Towards Benchmarking Stream Data Warehouses

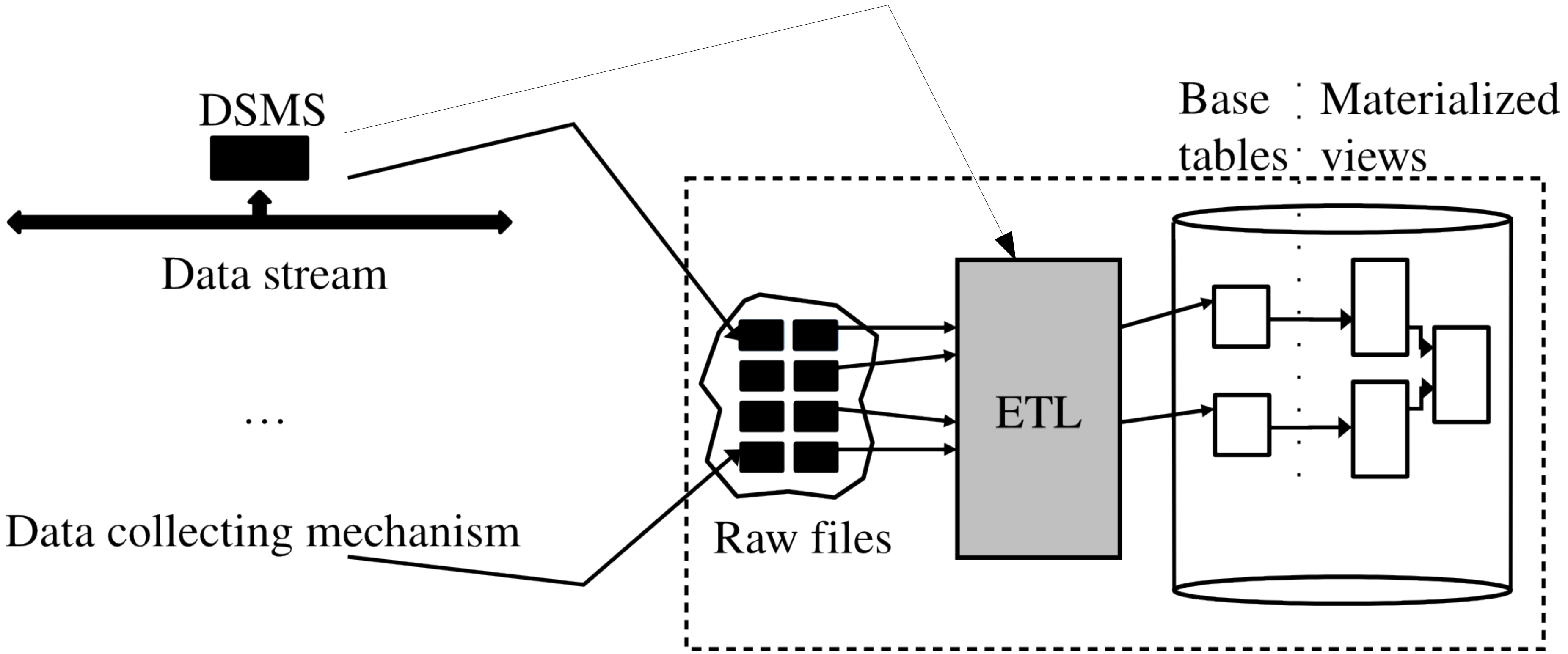
Arian Bär, Lukasz Golab

02.11.2012

Stream Data Warehouses

- A data warehouse that is (nearly) continuously loaded
- Enables real-time/historical analytics and applications

Stream Data Warehouses



Research Issues

- Goal: ensure data freshness

- Fast/streaming ETL
  - Streaming joins
- Fast data load and propagation
  - Temporal partitioning
  - Incremental view refresh
    - Golab et al, Stream warehousing with Data Depot, SIGMOD 2009
  - View update scheduling
    - Golab et al, Scalable scheduling of updates in stream data warehouses, TKDE 2012

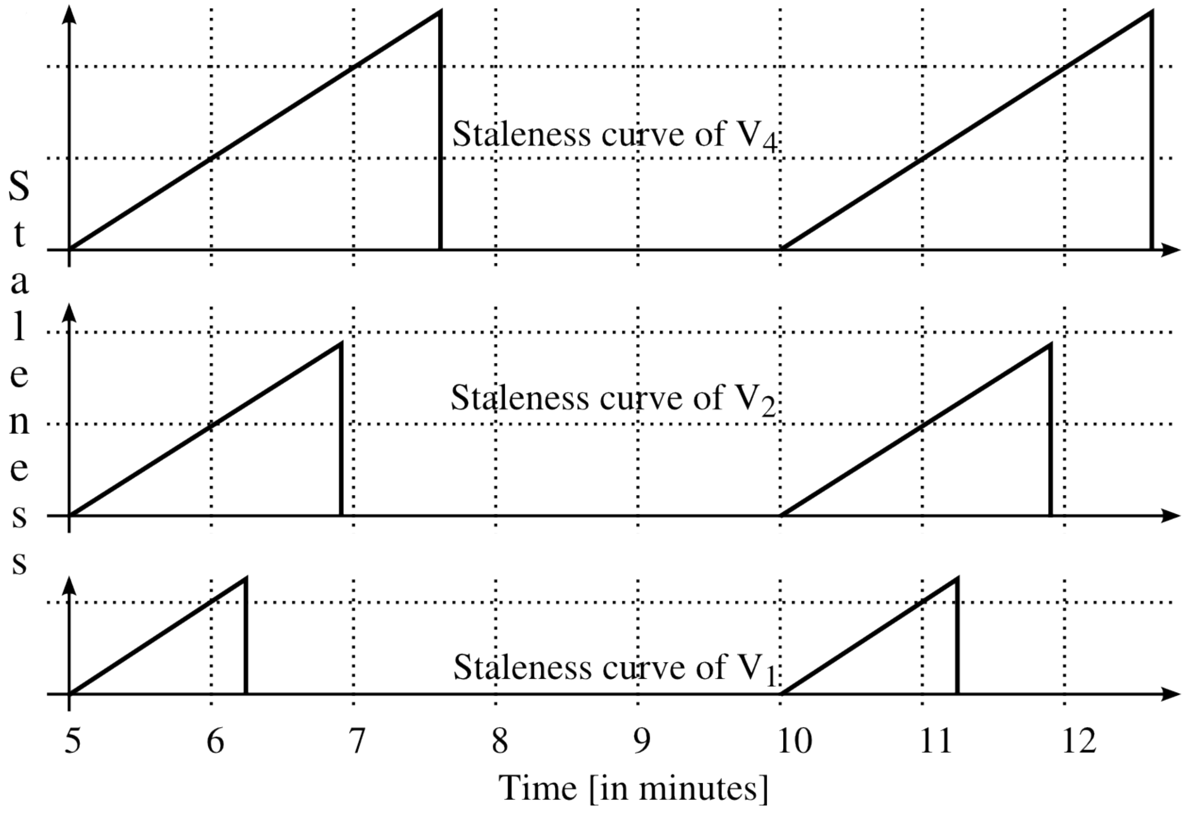
Measuring Freshness

- Use a data steam benchmark?
  - Focus on throughput; no persistent storage
- Use a data warehouse/OLAP benchmark?
  - Focus on query performance + periodic batch updates
- What we need
  - Translate metrics such as throughput and response time to data freshness/staleness

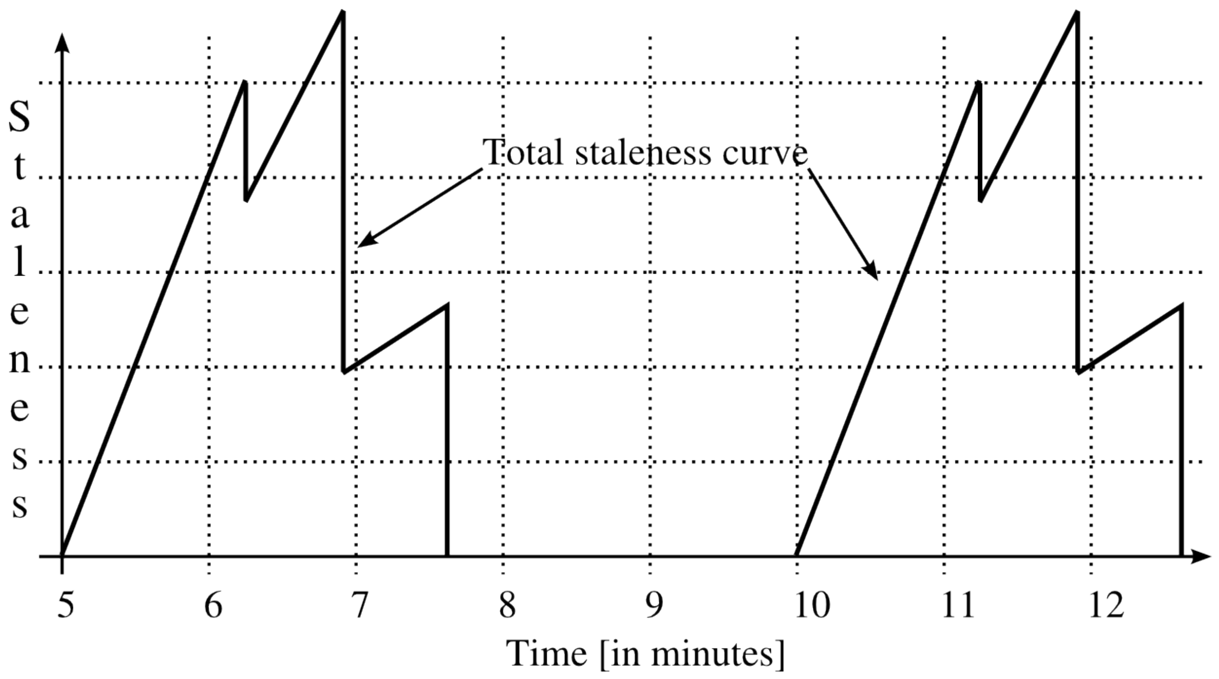
Basic Ingredients

- Define a staleness function wrt time
  - One per table; add up to get total for the warehouse
  - One implementation: staleness begins to accrue (for the base table and all associated views) when a new batch of data arrives
    - Many other definitions possible – e.g., binary
- Track over time
  - Get a staleness vs. time plot
- Return
  - Avg staleness per unit time
  - Min/max/variance over time
  - Priority-weighted staleness
  - The plot itself ...
  - … also query response times

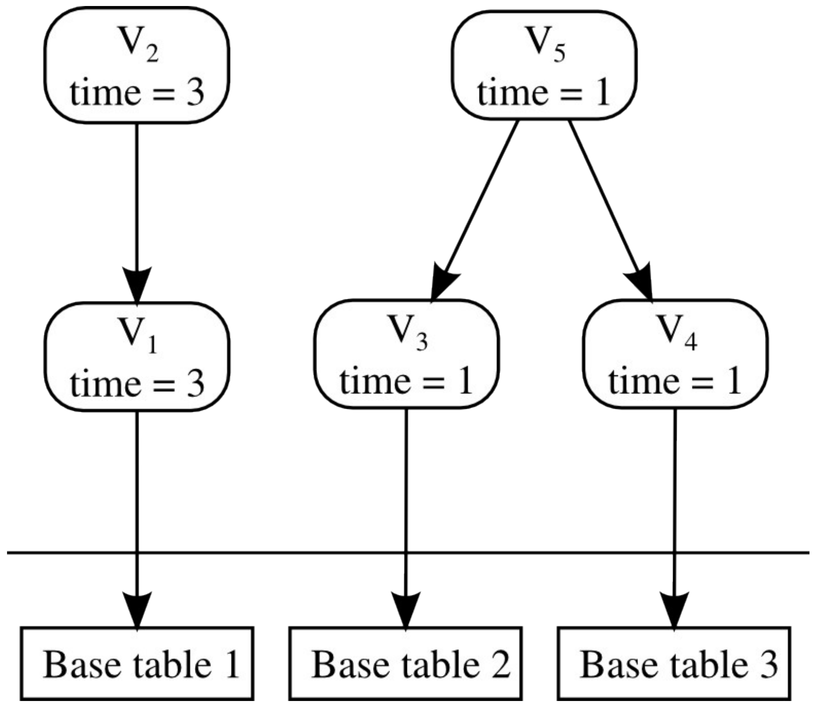
Staleness Plots

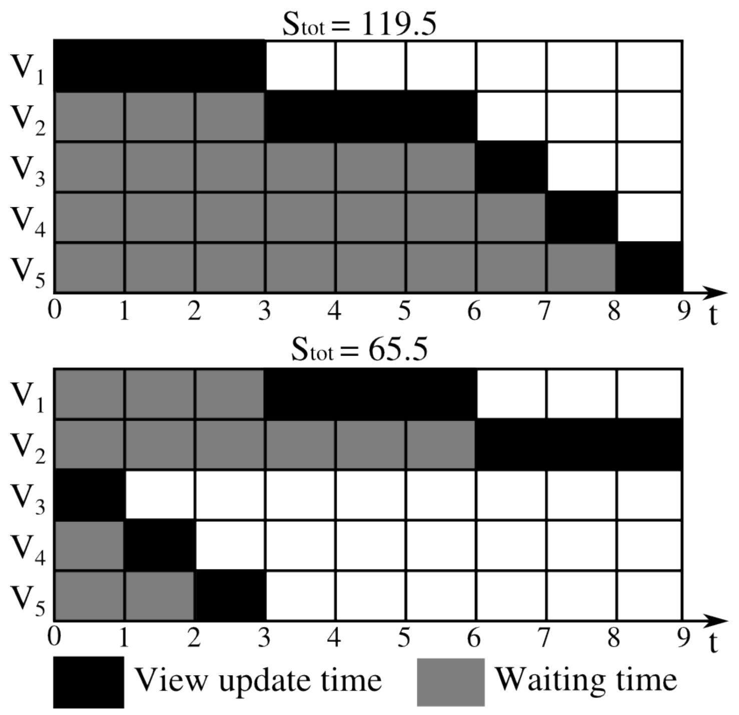


Total Staleness



Factors Influencing Staleness



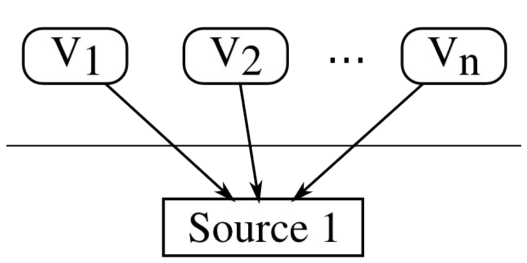


- ETL, data load, view update times
- Update order

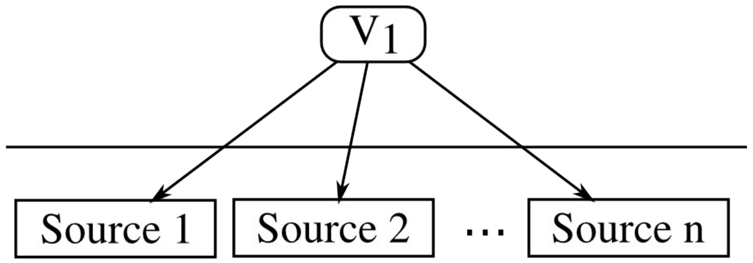
Benchmark Structure

- Data generator sends files to the SDW
- System executes a worload consisting of
  - Base table loads and materialized view updates (including indices) on arrival of newdata
  - Ad-hoc queries scheduled randomly
  - (Don't want to wait till the end to test query performance)
- Vary data speed and volume
  - Bursty workload will test overload performance
- Repeat for different view hierarchies

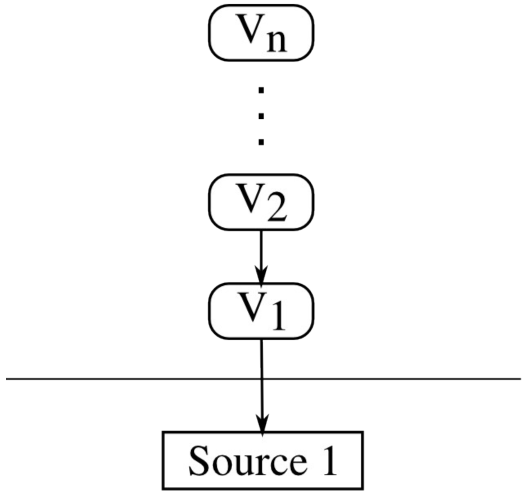
Example View Hierarchies



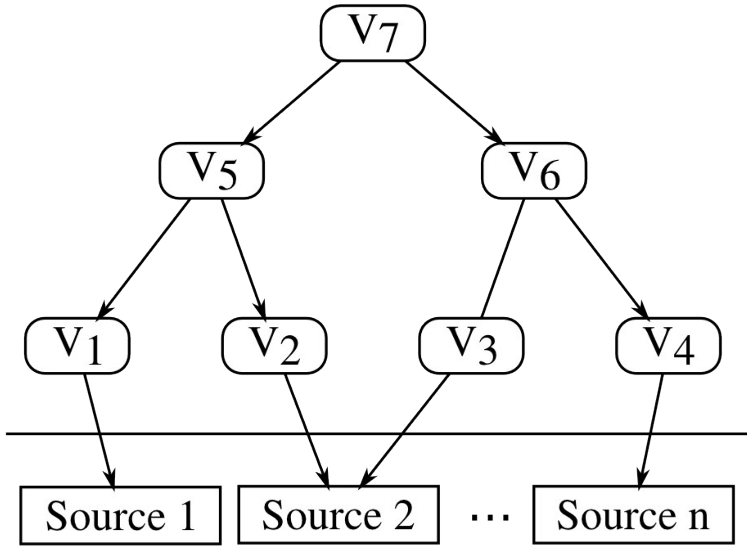
Many Views



Many Sources



Deep Hierarchy



Nested Hierarchy

Conclusions and Future/Ongoing Work

- Proposal for a SDW benchmark framework
  - Focus on data freshness over time
  - Interpretable results
- Ongoing work
  - Benchmark implementation
  - Efficient incremental view update
  - Freshness (and completeness) as data quality metric
  - Freshness in a distributed SDW