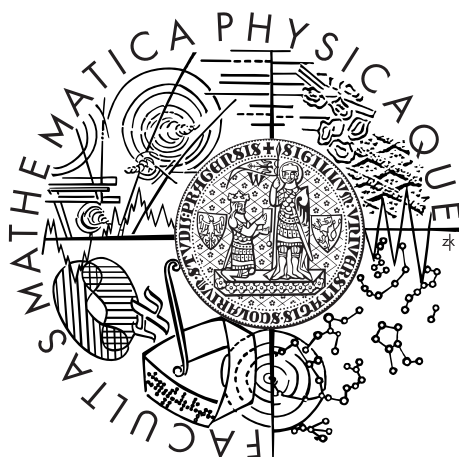


Charles University in Prague
Faculty of Mathematics and Physics

MASTER THESIS



Marcel Kikta

Evaluating relational queries in pipeline-based environment

Department of Software Engineering

Supervisor of the master thesis: David Bednárek

Study programme: Software systems

Specialization: Software engineering

Prague 2014

Dedication.

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In date

signature of the author

Název práce: Vyhodnocování relačních dotazů v proudově orientovaném prostředí

Autor: Marcel Kikta

Katedra: Katedra softwarového inženýrství

Vedoucí diplomové práce: RNDr. David Bednárek, Ph.D.

Abstrakt:

Klíčová slova: SQL, Překladač, Relační algebra, Optimalizator, Bobox

Title:

Author: Marcel Kikta

Department: Název katedry či ústavu, kde byla práce oficiálně zadána

Supervisor: RNDr. David Bednárek, Ph.D.

Abstract:

Keywords: SQL, Compiler, Relational algebra, optimizer, Bobox

Contents

Introduction	2
1 Architecture	3
1.1 Bobox	3
1.2 Bobolang	4
1.3 Architecture	4
1.4 Goal	4
2 Related work	5
2.1 SQL and relational algebra	5
2.2 Optimization and plan generation	5
3 Analysis	6
4 Implementation	7
Conclusion	8
Bibliography	9
List of Tables	10
List of Abbreviations	11
Attachments	12

Introduction

Today's processors have multiple cores and it's single core performance is improving only very slow because of physical limitations. On the other hand number of cores is still increasing and we can assume that it will continue. That's why developing parallel software is crucial for improving overall performance.

Parallelization can be achieved manually or using some framework designed for it. For example there are frameworks like OpenMP or Intel TBB. Department of Software Engineering at Charles University in Prague developed it's own parallelization framework called Bobox[1].

Bobox is designed for parallel processing large amounts of data. It was specifically created to simplify and speed up parallel programming of certain class of problems - data computations based on non-linear pipeline. It was created to evaluate queries over relational data but it was successfully used in implementation of XQuery and TriQuery engines.

Bobox contains from runtime environment and operators. These operators are called boxes and they are C++ implementation of data processing algorithm. Boxes use messages called envelopes to send processed data to each other.

Bobox takes as input execution plan written in special language Bobolang[2]. It allows to define used boxes and simply connect them into directed acyclic graph. Bobolang specifies the structure of whole application and also the inner structure of each box. It can create highly optimized evaluation, which is capable of using the most of the hardware resources. The language has been tested in several applications and it turned out to be very powerful tool in data processing massive parallel application.

Most used databases are relational databases. They are based on the view of data organized in tables called relations. SQL[3] ("Structured query language") is very important language based on relation databases. It is used for querying data, modifying content of tables and also the structure of tables. When we want to evaluate query we need to parse query text input into parse tree. This form will be transformed to relational algebra, which we call logical query plan. It will be optimized and physical plan is generated. Physical plan indicates not only operation performed, but also which order are they performed and what kind of algorithms are used for execution.

The main goal of this thesis is to implement part of SQL compiler. The input is query written in XML format in form of relational algebra. Program validates input, optimizes and transforms it to physical plan of given query. The output is execution plan for Bobox written in Bobolang.

1. Architecture

1.1 Bobox

In the section we describe basic architecture of Bobox. Information source for this chapter is Doctoral thesis Parallel Processing of Data[4].

Overall Bobox architecture is displayed in figure 1.1. Framework contains of Boxes. Box is basically a C++ class containing implementation of data processing algorithm. Box can have arbitrary number of inputs and outputs. All boxes are connected to a directed acyclic graph.

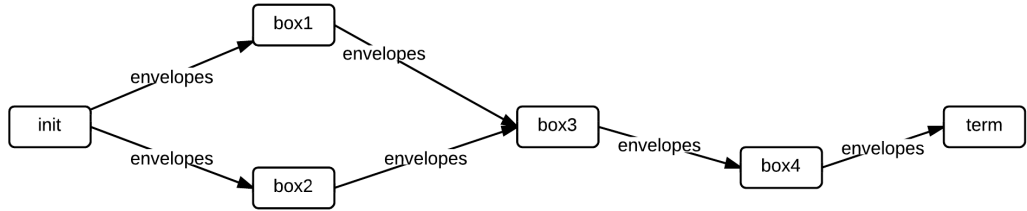


Figure 1.1: Bobox architecture.

Data streams are implemented as data units called enveloped. Envelope structure is displayed in figure 1.2. It consists of sequence tuples, but internally data are stored by columns, that means envelope contains from sequence of columns and it's data is stored in separate list. So to read all attributes of the i -th tuple we have to access all column lists and read it's i -th element. There is special type of envelope having poisoned pill. It is send after all valid data indicating end of data stream.

There are two special boxes, which have to be in every execution plan:

- *init* - first box in topological order and it indicates starting box of execution plan
- *term* - last box in topological order and indicates that plan has been completely evaluated

Evaluation starts with scheduling *init* box, which sends poisoned pills to all of its output. All of it's output boxes will be scheduled. They can read data from hard drive or network, process it and sent it to other boxes for further processing. Other boxes usually receives data in envelopes in their inputs. Box *term* waits for every it's input to receive poisoned pill.

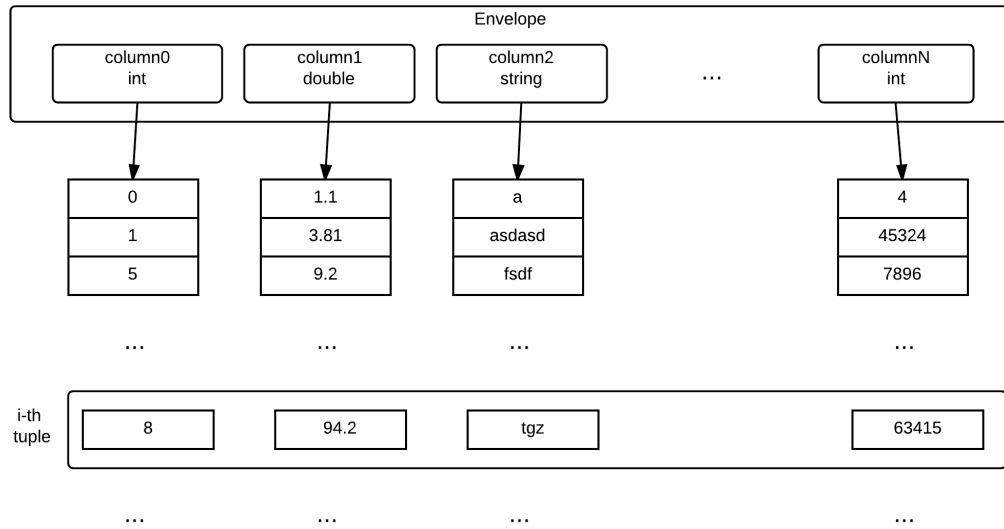


Figure 1.2: Envelope structure.

1.2 Bobolang

1.3 Architecture

1.4 Goal

2. Related work

2.1 SQL and relational algebra

2.2 Optimization and plan generation

3. Analysis

4. Implementation

Conclusion

Bibliography

- [1] D. Bednárek, J. Dokulil, J. Yaghob, and F. Zavoral. *Bobox: Parallelization framework for data processing. Advances in Information Technology and Applied Computing*, 2012.
- [2] Z. Falt, , D. Bednárek, K. Martin, J. Yaghob, and F. Zavoral. *Bobolang - a language for parallel streaming applications. In 23rd international symposium on High-Performance Parallel and Distributed Computing*. ACM, 2014.
- [3] H. Garcia-Molina, , J. D. Ullman, J. Widom. *Database Systems The Complete Book*. Prentice Hall, 2002, ISBN 0-13-031995-3.
- [4] Zbyněk Falt. *Parallel Processing of Data - Doctoral thesis*. Prague, 2013.

List of Tables

List of Abbreviations

Attachments