Aufgabenbeschreibung:
Optimising interactive distributed dataflows
(Masterarbeit)
Nikolas Göbel

Frank McSherry.
Systems Group, ETH Zürich.

Autumn Semster 2018

1 Introduction

The goal of this project is to formulate a framework in which the performance of interactive incremental dataflow computations can be predicted and continously optimised, as clients register new flows and retire old ones. We will consider optimisation w.r.t to the number of concurrent flows maintainable at high-throughput and low-latency.

As more and more industries adopt data-driven decision-making and companies are increasingly looking to provide digital services, the capability to perform diverse complex analyses on potentially unbounded data streams is becoming a critical competitive advantage.

While recent, purpose-built stream processing systems can meet the very high throughputs and near real-time latencies demanded, we are interested in the interactive interfaces, higher-level query languages, and automatic optimisation neccessary to support real-world use-cases.

Understanding overall system performance in such a dynamic environment warrants re-visiting established practices in query optimisation.

Work will start from the *Differential Dataflow* codebase, an incremental distributed stream processing system, and the *Declarative Dataflow* query engine for higher-level languages, written on top of it. We will develop models and algorithms to understand and improve their scalability to many concurrent, interactive clients.

Specifically,

- The project will initially focus on identifying trade-offs, degrees of freedom (in implementing flows), and metrics that differ from established knowledge in the field of query optimisation. E.g. whereas a traditional query planner would only look at the size of a relation, we must also take into account its rate of change.
- We will then explore the problem of finding common sub-computations across flows, and the various ways of exploiting this commonality. Together with the

insights gained in the first part of the project, we will devise a query planner that, given the current overall dataflow graph, should be capable of re-writing a newly registered flow into a more efficient one producing equivalent results.

Several additional challenges exist, should the problem of optimising common sub-structures across a dynamic dataflow environment yield insufficient research opportunity. A simpler problem is the optimisation of a single dataflow, taking into account characteristics unique to the stream processing setting, but otherwise heavily inspired by traditional query planning. A harder problem would be extending the optimisation efforts to cover queries over historical data and execution across heterogeneous workers.

2 Background

Murray, McSherry, et al. - Naiad A Timely Dataflow System

McSherry, Murray, et al. - Differential dataflow

github.com/frankmcsherry/timely-dataflow

github.com/frankmcsherry/differential-dataflow

github.com/comnik/declarative-dataflow

Literature on traditional query planning and database optimisation.

3 Work plan

The work consists of the following stages, some of which can be performed in parallel.

1a. Familiarization with the differential dataflow stack and the Rust language.

1b. Familiarization with existing approaches to query optimisation in databases.

1c. Familiarization with existing approaches to optimisation in stream processing sytems.

2. Develop exemplary target applications in need of many concurrent, interactive dataflows. As an example, many companies wish to make sensor data queryable in near real-time, in order to support their analysts, while respecting individual access-control policies.

3a. Analysis of performance variation with characteristics such as dataset size, rate of change, percentage of shared sub-structures, and the degree of parallelism.

3b. Utilization of the timely dataflow logging utilities to monitor indicators of associated performance degradation.

3c. Analysis of the correspondence of performance variations with monitored indicators.

4. Proposal of a model for structured analysis and optimisation of query plans.

5. Implementation of a source-to-source query optimiser comparing new queries to existing flows, s.t. as long as flows leave and re-join from time to time, the whole system will tend to a better performing overall dataflow, without the need to re-structure running flows.
6. Validation of the efficacy of the proposals, and accuracy of their analysis.

There is the opportunity to arbitrarily refine the model and corresponding optimisation policies (steps 4-6), as initial attempts fail, or new avenues are discovered.

It should be stressed that the primary goal of the project is not to produce the single best performing dataflow for a specific computation, but rather to gain insight and understanding into how a modern dataflow system can support a wide range of non-trivial queries in demanding interactive environments while retaining its overall performance characteristics.

4 Deliverables

The project should result in the following concrete deliverables:

1. Thesis dissertation. This should follow the normal format for such a thesis, and contain at least the following:
   a. Introduction / motivation of the problem
   b. Survey of related work, approaches to dataflow optimisation and query planning
   c. Description of the approach and methodology
   d. Details of the design and implementation
   e. Performance evaluation, and validation of hypotheses
   f. Discussion of broader implications, opportunities for future work, unresolved issues
2. Complete source code for:
   a. Workloads (real and synthetic) demonstrating various performance characteristics
   b. The optimiser implementation
   c. Scripts used to conduct experiments
   d. Any patches to timely's monitoring infrastructure
   e. Any tooling used to analyze experimental results
   f. Any patches introducing infrastructural improvements
3. Presentation of thesis results and demonstration of functionality

5 Notenschema

The minimum requirements for a grade of 5.0 are as follows:

- Demonstration of model predictions and corresponding analysis on representative workloads, as outlined in work units 1-6 above.
- Optimisation policies improving the above performance based on operational metrics established in work units 3 and 4, for at least one non-trivial workload.
- Completion of deliverables 1-3 to a satisfactory standard.

The grade will be reduced if these goals are not achieved, except in the case of extreme extenuating circumstances (such as an unforeseeable and unresolvable technical barrier to completing the work, accompanied by an acceptable alternative work item).

A grade of 5.50 will be awarded for the completion of the minimum work to an unusually high quality, or with the addition of extra research work accompanied by documentation and writeup in the thesis.