## Chapter 2.3 Speed Up Access to Auxiliary Storage

***Introduction:***

* The average time that one disk needs to access the disk block is nearly 10 ms. This doesn’t mean that the application sends out the data requirement to disk controller and after 10ms it will get back data.
* If there is one disk, and the disk maybe busy because of the same process or the access of other process. In the worst situation, if the number that the disk access requirement exceeds 10ms each time, then these requirements will block forever.

However, we can do something to decrease the average access time of disk, so that we can improve the *handling capacity (吞吐量) (The disk access times that the system can adapt to.)*

We will consider some technologies that are used to speed up the classical database access:

1. Make the blocks that need to be access altogether into the same cylinder, so that we can usually avoid the Seek Time and Rotation Delay.
2. Divide the data that needs to be access into several smaller disks rather than one big disk. Let more Head Assembler go and access the disk block and that can increase the disk access amount in unit time.
3. “Mirror Image” disk - put two or more data copy into different disk. This strategy, other than save the data, can be used to access multi - disks in one time.
4. In Operation System, DBMS or Disk Controller, using the Disk Scheduling Algorithm to choose the sequence of block to be red or wrote.
5. Get the disks that need to be accessed to the main memory.

### Chapter 2.3.1 I/O Model of Calculation

***Introduction:***

Let us image one simple computer that runs DBMS, and try to give the service to the users of Database by using the Query and Modification. Now, assume that we have one processor, one disk controller and one disk, and the database is too big to be placed in the main memory.*The key part of database can be put into the main memory, but normally, if the user wants to access each segment of database, then it must be indexed from the disk.*

The dominant place of I/O cost: *The time that cost to execute read/write on the disk is much longer than those which are used to manipulate the data from main memory. Therefore, the times of block access is the approximation of algorithm, and it must be minimized.*

For example, if we have one Relation R with the requirement that we need to access the tuple, while the tuple has a fixed key k.

Now it’s better to create an index in this Relation R which can be used to locate the disk block with the tuple of key k and whether the index can tell us the specific place in block of this tuple is not so much important.

Chapter 2.3.2 Organize Data According to Cylinder

***Definition:***

Since the Seek Time occupies nearly half of the average access time, then put some data that needs to be visited together, such as Relation, stored into one single Cylinder or several close Cylinder, which is meaningful.

Actually, if we choose to read and write all blocks in one single disk or one Cylinder continuously, then here we just need to consider the *first Seek Time (Locate to the Cylinder)* and the *first Rotation Waiting Time (Wait to the first Block move under the Disk Head)*, but ignore all other time. *Then read/write the data from the disk will nearly equals to the Transfer Time theoretically.*

Chapter 2.3.3 Using Multi - Disk

Chapter 2.3.4 Disk Mirror

Chapter 2.3.5 Disk Schedule and Elevator Algorithm

Chapter 2.3.6 Pre - Fetch and Buffering