CS3201/2: Software Engineering Project

Query Optimization

By: Dr. Bimlesh Wadhwa

Query Optimization Problem

- Describe how query evaluator would evaluate the following query. Subsequently, identify a few strategies to optimize the evaluation.

  assign a1, a2, a3; statement s1, s2, s3; variable v1, v2, v3;

  Select <s1, s2, v2> such that Uses (s3, v1) and Modifies (s3, “x”) and Follows (s1, s2) and Parent (s3, s1) and Uses (s2, v1)
  such that Uses (5, “y”) and Follows (3, 4) pattern a1 (v2, _“x+y”_)
  such that Affects (a1, a2) with a2.stmt# = 20
  such that Modifies (a3, v3) pattern a3 (“z”, _)

- Assume the following list of answers for selected clauses:

Uses (s3, v1)	
s3	**v1**
5	y
5	z
7	x
8	y
2	y

Modifies (s3, “x”)
s3
6
5
7

Follows (s1,s2)	
s1	**s2**
6	20
3	4
10	15
8	11

Parent (s3,s1)	
s3	**s1**
5	6
5	8
2	7
2	3
7	6

Uses (s2,v1)	
s2	**v1**
11	y
20	x
4	x
20	y
11	z
10	z

Simplistic Approach

- Initialize an empty intermediate result table
- Get the results for each clause one by one and merge them into the intermediate result table
- Produce the final results based on the intermediate result table and the select clause

Query

assign a1, a2, a3; statement s1, s2, s3; variable v1, v2, v3;

Select <s1, s2, v2> such that Uses (s3, v1) and Modifies (s3, “x”) and Follows (s1, s2) and Parent (s3, s1) and Uses (s2, v1)

such that Uses (5, “y”) and Follows (3, 4) pattern a1 (v2, _”x+y”_)

such that Affects (a1, a2) with a2.stmt# = 20

such that Modifies (a3, v3) pattern a3 (“z”, _)

Uses (s3, v1)	
s3	**v1**
5	y
5	z
7	x
8	y
2	y

Modifies (s3, “x”)
s3
6
5
7

Follows (s1,s2)	
s1	**s2**
6	20
3	4
10	15
8	11

Parent (s3,s1)	
s3	**s1**
5	6
5	8
2	7
2	3
7	6

Uses (s2,v1)	
s2	**v1**
11	y
20	x
4	x
20	y
11	z
10	z

Uses (s3, v1)	
s3	**v1**
5	y
5	z
7	x
8	y
2	y

Modifies (s3, “x”)
s3
6
5
7

Follows (s1, s2)	
s1	**s2**
6	20
3	4
10	15
8	11

Parent (s3, s1)	
s3	**s1**
5	6
5	8
2	7
2	3
7	6

Uses (s2, v1)	
s2	**v1**
11	y
20	x
4	x
20	y
11	z
10	z

Intermediate result table

Uses (s3, v1)	
s3	**v1**
5	y
5	z
7	x
8	y
2	y

Modifies (s3, “x”)
s3
6
5
7

Follows (s1, s2)	
s1	**s2**
6	20
3	4
10	15
8	11

Parent (s3, s1)	
s3	**s1**
5	6
5	8
2	7
2	3
7	6

Uses (s2, v1)	
s2	**v1**
11	y
20	x
4	x
20	y
11	z
10	z

Intermediate result table

s3	v1
5	y
5	z
7	x
8	y
2	y

Uses (s3, v1)	
s3	**v1**
5	y
5	z
7	x
8	y
2	y

Modifies (s3, “x”)
s3
6
5
7

Follows (s1, s2)	
s1	**s2**
6	20
3	4
10	15
8	11

Parent (s3, s1)	
s3	**s1**
5	6
5	8
2	7
2	3
7	6

Uses (s2, v1)	
s2	**v1**
11	y
20	x
4	x
20	y
11	z
10	z

Intermediate result table

s3	v1
5	y
5	z
7	x
~~8~~	~~y~~
~~2~~	~~y~~

Uses (s3, v1)	
s3	**v1**
5	y
5	z
7	x
8	y
2	y

Modifies (s3, “x”)
s3
6
5
7

Follows (s1, s2)	
s1	**s2**
6	20
3	4
10	15
8	11

Parent (s3, s1)	
s3	**s1**
5	6
5	8
2	7
2	3
7	6

Uses (s2, v1)	
s2	**v1**
11	y
20	x
4	x
20	y
11	z
10	z

Intermediate result table

s3	v1	s1	s2
5	y	6	20
5	z	6	20
7	x	6	20
5	y	3	4
5	z	3	4
7	x	3	4
5	y	10	15
5	z	10	15
7	x	10	15
5	y	8	11
5	z	8	11
7	x	8	11

Uses (s3, v1)	
s3	**v1**
5	y
5	z
7	x
8	y
2	y

Modifies (s3, “x”)
s3
6
5
7

Follows (s1, s2)	
s1	**s2**
6	20
3	4
10	15
8	11

Parent (s3, s1)	
s3	**s1**
5	6
5	8
2	7
2	3
7	6

Uses (s2, v1)	
s2	**v1**
11	y
20	x
4	x
20	y
11	z
10	z

Intermediate result table

s3	v1	s1	s2
5	y	6	20
5	z	6	20
7	x	6	20
~~5~~	~~y~~	~~3~~	~~4~~
~~5~~	~~z~~	~~3~~	~~4~~
~~7~~	~~x~~	~~3~~	~~4~~
~~5~~	~~y~~	~~10~~	~~15~~
~~5~~	~~z~~	~~10~~	~~15~~
~~7~~	~~x~~	~~10~~	~~15~~
5	y	8	11
5	z	8	11
~~7~~	~~x~~	~~8~~	~~11~~

Uses (s3, v1)	
s3	**v1**
5	y
5	z
7	x
8	y
2	y

Modifies (s3, “x”)
s3
6
5
7

Follows (s1, s2)	
s1	**s2**
6	20
3	4
10	15
8	11

Parent (s3, s1)	
s3	**s1**
5	6
5	8
2	7
2	3
7	6

Uses (s2, v1)	
s2	**v1**
11	y
20	x
4	x
20	y
11	z
10	z

Intermediate result table

s3	v1	s1	s2
5	y	6	20
~~5~~	~~z~~	~~6~~	~~20~~
7	x	6	20
5	y	8	11
5	z	8	11

...

Query

assign a1, a2, a3; statement s1, s2, s3; variable v1, v2, v3;

Select <s1, s2, v2> such that Uses (s3, v1) and Modifies (s3, “x”) and Follows (s1, s2) and Parent (s3, s1) and Uses (s2, v1)

such that Uses (5, “y”) and Follows (3, 4) pattern a1 (v2, _”x+y”_)

such that Affects (a1, a2) with a2.stmt# = 20

such that Modifies (a3, v3) pattern a3 (“z”, _)

Uses(s3, v1)	
s3	**v1**
5	y
5	z
7	x
8	y
2	y

Modifies (s3, “x”)
s3
6
5
7

Follows(s1,s2)	
s1	**s2**
6	20
3	4
10	15
8	11

Parent(s3,s1)	
s3	**s1**
5	6
5	8
2	7
2	3
7	6

Uses(s2,v1)	
s2	**v1**
11	y
20	x
4	x
20	y
11	z
10	z

Query

assign a1, a2, a3; statement s1, s2, s3; variable v1, v2, v3;

TRUE/FALSE	• such that Follows (3, 4) and Uses (5, “y”)
<a3, v3> (TRUE/FALSE)	• pattern a3 (“z”, _) such that Modifies (a3, v3)
<s3, v1, s1, s2>	• such that Modifies (s3, “x”) and Parent (s3, s1) and Follows (s1, s2) and Uses (s3, v1) and Uses (s2, v1)
<a1, v2, a2>	• with a2.stmt# = 20 such that Affects (a1, a2) pattern a1 (v2, _“x+y”_)
Select <s1, s2, v2>	• Compute tuples <s1, s2, v2> (not <s3, v1, s1, s2, a1, v2, a2>

Query Optimizations Steps

1. Divide the clauses into multiple groups
   - Clauses without synonyms
   - Clauses with connected synonyms should be in the same group
2. Sort groups for evaluation
   - Start with clauses without synonyms
   - Prioritize groups that do not return results in Select

Query Optimizations Steps

3. Sort clauses inside the group
   - Prioritize clauses with one constant and one synonym
   - Prioritize clauses with less number of results: Follows, Modifies, etc.
   - Sort clauses such that at least one synonym has been computed in a previous clause
   - Prioritize with-clauses – more restrictive than such that clauses
   - Evaluating pattern-clauses – similar to any such that clause
   - Push Affects/* clauses on the last positions in a group

Query Optimizations Steps

4. Evaluation for each group
   - Make use of synonym values computed in one clause to compute results in a connected clause
   - Maintain the intermediate results for each group
5. Evaluation of Select
   - Join the intermediate results and / or select part of intermediate results to obtain final results