Building OLAP Data Analytics by Storing Path-Enumeration Keys into Sorted Sets of Key-value Store Databases

Luis Loyola, Fernando Wong, Daniel Pereira

SkillupJapan Corporation
Tokyo, Japan
Layout

- Introduction
- Related work
- Proposed scheme
- Tests and results
- Discussion and future work
OLAP (Online Analytical Processing)
Multidimensional database where measures or statistics of interest are pre-calculated in *fact tables*
Steps: Roll-up, Drill-down, Slicing & Dicing

OLAP CUBE

Problem definition

**OLAP is great but...**
- The initial cost and maintenance complexity of OLAP solutions makes it difficult for small/medium startups to acquire and benefit from it.
- Installing, setting up, configuring and maintaining an OLAP database usually demands training software engineers.
- We needed a simple and cost-effective analytics solution.

**Our original motivation**
Can we make OLAP-like data analytics simpler using an open-source database?
Introduction: key-value store

A rapid growth in recent years because of their simplicity to set up and high speed to answer queries.

- No-SQL databases (no SQL queries)
- Schema less (no tables)
- Different types: eventually consistent, hierarchical, key-value cache, etc.
- BigTable, MemCache, Amazon DynamoDB, LevelDB, Tokyo Cabinet, REDIS, etc.
Introduction: REDIS

REDIS
From the main page: http://redis.io

"Redis is an open-source, advanced key-value store. It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets and sorted sets"

Sets: Non repeating collections of strings
example_set_1013 => {"A", "B", "C"}

Sorted sets: identical to sets but every member is associated with a score. While members are unique, scores can be repeated.
example_sorted_set_1013 => {"A":10, "B":8, "C":7}

Operations: Union and Intersection
Problem definition

Relational Database (MySQL)

Example query
What was the number of clicks on banner X from Campaign C and Sponsor S when displayed on Ad Space S in Website W from Publisher P between 2012/06/17 and 2012/09/19?

LOTS OF EXPENSIVE TABLE JOINS TO ANSWER THIS QUERY

OLAP Database

Same Example Query

Very fast response since clicks have been pre-calculated for all these dimensions
Problem definition

Example query
What was the number of clicks on banner X from Campaign C and Sponsor S when displayed on Ad Space S in Website W from Publisher P between 2012/06/17 and 2012/09/19?

Relational Database (MySQL)

Lots of expensive table joins to answer this query

OLAP Database

Very fast response since clicks have been pre-calculated for all these dimensions
Problem definition

**Example query**
What was the number of clicks on banner X from Campaign C and Advertiser A when displayed on Ad Space S in Website W from Publisher P between 2012/06/17 and 2012/09/19?

Relational Database (MySQL)

Lots of expensive table joins to answer this query

Key-value store

{Sets, Sorted Sets}

with

{Union and Intersection}

Same Example Query

Very fast response since clicks have been pre-calculated for all these dimensions
Proposed Scheme: storing

Path enumeration model on tree => generate unique key per path

Build a set D with the deepest paths of the tree and their combinations

D1 = N1:N2:N3
D2 = N1:N4:N5
D3 = N1:N4:N6
D4 = D1:D2 = N1:N2:N3:N4:N5
D5 = D1:D3 = N1:N2:N3:N4:N6

D2:D3 does not help because they never happen together (disparate paths with common stem)

D = {D1, D2, D3, D4, D5}

Proposed Scheme: storing

Important:
All sets are created on the fly as events occur

Suppose there is a click on banner $R$ of campaign $C$ that belongs to advertiser $A$ when it is being displayed on ad space $S$ in website $W$ of publisher $P$
Proposed Scheme: storing and querying

**Approach 1:** Perform union operations among sorted sets with statistics stored for full branches and their combinations

**Approach 2:** Perform union operations only in time dimension among sorted sets with statistics for partial branches and their combinations

**Approach 3:** Name the sorted sets not in a generic fashion as in the previous two approaches but using the individual IDs of dimension records

**Some reasoning**
It is easy to imagine that in terms of memory usage the 1st approach is the best followed by the second and finally the third. In terms of query response speed the order is exactly the opposite.
Proposed Scheme: querying

Approach 1

Example query: what was the total number of clicks on all Ad Spaces in website W1, which belongs to publisher P1 and manager M1 between dates D1 and D2?

\[ Q' = M:P:W:S(m, p, w, \ast) \cap Q \]

\[ Q = \left\{ \begin{array}{l}
  m_1:p_1:w_1:s_1\text{clicks} = 2, m_1:p_1:w_1:s_2\text{clicks} = 7, \\
  m_1:p_2:w_2:s_3\text{clicks} = 4, m_1:p_2:w_2:s_5\text{clicks} = 5, \\
  m_2:p_3:w_3:s_7\text{clicks} = 3, m_2:p_3:w_4:s_8\text{clicks} = 6 \\
\end{array} \right\} \]

\[ M:P:W:S(m_1, p_2, w_2, \ast) = \left\{ \begin{array}{l}
  m_1:p_2:w_2:s_3\text{clicks}, \\
  m_1:p_2:w_2:s_4\text{clicks}, \\
  m_1:p_2:w_2:s_5\text{clicks}, \\
  m_1:p_2:w_2:s_6\text{clicks} \end{array} \right\} \]

\[ Q = \bigcup_{d=\text{date}_1}^{\text{date}_2} M:P:W:S(d) \]
Proposed Scheme: querying

**Approach 2**

Example query: what was the total number of clicks on all Ad Spaces in website W1, which belongs to publisher P1 and manager M1 between dates D1 and D2?

\[ Q' = \bigcup_{d=\text{date}_1}^{\text{date}_2} \text{M:P:W}(d) \]
Proposed Scheme: querying

**Approach 3**

Example query: what was the total number of clicks on all Ad Spaces in website W1, which belongs to publisher P1 and manager M1 between dates D1 and D2?

\[
Q' = \bigcup_{d=d_{date_1}}^{d_{date_2}} M:P:W(m, p, w, d)
\]
Proposed Scheme: querying

Non-aggregate Events

Example
What is the number of unique users that clicked on campaign resource r from campaign c ran by advertiser a between date1 and date2?

Solution: use simply sets, not sorted sets

\[
|U| = \bigcup_{d=\text{date}_1}^{\text{date}_2} \text{Users:}\text{A:}\text{C:}\text{R}(a, c, r, d)
\]
Tests and Results

• Seven representative queries for performance evaluation

• The data used for all queries correspond to real production data taken from September 1st, 2011 through November 30th, 2011, i.e. exactly three months

• Our approach is compared to a ROLAP Pentaho Mondrian database running on top of MySQL

• All following queries use $date1 = Sep. 1st, 2011$ & $date2 = Nov. 30th, 2011$
Tests and Results

Performance comparison for Query #1
What is the total number of clicks and views between dates \texttt{date1} and \texttt{date2} for campaign resource \texttt{R} of campaign \texttt{C} owned by Advertiser \texttt{A} and Manager \texttt{M}?

Performance comparison for Query #2
What is the total number of clicks and views between dates \texttt{date1} and \texttt{date2} on the advertising space \texttt{S} owned by Publisher \texttt{P} and Manager \texttt{M} when shown on applications?
Tests and Results

Performance comparison for Query #3

What is the number of clicks and views between dates $\text{date}_1$ and $\text{date}_2$ for campaign resource $R$ from campaign $C$ owned by Advertiser $A$ when shown in web sites on the ad space $S$ owned by Publisher $P$ and manager $M$?

Performance comparison for Query #4

What is the number of clicks and views between dates $\text{date}_1$ and $\text{date}_2$ for campaign resource $R$ from campaign $C$ owned by advertiser $A$ when shown on all ad spaces of all websites owned by publisher $P$ and manager $M$?
Tests and Results

Performance comparison for Query #5

What is the number of clicks and views between dates \textit{date1} and \textit{date2} for the whole campaign \textit{C} owned by advertiser \textit{A} when shown in applications on the ad space \textit{S} owned by publisher \textit{P} and manager \textit{M}?

Performance comparison for Query #6

What is the number of unique users that clicked on any campaign resource from campaign \textit{C} owned by advertiser \textit{A} when shown on applications in ad space \textit{S} owned by publisher \textit{P} and manager \textit{M} between \textit{date1} and \textit{date2}?
Performance comparison for Query #7

What is the number of unique users that clicked on all campaign resources owned by advertiser A when shown on all ad spaces owned by publisher P and manager M between date1 and date2?
We have proposed a novel scheme to perform OLAP-like statistics using sorted sets in key-value store.

Our system maps any relational database to key-value store in order to obtain accumulated measures of events of interest.

The system is efficient in terms of memory usage and very fast for responding complex queries involving aggregate events, outperforming in most cases a Pentaho Mondrian ROLAP database running on top of a properly indexed MySQL.

Out of the three proposed approaches the best one in practice for our case was Approach 2 since it gave us the best performance tradeoff between query response’s speed and memory usage.

The system performs very well for aggregate events but uses too much memory for non-aggregate events such as unique users counting.
Future work

- Analyze other types of data structures such as Hashes or Lists to tackle other specific problems
- Find more efficient ways to reduce memory consumption in case of non-aggregate events
- Apply our concept in other services offered by our company (e.g. Geotargeting)
THANKS FOR YOUR ATTENTION