

ColdFusion 9.01 Server Monitoring Enhancements

Posted At : July 13, 2010 10:38 AM | Posted By : Steven Erat

Related Categories: Adobe, Quality Assurance, ColdFusion

I began this thought as a comment to Adobe CF QA engineer [Sagar Ganatra's blog entry](#) describing the new Server Monitor enhancements in ColdFusion 9.01 updater, however, as it grew lengthy I decided my own blog post would be a more appropriate venue.

I'll add that the main reason for **why** one would want to run the ColdFusion Server Monitor on its own port via the Jetty implementation is that until now requests from the Server Monitor would go through the JRPP request pool, thereby adding additional traffic to the JRun active request pool, but more importantly if the JRun active request pool was queuing then the data refreshes in the Server Monitor would also queue and the Server Monitor may appear to hang as well. By establishing a separate request pool and port for Server Monitor requests in ColdFusion 9.01, the Server Monitor will not encounter a blocking situation as it would do previously.

Any general discussion of the Server Monitor should include the caveat that the use of Profiling, Monitoring, and Memory Tracking are not intended for production use (*blanket statement: see comments for more on that*). Moreover, if Memory Tracking is enabled in production, perhaps to help diagnose a prod performance problem, that it will only further decrease server performance. On 3 occasions in the last year alone I've helped ColdFusion shops that shot themselves in the foot by doing this. The impact to performance was substantially worse when enabled, and having them disable it immediately alleviated most of the problems, albeit not the problem that initially prompted them to enable the tracking.

Nice enhancements to Server Monitor in the future might include:

- **Persistent Metrics:** The ability to persist the Server Monitor data to a database. A restart of CF will clear data as of now. A use case would be for load testing scenarios where the a variety of metrics need to be quantitatively analyzed, which cannot presently done easily. Ideally, you'd want to know performance metrics at different points into a load test such as during the ramp up, after X minute intervals, and during cool down.

A second use case would be the ability to produce reports to monitor server health over time, perhaps by providing the ability to generate weekly reports of key data, possibly with green and red arrow indicator to visually identify metrics that have improved or worsened.

- **CF Request Pool Ratios:** Add the ability to analyze incoming requests to determine the real time ratio of CF request types such as CF Templates, CFCs, Flash Remoting, and Web Services. When expressed as a percentage, it could used to correctly determine the values to use in the **Request Tuning** part of the ColdFusion Administrator. Throughout much of my experience I have found that

CF shops rarely set the Request Tuning values to an appropriate range, either letting them remain unchanged at the default, or increasing them way out of range (into the hundreds even).

- **Request Tuning Calculator:** Provide a real suggested starting point for all Request tuning parameters including the JRun active and queued sizes based on the number of CPUs/cores and processing speeds. Presently, even in CF 901, a server will install with a default set of values that will be the same on a small box as it would on a beefy production box. To do this correctly, the total number of instances used to process production load would also have to be incorporated for proper per-instance tuning. Having a Server Monitor Request Tuning Calculator would be a big plus towards helping server admins **find the right starting range** for their particular hardware.
- **Out of Process Memory Tracking:** Since Memory Tracking is known to consume significant resources while tracking (a Heisenberg conundrum?), perhaps Memory Tracking could be done out of process over RMI or similar, akin to the **JRockit Mission Control memory analyzer** (which I've used for ColdFusion, but interpreting the data is not very intuitive).