

Webserver and Throughput

Contrary to what many in the field think, proper performance from the web server is crucial to overall server side performance in a PeopleSoft implementation. Since the web layer depends heavily on the underlying JVM performance, throughput of webserver is nothing but the throughput of the JVM itself. Unlike a database server where there are several options available to change the desired throughput, the webserver has only two parameters in essence. The JVM heap size and the number of runnable threads dictate the throughput.

JVM heap

When the JVM is started it allocates the minimum specified heap in the virtual storage and continues to grow. The frequency of Full garbage collection and the number of Full garbage collection determine the service period of the server for the useful work. If in a given hour the disruption due to Full garbage collection is 10 minutes then the service period is only 50 minutes. If the machine is a multi CPU system, then the 10 minutes clock time could translate into several service units. That is a big concern for architects.

A standard workable heap size is 384 to 512 MB. If you were to decrease the heap size below 384, to 256 MB then you will need more JVMs. (Note that 256 MB is a bare minimum heap size for a PROD implementation.) These JVMs can reside in the same machine or in a different machine. If they reside on the same machine, then such architecture will drive the machine better. A 256 MB JVM with three or four Webserver instances is certainly one of the best possible designs. However problems arise when the number of webserver is limited and the heap size is also low. There is a transient nature of the PeopleSoft application that will actually increase the heap demand over a period of time. At peak load, you may very well observe that it is always operating close to the maximum heap size. This behavior almost certainly guarantees that the Full GC is inevitable. While a Full GC is just a normal JVM operation, its frequency is what is of significance.

Under similar conditions, a machine hosting one JVM with 512MB heap size will yield lesser throughput compared with the setup involving (sometimes even up to 30 – 40% improvement) two JVMs each with 256MB heap size, hosted in the same machine. To a large extent, the results are installation dependent. During stress testing you have to vary the heap size and measure the throughput. A major obstacle to the throughput is the frequency of full garbage collection.

In order to provide fail-over you may want to design multiple web servers. If you have enterprise PORTAL then you may want to leverage the same machine to house both the PORTAL and PIA webserver.

Runnable Threads

The execute thread count is another parameter that governs the throughput. The more the number of runnable threads, the better the throughput. Increasing the runnable threads after a certain threshold decreases the throughput because it shifts the problem at a different layer. This is more felt in operating systems that have one to one mapping for the JVM threads and OS threads. A good rule of thumb for PeopleSoft is 80. If you have enterprise PORTAL, then the number and structure of contents will determine the number of threads required.

Design your JVM and the architecture with the throughput in mind.

If you want to discuss more on this topic, feel free to talk directly to our Principal Consultant by calling 866-DB2-PSADMIN. He can also be reached at venkat@hewittandlarsen.com.