

Uni Freiburg, Web Science Group
Prof. Peter Fischer
Systems Infrastructure for Data Science - Winter 2013/14

Exercise Sheet #8: Distributed Query Processing

December 18, 2013

1 Data Localization

Consider that the relation *Reviewers* is horizontally fragmented as follows:

$$\begin{aligned} Reviewers_1 &= \sigma_{reviewer_id \leq '20000'}(Reviewers) \\ Reviewers_2 &= \sigma_{reviewer_id > '20000'}(Reviewers) \end{aligned}$$

Now, consider a derived horizontal fragmentation of relation *Movie_Reviews*:

$$\begin{aligned} Movie_Reviews_1 &= Movie_Reviews \bowtie_{reviewer_id} Reviewers_1 \\ Movie_Reviews_2 &= Movie_Reviews \bowtie_{reviewer_id} Reviewers_2 \end{aligned}$$

Furthermore, the relation *Movies* is vertically fragmented as:

$$\begin{aligned} Movies_1 &= \Pi_{movie_id, title, release_year}(Movies) \\ Movies_2 &= \Pi_{movie_id, star_rating, era_id}(Movies) \end{aligned}$$

Transform the following query into a reduced query on fragments.

```
SELECT m.title
FROM Movies m, Reviewers r, Movie_Reviews mr
WHERE m.movie_id = mr.movie_id AND r.reviewer_id = mr.reviewer_id
      AND r.name = 'Cagri'
```

2 Query Optimization

- A. Consider a join among tables *Reviews*, *Movie_Reviews* and *Movies* from the previous example. Figure 1 shows both the join graph and the distribution onto three sites.

$$(Movies \bowtie_{movie_id} Movie_Reviews \bowtie_{reviewer_id} Reviewers)$$

- (i) Given the following information: $size(Movies)=100$, $size(Movie_Reviews)=200$, $size(Reviewers)=300$, $size(Movies \bowtie Movie_Reviews)=300$, $size(Movie_Reviews \bowtie Reviewers) = 200$, describe several alternatives for building a join ordering program.
- (ii) What is the optimal ordering that minimizes query response time (consider communication only)?

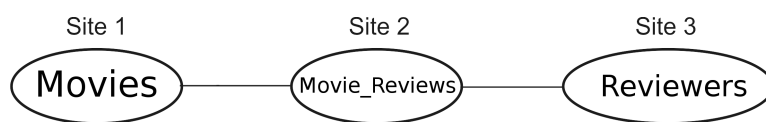


Figure 1: Join graph