Scuba: Diving into Data at Facebook

Presenter: Lavanya Subramanian

Need for Data Analysis

- **Performance monitoring**
  - Detect unexpected performance drops/rises
- **Pattern mining**
  - Understand user response to new features
- **Ad revenue monitoring**
  - Identify regional drops/rises in ad clicks and revenue

Data Analysis at Facebook

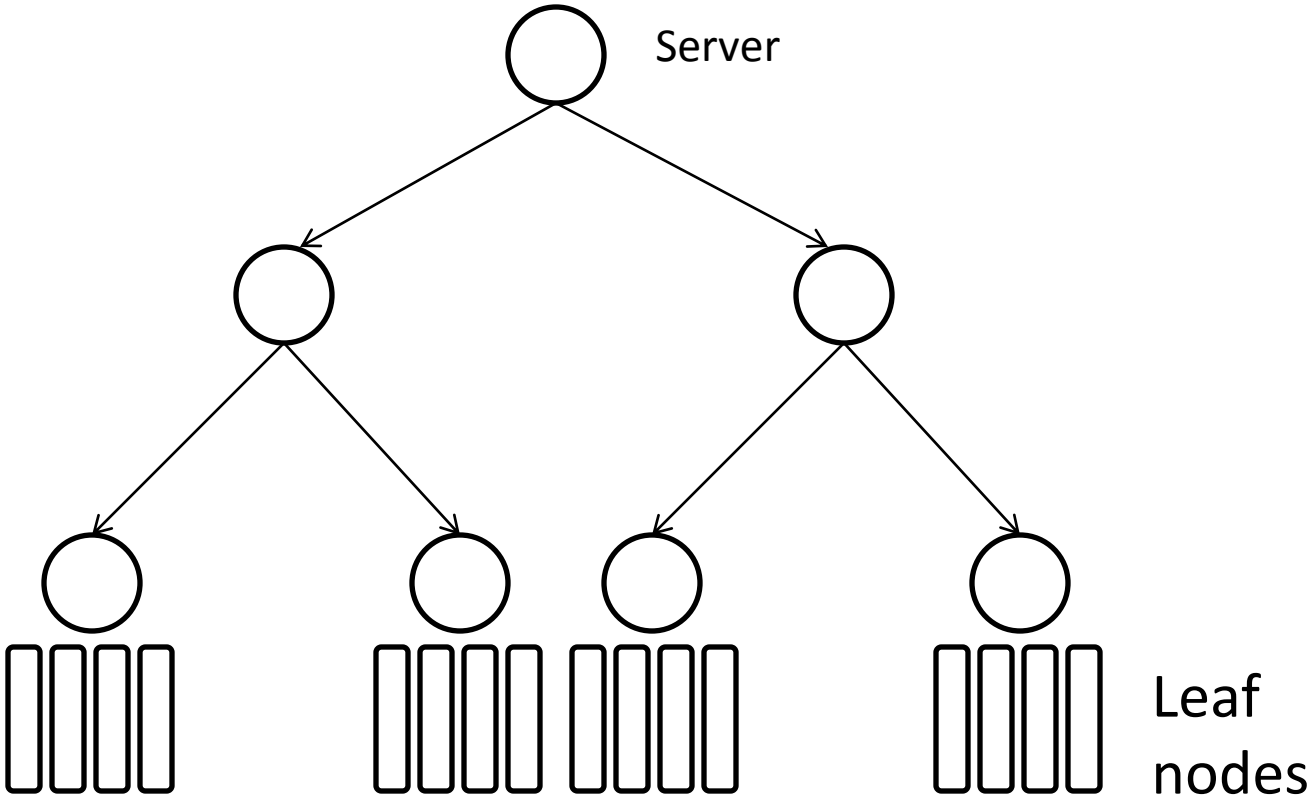
- Large data volumes
- **Real time analysis** of this data
- Key Requirements
  - Low latency
  - Flexibility
  - Scalability

Proposed Solution: Scuba

- **Structure**
  - In-memory database
  - Across hundreds of servers

- **How does it work?**
  - Holds and processes sampled real-time data
  - Query interface to access data
  - Visualization interface to analyze data

Architecture



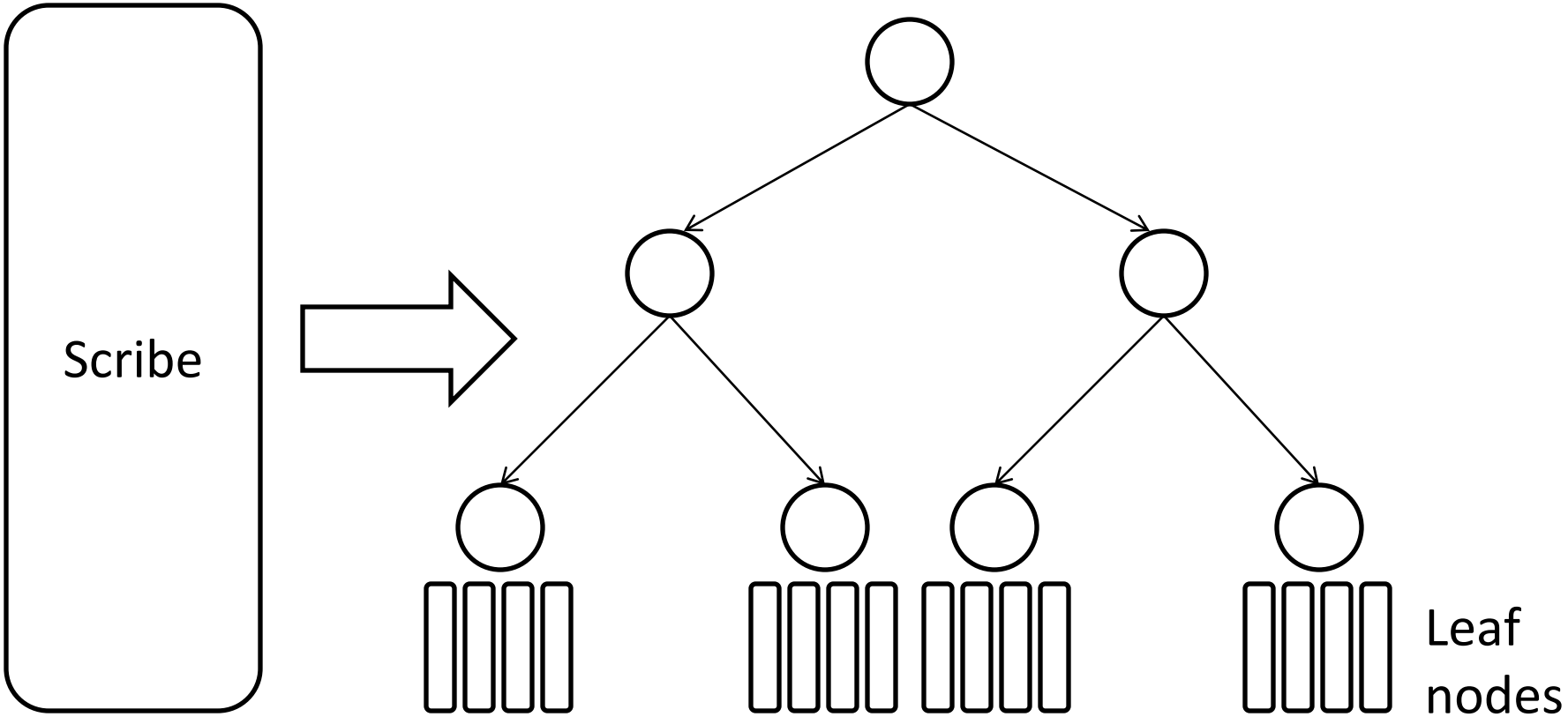
Data Layout

- Data stored in tables
- Data types supported
  - Integers, strings, sets of strings, vectors of strings
- Different compression for different data types

Table Characteristics

- Table is created upon data arrival at a leaf node
- Table can have empty columns; treated as null

Data Ingestion into Scuba



Data Ingestion into Scuba

- Events are **sampled** to reduce the data volume
- Use Scribe, a distributed messaging system to
  - Collect, aggregate and deliver data to Scuba
- For each batch of incoming data
  - Pick **two leaf nodes at random**
  - Send the batch to the node with more free memory
- Data compressed and sent to disk
- **Data then read back and stored in memory**

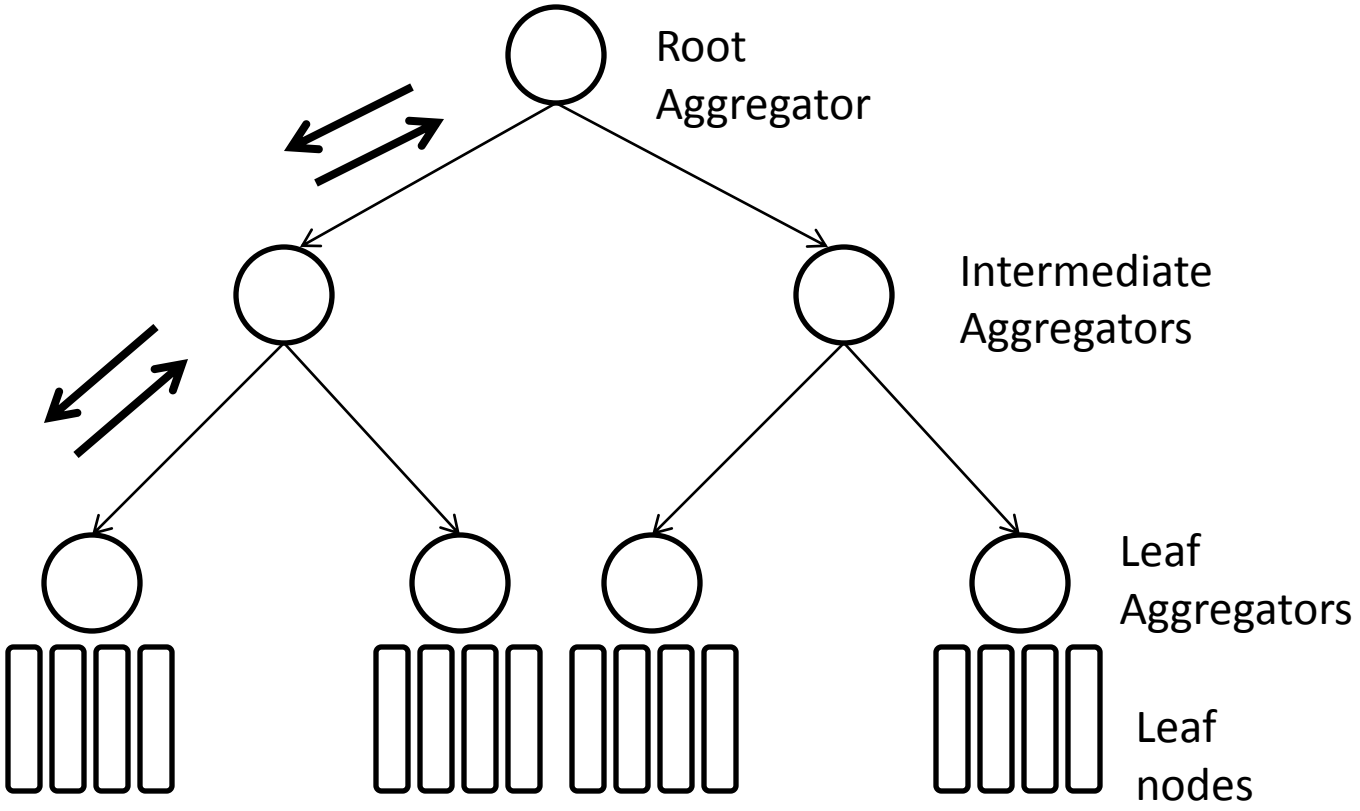
Dealing with Old Data

- Memory capacity is a concern
- Need to add new servers every 2-3 weeks
- Delete data based on
  - Age: Sample and preserve a fraction of old data
  - Space: When exceeding space limits, delete old data

Querying Scuba

- **Three kinds of interfaces**
  - Web-based
  - SQL
  - API to support querying from application code
- **Queries supported**
  - Different forms of aggregation
  - Percentiles, histograms
- Joins not supported by Scuba

Query Execution



Query Execution

- Leaf node may or may not contain a table's data
  - Depends on the table size and age
- Data scanning is usually by time range
  - Time is Scuba's only notion of index
- Results of a node are omitted beyond a time out
  - Small missing pieces of data do not affect accuracy of computations much
  - **Lower response time is a bigger requirement**

Performance Model

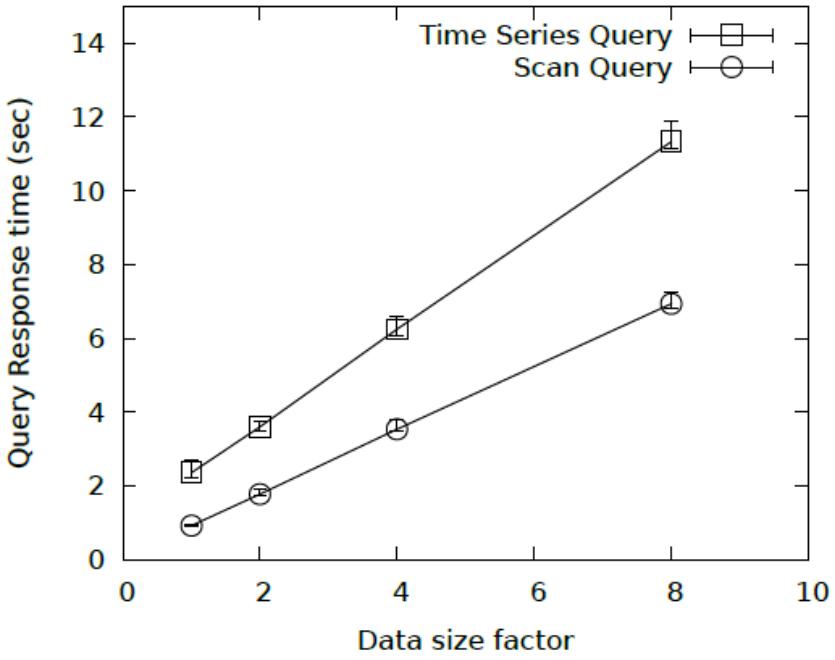
- Breaks down the latencies of different components

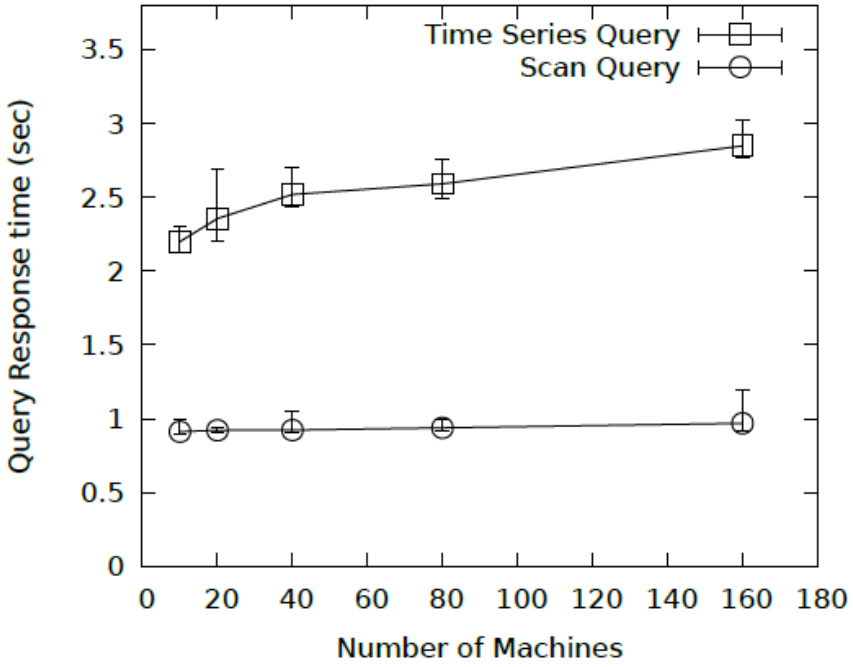
- Function of fanout, processing time at each aggregator, depth of tree

Experimental Setup and Queries

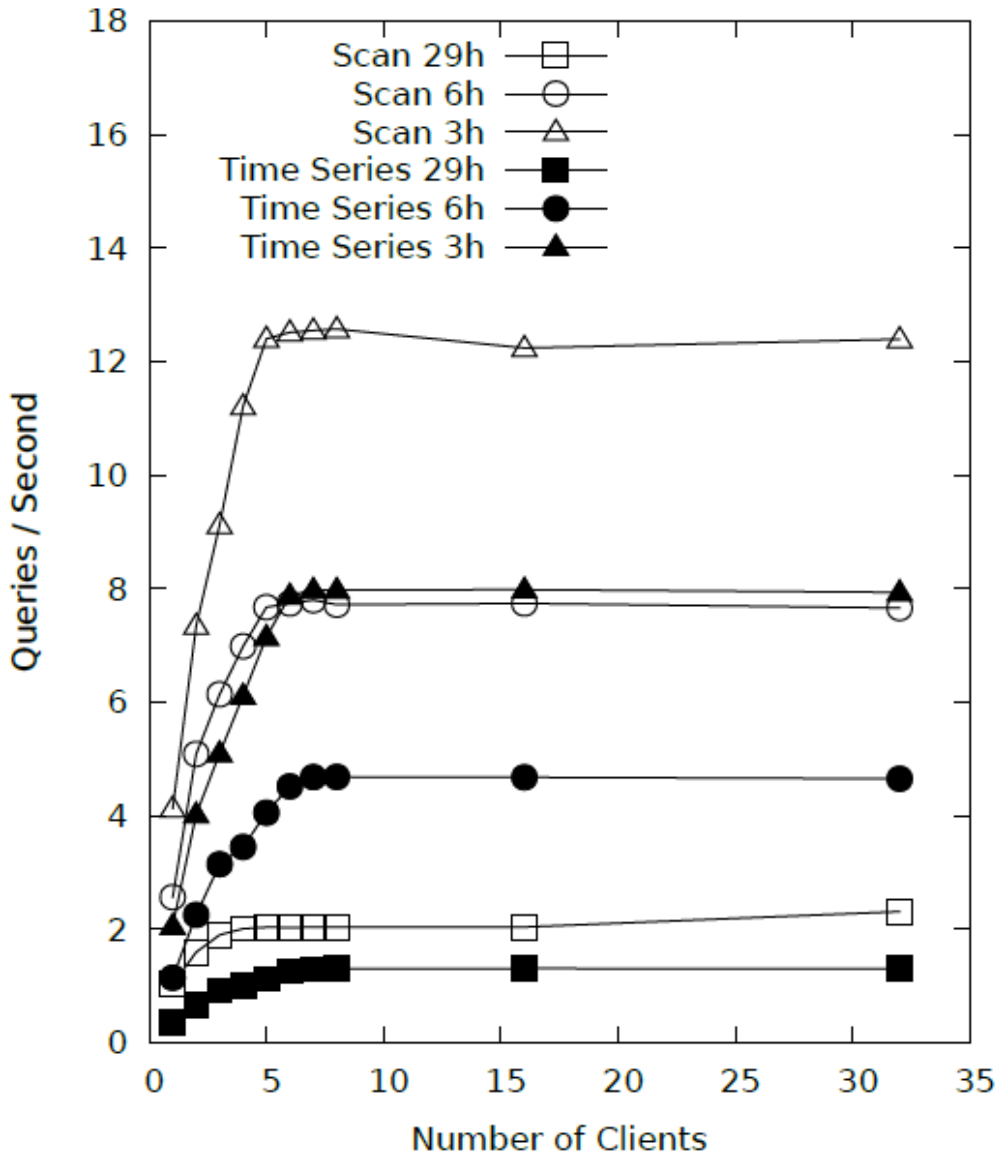
- 4 racks of 40 machines
- Machine configuration
  - Intel Xeon E5-2660
  - 2.2 GHz
  - 144 GB DRAM memory
- 10G ethernet
- Scan query, Time series query

Speedup and Scaleup





Throughput


Scan 29h
Scan 6h
Scan 3h
Time Series 29h
Time Series 6h
Time Series 3h
Queries / Second
Number of Clients
0
2
4
6
8
10
12
14
16
18
0
5
10
15
20
25
30
35

Discussion

- Details on the kind of data stored and analyzed
- Performance numbers for a wider set of queries
- Are these query throughputs good enough?
  - Might be fine for an internal system