Analysis of the Blocking Behaviour of Schema Transformations in Relational Database Systems

Lesley Wevers
Menno Tammens
Matthijs Hofstra
Marieke Huisman
Maurice van Keulen

University of Twente

ADBIS 2015

What is the problem?

WikiMedia schema revisions:

Source: http://yellowstone.cs.ucla.edu/schema-evolution/index.php/Schema_Evolution_Benchmark

Curino et al.

What is the problem?

WikiMedia schema revisions:

- 90% require a write lock.

Source: http://yellowstone.cs.ucla.edu/schema-evolution/index.php/Schema_Evolution_Benchmark

Curino et al.

What is the problem?

WikiMedia schema revisions:

- 90% require a write lock.
- Largest took 22 hours to complete for wikipedia.

Source: http://yellowstone.cs.ucla.edu/schema-evolution/index.php/Schema_Evolution_Benchmark

Curino et al.

What is the problem?

Some systems can not go offline:

- Telecom systems
- Payment systems
- Airline reservation systems
- Online services

Contributions

- An experimental investigation of the existing solutions for online schema changes.
  - What solutions are currently available?
  - To what degree do these work?
  - What are the open problems?
- A solution sketch for complex online schema transformations.

Earlier work

- Requirements for online schema changes
- Important classes of relational transformations
- A benchmark for online schema transformations based on TPC-C

Lesley Wevers, Menno Tammens, Matthijs Hofstra, Marieke Huisman and Maurice van Keulen. A Benchmark for Non-blocking Schema Transformations. DATA 2015.

Requirements for schema changes

Functionality

Transformations should:

- Provide ACID guarantees
- Be specified declaratively
- Include program updates

Performance

Transformations should:

- Have minimal impact on concurrent transactions:
  - Blocking
  - Slowdown
  - Aborts
- Commit as fast as possible

Schema Transformations

Semantic:

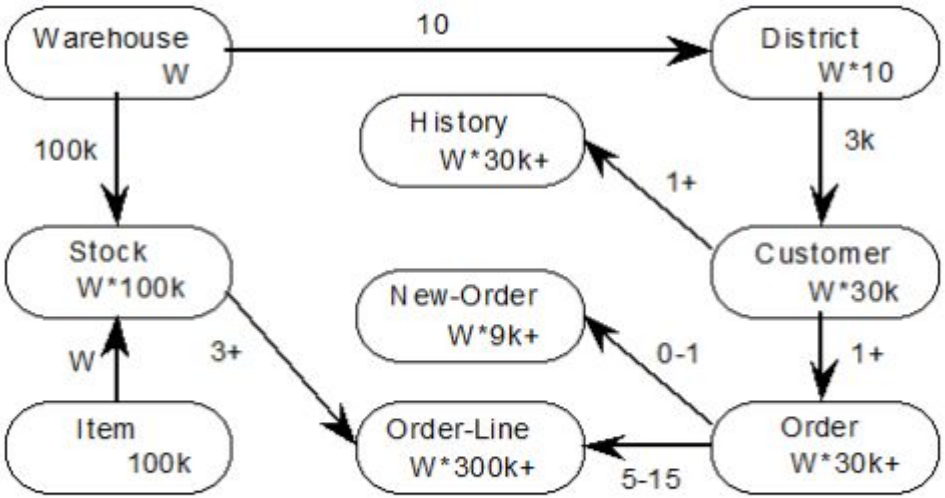
- Add / remove:
  - Relations
  - Columns
  - Constraints
- Change the cardinality of a relationship
- Use surrogate keys instead of natural keys
- ...

Implementation / performance:

- Add / remove indices
- Precompute aggregates
- Change normalization
- Change encoding of relationships
- ...

Benchmark

TPC-C



New order

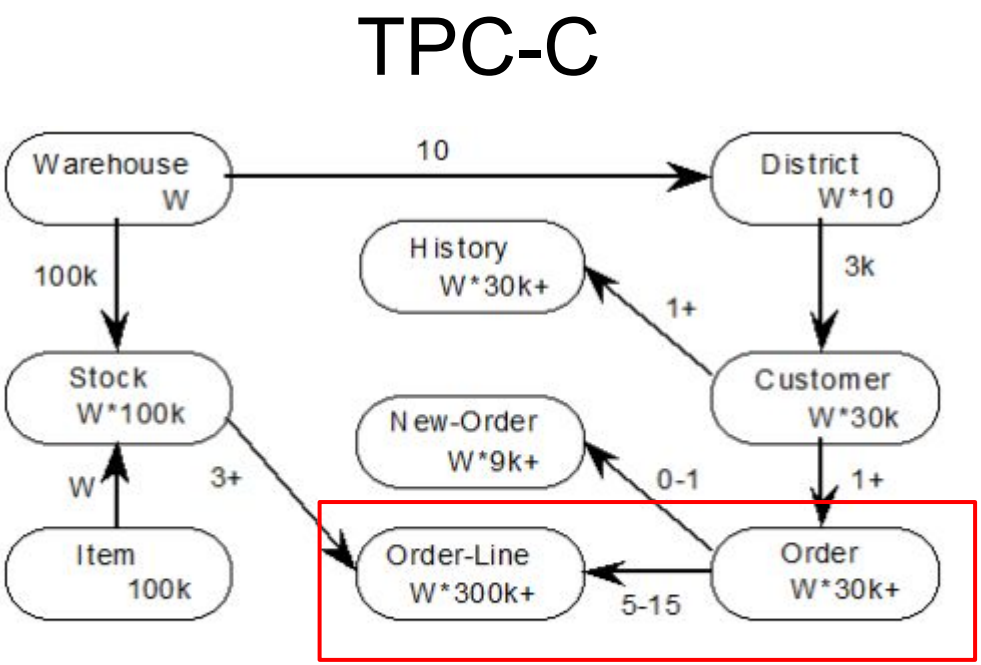
Payment

Order status

Delivery

Stock level

Benchmark



New order

Payment

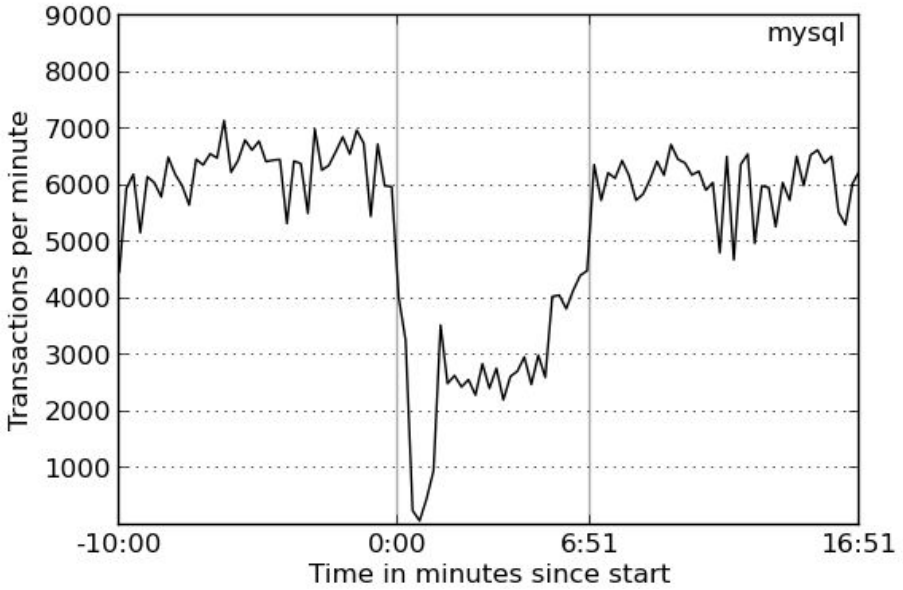
Order status

Delivery

Stock level

Benchmark Process

- Setup database
- Start TPC-C
- Intro period
- Transform:
  - Schema
  - Stored procedures
- Outro period
- Stop TPC-C



Contenders

Data definition language:

- PostgreSQL: Partial Instantaneous DDL
- MySQL: Partial Online DDL
- Oracle 11g

Ronströms method:

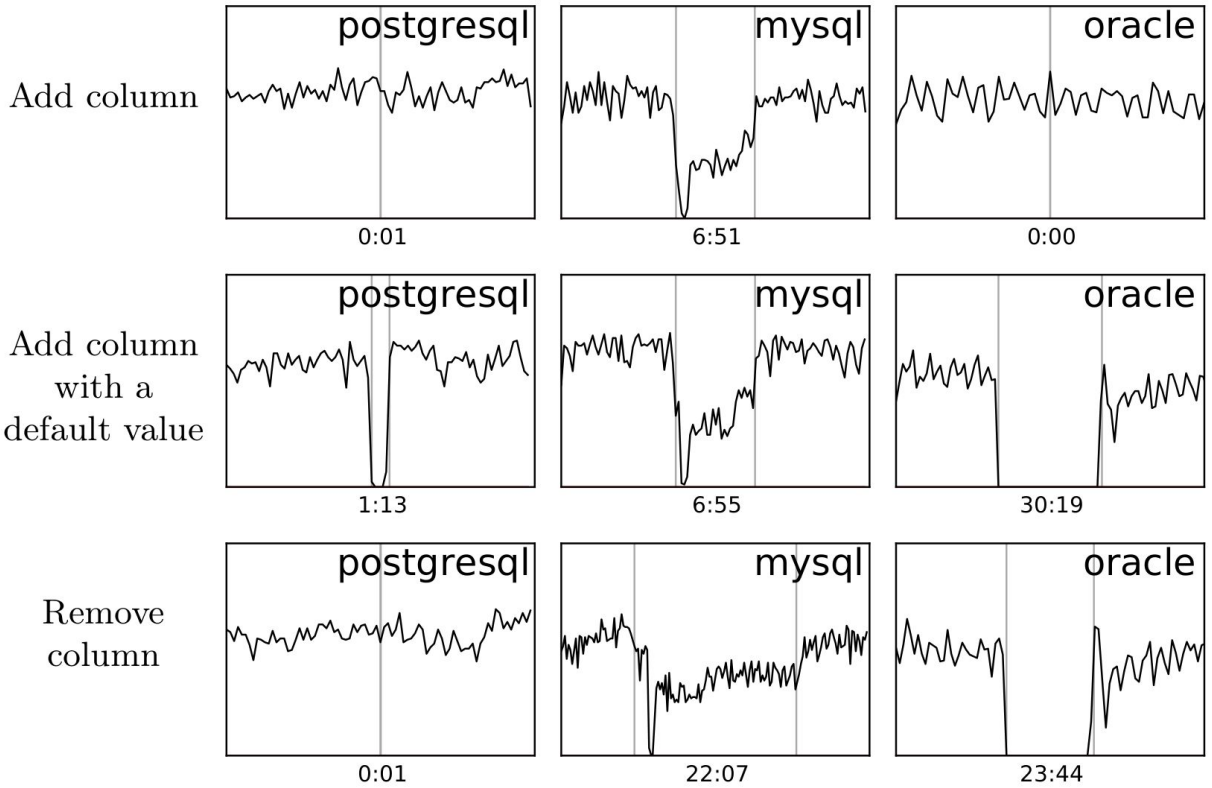
- Third-party tools (MySQL):
  - **pt-online-schema-change**
  - oak-online-alter-table
  - Facebook’s online-schema-change
- Oracle’s DBMS_PARALLEL_COPY

Experimental Setup

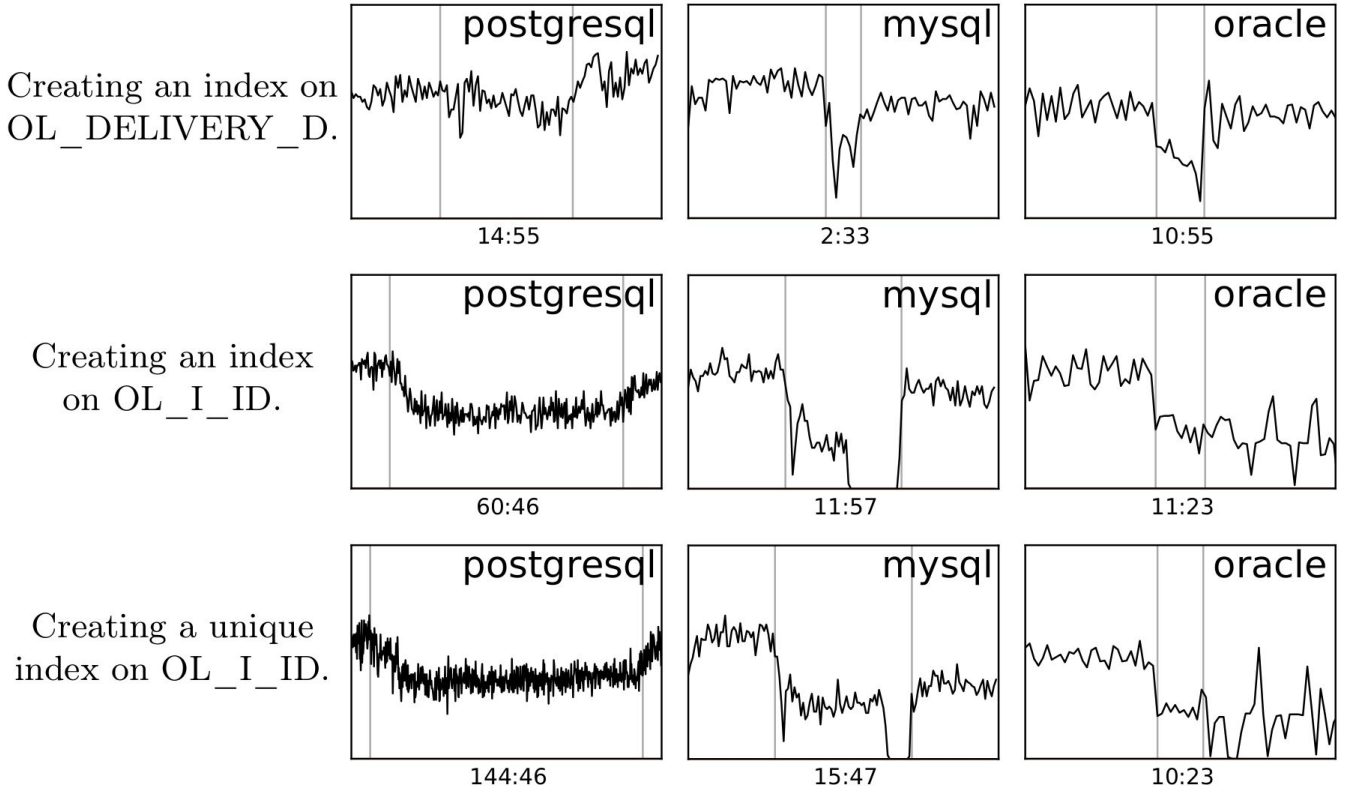
- HammerDB + Custom driver script
- Database: 30 warehouses
- Workload: 64 threads
- Out of the box behaviour: no tuning

Experimental Results

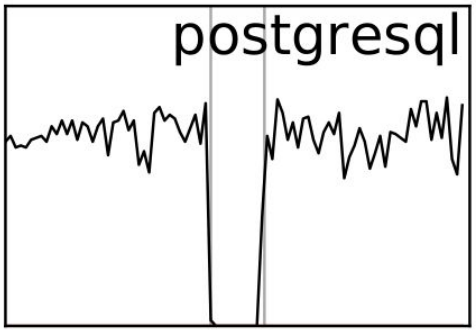
DDL: Column Operations



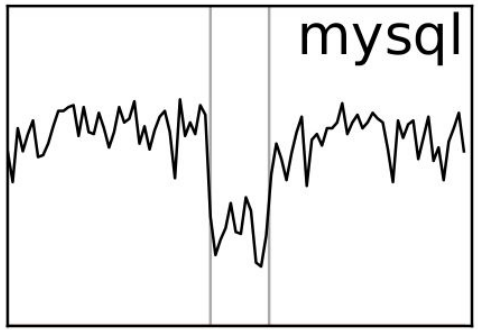
DDL: Index Operations



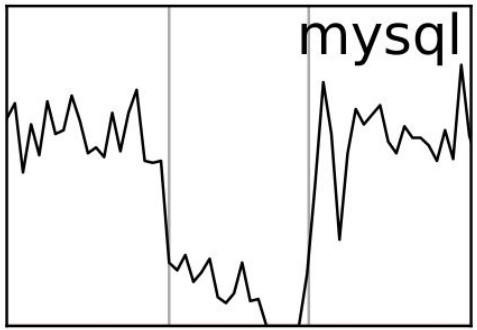
DDL: Bulk Data Transformations



OL_AMOUNT

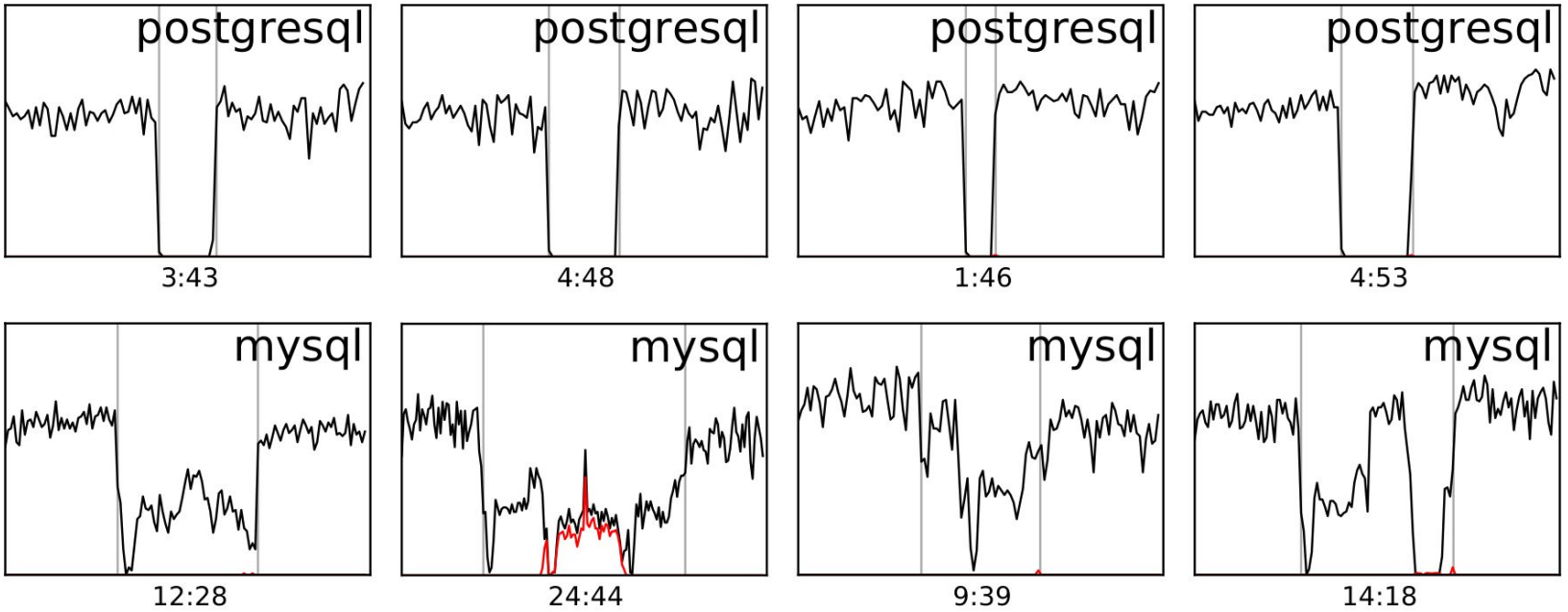


OL_AMOUNT



C_BALANCE

DDL: Complex Transformations



Adding a derived column OL_TAX.

Splitting OL_DIST_INFO.

Giving each order line its own carrier.

Giving each order the carrier of the first order line.

DDL Conclusions

Basic transformations:

- Mixed results
- Online DDL doesn’t always do what you expect.

Complex transformations:

- Every complex operation depends on data transformations, which can not be performed online.

Ronström’s method

id	street	number
1	A str	4
2	B str	78
3	C str	5b
4	D str	1

Ronström’s method

id	street	number
1	A str	4
2	B str	78
3	C str	5b
4	D str	1

id	address

Ronström's method

id	street	number
1	A str	4
2	B str	78
3	C str	5b
4	D str	1

id	address

```
ON UPDATE ...
ON INSERT ...
ON DELETE ...
```

Ronström's method

id	street	number
1	A str	4
2	B str	78
3	C str	5b
4	D str	1

id	address
1	A str 4
2	B str 78

```
ON UPDATE ...
ON INSERT ...
ON DELETE ...
```

Ronström's method

id	street	number
1	A str	4
2	B str	78
3	C str	5b
4	**Anotherstreet**	**87**

id	address
1	A str 4
2	B str 78

```
ON UPDATE ...
ON INSERT ...
ON DELETE ...
```

Ronström's method

id	street	number
1	A str	4
2	B str	78
3	C str	5b
4	**Anotherstreet**	**87**

id	address
1	A str 4
2	B str 78
4	Anotherstreet 87

```
ON UPDATE ...
ON INSERT ...
ON DELETE ...
```

Ronström's method

id	street	number
1	A str	4
2	B str	78
3	C str	5b
4	Anotherstreet	87

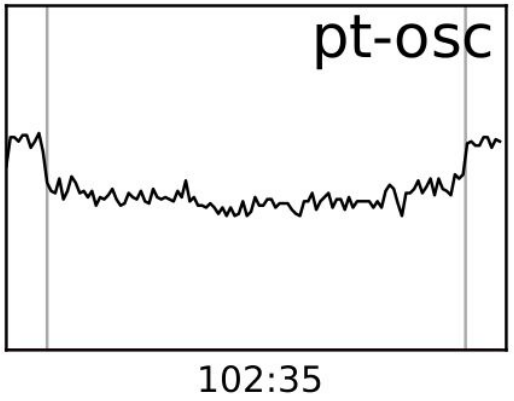
id	address
1	A str 4
2	B str 78
3	C str 5b
4	Anotherstreet 87

```
ON UPDATE ...
ON INSERT ...
ON DELETE ...
```

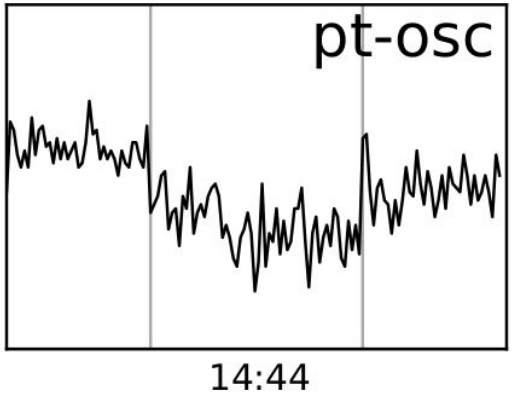
Ronström’s method

id	address
1	A str 4
2	B str 78
3	C str 5b
4	Anotherstreet 87

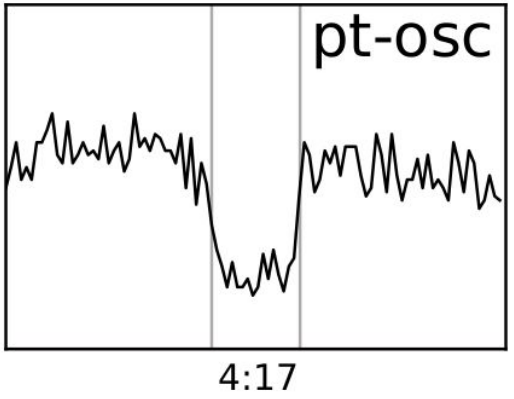
pt-osc: parameters



(a) Chunk size 1000, load 64

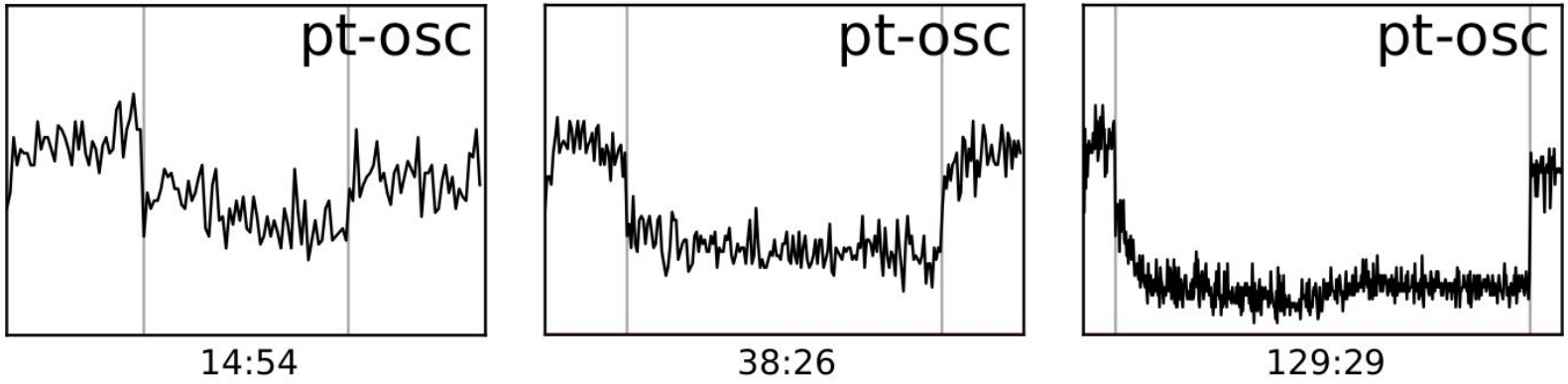


(b) Chunk size 1000, load 4



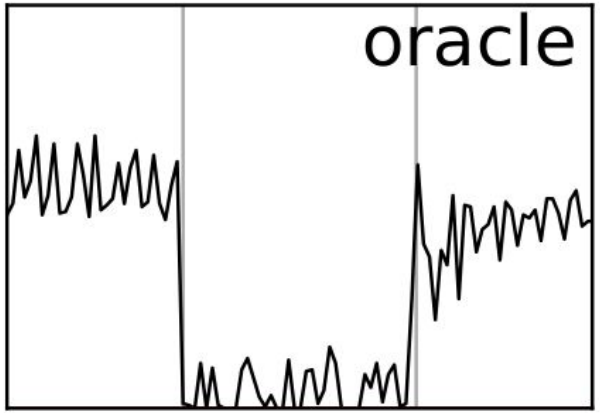
(c) Chunk size 10000, load 4

pt-osc: DDL operations

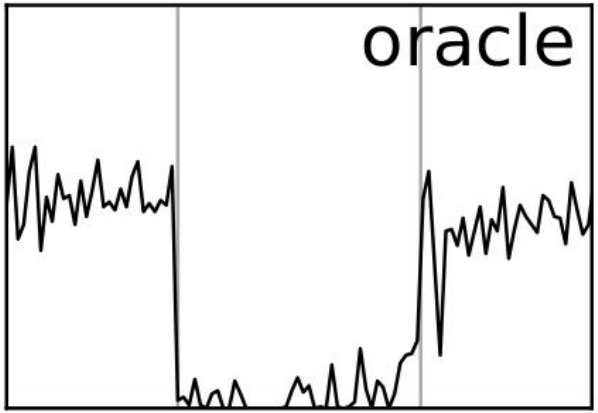


(a) Remove column (b) Create index (c) Create unique

Oracle's DBMS_PARALLEL_COPY



Performing a bulk data update.



Adding a derived column OL_TAX.

Ronstroms method: Conclusions

- Good results for basic transformations
- Support for some complex transformations: derived columns, splitting / merging tables
- Only needs support for triggers in the DBMS.

There are no techniques for transactional composition

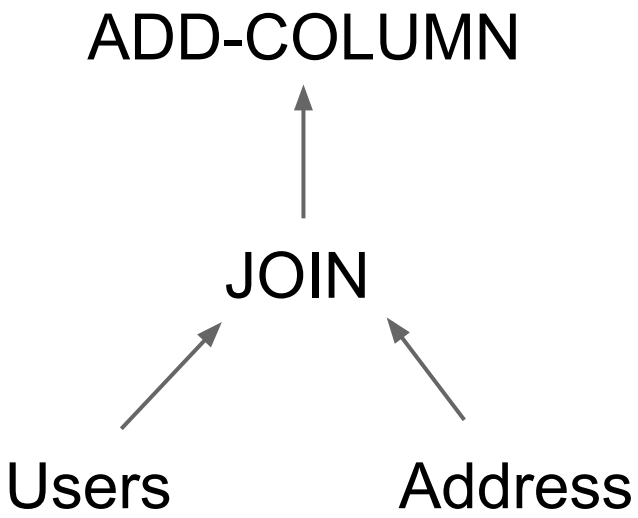
Conclusions

- DDL provided by DBMS has very limited support for online transformations.

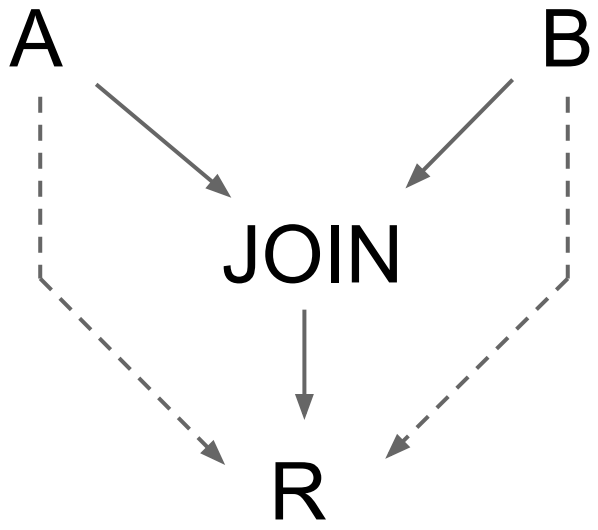
- Ronström's approach is promising, but lacks support for composition.

Solution Sketch for Complex Transformations using Ronström's method

Algebraic Transformations



Propagation using Ronström's method



We need to know how to propagate updates in the source of an operator to updates in the

Propagation must be efficient: We may need to build indices before starting propagation.

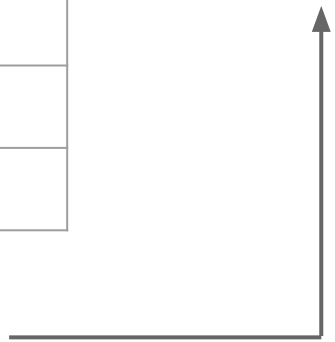
Composition in Ronström's method

id	street	number
1	A str	4
2	B str	78
3	C str	5b
4	D str	1

id	address

id	address

```
ON UPDATE ...
ON INSERT ...
ON DELETE ...
```



Composition in Ronström's method

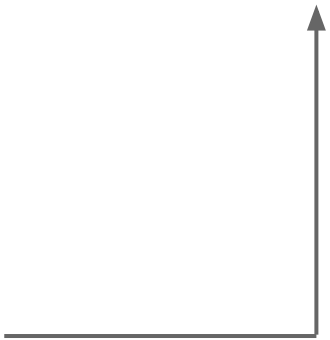
id	street	number
1	A str	4
2	B str	78
3	C str	5b
4	D str	1

id	address

id	address

```
ON UPDATE ...
ON INSERT ...
ON DELETE ...
```

```
ON UPDATE ...
ON INSERT ...
ON DELETE ...
```



Composition in Ronström's method

id	street	number
1	A str	4
2	B str	78
3	C str	5b
4	D str	1

id	address

```
ON UPDATE ...
ON INSERT ...
ON DELETE ...
```

Discussion

DBMSs can provide:

- More efficient alternatives to Ronström's method.
- Instantaneous operations.
- On-the-fly operations.

Outlook

Optimizer for online transformations: generate execution plans for algebraic transformations.

- Select materialization approaches for every operator.
- Optimize for minimal time to commit.

For more information

Read the paper:

- Analysis of the Blocking Behaviour of Schema Transformations in Relational Database Systems

Download the benchmark implementations:

- http://wwwhome.ewi.utwente.nl/~weversl2/?page=ost

Benchmark specification:

- Lesley Wevers, Menno Tammens, Matthijs Hofstra, Marieke Huisman and Maurice van Keulen. A Benchmark for Non-blocking Schema Transformations. DATA 2015.