Scheduling

我们先回顾下operator/task的概念

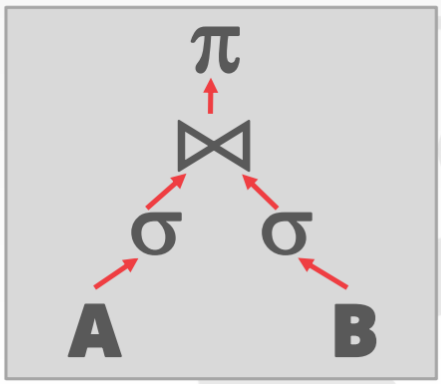
QUERY EXECUTION

A query plan is comprised of **operators**.

An **operator instance** is an invocation of an operator on some segment of data.

A **task** is the execution of a sequence of one or more operator instances (also sometimes referred to as a **pipeline**).

```
SELECT A.id, B.value
  FROM A, B
 WHERE A.id = B.id
   AND A.value < 99
   AND B.value > 100
```



我们今天讨论的调度：就是讨论task时怎么调度的。

SCHEDULING

For each query plan, the DBMS must decide where, when, and how to execute it.

→ How many tasks should it use?
→ How many CPU cores should it use?
→ What CPU core should the tasks execute on?
→ Where should a task store its output?

The DBMS ***always*** knows more than the OS.

Process Model

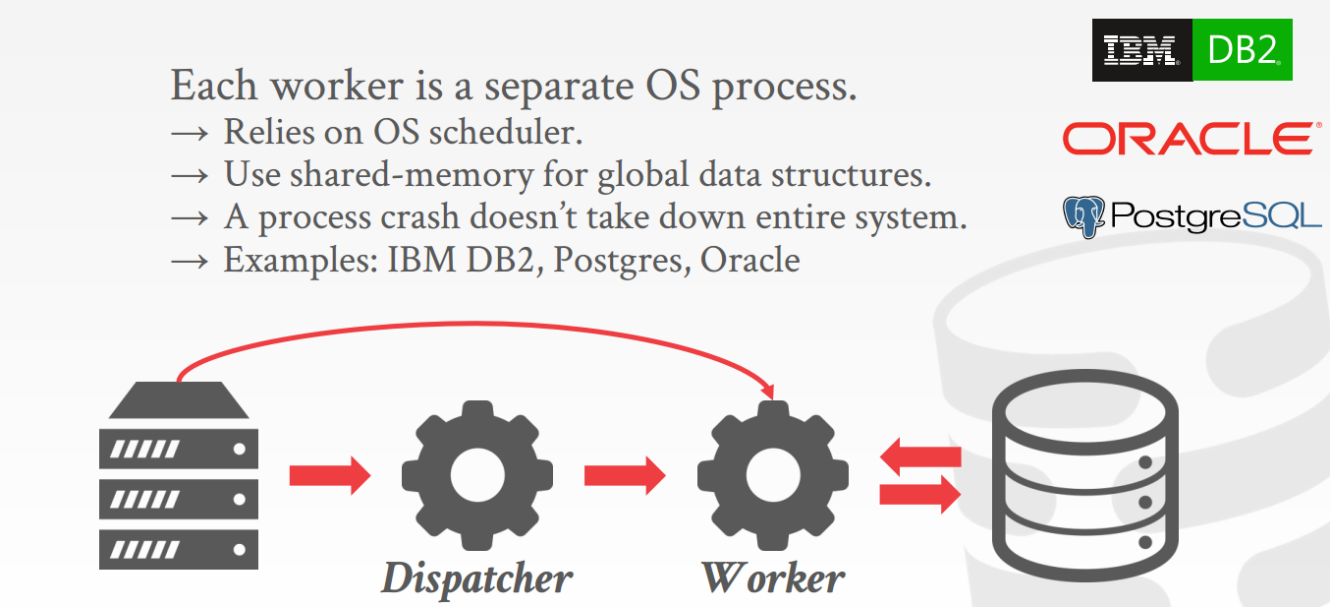
进程模型决定了系统如何支持多用户的并发请求

worker是DBMS中负责执行task并且向用户返回结果的组件

第一节课要求读的paper《Architecture of DB》有讲到三种进程模型：

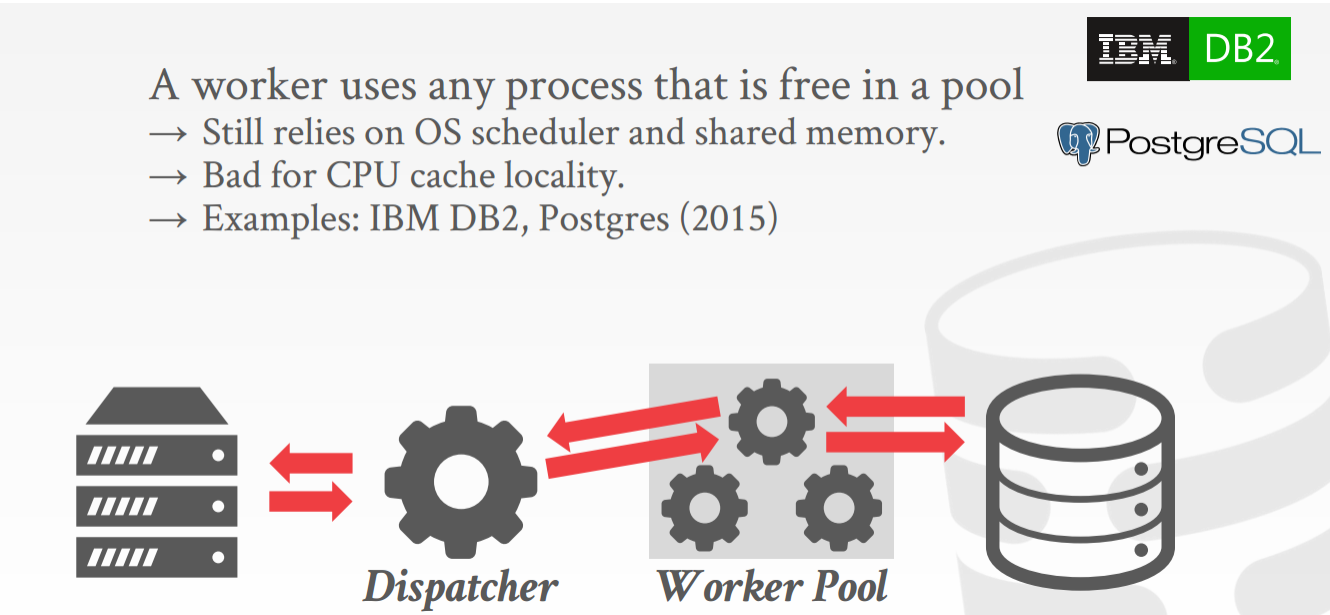
- Approach1 Process per DBMS worker

Fork to stat a new process,所以是操作系统来承担worker调度的工作



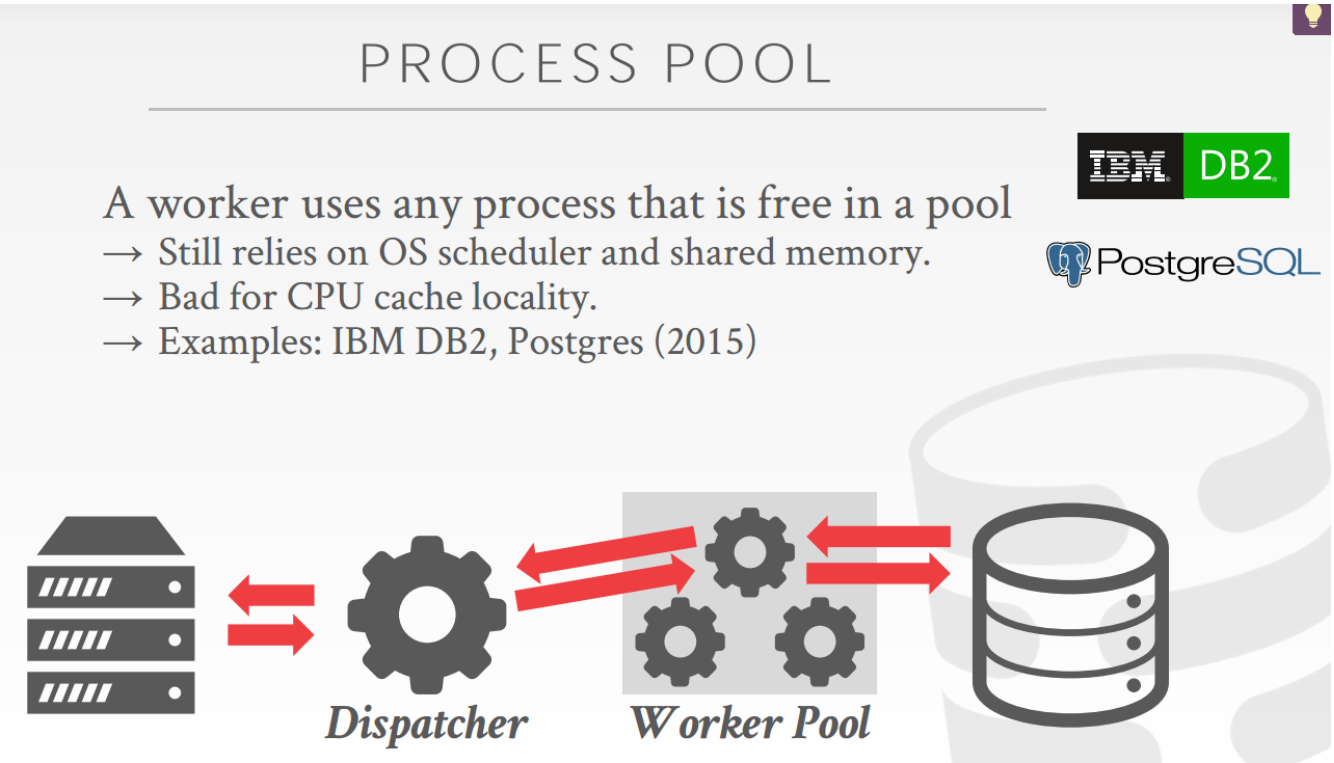
- Approach2 Process Pool

单个worker的socket可能要应付不同的并发读写请求，socket是有缓存的, 那就可能会导致维护缓存一致性导致的性能瓶颈,如cache miss等。



- Approach3 Thread per DBMS Worker

最常见的模型



好处：

- 上下文切换开销更少
- 不需要去管理共享内存

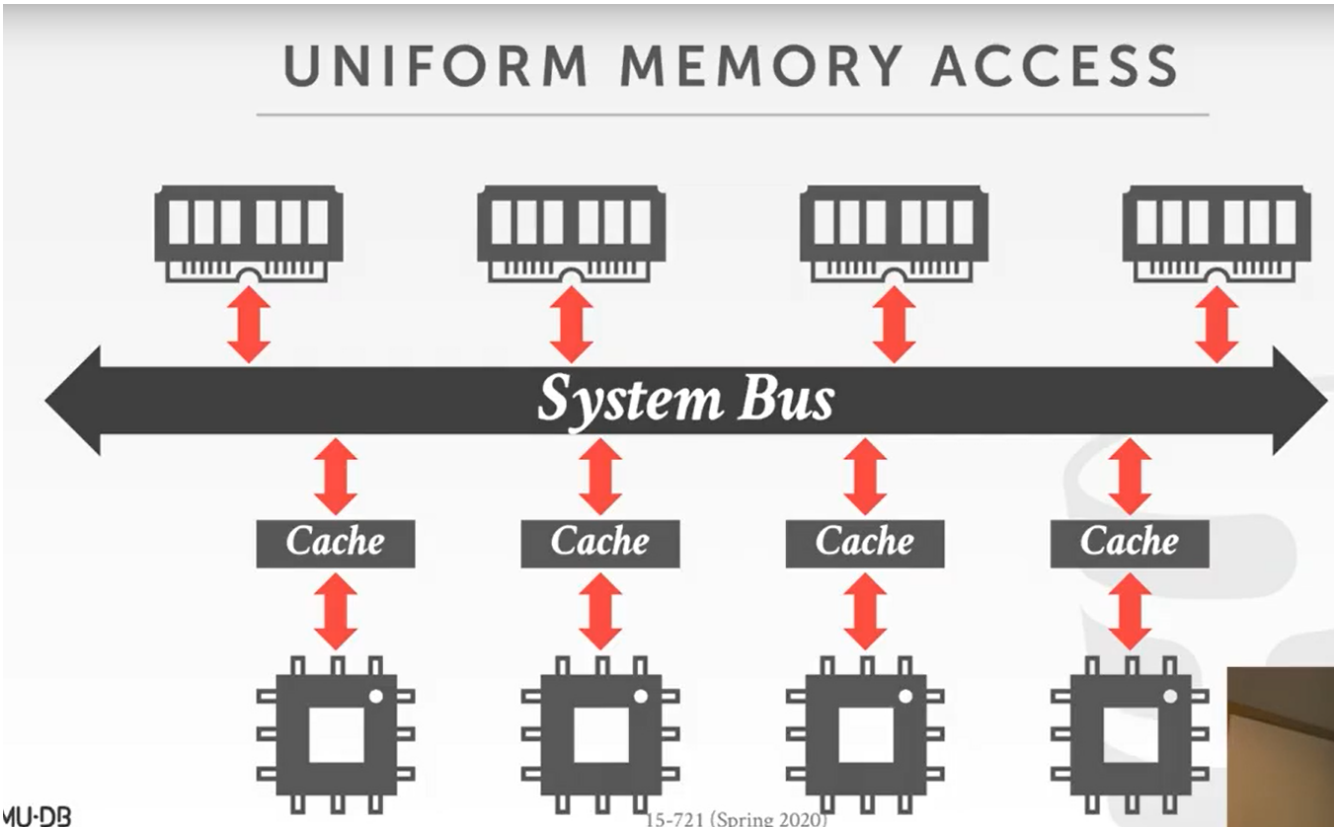
Data Placement

worker都是对local data做操作的，所以worker必须要意识到硬件的内存管理

SMP （Uniform Memory Access)

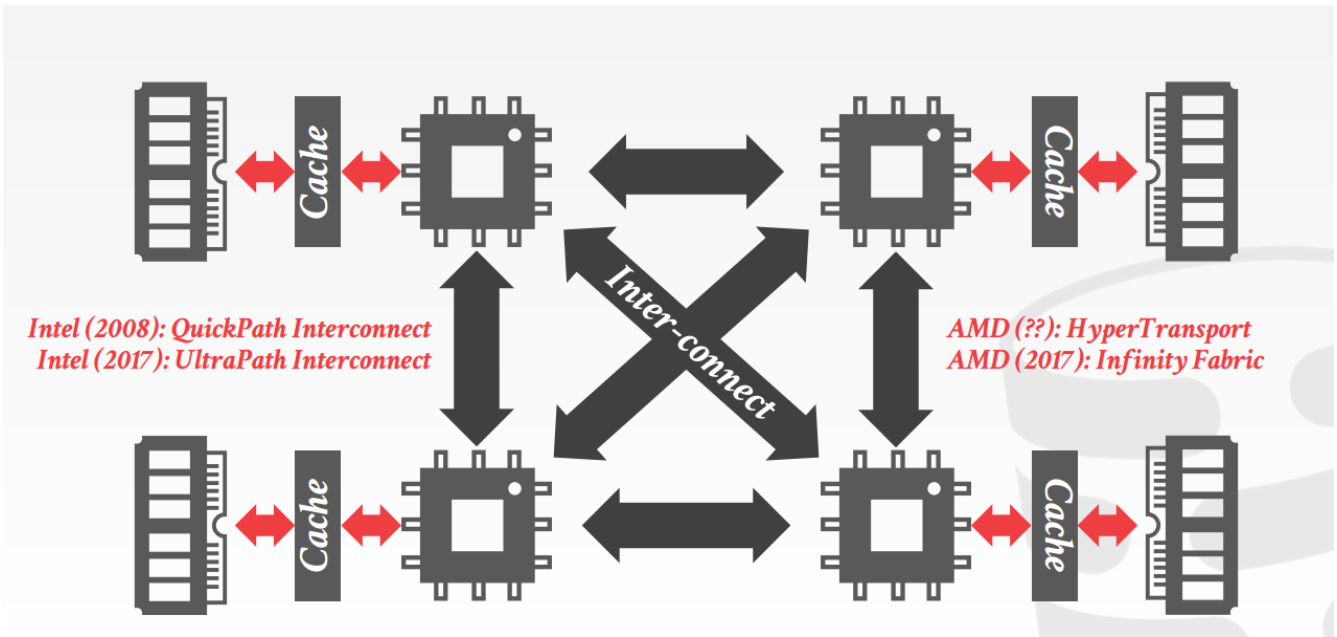
多个CPU共享内存地址空间

总线带宽是瓶颈

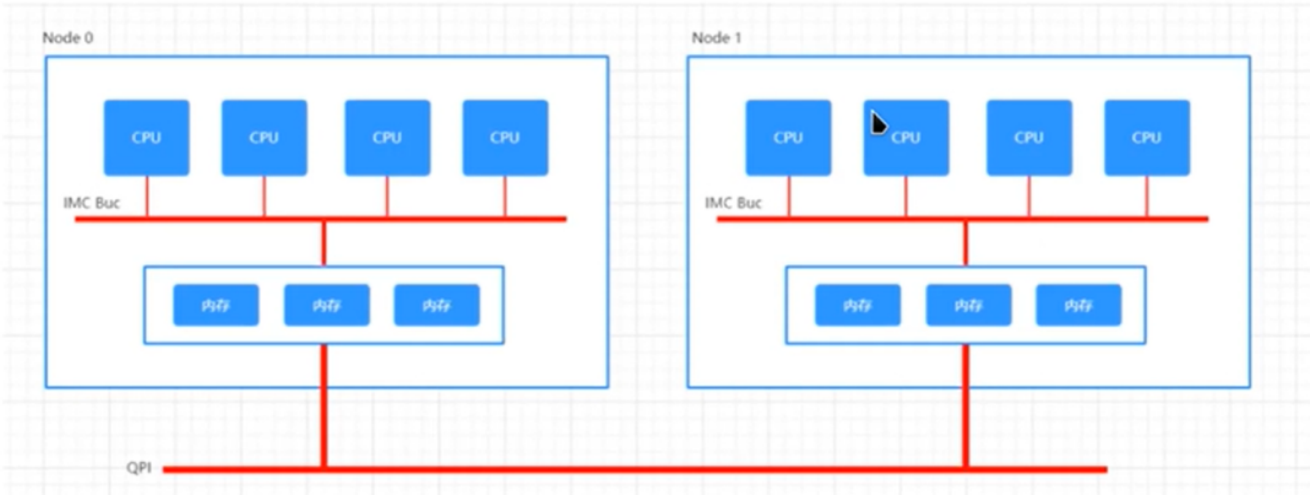


NUMA

避免对内存共享系统总线的占用

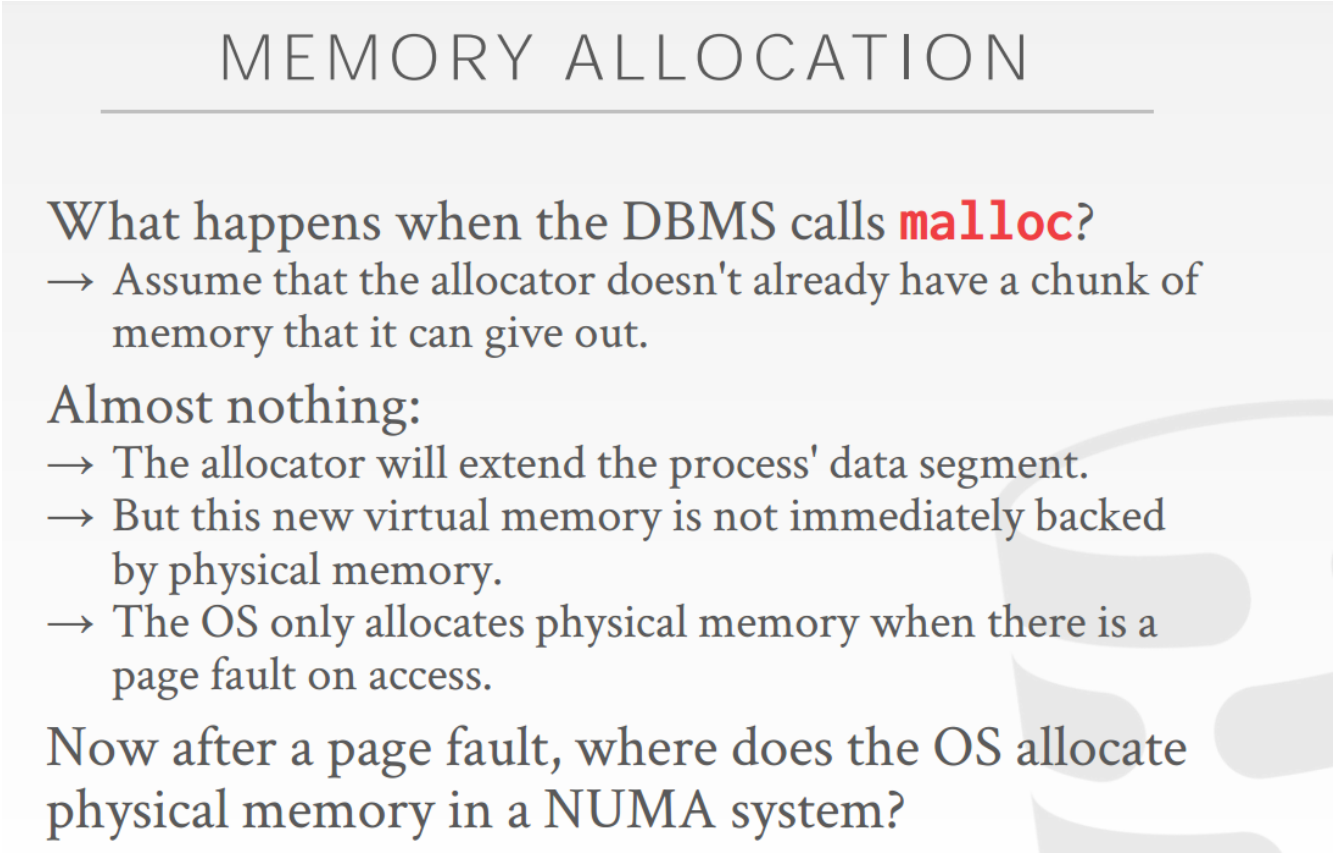


一个Node内不同的CPU的L1cache是独占的(通常分为数据cache和指令cache),而L2,L3Cache共享



DBMS应该是负责把memory做划分，并且分配给不同的CPU

然后把operator调度到距离其最近的CPU核上



内存分配的策略：

- Interleaving

内存分配平均分布在不同的CPU(socket)上。

- First-Touch

分配在引起Page Fault的那个线程对应的CPU(socket)上

Partition & Placement

前者指的是分区：

分区是把DBMS的数据按照某种策略水平划分出不同部分：

- 哈希
- Round-Robin
- Range

后者指的是怎么把这些数据放置到合适的地方

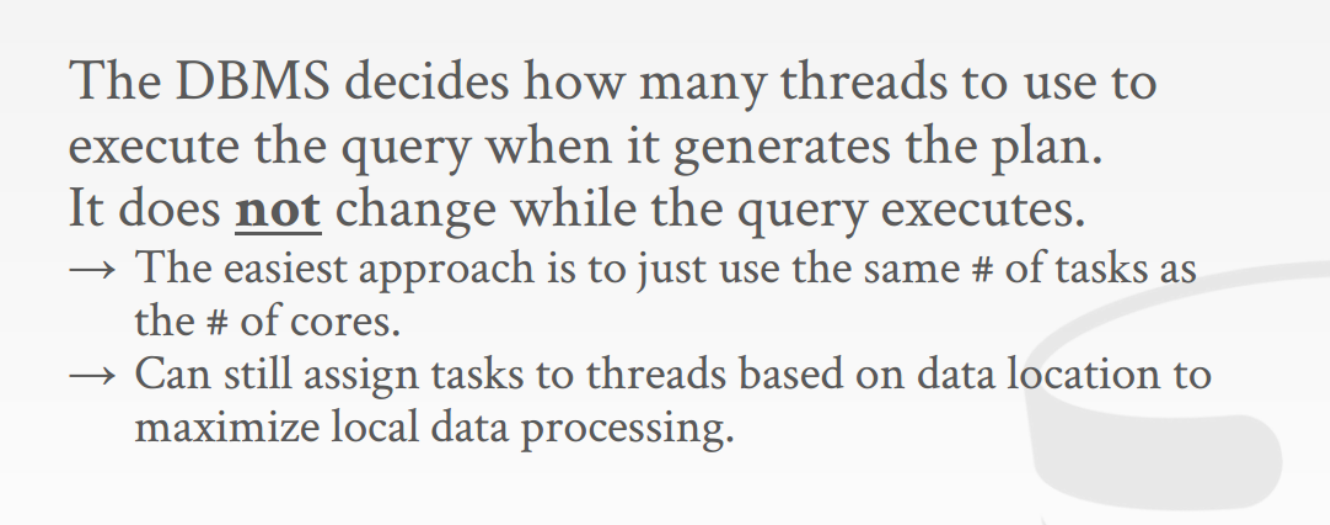
- Roung-Robin
- 均分(Interleaving)

现在我们来讨论如何把Logical Query Plan转化成实际运行的Task?

Scheduling

静态调度

在逻辑计划生成的时候固定好执行query所需要线程数量



动态调度

下面的模型是HyPer采取的一种pull模型

如果是dispatcher的话则为push模型，像这种由worker来去选择执行的task则为pull模型

No separate dispatcher thread.

The workers perform cooperative scheduling for each query plan using a single task queue.

→ Each worker tries to select tasks that will execute on morsels that are local to it.

→ If there are no local tasks, then the worker just pulls the next task from the global work queue.

而下面是SAP HANA选择的策略

Pull-based scheduling with multiple worker threads that are organized into groups (pools).

→ Each CPU can have multiple groups.

→ Each group has a soft and hard priority queue.

Uses a separate "watchdog" thread to check whether groups are saturated and can reassign tasks dynamically.

Dynamically adjust thread pinning based on whether a task is CPU or memory bound.

Found that work stealing was not as beneficial for systems with a larger number of sockets.

Using thread groups allows cores to execute other tasks instead of just only queries.

SQLOS(Used by SQL SERVER)

用户态的,NUMA-aware并且**运行在数据库内核的操作系统层**

决定task分配到什么线程

管理IO的调度与数据库的锁

FLOW CONTROL

如果请求到达速度远快于DBMS可以执行的速度，数据库很容易进入满载状态。