Staying FIT: Efficient Load Shedding Techniques for Distributed Stream Processing

Nesime Tatbul

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

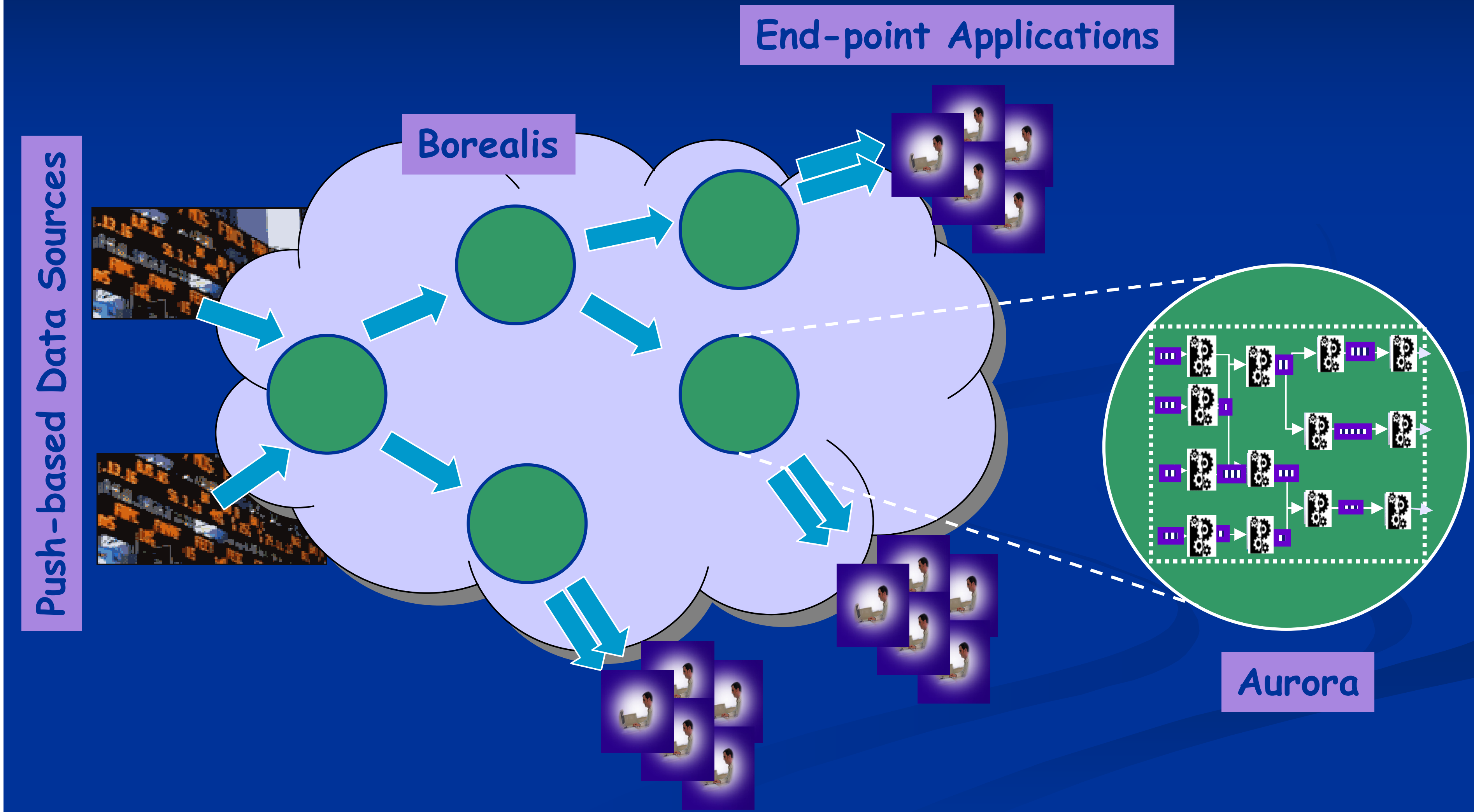
Uğur Çetintemel
Stan Zdonik

BROWN

Talk Outline

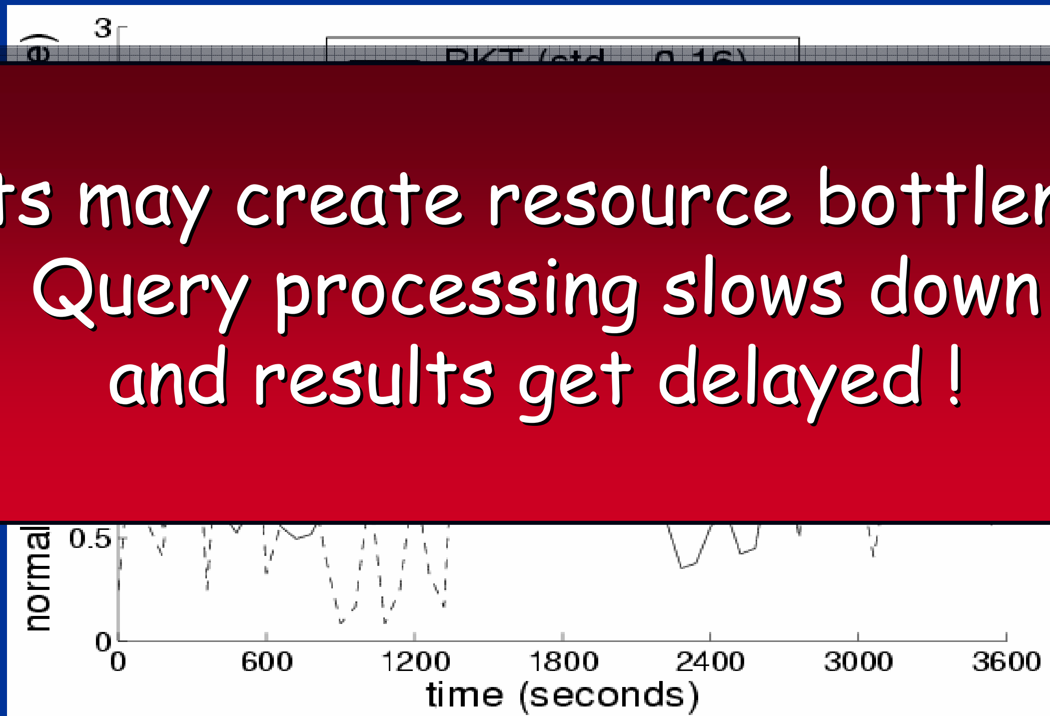
- Problem Introduction
- Approach Overview
- Advance Planning with an LP Solver
- Advance Planning with FIT
- Performance Results
- Related Work
- Conclusions and Future Work

Distributed Stream Processing
The Aurora/Borealis System



Bursty Workload

- Data can arrive fast, in unpredictable bursts
- Example: Network traffic data



Bursts may create resource bottlenecks:
Query processing slows down
and results get delayed !

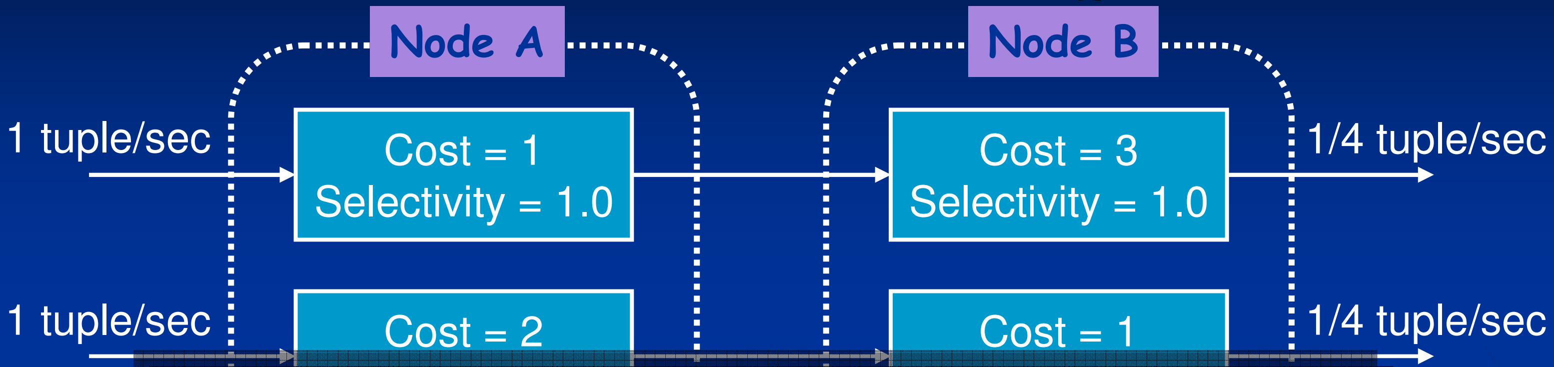
Source: Internet Traffic Archive, http://ita.ee.lbl.gov/

Models and Assumptions

- We focus on CPU as the limited resource.
- Load shedding is achieved by inserting probabilistic drop operators into query plans.
  - Random Drop [VLDB'03], Window Drop [VLDB'06]
  - Approximate result is a subset of the original result.
- The goal is to maximize the total weighted query throughput (e.g., [Ayad et al, SIGMOD'04, Amini et al, ICDCS'06]).
- Servers are arranged in a tree-like topology.

Distributed Load Shedding

Key Observation: Load Dependency



Server nodes must coordinate to achieve high-quality results!

Plan	Rates at A	A.load	A.throughput	B.load	B.throughput
0	1, 1	3	1/3, 1/3	4/3	1/4, 1/4
1	1, 0	1	1, 0	3	1/3, 0
2	0, 1/2	1	0, 1/2	1/2	0, 1/2
3	1/5, 2/5	1	1/5, 2/5	1	1/5, 2/5

optimal for A

feasible for both

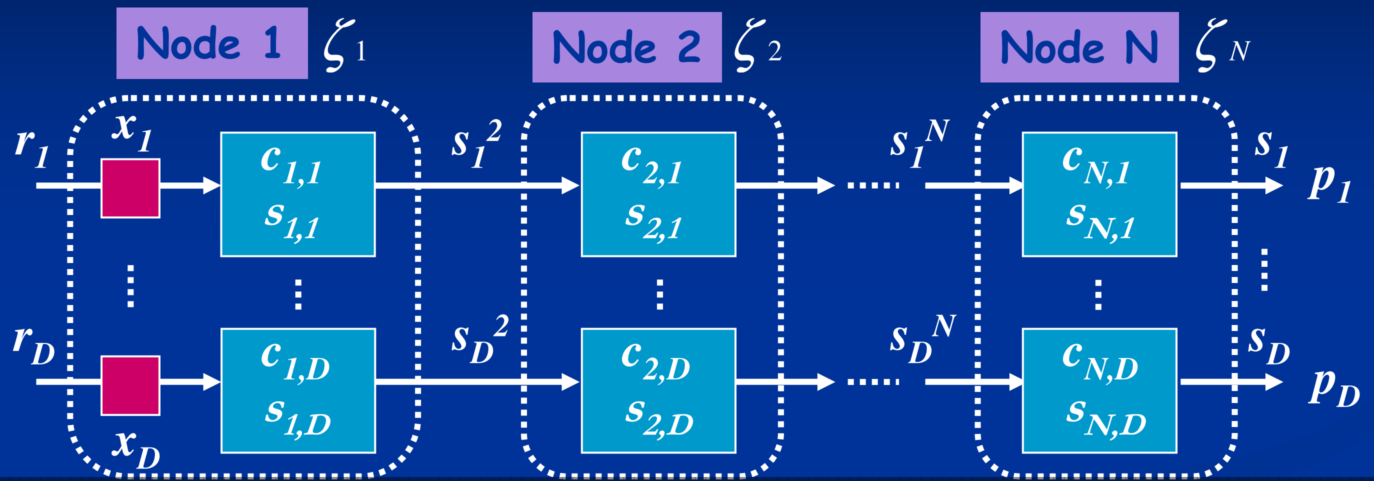
optimal for both

A.load ≤ 1

B.load ≤ 1

B.throughput: maximize !

Distributed Load Shedding
as a Linear Optimization Problem



Problem formulation for non-linear query plans (i.e., with operator splits and merges) is in the paper.

$$0 \leq x_j \leq 1$$

$$\sum_{j=1}^{D} r_j \times x_j \times s_j \times p_j \text{ is maximized.}$$

Talk Outline

- Problem Introduction
- Approach Overview
- Advance Planning with an LP Solver
- Advance Planning with FIT
- Performance Results
- Related Work
- Conclusions and Future Work

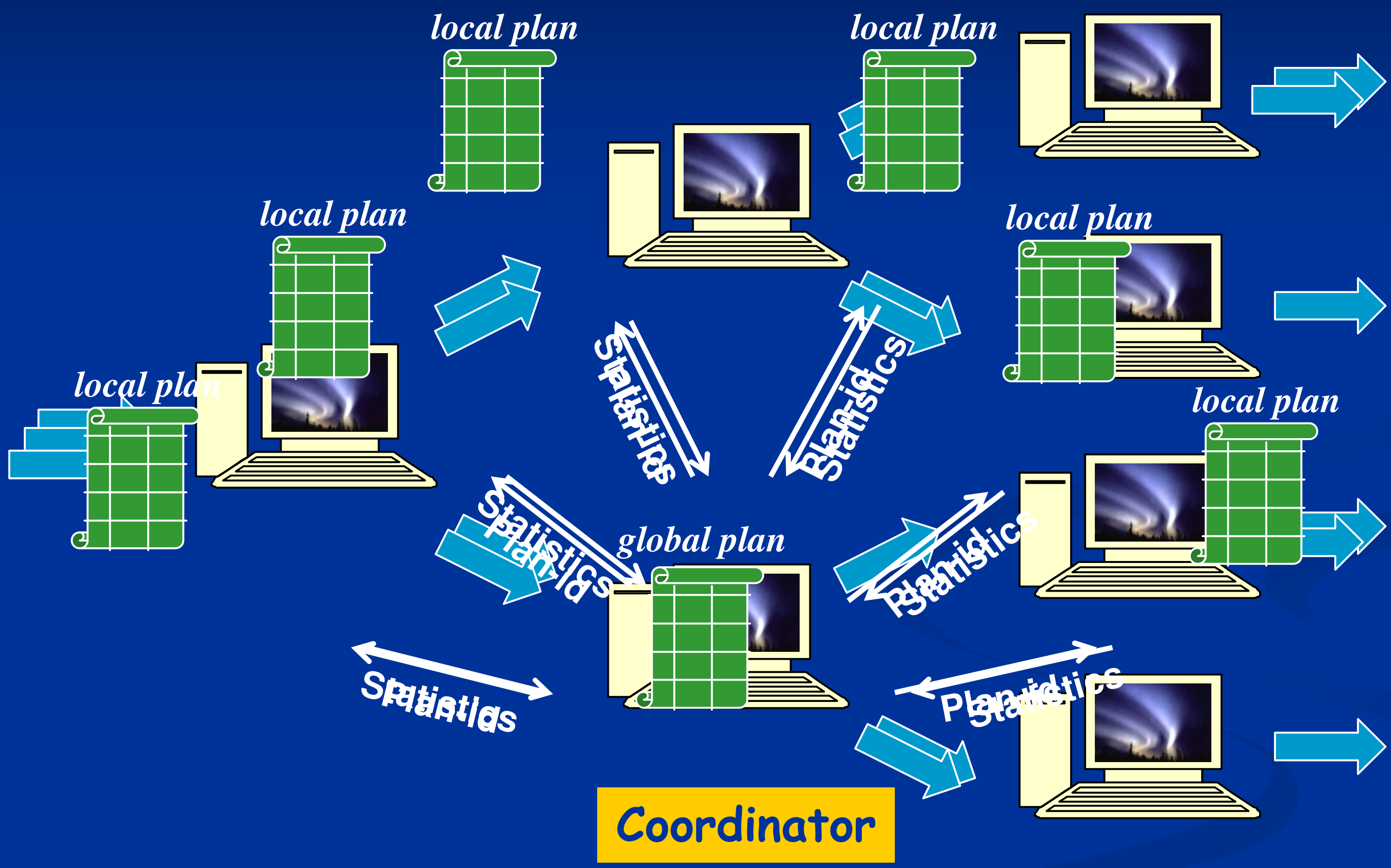
Architectural Overview

Centralized vs. Distributed

Load Shedding Phases	Distributed	Centralized
Advance Planning	All	Coordinator
Load Monitoring	All	Coordinator
Plan Selection	All	Coordinator
Plan Implementation	All	All

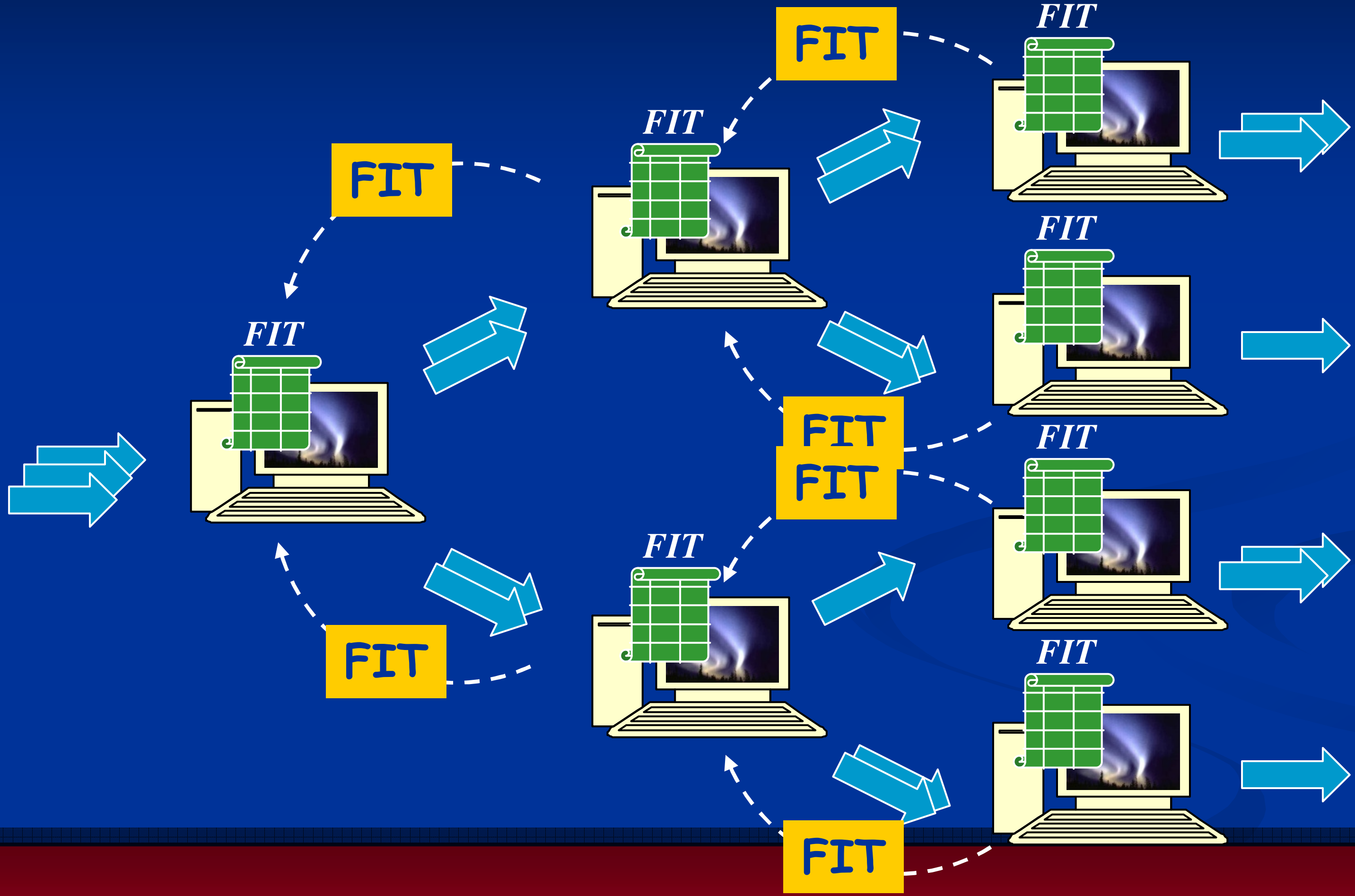
Architectural Overview

Centralized Approach



Architectural Overview

Distributed Approach

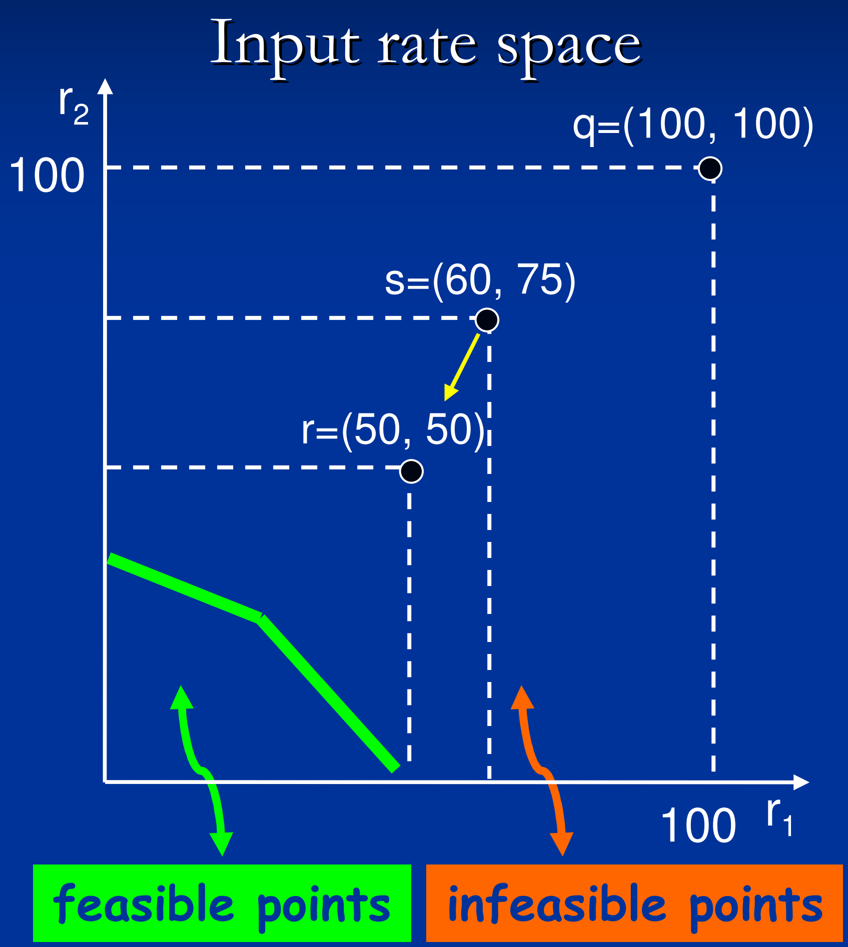


Feasible Input Table : (r_1, .., r_n, [local plan], quality)

Talk Outline

- Problem Introduction
- Approach Overview
- Advance Planning with an LP Solver
- Advance Planning with FIT
- Performance Results
- Related Work
- Conclusions and Future Work

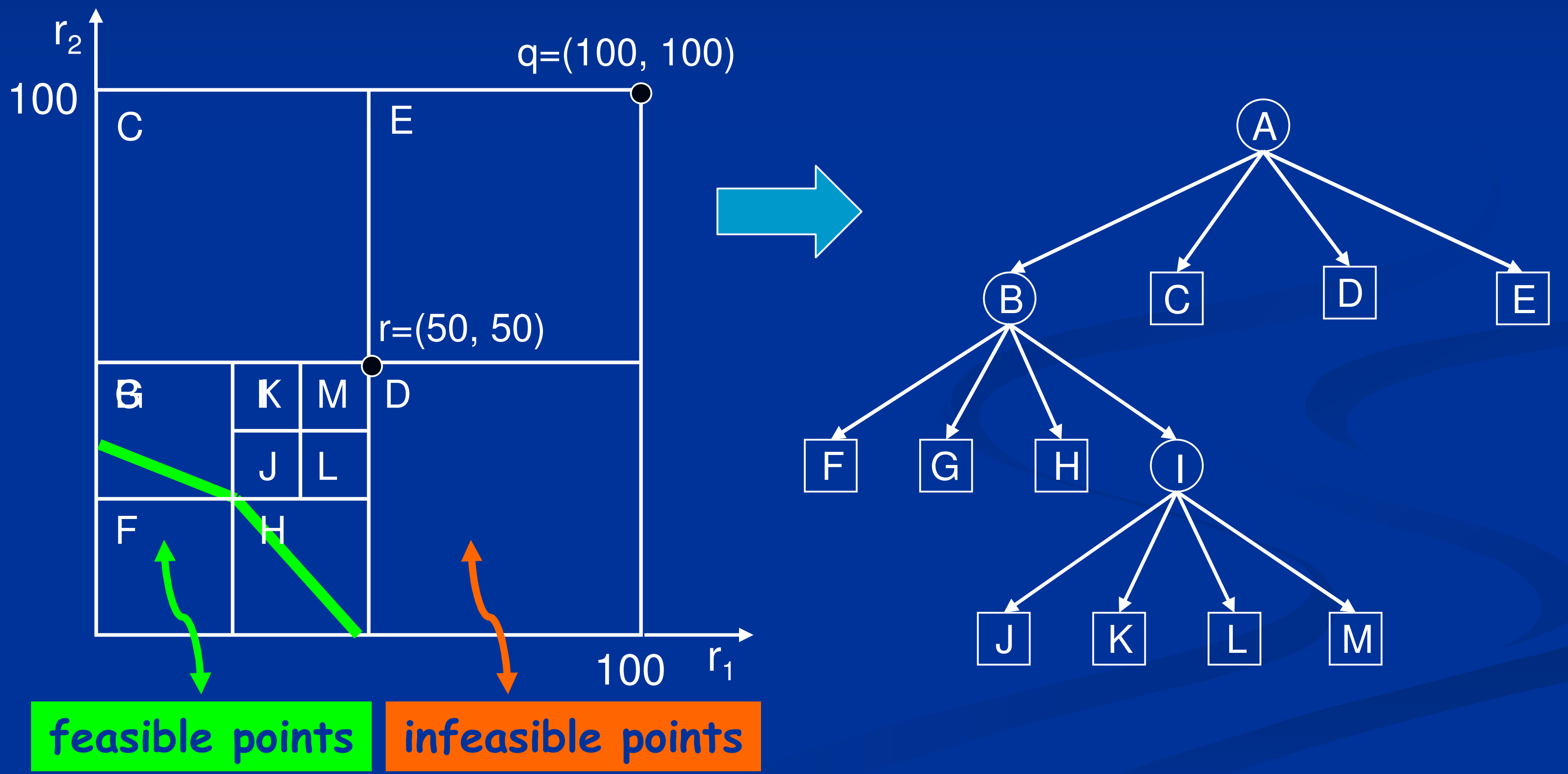
Advance Planning with an LP Solver
Approximate Load Shedding Plans



- Given an infeasible point, the Solver generates an optimal plan.
- We don't want to call the Solver for each infeasible point.
- Key observation:
  - $\text{Quality}_r \leq \text{Quality}_q$
- Assume an **error threshold "ε"** in quality. Given any infeasible point s such that r < s < q :
  - If $(\text{Quality}_q - \text{Quality}_r)/\text{Quality}_q \leq \varepsilon$, then s can use the load shedding plan for r, <u>with a minor modification</u>.

Advance Planning with an LP Solver
QuadTree-based Plan Index

- Use a Region QuadTree to divide and index the input rate space.



Advance Planning with an LP Solver

Exploiting Non-uniform Input Workload

- Infeasible points may be observed with different probabilities, i.e., some regions may have higher expected probability.

- Given a region with expected probability p,

  the expected maximum error for this region is:
  - $E[Error_{max}] = p * (Quality_{max} – Quality_{min}) / Quality_{max}$

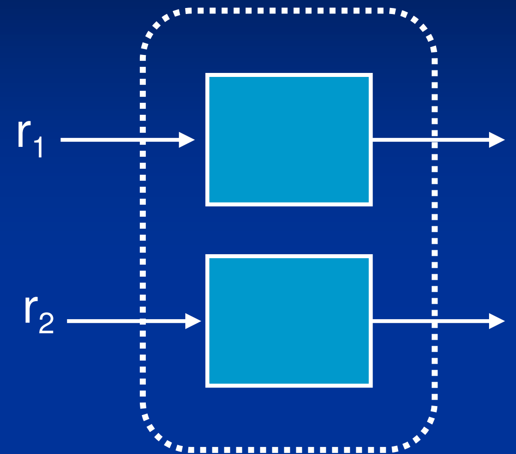
- For all regions, we must make sure that:
  - $Total(E[Error_{max}]) \leq \varepsilon$

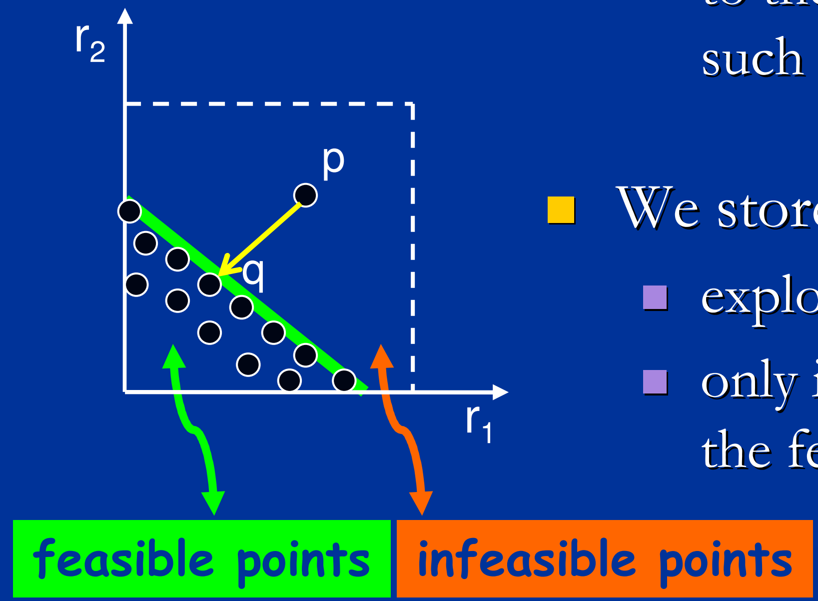
Talk Outline

- Problem Introduction
- Approach Overview
- Advance Planning with an LP Solver
- Advance Planning with FIT
- Performance Results
- Related Work
- Conclusions and Future Work

Advance Planning with FIT

FIT Basics

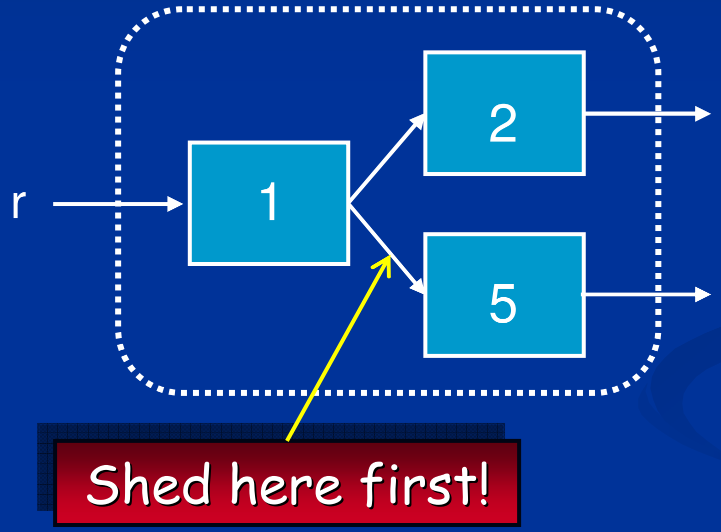




- We store feasible points in FIT:
  - (r_1, r_2, [local plan], quality)
- FIT-based load shedding:
  - Given an infeasible point p, p must be mapped to the highest quality feasible point q in FIT, such that $q \leq p$.
- We store a reduced number of FIT points by:
  - exploiting the "ε" error tolerance threshold, and
  - only including the FIT points that are "close" to the feasibility boundary.

Advance Planning with FIT
Complementary Local Plans

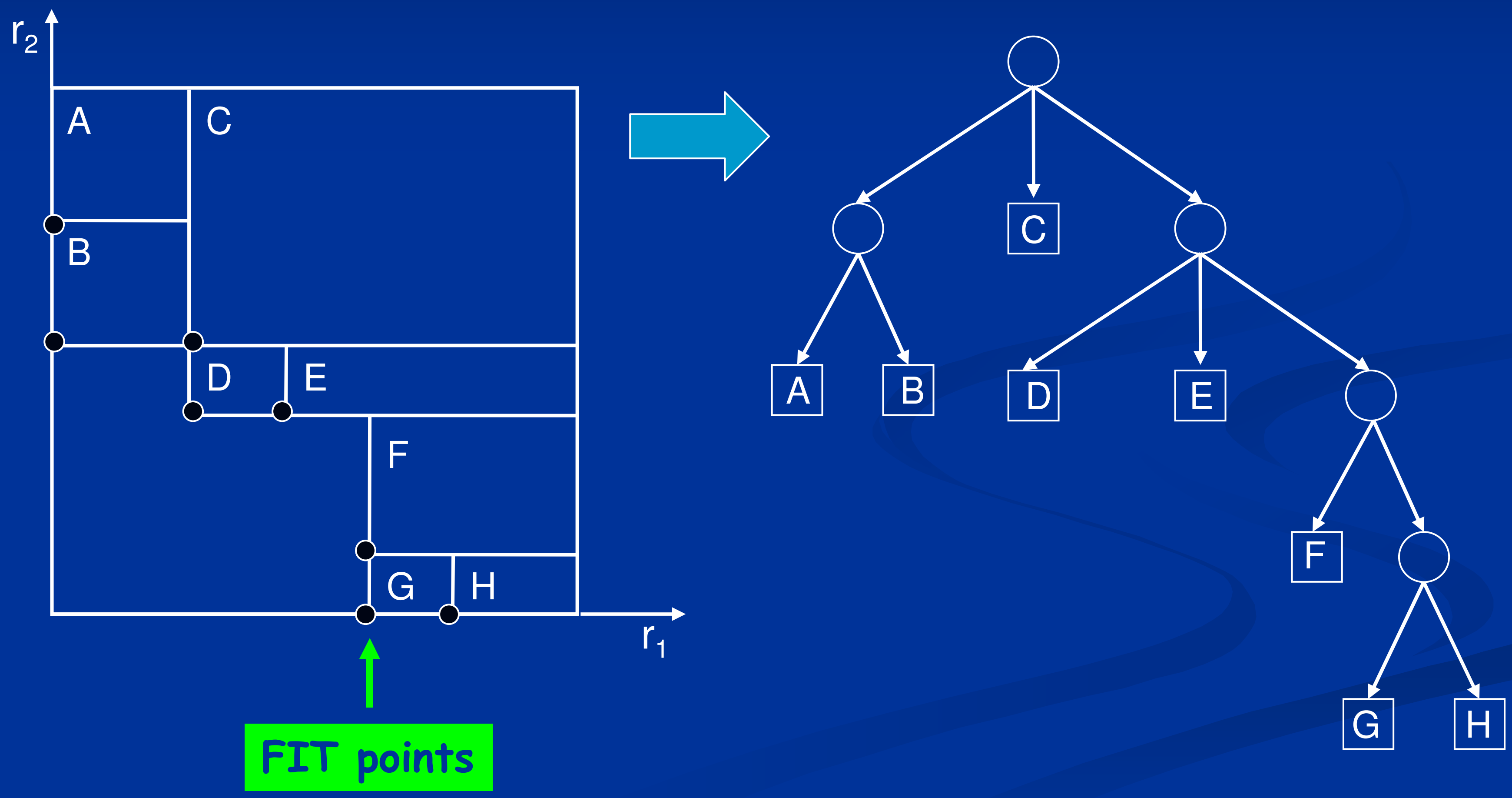
- Complementary local load shedding plans may be needed for nodes with operator splits.
- Example:



- Local plans are not propagated upstream.

Advance Planning with FIT
QuadTree-based Plan Index

- Use a Point QuadTree to divide and index the input rate space.



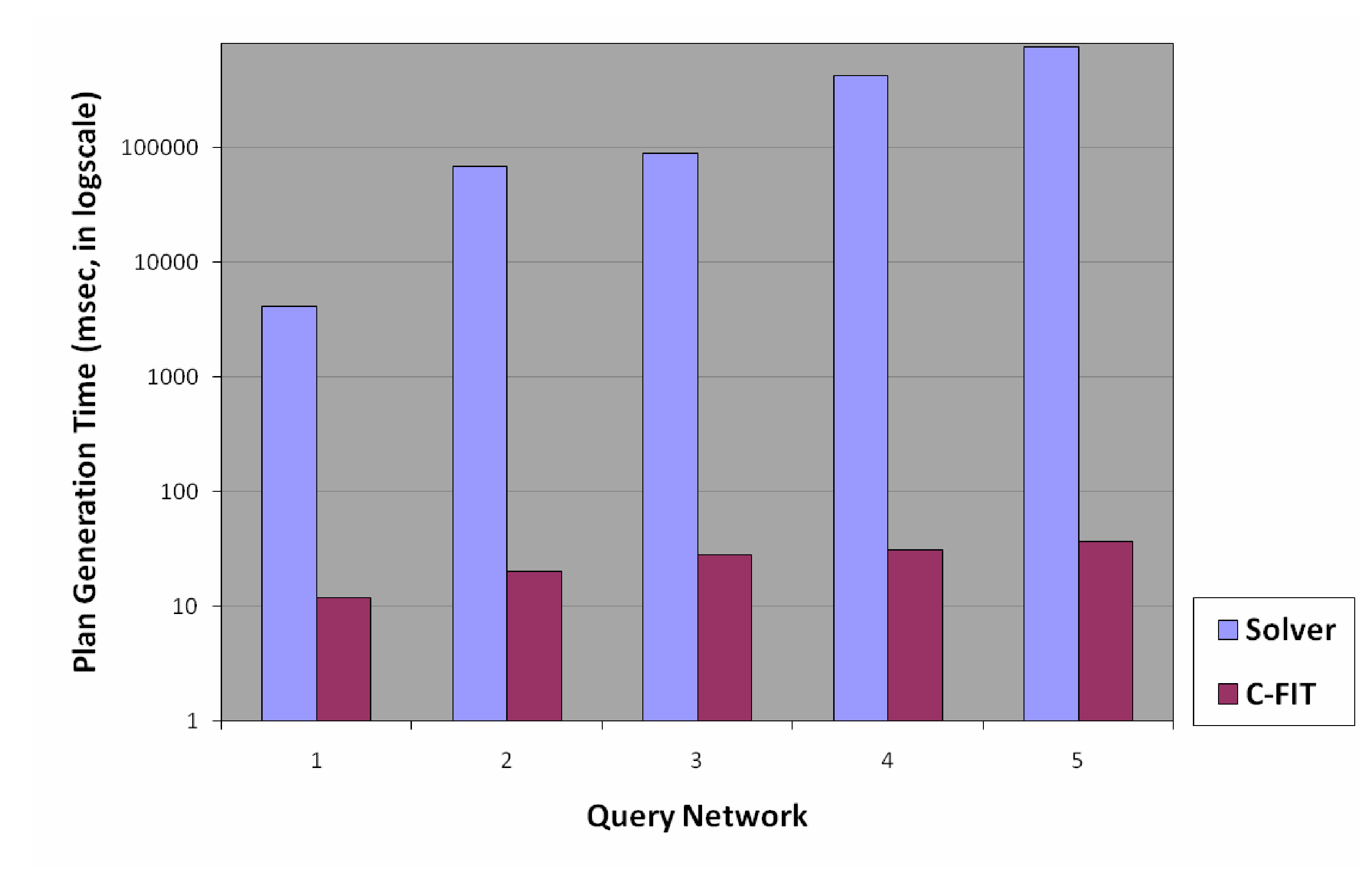
Talk Outline

- Problem Introduction
- Approach Overview
- Advance Planning with an LP Solver
- Advance Planning with FIT
- Performance Results
- Related Work
- Conclusions and Future Work

Experimental Setup

- Implemented on Borealis
- Query networks with Delay(cost, selectivity) operators
- Two input workloads:
  - Synthetic: Exponential distribution
  - Network traffic traces from the Internet Traffic Archive
- Goals:
  - Analyze plan generation efficiency for
    - Solver, Solver-W, C-FIT
  - Analyze communication overhead for
    - D-FIT

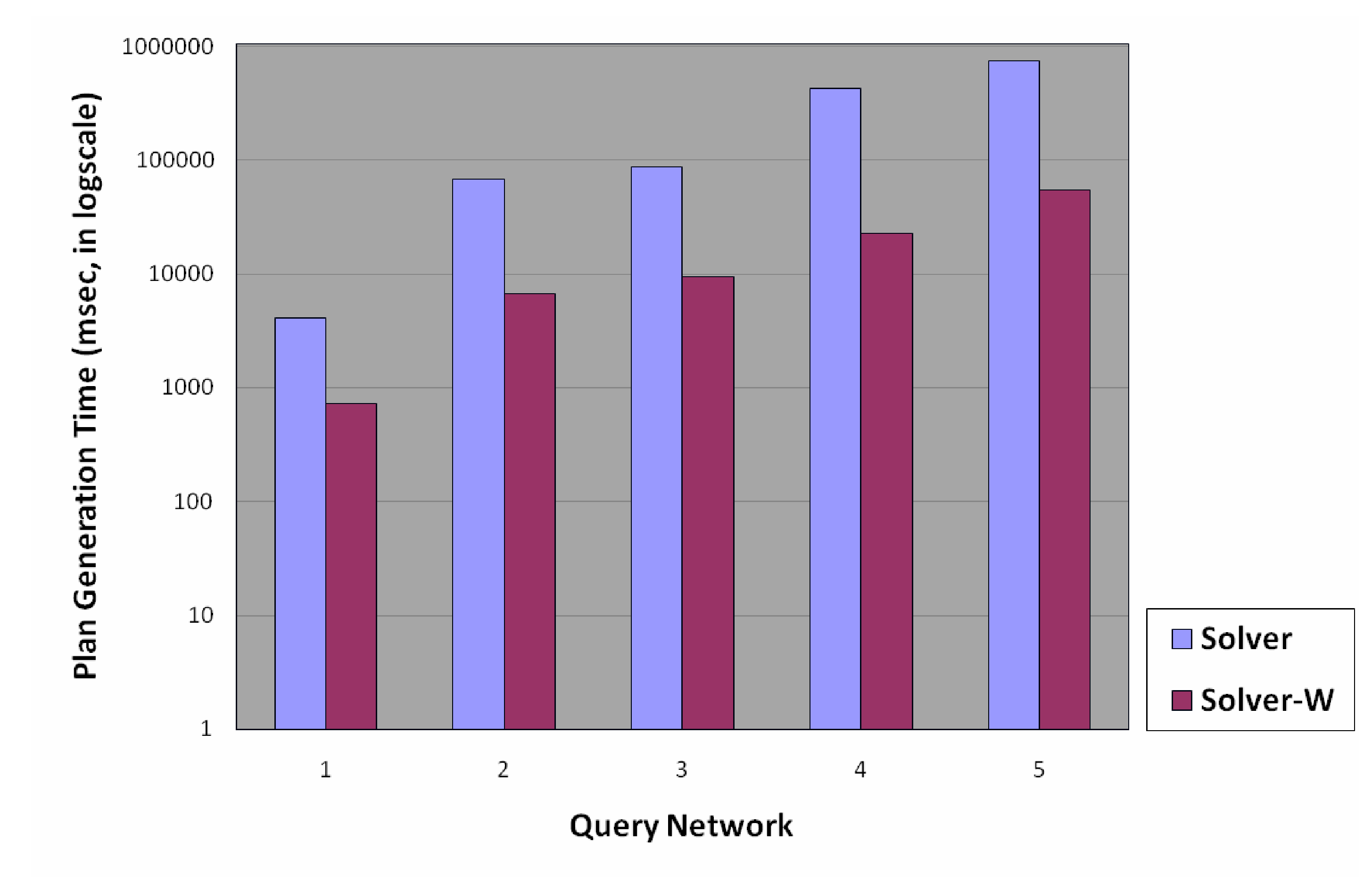
Plan Generation Performance
Solver vs. C-FIT



2 x 2 query networks with different cost distributions
(Maximum Error ε = 0.05)

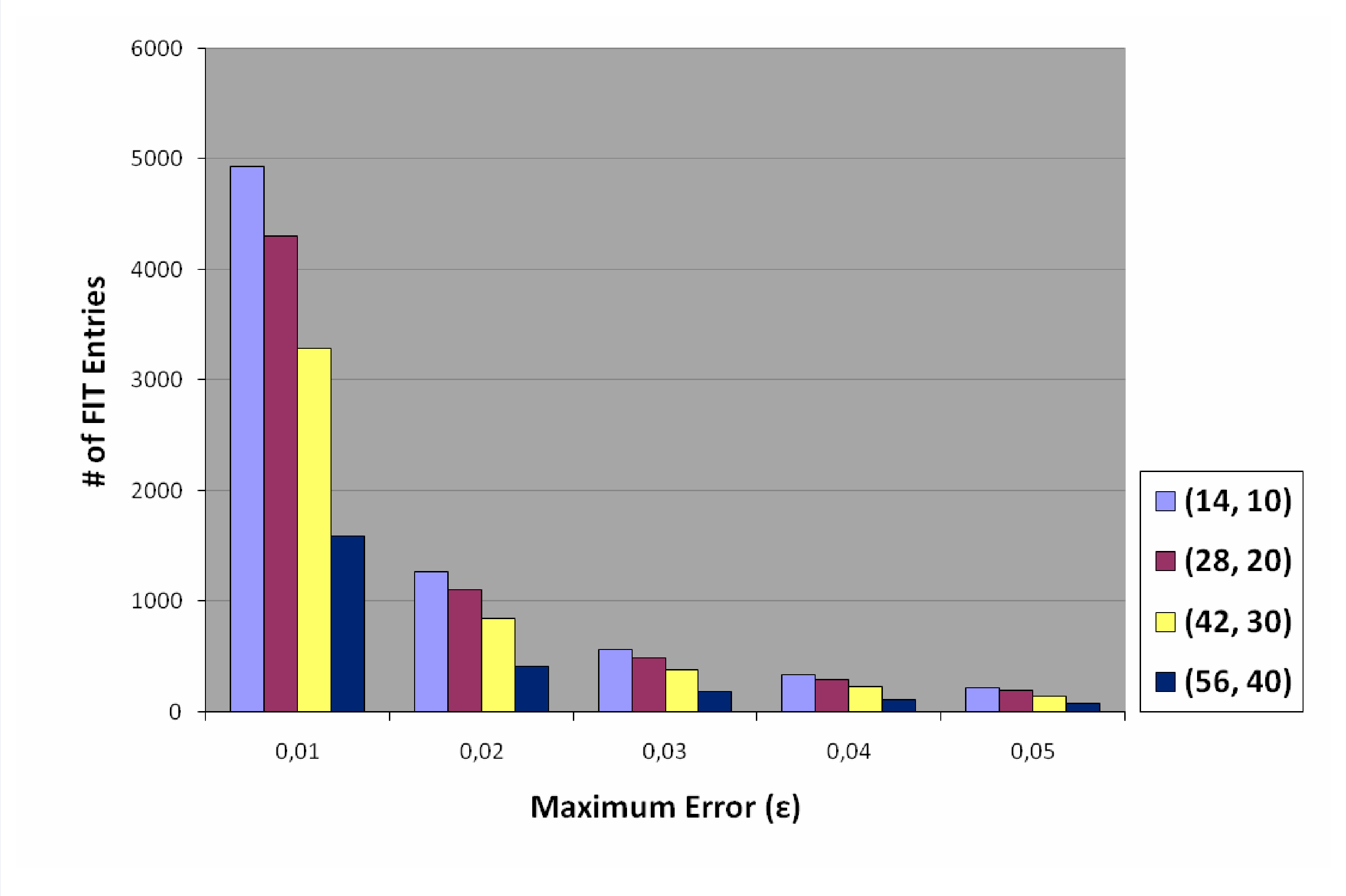
Plan Generation Performance

Solver vs. Solver-W



2 x 2 query networks with different cost distributions
(Expected Maximum Error E[ε] ~ 0.015)

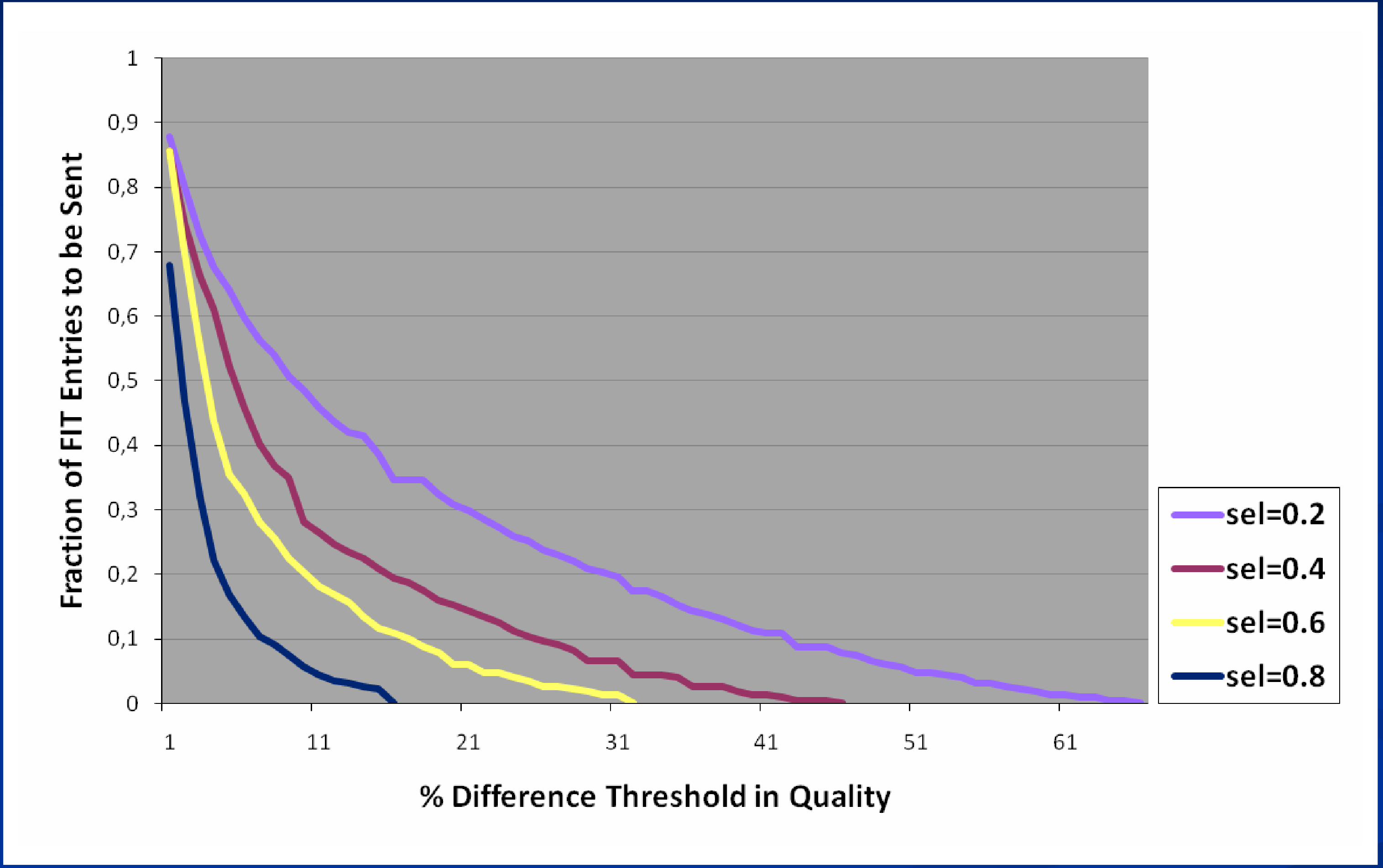
D-FIT Communication Overhead
Effect of Query Cost and Error Tolerance



2 x 2 query networks with different query costs

D-FIT Communication Overhead

Sensitivity to Selectivity Change



2 x 2 query networks with different operator selectivity
(Initial selectivity = 1.0)

Talk Outline

- Problem Introduction
- Approach Overview
- Advance Planning with an LP Solver
- Advance Planning with FIT
- Performance Results
- Related Work
- Conclusions and Future Work

Related Work

- Load shedding for the single server case. Examples: [Tatbul et al, VLDB'03/VLDB'06], [Babcock et al, ICDE'04] [Ayad et al, SIGMOD'04], [Reiss et al, ICDE'05], …
- Control-based load management in System S [Amini et al, ICDCS'06]
- Aggregate congestion control against DoS attacks [Mahajan et al, SIGCOMM CCR'02]
- Parametric query optimization [Ioannidies et al, VLDB'92], [Ganguly, VLDB'98], [Hulgeri et al, VLDB'02]

Conclusions

- Distributed load shedding requires coordination among the servers.
- We provide centralized and distributed alternatives.
- We propose efficient techniques for advance generation of load shedding plans:
  - Approximate load shedding plans
  - QuadTree-based plan indexing
  - Exploiting input workload distribution
- Distributed FIT is better for dynamic environments.

Future Work

- Performance on larger scale networks
- Bandwidth bottlenecks
- Non-tree server topologies
- Hybrid approaches
  - Centralized + Distributed
  - Local plan refinement
- Other quality metrics

More information:

http://www.cs.brown.edu/research/borealis/

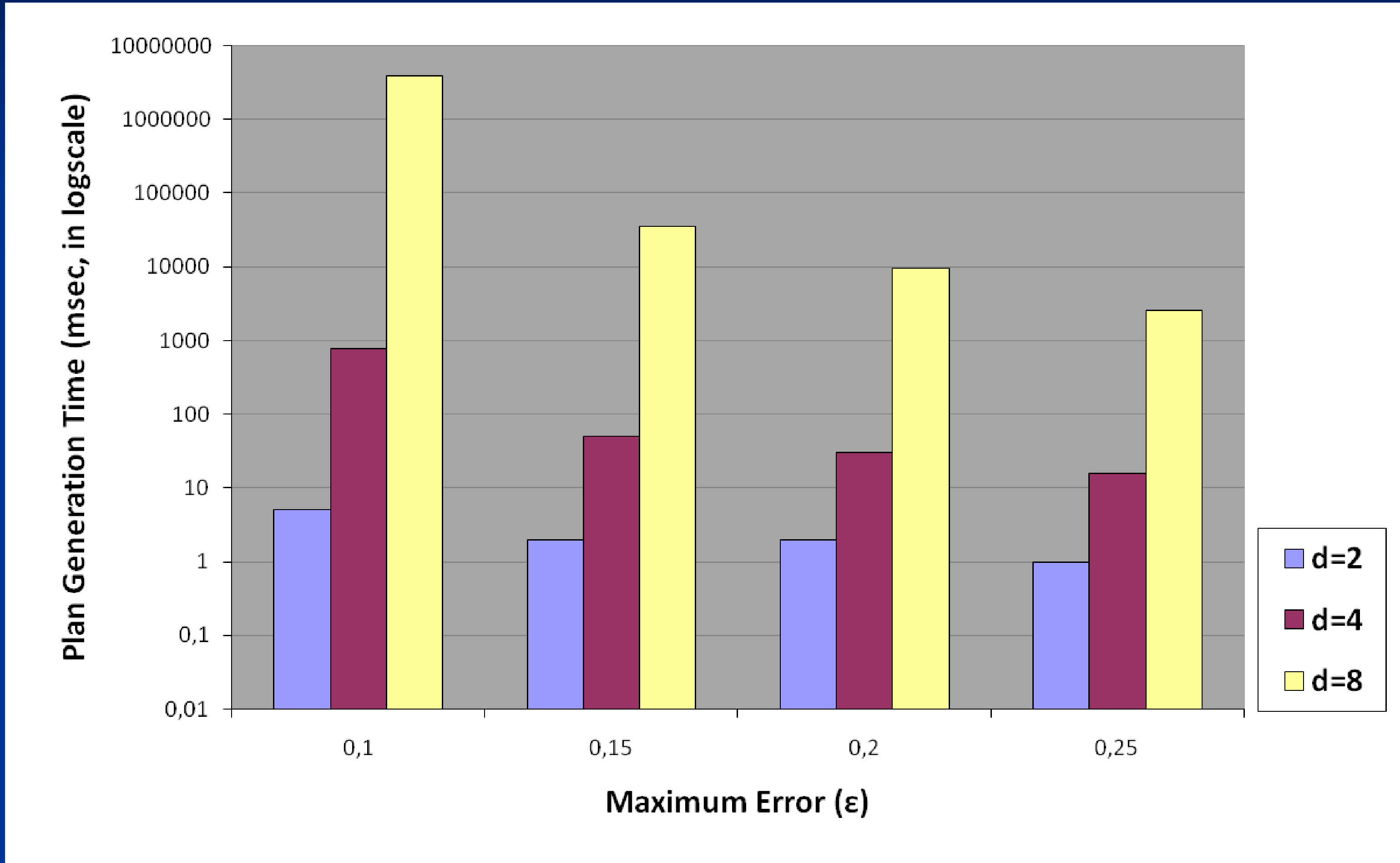
http://www.inf.ethz.ch/~tatbul

Questions?

Advance Planning with FIT
Upstream FIT Propagation

- Each leaf node generates its FIT from scratch, and propagates it to its upstream parent.
- Each non-leaf node, upon receiving FITs from its children:
  1. Maps the FIT rates from its outputs to its own inputs (<u>Note:</u> Mapping across splits may result in local plans).
  2. Merges multiple FITs into a single FIT.
  3. Removes the FIT entries that are infeasible for itself.
  4. Propagates the resulting FIT further upstream.

Plan Generation Performance
Effect of Input Dimensionality (C-FIT)



2 x 2, 4 x 2, 8 x 2 query networks with the same total cost