3.7 Memory and Storage

Return to DIDO CLI Background

Memory and Computer Storage is not a single thing, it is a hierarchy based upon how much memory needs to be accessed and how fast it needs to be accessed. Figure 1 summarizes the hierarchy as a pyramid. the width of each band in the pyramid represent the amount of data that can be stored. As a general rule, if there is a lot of data, then to keep the costs affordable, the slower the access is to the memory. At the lowest level, the memory persists after the power is turned off. At the top of the pyramid, the data is lost when the power is turned off.

Designing for high performance requires considering the restrictions of the memory hierarchy, i.e. the size and capabilities of each component. Each of the various components can be viewed as part of a hierarchy of memories (m1,m2,...,mn) in which each member mi is typically smaller and faster than the next highest member mi+1 of the hierarchy. To limit waiting by higher levels, a lower level will respond by filling a buffer and then signaling for activating the transfer.

There are four major storage levels[1)]:

- **Internal** – Processor registers and cache
- **Main** – the system RAM and controller cards
- **On-line mass storage** – Secondary storage
- **Off-line bulk storage** – Tertiary and Off-line storage

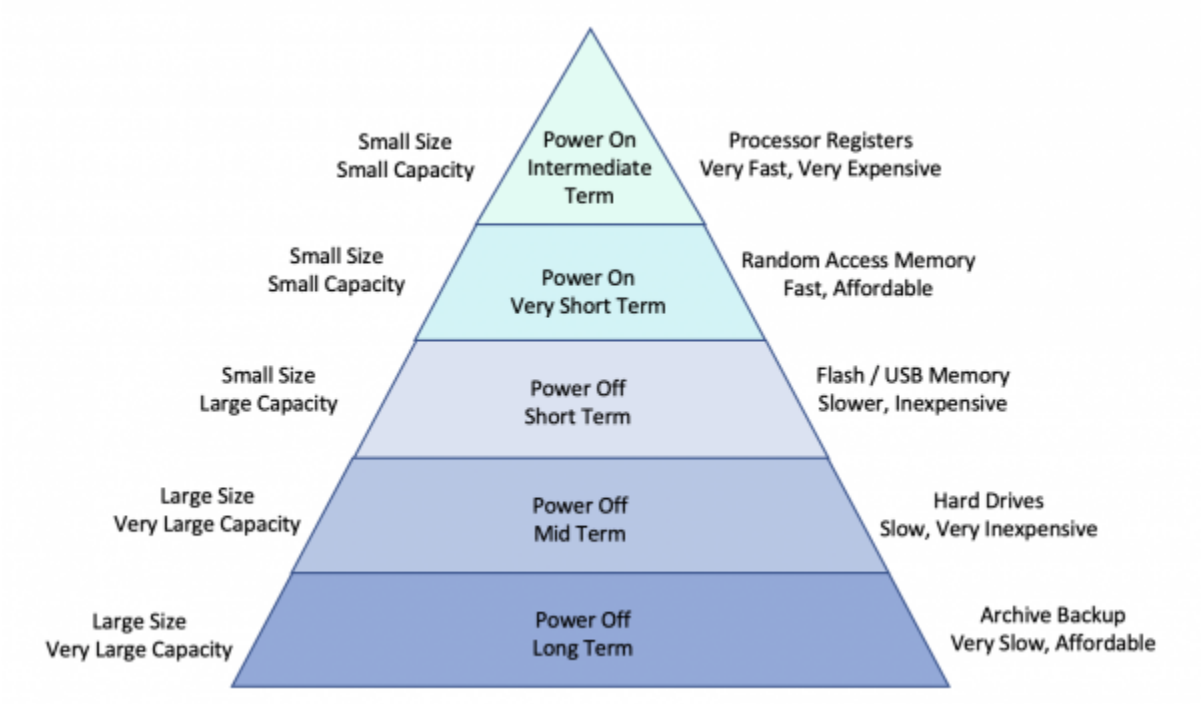

Figure 1: Traditional Computer Memory Hierarchy

This Traditional Computer Memory Hierarchy model is best when applied to a single computer. Within the DIDO, that applies to each of the nodes within the Distributed Network. However, the DIDO-CLI needs to consider a memory model that covers all the nodes and considers that each node must software

deterministically (i.e., all the software on all the nodes must produce the same output). Therefore, the Memory should rely extensively on memory that can persist after the power has been turned off. Although simple DIDO transactions routinely “carry” all the required data within the transaction, as the transactions get more complicated that may not always be possible or desirable since the size of the data that needs to be included in the transaction can become large.

DEFINE, CREATE, versus DECLARE

- DEFINE defines an DIDO Base Object in the Heap
- CREATE Creates and instance of something that has been defined
- DECLARE Creates a short term instance of something that has been defined

1)

Toy, Wing; Zee, Benjamin Computer Hardware/Software Architecture, January 1st, 1986, Prentice Hall. p. 30, ISBN 0-13-163502-6, https://isbndata.org/978-0-13-163502-9/computer-hardwaresoftware-architecture

From:
https://www.omgwiki.org/dido/ - **DIDO Wiki**

Permanent link:
https://www.omgwiki.org/dido/doku.php?id=dido:public:s_cli:05_contents:01_prt:03_langconst:07_memandstor:start

Last update: **2021/08/13 13:17**