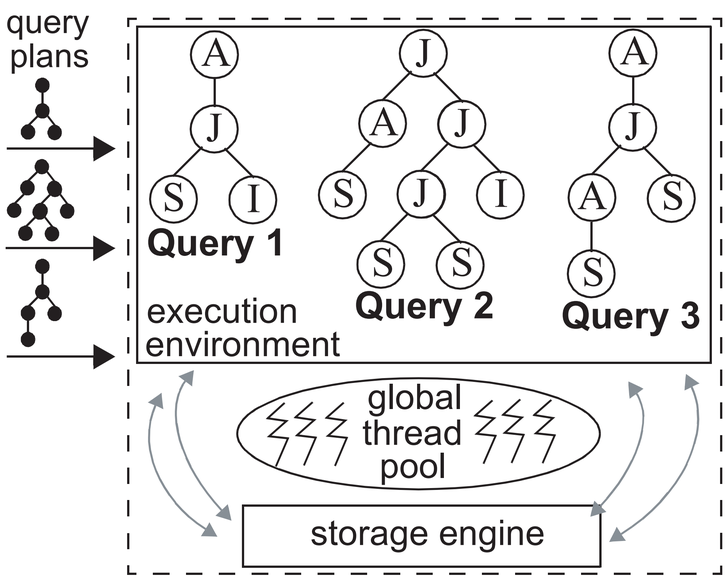
Simultaneous Pipelining in QPipe: Exploiting Work Sharing Opportunities Across Queries

Debabrata Dash, Kun Gao, Nikos Hardavellas, Stavros Harizopoulos, Ryan Johnson,
Naju Mancheril, Ippokratis Pandis, Vladislav Shkapenyuk, Anastassia Ailamaki

Conventional Query Engines

- **Conventional model: "one-query many-operators"**
- **Queries may exhibit data & computation overlap**
- **Run-time sharing only applies to the storage engine**
- **Storage engine sees uncoordinated page requests**

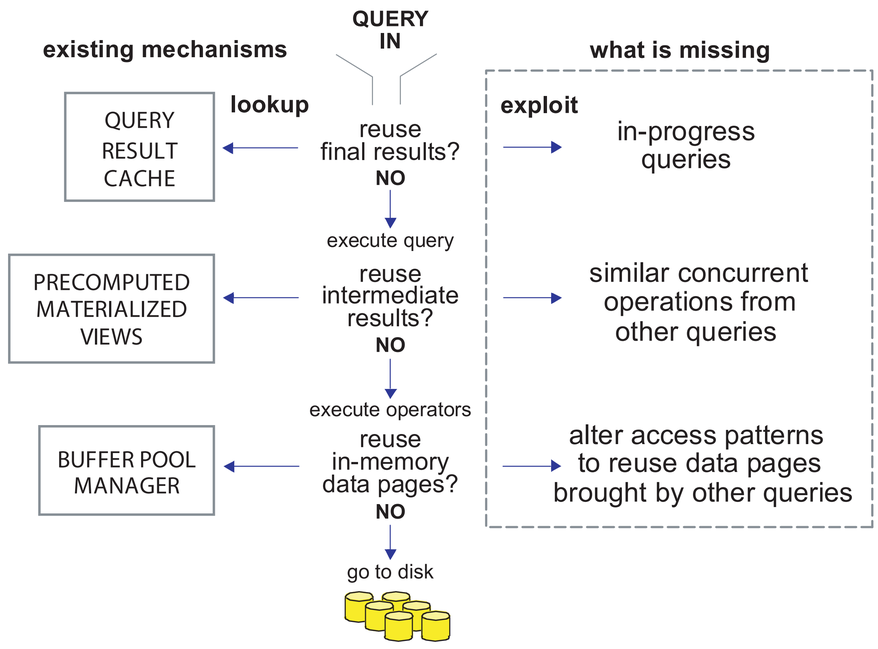


Data & Work Sharing Limitations

- **High concurrency increases sharing opportunity**

However:

- **Queries are evaluated independently**
- **Existing mechanisms for sharing are opportunistic**



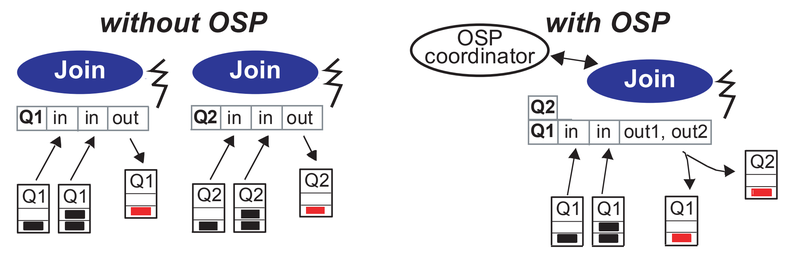
We need new execution model to expose work sharing

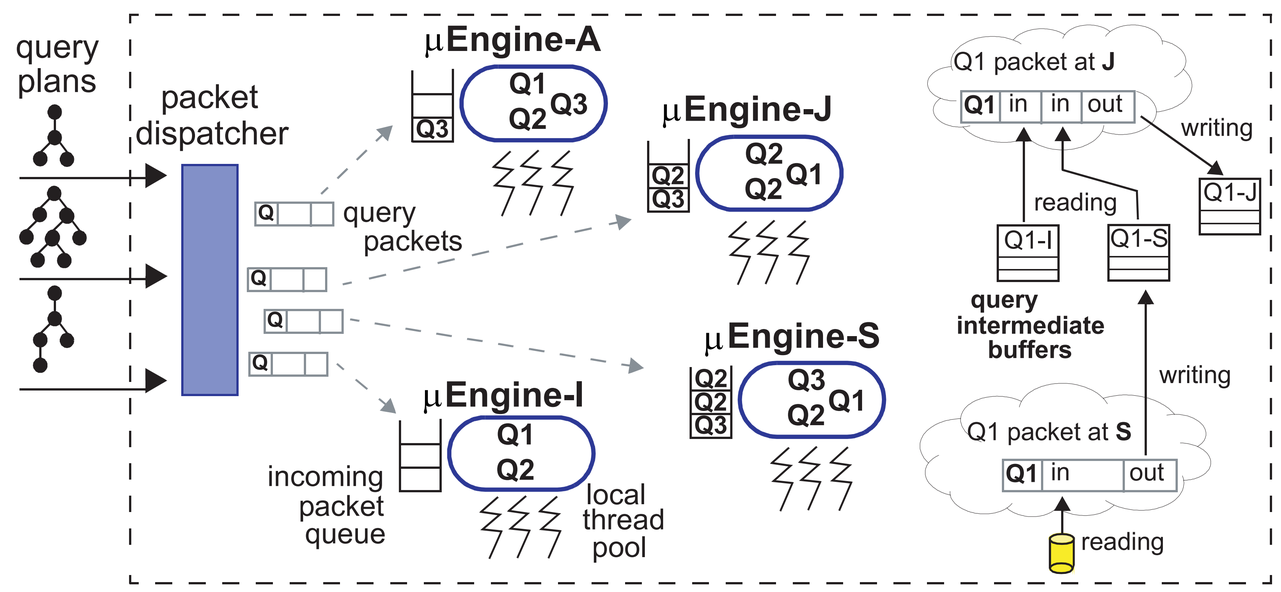
QPipe: a Staged Query Execution Engine

- **New philosophy: "one-operator many-queries"**
- **Relational operators become micro-Engines**
- **Queries break in packets, queue up in μEngines**
- **Exposes sharing opportunities at run time**

OSP: On-demand Simultaneous Pipelining

- **μEngines detect overlap at run time**
- **Results *simultaneously pipelined* to consumers**



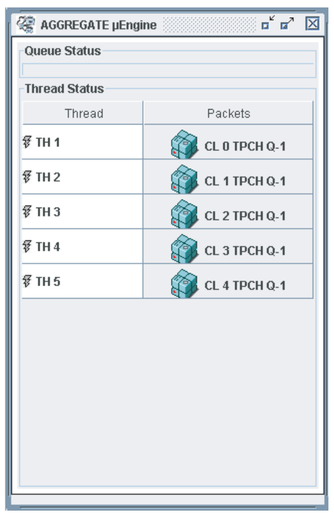


Demonstration

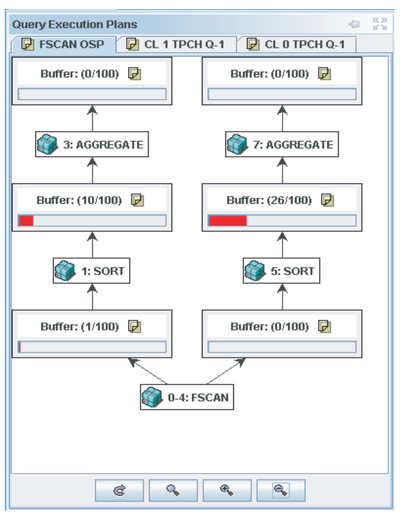
- **QPipe is built on top of BerkeleyDB**
- **Experiment with a subset of TPC-H queries**

- **Demonstration features:**
  - **Introduction to QPipe**
    - **Resource utilization and query progress**
  - **Demonstration of OSP**
    - **Simultaneously pipelined query execution**
  - **Interactive mode: submission of ad-hoc packets**

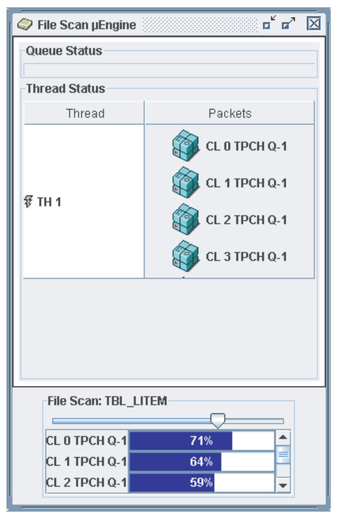
μEngine view



Query plan and intermediate buffers



μEngine OSP view



Databases
@ Carnegie Mellon

http://www.cs.cmu.edu/~StagedDB