Lecture 13: Query Execution II
Lecture 13 查询执行 II

I/O Parallelism I/O 并行

Using additional processes/threads to execute queries in parallel will not improve performance if the disk is always the main bottleneck. Thus, we need a way to split the database up across multiple storage devices.
如果磁盘是主要瓶颈，那么使用额外的进程和线程去执行并行的查询并不能提高系统整体的性能。因此，我们需要一个在多个存储设备上拆分数据库的办法。

Multi-Disk Parallelism 多磁盘并行化

Configure OS/hardware to store the DBMS's files across multiple storage devices. Can be done through storage appliances and RAID configuration. This is transparent to the DBMS. It cannot have workers operate on different devices because it is unaware of the underlying parallelism.
配置操作系统/硬件设备去跨越多个存储设备存储 DBMS 的文件。可以通过存储设备和 RAID 配置来完成。这对 DBMS 来说是透明的。它不能让 worker 在不同的设备上操作因为它并不了解底层的并行性。

File-based Partitioning 基于文件的分区

Some DBMSs allow you to specify the disk location of each individual database. The buffer pool manager maps a page to a disk location. This is also easy to do at the file-system level if the DBMS stores each database in a separate directory. However, the log file might be shared.
有些 DBMS 允许你指定每个数据库磁盘的位置，缓冲池将页面映射到磁盘。如果数据库管理系统将不同的数据库存储到不同的目录，那么在文件系统级别很容易做到这一点。但是，日志文件可能是共享的。

Logical Partitioning 逻辑分区

Split single logical table into disjoint physical segments that are stored/managed separately. Such partitioning is ideally transparent to the application. That is, the application should be able to access logical tables without caring how things are stored.
将单个逻辑表拆分成独立存储/管理不相交的物理段，这种分区对应用程序来说是透明的。也就是说应用程序去访问逻辑表可以不关心底层的存储形式。

Vertical Partitioning 垂直分区

• Store a table's attributes in a separate location (like a column store).
将表的属性分开存储到不同的位置（像列存储）
• Have to store tuple information to reconstruct the original record.
必须存储元信息以重建原始记录。

Horizontal Partitioning 水平分区

• Divide the tuples of a table into disjoint segments based on some partitioning keys.
根据一些分区键将表的元组划分为不相交的段。
• There are different ways to decide how to partition (e.g., hash, range, or predicate partitioning). The efficacy of each approach depends on the queries.
有很多的方法来决定如何分区，(例如：哈希，范围，谓词分区)。每种方法的效果取决于具体的查询。