University of Zurich^UZH

Department of Informatics

University of Zürich
Department of Informatics
Binzmühlestr. 14
CH-8050 Zürich
Phone. +41 44 635 43 11
Fax +41 44 635 68 09
www.ifi.uzh.ch/dbtg

UZH, Dept. of Informatics, Binzmühlestr. 14, CH-8050 Zürich

Prof. Dr. Michael Böhlen
Professor
Phone +41 44 635 43 33
Fax +41 44 635 68 09
boehlen@ifi.uzh.ch

Zürich, 9. Juni 2023

Master Thesis (30 ECTS)
Database Technology

Topic: Optimizing linear algebra computations in MonetDB

There are significant amounts of curated business data that are maintained in databases. This data must be analyzed to extract actionable business intelligence. The goal of this Master thesis is to integrate linear algebra and iterations into the MonetDB column store system, implement efficient evaluation techniques, and evaluate the solution on large data sets.

Many common analysis tasks combine linear regression with gradient descent to estimate the relationship between the independent variables and a dependent variable. Linear regression, in turn, can be reduced to two linear algebra operations: matrix transpose (`TRA`) and matrix multiplication (`MMU`). In column store systems some linear algebra operations, e.g., matrix addition (`ADD`), can naturally be expressed as column vector operations. Other operations, however, such as `TRA` and `MMU`, have to access the elements row-by-row, which is an access pattern that column vectors do not support efficiently. To address this mismatch new methods must be designed that make it possible to efficiently evaluate operations with row access patterns in column store systems [1, 3].

To analyze real world data it is not sufficient to consider linear algebra operations in isolation. Instead we have to combine linear algebra operations with relational algebra operations and iterations [2] to formulate typical data analysis tasks. For instance, joins and aggregations are needed to preprocess data while iterations are needed for gradient descent computations. During iterative computations it is important to reuse column vectors and avoid the slow repetitive allocation of new memory. This requires changes at the system level of MonetDB since relational algebra operations are implemented as out-of-place operations.

As an example analysis task we consider the prediction of the price of a property for a given set of features, such as crime rate (CR), nitric oxide (NO), number of rooms (NR), distance to center (DC), etc. CR and NO data are available for each area (identified by Zip code) in relation d. The features of a property, such as NR, DC and Price, are available in relation p.

d

Zip	CR	NO
8001	0.006	0.538
8002	0.027	0.469
8003	0.047	0.573
...	...	...

p

id	Zip	NR	DC	Price	...
n001	8001	6	4.09	24.00	...
n002	8001	5	4.967	21.60	...
n003	8002	4	2.505	11.90	...
...	...	...	...	...	...

Over this database we want to train a linear model with a number of independent variables (CR, NO, NR, DC) determining a dependent variable (Price). We use SQL to gather the independent and dependent variables from the database in relations X and y. We randomly initialize the contributions (weight) of the independent variables in relation W. During the linear regression computation the weights are iteratively refined according to the gradient until the approximation of $y = XW$ is sufficiently precise.

X

id	CR	NO	NR	DC	...
n001	0.006	0.538	6	4.09	...
n002	0.027	0.469	5	4.967	...
n003	0.047	0.573	4	2.505	...
...	...	...	...	...	...

y

id	price
n001	24.00
n002	21.60
n003	11.90
...	...

W

feature	weight
CR	-0.1
NO	-0.2
...	...

The individual tasks of the MSc thesis are described below. A carefully worked out MSc thesis that describes the solutions to these tasks must be handed in. The results must be presented at a DBTG meeting.

Task 1:

Implement and integrate the transpose operation into MonetDB. Particular attention must be paid to the handling of schema information. Since the schema of a transposed relation can neither be derived from existing schema information nor from the query a new evaluation approach must be implemented. The integration of the transpose operation into the query evaluation pipeline of MonetDB must be designed so that transpose can be used in nested sequences of operations (e.g., $\pi_{n002}(\texttt{TRA}(d \bowtie p, id))$ and $\texttt{TRA}(\texttt{TRA}(d \bowtie p, id), c)$). To achieve this the following elements of MonetDB must be studied and extended:

- overview, deployment
- SQL syntax, symbol tree (sql_parser.y)
- relation tree (rel_select.c)
- statement tree (rel_bin.c)
- execution backend (batcalc.c, gdk_calc.c)

Task 2:

Matrix multiplication is a crucial linear algebra operation that is difficult to implement efficiently since a direct implementation of the mathematical definition accesses elements row-by-row as well as column-by-column. The goal of this task is to investigate, empirically compare and precisely quantify different approaches to implement matrix multiplication in MonetDB. The solution shall leverage MonetDB's column vectors and pay attention to memory access patterns and cache usage.

Task 3:

A gradient descent computation is an iterative process that uses the gradient of the loss function to update weight matrix W. Thus, for learning rate η and loss function L we iteratively update the weight matrix according to $W = W - \eta * \nabla L(W)$. For linear regression we get $\nabla L(w) = \frac{1}{N} X^t(XW - y)$ with N being the number of samples. The solutions developed in tasks 1 and 2 shall be leveraged to integrate efficient linear regression computations into MonetDB together with an in-depth analysis of the memory usage during iteration over large database instances. In addition, the use of your solution in reinforcement learning applications shall be investigated.

References

[1] Peter A Boncz, Marcin Zukowski, and Niels Nes. Monetdb/x100: Hyper-pipelining query execution. In *Cidr*, volume 5, pages 225–237, 2005.

[2] Oksana Dolmatova, Nikolaus Augsten, and Michael H Böhlen. A relational matrix algebra and its implementation in a column store. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 2573–2587, 2020.

[3] Milena G Ivanova, Martin L Kersten, Niels J Nes, and Romulo AP Gonçalves. An architecture for recycling intermediates in a column-store. *ACM Transactions on Database Systems (TODS)*, 35(4):1–43, 2010.

Supervisor: Xinyu Zhu (`xinyu.zhu@uzh.ch`)

Start date: May 15, 2023

End date: November 15, 2023

University of Zurich
Department of Informatics

[illegible]

Prof. Dr. Michael Böhlen
Professor