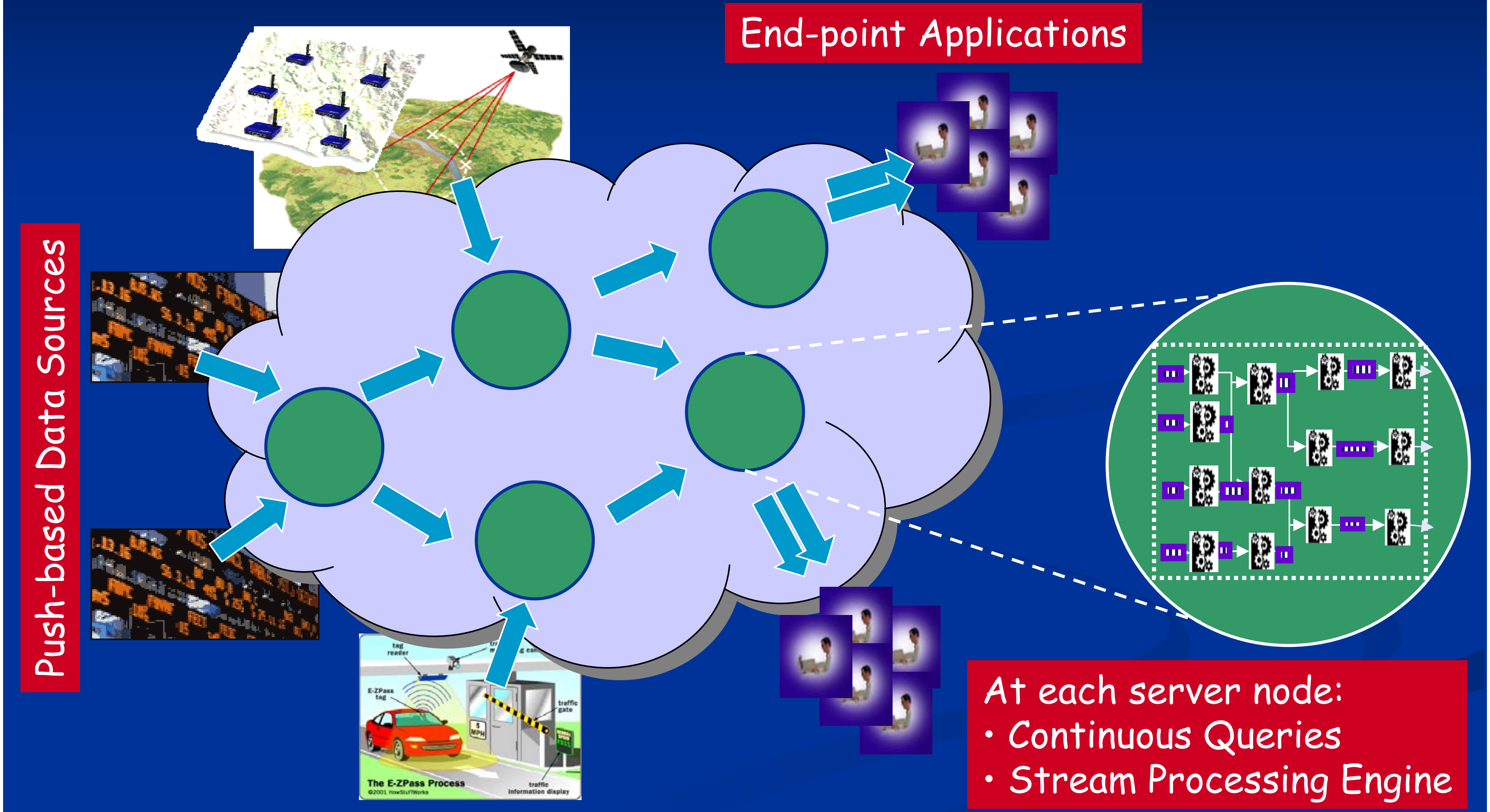
Dealing with Overload in Distributed Stream Processing Systems

Nesime Tatbul Stan Zdonik

BROWN

Distributed Stream Processing



End-point Applications

Push-based Data Sources

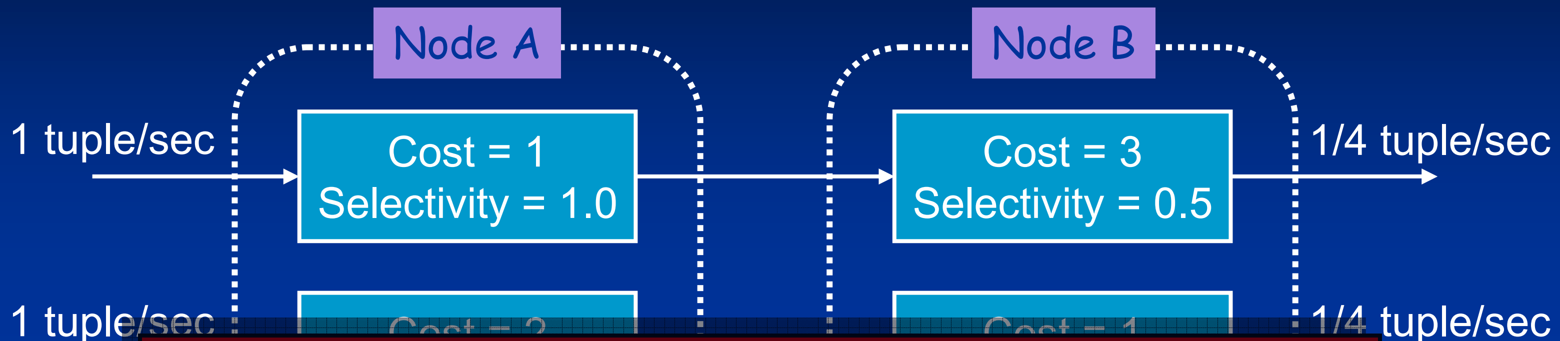
At each server node:
- Continuous Queries
- Stream Processing Engine

The Overload Problem

- Bursty data arrival
- Insufficient resources (e.g., CPU, bandwidth)
- Bottlenecks along the server chain
- Delayed query results

- Given a load distribution, how can we best **shed load** that minimizes degradation in result quality?

Load Dependency



Server nodes must coordinate in their load shedding decisions to achieve high-quality results.

Plan					nput
0					/4
1					/4
2	1, 0	1	1, 0	3	1/3, 0
3	0, 1/2	1	0, 1/2	1/2	0, 1/2
4	1/5, 2/5	1	1/5, 2/5	1	1/5, 2/5

optimal for A

feasible for both

optimal for both

≤ 1

≤ 1

maximize !

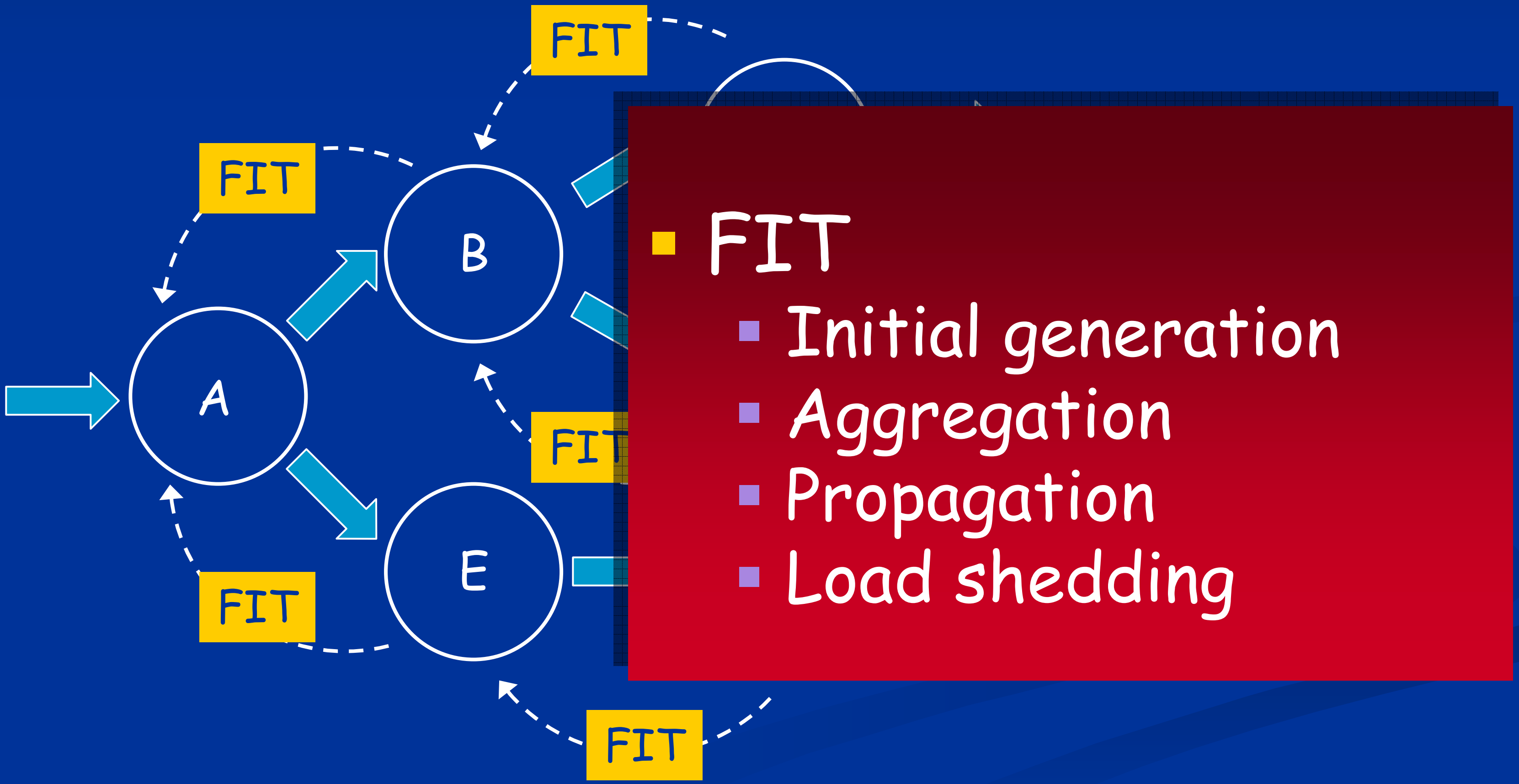
Design Goals

- Fast reactivity to load
- Global control on output quality
- Scalability
  - Number of server nodes
  - Number of input streams
  - Amount of query branching

Centralized or Distributed?

Distributed Coordination by Metadata Exchange

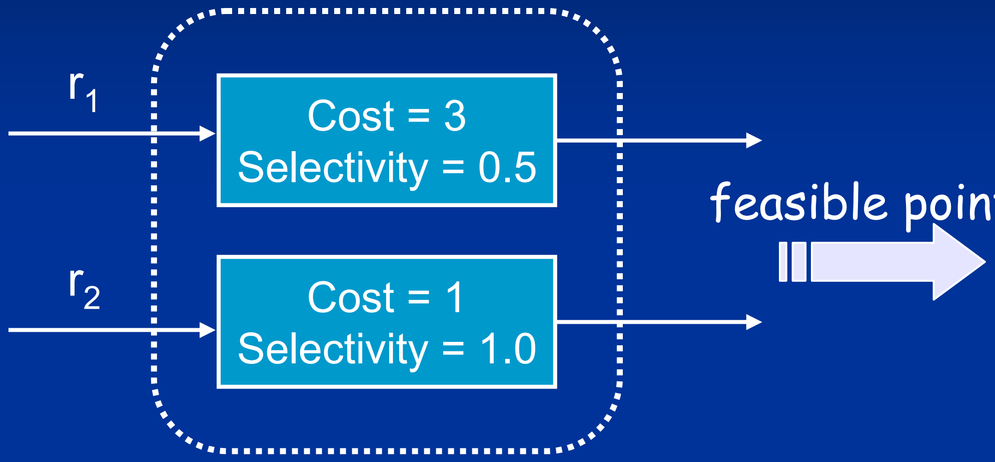
- Feasible Input Table (FIT): (r_1, .., r_m, quality-score)

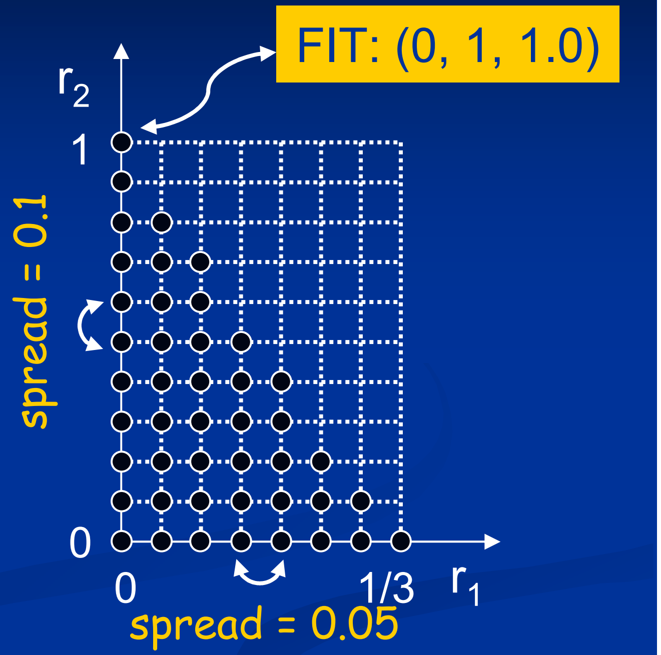


- FIT
  - Initial generation
  - Aggregation
  - Propagation
  - Load shedding

Generating FIT

- At each leaf node:

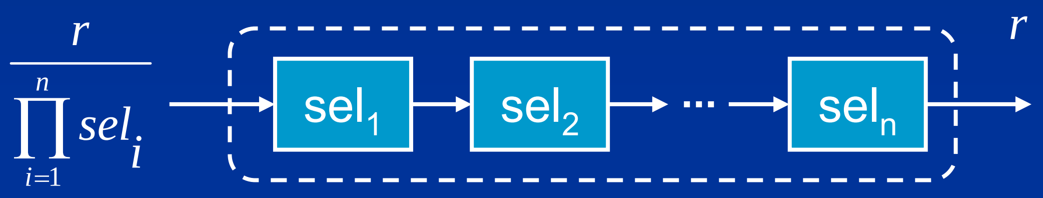




- Spread can be adjusted based on:
  - a fixed maximum error from the optimal, or
  - based on a fixed FIT size.
- If splits in the local query plan:
  - use an additional local plan that complements a FIT entry.

Aggregating and Propagating FIT

- When parent A receives FIT from child B:
  - Maps FIT entries from A's output to A's input.



  - Eliminates entries that are infeasible for A.
  - If splits along a path, propagates the maximum rate and keeps the rest as an additional local plan.
- If parent A has multiple child nodes:
  - Merges FIT entries pairwise.
  - Adds the quality scores.

FIT-based Load Shedding

- A node observes input rates (r_1, .., r_m).
- If there exists F in FIT where for all i, $F.r_i \geq r_i$ with no local plan:
  - Do nothing.
- Else:
  - Find F in FIT with the highest quality-score such that for all i, $F.r_i \leq r_i$.
  - Reduce r_i by $1\text{-}F.r_i/r_i$.
  - Apply the associated local plan if any.

Open Challenges

- Metadata (FIT) maintenance
- Fairness and Priorities
- Server topology
- Bandwidth bottlenecks
- Centralized vs. Distributed tradeoffs

Summary

- There is **load dependency** among the nodes of a distributed stream processing system.
- Distributed load shedding requires **global coordination** among nodes to ensure optimality.
- We can provide this coordination by **upstream metadata aggregation and propagation**.
- Results can be improved by using **additional local plans** that complement the metadata.

Questions?