Industry Trends

http://goo.gl/7vHXb0

CS5200 DBMS
Bruce Chhay

Industry Trends

L1. Data Analysis Systems

NoSQL

- Non/Not only SQL: highly scalable and available, usually key-value storage instead of relational.
- Semi-structured, so type is not guaranteed. A “type” can have thousands of columns.
- SQL-like, but no standard language.
- Compromises consistency in favor of availability.

CAP & BASE

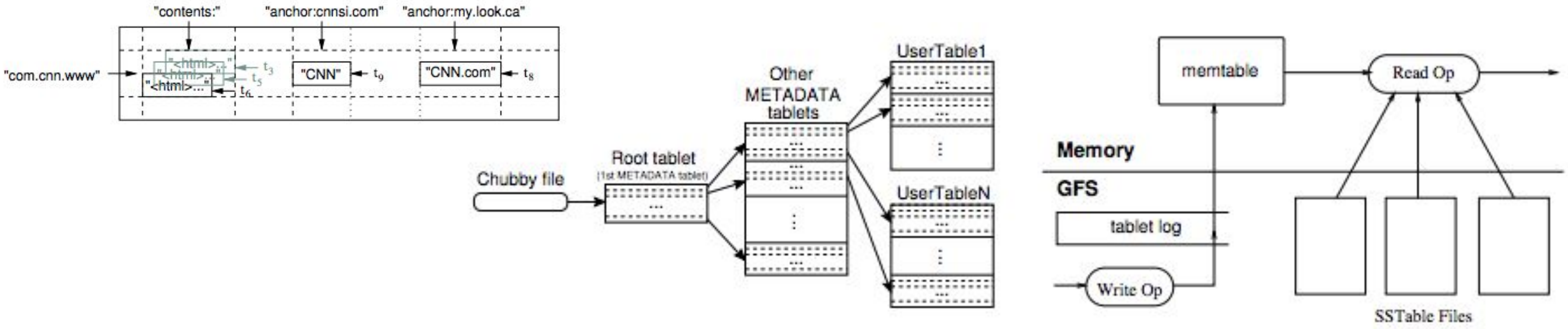
- CAP Theorem: a distributed system cannot simultaneously provide all three of the guarantees:
  - Consistency - all nodes have the same copy of data (note this is a little different than ACID consistency, which ensures integrity).
  - Availability - every request is serviced.
  - Partition tolerance - system operation despite message loss and/or failure.
  - Scenario: link between nodes breaks, and each side receives conflicting requests.
- Consistency is compromised for high availability and partition tolerance.

CAP & BASE

- Weaker consistency with BASE: Basically Available Soft-state Eventual-consistency.
  - State/data replication will eventually converge when no new updates are made to a given object.
  - Convergence must resolve conflicts (e.g. last writer wins).
  - Compare to ACID isolation (slide). BASE is for different use cases.

BigTable

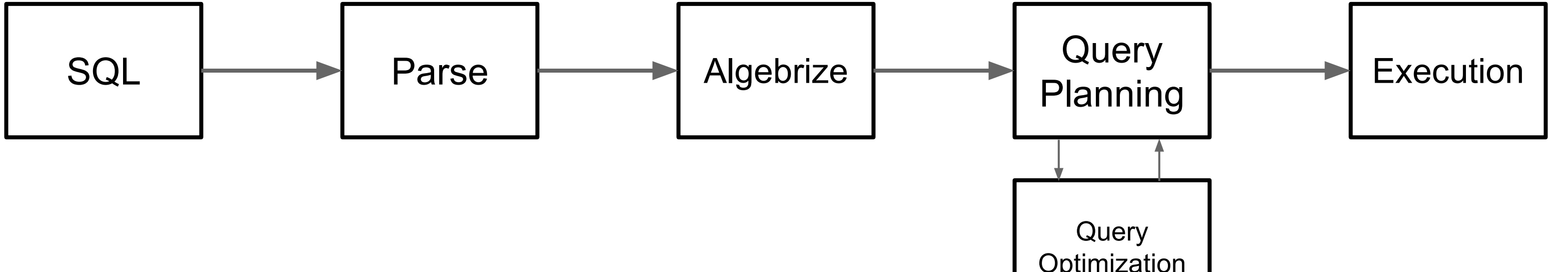
- Similarly HBase, Cassandra, …
- A “row” is a dictionary that consists of “columns” (actually grouped into column families, which specifies type).
- Data model: (row:string, column:string, time:int64) → string.
- Updates per row are atomic. Multi-version concurrency control by adding a timestamp for each cell.
- Caching in memtables, storage in SSTable (sorted by keys, plus an index)



http://static.googleusercontent.com/media/research.google.com/en/us/archive/bigtable-osdi06.pdf

NewSQL

- Highly consistent and available.
- Support for a relational data model, SQL, transactions (ACID).
- Spanner:
  - https://cloud.google.com/spanner/
  - TrueTime uses atomic clocks/GPS for synchronizing time across distributed nodes.
  - Paxos for data replication. Elected leader resolves concurrent transactions. As long as majority of replicas are alive, processing continues (share nothing). Data is replicated in shards.



Industry Trends

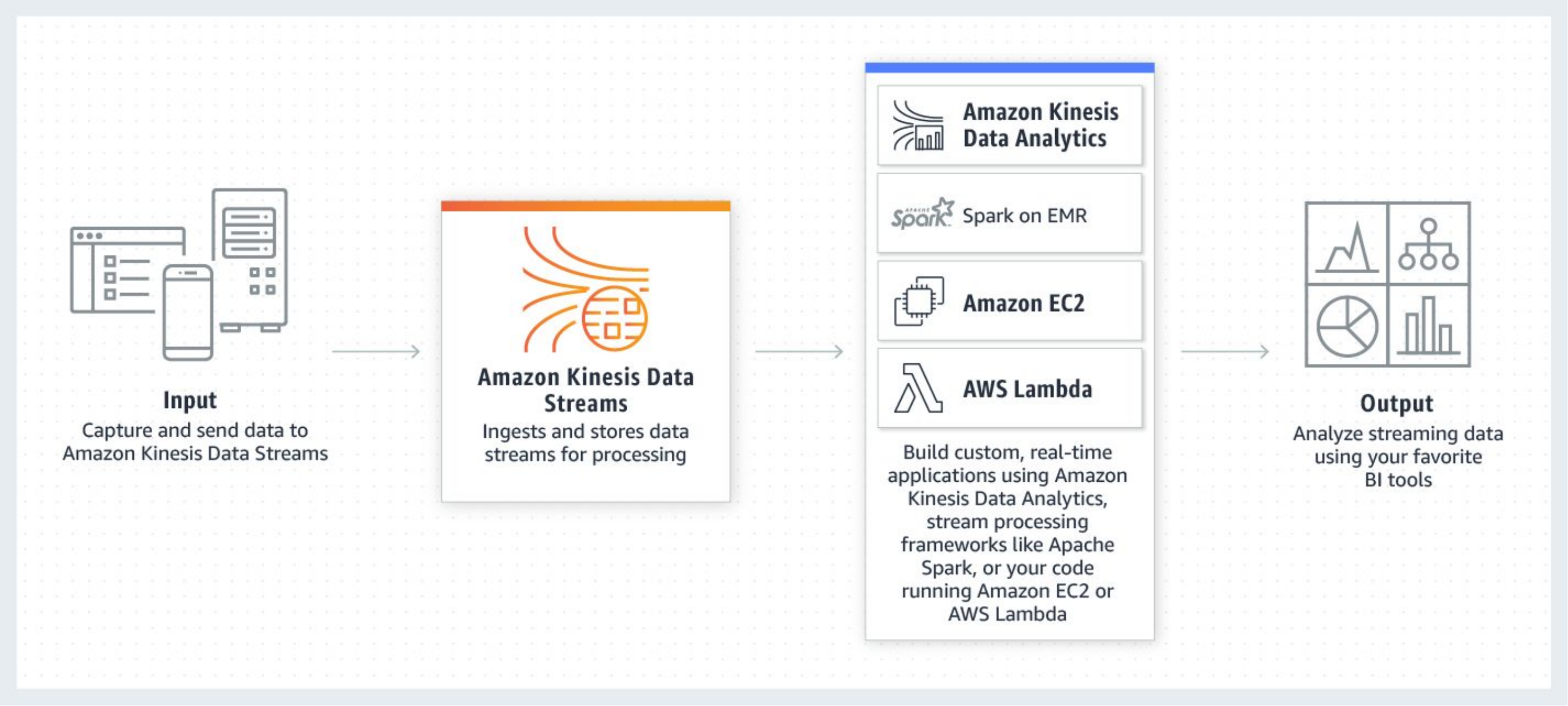
L2. Big Data Architectures

Big Data

- Existing tools are limited in the ability to derive insight from the exponential growth in data.
  Estimates that our data doubles every two years.
- Big Data V’s: volume, velocity, variety.
- Trends:
  - Capture - orchestrate millions of input sources
  - Transform (and store) - large-scale processing
  - Analysis - predictive analytics

Architecture Example

- Amazon Kinesis: real-time streaming analytics.



Input

Capture and send data to Amazon Kinesis Data Streams

→

Amazon Kinesis Data Streams

Ingests and stores data streams for processing

→

Amazon Kinesis Data Analytics

Spark on EMR

Amazon EC2

AWS Lambda

Build custom, real-time applications using Amazon Kinesis Data Analytics, stream processing frameworks like Apache Spark, or your code running Amazon EC2 or AWS Lambda

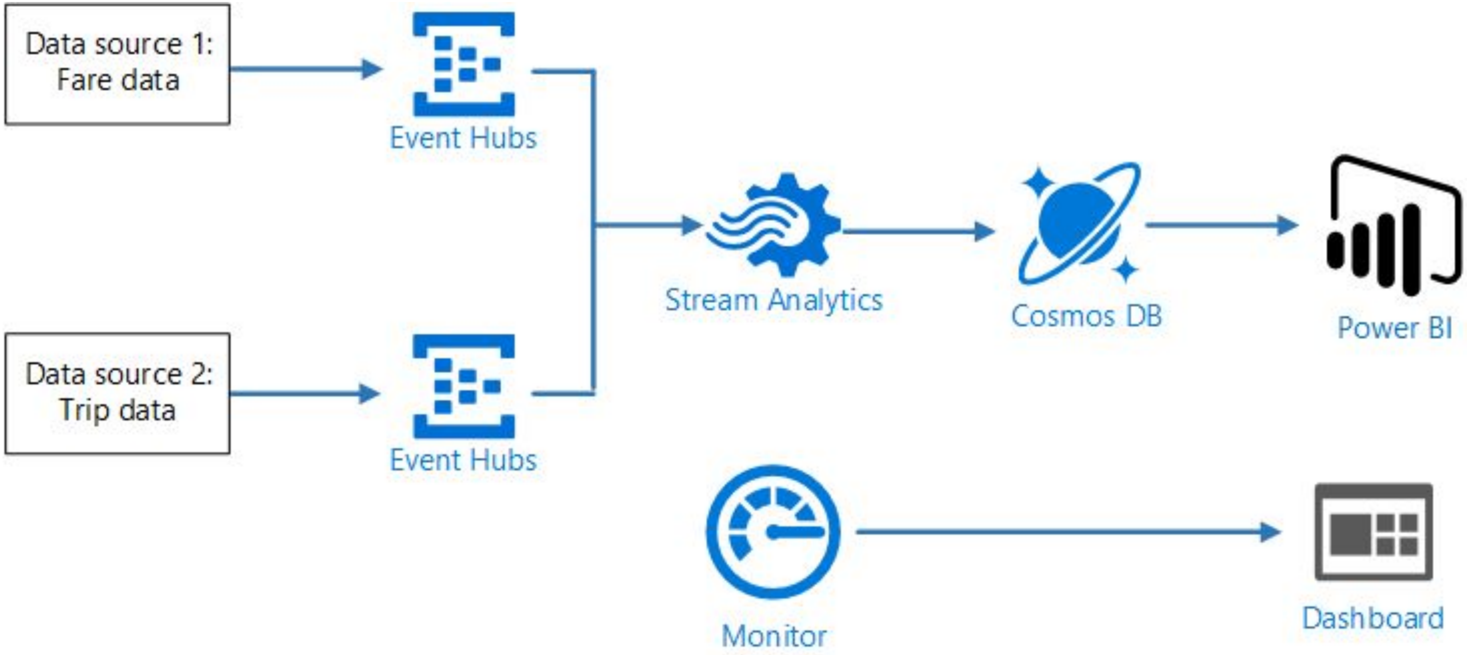
→

Output

Analyze streaming data using your favorite BI tools

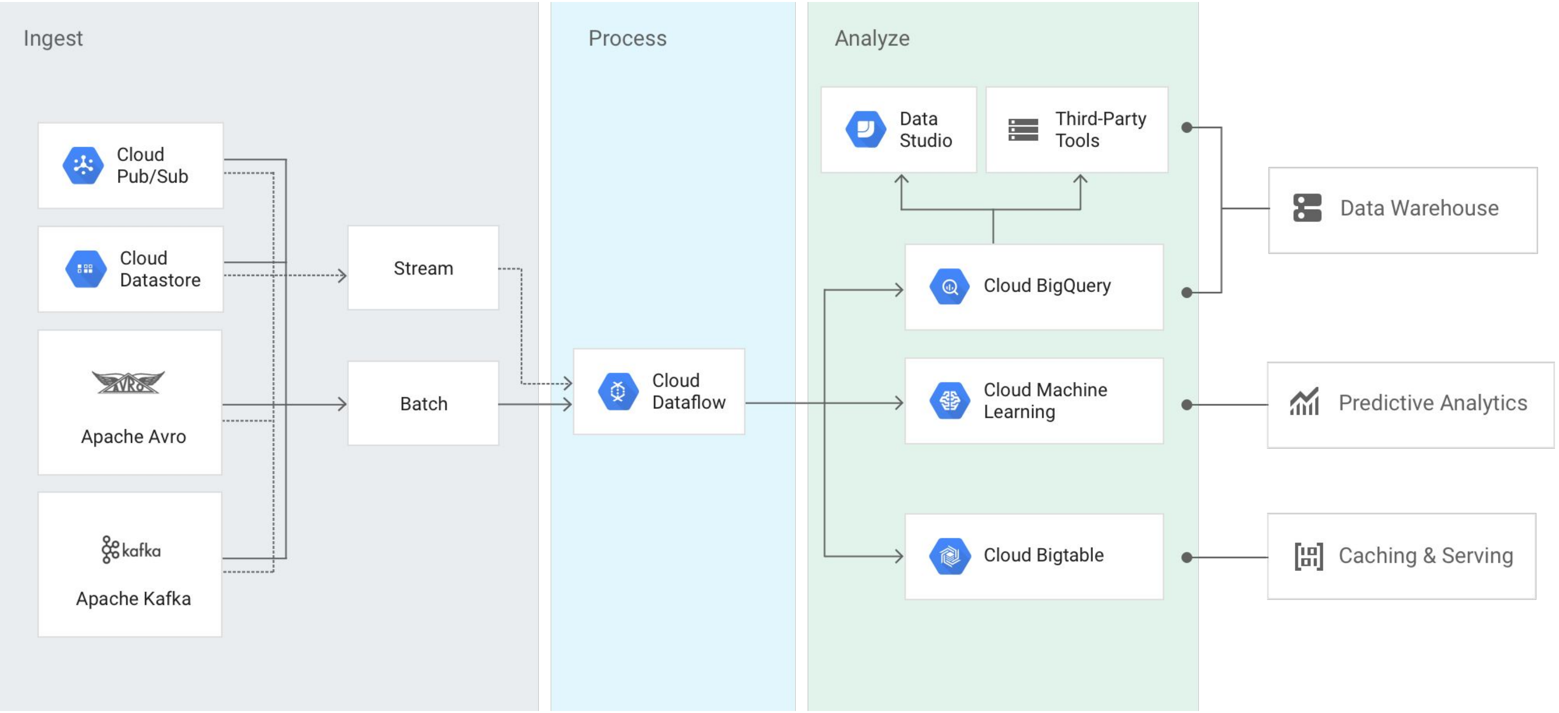
Architecture Example

- (Microsoft) Azure Stream Analytics: real-time streaming analytics.



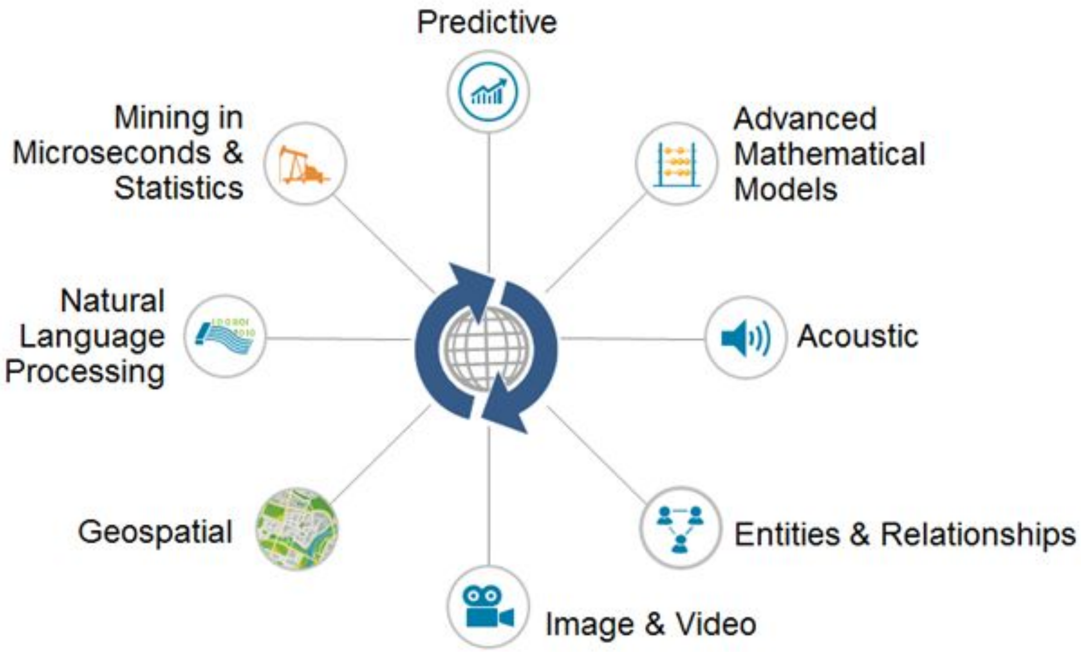
Architecture Example

- (Google) Cloud Dataflow: real-time streaming analytics.



Architecture Example

- (IBM) Stream Computing: real-time streaming analytics.


Predictive
Mining in Microseconds & Statistics
Advanced Mathematical Models
Natural Language Processing
Acoustic
Geospatial
Entities & Relationships
Image & Video

Industry Trends

L3. Applications

Predictive Analytics

- Statistical inference vs descriptive statistics:
  Inferring properties (based on probability theory) and testing hypotheses (prediction) vs quantitative summarization.
- Implementations:
  http://aws.amazon.com/machine-learning/
  https://azure.microsoft.com/en-us/services/machine-learning-studio/
  https://cloud.google.com/ml-engine/ https://www.tensorflow.org/

Example

- Netflix Prize: $1M to improve movie predictions based on preferences.
  http://www.netflixprize.com/
- 100M ratings training set: {user,movie,date,grade},
  2.8M ratings qualifying set: {user,movie,date,?}.
- Evaluated on root-mean-square error of predicted grades and actual grades (estimate of standard deviation of the differences).
- Collaborative filtering: make a prediction (filtering) based on a user's preferences.
  - Many users and many preference inputs (possibly many different sources).
  - Prediction for user A can be based on user B preferences when A and B are determined to be similar.
  - Other uses: sensors, financial, marketing.

MapReduce

- Functional programming:
  - map(function, list): Return a list where function is applied to each item in list.
    Example: map(lambda x: x+1, [1,2,3,4]) →
    [1,2,3,4]: 1+1 = 2
    [1,2,3,4]: 2+1 = 3
    [1,2,3,4]: 3+1 = 4
    [1,2,3,4]: 4+1 = 5
    == [2,3,4,5]
  - reduce(function, list): Return the value of cumulatively applying function to each item in list.
    Example: reduce(lambda x,y: x+y, [1,2,3,4]) →
    1 +
    2 +
    3 +
    4
    == 10

MapReduce

- Distributed:
    - map(k1,v1) -> list(k2,v2): apply map function to every k,v pair and write a new list of k,v pairs to temporary storage.
    - shuffle: redistribute temporary output of map() based on output keys, k2.
    - reduce(k2, list(v2)) -> list(v3): one worker processes a list of output values v2 for a given output key k2, and produces a single v3. The output of all reduce workers are collected in list(v3).
    - For massively parallelizable problems (potentially large communication costs).
    - Implementations: Google MapReduce, Apache Hadoop, Amazon EMR.

Example

- Word count across all works of Shakespeare.

```
map(doc_name, doc_content)
    yield (word,1) for word in doc_content;
→ [(anon,1), (brave,1), … (anon,1) ... ]

shuffle (including partition and comparison);

reduce(word, occurrences)
    subtotal=0;
    subtotal += occurrence for occurrence in occurrences;
    yield (word,subtotal);
→ [(anon,9),(brave,27), … ]
```

Example

- Word count across all works of Shakespeare.

```
map(doc_name, doc_content)
    yield (word,1) for word in doc_content;
→ [(anon,1), (brave,1), … (anon,1) ... ]

shuffle (including partition and comparison);

reduce(word, occurrences)
    subtotal=0;
    subtotal += occurrence for occurrence in occurrences;
    yield (word,subtotal);
→ [(anon,9),(brave,27), … ]
```

Example

- Word count across all works of Shakespeare.

```
map(doc_name, doc_content)
      yield (word,1) for word in doc_content;
→ [(anon,1), (brave,1), … (anon,1) ... ]

shuffle (including partition and comparison);

reduce(word, occurrences)
      subtotal=0;
      subtotal += occurrence for occurrence in occurrences;
      yield (word,subtotal);
→ [(anon,9),(brave,27), … ]
```

Example

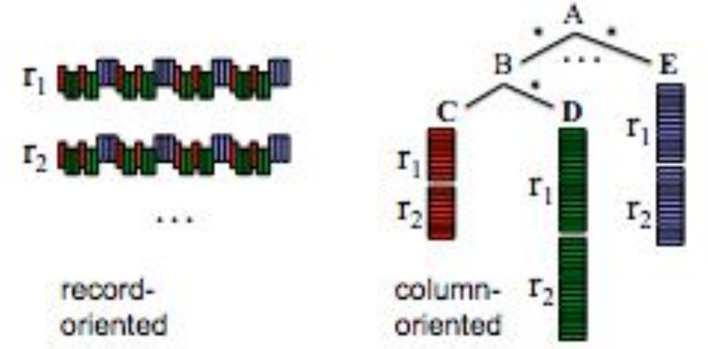
- SQL Aggregation.
  MongoDB: Binary JSON doc storage with dynamic schema, can use MR for GROUP BY.

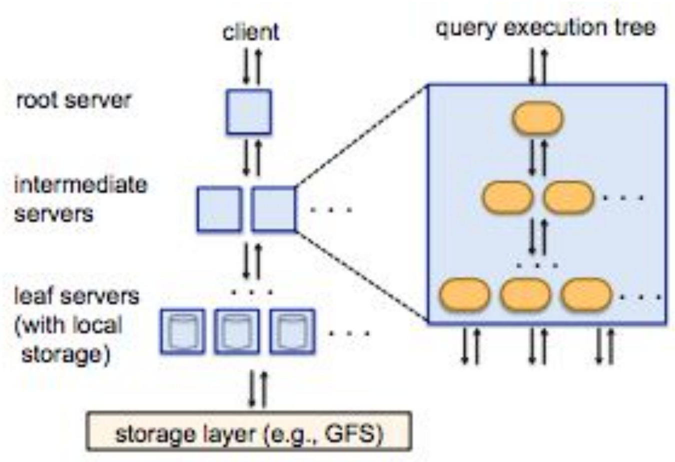
  ```
  SELECT word, COUNT(*) AS subtotal
  FROM Shakespeare
  GROUP BY word
  ORDER BY subtotal DESC;
  ```

- WHERE: can this be implemented in map()? Per-row selection filtering.
- JOIN: shuffle() and reduce()? Redistribution for set operations and produce output for a given key; potentially multiple passes.

Google BigQuery

- Real-time, interactive big data service; analyze TBs in seconds with SQL.
- Based on Dremel: tree-like architecture, columnar storage.
- Query evaluation similar to a relational database.





http://static.googleusercontent.com/media/research.google.com/en/us/pubs/archive/36632.pdf

Google BigQuery

- Scale: Current SATA disk controllers take 25 minutes to scan 1TB (600MB/s).
  Sample query: 2TB table, 1.2TB scanned, regexp, aggregation, sorting. < 10 sec.

```
SELECT title, SUM(views) as views
FROM [bigquery-samples:wikipedia_pageviews.201007]
WHERE
  NOT title CONTAINS ':'  AND wikimedia_project='wp'
  AND language='en'  AND REGEXP_MATCH(title, r'^G.*o.*g.*e$')
GROUP BY title
ORDER BY views DESC
LIMIT 100;
```

- ETL vs ELT

Discussion

Could you see your project moving to a big data architecture? If so, describe how. Also, what applications would be relevant for your project? For example, can you apply predictive analytics? If so, describe the “predictive” functionality you would deliver and the value for your users.

TRACE Evaluations

- You can use your laptop, smartphone, tablet, etc.
- You received an email announcement with a link to the login page for completing the TRACE survey. Answers are completely anonymous and confidential.
- Important for improving teaching, in this class and NEU.
- If you would like to have a conversation, then please email, schedule a meeting, chat during break, etc.
- Any and all feedback is appreciated!

Teaching Assistant

- Responsibilities: assignment grading and helping the students succeed!
- Time commitment: 5-15 hours a week.
- Apply to the pool, and a match will be selected.
- Preference for students that recently completed the course.

Projects

- Milestone 6