* **I/O parallelism** refers to reducing the time required to retrieve relations from disk by partitioning the relations over multiple disks.
* Horizontal partition - In **horizontal partitioning**, the tuples of a relation are divided (or declustered) among many disks, so that each tuple resides on one disk.
* Partition techniques
  + Round-robin. This strategy scans the relation in any order and sends then mod Diith tuple to disk number
* Access of data classified as
  + Scanning entire relation
  + Point queries – seek tuple that has specific value
  + Range queries - Locating all tuples for which the value of a given attribute lies within a

specified range

best suited for

* Round robin - The scheme is ideally suited for applications that wish to read

the entire relation sequentially for each query. Here point and range queries are complicated to process

* Hash partitioning – for point queries based on partition attribute but The scheme, however, is notwell suited for point queries on nonpartitioning attributes.
* Range partition - This scheme is well suited for point and range queries on the partitioning attribute.
* **What is skew -** When a relation is partitioned (by a technique other than round-robin), theremay be a **skew** in the distribution of tuples, with a high percentage of tuples placed in some partitions and fewer tuples in other partitions.
* Attribute value skew - All the tuples with the same value for the partitioning attribute end up in the same partition, resulting in skew.
* Partition skew - refers to the fact that there may be load imbalance in the partitioning, even when there is no attribute skew.
* **Small skew can result in significant decrease in performance**
* **Skew becomes an increasing problem with a higher degree of parallelism.**
* **Ways to minimize skew** 
  + **Reminaning**
* **Interquerry parallelism**
  + different queries or transactions execute in parallel with one another.
  + Transaction throughput can be increased by this form of parallelism.
  + Interquery parallelism is the easiest form of parallelism to suppor in a database system particularly in a shared-memory parallel system
  + **Cache coherency problem -** The problem of ensuring that the lates version of data in processor buffere pool when processor access or update from db system
  + **Protocol available to check cache coherency –** 
    - Shared disk system
      * Before any read or write access to a page, a transaction locks the page In shared or exclusivemode, as appropriate. Immediately after the transaction obtains either a shared or exclusive lock on a page, it also reads the most recent copy of the page from the shared disk.
      * Before a transaction releases an exclusive lock on a page, it flushes the page to the shared disk; then, it releases the lock
* Intraquery parallelism - execution of single query in paraller on multiple processor or disks
  + It speedup long running queries
  + **Intraoperation parallelism -** speed up processing of a query by parallelizing the execution of each individual operation, such as sort, select, project, and join
  + Interoperation parallelism. - We can speed up processing of a query by executing in parallel the different operations in a query expression