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

Process Models 处理模型

A DBMS process model defines how the system supports concurrent requests from a multi-user application/environment. The DBMS is comprised of more or more workers that are responsible for executing tasks on behalf of the client and returning the results.
DBMS 的处理模型定义了系统如何支持来自多用户应用程序/环境的并发请求。DBMS 由一个或者多个 worker 组成，它们负责执行任务并返回给客户端结果。

Approach #1 –Process per Worker: 每个 worker 由独立进程实现

• Each worker is a separate OS process, and thus relies on OS scheduler.
每个 worker 都是一个独立的操作系统进程，因此这会依赖操作系统的调度器。
• Use shared memory for global data structures.
对全局数据结构存储使用共享内存（可以用于多个 worker 进程之间通信)。
• A process crash does not take down entire system.
一个进程崩溃不会导致整个系统崩溃。

Approach #2 –Process Pool: 进程池

• A worker uses any process that is free in a pool.
worker 可以使用池中任何空闲的进程。
• Still relies on OS scheduler and shared memory.
仍然是依赖操作系统调度以及共享内存。
• This approach can be bad for CPU cache locality due to no guarantee of using the same process between queries.
由于无法保证在查询之间使用相同的进程，因此这种方法可能会对 CPU 的缓存 局部性 造成不利的影响。

Approach #3 –Thread per Worker: 每个 worker 由独立线程实现

• Single process with multiple worker threads.
有多个 worker 线程组成的独立进程实现。
• DBMS has to manage its own scheduling.
DBMS 需要管理它自己的调度策略。
• May or may not use a dispatcher thread.
可以选择使用或者不使用一个单独的调度线程。
• Although a thread crash (may) kill the entire system, we have to make sure that we write high-quality code to ensure that this does not happen.
一个线程崩溃（可能）会杀死整个系统，我们必须确保编写高质量的代码，以确保不会发生这种情况。
Using a multi-threaded architecture has advantages that there is less overhead per context switch and you do not have to manage shared model. The thread per worker model does not mean that you have intra-query parallelism.
使用多线程体系结构的优点是：每个上下文切换的开销较小，并且你不需要管理共享模型。每个 worker 一个线程的模型并不意味着你有 intra-query 的并行性。
For each query plan, the DBMS has to decide where, when, and how to execute:
对于每个查询计划，DBMS 必须决定在何处、何时以及如何执行它？
• How many tasks should it use?
它需要使用多少个任务？
• How many CPU cores should it use?
它需要使用多少 CPU 核心数？
• What CPU core should the tasks execute on?
任务应该在哪个 CPU 内核上执行？
• Where should a task store its output?
任务应该在哪里存储其输出？