

## Connection concentrator in a PeopleSoft environment

IBM is constantly working on enhancing the connection mechanism in the network computing environment or DRDA. The primary objective is to improve resource utilization and throughput. Connection concentrator, introduced with DB2 Connect EE version 7.0 tries to solve the issue of load balancing the host connections in a distributed environment. We shall see how the connection concentrator helps to load balance the PeopleSoft online workload. Even though we are talking about the PeopleSoft workload, the discussion applies to non-PeopleSoft workload as well.

### PeopleSoft online workload

PeopleSoft online workload primarily comprises of the application server processes such as PSAPPSRV.EXE, PSQRYSRV.EXE, and PSSAMSRV.EXE etc. PSAPPSRV.EXE handles a major part of the workload. In a typical PeopleSoft workload, these application server processes issue the database connects, SQL requests and the business logic processing. In order to control the throughput and efficiency of processing, a significant effort in the tuning process is to right size the number of such processes.

With the introduction of applications such as enterprise portal and various self-service type applications, almost all the employees of an enterprise are considered potential users of the system. The need for more and more application server processes grows with the introduction of more application modules. The more application server processes, the more the number of host connections. How are we going to manage these connections? Connection concentrator tries to address this issue. In order to review connection concentrator, we must review connection pooling first.

### Connection Pooling

When a client requests to communicate with the host, an agent task needs to be created. The request should be formatted into DRDA protocol and sent over TCP/IP to the host. The host processes the message and sends a reply. The agent then receives the reply and formats it to the client. At this point, prior to connection pooling, if there was any disconnect request as might happen from an application server recycle process, the agent disconnects the connection to the host connection to the client and terminates the server process. An agent is an engine dispatchable process in a UNIX environment and a thread in a Windows environment.

Since terminating a connection and then later re-initiating it is a resource intensive process, DB2 connect V 6 introduced the connection pooling mechanism. Connection pooling is achieved with the introduction of a new parameter called NUM\_POOLAGENTS. The default value of NUM\_POOLAGENTS is set to 2% of the MAXAGENTS value. In connection pooling, when a distributed process requests a disconnect request, the agent first checks to see if the value of NUM\_POOLAGENTS is reached. If so, the agent disconnects the connection to the host connection to the client and terminates the server process. If the value of NUM\_POOLAGENTS is not reached, it just terminates the client connection. The server process and the host connection are still available to the incoming workload, resulting in a possible reuse of connections.

In a PeopleSoft application, many installations do not pay attention to this minute fact and usually the NUM\_POOLAGENTS is left to default to 2% of the MAXAGENTS. In DB2 connect Ver 8 this default value has now been increased to 50% of the MAXAGENTS. For PeopleSoft, It is recommended to set NUM\_POOLAGENTS to a value equal to the number of PSAPPSRV processes.

Connection pooling has another big challenge in a PeopleSoft environment. While it tries to achieve improvement in resource utilization on the DB2 connect server, it does not offer good workload management in a data-sharing environment. It does not try to spread the workload evenly. In order to distribute the workload evenly it needs to know the WLM information. The WLM information is made available to it only after a database connect request. During a database connect request, it caches the WLM information so the WLM information is available to subsequent connect requests. Since it relies on previously cached information, it does not translate into accurate workload balancing. Moreover, the connection pooling effect as described above, tries to reuse a host connection if it is already available in the pool. In PeopleSoft, the database connect request is made at the application server startup time or when an application server process recycles itself. There is no explicit disconnect request. Let us consider this practical scenario. Typically the application servers are brought down for preventive maintenance and usually the maintenance window is when the systems are least loaded. Please click on the [link here for a schematic representation of connection pooling](#).

When the application server is booted, the cache is checked to connect the agent task to that member of the data sharing group which is least loaded. Since we are doing maintenance during off-hours, it is possible that all connections end up in one member. This is perfectly valid because at this time that member is least loaded. Subsequently when intensity of activity resumes, the host DB2 subsystem and the LPAR faces intense utilization. The sysplex is not evenly managed. One LPAR may shoot to 99% CPU busy rate while there is room in the other LPAR, which is at 30%.

### **Connection concentrator**

With connection concentrator introduced in DB2 connect ver 7, the problem of uneven workload balancing is solved. To enable connection concentrator, the parameter MAX\_LOGICAGENTS needs to be set higher than MAXAGENTS. DB2 connect should be SYSPLEX enabled.

This reduces the number of resources that maintain the host connections. At commit point, the logical agent breaks its association with the real agent and it can get associated with another real agent during subsequent database call needs. Logical agent scheduler associates a logical agent with the real agent or worker agent. Furthermore the WLM information is used to rebalance the workload. DB2 connect does not have to wait until a new connection to get the updated WLM information. PeopleSoft PSAPPSRV process manages the bulk of the SQL processing. The design of the PSAPPSRV process is in such a way that there are frequent commits. Hence it results in more frequent re-balancing of the workload. This is the single biggest advantage of the connection concentrator. It also needs fewer resources that hold database connections. But this is really not a big advantage for PeopleSoft workload because in the PeopleSoft architecture it is the application server processes that maintain the database connections and the end users do not usually maintain any connections directly. It is possible to support 1000 users with 20 or so PSAPPSRV processes. So you only need 20 database connections anyway. Please click on the [link here for a schematic representation of connection concentrator](#).

While trying to do a workload balancing, there is a lot of heuristic processing involved. It does not instantaneously react by switching the connections for each variation in the host workload. With connection concentrator enabled, one can achieve a high degree of workload balancing by properly adjusting the recycle count of the PSAPPSRV process. When you lower the recycle count, you risk losing the benefits of 'in memory caching' for the PSAPPSRV process. PSAPPSRV has to read the contents from the disk cache or from the database itself. To guard against additional database calls, make sure that the objects are staged and covered in the buffer pool. Additionally, you can move the application server cache file structures to a device with more controller cache, to improve the I/O response time.

You can implement the server cache mechanism, a delivered process within the PeopleSoft architecture or if you know your frequently used PeopleTools objects such as panels, records and other associated metadata, you can put them in a project and validate using three-tier mode. This will generate the cache files that are just necessary for the application.

Connection concentrator does not perform connection balancing and all it does is to favor workload distribution in a sysplex.

### **Restrictions**

Connection concentrator comes with certain restrictions. It may not be suitable for SQR and other batch programs. These threads are more than likely to get a dedicated worker agent.

You don't need to activate connection concentrator if you don't have data sharing for DB2. You can call Venkat srinivasan at 866-DB2-PSADMIN for further information if you are planning to consider connection concentrator. He can also be reached at [venkat@hewittandlarsen.com](mailto:venkat@hewittandlarsen.com).