

Global analytics in the face of bandwidth and regulatory constraints

Geode

Ashish Vulimiri, Thomas Jungblut, Carlo Curino, Jitu Padhye, Brighten Godfrey, George Varghese

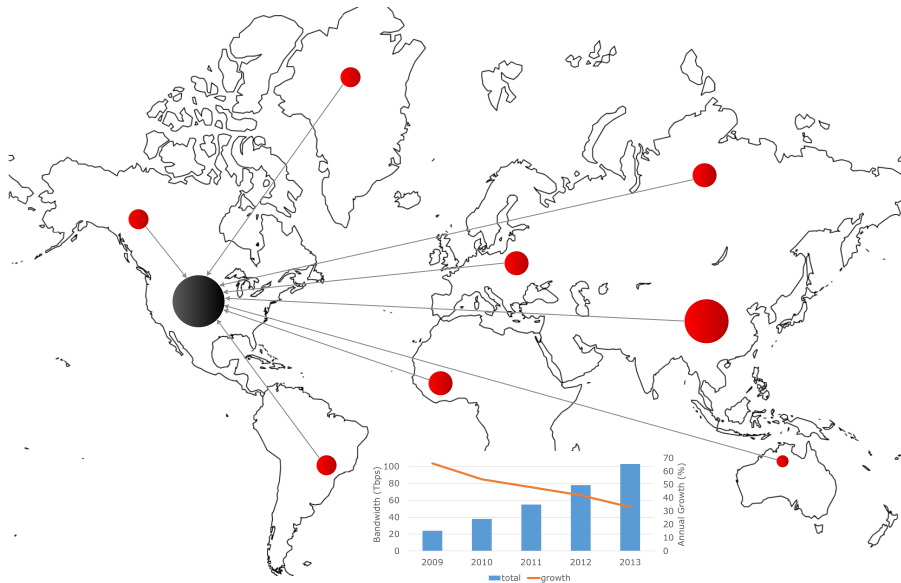
Presented by Claudio Scheer

Table of Contents

1 Overview

2 Geode

Wide-Area Big Data



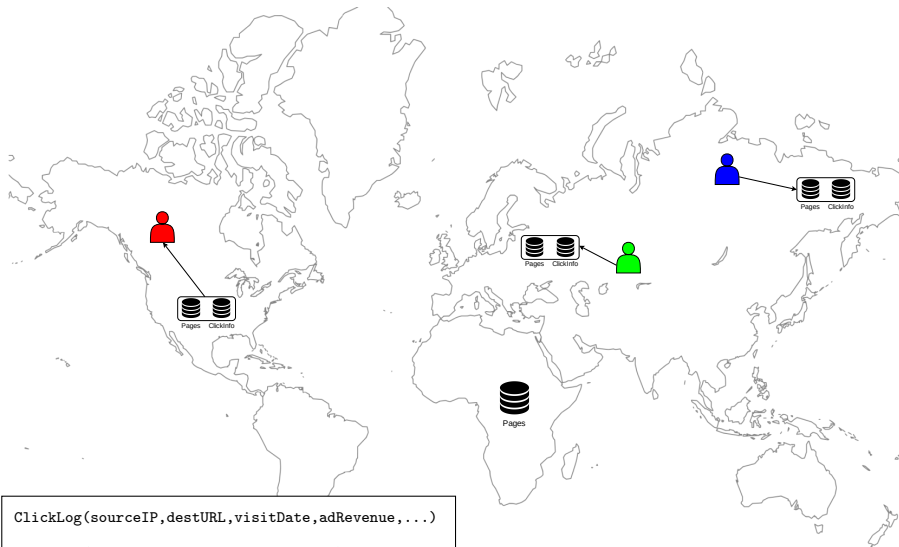
- SQL analytics over geo-distributed data;
- Support joins over geo-distributed data centers;
- “What’s the best join order?” to reduce bandwidth;
- No attempt to minimize execution latency;

Table of Contents

1 Overview

2 Geode

Example



Example - centralized

```
SELECT sourceIP, sum(adRevenue), avg(pageRank)
FROM ClickLog cl
JOIN PageInfo pi ON cl.destURL = pi.pageURL
WHERE pi.pageCategory = 'Entertainment'
GROUP BY sourceIP
HAVING sum(adRevenue) >= 100;
```

1B users

6 pages visited per user/day

200 bytes per ClickLog row

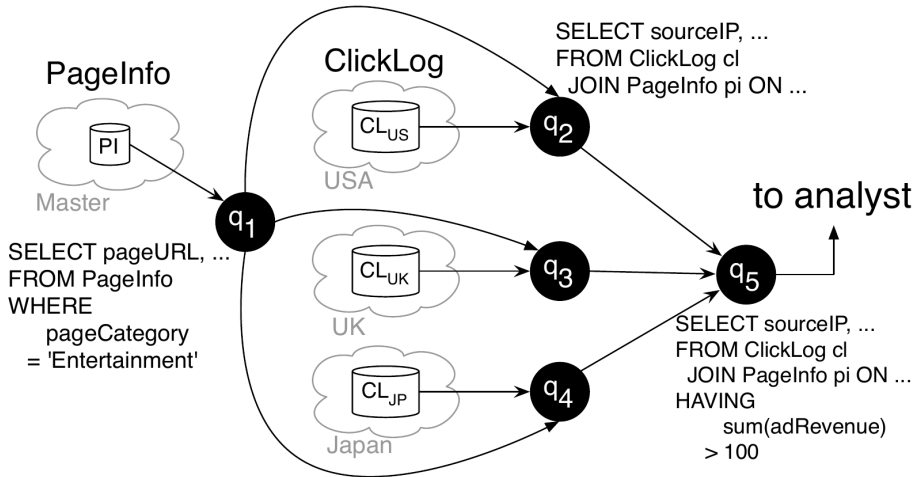
= 1.2 TB

Example - Geode

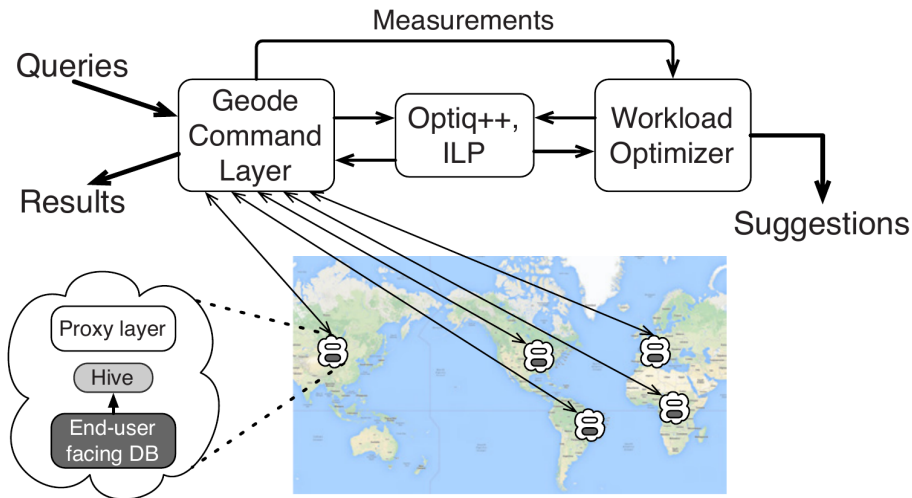
```
SELECT sourceIP, sum(adRevenue), avg(pageRank)
FROM ClickLog cl
JOIN PageInfo pi ON cl.destURL = pi.pageURL
WHERE pi.pageCategory = 'Entertainment'
GROUP BY sourceIP
HAVING sum(adRevenue) >= 100;
```

1B users
100M pages visited per day
each q1 tuple is 20 bytes
each q2, q3, q4 tuple is 12 bytes
= 14 GB

Example - Geode



Architecture

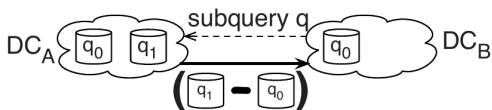
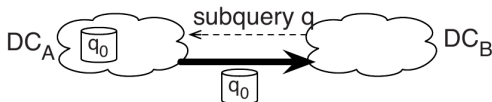


time

0 — DC_B asks DC_A for result of running subquery q ; DC_A computes and sends q_0

After transfer, both keep cached copy

1 — DC_B again asks DC_A for result of running subquery q ; DC_A computes new results q_1 , but only sends diff



- Centralized approach performs better when update rates are low;
- At high update rates, caching is ineffective;
- Distributed execution outperformed the centralized approach in four of six workloads (250x - 360x);

Open challenges

- Latency;
- They limit where data can be copied, but not data movement through queries;
- Support for other data models, such as MapReduce;