BlinkDB - Queries with Bounded Errors and Bounded Response Times on Very Large Data

Paweł Kamiński

18 listopada 2015
Problem

Situation

Huge database consists of one table.
### Problem

#### Situation

Huge database consists of one table.

#### Blockers

- Query execution time.
- Space taken by database.
Example

```
SELECT AVG(SessionTime)
FROM Sessions
WHERE city = 'London'

SELECT AVG(SessionTime)
FROM Sessions
WHERE city = 'Pontefract'
```
Solution

Average case

In most cases approximated answer is enough. So let’s work on samples.
Query Column Set

The set of columns used by queries in WHERE, GROUP BY, HAVING.
Approaches

- Uniform Sampling
- Online Aggregation
- Sketches
unique - all

Fraction of Queries (CDF)

Unique Query Templates (%)

Conviva Queries (2 Years)

Facebook Queries (1 week)
Big joins

Fraction of Join Queries (CDF)

Size of Dimension Table(s) (GB)

Facebook Queries (1 Week)
BlinkDB architecture

SELECT COUNT(*)
FROM TABLE
WHERE (city="NY")
LIMIT 1s;

Sample Selection

Shark

Result: 1,101,822 ± 2,105
(95% confidence)

HiveQL/SQL Query

TABLE

Original Data

Sample Creation & Maintenance

Distributed Filesystem

Distributed Cache
Example

```
SELECT COUNT(*)
FROM Sessions
WHERE city='London'
GROUP BY OS
ERROR WITHIN 10% AT CONFIDENCE 95%
/WITHIN 5 SECONDS
```
Samples

- Uniform Samples
- Single-Column Stratified Samples
- Multi-Column Stratified Samples
Samples

- Uniform Samples GROUP
- Single-Column Stratified Samples efficiency
- Multi-Column Stratified Samples
Stratified Samples - cutting
Machine memory spacing
Choose QCS

- Use less memory than we have in whole system
- Coverage as many query QCS as possible
- Try to not sample big groups
Selecting Sample

- satisfiability vs all in memory
- disk vs memory
Selecting Right Sample

- Error Profile - variance $\sim \frac{1}{\text{numberofrows}}$

- Latency Profile - I/O scales lineary + initial cost
Example

Task: get average today's connection time in Pontefract

QCS:

- date/country
- date/designated media
- date/ended flag
Query Response Time
Statistical Error

Statistical Error (%)
Query Latency (seconds)

Sample Size (in MB)
Implementation stack

- BlinkDB Query Interface
- Hive Query Engine
  - Uncertainty Propagation
  - Sample Selection
    - Hadoop MapReduce
    - Shark (Hive on Spark)
    - Spark
    - BlinkDB Metastore
  - Sample Creation and Maintenance
- Hadoop Distributed File System (HDFS)
EC2 100 nodes 8 CPU 2.66 GHZ 70 GB RAM and 800 GB disc, what gives 75 TB discs storage and 6TB distributed RAM cache.

Tests - Conviva 17TB, TPC-H 1TB

Samples +50% memory
BlinkDB vs No sampling

- Hive
- Hive on Spark (without caching)
- Hive on Spark (with caching)
- BlinkDB (1% relative error)

Query Service Time (seconds)

Input Data Size (TB)

- 2.5TB
- 7.5TB
Error comparison (Conviva)

![Graph showing error comparison for different QCS configurations.](image-url)
Benchmark - scaling up

Actual Response Time (seconds)

Requested Response Time (seconds)
Benchmark - scaling up

![Graph showing actual error (%) vs requested error bound (%). The graph indicates an increase in actual error as the requested error bound increases.](image-url)