

“Abracadabra” of Application server performance

The Application server performance is very important in the online PeopleSoft workload. Proper server design, machine choice and load balancing options should be taken into considerations while designing the application server.

Etymology first

It is believed to have come into English via French from a Greek word abrasadabra (the change from s to c seems to have been through a confused transliteration of the Greek). It originated as a secret and mystical word with a Gnostic sect in Alexandria called the Basilidians (named after their founder Basilides of Egypt). It was probably based on Abrasax, the name of their supreme deity (Abraxas in Latin sources), but is sometimes said to have been constructed from the initial letters of three Hebrew Words: Ab, the father, Ben, the son, and Acadsch, the Holy Spirit. It was used as a charm, written in the shape of a triangle on a piece of parchment worn round the neck, and was believed to have the power to cure toothaches, malaria and other scourges. And Abraxas itself was said to have magical powers of its own, as a word that represented the number of days in the year, 365. Adding up the numerical values of its seven Greek letters by a process called gematria derived this. For this reason, it was often engraved on amulets and precious stones. It means difficult to comprehend.

The following is a discussion, which suits the title. I have tried to model certain main inner intricacies on which the throughput directly depends. Deliberately I have left out the normal performance boosters / topics that are easily understood such as CPU, memory and load balancing etc. Many documents can explain those easily understood concepts.

In a goal oriented tuning approach the application server throughput depends on the following three factors.

- 1.0 Number of server processes
- 2.0 Read I/O rate on cache file system
- 3.0 Working set of PSAPPSRV

Queuing Model

Application server processes managed by Tuxedo works based on a queuing model. The workload goes into a queue from where it is de-queued. It is this queuing model that gives concurrency and parallelism within the application server.

There fore the first tuning goal is to determine the optimal number of server processes required to meet the workload at peak conditions.

When a server process queue is filled up and there are no available server processes to satisfy the requests, the delays and performance issues are imminent until one of the server processes frees up and is ready to serve the request.

Monitoring the queue length over peak period helps to size the server processes that would be necessary to meet adequate response time.

Caching Model

During normal processing, the application server PSAPPSRV processes obtain any structural information about various PeopleTools objects by querying the PeopleTools metadata. In order to speedup processing, the PSAPPSRV server process maintains a memory cache and additionally a DISK cache. The integrity of the cache structures is automatically checked every time such a need arises. (Every time a SQL SELECT is done against PSVERSION it is to maintain the cache coherency.)

Presumably this technique improves the response time. While the response time certainly improves by finding the object in memory cache, the response time can be poor if the DISK systems are accessed, especially in slow devices. In fact it could be poorer than accessing the database itself. Database systems have usually superior memory caching implementation and if all objects could be found in the buffers, the response time could very well be superior. Hence it is very important to make sure that we deploy the application server CACHE file system to fast DISK devices with a large controller cache. A large controller cache helps to improve the overall DISK response time. Defining the cache file system within the local DISK will almost certainly guarantee poor performance. Local Disks usually have less controller cache.

Each PSAPPSRV service will maintain its own cache structure. As you decrease the I/O response time, the response time will improve.

Shared Cache

Starting Version 8.14 PeopleSoft delivered a mechanism to share the cache files, which lets all the PSAPPSRV services to share the cache. The concept is good. But just by enabling the shared cache system, you will not get performance benefits. A shared cache when complemented with a large controller cache will increase the throughput on the application server. You will minimize the wait times and blocking conditions. As you remove those waits, the CPU could be put to more work. Under shared cache mode there is no Write operation to the disk because the cache is presumed preloaded.

Frequency of Recycle

Due to the memory cache implementation, the working set of PSAPPSRV continues to grow. In order to guard against the ill effects of an uncontrolled growth, PSAPPSRV services recycle themselves after a pre-determined number of service requests. This process clears the memory and controls the working set of PSAPPSRV service. As a general rule, monitor the average working set of PSAPPSRV service. If it grows beyond 150Mb then adjust the Recycle count downward. 150 MB is a good estimate and can serve as a guidance to size the memory in the system. PSAPPSRV and JSH are the two major processes consuming a lot of memory in application server. Since the recycle process clears the memory cache, frequent recycling will increase the I/O rate on the system. A good balance is necessary. The Recycle Count is specified in the application server configuration file. During peak workloads, if an unusual number of server processes start to recycle there will be brief pause in the effective throughput. It takes several SYSCALL events and CPU cycles to complete a RESTART. Proper value for Recycle count is also an important tuning effort.

If you use more than one application server domain, possibly in a load balancing setup, then choose different Recycle count values for each of them.

Overall Server Performance

Sufficient CPU, memory and network throughput is also crucial to application server performance. The network between the application server and the database should preferably be on the same LAN interconnected on a Gb backbone.

No wonder application server needs to be more powerful than the web server.

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.