MonetDB/X100

Presented by
NIKHIL P.
MCA S5

Contents

- Introduction
- What is X100?
- Background
- MonetDB Design
- X100 Query processor
- Data Storage
- Related Works
- Conclusion
- References

INTRODUCTION

- MonetDB is an open-source Database Management System(DBMS)

- MonetDB is designed for high performance applications in data mining, business intelligence, OLAP, scientific databases, XML query, text and multimedia retrieval, etc.

INTRODUCTION

- It was designed primarily for data warehouse applications

- MonetDB achieves significant speed up compared to traditional designs by innovations at all layers of a DBMS.

INTRODUCTION

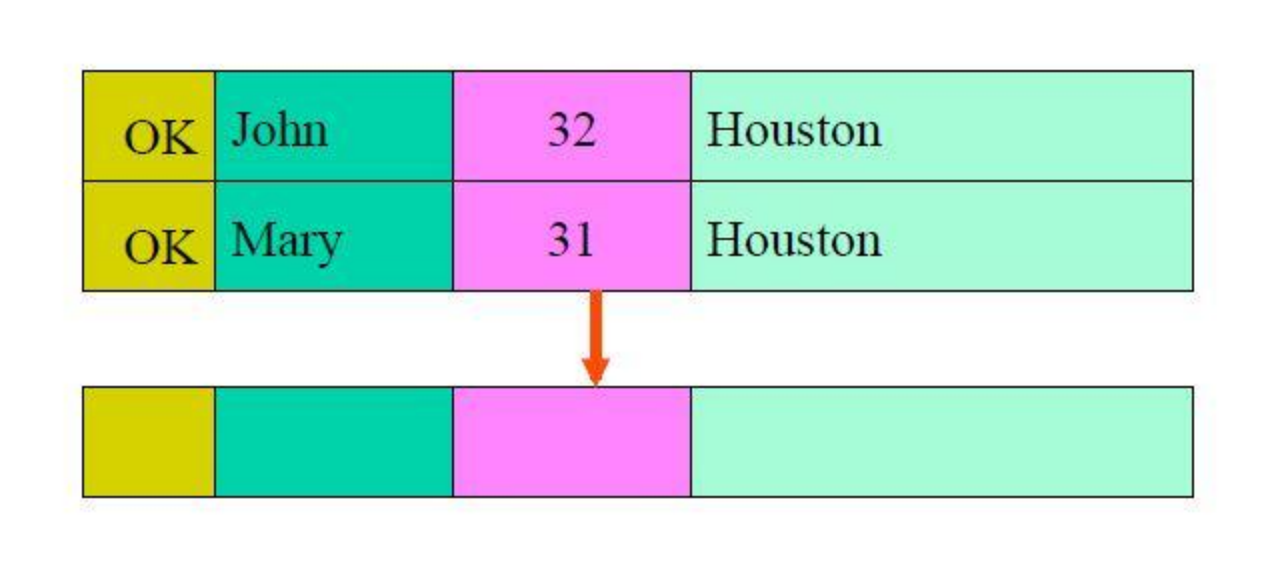
- a storage model based on vertical fragmentation
- a modern CPU-tuned query execution architecture
- automatic and adaptive indices
- run-time query optimization
- a modular software architecture.

What is X100?

- X100 is a new query processing engine developed for MonetDB.

Background

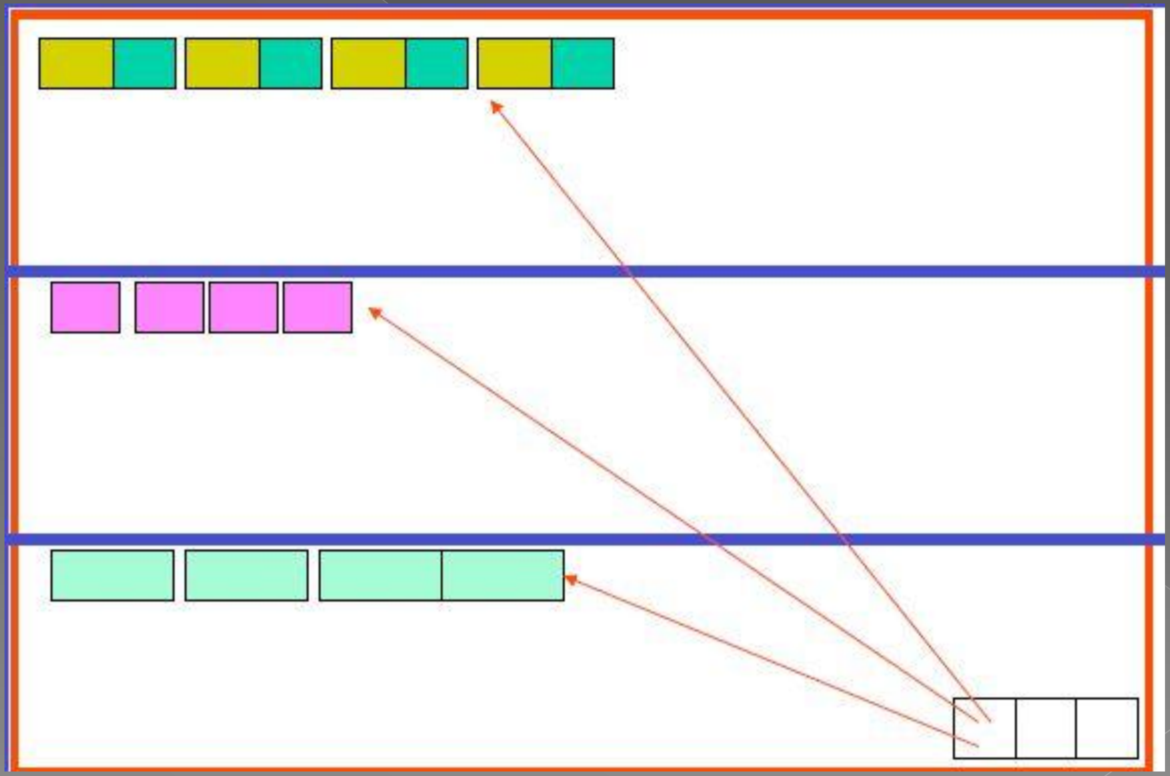
- Early 80s:Tuple storage structures for PCs were simple



OK	John	32	Houston
OK	Mary	31	Houston

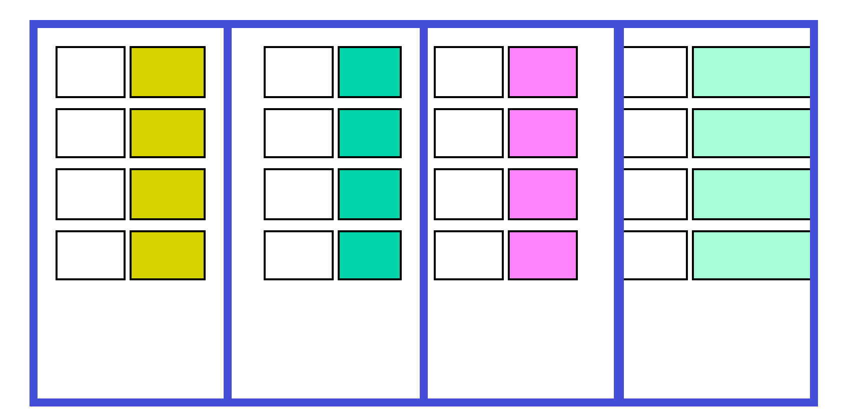
Background

- Not all attributes are equally important



Background

- A column orientation is as simple and it acts like an array.



- Attributes of a tuple are correlated by offset

MonetDB Design

- MonetDB is a full-fledged relational DBMS that supports SQL:2003 and provides standard client interfaces such as ODBC and JDBC.

- Application programming interfaces for various programming languages including C, Python, Java, Ruby, Perl and PHP.

MonetDB Design

- It is designed to exploit the large main memories of modern computers during query processing.
- It is one of the first publicly available DBMS designed to exploit column store technology.

Physical Data Model

- Instead of storing all attributes of each relational tuple together in one record, MonetDB represents relational tables using vertical fragmentation, by storing each column in a separate table called BAT.

- The left column is called ‘head’ and the right column holding actual attribute values is called ‘tail’.

Physical Data Model

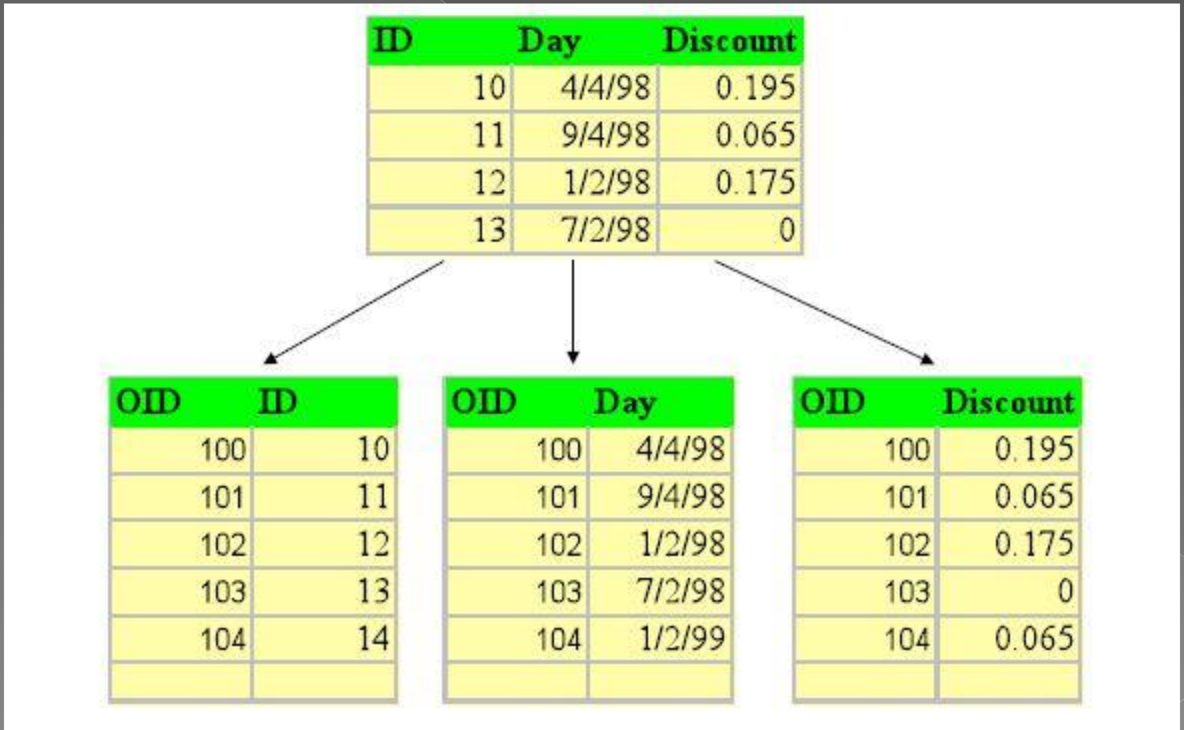
- Every relational table is internally represented as collection of BAT(Binary Association Table)s.

- For a Relation R of ‘k’ attributes, there exists k BATs each BAT stores the attribute as (OID, value) pairs.

- System generated OID value identifies the relational tuple that the attribute value belongs to, ie, all attribute values of a single tuple are assigned same OID.

Physical Data Model

- Binary Association Tables



ID	Day	Discount
10	4/4/98	0.195
11	9/4/98	0.065
12	1/2/98	0.175
13	7/2/98	0

OID	ID
100	10
101	11
102	12
103	13
104	14

OID	Day
100	4/4/98
101	9/4/98
102	1/2/98
103	7/2/98
104	1/2/99

OID	Discount
100	0.195
101	0.065
102	0.175
103	0
104	0.065

Physical Data Model

- For fixed width data types (eg: int) MonetDB uses a plain C array of the respective type to store the value column of a BAT.

- For variable-width data types (eg:strings) MonetDB applies a kind of dictionary encoding.

Physical Data Model

- MonetDB uses OS’s memory mapped files support to load data in main memory and exploit extended virtual memory. Thus, all data structures are represented in the same binary format on disk and in memory.

- It uses late tuple reconstruction, i.e., during the entire query evaluation all intermediate results are in a column format.

Execution Model

- MonetDB kernel is an abstract machine, programmed in the MonetDB Assembly Language(MAL).

- The core of MAL is formed by a closed low level two-column relational algebra on BATs.

- Complex operations are broken into a sequence of BAT algebra operators that each perform a simple operation on an entire column of values.

System Architecture

- MonetDB's query processing scheme is centered around three software layers:
- Front end: It provides the user-level data model and query language.
  - The front end's task is to map the user space data model to MonetDB's BATs and to translate the user space query language to MAL.

System Architecture

- Back end:
  - It consists of the MAL optimizers framework and MAL interpreter as textual interface to the kernel.
  - The MAL optimizers framework consists of a collection of optimizer modules that each transform a MAL program into a more efficient one, possibly adding resource management directives.
  - Operating on the common binary relational back-end algebra, these optimizer modules are shared by all front-end data models and query languages.

System Architecture

- Kernel:
  - The bottom layer provides BATs as MonetDB’s important data structure.

X100: query Processor

- Goal of X100 is to:
  - Execute high volume queries at high CPU efficiency.
  - Extensible to other application domains and achieve those same efficiency on extensible code.
  - Scale with the size of the lowest storage hierarchy.

To achieve these goals, X100 must fight with entire computer memory architecture

X100: query Processor

- Disk
  - It uses a vertically fragmented data layout, sometimes is enhanced with lightweight data compression

- RAM
  - The same vertically partitioned and compressed disk data layout is used in RAM to save space and bandwidth.

X100: query Processor

- Cache
  - Vertical chunks of cache-resident data items called ‘vectors’ are the unit of operation for X100 execution primitives
  - X100 query processing operators should be cache-conscious and fragment huge datasets efficiently into cache-chunks and perform random data access only in the cache.
- CPU
  - X100 primitives expose to the compiler that processing a tuple is independent of the previous and next tuples.

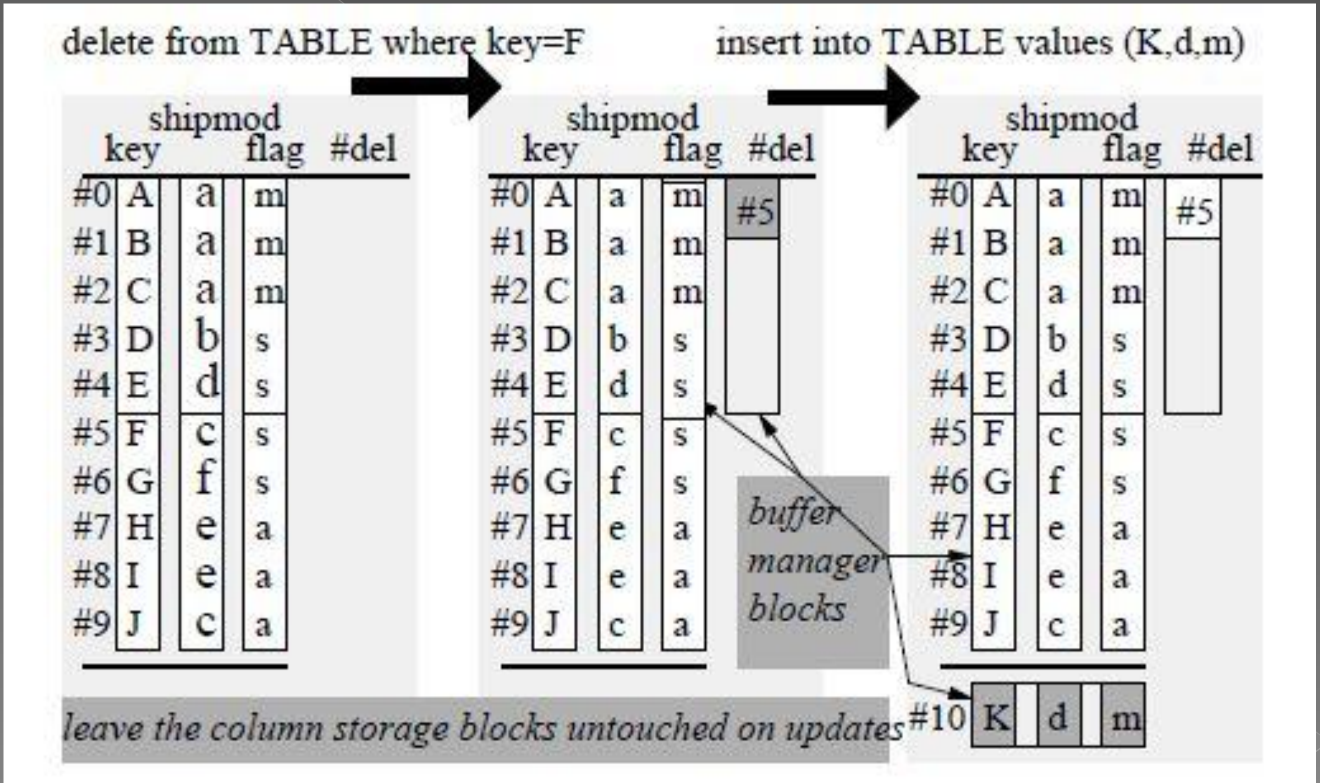
Data Storage

- MonetDB/X100 stores all tables in vertically fragmented form
- MonetDB stores each BAT in a single contiguous file, where columnBM partitions those files in large chunks.
- A disadvantage of vertical storage is an increased update cost: a single row update or delete must perform one I/O for each column.

Data Storage

- MonetDB solves this by treating the vertical fragments as immutable objects, updates goto delta structures instead.
- Updates make the delta columns grow, whenever the size exceeds, data storage should be reorganized, ie., the vertical storage is up-to date again and delta columns are empty.

Data Storage



delete from TABLE where key=F

insert into TABLE values (K,d,m)

shipmod

	key		flag	#del
#0	A	a	m	
#1	B	a	m	
#2	C	a	m	
#3	D	b	s	
#4	E	d	s	
#5	F	c	s	
#6	G	f	s	
#7	H	e	a	
#8	I	e	a	
#9	J	c	a	

shipmod

	key		flag	#del
#0	A	a	m	#5
#1	B	a	m	
#2	C	a	m	
#3	D	b	s	
#4	E	d	s	
#5	F	c	s	
#6	G	f	s	
#7	H	e	a	
#8	I	e	a	
#9	J	c	a	

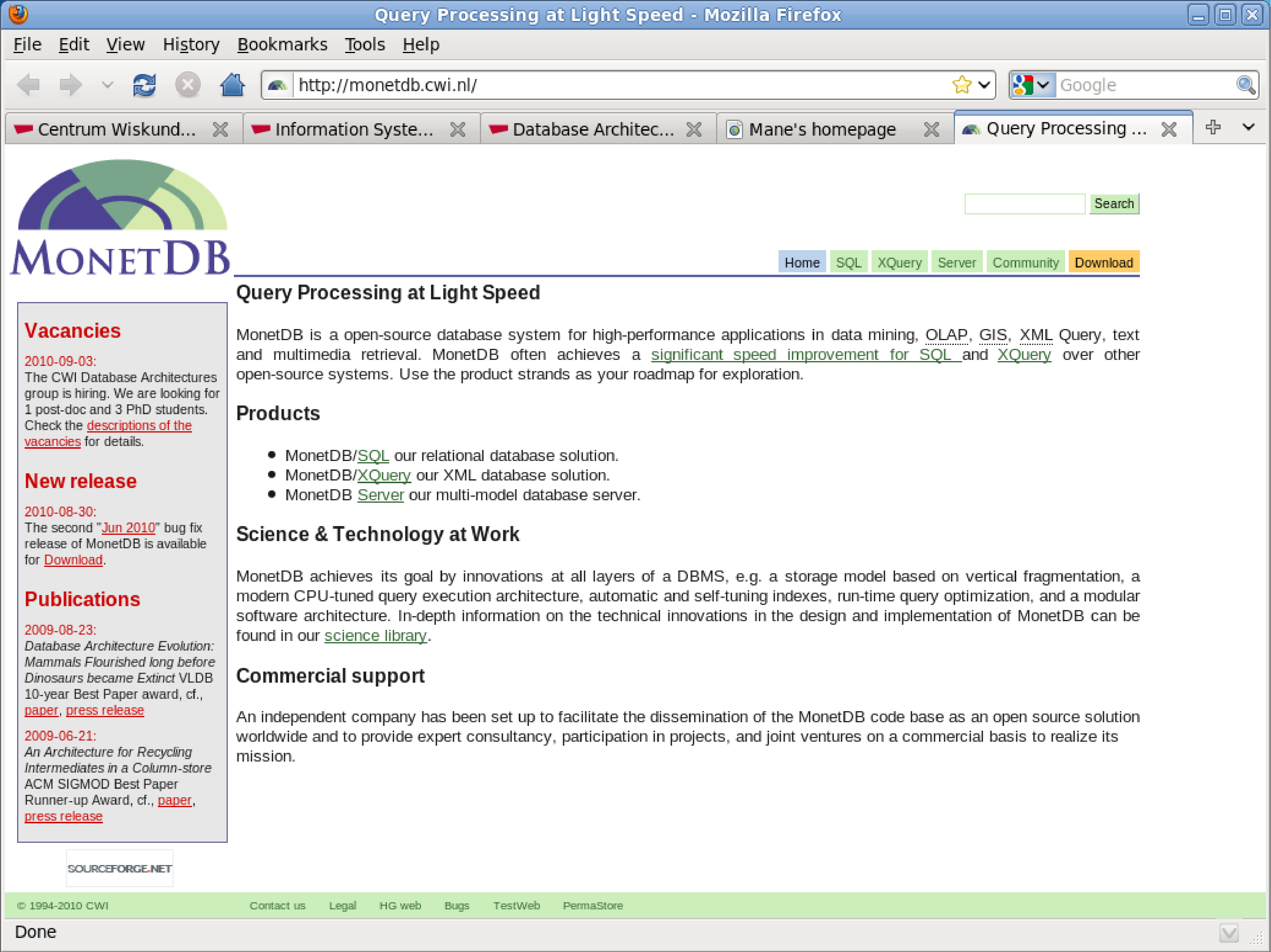
shipmod

	key		flag	#del
#0	A	a	m	#5
#1	B	a	m	
#2	C	a	m	
#3	D	b	s	
#4	E	d	s	
#5	F	c	s	
#6	G	f	s	
#7	H	e	a	
#8	I	e	a	
#9	J	c	a	
#10	K	d	m	

leave the column storage blocks untouched on updates

Data Storage

- An advantage of vertical storage is that queries that access many tuples but not all columns saves bandwidth.



Vacancies

2010-09-03:
The CWI Database Architectures group is hiring. We are looking for 1 post-doc and 3 PhD students. Check the descriptions of the vacancies for details.

New release

2010-08-30:
The second "Jun 2010" bug fix release of MonetDB is available for Download.

Publications

2009-08-23:
Database Architecture Evolution: Mammals Flourished long before Dinosaurs became Extinct VLDB 10-year Best Paper award, cf., paper, press release

2009-06-21:
An Architecture for Recycling Intermediates in a Column-store ACM SIGMOD Best Paper Runner-up Award, cf., paper, press release

Query Processing at Light Speed

MonetDB is a open-source database system for high-performance applications in data mining, OLAP, GIS, XML Query, text and multimedia retrieval. MonetDB often achieves a significant speed improvement for SQL and XQuery over other open-source systems. Use the product strands as your roadmap for exploration.

Products

- MonetDB/SQL our relational database solution.
- MonetDB/XQuery our XML database solution.
- MonetDB Server our multi-model database server.

Science & Technology at Work

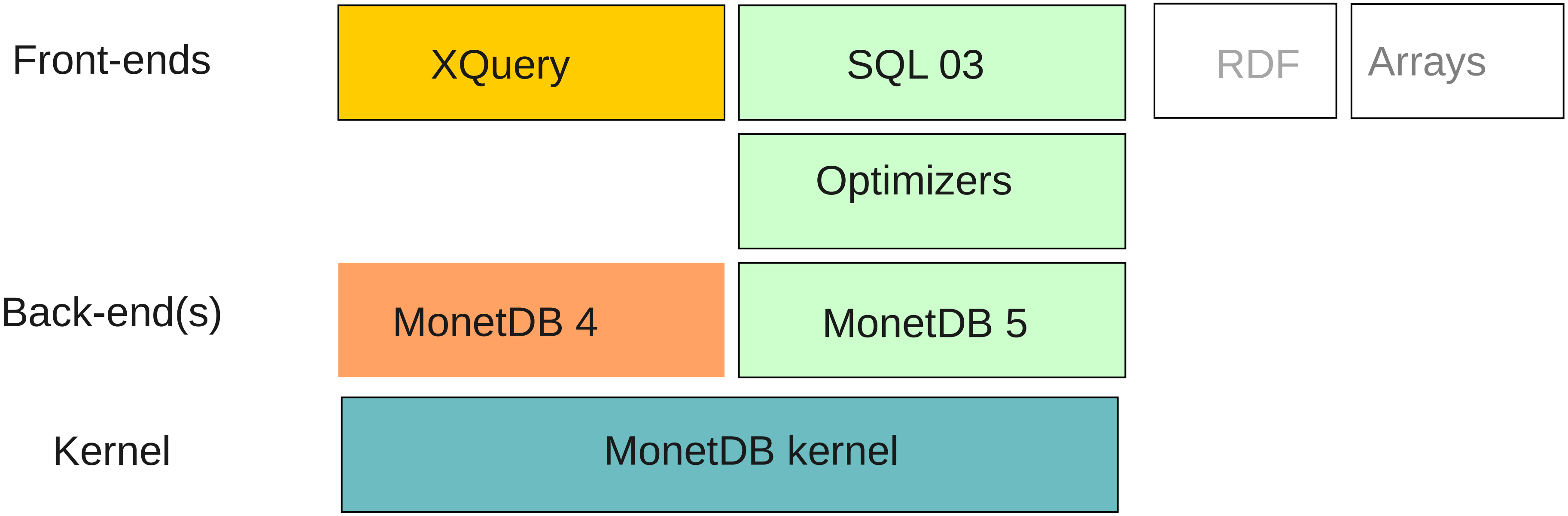
MonetDB achieves its goal by innovations at all layers of a DBMS, e.g. a storage model based on vertical fragmentation, a modern CPU-tuned query execution architecture, automatic and self-tuning indexes, run-time query optimization, and a modular software architecture. In-depth information on the technical innovations in the design and implementation of MonetDB can be found in our science library.

Commercial support

An independent company has been set up to facilitate the dissemination of the MonetDB code base as an open source solution worldwide and to provide expert consultancy, participation in projects, and joint ventures on a commercial basis to realize its mission.

© 1994-2010 CWI Contact us Legal HG web Bugs TestWeb PermaStore

The MonetDB Software Stack



Front-ends

XQuery

SQL 03

RDF

Arrays

Optimizers

Back-end(s)

MonetDB 4

MonetDB 5

Kernel

MonetDB kernel

Related Works

- MIT Column Store
  - First column store to implement the columnar-oriented database system.
  - Column store maps a table to projects, and thus allows redundant columns that appear inside multiple projects. Each column in the project is stored with the column-wise storage layout.

Related Works

- Microsoft SQL Server 2012
  - The recent version supports columnar storage and efficient batch-at-a-time processing.
  - Comparing with MonetDB, SQL server 2012 allows only the column index and it is unclear whether the underlying storage layout of data value is also designed for the column storage.

Related Works

- Main Memory Hybrid Column Store
  - Is a main memory database system and it automatically partition tables into vertical partitions of varying widths.
  - It is similar to the column storage of MonetDB.

Related Works

- Google BigTable
  - It is designed to scale for petabytes of strutured data and thousands of commodity servers.
  - Bigtable allows client to group multiple column families together into a locality group.
  - The locality groups of BigTable does not support CPU-cache-level optimizations that are used in MonetDB.

Conclusion

The comparison with other column store approaches provides its importance over other technologies. The column store approach is becoming widely accepted among everything and it indicates that MonetDB is going to be widely accepted and used among all database related frameworks.

References

- [1] Maarten Vermeij1, “MonetDB, a novel spatial column-store DBMS”, TUDelft, OTB, section GIS-technology

- [2] Peter Boncz, “MonetDB/X100: Hyper-Pipelining Query Execution”, CWI, Amsterdam, The Netherlands, 2005

- [3] Weixiong “MonetDB and the application for IR Searches”, Rao Department of Computer Science University of Helsinki, Finland, 2012