PLAN SELECTION based on QUERY CLUSTERING

Antara Ghosh   Jignashu Parikh   Vibhuti Sengar
Jayant Haritsa
Computer Science & Automation
Indian Institute of Science
Bangalore, INDIA

THANKS TO

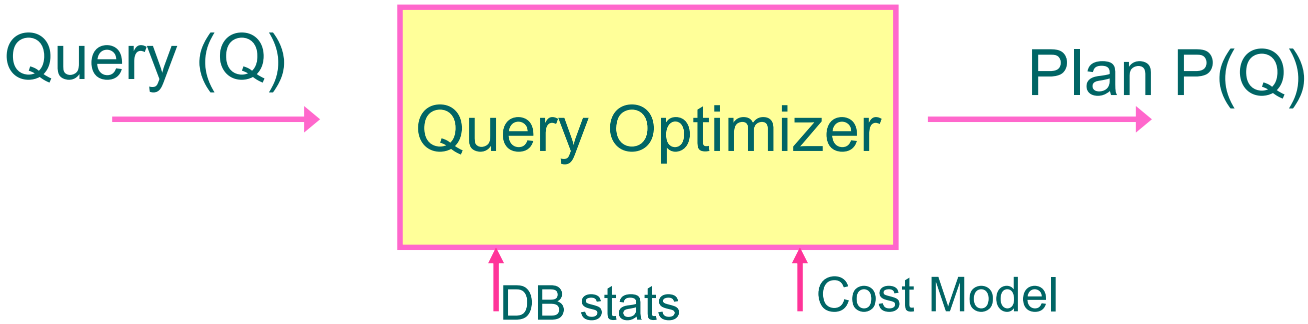
- Computer Society of India
- Indian Institute of Science
- IBM India Research Lab

TALK ORGANIZATION

- Overview
- Details
  - Query Feature Vector
  - Query Similarity
  - Query Clustering
- Performance Study
- Applicability of PLASTIC
- Closing Remarks

Query Plan Generation

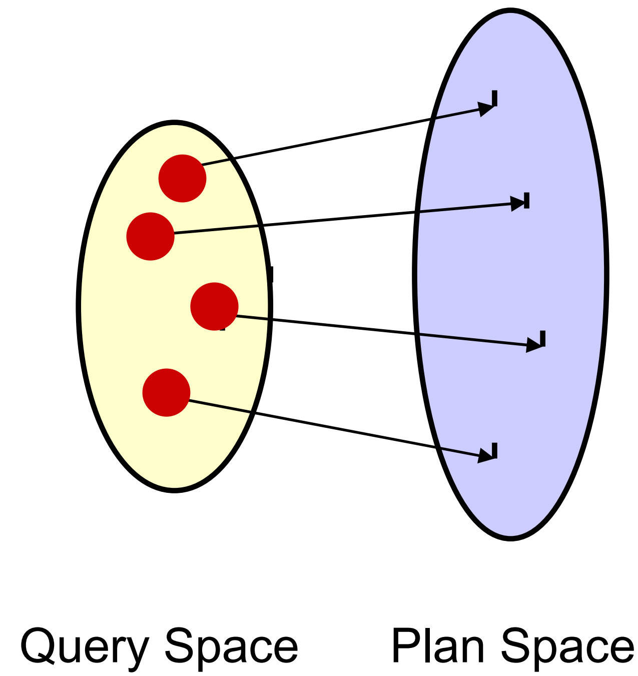
- Standard technique



- Computationally expensive since large number of plan candidates for queries
- Difference between right choice of plan and a sub-optimal choice can be enormous

Reduction of Optimization Overhead

- Plan Cacheing
  - Exact Match: Current commercial optimizers
    - E.g. Oracle's Stored_Outlines
    - Very limited scope
  - Similarity Match:
    PLASTIC (PLAn Selection Through Incremental Clustering)
    - Based on query clustering
    - Deals with plan templates, not plans (a plan template is the operator tree with variables for the operands – relations/attributes)
    - Facilitates plan sharing



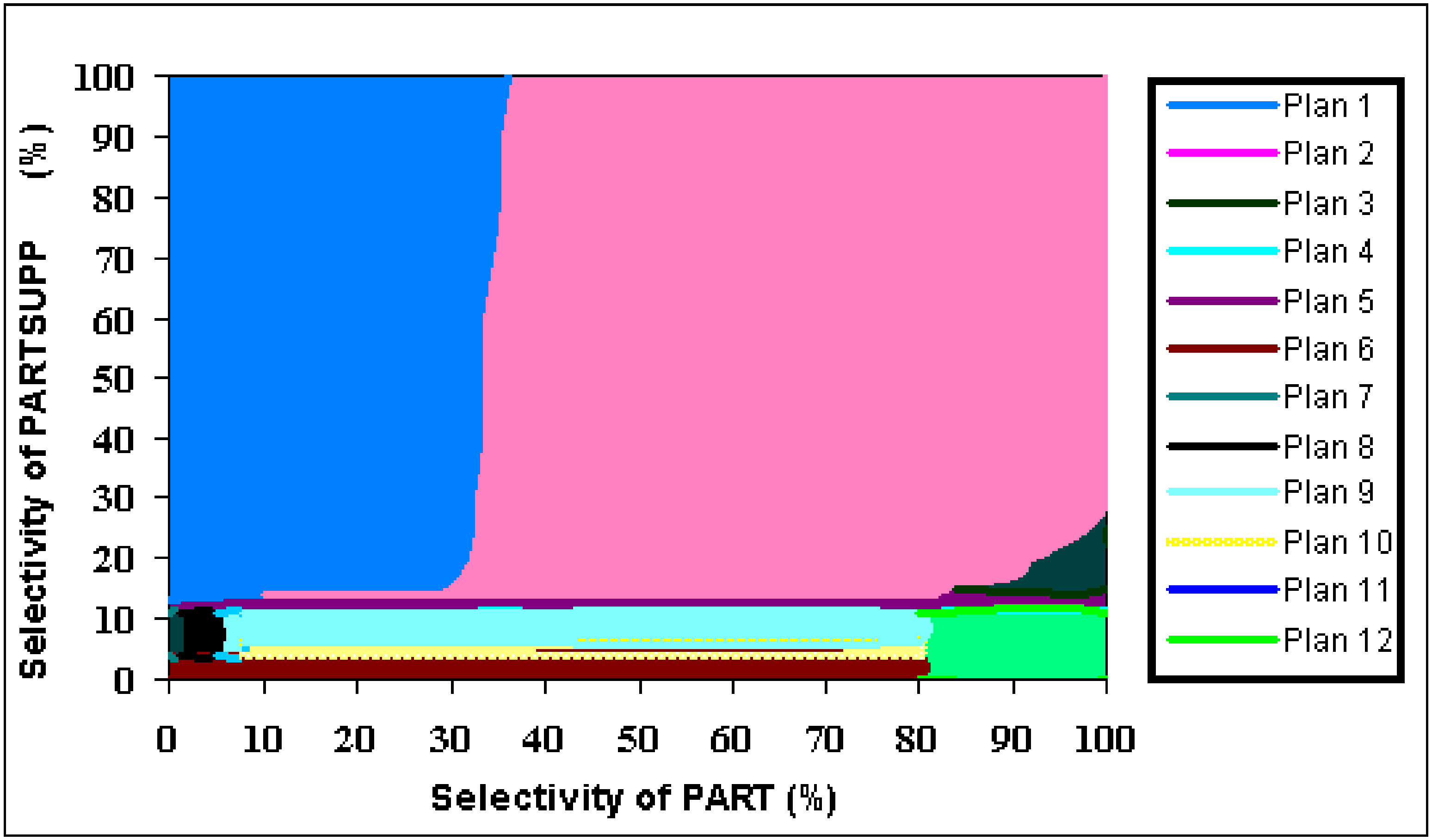
Major Benefits of Similarity Approach

- Significant improvements in optimization time due to broad-based plan reuse
- Improvements to the plan associated with the cluster representative (e.g. Plan Hints) automatically percolate to <u>all</u> cluster members
  – Makes it affordable to run optimizers at their highest optimization level since only cluster representatives have to be explicitly optimized
  – Reduces workload on DBAs
- Data updates are automatically reflected in change of plans due to changes in cluster assignments

Motivating Query

```
Select
    s_acctbal, s_name, n_name, p_partkey,
    p_mfgr, s_address, s_phone, s_comment
From
    part p, supplier s, partsupp ps, nation n, region r
Where
    p_partkey = ps_partkey and
    s_suppkey= ps_suppkey and
    p_size := :1 and p_type like :2 and
    s_nationkey = n_nationkey and
    n_regionkey = r_regionkey and
    r_name := :3 and ps_supplycost := :4
```

Associated Plan Diagram

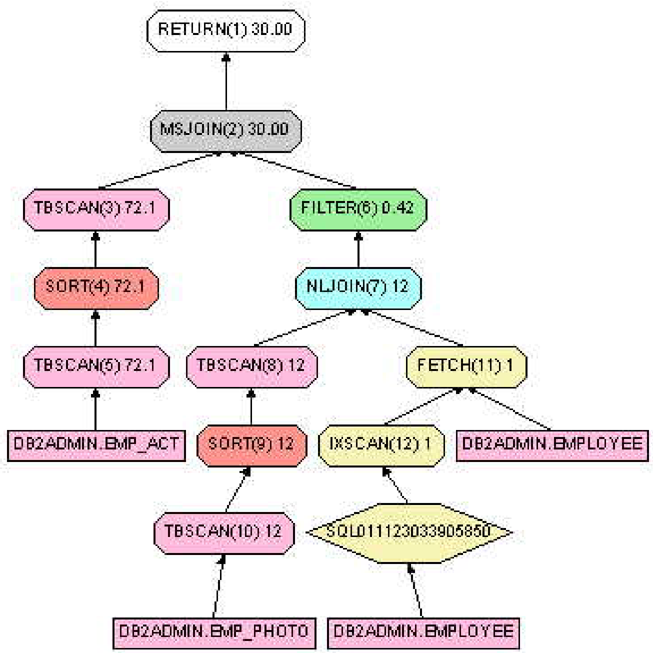


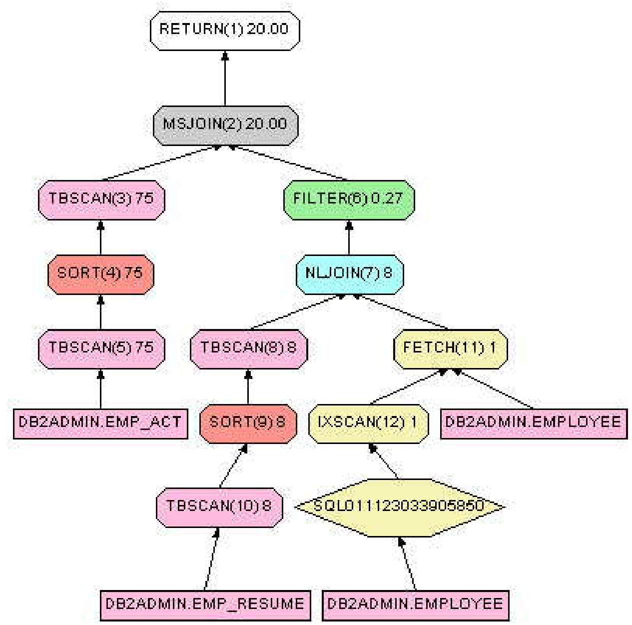
Note: 80% of space occupied by 20% of the Plans

Query Clustering (First Cut)

- Cluster Definition: Two queries belong to the same cluster if their plan templates are the same
- Problem: queries that are very different may have the same plan template
  - Results in heterogeneous clusters making it difficult to classify new queries

Different looking Queries- Similar Plan Templates





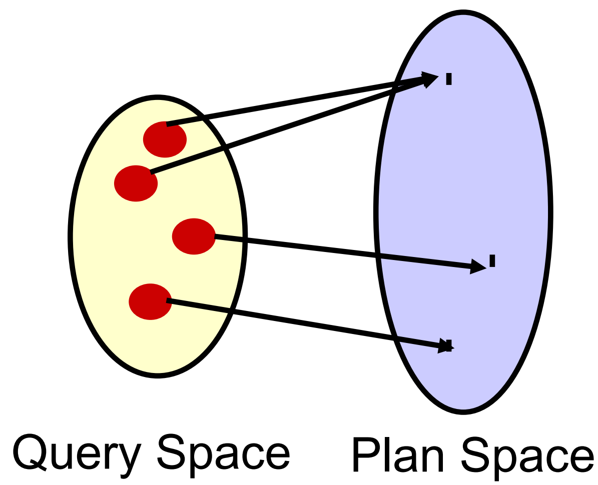
select * from employee as a, emp_act as b, emp_photo as c where a.empno=b.empno and b.empno=c.empno and a.empno>'000000' and b.empno<'000400‘ and c.empno between '000010' and '000390'	select a.firstname, a.lastname ,b.projno, c.resume from employee as a, emp_act as b, emp_resume as c where a.empno=b.empno and b.empno=c.empno

Observation

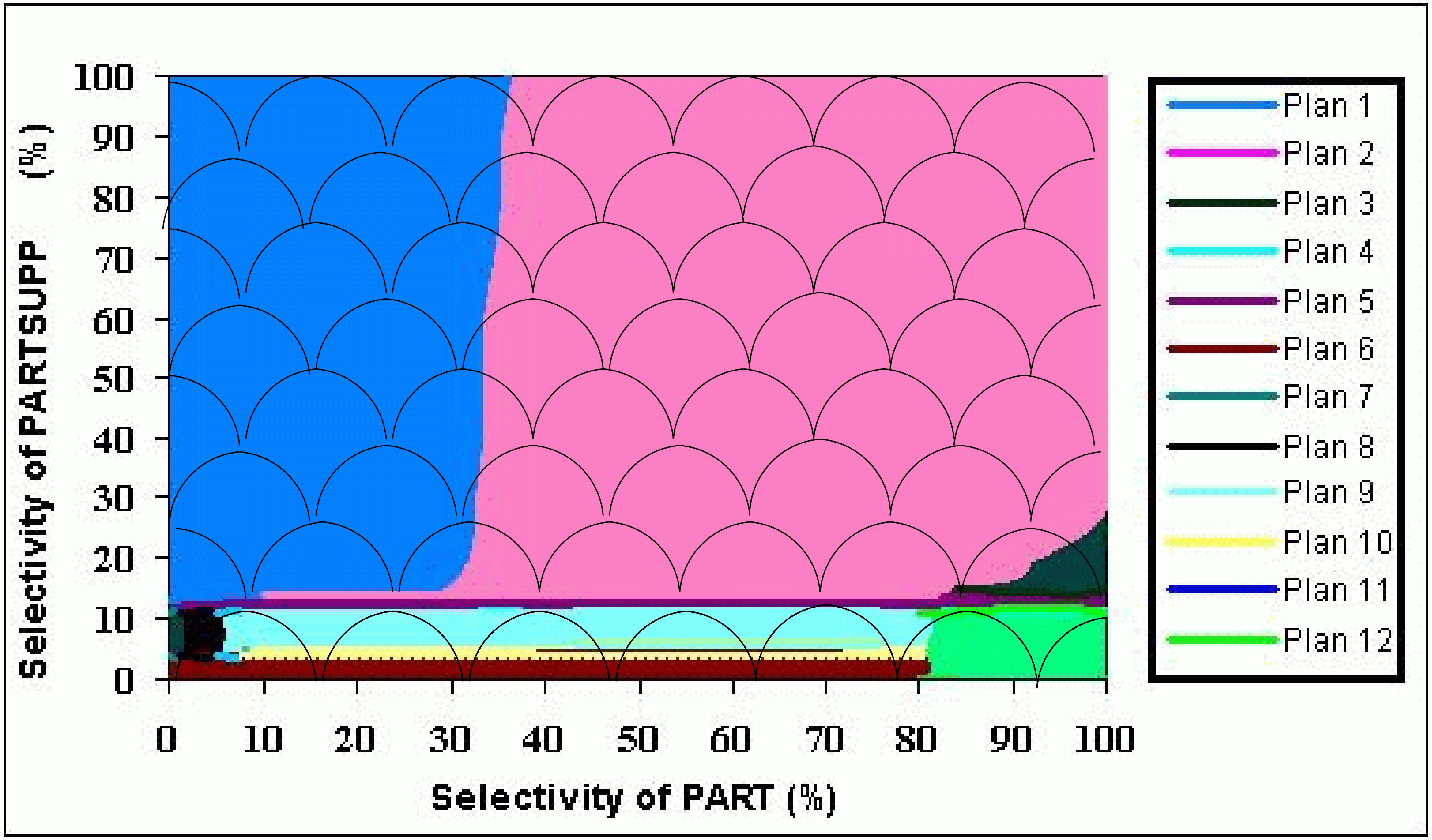
Clustering in Plan Space makes Classification in Query Space difficult …

Query Clustering: PLASTIC Approach

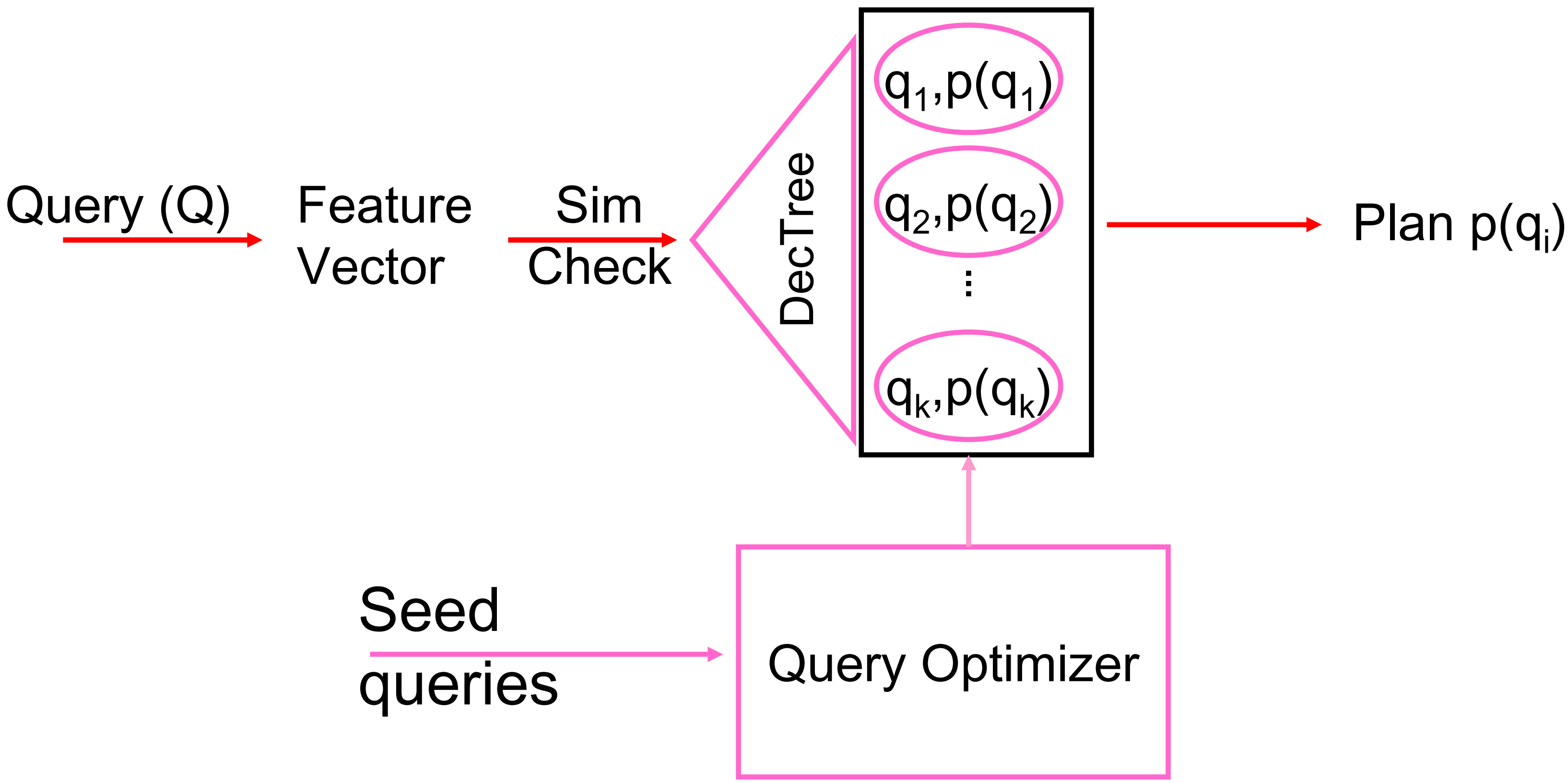
- Cluster Definition: Two queries belong to the same cluster if their Feature Vectors in Query Space are similar
  - Feature vectors have structural + statistical components (explained later)
  - Each cluster is defined by a single representative query
  - Clustering in Query Space may result in multiple clusters mapping to the same plan template



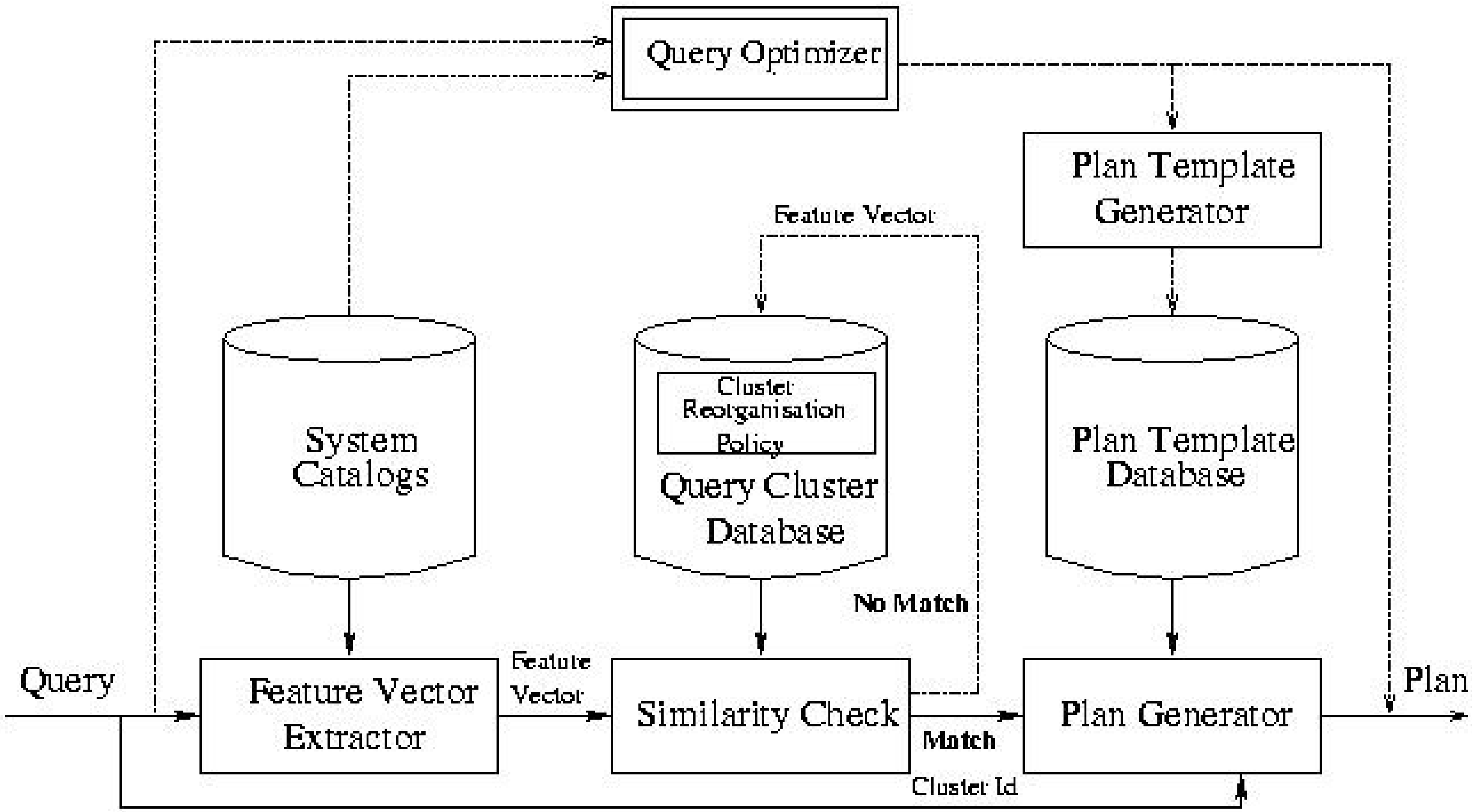
Cluster Diagram for Sample Query



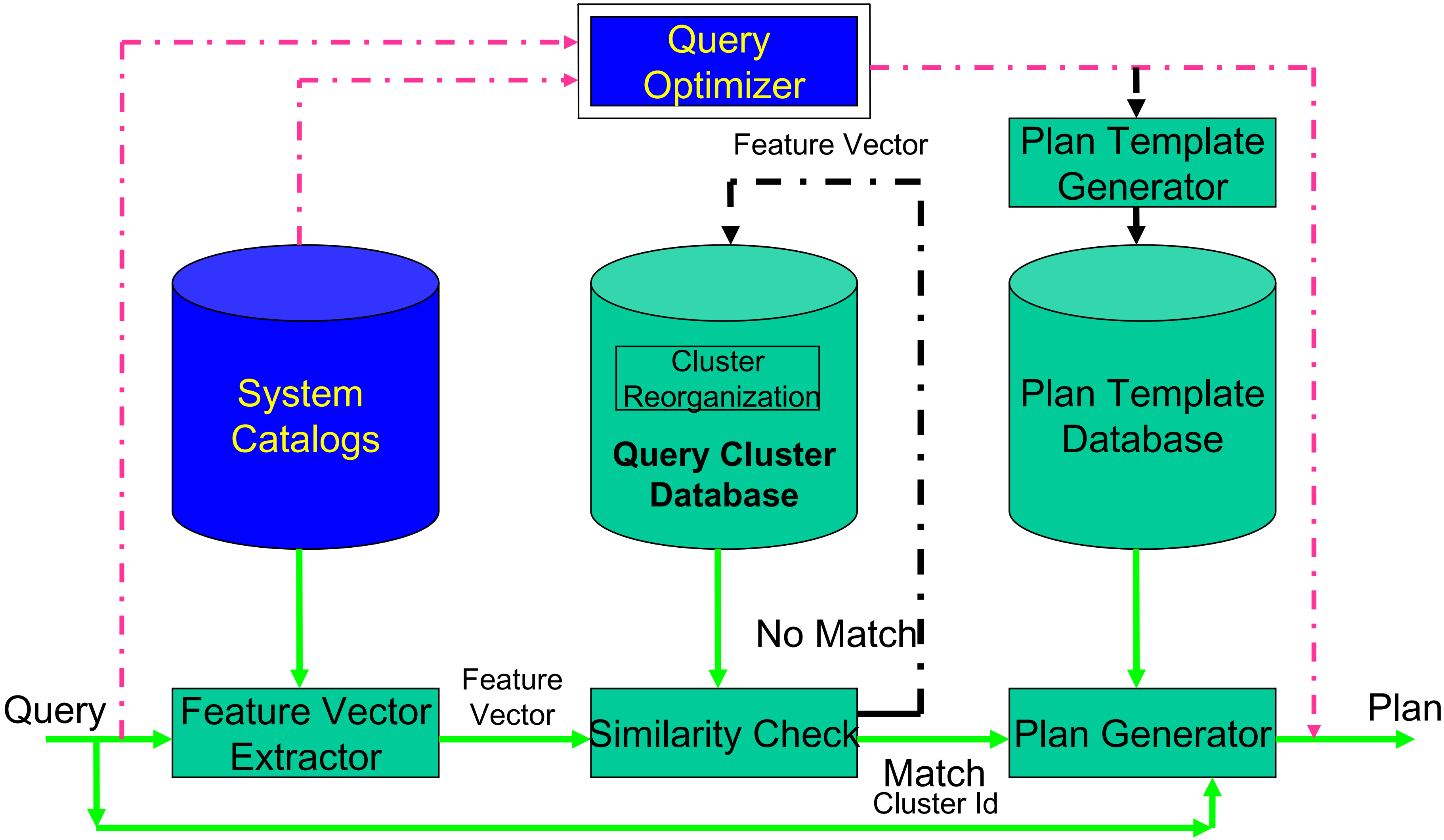
THE PLASTIC SCHEME



Proposed Optimizer Architecture



Proposed Optimizer Architecture



TALK ORGANIZATION

- Overview
- Details
  - Query Feature Vector
  - Query Similarity
  - Query Clustering
- Performance Study
- Applicability of PLASTIC
- Closing Remarks

Query Feature Vector

- Two components
  - Structural Features
    - Determined from the query and DB schema catalogs
  - Statistical Features
    - Derived from DB statistics module
- Feature selection based on
  - study of query optimization literature
  - characteristics of plans generated by commercial optimizers
  - not involving computation of any plan specific information
  - not requiring additional inputs beyond those already available to the optimizer

Structural Features (per Table)

- Degree of the Table (DT)
  - No. of Join Predicates in which the table is involved
- Join Predicate Index Counts (JIC)
  - JIC[k] = Number of join predicates (in which the table participates) having k indexed attributes in the join predicate
    k = 0, 1 or 2
- Predicate Counts of a Table (PC)
  - Count of SARGable and Non-SARGable predicates in which the table is involved
- Index Flag of a Table (IF)
  - Set if all the selection attributes and projections on that table can be evaluated through indexes only (i.e. Required information can be obtained solely from the indexes without accessing the actual data tables)

Statistical Features (per Table)

- Table Size (TS)
  - Total size (disk occupancy) of the table
- Effective Table Size (ETS)
  - Calculated by estimating the impact of pushing down all the projections and selections on the table in the query

Example Feature Vector

Select A.a1,B.b1
from A, B
Where A.a1 = B.b2 and
A.a2 >100 and
B.b3 <25

- Combined index on (a1,a2) of Table A
- Index on b2 of Table B
- A2 > 100 has selectivity 0.5
- B3 < 25 has selectivity .005

Feature	Table A	Table B
DT	1	1
IF	1	0
PCsarg	1	1
PCnsarg	0	0
JIC	{0, 0,1}	{0, 0,1}
TS	400000	100000
ETS	200000	5000

TALK ORGANIZATION

- Overview
- Details
  - Query Feature Vector
  - Query Similarity
  - Query Clustering
- Performance Study
- Applicability of PLASTIC
- Closing Remarks

Step 1: Structural Comparison

- Equality Checks based on Aggregate Structural Features like
  - Number of tables participating in the query
    - Obvious
  - Degree Sequence (Vector of Table Degrees)
    - Should be same else the plan templates will perforce be different
  - Sum of Index flags
    - Data gathering differs based on flag setting

Step 2: Statistical Similarity (Mapping Tables)

- Query 1 has R1 and R2
- Query 2 has S1 and S2
- Could map R1 to S1 and R2 to S2 or R1 to S2 and R2 to S1

- N! possibilities
  - Reduced by grouping tables with identical structural features and considering only intra-group mappings

Table Distance Function

$$dist_{ij}(T_1^i, T_2^j) = \frac{w_1 * (TS_1^i - TS_2^j) + w_2 * (ETS_1^i - ETS_2^j)}{max(TS_1^i, TS_2^j)},$$

- Tables are numbered according to mapping
- TS_k^i = Table size of i^{th} Table of Query k
- ETS_k^i = Estimated Table size of i^{th} Table of Query k
- w_1 and w_2 are weights
  - w_1, w_2 ε [0,1] and $w_2 = 1 - w_1$
- Normalization ensures $dist_{ij}$ is in $(0,1)$
- After all mappings (within the group) are evaluated the mapping with the *mindist (*minimum aggregate value of *dist*) is selected

Query Distance Function

- Let $mindist_g$ be the distance between the g^{th} group mapping between two queries

$$TotalDist = \sum_{g \in G} mindist_g$$

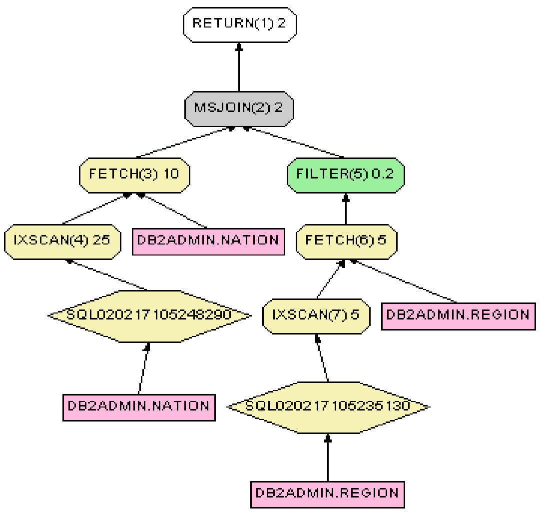
- Queries are similar only if *TotalDist* is less than a predefined *Threshold*

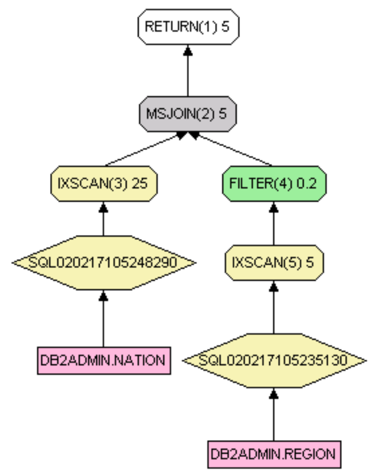
Distance Function Design

- Our investigation of plan choices by optimizers indicates that, given structural compatibility, TS and ETS play a crucial role in determining the plan choices
- Choices of w_1 and w_2 determine the relative impacts of TS and ETS
- Threshold determines the stretch of individual clusters. Lower threshold values result in
  - smaller percentage of error-causing clusters (i.e. clusters straddling plan boundaries in the plan diagram),
  - larger number of clusters increases the search space for classification

Similarity Examples

Q1: select * from nation, region
where
n_nationkey=r_regionkey
Q2: select n_nationkey
from nation, region
where n_nationkey =
r_regionkey
Q3: select n_comment, r_comment
from nation, region





- DB2 produces different plans for Q1 and Q2 although they *look* similar!!
- Same plan for both these queries Q1 and Q3 although they seem different!

TALK ORGANIZATION

- Overview
- Details
  - Query Feature Vector
  - Query Similarity
  - Query Clustering
- Performance Study
- Applicability of PLASTIC
- Closing Remarks

Leader Algorithm [Hartigan 1975]

- Algorithm:
  - Match a query with existing cluster leaders and if no match is found, make the query a new leader.
- Leader is an incremental algorithm and we therefore use it for classification also
- Classification becomes slow if large number of clusters
  - Inducing a decision tree on the clusters reduces this problem substantially

TALK ORGANIZATION

- Overview
- Details
  - Query Feature Vector
  - Query Similarity
  - Query Clustering
- Performance Study
- Applicability of PLASTIC
- Closing Remarks

Metrics

- Prediction Efficiency
  - Time required for predictions
- Prediction Accuracy
  - How often do we guess right?
- Prediction Risk Factor
  - Penalty for wrong choices
- Plan Cache Space Overhead
  - Storage required by query representatives and their plans

Testbed

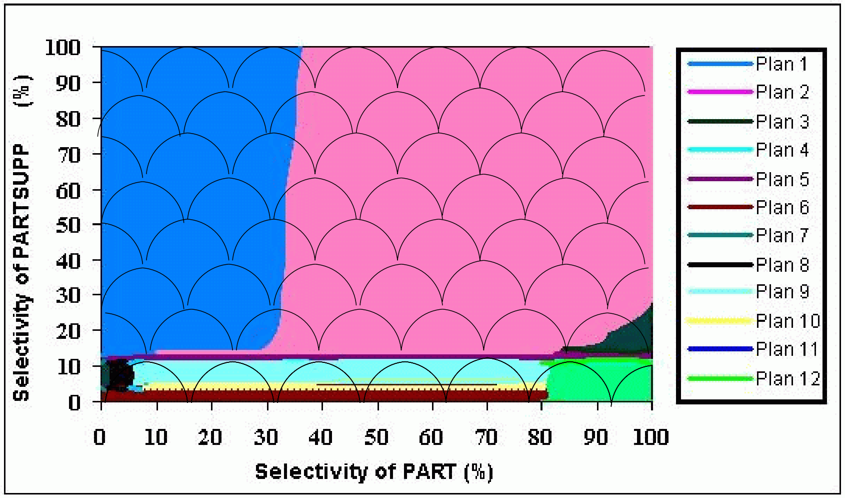
- DBMS: DB2 Universal Database Version 7
  - Default optimization class of DB2 (level 5)
- PLATFORM: P-III / Windows 2000 machine
- DATABASE: TPC-H database on scale 1 (1GB)
- QUERIES: Simplified (pure SPJ) versions of TPC-H Queries
- ASSUMPTIONS
  - Queries are uniformly distributed over the selectivity space (limited to 2D)
  - Static resource configuration

Clustering on Example Query (Q2')

Select
- s_acctbal, s_name, n_name, p_partkey,
- p_mfgr, s_address, s_phone, s_comment

From
- part p, supplier s, partsupp ps, nation n, region r

Where
- p_partkey = ps_partkey and
- s_suppkey= ps_suppkey and
- p_size := :1 and p_type like :2 and
- s_nationkey = n_nationkey and
- n_regionkey = r_regionkey and
- r_name := :3 and ps_supplycost := :4



65 Clusters Generated with Threshold value of 0.01
$W_1 = 0.7$ and $W_2 = 0.3$

P-DB2 Performance on Example Query

Metric	DB2	P-DB2 Leader	P-DB2 Decision Tree
Accuracy	100%	90.76%	88.8%
Efficiency	0.1s	0.004s	0.00025 s
Space	-	1.97KB	3.96KB

Risk Factor

Error Case	DB2 Cost timeron	P-DB2 Cost timeron	Risk Factor (%)
1	261209	266260	1.9
2	241054	246000	2
3	173913	188684	1.1
4	158577	158681	0
5	161814	159078	-0.02

Summary of Results

- For SPJ queries and static resource availability,
  PLASTIC provides x10 improvement in query optimization time with 90% accuracy in correct plan prediction
- Mistakes are not expensive since they occur on plan boundaries (< 10% error penalty)
- Space overhead is miniscule

TALK ORGANIZATION

- Overview
- Details
  - Query Feature Vector
  - Query Similarity
  - Query Clustering
- Performance Study
- Applicability of PLASTIC
- Closing Remarks

Inter-query Plan Sharing

- PLASTIC works across queries with
  - Different Selection Predicates
  - Different Projection Attributes
  - Different Join Attributes
  - Different Tables

- PLASTIC broadens the scope of plan sharing beyond mere syntactic matching

Example (Different Join Attributes)

```
Select l_extendedprice, l_discount
From customer, orders, lineitem, supplier, nation, region
Where
        c_custkey = o_orderkey
  and L_COMMITDATE = O_ORDERDATE
  and l_suppkey = s_suppkey
  and c_nationkey = s_nationkey
  and s_nationkey = n_nationkey
  and n_regionkey = r_regionkey
  and r_name = 'AFRICA'
  and o_orderdate >= date ('1997-01-01')
  and year(o_orderdate) < (year ('1997-01-01')+1);
```

Example (Different Join Attributes)

```
Select l_extendedprice, l_discount
From customer, orders, lineitem, supplier, nation, region
Where
        c_custkey = o_orderkey
   and L_COMMITDATE = O_ORDERDATE
   and l_suppkey = s_suppkey
   and c_nationkey = s_nationkey
   and s_nationkey = n_nationkey
   and n_regionkey = r_regionkey
   and r_name = 'AFRICA'
   and o_orderdate >= date ('1997-01-01')
   and year(o_orderdate) < (year ('1997-01-01')+1);
```

Example (Different Join Attributes)

```
Select l_extendedprice, l_discount
From customer, orders, lineitem, supplier, nation, region
Where
        c_custkey = o_orderkey
    and L_SHIPDATE = O_ORDERDATE
    and l_suppkey = s_suppkey
    and c_nationkey = s_nationkey
    and s_nationkey = n_nationkey
    and n_regionkey = r_regionkey
    and r_name = 'AFRICA'
    and o_orderdate >= date ('1997-01-01')
    and year(o_orderdate) < (year ('1997-01-01')+1);
```

- No change in plan generated by DB2
- PLASTIC correctly identifies this since the Join Index Counts in both queries remain same

TALK ORGANIZATION

- Overview
- Details
  - Query Feature Vector
  - Query Similarity
  - Query Clustering
- Performance Study
- Applicability of PLASTIC
- Closing Remarks

Future Work

- Need to extend PLASTIC to
  - handle correlated nested queries, as well as GROUP BY and HAVING clauses
  - handle changes in the system resource availability between training and operational stages
- Variable-sized clusters
  - Error varies with table selectivities
  - Cluster sizes should thus be made sensitive to selectivities
- Automated parameter settings (w_1, w_2 and T)

Comparison with Related Work

- Unlike MQO
  - No attempt to *optimize* Queries
  - Instead, we aim to *reuse* previous optimization results
  - PLASTIC's plan selection is not specific to a temporal window of queries
- Unlike PQO
  - We do not try to characterize the plan space for a given query
  - Our approach extends to *sharing* of plans across similar queries

Take Away

- PLASTIC significantly increases the scope of “plan recycling”, thereby substantially improving the utility of plan cacheing
- A query optimizer’s best friend