

DS6800

**Memory
And
DATA
Flow**

Explanation

Bob L Xiao 蒂斐

Storage Product Field Engineer

Email: xiaoling@cn.ibm.com

[Tel:86-21-60924242](tel:86-21-60924242) Teline: 24242

9F, Building 10, No.399 Ke Yuan Road, Zhang Jiang High Tech
Park, Shanghai, China (201203)

From IBM Storage Product Field Engineer Bob.

Per your issue, I will try to demonstrate you an overall picture of the cache in DS6800. Hope it will be helpful to resolve your issue.

There are 2 Processor Cards in the Server Enclosure of DS6800 and each one is redundant for another in functionalities. For each controller card, there is a 2 GB built-in memory which is acting as cache for DS6800. This 2 GB memory is divided into 2 parts as Volatile Memory and Persistent Memory (NVS). This kind of division is based on the redundant architecture of DS6800 as figure 1.

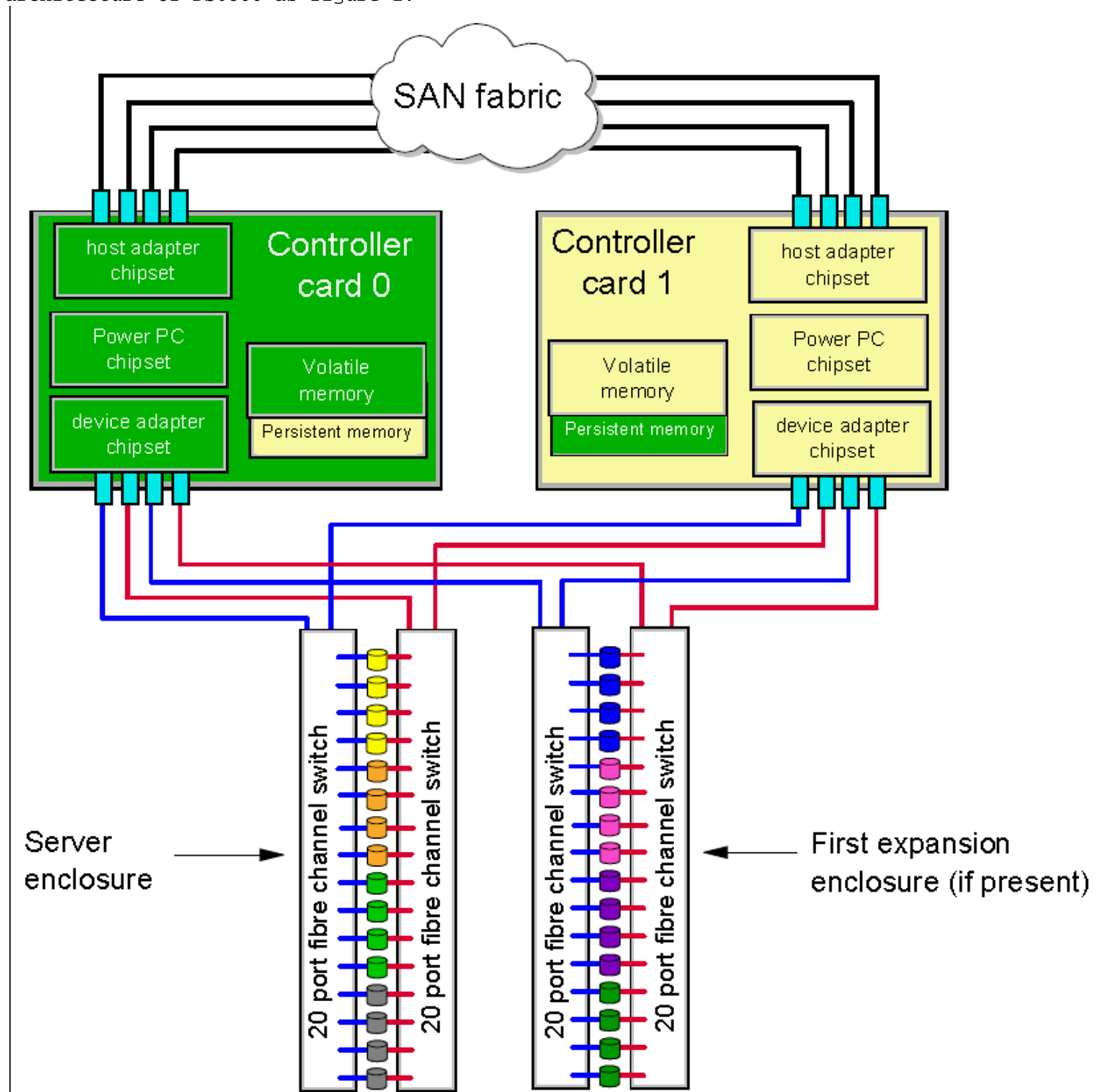


Figure 1 DS6800 Architecture

If a host which is connected to DS6800 is initializing a write, for a normal data flow, there will be 2 copies of each data inside of DS6800 in order to achieve this redundancy. Let's look into details. When a write is issued into a volume, the writes normally are issued to the controller that owns this volume (Controller 0 owns Even LSSs (logical Subsystems) and controller 1 owns Odd LSSs.) However, the write data is also placed into NVS (Persistent Memory) in the alternate controller as depicted as figure 2. The reason we need to keep 2 copies of each data is to achieve redundancy in case on controller gets into failed.

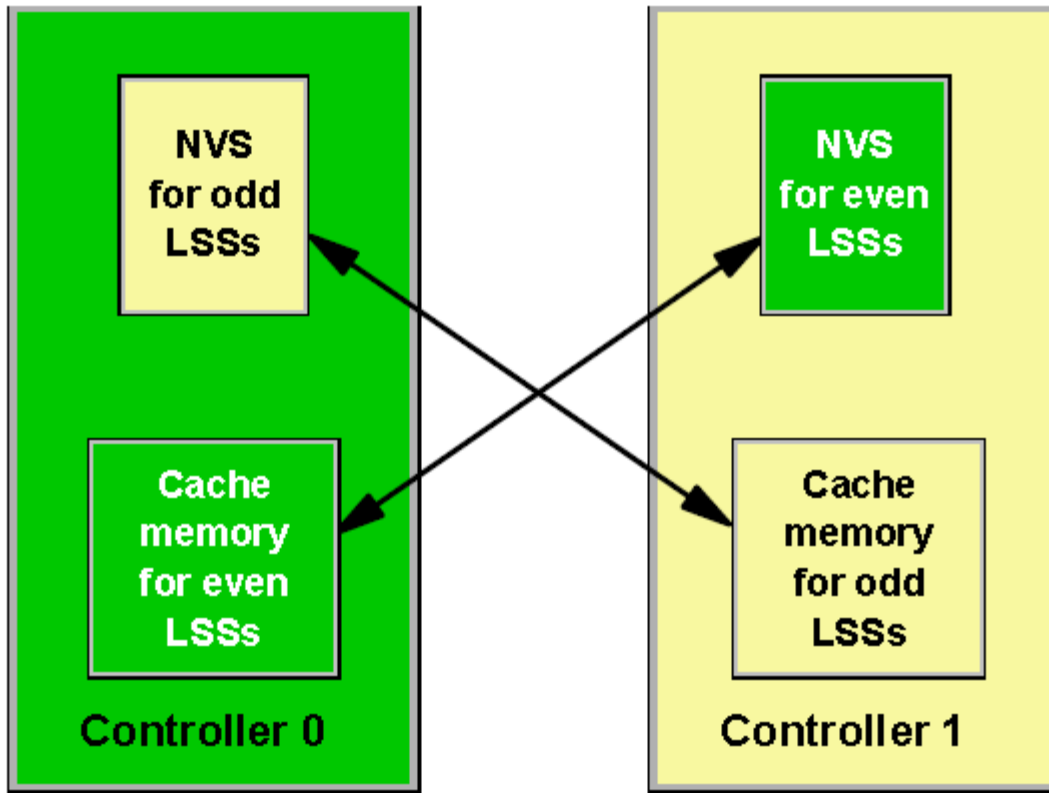


Figure 2 Normal Data Flow for DS6800

So the normal data flow of data for a write is as follows. Reading data of DS6800 is working in the similar way except that storage server is providing data to host.

1. Data is written to cache memory in the owning controller.
2. Data is written to NVS memory of the alternate controller.
3. The write is reported to the attached host as having been completed.
4. The write is de-staged from the cache memory to disk.
5. The write is then discarded from the NVS memory of the alternate controller.

This kind of redundancy is transparent to user of DS6800. The cache size you can see from the host side should be the size of "all cache" of DS6800. It is determined by the software/hardware design inside of DS6800. Usually, it equals the size of sum of volatile memory on both controller cards (They are the cache memory for even LSSs and odd LSSs in figure 2). Currently, there is no interface provided to user to adjust this size. However, you may encounter the scenario that the size changed to the half of the original if one controller gets failed. If one controller gets failed, DS6800 is working in **failover** mode as the following figure. During this time for our case, all LSSs will be taken over by controller 1. Moreover, the cache/NVS for Controller 1 will be split into 2 parts each for IO of even and odd LSSs. If the failed controller gets back online, then there will be **failback** process to let both controllers come back as normal status in figure 2.

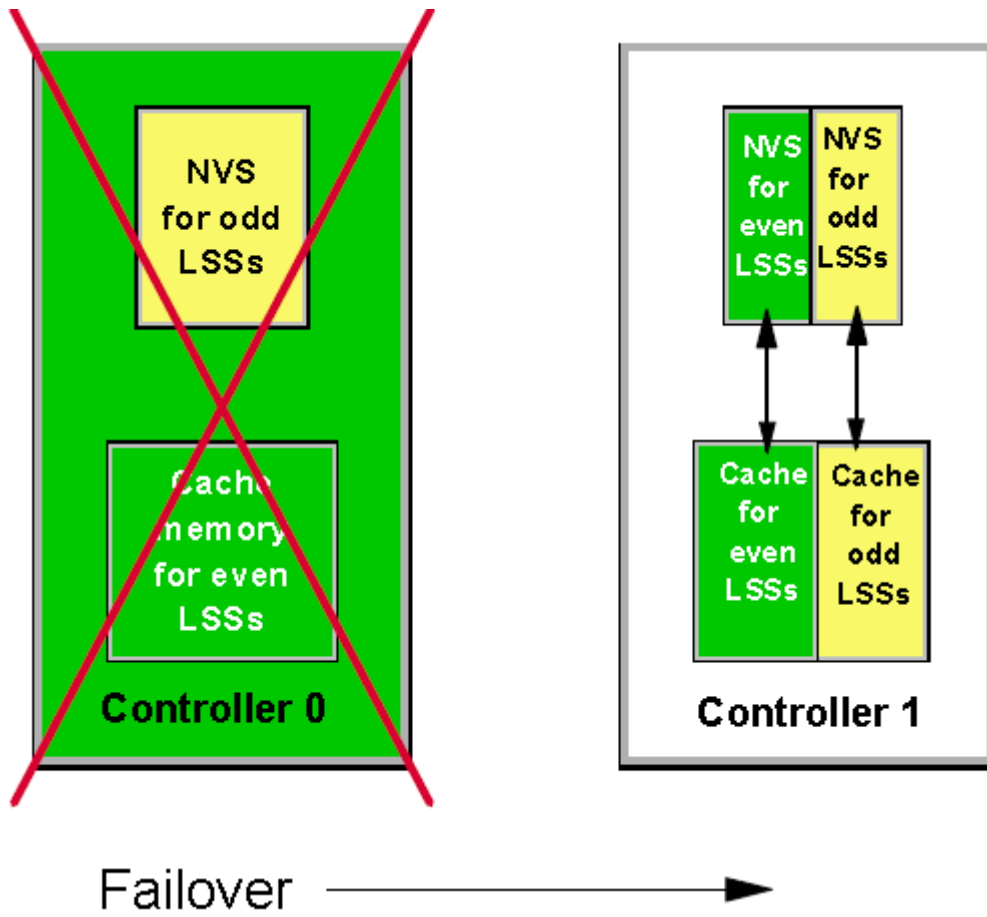


Figure 3 Failover of DS6800