Selecting Application Code Profiling Tools

Profiling is a task confusing to some database administrators, but it is important because it allows you to absolutely see — without guessing or conjecture — what actually is happening with your application code.

You need the right tool for the job and context, whether it's a server profiling tool or a client profiling tool.

We need to regularly profile our code and examine the code running in our environments, otherwise, we're making assumptions about what's running. We need to work with facts. If we're having problems, we need to understand why. If we don't use profiling tools, then we can never really get to the root of the problem.

A recent study found the majority of the causes of failure in Web applications result from ill-planned changes, some of which are made by developers. Two-thirds of availability and performance errors result from misconfiguration. Problems in an enterprise stem not only from configuration problems or changes in the code, but also can result from hand-off issues, deployment problems, or issues with the application code itself.

The Human Factor

Automated tools can help us with profiling, but they aren't able to analyze nuances in the environment. A tool may be able to lead us to the source of the problem, but has no means to check whether there is a process in place or to follow through with improvements. You need the human factor, for example, to iterate through change management procedures that allow you to maintain a fix in an environment.

You have to tune through people using processes to make your code better. It takes the human factor to successfully profile your code. You can't solely rely on an automated tool.

Which Tools Are Right?

As we look at these tools available, what kind of questions do we need to ask? What should we consider when we are selecting the right tool for this job?

Here are a few questions I've asked myself:

- How hard is the tool to set up?
- How hard is the tool to manage?
- Is it something that we're just going to use one time, or is it the kind of tool that we want to set up? Maybe it's something you want to put in production for six months or maybe you want it integrated in your application development environment, so you're doing development and operations together.

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

- What does it profile?
- Does it profile resources, threading, and memory?
- What kind of reports will this product generate?
- Is the report something that I can hand to my superiors?
- And then of course, what's the budget? How much does the product cost?

Among the prevailing problems tools can address include garbage collection, memory problems, and thread and resource contention.

Useful Tools

Some of the useful tools I've found to handle various profiling tasks include jstack, jstat, and jmap, as well as Splunk, HP Diagnostics, and AppDynamics. There is a whole range of useful tools available. You might, for example, find Fiddler most appropriate if you don't have much time to work with a customer, or if you're working in a development environment in which the browser is on one side and you have Fiddler on the other.

With the exception of jstack, jstat and jmap, all of these products will work within a .net SQL environment, enabling them to be used in other environments, including Java JVM, and PHP.

There are a lot of different tools available for different jobs and different coding solutions. You'd do well to search online for these products to determine for yourself how they might help you profile your application code. You need to proactively determine what works best for you and your customers. You don't want to make assumptions. You do want to determine how to best profile your code within the context of transactions from a customer's point of view.

Are there tools you use you think work well? Let us know, we would love to hear from you.

About the Author: Chuck Ezell
Vice President & Practice Lead, Development, Tuning & Automation, Datavail

With more than 22 years of experience, Chuck Ezell supports some of the world's infrastructures with database performance tuning for their applications, automating processes, developing new database application solutions and tuning their ETLs. Chuck helps locate and eliminate system bottlenecks, while uncovering opportunities to improve performance on all application tiers. Although, he excels at optimizing and customizing Oracle systems, Chuck works with .NET, T-SQL, C#, PHP, javascript, Java, ANSI SQL, PL/SQL, APEX, and many other languages on systems by Oracle, HP, IBM, Linux and Windows. He uses SSMS, Oracle OEM, AppDynamics, Splunk, Visual VM along with many others, for performance tuning operations.