Query Language Support for Timely Data Deletion

Subhadeep Sarkar
Manos Athanassoulis
exponential growth of data size
IoT
edge computing
autonomous vehicles
smart healthcare

5G
smart city

5G
fast writes

fast reads
Out-of-place paradigm
Out-of-place **systems**

Relational & Array-based
- SciDB
- VERTICA
- [tile]DB
- SQLite
- monetdb
- influxdb
- SAP HANA

NoSQL
- Bigtable
- cassandra
- Apache HBase
- RocksDB
- WT
- Scylla
- DynamoDB
- levelDB
- accumulo
- riak
- QuasarDB
- tarantool
Out-of-place **systems**

**Relational & Array-based**
- SciDB
- Vertica
- tileDB
- SQLite
- monetdb
- influxdb
- SAP HANA
- QuasarDB

**NoSQL**
- Cassandra
- HBase
- Bigtable
- RocksDB
- Scylla
- WT
- levelDB
- DynamoDB
- riak
- tarantool
Out-of-place paradigm

Realizes updates & deletes through new inserts

 Applies updates/deletes lazily to base data

Maintains ingestion-order of entries
Deletes in out-of-place systems

<table>
<thead>
<tr>
<th>ID</th>
<th>Flag</th>
<th>Name</th>
<th>Dept</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0</td>
<td>John</td>
<td>CS</td>
<td>40000</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>Joe</td>
<td>Math</td>
<td>35000</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>Amar</td>
<td>Math</td>
<td>30000</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>Saka</td>
<td>CS</td>
<td>30000</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>Li</td>
<td>Math</td>
<td>60000</td>
</tr>
</tbody>
</table>

DELETE FROM table
WHERE ID = 7;
### Deletes in out-of-place systems

**DELETE FROM** table **WHERE** ID = 7;

**INSERT INTO** table (ID, Flag) **VALUES** (7, 1);

<table>
<thead>
<tr>
<th>ID</th>
<th>Flag</th>
<th>Name</th>
<th>Dept</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>John</td>
<td>CS</td>
<td>40000</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>Joe</td>
<td>Math</td>
<td>35000</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>Amar</td>
<td>Math</td>
<td>30000</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>Saka</td>
<td>CS</td>
<td>30000</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>Li</td>
<td>Math</td>
<td>60000</td>
</tr>
</tbody>
</table>
### Deletes in out-of-place systems

<table>
<thead>
<tr>
<th>ID</th>
<th>Flag</th>
<th>Name</th>
<th>Dept</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>John</td>
<td>CS</td>
<td>40000</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>Joe</td>
<td>Math</td>
<td>35000</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>Amar</td>
<td>Math</td>
<td>30000</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>Saka</td>
<td>CS</td>
<td>30000</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>Li</td>
<td>Math</td>
<td>60000</td>
</tr>
</tbody>
</table>

**DELETE** FROM table WHERE ID = 7;

**INSERT** INTO table (ID, Flag) VALUES (7, 1);
Deletes in out-of-place systems

```
DELETE FROM table
WHERE ID = 7;

INSERT INTO table (ID, Flag)
VALUES (7, 1);
```
Policy layer

Right to be forgotten

Right to delete

Deletion right

System layer

Data layout re-organization

Data deletion algorithms

Sarkar et al., SIGMOD '20

Cohn-Gorrordon et al., USENIX Security Symp. '20
deletion requirements
Policy layer

deletion requirements

System layer

requirement translation
Deletion requirements

- retention-based: delete all data older than $D$ days
- on-demand: delete data object $X$ within $D$ days
Retention-based deletes

CREATE TABLE R (ID int, Name varchar(255), ...) WITH FIXED RET_DUR (t1 180, t2 365);

INSERT INTO R (32, Aaron, ...) WHERE RET_DUR = t1;
CREATE TABLE R (ID int, Name varchar(255), ...) WITH FIXED RET_DUR (t1 180, t2 365);

INSERT INTO R (32, Aaron, ...) WHERE RET_DUR = t1;
Retention-based deletes

CREATE TABLE R (ID int, Name varchar(255), ...) WITH ARBITRARY RET_DUR;

INSERT INTO R (32, Aaron, ...) WITH RET_DUR 90;
On-demand deletes

CREATE TABLE R (ID int, Name varchar(255), ...)  
WITH FIXED DPT (d1 30, d2 45, d3 60);

DELETE FROM R  
WHERE ID = 32  
WITH DPT d2;
CREATE TABLE R (column1 type1, column2 type2, ...) WITH RET_DUR {ARBITRARY|FIXED (t1 <ret1>, t1 <ret1>, ...)} WITH DPT {ARBITRARY|FIXED (d1 <dpt1>, d1 <dpt1>, ...)};

INSERT INTO R (val1, val2, ...) WITH RET_DUR {<t>|t<i>};

DELETE FROM R WHERE (...) WITH DPT {<d>|d<i>};}
Policy layer:
- Right to be forgotten
- Right to delete
- Deletion right

System layer:
- Data layout re-organization
- Data deletion algorithms
<table>
<thead>
<tr>
<th>Policy layer</th>
<th>Requirements layer</th>
<th>Application layer</th>
<th>System layer</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Right to be forgotten" /></td>
<td><img src="image" alt="Retention-based Deletes" /></td>
<td>![CREATE TABLE R (...) WITH RET_DUR {ARBITRARY</td>
<td>FIXED(...)} WITH DPT {ARBITRARY</td>
</tr>
<tr>
<td><img src="image" alt="Right to delete" /></td>
<td><img src="image" alt="On-demand Deletes" /></td>
<td>![INSERT INTO R (...) WITH RET_DUR {&lt;t&gt;</td>
<td>t&lt;i&gt;};](INSERT_INTO_R())</td>
</tr>
</tbody>
</table>
APIs to express system deletion requirements

Persistent deletion from file systems

Secure & persistent deletion at device level
Toward **deletion-compliant** data systems

disc.bu.edu/persistent-deletes

Thank You!