

## **DB2 Checkpoint, Why should you adjust this parameter?**

When a checkpoint happens, DB2 has to gather and write the following information to the log dataset.

- Begin Checkpoint.
- Each incomplete Unit of Recovery status will be recorded in the log. This is commonly known as a Begin\_UR record.
- A set of object (page set) summary records representing the status of the object.
- An end checkpoint record.
- The checkpoint RBA is written to the BSDS as well.
- It schedules writes for the objects that have updated pages.

### **Purpose of checkpoint**

The sole purpose of the checkpoint is to make a system wide consistency point so a restart can be painless. When DB2 restarts, one of the important phases is the current status rebuild during which the status of the objects as existed when DB2 came down either normally or abnormally is automatically regenerated. It reads the BSDS to determine the last complete checkpoint. Depending on the DB2 start parameter (which is either RESTART or DEFER) it has to process the active log dataset. Starting from where the last checkpoint was found, it reads the log forward and finds all the begin UR records that don't have an associated end UR record. At the end of this task, it has determined the status of all open objects.

### **How can it be disruptive for PeopleSoft?**

At checkpoint time, DB2 will externalize all those pages for which there is a WRITE INTENT. Checkpoint is just a regular system event, which apart from externalizing the pages the checkpoint process updates the BSDS with the Checkpoint RBA, denoting a system wide consistent point used in restart and recovery. As many believe, there is nothing that stops in DB2 during the checkpoint. Checkpoints can be disruptive if many pages are to be written asynchronously. That is why it is necessary to control the writes with the VDWQT & DWQT thresholds. I have written a separate document about those thresholds.

There is a very important phenomenon which can potentially be disastrous for a typical Peoplesoft batch workload. In a batch process, PeopleSoft processes often read a subset of records from a master table into a temporary table and then subsequently a read is done on that subset of records in the temp tables. If a checkpoint is being done too frequently, say at 50000 log records, the number of pages waiting to be written to the disk could be quite high. So during the subsequent read as described above, it becomes necessary to remove that page from the checkpoint queue, satisfy the read request so that it doesn't wait and then later when the RELPAGE is issued that page will be written synchronously. During the whole process a significant time is accounted under head 'Asynchronous write wait' to the process. It is very noticeable especially if the page is actually in the process of being written.

Therefore it is very important to control the checkpoint interval. Ideally you should checkpoint every 10 – 15 minutes. 50000 may very well be a good value, if the update rate is too low. It is very uncommon in a Peoplesoft environment, though. A longer checkpoint will potentially delay the DB2 restart. Depending on your update rate it may elongate the time taken for RESTART.

Writing data more frequently will also help the NVS in the storage controllers. It can help reduce the restart time of the sub-system following an abnormal termination. During restart DB2 does not have to wait for the status rebuild and write processing for the pending pages.