Reporting Technologies
Static and Dynamic Reporting

Michael Nissen
michaeln@diku.dk

Department of Computer Science, University of Copenhagen

Nov 18. 2008
1 Reporting

2 Technologies of Today
   - Materialized Views
   - OLAP
   - SIFT
   - Summary

3 Technologies of Tomorrow?
   - FunSETL
   - Map-Reduce

4 Summary
What is Reporting?

Definition (Report Function)

A report function is a function on transactional data.

Reporting is the discipline of

- *Applying* report functions, that is, executing their specification on actual data.
- *Expressing* report functions, that is, describe them in a specification- or programming language.

Note: Presentation of results is NOT included in the definition.
Static and Dynamic Report Functions

Concept (Static and Dynamic Report Functions)

- **A Static Report Function is a report function, which we know in advance that we want to compute at some point.**
- **A Dynamic Report Function is a report function, which we do NOT know in advance that we want to compute at some point.**
Reporting Today

- Report functions are usually expressed using fx. SQL, OLAP, SIFT (Microsoft NAV) or in a general purpose programming language (for instance, X++ or C/AL).
- ERP systems contain a lot of data.
- ERP systems primarily accumulate data.
- Many report functions are conceptually simple.
- Many report functions are computed from scratch.
What are the problems and what do we want?

- Computing report functions is time consuming.
- Expressing report functions can be hard in the existing specification- and programming languages.
- Real-time or near-real-time (dash-boarding) computations of report functions are preferable.
- The responsibility of efficient computation of report functions should be moved away from the developer.
What are the problems and what do we want?

- Computing report functions is time consuming.
- Expressing report functions can be hard in the existing specification- and programming languages.
- Real-time or near-real-time (dash-boarding) computations of report functions are preferable.
- The responsibility of efficient computation of report functions should be moved away from the developer.
Realized Technologies

- Materialized Views
- OLAP
- SIFT (Microsoft NAV)
- Google’s Map-Reduce
- FunSETL
Realized Technologies

- Materialized Views
- OLAP
- SIFT (Microsoft NAV)
- Google’s Map-Reduce.
- FunSETL
Materialized Views

- **What?**: Storage of virtual relations.
- **Why?**: Faster access to virtual relations.
### Bicycle Business - Example

<table>
<thead>
<tr>
<th>Branch</th>
<th>Color</th>
<th>Time_Id</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Valby</td>
<td>Red</td>
<td>T1</td>
<td>1599</td>
</tr>
<tr>
<td>Frederiksberg</td>
<td>Red</td>
<td>T2</td>
<td>1799</td>
</tr>
<tr>
<td>Valby</td>
<td>Red</td>
<td>T3</td>
<td>1399</td>
</tr>
<tr>
<td>Frederiksberg</td>
<td>Blue</td>
<td>T4</td>
<td>2199</td>
</tr>
<tr>
<td>Valby</td>
<td>Red</td>
<td>T5</td>
<td>1299</td>
</tr>
<tr>
<td>Frederiksberg</td>
<td>Blue</td>
<td>T6</td>
<td>1299</td>
</tr>
<tr>
<td>Frederiksberg</td>
<td>Blue</td>
<td>T7</td>
<td>2399</td>
</tr>
</tbody>
</table>
Materialized Views - Example

Example

Declare a view totalsales that holds the sum of the sales for each branch.

```sql
create view totalsales(branch, amount) as
select Branch, sum(Price)
from sale
group by Branch
```

<table>
<thead>
<tr>
<th>branch</th>
<th>amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frederiksberg</td>
<td>7696</td>
</tr>
<tr>
<td>Valby</td>
<td>4297</td>
</tr>
</tbody>
</table>
Materialized Views - Issues

- **View Maintenance.**
  - How should a materialized view be updated when the data it depends on is changed?
  - The example view can be updated incrementally.

- **Purging unused views.**

- **Can** in some cases **be used to do real-time report function computation:**
  - A materialized view can be declared to maintain results needed by a static report function.
  - We can get lucky and use a materialized view in the computation of a dynamic report function.
OLAP - OnLine Analytical Processing

- **What?**: Special kind of materialized views. (Union of GROUP BY SQL statements).
- **Why?**: Speedup computation time of queries that benefit from these kind of views.
OLAP - Issues

- OLAP cube relations can be as big (or even bigger) than the source tables they stem from.
- Updating OLAP cubes has the same problems as Materialized Views.
- Can in some cases be used to do real-time report function computation:
  - An OLAP cube can be declared to maintain results needed by a static report function.
  - We can get lucky and use an OLAP cube in the computation of a dynamic report function.
SIFT

- *What?*: Virtual fields on existing tables containing aggregate information.
- *Why?*: To speedup the computation of report functions.
SIFT - Issues

- Updating FlowFields.
- Purging unused FlowFields.
- Some static report functions can be computed in real-time using FlowFields.
Summary

The technologies presented so far:

- Some static report functions can benefit from these technologies.
- Can maintain unnecessary information, which however gives some possibility of dynamic report function computation.
- Unclear when real-time computation can be performed (the developers responsibility to identify this).
Technologies of Tomorrow?

- Why only use Relational Database Technologies?
- Relational databases do not have a distinction of static and dynamic queries.
- Generally low support for real-time computation.
FunSETL

- Declarative specification of report functions.
- Automatic transformation to incremental specification (often real-time).
- Asymptotic improvement in many cases.
- Only maintaining the necessary information.
- Suited for static report functions.
Map-Reduce

- **What?** C++ library.
- **Why?** Automatic parallelization of computations.
- **How?** Execute on many low price machines.
Map-Reduce - Example

Example

Compute the total number of bicycles sold of each color.

map and reduce functions declared as (written in pseudo code).

1: map (String branch, String color):
2:   EmitIntermediate(color, 1);
3: 
4: reduce (String color, Iterator values):
5:   int result = 0;
6:   foreach v in values:
7:     result += v;
8:   Emit(result);
Map-Reduce Comments

- Current Map-Reduce not suited for real-time computation (maybe it can be adapted).
- Suited for dynamic report functions.
- Removes responsibility of efficient computation away from the developer.
Summary

Relational Databases, Materialized Views, OLAP and SIFT does not provide good support for

- Real-time or near-real-time computation of report functions.

Idea

*Split the specification of report functions in two classes:*

- **Dynamic:** Specification that guarantees parallelization of the computation.
- **Static:** Specification that guarantees that the results are maintained (incrementally) in real-time or near-real-time.
Example

OLAP Cube with Color and Quarter and aggregate Sum.

```sql
select sale.Color, time.Quarter, sum(sale.Price)
from sale, time
where sale.Time_id = time.Time_id
group by cube(sale.Color, time.Quarter)
```
## OLAP - Example - Result

<table>
<thead>
<tr>
<th>Color</th>
<th>Quarter</th>
<th>sum(Price)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>1</td>
<td>4797</td>
</tr>
<tr>
<td>Blue</td>
<td>1</td>
<td>2199</td>
</tr>
<tr>
<td>Red</td>
<td>2</td>
<td>1299</td>
</tr>
<tr>
<td>Blue</td>
<td>2</td>
<td>3698</td>
</tr>
<tr>
<td>Blue</td>
<td>-</td>
<td>5897</td>
</tr>
<tr>
<td>Red</td>
<td>-</td>
<td>6096</td>
</tr>
<tr>
<td>-</td>
<td>1</td>
<td>6996</td>
</tr>
<tr>
<td>-</td>
<td>2</td>
<td>4997</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>11993</td>
</tr>
</tbody>
</table>
FunSETL - Financial Statement

- Non-incremental
- Incremental

Graph showing the relationship between Events and Seconds for both Non-incremental and Incremental processes.