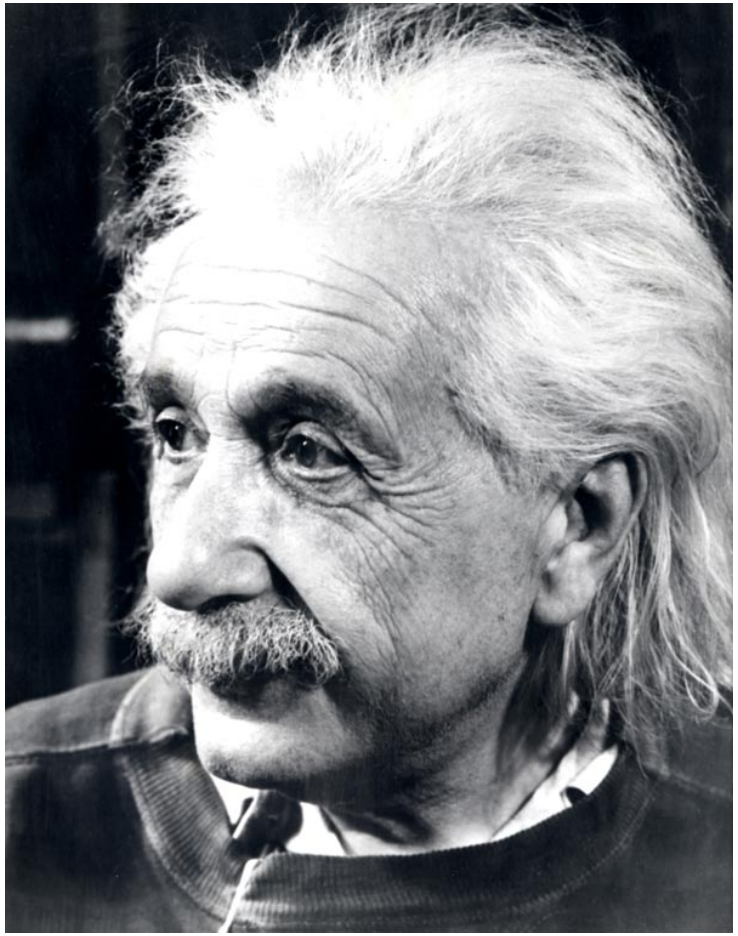
MonetDB:

Reaching the stars step by step

Martin Kersten
Milena Ivanova
Arjen de Rijke
http://www.monetdb.org/

2010 NGSS Sky Survey Data Management Workshop, Edinburgh

“We can't solve problems by using the same kind of thinking we used when we created them.”



The world of column stores

Functionality and performance of MonetDB

Roadmap for a Science Database System

The landscape

Motivation

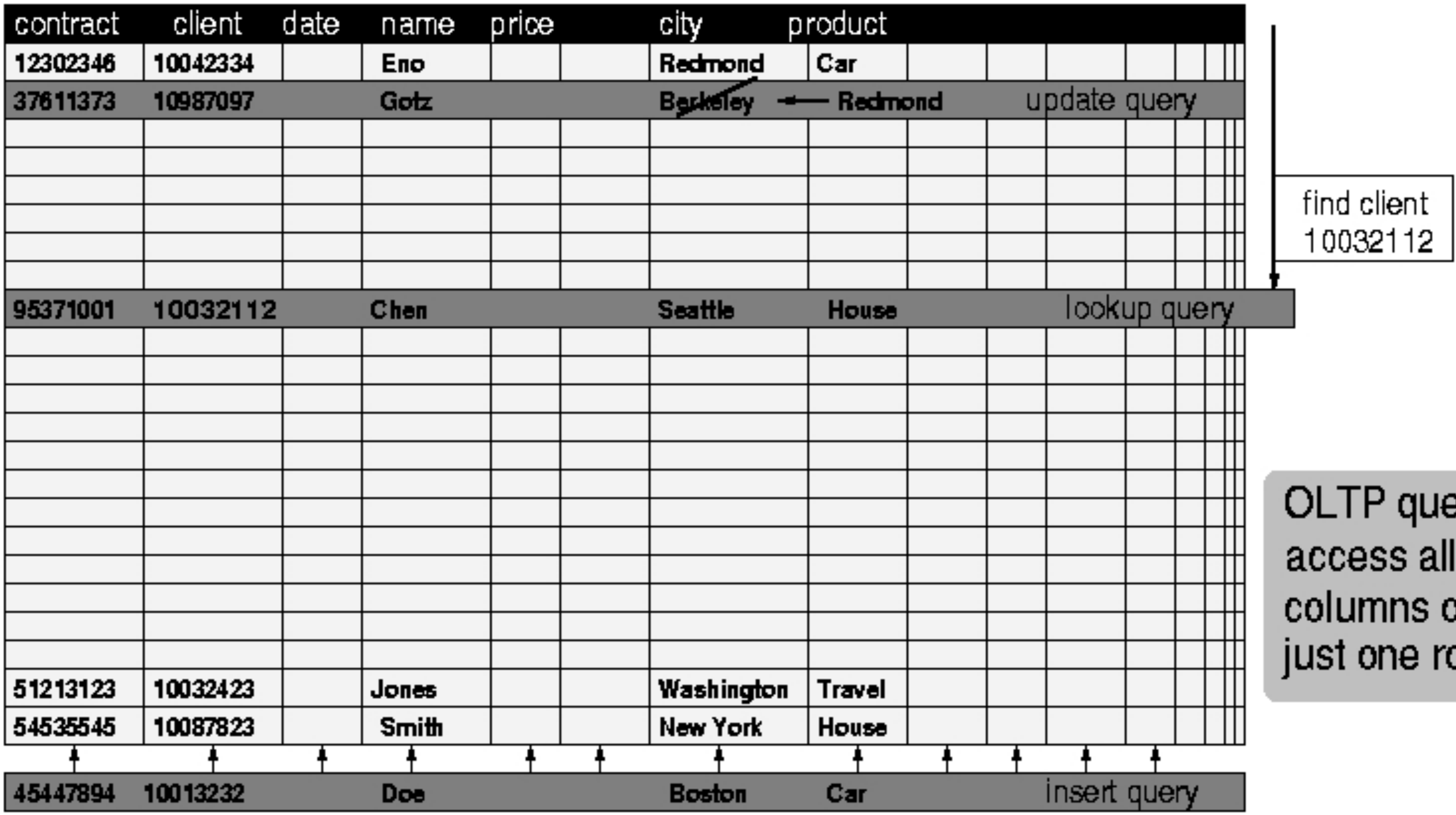
- Relational DBMSs dominate since the late 1970's / 1980's
  - Transactional workloads (OLTP, row-wise access)
  - I/O based processing
  - Ingres,Postgresql, MySQL, Oracle, SQLserver, DB2, ...

- Column stores dominate product development since 2008
  - Datawarehouses and business intelligence applications
  - Startups: Infobright, Aster Data, Greenplum, LucidDB,..
  - Commercial: Microsoft, IBM, SAP,…

MonetDB, the pioneer

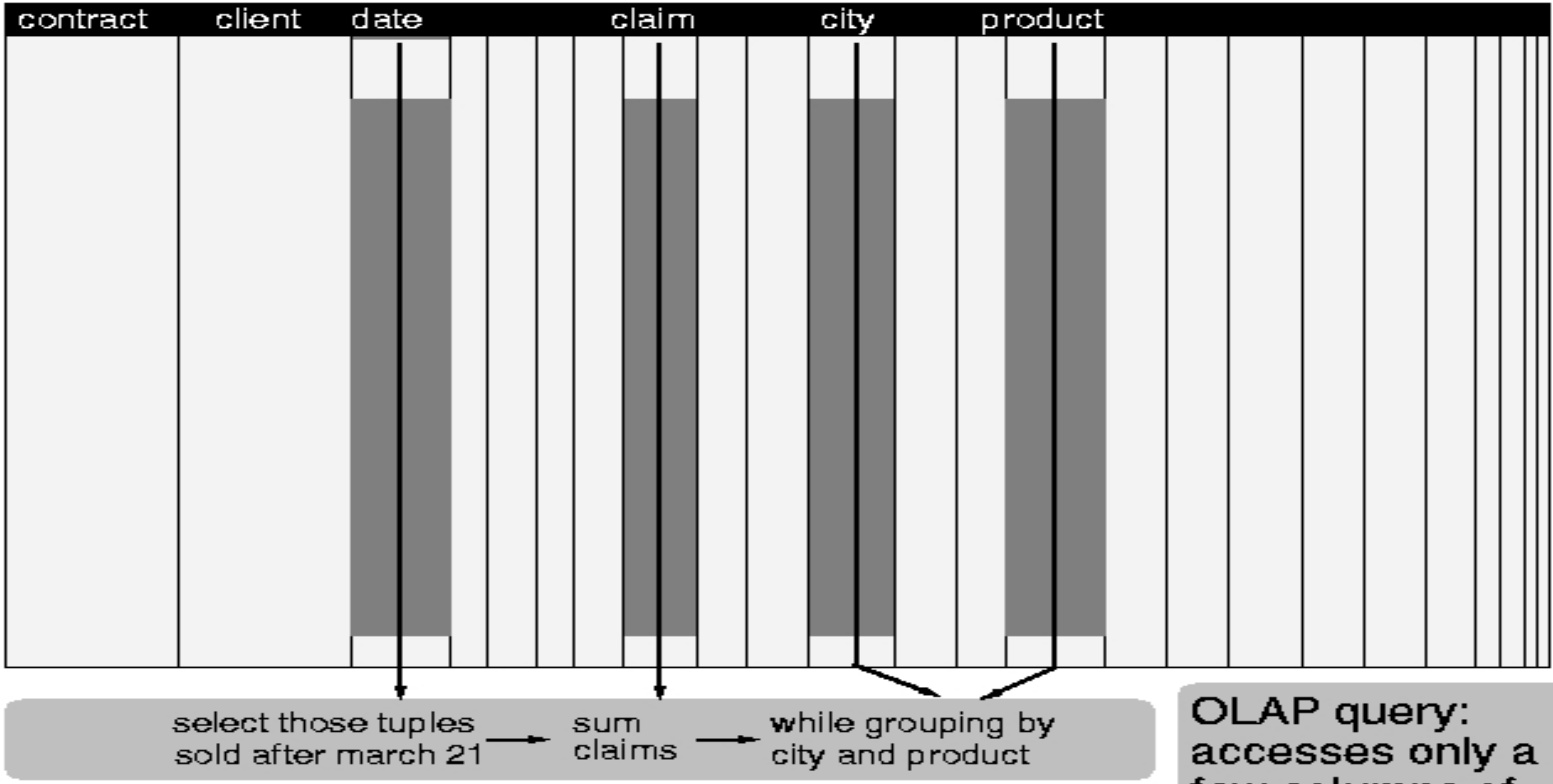
The world of column stores

Workload changes: Transactions (OLTP) vs ...



OLTP queries: access all columns of just one row.

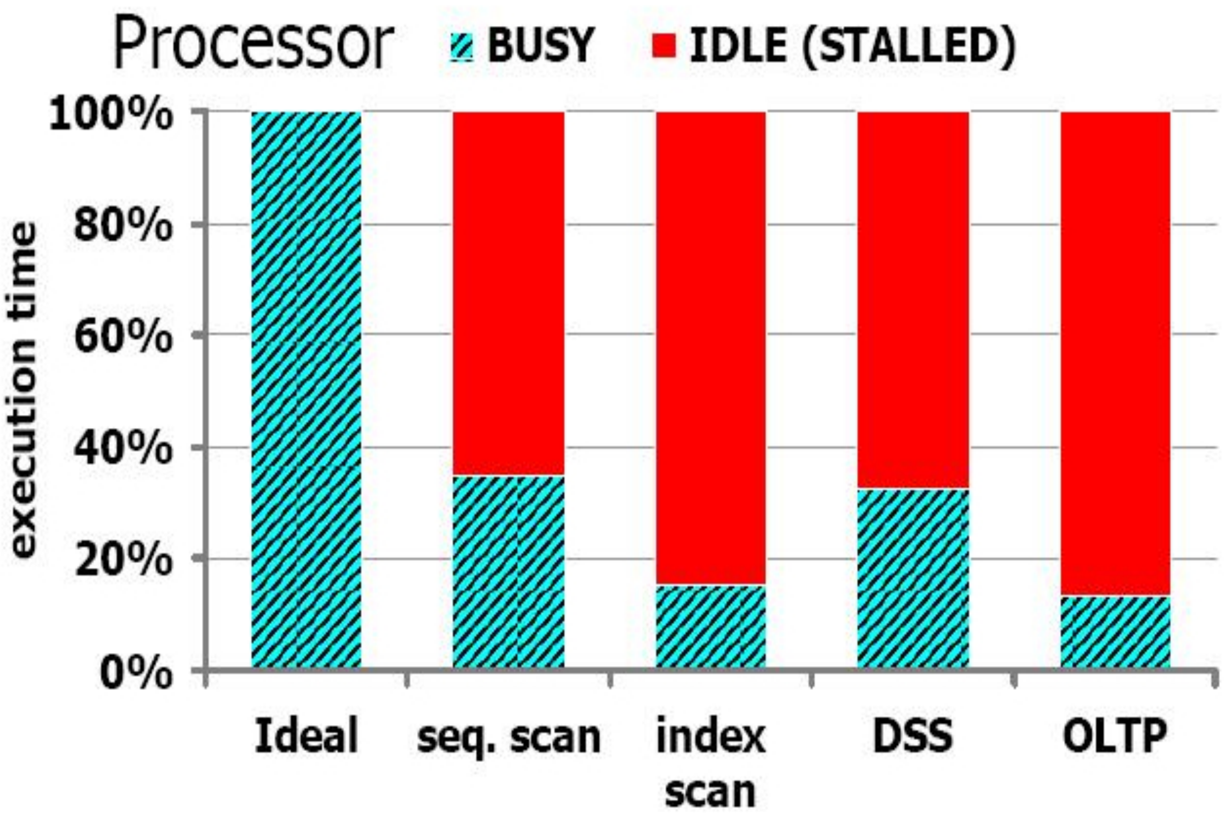
Workload changes: ... vs OLAP, BI, Data Mining, ...



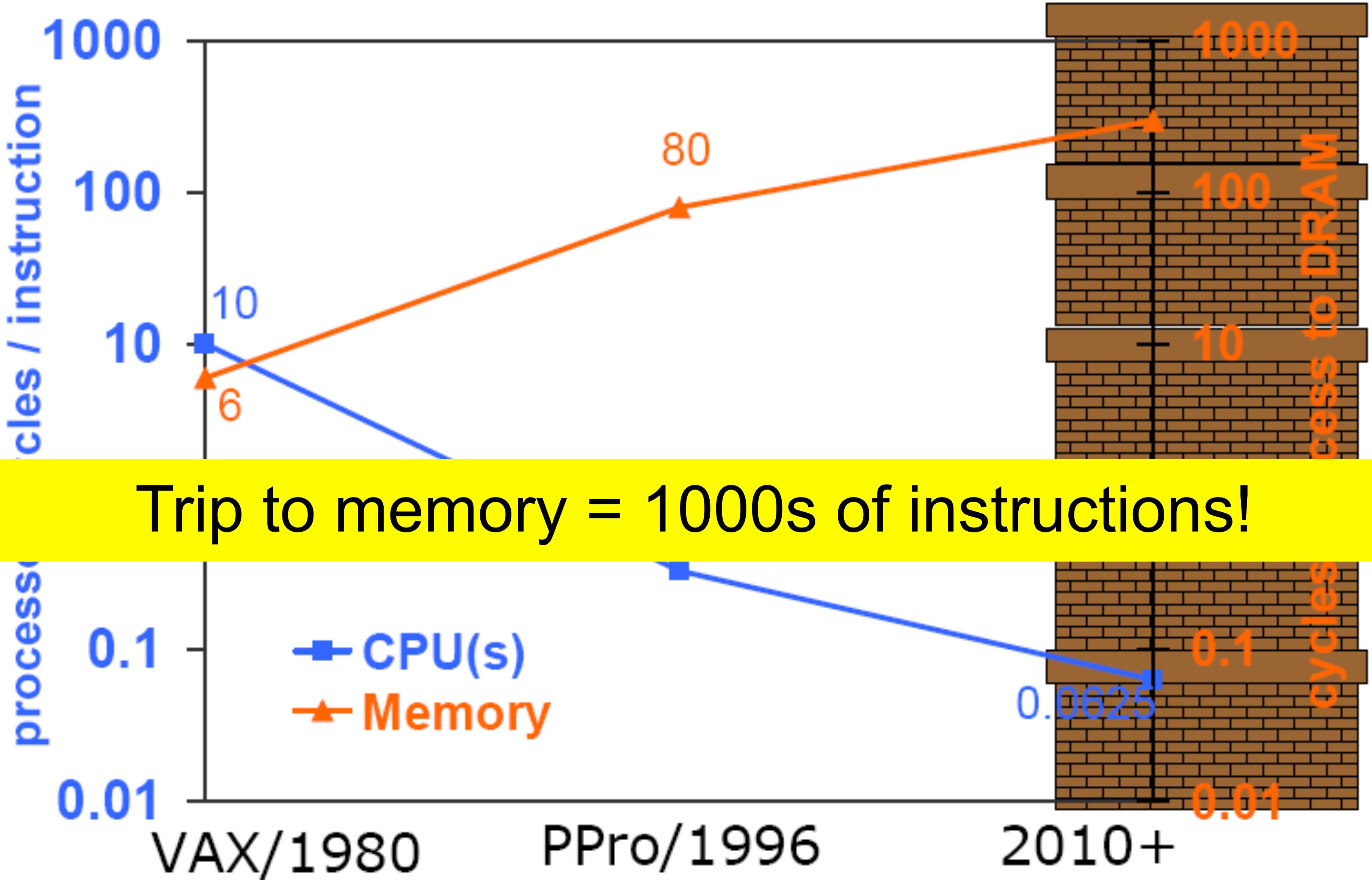
OLAP query: accesses only a few columns of almost all rows.

Databases hit The Memory Wall

- Detailed and exhaustive analysis for different workloads using 4 RDBMSs by Ailamaki, DeWitt, Hill,, Wood in VLDB 1999: "*DBMSs On A Modern Processor: Where Does Time Go?*"
- CPU is 60%-90% idle, waiting for memory:
  - L1 data stalls
  - L1 instruction stalls
  - L2 data stalls
  - TLB stalls
  - Branch mispredictions
  - Resource stalls



Hardware Changes: The Memory Wall



Trip to memory = 1000s of instructions!

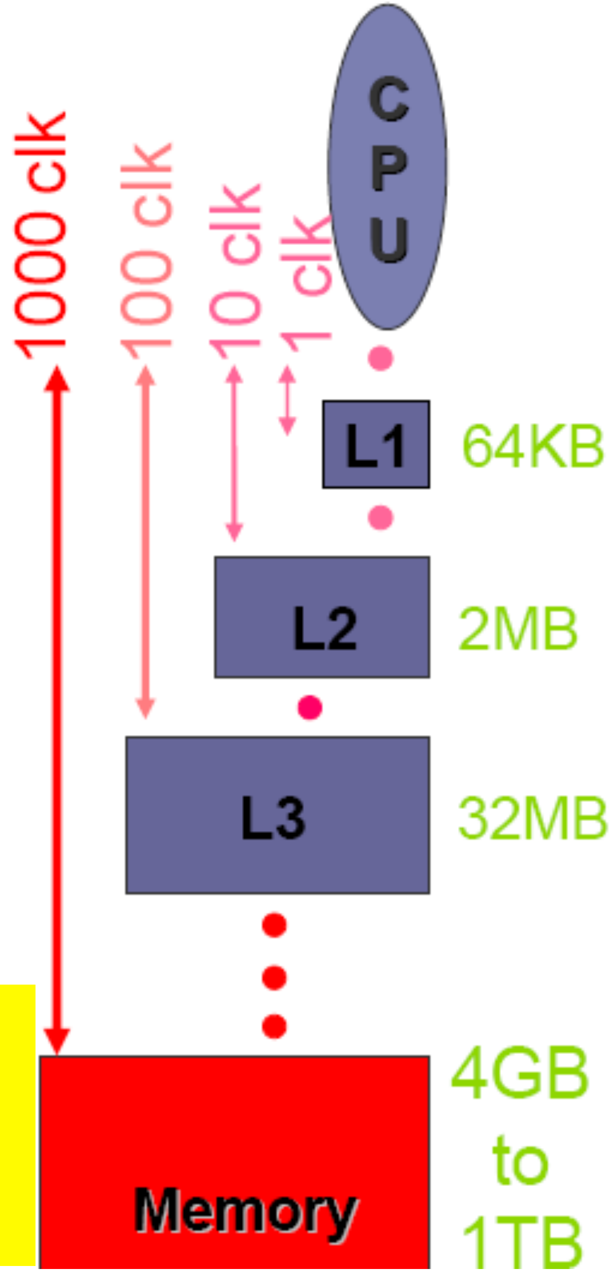
Hardware Changes: Memory Hierarchies

- Caches trade off capacity for speed
- Exploit instruction/data locality
- Demand fetch/wait for data

[ADH99]:

- Running top 4 database systems
- **At most 50% CPU utilization**

+Transition Lookaside Buffer (TLB)
Cache for VM address translation ➔
only 64 entries!



Evolution

It is not the strongest of the species that survives, nor the most intelligent, but the one most responsive to change.

Charles Darwin **(1809 - 1882)**

MonetDB

- Database kernel developed at CWI since 1993
  - *Research prototype turned into open-source product*
- Pioneering columnar database architecture
  - *Complete Relational/SQL & XML/XQuery DBMS*
- Design focus on large memory
  - *Data is kept persistent on disk and can exceed memory limits*
- Aiming at OLAP, BI & Data Mining workloads ("read-dominated")
  - *Supporting ACID transactions (WAL, optimistic CC)*
- Platform for database architecture research
  - *Used in academia (research & teaching) & commercial environments*
- Back-end for various DB research projects:

  Multi-Media DB & IR ("Tijah"), XML/XQuery ("Pathfinder"),

  Data Mining ("Proximity"), Digital Forensics ("XIRAF"),

  GIS ("OSM"), ...

How is MonetDB Different

- full vertical fragmentation: always!
  - everything in binary (2-column) tables (**B**inary **A**ssociation **T**able)
  - saves you from table scan hell in OLAP and Data Mining

- RISC approach to databases
  - simple back-end data model
  - simple back-end query language (binary/columnar relational algebra)
  - don't need (to pay for) a buffer manager => manage virtual memory
  - explicit transaction management => DIY approach to ACID
  -

- Multiple user data models & query languages
  - SQL, XML/XQuery, SciQL, RDF/SPARQL
  - front-ends map data models and query languages

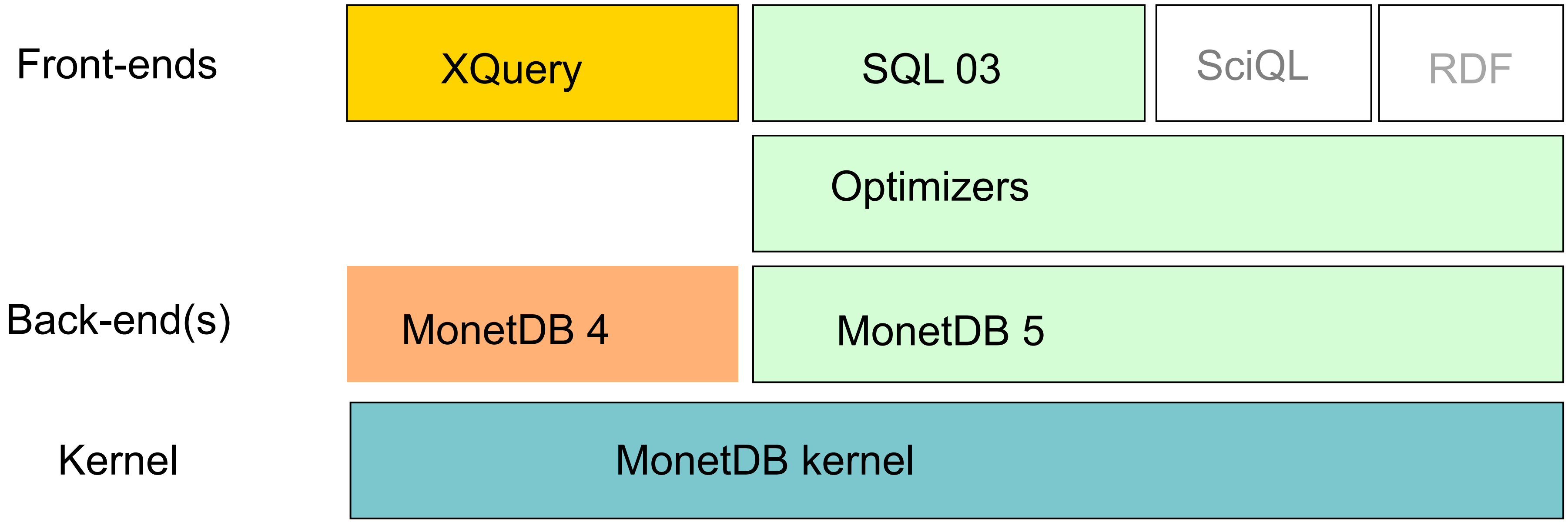
How is MonetDB Different

- optimized for large memory hierarchies
  - cache-conscious algorithms
  - exploit the persistence storage (disk,network,SSD)
- operator-at-a-time bulk processing
  - avoids tuple-at-a-time management overhead
- CPU and memory cache optimized
  - programming team experienced in main memory DBMS techniques
  - use of scientific programming optimizations (loop unrolling)

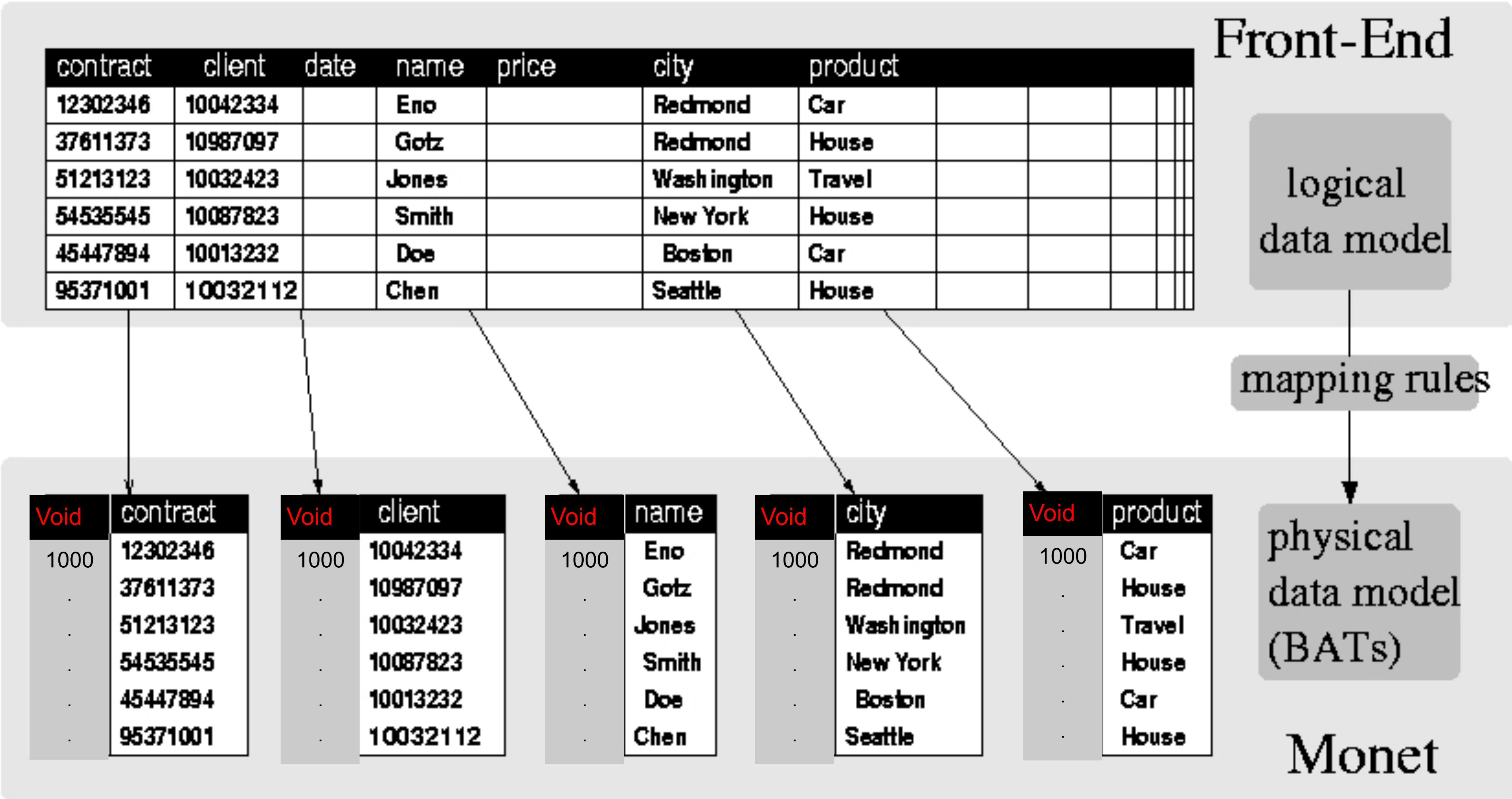
MonetDB vs Traditional DBMS Architecture

- **Architecture-Conscious Query Processing**
  - **vs Magnetic disk I/O conscious processing**
  - *Data layout, algorithms, cost models*
- **RISC Relational Algebra (operator-at-a-time)**
  - **vs Tuple-at-a-time Iterator Model**
  - *Faster through simplicity: no tuple expression interpreter*
- **Multi-Model: ODMG, SQL, XML/XQuery, ..., RDF/SPARQL**
  - **vs Relational with Bolt-on Subsystems**
  - *Columns as the building block for complex data structures*
- **Decoupling of Transactions from Execution/Buffering**
  - **vs ARIES integrated into Execution/Buffering/Indexing**
  - *ACID, but not ARIES.. Pay as you need transaction overhead.*
- **Run-Time Indexing and Query Optimization**
  - **vs Static DBA/Workload-driven Optimization & Indexing**
  - *Extensible Optimizer Framework;*
  - *cracking, recycling, sampling-based runtime optimization*

The MonetDB Software Stack

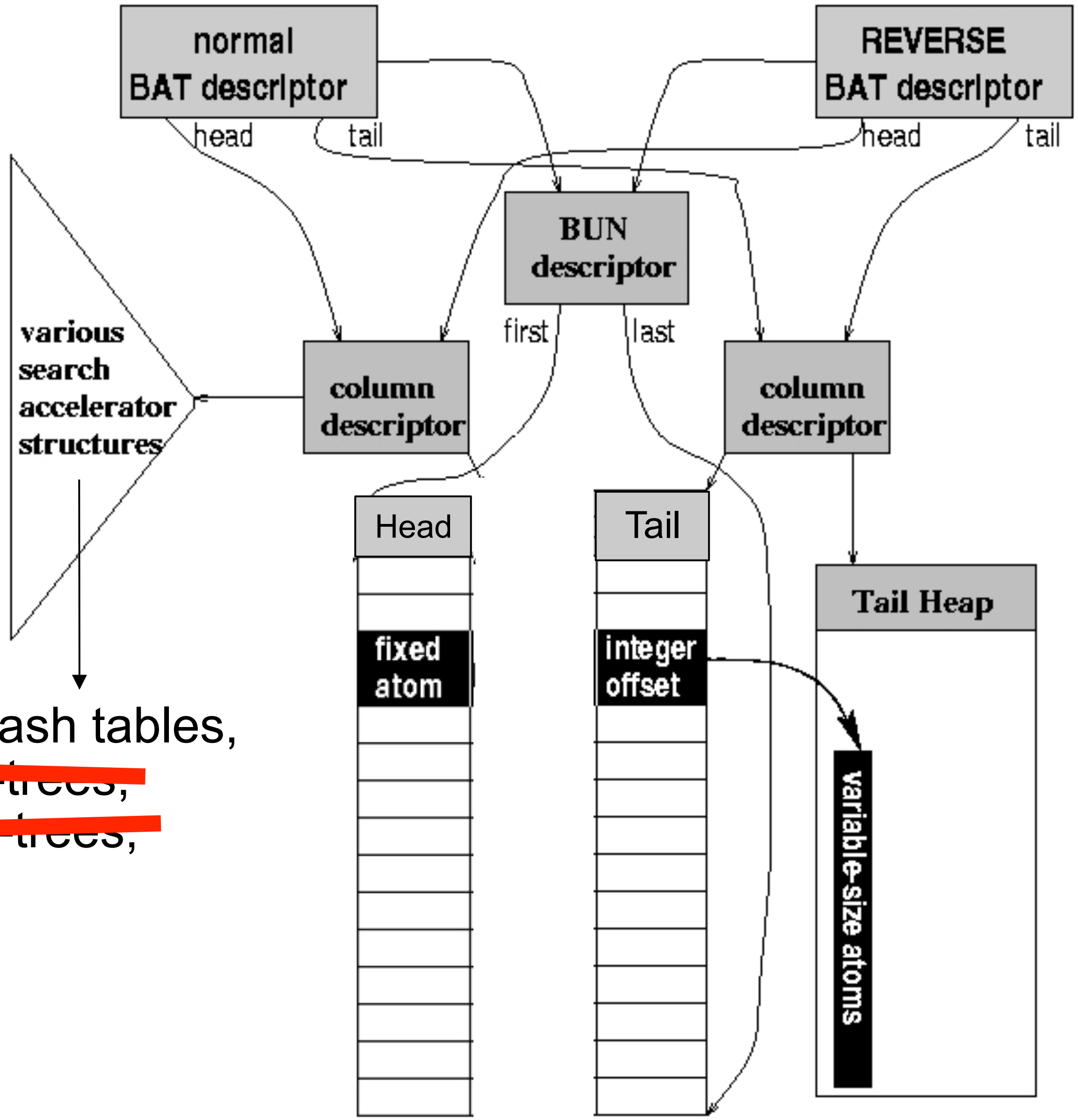


Storing Relations in MonetDB



Virtual OID: seqbase=**1000** (increment=1)

BAT Data Structure



Hash tables,
~~T-trees,~~
~~R-trees,~~
...

BAT:
binary association table

BUN:
binary unit

Head & Tail:
- consecutive memory blocks (arrays)
- memory-mapped files

Tail Heap:
- best-effort duplicate elimination for strings (~ dictionary encoding)

RISC Relational Algebra

SELECT	id, name, (age-30)*50 as bonus
FROM	people
WHERE	age > 30

```
batcalc_minus_int(int* res,
                  int* col,
                  int  val,
                      int n)
{
     for(i=0; i<n; i++)
          res[i] = col[i] - val;
}
```

CPU ☺? Give it "nice" code !

- few dependencies (control,data)
- CPU gets out-of-order execution
- compiler can e.g. generate SIMD

One loop for an entire column

- no per-tuple interpretation
- arrays: no record navigation
- better instruction cache locality

Simple, hard-coded semantics in operators

MATERIALIZED intermediate results

Processing Model (MonetDB Kernel)

- **Bulk processing:**
  - full materialization of all intermediate results

- **Binary (i.e., 2-column) algebra core:**
  - select, join, semijoin, outerjoin
  - union, intersection, diff (BAT-wise & column-wise)
  - group, count, max, min, sum, avg
  - reverse, mirror, mark

- **Runtime *operational* optimization:**
  - Choosing optimal algorithm & implementation according to input properties and system status

Processing Model (MonetDB Kernel)

- Heavy use of code expansion to reduce cost

1 algebra operator

select()

3 overloaded operators

select("=",value) select("between",L,H) select("fcn",parm)

10 operator algorithms

scan hash-lookup bin-search bin-tree pos-lookup

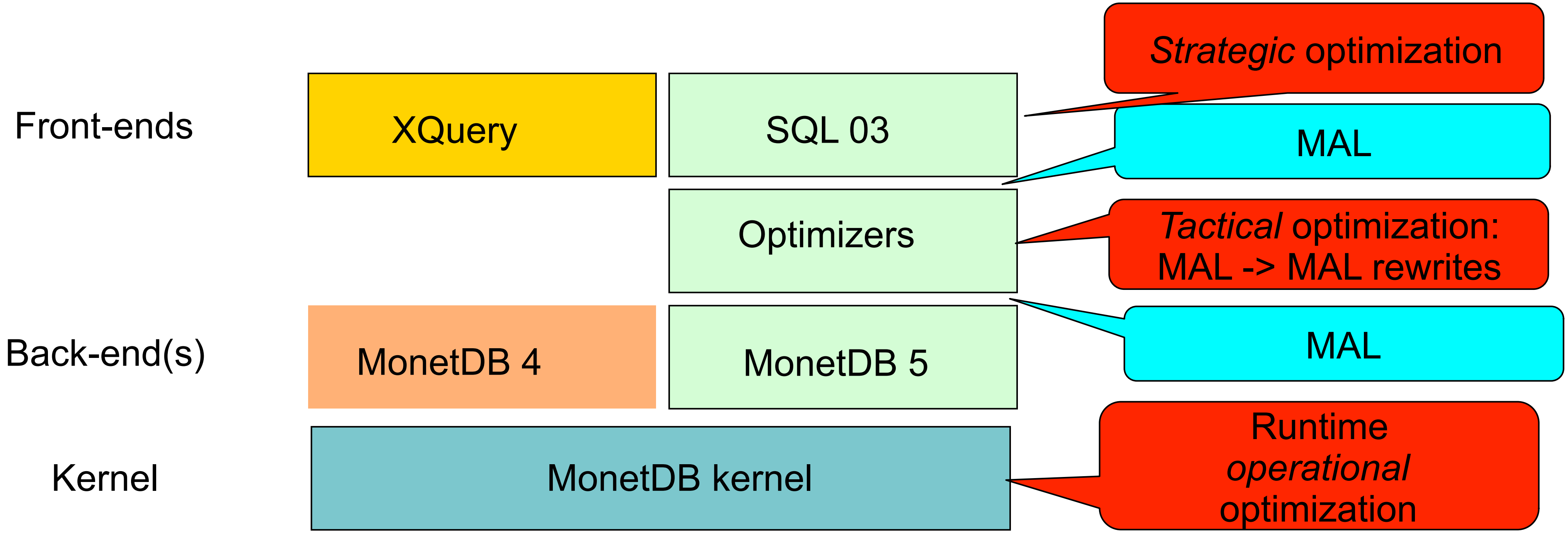
~1500(!) routines

(macro expansion)

scan_range_select_oid_int(), hash_equi_select_void_str(), …

- ~1500 selection routines
- 149 unary operations
- 335 join/group operations
- ...

The MonetDB Software Stack



MonetDB/5 Back-end: MAL

- **MAL: Monet Assembly Language**
  - textual interface
  - Interpreted language
- **Designed as system interface language**
  - Reduced, concise syntax
  - Strict typing
  - Meant for automatic generation and parsing/rewriting/processing
  - Not meant to be typed by humans
- **Efficient parser**
  - Low overhead
  - Inherent support for *tactical* optimization: MAL -> MAL
  - Support for optimizer plug-ins
  - Support for runtime schedulers
- **Binary-algebra core**
- **Flow control** (MAL is computational complete)

MonetDB Front-end: SQL

EXPLAIN SELECT a, z FROM t, s WHERE t.c = s.x;

```
function user.s2_1():void;
barrier _73 := language.dataflow();
    _2:bat[:oid,:int]  := sql.bind("sys","t","c",0);
    _7:bat[:oid,:int]  := sql.bind("sys","s","x",0);
    _10 := bat.reverse(_7);
    _11 := algebra.join(_2,_10);
    _13 := algebra.markT(_11,0@0);
    _14 := bat.reverse(_13);
    _15:bat[:oid,:int]  := sql.bind("sys","t","a",0);
    _17 := algebra.leftjoin(_14,_15);
    _18 := bat.reverse(_11);
    _19 := algebra.markT(_18,0@0);
    _20 := bat.reverse(_19);
    _21:bat[:oid,:int]  := sql.bind("sys","s","z",0);
    _23 := algebra.leftjoin(_20,_21);
exit _73;
    _24 := sql.resultSet(2,1,_17);
    sql.rsColumn(_24,"sys.t","a","int",32,0,_17);
    sql.rsColumn(_24,"sys.s","z","int",32,0,_23);
   _33 := io.stdout();
    sql.exportResult(_33,_24);
end s2_1;
```

MonetDB: MAL Optimizers

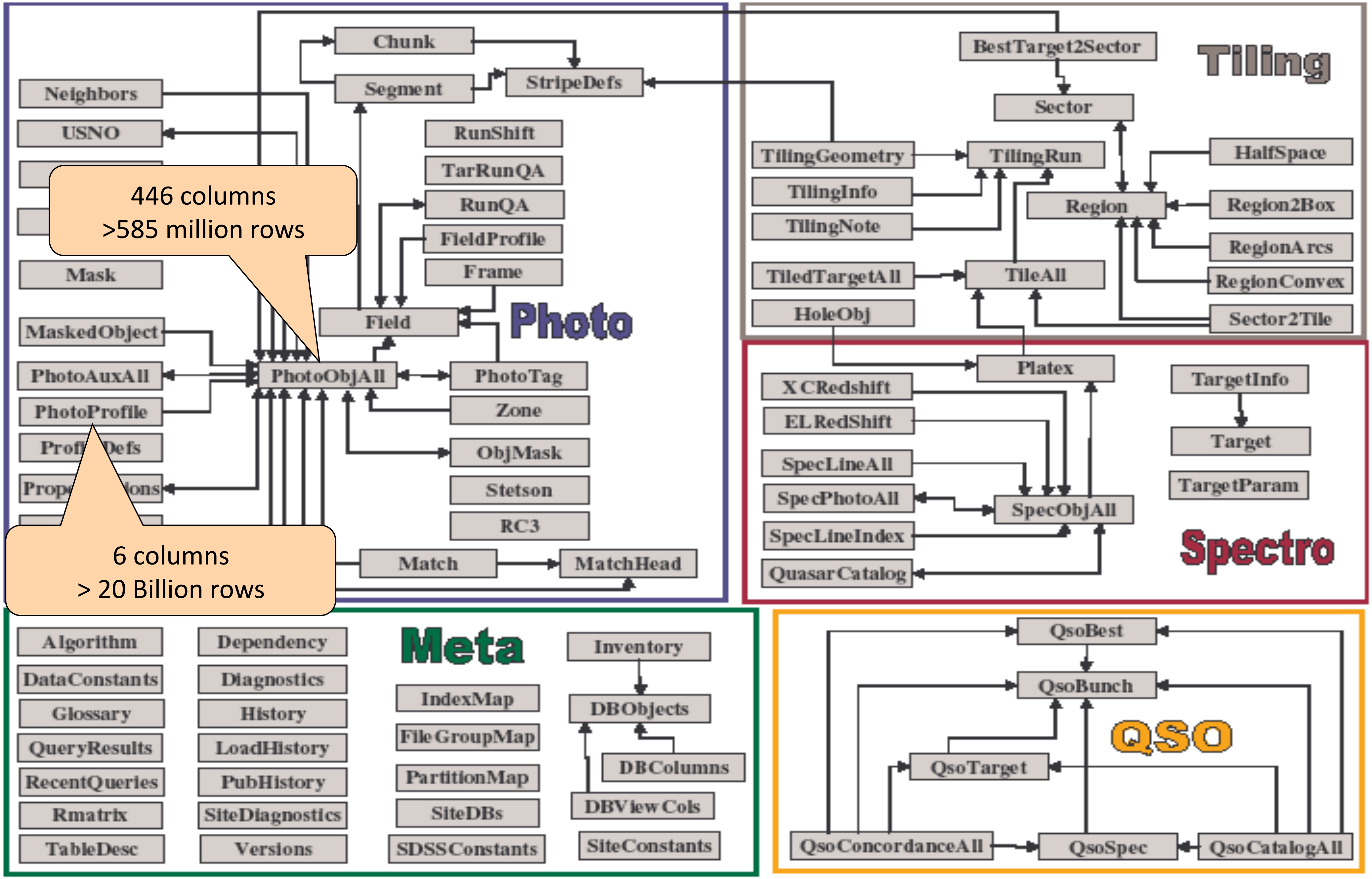
- **General front-end independent MAL -> MAL rewriting**
  - Implemented once, shared by all (future) front-ends
- **Examples**:
  - Constant propagation
  - Scalar expression evaluation
  - Dead-code elimination
  - Common sub-expression elimination
  - Reordering to optimize intermediate result usage
  - Reordering of linear (projection-) join chains
  - Parallelization:
    - Dataflow analysis
    - Horizontal partitioning
    - Remote execution
  - *Cracking*
  - *Recycling*
  - *...*

MonetDB Front-end: SQL

- **SQL 2003**
- **Parse SQL into logical n-ary relational algebra tree**
- **Translate n-ary relational algebra into logical 2-ary relational algebra**
- **Turn logical 2-ary plan into physical 2-ary plan (MAL program)**
  - Generate internal tree representation, not textual MAL program
- **Front-end specific *strategic* optimization:**
  - Heuristic optimization during all three previous steps
- **Primary key and distinct constraints:**
  - Create and maintain hash indices
- **Foreign key constraints**
  - Create and maintain foreign key join indices
- **Exploit both above indices during query evaluation**


File Edit View Go Bookmarks Tools Help
http://cas.sdss.org/dr6/en/
Go
Getting Started
Latest Headlines
MonetDB, Query Pro...
CWI
Sloan Digital Sky Survey / SkyServer
SDSS
Home
Tools
Schema
Projects
Astronomy
SDSS
Contact Us
Download
Site Search
Help
Welcome to the DR6 site!!
The Sixth Data Release is dedicated to Jim Gray for his fundamental contribution to the SDSS project and the extraordinary energy and passion he shared with everybody!
This website presents data from the Sloan Digital Sky Survey, a project to make a map of a large part of the universe. We would like to show you the beauty of the universe, and share with you our excitement as we build the largest map in the history of the world.
News
The site hosts data from Data Release 6 (DR6). What's new in DR6, what's new on this site, and known problems.
More...
For Astronomers
A separate branch of this website for professional astronomers (English)
More...
SDSS is supported by
NSF
NASA
MEXT
Powered by
Microsoft
Site Traffic
Privacy Policy
SkyServer Tools
Famous places
Get images
Visual Tools
Explore
Search
Object upload
CasJobs
Science Projects
Basic
Advanced
Challenges
For Kids
Games and Contests
Teachers
Links to other projects
Info Links
About Astronomy
About the SDSS
About the SkyServer
SDSS Data Release 6
SDSS Project Website
Open SkyQuery
Images of RC3 Galaxies
Help
Getting Started
FAQ
How To
Glossary
Schema Browser
Sample SQL Queries
Details of SDSS Data
Contact Us
Done

SkyServer Schema



CWI

Recycler
motivation & idea

"An architecture for recycling intermediates in a column-store". Ivanova, Kersten, Nes, Goncalves. ACM TODS 35(4), Dec. 2010

MonetDB

Motivation:

- scientific databases, data analytics
- Terabytes of data (observational , transactional)
- Prevailing read-only workload
- Ad-hoc queries with commonalities

Background:

- Operator-at-a-time execution paradigm
  - Automatic materialization of intermediates
- Canonical column-store organization
  - Intermediates have reduced dimensionality and finer granularity
  - Simplified overlap analysis

Recycling idea:

- instead of garbage collecting,
  keep the intermediates and reuse them
  - speed up query streams with commonalities
  - low cost and self-organization

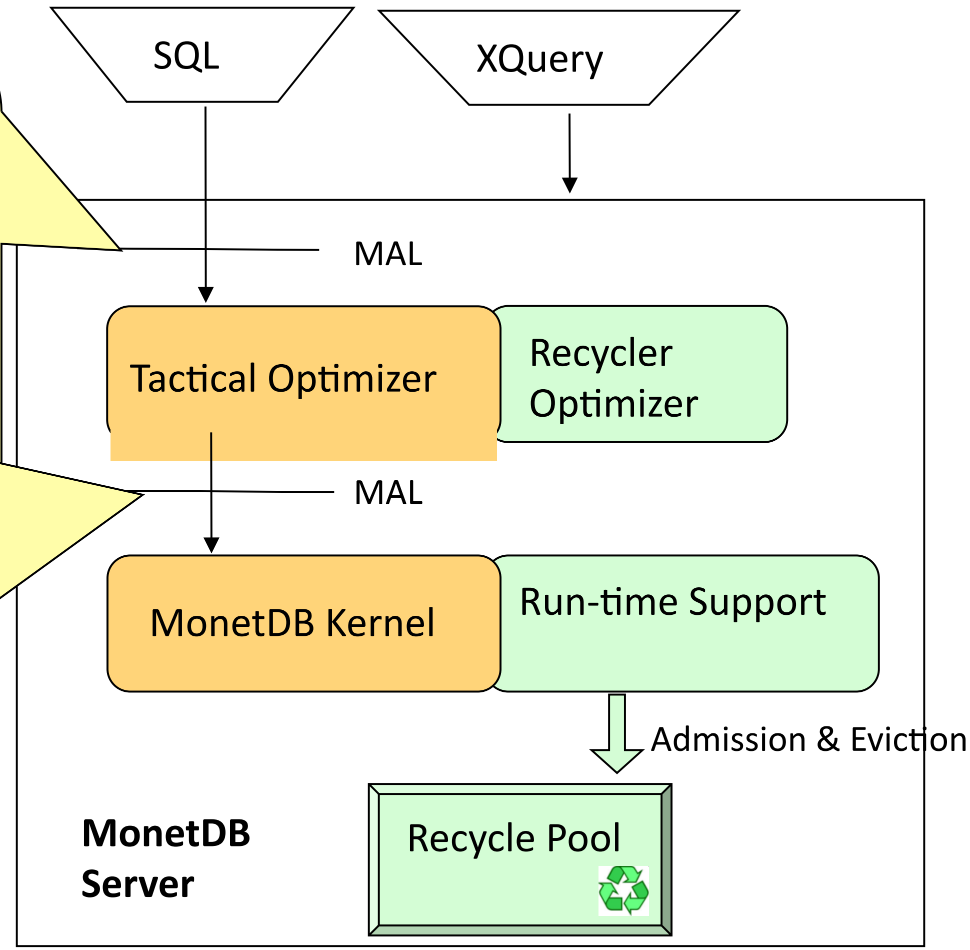
Recycler

fit into MonetDB

"An architecture for recycling intermediates in a column-store". Ivanova, Kersten, Nes, Goncalves. ACM TODS 35(4), Dec. 2010

```
function user.s1_2(A0:date, ...):void;
  X5 := sql.bind("sys","lineitem",...);
  X10 := algebra.select(X5,A0);
  X12 := sql.bindIdx("sys","lineitem",...);
  X15 := algebra.join(X10,X12);
  X25 := mtime.addmonths(A1,A2);
  ...
```

```
function user.s1_2(A0:date, ...):void;
  X5 := sql.bind("sys","lineitem",...);
  X10 := algebra.select(X5,A0);
  X12 := sql.bindIdx("sys","lineitem",...);
  X15 := algebra.join(X10,X12);
  X25 := mtime.addmonths(A1,A2);
  ...
```



CWI

Recycler
instruction matching

“An architecture for recycling intermediates in a column-store”. Ivanova, Kersten, Nes, Goncalves. ACM TODS 35(4), Dec. 2010

MONETDB

Run time comparison of

- instruction types
- argument values

Exact matching

```
Y3 := sql.bind("sys","orders","o_orderdate",0);
```

```
X1 := sql.bind("sys","orders","o_orderdate",0);
…
```

Name	Value	Data type	Size
X1	10	:bat[:oid,:date]	
T1	“sys”	:str	
T2	“orders”	:str	
…			

Recycler

instruction subsumption

"An architecture for recycling intermediates in a column-store". Ivanova, Kersten, Nes, Goncalves. ACM TODS 35(4), Dec. 2010

```
Y3 := algebra.select(X1,20,45);
```

```
X3 := algebra.select(X1,10,80);
…
X5 := algebra.select(X1,20,60);
```

Name	Value	Data type	Size
X1	10	:bat[:oid,:int]	2000
X3	130	:bat[:oid,:int]	700
X5	150	:bat[:oid,:int]	350
…			

Recycler
admission policies

“An architecture for recycling intermediates in a column-store”. Ivanova, Kersten, Nes, Goncalves. ACM TODS 35(4), Dec. 2010

Decide about storing the results

- KEEPALL
  - all instructions advised by the optimizer
- CREDIT
  - instructions supplied with credits
  - storage ‘paid’ with 1 credit
  - reuse returns credits
  - lack of reuse limits admission and resource claims

Recycler
cache policies

“An architecture for recycling intermediates in a column-store”. Ivanova, Kersten, Nes, Goncalves. ACM TODS 35(4), Dec. 2010

Decide about eviction of intermediates

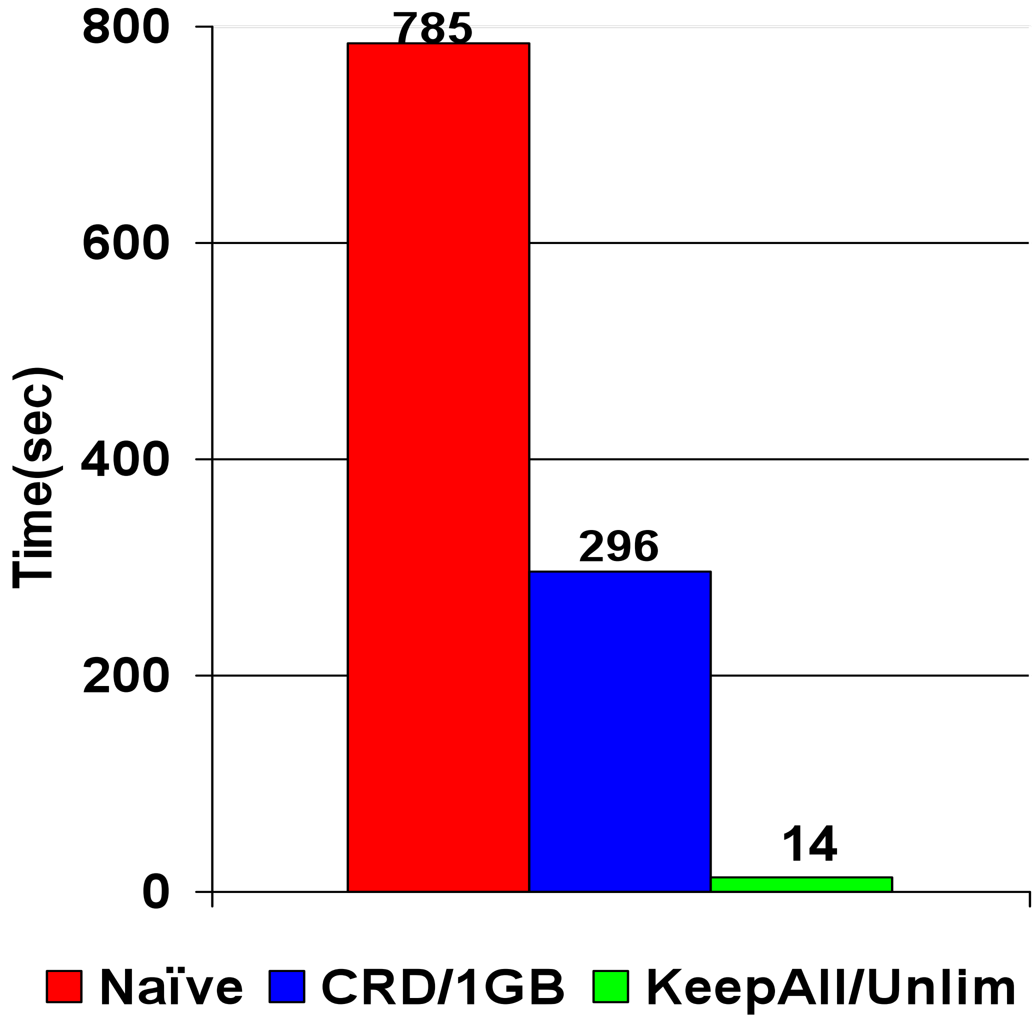
- Pick instructions with smallest utility
  - LRU : time of computation or last reuse
  - BENEFIT : estimated contribution to performance: CPU and I/O costs, recycling
- Triggered by resource limitations (memory or entries)

Recycler
SkyServer evaluation

"An architecture for recycling intermediates in a column-store". Ivanova, Kersten, Nes, Goncalves. ACM TODS 35(4), Dec. 2010

Sloan Digital Sky Survey / SkyServer
http://cas.sdss.org

- 100 GB subset of DR4
- 100-query batch from January 2008 log
- 1.5GB intermediates, 99% reuse
- Join intermediates major consumer of memory and major contributor to savings



Project portfolio

Commercial: 0.5 PB telco warehouse

Commercial: DNA warehouses

LOFAR : Transient detection

Emili: Streaming in sensor-based systems

TELEIOS: Remote sensing virtual observatory

Planetdata,Lod2: Semantic web, linked open data

NWO: Biased sampling for science database

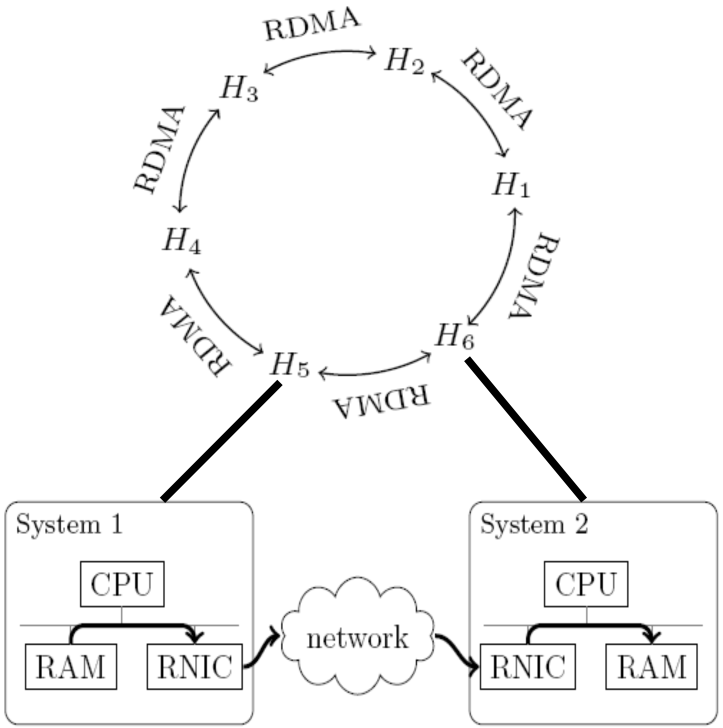
SciLens: Dissemination and coordination

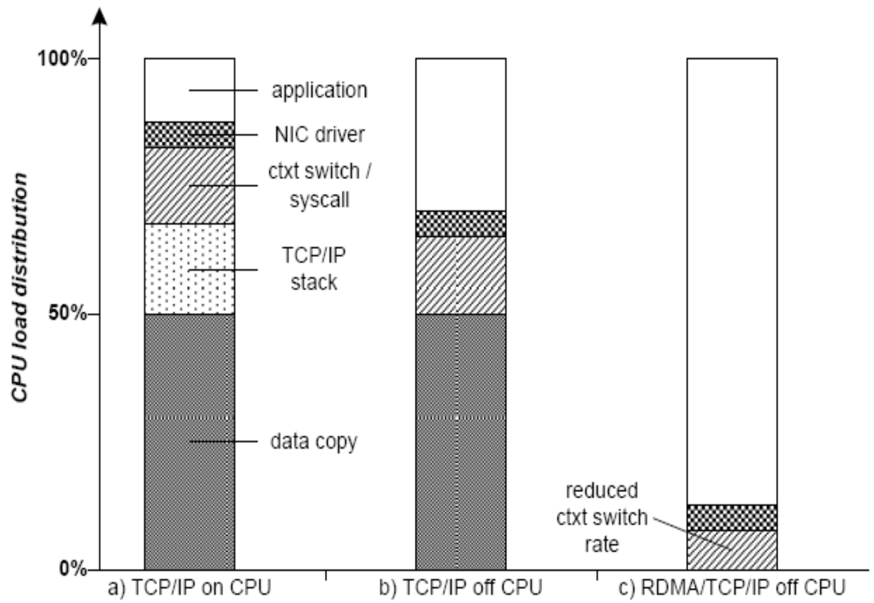
Datacyclotron: Novel distributed architectures

Remote Direct Memory Access (RDMA)

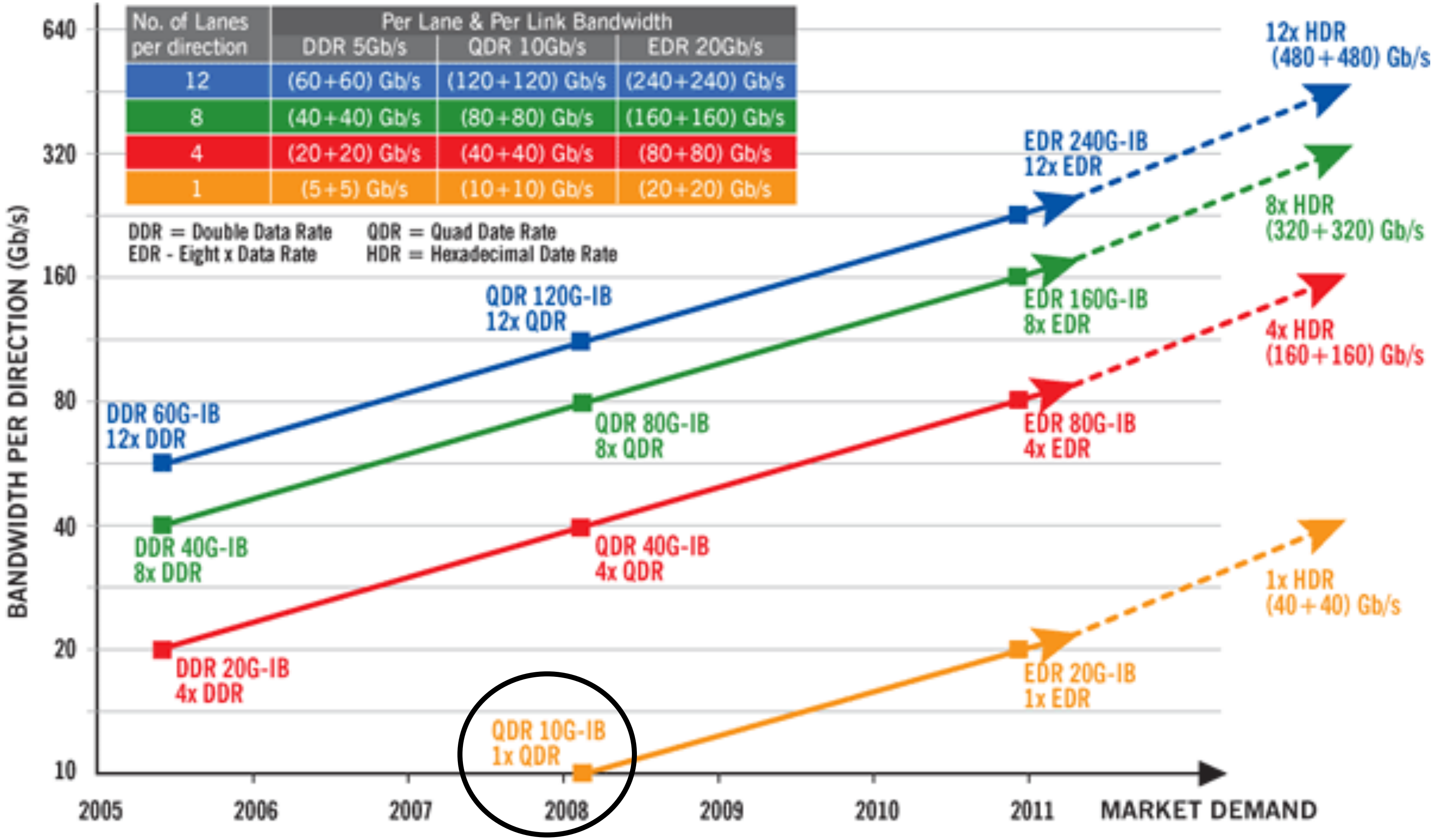
- Remote Memory at Your Finger Tips.

- RDMA Benefits.
  - Cpu Load
  - Reduced Memory Bus Traffic



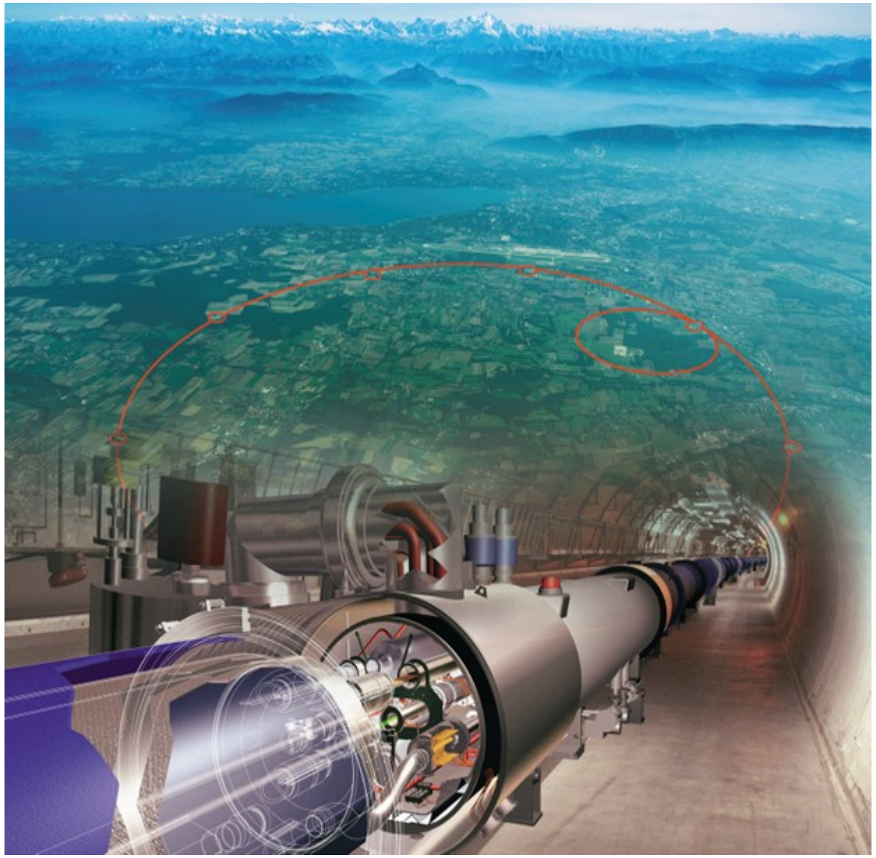


Road-map for RDMA



The topology.

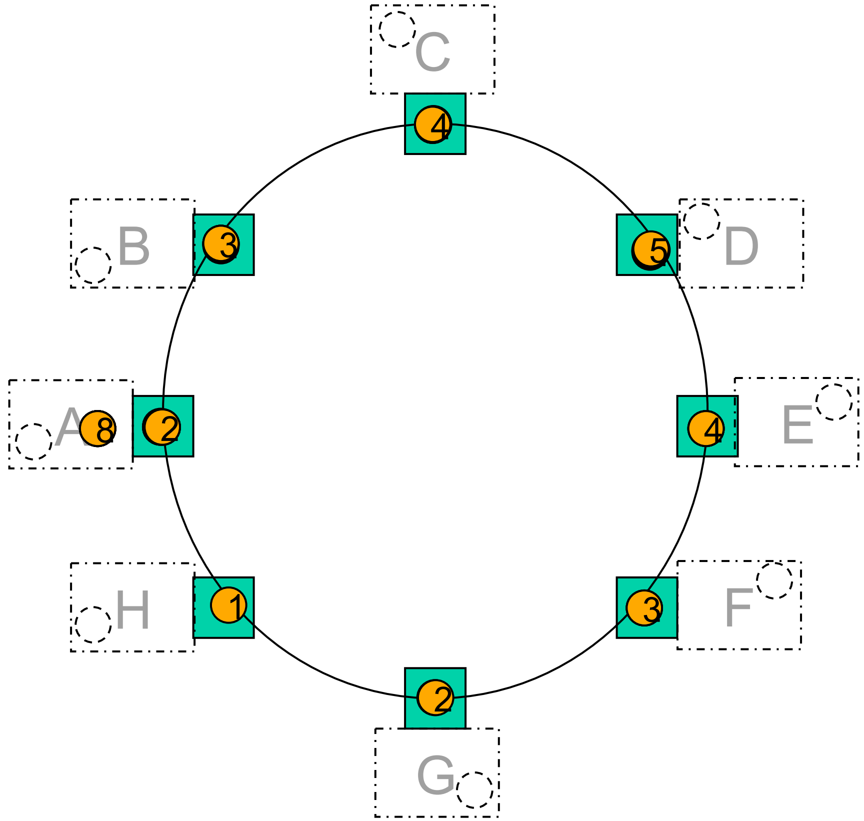
- Swiss one (LHC)



- Dutch one (DaCy)

The data acceleration.

- A chunk is loaded by a node into the ring.
- It flows clockwise...
- It continuously hops from node to node... until it is removed...



MonetDB SciQL

SciQL (*pronounced 'cycle'*)

- A backward compatible extension of SQL'03
- Symbiosis of relational and array paradigm
- Flexible structure-based grouping
- Capitalizes the MonetDB array storage
  - Recycling, an adaptive 'materialized view'
  - Zero-cost attachment contract for cooperative clients

MonetDB Vaults

A contract between MonetDB and file repository of volumeous scientific data

- provide seamless SQL access to foreign file formats using SciQL views
- zero cost, adaptive loading and replication
- Capitalize libraries as UDFs (linpack, R,..)
- Short term targets:
  - MSEED, FITS, NETCDF, csv

MonetDB Octopus

- Distributed SQL processing without a DBA
- Merovingian, managing a cluster of servers
- Cloud-based infrastructure with fail-over
- Partial/full replication adaptive to query needs.
- Recycling, a basis for distribution and load scheduling

eScience- landscape

Science Domain Workbench

General Workflow Systems

General Purpose Database Management

Data acquisition systems

Data scrubbing, cleaning

Data refinement, enrichment

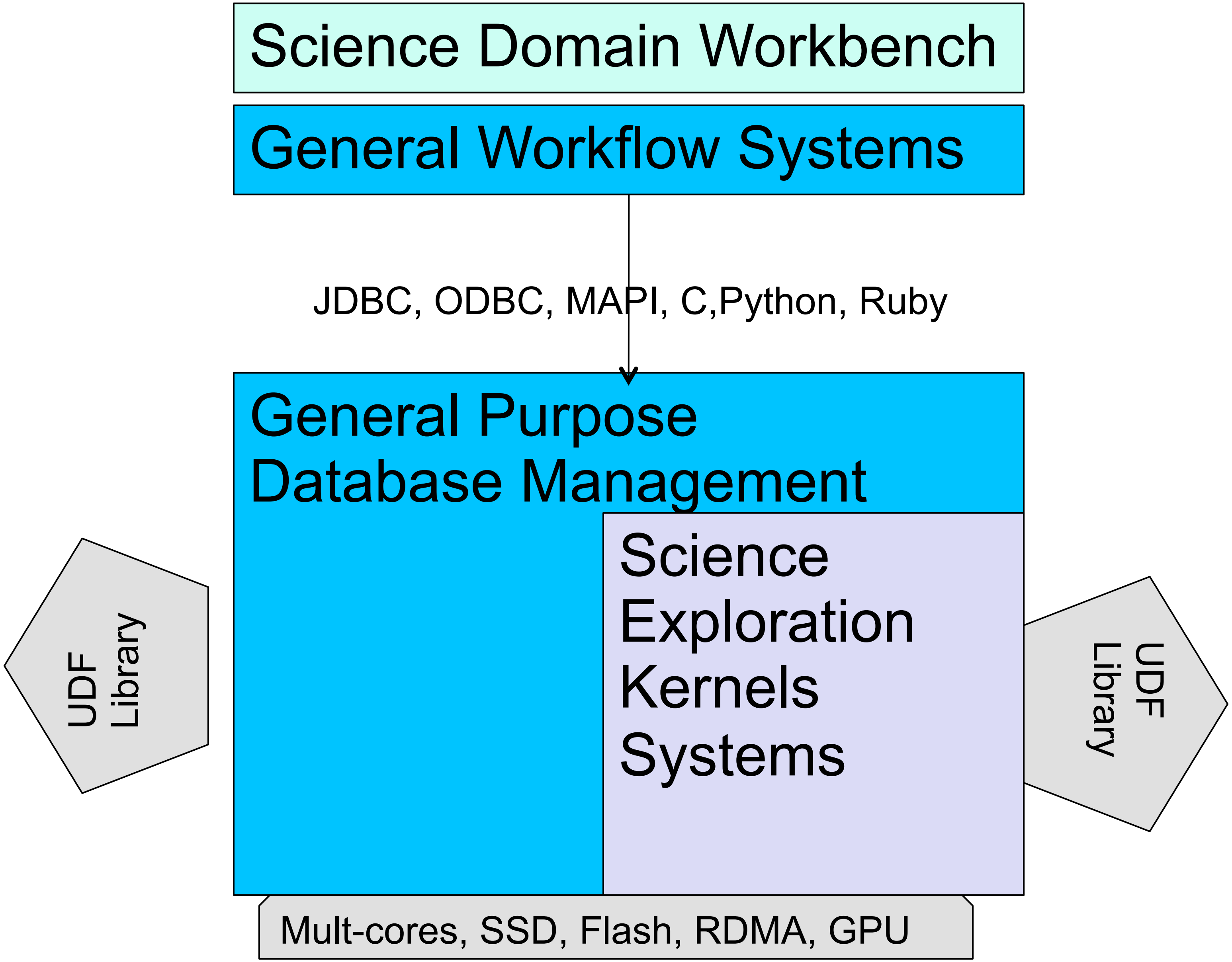
Data catalogues, meta-data

Data exploration, mining

Data visualisation

Science Exploration Database Systems

MonetDB for Science



Science DBMS

	MonetDB 5.23	SciDB 0.5
Open source	Mozilla License	GPL 3.0 + Commercial
Downloads	>12.000 /month	Tens up to now
SQL compliance	SQL 2003	??
Interoperability	JDBC, ODBC, MAPI, C, Python, Ruby, C++	C++ UDF
Array model	SciQL	AQL
Science support	Linked libraries	Linked libraries
Foreign files	Vaults to FITS, NETCDF, MSEED	??
Distribution	50 node cluster Octopus 200 node cluster Cyclotron	4 node cluster
Distribution tech	Dynamic partial replication	Static fragmentation
	Distributed query, map-reduce, streaming, multi-core	Map-reduce
Largest local demo	Skyserver SDSS 6 3TB	---

Open-Source Development

- Feature releases: 3-4 per year
  - Research results
  - User requests
- Bug-fix releases: monthly
- QA
  - Automated nightly testing on >20 platforms
  - Ensure correctness & stability
  - Ensure portability
  - Bug reports become test cases
  - Semi-automatic performance monitoring
  - Passed static code verification by Coverity with only minor problems

File Edit View History Bookmarks Tools Help

http://monetdb.cwi.nl/testing/projects/monetdb/Stable/sql/.mTests103/index.html

Google

Most Visited Release Notes Fedora Project Red Hat Free Content

clients Stable t... | MonetDB4 Sta... | MonetDB5 Sta... | template/4 Sta... | template/5 Sta... | sql Stable test ... | geom Stable t... | pathfinder/ Sta... | pathfinder/g St...

Testing results of 2010.01.17_00-09-02 (on various platforms):

"stdout stderr": "o" = No differences. "x" = Minor differences. "X" = *Major differences*. "K" = *Killed*. "T" = *Timeout*. "S" = *Socket not released*. "C" = *Crash*. "-" = No output.

Below: Complete list showing all tests. (*Alternative:* Shortened list showing only tests with unexpected outcome on at least one platform.)

	Debian4.0 G.32.32.d.1	Fedora10 G.32.32.d.1	Fedora10 G.64.32.d.1	Fedora10 G.64.64.d.0	Fedora10 G.64.64.d.1	Fedora10 G.64.64.s.1	Fedora10 I.64.32.d.1	Fedora10 I.64.64.d.1	Fedora11 G.64.64.d.0	Gentoo2.0.1 G.64.64.d.1	SunOS5.10 G.32.32.d.0	SunOS5.10 G.64.32.d.0	SunOS5.10 G.64.64.d.0	SunOS5.11 G.32.32.d.1	SunOS5.11 G.64.64.d.1	SunOS5.11 S.32.32.d.1	Windows5.1 I.32.32.d.1	Windows6.0 I.64.32.d.1	Windows6.0 I.64.64.d.1	Windows6.0 M.32.32.d.1	Windows6.0 M.64.32.d.1	Windows6.0 M.64.64.d.1	
conf	Debian4.0 G.32.32.d.1	Fedora10 G.32.32.d.1	Fedora10 G.64.32.d.1	Fedora10 G.64.64.d.0	Fedora10 G.64.64.d.1	Fedora10 G.64.64.s.1	Fedora10 I.64.32.d.1	Fedora10 I.64.64.d.1	Fedora11 G.64.64.d.0	*Gentoo2.0.1 G.64.64.d.1*	SunOS5.10 G.32.32.d.0	SunOS5.10 G.64.32.d.0	SunOS5.10 G.64.64.d.0	SunOS5.11 G.32.32.d.1	SunOS5.11 G.64.64.d.1	SunOS5.11 S.32.32.d.1	Windows5.1 I.32.32.d.1	Windows6.0 I.64.32.d.1	Windows6.0 I.64.64.d.1	Windows6.0 M.32.32.d.1	Windows6.0 M.64.32.d.1	Windows6.0 M.64.64.d.1	*conf*
monetdb-sql-conds	o o	- -	o o	o o	o o	- -	o o	o o	o o	X o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	*monetdb-sql-conds*
src/backends/monet5	Debian4.0 G.32.32.d.1	Fedora10 G.32.32.d.1	Fedora10 G.64.32.d.1	Fedora10 G.64.64.d.0	Fedora10 G.64.64.d.1	Fedora10 G.64.64.s.1	Fedora10 I.64.32.d.1	Fedora10 I.64.64.d.1	Fedora11 G.64.64.d.0	Gentoo2.0.1 G.64.64.d.1	SunOS5.10 G.32.32.d.0	SunOS5.10 G.64.32.d.0	SunOS5.10 G.64.64.d.0	SunOS5.11 G.32.32.d.1	SunOS5.11 G.64.64.d.1	SunOS5.11 S.32.32.d.1	Windows5.1 I.32.32.d.1	Windows6.0 I.64.32.d.1	Windows6.0 I.64.64.d.1	Windows6.0 M.32.32.d.1	Windows6.0 M.64.32.d.1	Windows6.0 M.64.64.d.1	src/backends/monet5
optimizers	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	optimizers
Mbeddedsql5--help	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	Mbeddedsql5--help
src/benchmarks/ATIS	Debian4.0 G.32.32.d.1	Fedora10 G.32.32.d.1	Fedora10 G.64.32.d.1	Fedora10 G.64.64.d.0	Fedora10 G.64.64.d.1	Fedora10 G.64.64.s.1	Fedora10 I.64.32.d.1	Fedora10 I.64.64.d.1	Fedora11 G.64.64.d.0	Gentoo2.0.1 G.64.64.d.1	SunOS5.10 G.32.32.d.0	SunOS5.10 G.64.32.d.0	SunOS5.10 G.64.64.d.0	SunOS5.11 G.32.32.d.1	SunOS5.11 G.64.64.d.1	SunOS5.11 S.32.32.d.1	Windows5.1 I.32.32.d.1	Windows6.0 I.64.32.d.1	Windows6.0 I.64.64.d.1	Windows6.0 M.32.32.d.1	Windows6.0 M.64.32.d.1	Windows6.0 M.64.64.d.1	src/benchmarks/ATIS
cache	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	cache
load	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	load
select_simple_join	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	select_simple_join
select_join	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	select_join
select_key_prefix_join	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	select_key_prefix_join
select_distinct	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	select_distinct
select_group	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	select_group
src/benchmarks/arno	Debian4.0 G.32.32.d.1	Fedora10 G.32.32.d.1	Fedora10 G.64.32.d.1	Fedora10 G.64.64.d.0	Fedora10 G.64.64.d.1	Fedora10 G.64.64.s.1	Fedora10 I.64.32.d.1	Fedora10 I.64.64.d.1	Fedora11 G.64.64.d.0	Gentoo2.0.1 G.64.64.d.1	SunOS5.10 G.32.32.d.0	SunOS5.10 G.64.32.d.0	SunOS5.10 G.64.64.d.0	SunOS5.11 G.32.32.d.1	SunOS5.11 G.64.64.d.1	SunOS5.11 S.32.32.d.1	Windows5.1 I.32.32.d.1	Windows6.0 I.64.32.d.1	Windows6.0 I.64.64.d.1	Windows6.0 M.32.32.d.1	Windows6.0 M.64.32.d.1	Windows6.0 M.64.64.d.1	src/benchmarks/arno
create	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	create
check0	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	check0
insert_MODEL	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	insert_MODEL
insert_ATOM	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	insert_ATOM
insert_BOND	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	insert_BOND
check1	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	check1
01	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	01
02	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	02
03	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	03
04	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	04
05	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	05
06	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	06
07	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	07
08	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	08
09	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	09
10	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	10
11	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	11
12	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	12
13	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	13
14	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	14
15	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	15
16	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	16
17	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	17
18	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	18
19	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	19
20	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	20
21	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	21
22	o o	- -	o o	o o	o o	- -	o o	o o	o o	o o	o o	o o	o o	- -	o o	- -	o o	o o	o o	- -	- -	- -	22

Done

Stefan Manegold | 9292ov.nl: OV-reisinfo... | sql Stable make Sun J... | sql Stable test Sun Jan ... | 2 °C | 6 °C | Sun Jan 17, 16:28:32

MonetDB,

full-functional open-source product

a mature modern column-store

proven track record in (science) applications

strong and committed development team

close interaction with application developers

Reaching the stars step by step