. Why is Microsoft Office installed on your server anyway? Shut down services that are not being used. Do you need DFS? The Spooler? The server should be carefully tuned to be running *only what it needs to accomplish it's tasks*. It surprising how many folks install a Windows server with all the default services enabled.
- **Traffic Patterns** - Look in the web logs for spikes in traffic or excessive activity from web bots or crawlers. For example, [recent SQLi attacks](#) have resulted in an exponential increase in requests targeting ColdFusion servers who have a significant number of pages indexed on search engines.
- **Database Locks** - Look in your DB tools for lock escalation problems. The MS SQL2005 "activity monitor" for example shows running processes and locking levels. It can also indicate blocks due to escalation (where a lock on 1 resource is blocking a request for a lock on another - usually more granular resource. These can over work your DB server as well as peg your web server as the system queues

up waiting for resources to be released. Usually such situations indicate either poor schema design or a resource constrained database server.

3rd Party Tools

Finally, there are 2 (possibly more) excellent third party tools out there that can be of immeasurable value. They are [See Fusion](#) and [Fusion Reactor](#). Both of these allow you to monitor threads, fire off garbage collection and set alerts. They each have strengths and weaknesses, but either of them will give you a great jump start toward digging into your server and its behavior. Of course, ColdFusion 8 Enterprise ships with a monitor that is pretty good as well - though I find the main "overview" of the other 2 tools gives me a better immediate snapshot of what is happening at this very moment.

Ok... you've tried everything above and you are still having problems. What's next? Next is the stuff that is hard to find, hard to test and hard to fix. It's the really sticky stuff.