Reducing the Wait for Garbage Collection

When we're profiling application code, there's one common problem in the Java operating environment that has the potential to affect the entire application: Garbage Collection.

In code creation, Garbage Collection is a function of memory management used to eliminate memory allocation issues and increase productivity. Although the process should free up memory in use, it sometimes creates more problems than it solves. It can create latency and other performance issues.

Chuck Ezell, senior applications tuner at Datavail, explains:

When Garbage Collection happens, there are a lot of CPUs consumed. Everyone is waiting. All of the other JVMs, processes, and the threading that's going on within the JVMs will have to wait until the Garbage Collection process is complete. It can cause a lot of slowness and it can happen across the entire application. And if we're not monitoring for that — if we have no way to profile our code to see if it's a Garbage Collection problem — then we are lost.

Using JDK Tools

Some of the tools in the Java Developer Kit — jstack, jstat, and jmap — are easy to use and are free. They can be used to profile application code. These tools only run on the terminal and generate useful information, but don't retain any information. You can resolve this by creating a file of the raw data. Before you go to a new session or revert to a different directory, you can easily move the raw data into a file.

Ezell explains his process:

I'll name the file, I'll give it a server name, I'll give it a day. That way I can keep the history of what I've done. There's no analysis provided. It's just the raw data. But it does provide running threads. You can see the memory footprints of JVMs. You can also see Garbage Collection with jstat. You can actually pull live threads that allow you to look at Garbage Collection in a live JVM, in a production environment.

One means to do this is to use jmap to see the memory footprint, then look at the ebb and flow of memory allocation — which helps monitor Garbage Collection — in jstat. You can use it to look at Garbage Collection and different arguments, or the old memory capacity. By adding a timestamp, says Ezell, you can sample and discover what is occurring over time. This can be applied to a particular JVM to see what the Garbage Collection process is doing to the virtual machine over time.

11800 Ridge Pkwy., Suite 125, Broomfield, CO 80021
Phone: 877.722.8247 | Email: info@datavail.com
Website: www.datavail.com

Reactive, Not Proactive

These specific tools are reactive rather than proactive. You must be there when running a JDK tool and it has to be run when there is a problem. Adds Ezell:

They are great tools for firefighting in a reactive situation. They don't maintain any history or reporting. The output is essentially what you see. It will help you answer questions and it will help you identify real problems now.

Some other tools able to help monitor the process, but offering differing diagnostic capabilities, are HP Diagnostics and AppDynamics. AppDynamics can profile code, for example, down to the thread stack and exceptions without the need for instrumentation. It also works well with Splunk. You can also schedule automatic and diagnostic sessions that will send you e-mail notifications if too many Garbage Collection events are occurring.

This information should make those chores associated with diagnosing and resolving Garbage Collection issues much easier.

If you need assistance or ongoing help with profiling your code, please contact Datavail to discuss a custom solution for your enterprise.

About the Author: John Kaufling
Vice President and Practice Leader of Application Services, Datavail

John Kaufling has more than 20 years of experience in the IT industry, including more than 12 years as an Oracle EBS database administrator at Level 3 Communications and at Oracle Corporation. His specialties include implementations, upgrades, performance tuning and extensive capability to support the product. John's work with Oracle apps database administration has included experience with SOA suite, Veritas Cluster, Oracle DataGuard, Load Balancing from Resonate, Cisco and BigIP and extensive experience with Oracle self-service applications and self-service framework technology.