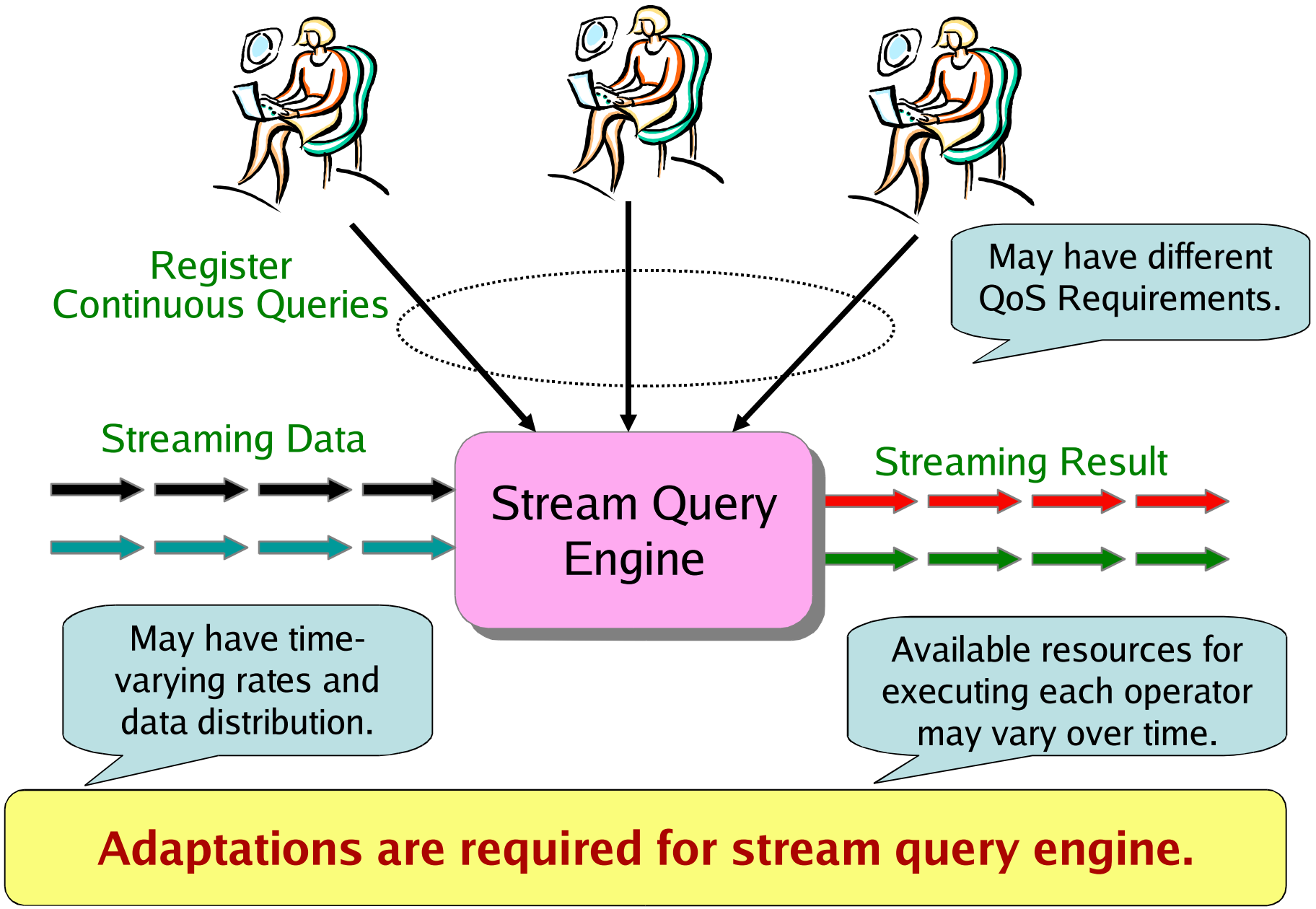
CAPE: Continuous Query Engine with Heterogeneous-Grained Adaptivity

Elke A. Rundensteiner, Luping Ding, Timothy Sutherland, Yali Zhu
Brad Pielech, Nishant Mehta
Natasha Bogdanova, Mariana Jbantova

Department of Computer Science, Worcester Polytechnic Institute
100 Institute Road, Worcester, MA 01609
Tel: 1-508-831-5857, Fax: 1-508-831-5776
{rundenst, lisading, tims, yaliz, winners, nishantm, natasha, jbantova}
@cs.wpi.edu

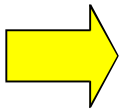
http://davis.wpi.edu/dsrg/CAPE/index.html

Uncertainties in Stream Query Processing



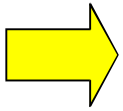
What is CAPE?

Constraint-aware



Exploit semantic constraints such as sliding windows and punctuations to reduce resource usage and improve response time.

Adaptive



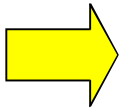
Incorporate ***heterogeneous-grained adaptivity*** at all query processing levels.

- **Adaptive query operator execution**
- **Adaptive query plan re-optimization**
- **Adaptive operator scheduling**
- **Adaptive query plan distribution**

Continuous Query

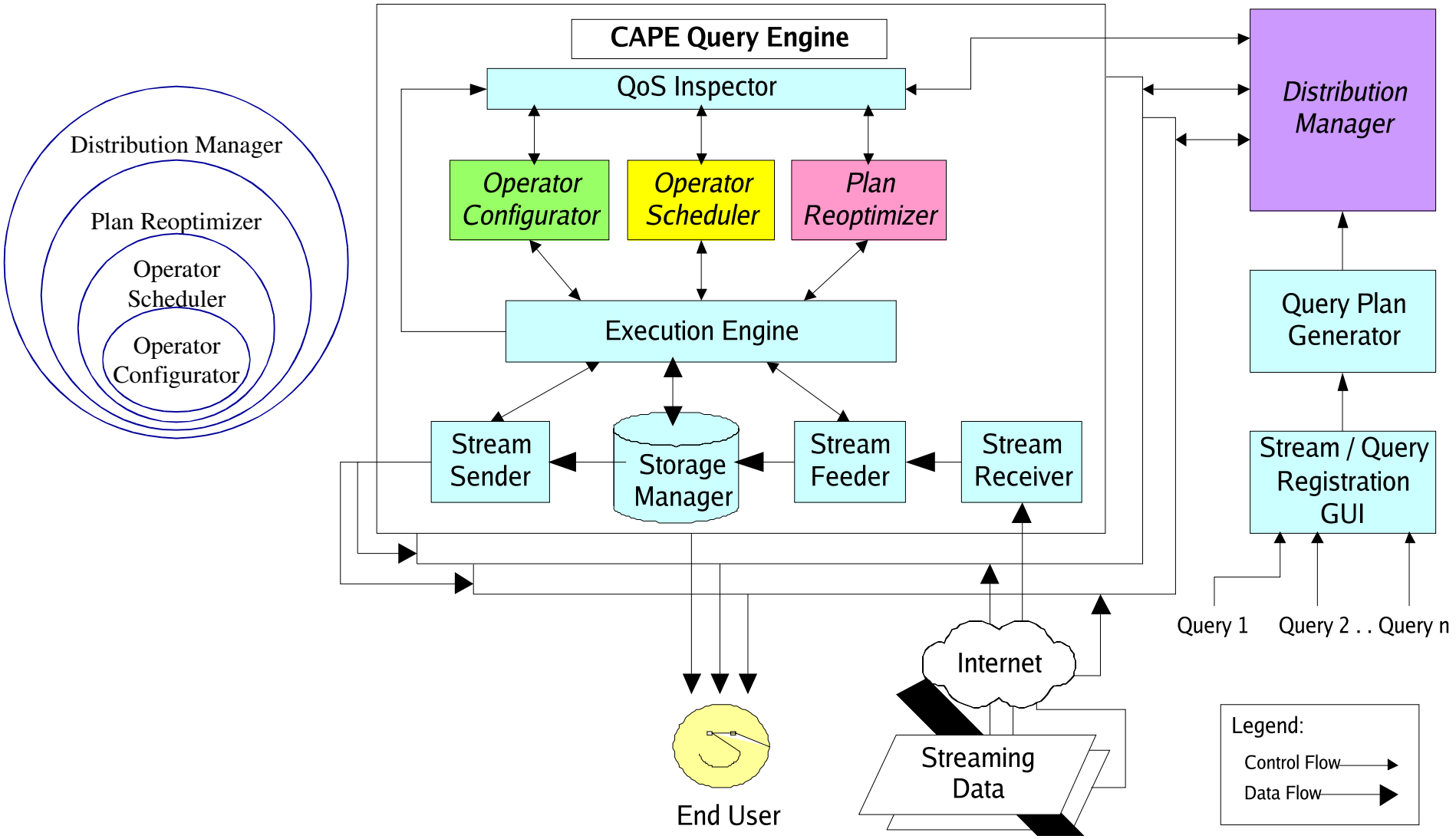
Processing

Engine



Process queries in a real-time manner by employing well-coordinated heterogeneous-grained adaptations.

CAPE System Architecture



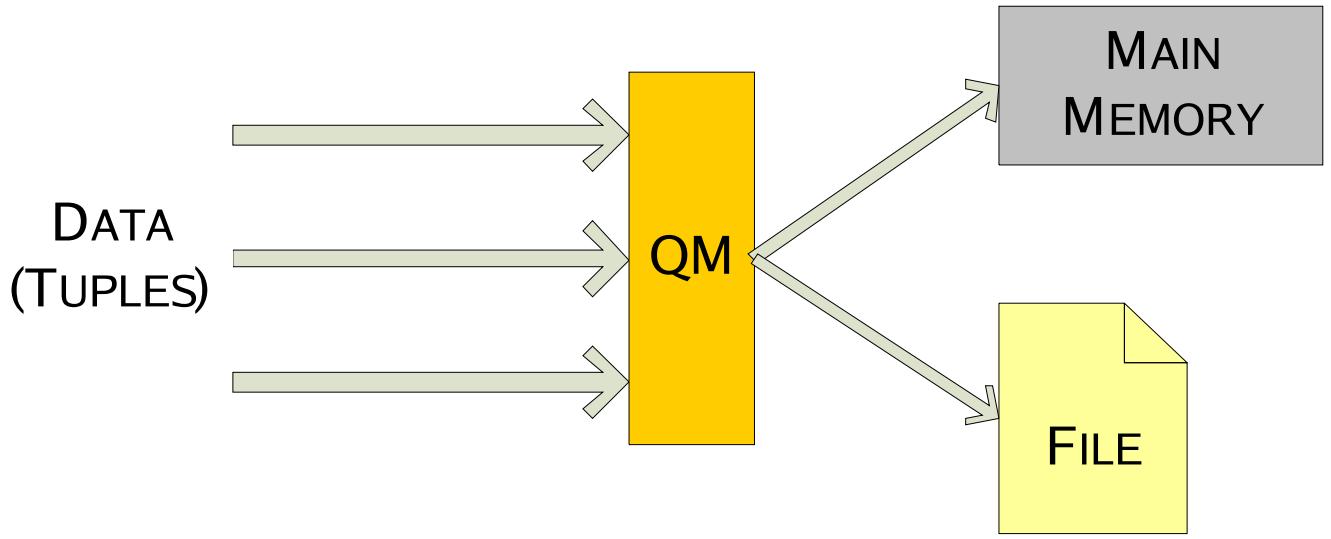
Queue Manager: Purpose

PROBLEM:

CAPE handles very large amounts of data, so need backup method when it runs out of memory

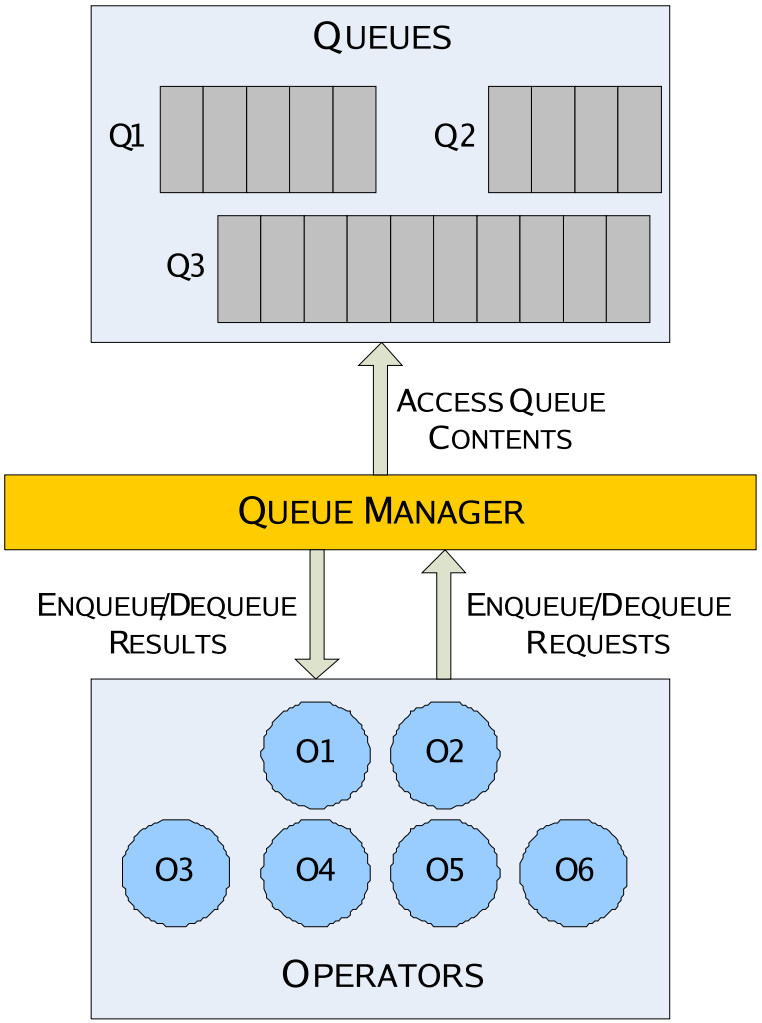
SOLUTION:

Queue Manager, which decides whether data in queue needs to go to file or remain in memory

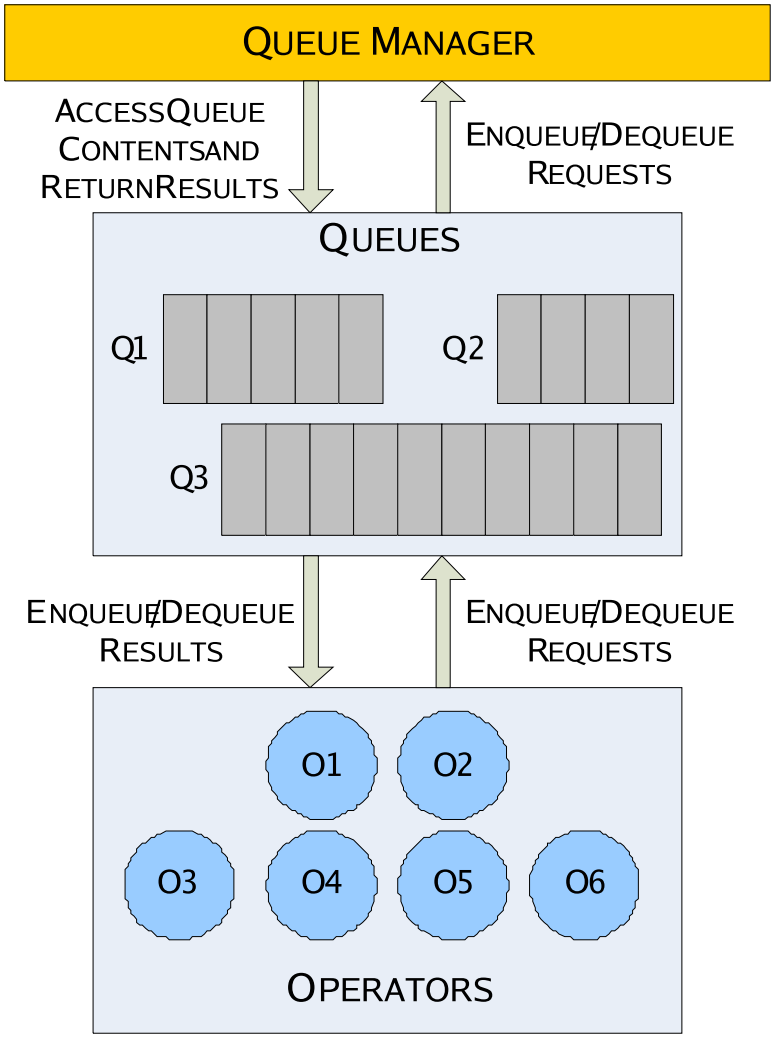


Queue Manager: Structure

Ideally:



Really:



Queue Manager: Decision Making

- Keep track of a memory threshold variable
  - How much memory we want to keep free
  - Once available memory goes below threshold, tuples are sent to disk
- Have an update method, which is called every time QM needs to make a decision
  - Ensures most recent memory info is used
- Use Storage Manager when tuples need to go to file to minimize I/O costs

Storage Manager: Overview

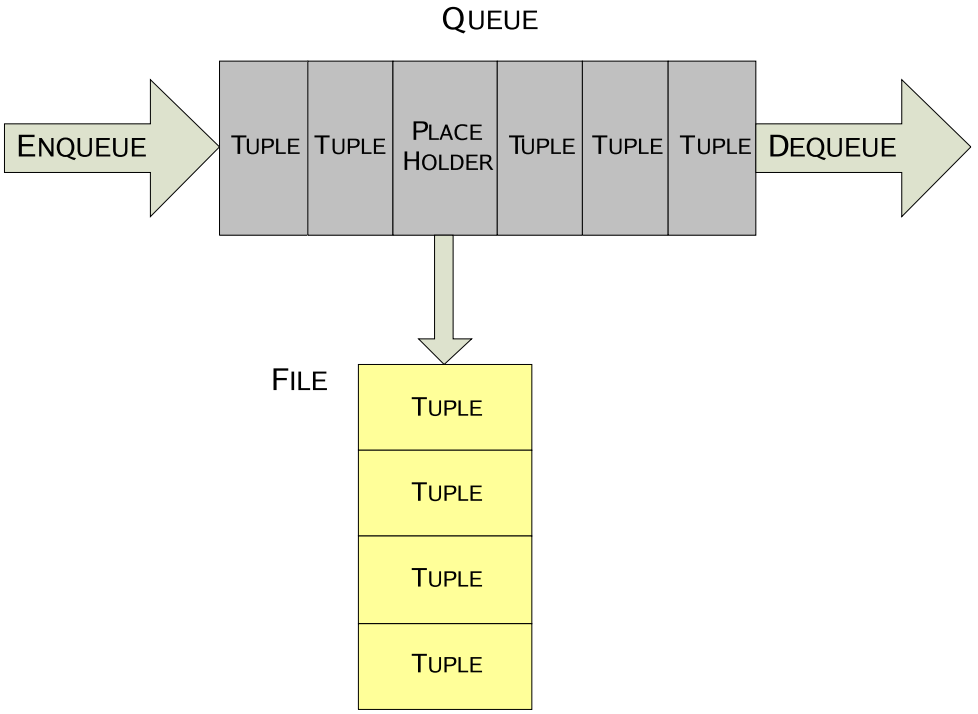
Storage Manager is called by QM when tuples need to be written to/ read from disk

(Adapted for CAPE from Nishant Mehta's Storage Manager)

- Parses tuples and generates symbol trees based on schema
  - Side Effect: Need a new instance of Storage Manager for every schema
- Provides an efficient way to read/write files
  - Implements random access for tuple files

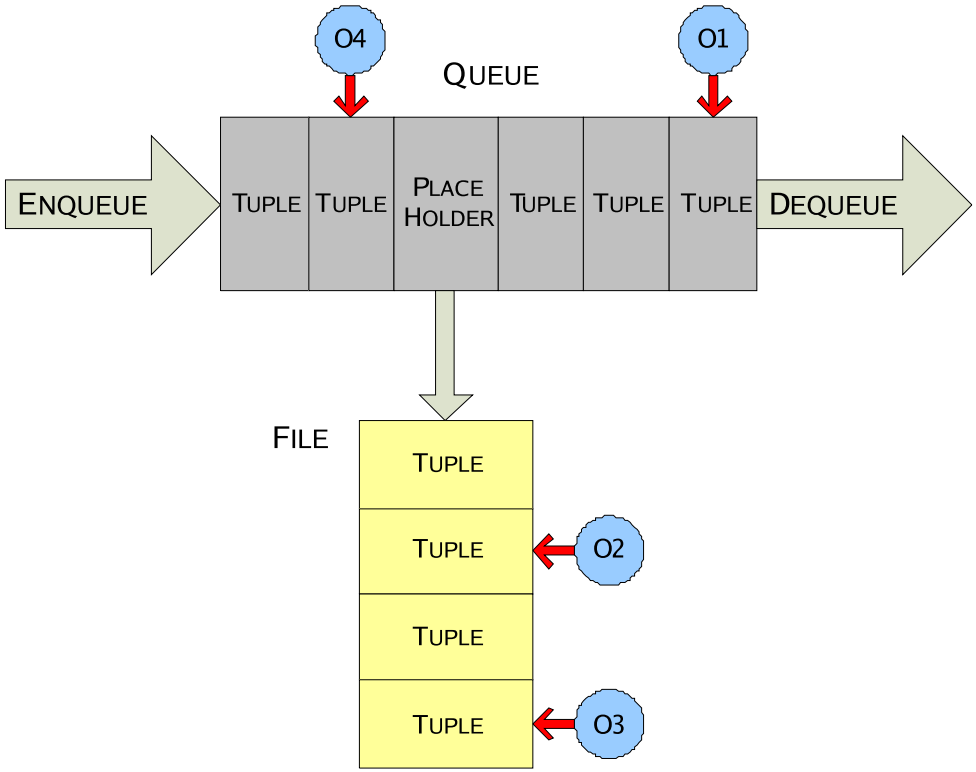
Queue Manager: Enqueue/Dequeue

- Tuples are stored in queue (main memory) until memory threshold is reached
- Then, tuples are written to file and a place holder is put in the queue
- Dequeue simply reads off the tuples from the front of the queue and from file if necessary



Queue Manager: Cursors

- Cursors allow multiple operators to access a queue at the same time
- If one operator reads from file, those tuples are put in main memory so other operators do not need to read from file again



CAPE Publications, TRs & URLs

[RDZ04] E. A. Rundensteiner, L. Ding, Y. Zhu, T. Sutherland and B. Pielech, "**CAPE: A Constraint-Aware Adaptive Stream Processing Engine**". Invited Book Chapter. http://www.cs.uno.edu/~nauman/streamBook/. July 2004.

[ZRH04] Y. Zhu, E. A. Rundensteiner and G. T. Heineman, "**Dynamic Plan Migration for Continuous Queries Over Data Streams**". SIGMOD 2004, pages 431-442.

[DMR+04] L. Ding, N. Mehta, E. A. Rundensteiner and G. T. Heineman, "**Joining Punctuated Streams**". EDBT 2004, pages 587-604.

[DR04] L. Ding and E. A. Rundensteiner, "**Evaluating Window Joins over Punctuated Streams**". CIKM 2004, to appear.

[DRH03] L. Ding, E. A. Rundensteiner and G. T. Heineman, "**MJoin: A Metadata-Aware Stream Join Operator**". DEBS 2003.

[SPR04] T. Sutherland, B. Pielech and E. A. Rundensteiner, "**Adaptive Scheduling Framework for A Continuous Query System**". Tech Report, WPI-CS-TR-04-16, 2004.

[SR04] T. Sutherland and E. A. Rundensteiner, "**D-CAPE: A Self-Tuning Continuous Query Plan Distribution Architecture**". Tech Report, WPI-CS-TR-04-18, 2004.

CAPE Project: http://davis.wpi.edu/dsrg/CAPE/index.html