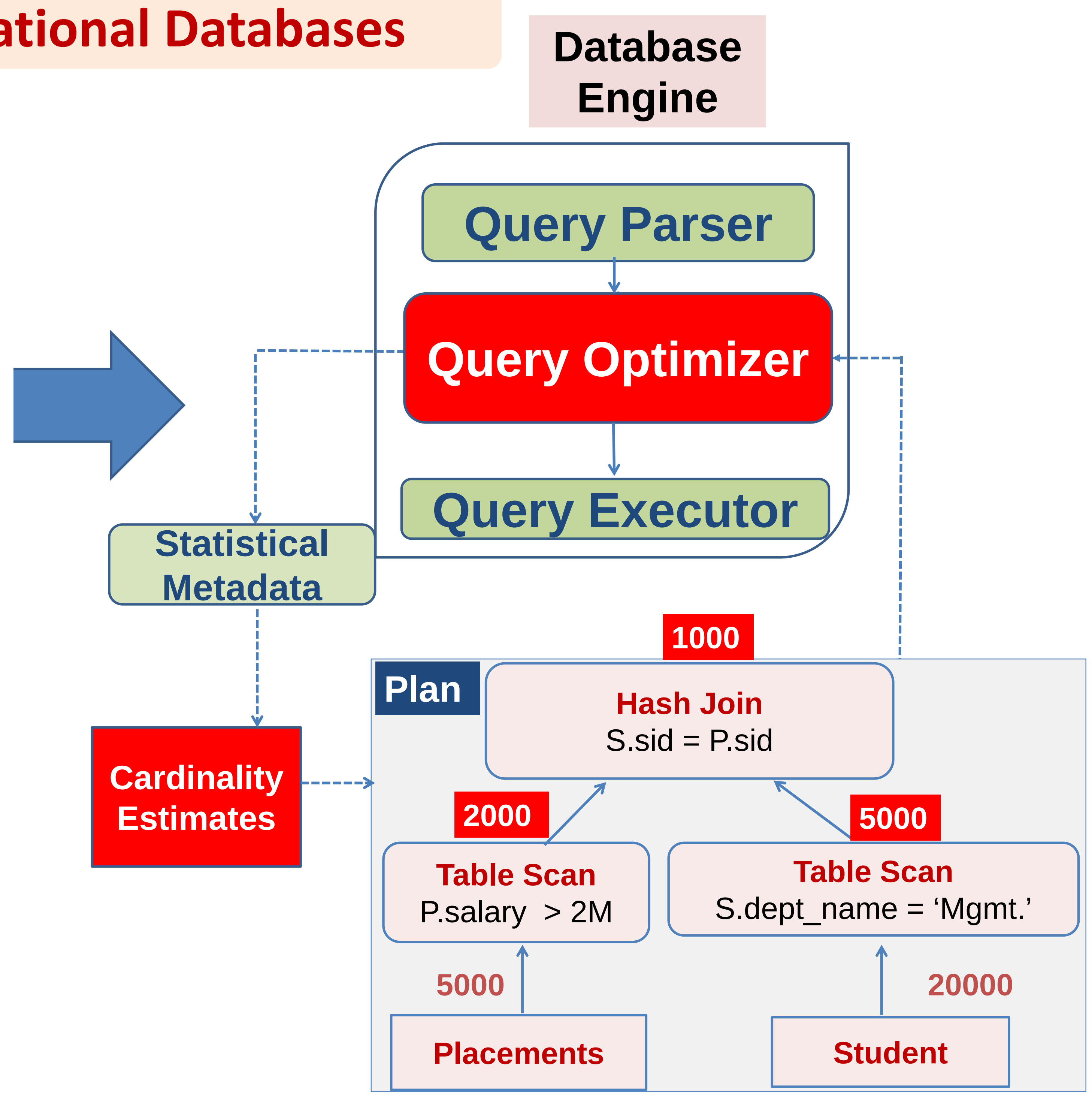
Database Engine Design for Robust Query Processing

Srinivas Karthik
Advisor: Prof. Jayant Haritsa
Indian Institute of Science

Database Systems Laboratory
Indian Institute of Science

Query Processing in Relational Databases



(SQL version)
Select Count (S.sid)
From Student S, Placements P
Where S.sid = P.sid
and S.dept_name = 'Management'
and P.salary > 2M

Problem of Cardinality Mis-estimates

Compile-time cardinality estimation errors cause orders of magnitude slower run-time, which can reach the millions!

Robustness Metric

$$SubOpt(est, act) = \frac{Cost(Optimizer\ Chosen\ Plan\ \ at\ act)}{Cost(Optimal\ Plan\ \ at\ act)}$$

$$\text{MaxSubOpt (MSO)} = Max[\text{ SubOpt}(est, act)] \quad \forall\ est\ ,\ act$$

Worst case impact of estimation errors

MSO ranges over [1, ∞)

GOAL: Propose query processing algorithms that provides MSO guarantee as close to 1 as possible

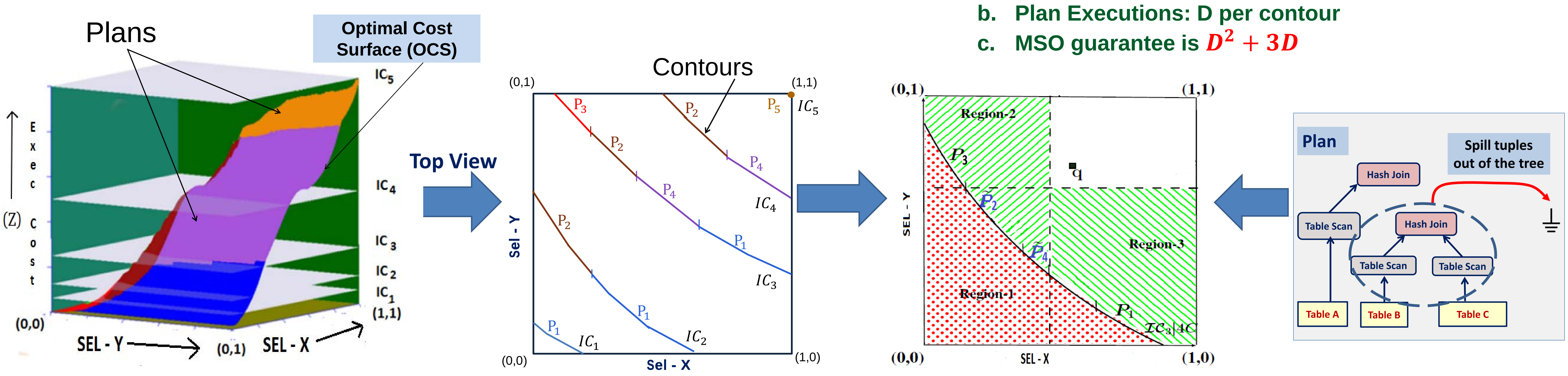
SpillBound Algorithm
[IEEE ICDE '16 : Best Student Paper Award]

Compile-time

1. Identify predicates prone to estimation errors (D)
2. Construct OCS
3. Cut OCS with isocost planes having doubling cost

Execution-time

1. Naïve:
   a. Hypograph Pruning
   b. Plan Executions: Execute all plans in a contour
2. SpillBound:
   a. Half-space Pruning – spilling mode plan executions
   b. Plan Executions: D per contour
   c. MSO guarantee is $D^2 + 3D$

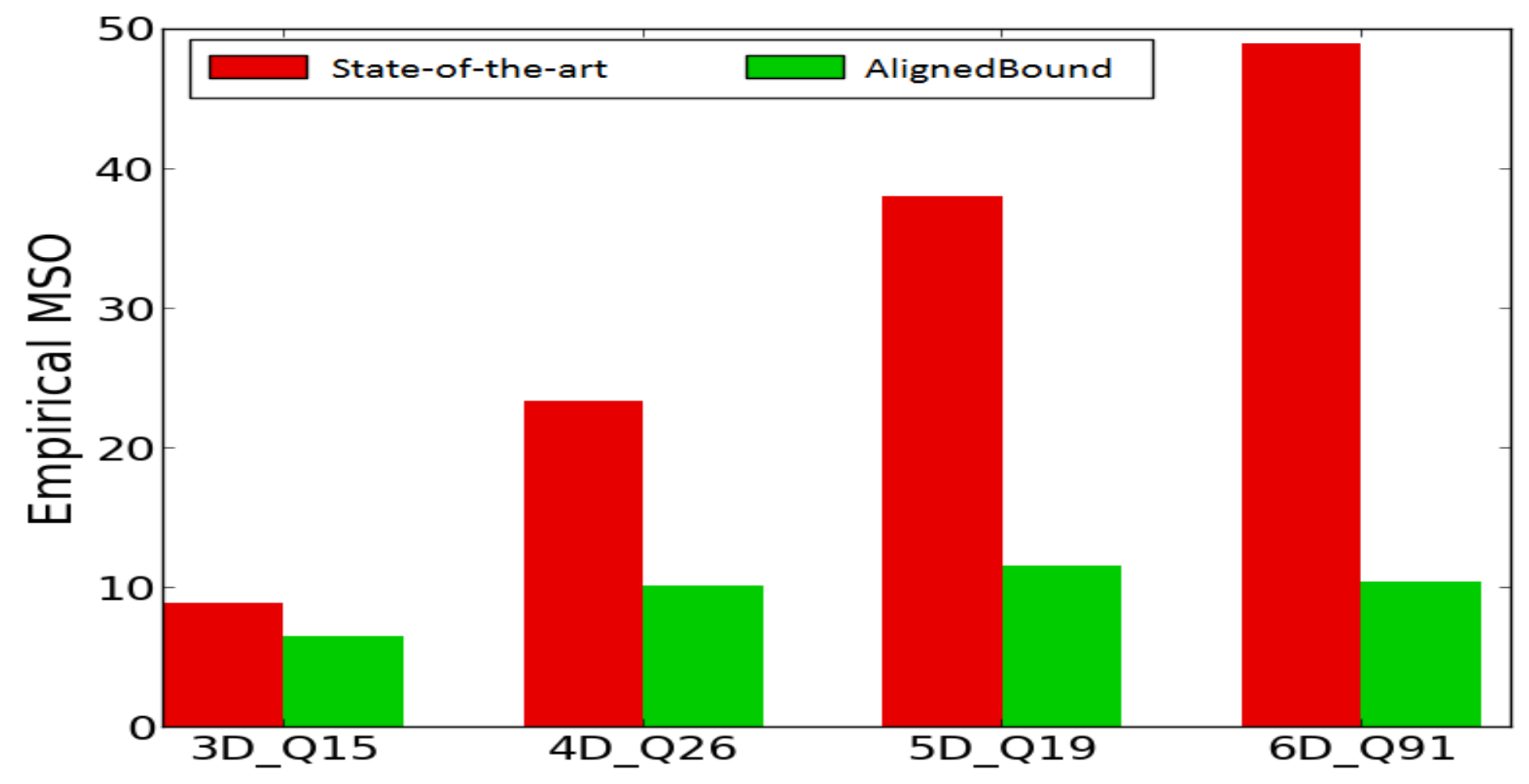


Additional Results

Lower Bound [ICDE '16]
MSO Lower Bound of *Ω(D)*

AlignedBound [TKDE '17]
Execute, for most queries, atmost **1** execution per contour, thus empirically matching MSO guarantee of $2D + 2$

Robustness Results
[PostgreSQL/TPC-DS]



Observations

1. Empirical performance of AlignedBound significantly better than state-of-the-art
2. Algorithms collapse the enormous MSO (in millions) down to a *single order of magnitude*

Publications

- S. Karthik et al. "Platform-independent Robust Query Processing" IEEE ICDE 2016
- S. Karthik et al. "Platform-independent Robust Query Processing" IEEE TKDE Journal 2017

Ongoing

- **Online PlanBouquet:** Handling dynamic queries wherein the expensive pre-processing efforts are unviable.
- **Dimensionality reduction:** We observed that some of the dimensions in a query could be removed while reducing the MSO guarantee.

TAKEAWAY Our proposed algorithms provides a significant step forward in robust query processing!

Database Design for Robust Query Processing

Srinivas Karthik

Advisor: Prof. Jayant Haritsa

CSA, IISc

DBMS

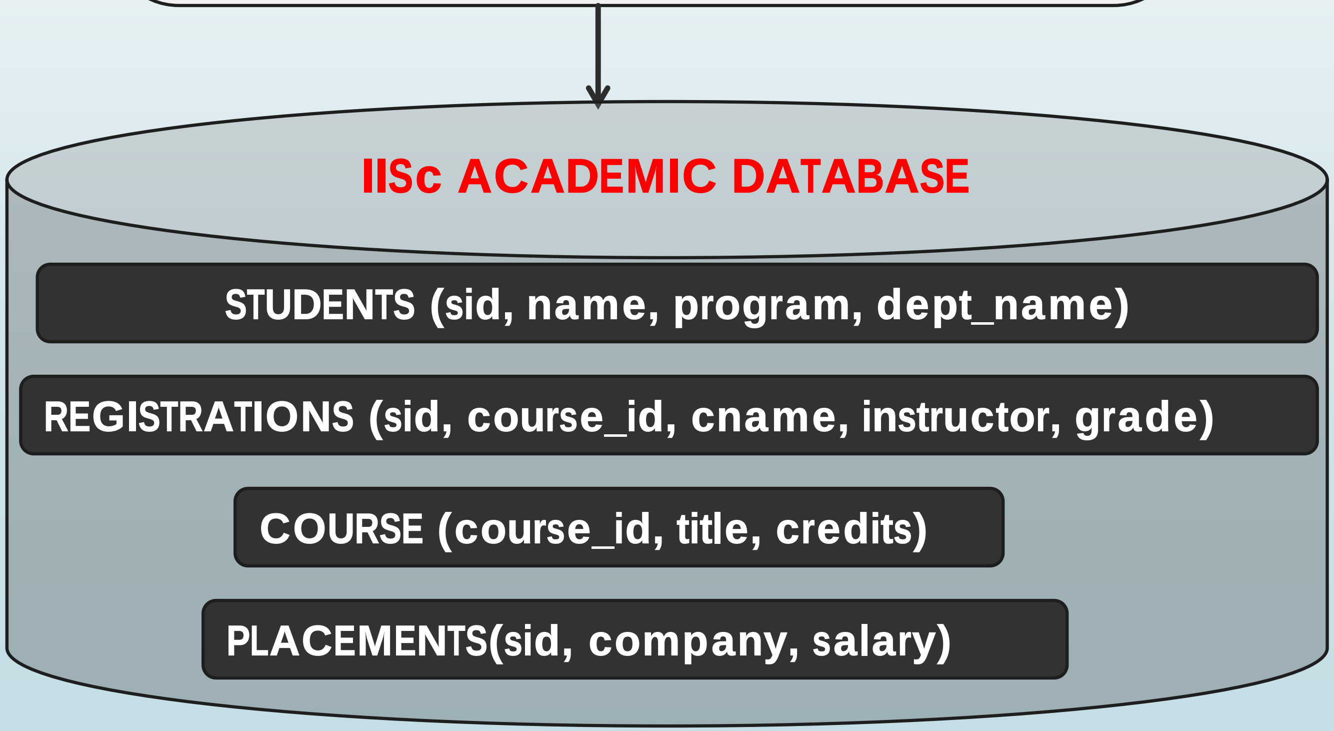
Declarative not Procedural

How many management students secured more than 2M salary?

(SQL version)

```
Select    count(S.sid )
From     Students S,  Placements P
Where   S.sid = P.sid
    and   S.dept_name  = ‘Management’
    and   P.salary  > 2M
```

Student



IISc ACADEMIC DATABASE

STUDENTS (sid, name, program, dept_name)

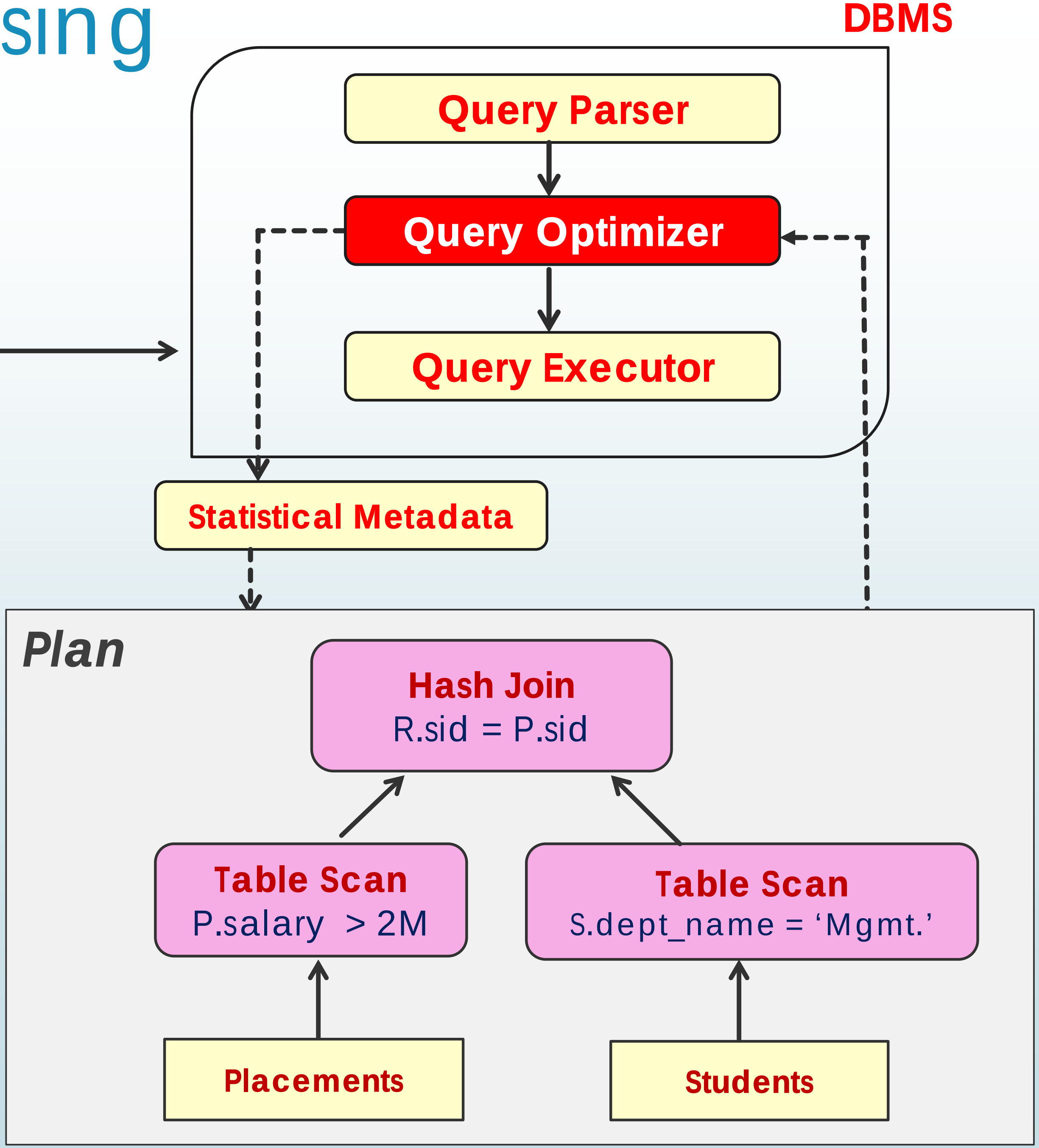
REGISTRATIONS (sid, course_id, cname, instructor, grade)

COURSE (course_id, title, credits)

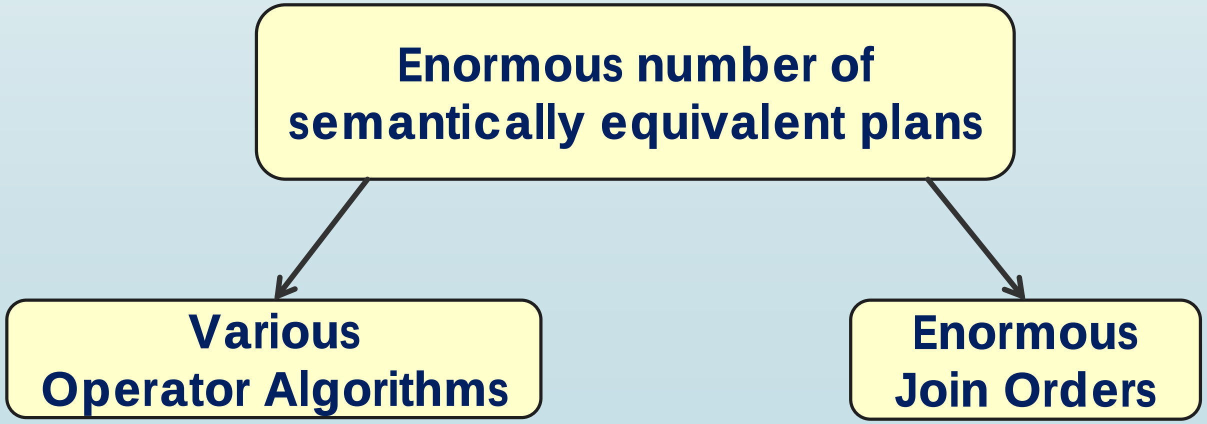
PLACEMENTS(sid, company, salary)

Declarative Query Processing

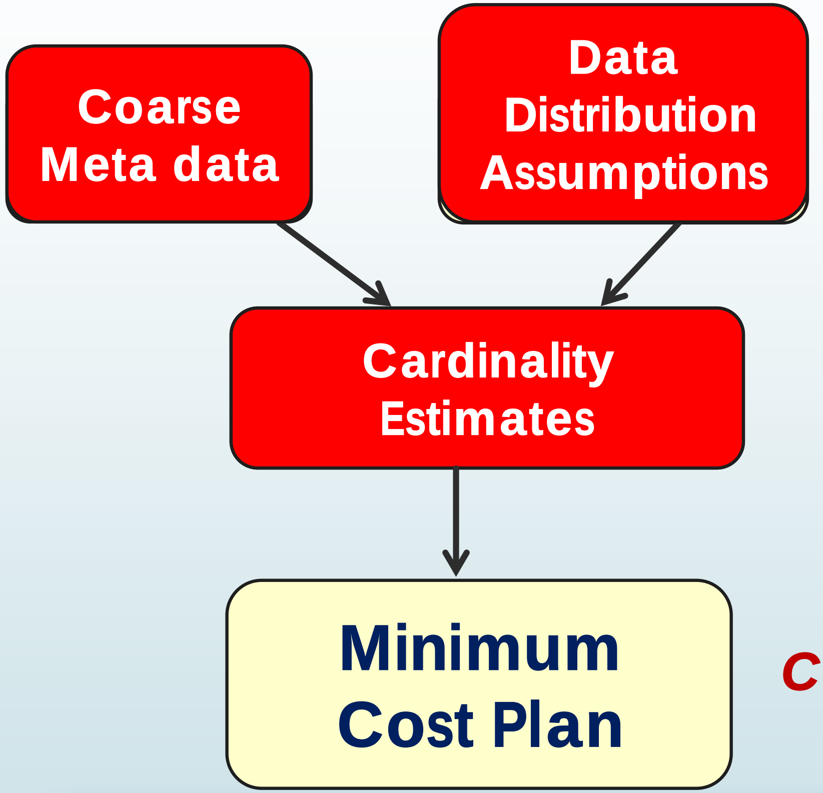
```
Select   count (S.sid )
From  Students S, Placements P
Where   S.sid = P.sid
    and   S.dept_name = ‘Management’
    and   P.salary  > 2M
```



Plan: tree of operators to process the data


Enormous number of semantically equivalent plans
Various Operator Algorithms
Enormous Join Orders

Query Optimizer



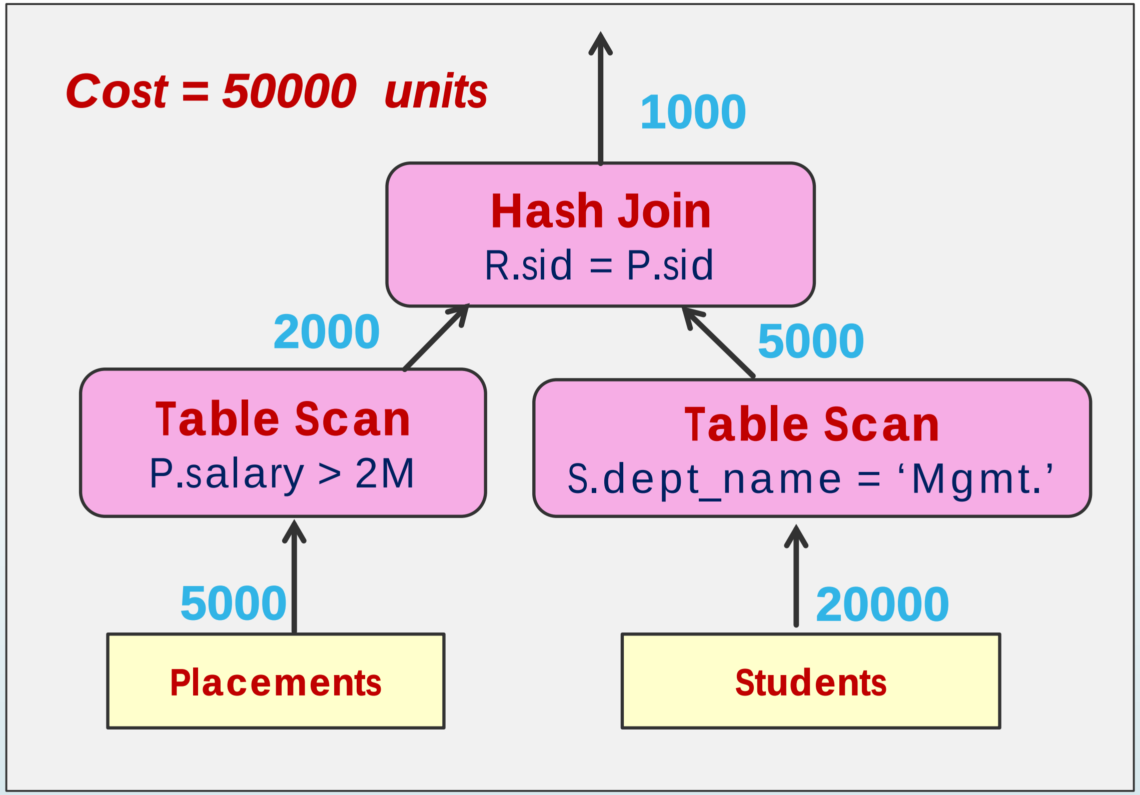
Coarse Meta data

Data Distribution Assumptions

Cardinality Estimates

Minimum Cost Plan

Cost: *Measure of query response time*



Cost = 50000 units

1000

Hash Join
R.sid = P.sid

2000

5000

Table Scan
P.salary > 2M

Table Scan
S.dept_name = 'Mgmt.'

5000

20000

Placements

Students

Compile-time plan can be highly sub-optimal at run-time (even in orders of magnitude)

GOAL

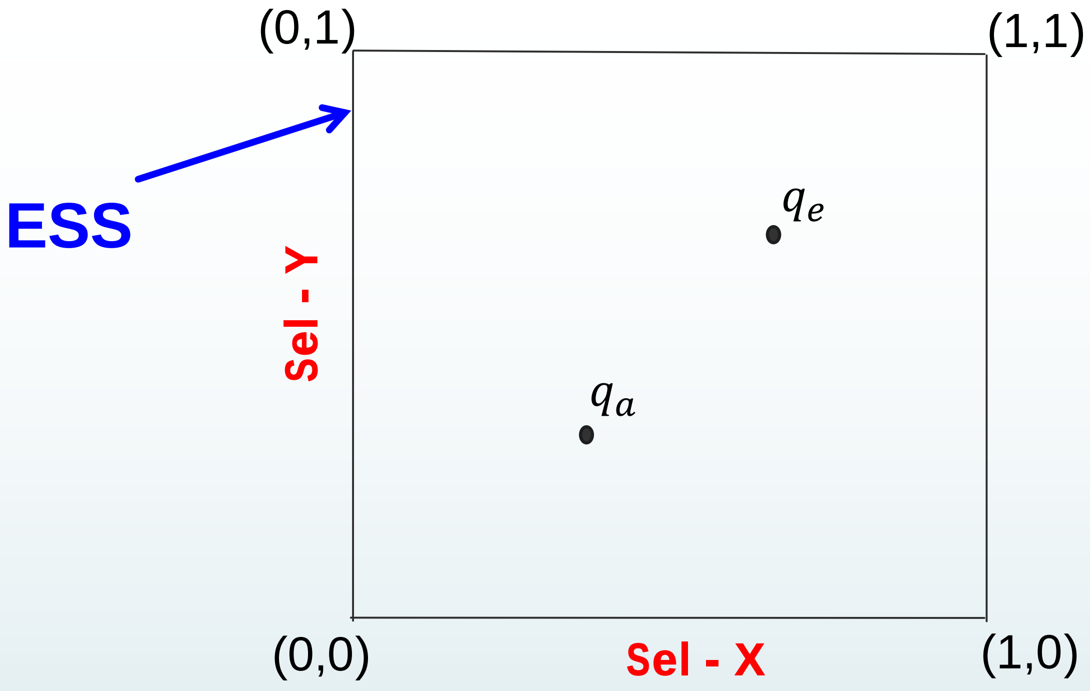
Design Robust Query Processing Algorithms

Robustness Metric: MSO

- ✓ Selectivity = normalized cardinalities in the range [0,1]
- ✓ $\mathbf{q_a}$: actual selectivity encountered during execution
- ✓ $\mathbf{q_e}$: optimizer's estimated selectivity
- ✓ SubOptimality (denoted by SubOpt) incurred by optimizer chosen plan instead of optimal plan

$$SubOpt(\mathbf{q_e}, \mathbf{q_a}) = \frac{Cost(Optimizer\ Chosen\ Plan\ \ at\ \mathbf{q_a})}{Cost(Optimal\ Plan\ \ at\ \mathbf{q_a})}$$

$$MaxSubOpt\ (MSO) = MAX[SubOpt(q_e, q_a)]\ \ \forall q_e, q_a \in ESS$$



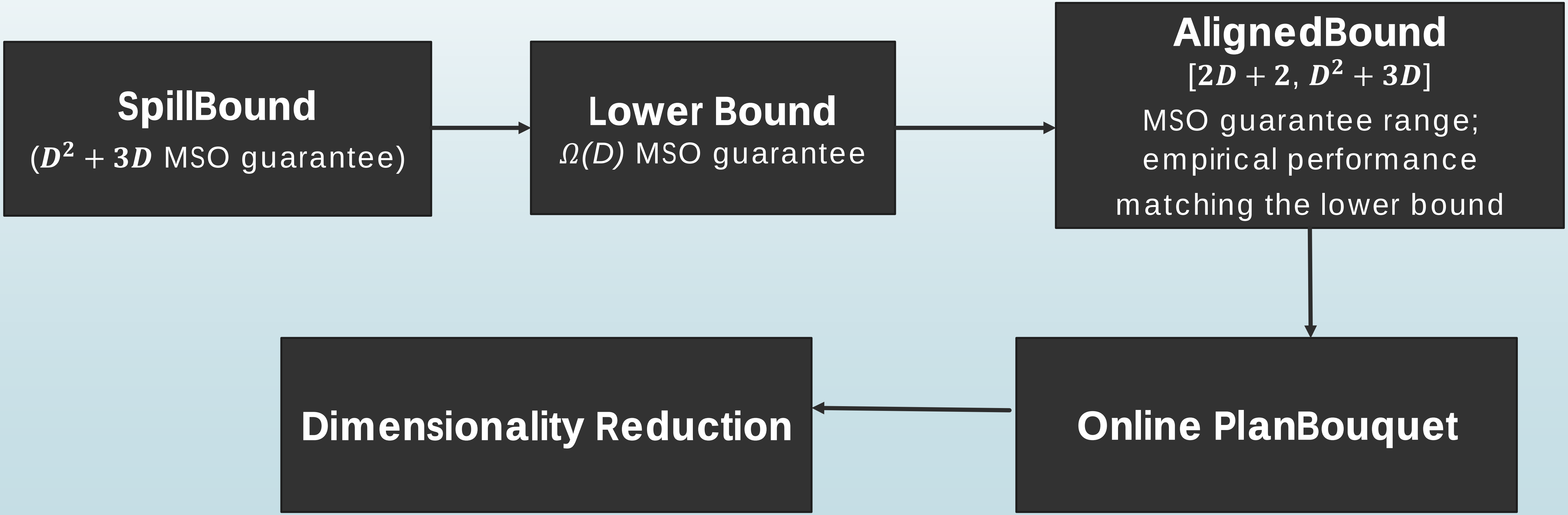
- **Error-prone predicates:**
  - ➢ predicate - X and predicate – Y
- **Error-prone Selectivity Space (ESS)**

The worst case impact on suboptimality across all estimation errors

Thesis Overview

Goal:

Propose query processing algorithms that provides MSO guarantee as close to 1 as possible



SpillBound

$(\boldsymbol{D^2 + 3D}$ MSO guarantee)

Lower Bound

$\Omega(D)$ MSO guarantee

AlignedBound

$[\boldsymbol{2D + 2,\ D^2 + 3D}]$

MSO guarantee range; empirical performance matching the lower bound

Online PlanBouquet

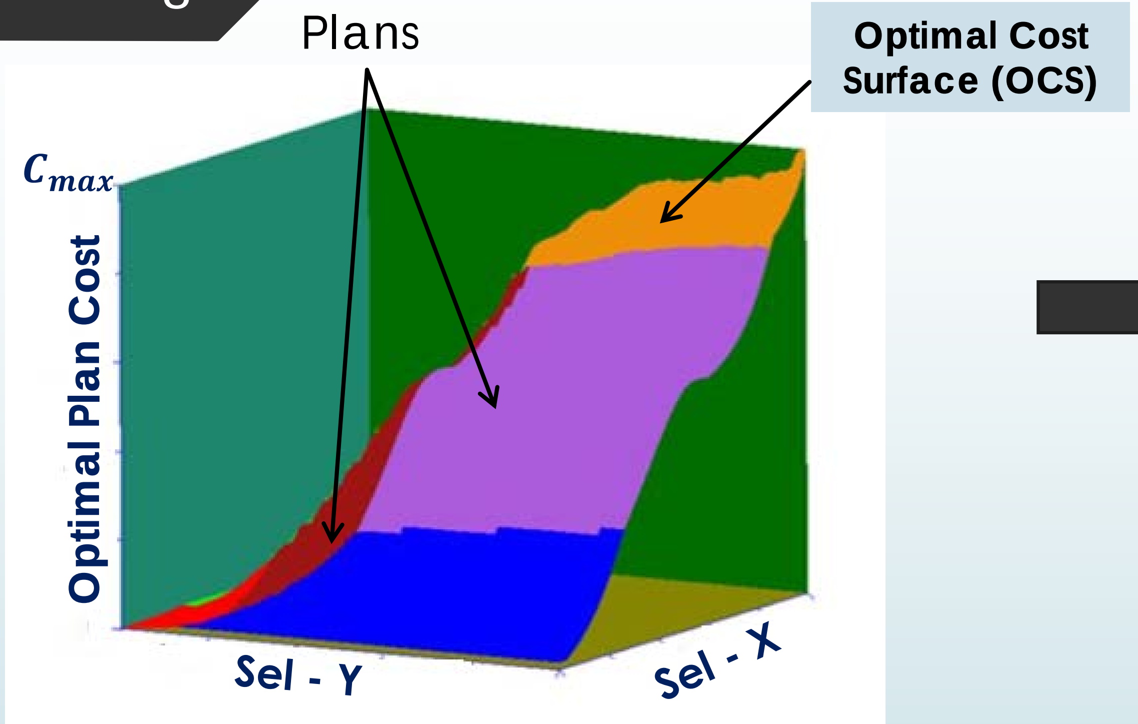
Dimensionality Reduction

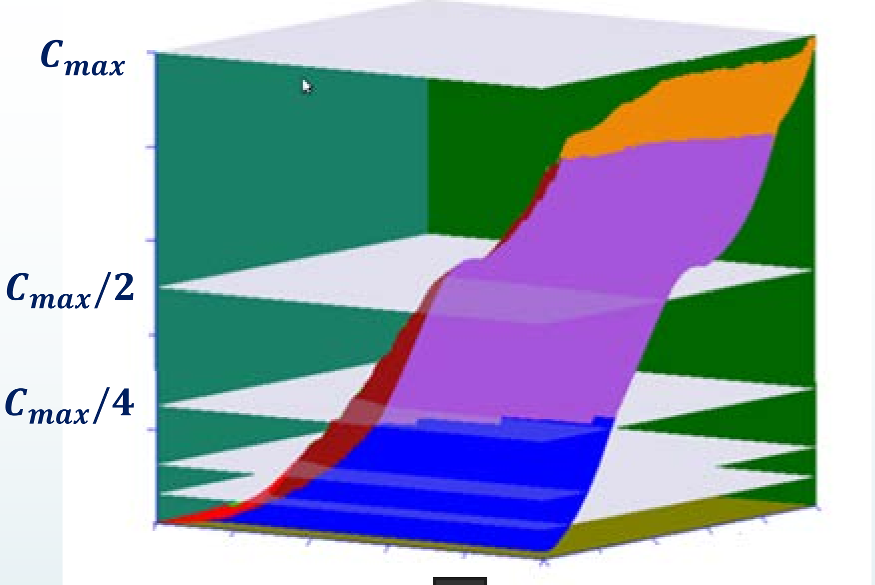
Our Proposed Algorithms (SpillBound/AlignedBound)

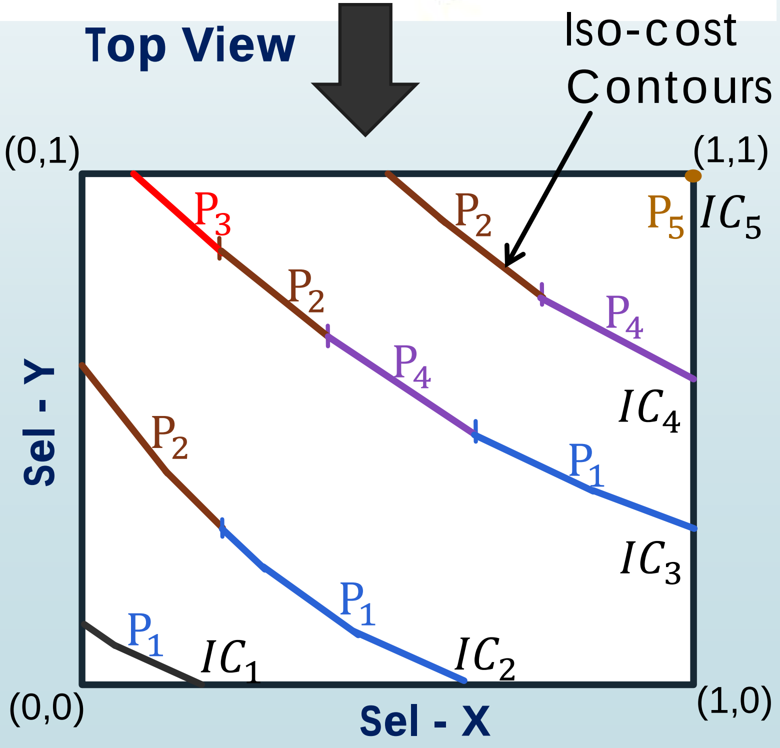
1. Compile-time Phase

2. Execution Phase

Compile-time Phase







Step 1: Construct ESS

Step 2: Cut OCS with isocost planes having **doubling cost**

Step 3: **PlanBouquet** - set of plans in the intersection of these cuts with OCS

***ρ* is the maximum number of plans in any contour.**

Results

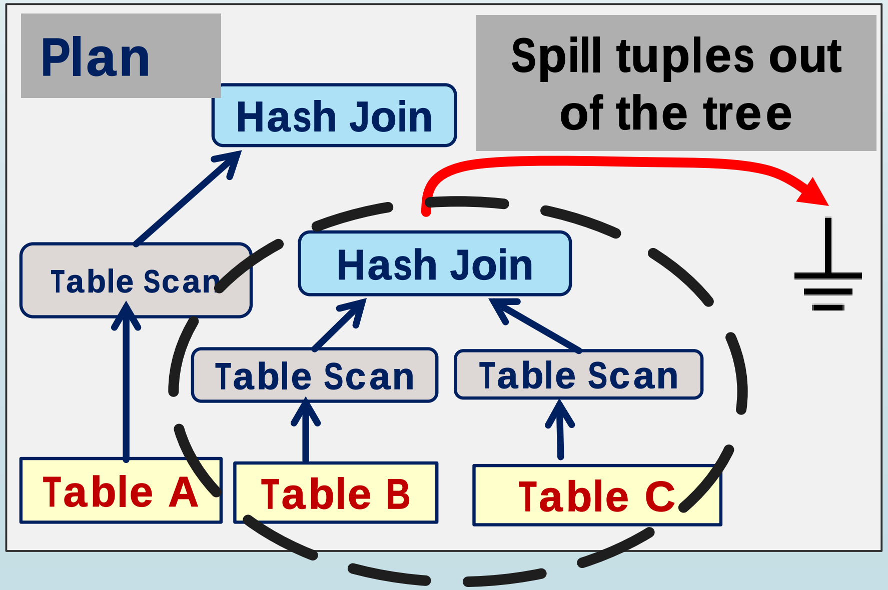
NAIVE: During execution phase, execute all plans in every contour until completion. Resulting in MSO guarantee of $\boldsymbol{4 * \rho}$

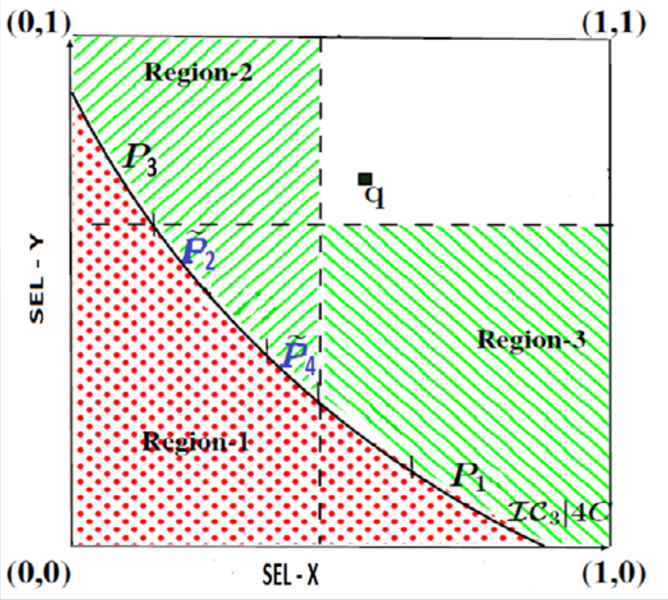
SpillBound: MSO guarantee of $\boldsymbol{D^2 + 3D}$

Spill-mode
execution of plans

Half-space pruning
of ESS

Execute at most D plans in every contour





More Results

Lower Bound: $\Omega(D)$ on the MSO guarantee

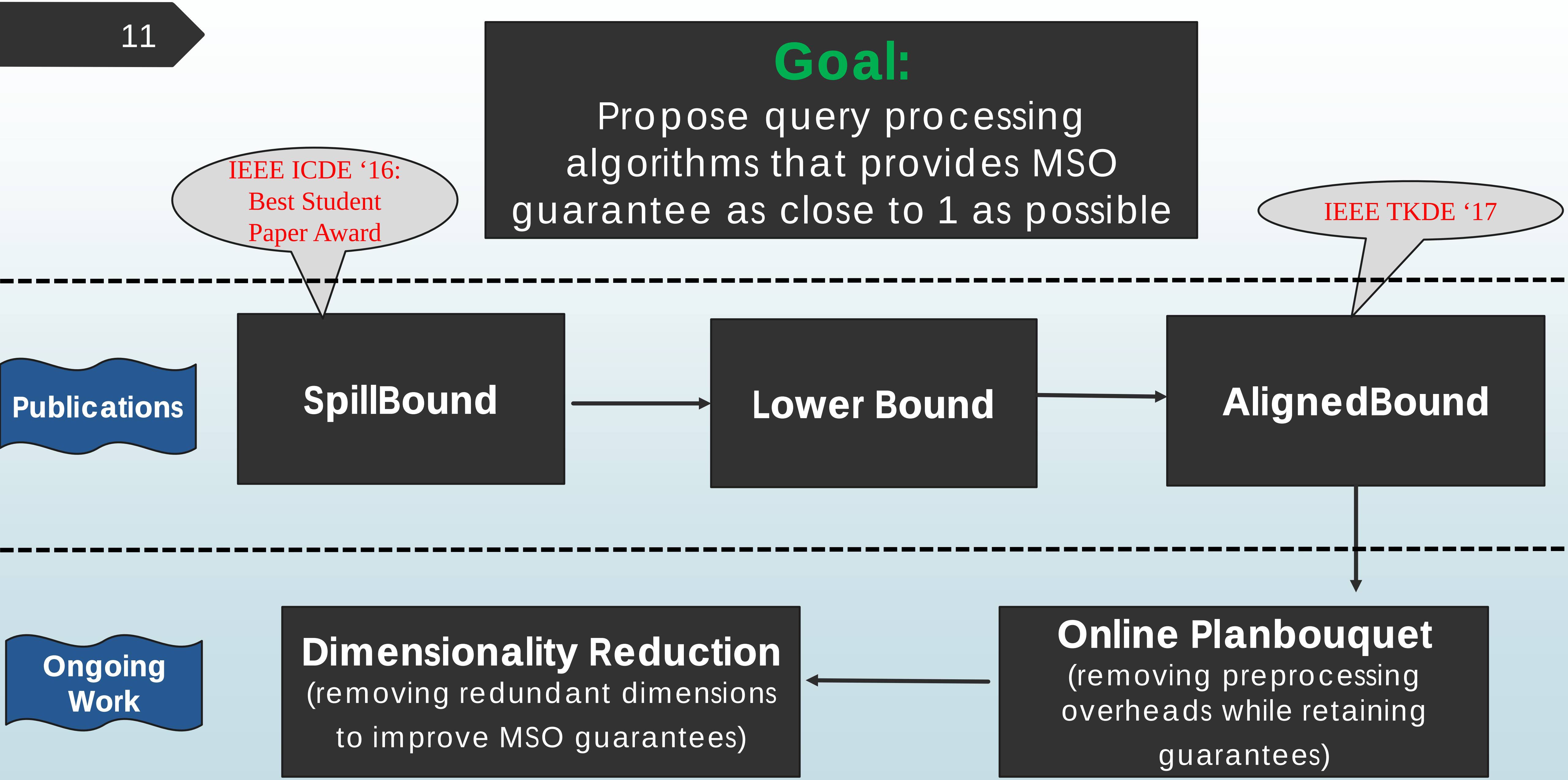
AlignedBound: During execution phase, for most queries, atmost $\mathbf{1}$ execution per contour, thus empirically matching MSO guarantee of $\boldsymbol{2D+2}$

Empirically: Evaluated on opensource PostgreSQL database engine, industrial strength benchmark dataset and queries.

MSO is less than around **10; significantly** improving over the state-of-the-art

Our proposed Algorithms collapse the enormous MSO (in millions) down to a *single order of magnitude*

Thesis Overview



Goal:

Propose query processing algorithms that provides MSO guarantee as close to 1 as possible

IEEE ICDE ‘16: Best Student Paper Award

IEEE TKDE ‘17

Publications

SpillBound

Lower Bound

AlignedBound

Ongoing Work

Online Planbouquet
(removing preprocessing overheads while retaining guarantees)

Dimensionality Reduction
(removing redundant dimensions to improve MSO guarantees)

Thank you!