



a) Total Execution Time

b) Average Tuple Latency

c) Cumulative Throughput over 5 min

Figure 1: Experimental results.

1 Experimental Evaluation

Experimental Setup: All our experiments are run on a machine with Java 6, 64-bit Windows 7 with Intel(R) Core(TM) Duo CPU @2.27 GHz processor and 4GB of RAM. We compare the performance of *QueryMesh* (i.e., multi-route) against the traditional “single plan for all data” system (i.e., single route).

Our experiments include the measurement of several metrics: 1) the total execution time required to process a data set, 2) the average tuple latency, and 3) the cumulative throughput.

1.1 Data set and Query

The tuples of each stream are created such that the selectivities of joining subsets of one stream to the other streams varies. This causes multiple optimal routes to be available for this stream and hence multiple optimal processing routes may be required.

The experiments ran a 3 way join query across 3 synthetic data streams. Each experiment was run 10 times. The results are the averaged output of these runs.

1.2 Results and Analysis

Total Execution Time: We now compare the total execution time of *QueryMesh* to the single route model using the synthetic data outlined above. Figure 1 a indicates the total execution time of Query Mesh (i.e., multi-route) and single route model to run the synthetic data outlined above. We can observe that Query Mesh executed the data set 170.62% faster than the single route solution.

Average Tuple Latency: We now compare the average execution time of tuples through *QueryMesh* to the single route model using the synthetic data outlined above. Figure 1 b indicates the average execution time of Query Mesh (i.e., multi-route) and single route model to run the synthetic data outlined above. We can observe that the average execution time of Query Mesh is 107.21% faster than the single route solution.

Cumulative Throughput: Figure 1 c indicates the total number of tuples produced by the Query Mesh (i.e., multi-route) and the single route model over time using the synthetic data outlined above. We can observe that the cumulative throughput of Query Mesh remains significantly larger over time than the single route solution. In particular, Query Mesh produced 36.82% more results.

1.3 Summary of Experimental Results

The main findings of our experimental study are: 1) the total execution time for Query Mesh was 170.62% faster than the single route solution, 2) the average execution time for Query Mesh was 107.21% faster than the single route solution, and 3) *Query Mesh* improved the cumulative throughput rate by 36.82% better than that experienced by the single route plan.