The Firebird Cache

Everything in the database is organised as fixed sized pages of known structure - there are nine different types of pages. The page cache is the intermediate between the "working" parts of the database and the disk.

When initially started, Firebird reads the database header page, then the first pointer page of a system table called RDB$PAGES from which it then knows where to find the pointer pages, amongst other things, for the system and user tables within the database.

As soon as the *application* accesses the database and begins a transaction, Firebird reads the pointer pages telling it where to find the data pages for the tables involved in the application's transactions. All the pages go into the cache and stay there until the cache is completely full. When there's no place to put the next page, Firebird releases the *least recently used page* - not the first one read, but the one that has been referenced least recently and has not been modified.

When a transaction commits - and at a few other times - Firebird writes the pages changed by that transaction to disk, but does not release them from the cache - this helps avoid an extra read if the page is required again soon.

Over time and with luck, you end up with the most frequently changed and referenced pages resident in the cache - those would be transaction inventory pages, pointer pages for active tables, the header page, the top levels of indexes, etc. Data and lower level index pages swap in and out as required - but the cache will always be full.

You can use the MON$ tables (in Firebird 2.1 onwards, especially MON$IO_STATS) to figure out how well your cache is being used. They will report on the number of *Fetches* vs. *Reads* which is the number of times pages were accessed versus the number of times they had to be read from disk. When the number of reads goes up dramatically, you've reduced the cache too much.

The MON$ tables will also give you the number of *Marks* vs. *Writes*, which tells you the number of times pages are changed versus the number of times the pages were written to disk. When you see the disk writes go up, you've probably reduced the cache too much.