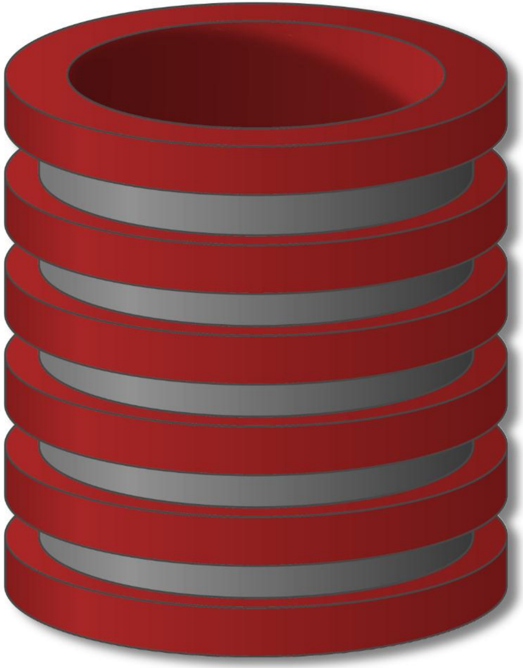


NoSQL Systems

Overview
(as of November 2011)

Jennifer Widom

NoSQL Systems

- Not every data management/analysis problem is best solved *exclusively* using a traditional DBMS
- "NoSQL" = "Not Only SQL"

NoSQL Systems

Alternative to traditional relational DBMS

+ Flexible schema ✓
+ Quicker/cheaper to set up ✓
+ Massive scalability ✓
+ Relaxed consistency ✓ → higher performance & availability

– No declarative query language → more programming
– Relaxed consistency → fewer guarantees

NoSQL Systems

Several incarnations

- MapReduce framework ~ OLAP
- Key-value stores ~ OLTP → Column Stores
- Document stores
- Graph database systems

MapReduce Framework

Originally from Google, open source Hadoop

- No data model, data stored in files

  GFS
  HDFS

- User provides specific functions

  map( ) reduce( )
  reader( ) writer( ) combiner( )

- System provides data processing "glue", fault-tolerance, scalability

Map and Reduce Functions

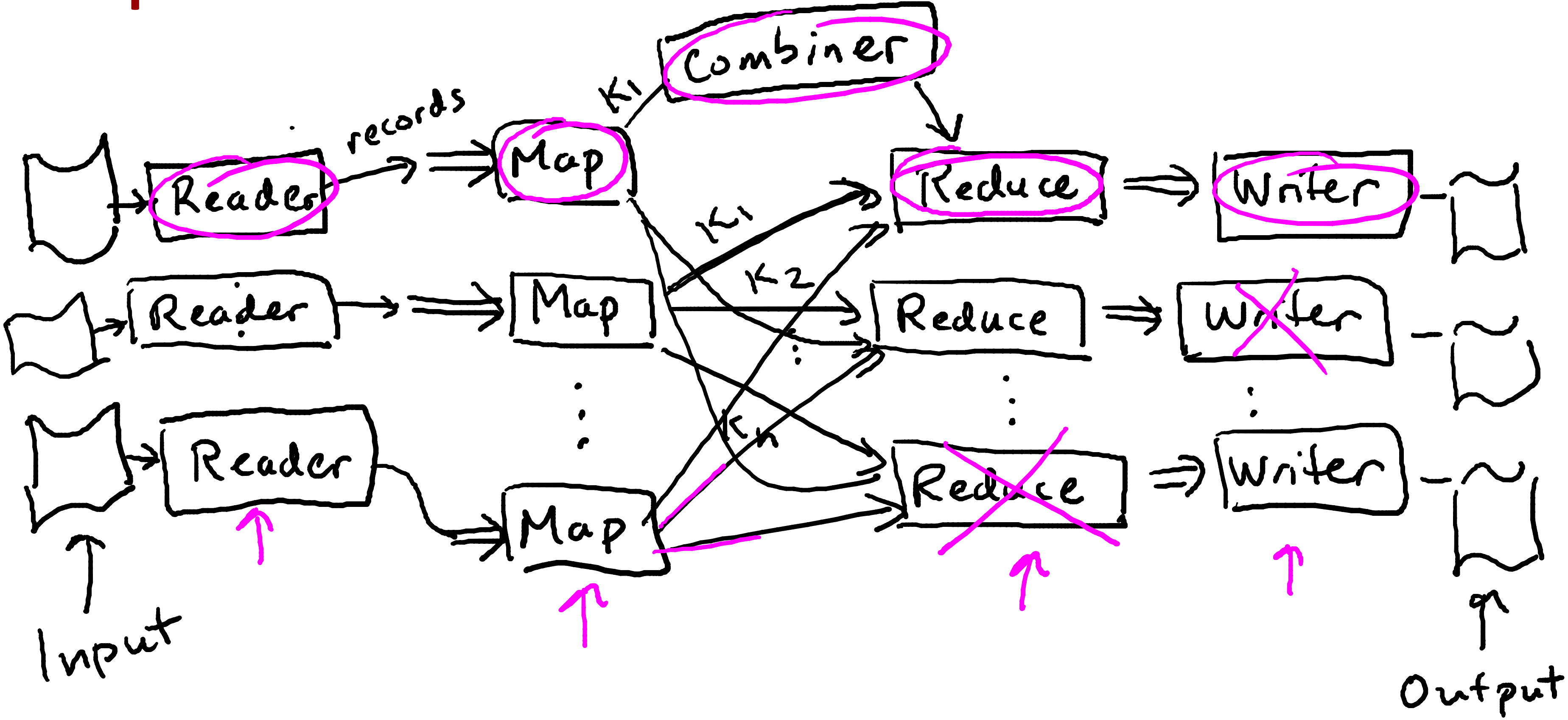
Map: Divide problem into subproblems

map(item) → 0 or more <key, value> pairs

Reduce: Do work on subproblems, combine results

reduce(key, list-of-values) → 0 or more records

MapReduce Architecture



MapReduce Example: Web log analysis

Each record: UserID, URL, timestamp, additional-info ←

Task: Count number of accesses for each domain (inside URL)

map(record) → $\langle \underbrace{\text{domain}}_{\text{key}}, \text{NULL} \rangle$

reduce($\underbrace{\text{domain}}_{\text{key}}$, list ~~of NULL~~ list of counts) → $\langle$domain, ~~count~~ sum$\rangle$

combine(domain, list of NULL) → $\langle$domain, count$\rangle$

MapReduce Example (modified #1)

Each record: UserID, URL, timestamp, additional-info

Task: Total "value" of accesses for each domain based on additional-info ←

map(record) → ⟨domain, score⟩

reduce (domain, list of scores) → ⟨domain, sum⟩

MapReduce Example (modified #2)

Each record: UserID, URL, timestamp, additional-info

Separate records: UserID, name, age, gender, …

) ⋈

Task: Total "value" of accesses for each domain based on user attributes

MapReduce Framework

- No data model, data stored in files ✓
- User provides specific functions ✓
- System provides data processing “glue”, fault-tolerance, scalability ✓

MapReduce Framework

Schemas and declarative queries are missed

- **Hive** – schemas, SQL-like query language
- **Pig** – more imperative but with relational operators
  - Both compile to "workflow" of Hadoop (MapReduce) jobs

Dryad allows user to specify workflow

- Also DryadLINQ language

Key-Value Stores "OLTP"

Extremely simple interface

- Data model: (key, value) pairs
- Operations: Insert(key,value), Fetch(key),
  Update(key), Delete(key)

Implementation: efficiency, scalability, fault-tolerance

- Records distributed to nodes based on key
- Replication
- Single-record transactions, "eventual consistency"

Key-Value Stores

Extremely simple interface

structure

- Data model: (key, value) pairs
- Operations: Insert(key,value), Fetch(key),
  Update(key), Delete(key)
- Some allow (non-uniform) columns within value
- Some allow Fetch on range of keys

$2 < key < 10$

Example systems

- Google BigTable, Amazon Dynamo, Cassandra, Voldemort, HBase, …

Document Stores

Like Key-Value Stores except value is document

- Data model: (key, document) pairs
- Document: JSON, XML, other semistructured formats
- Basic operations: Insert(key,document), Fetch(key),
  Update(key), Delete(key)
- Also Fetch based on document contents

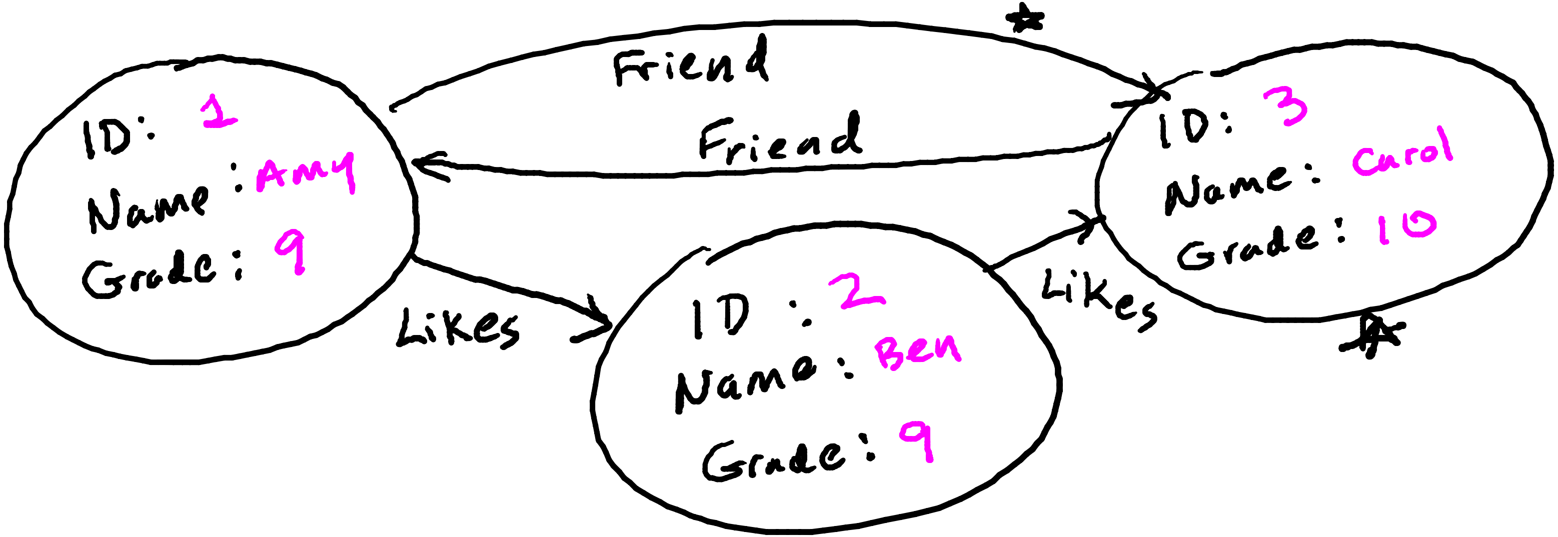
System/format specific

Example systems

- CouchDB, MongoDB, SimpleDB, …

Graph Database Systems

- Data model: nodes and edges
- Nodes may have properties (including ID)
- Edges may have labels or roles



Graph Database Systems

- Interfaces and query languages vary
- Single-step versus "path expressions" versus full recursion
- Example systems

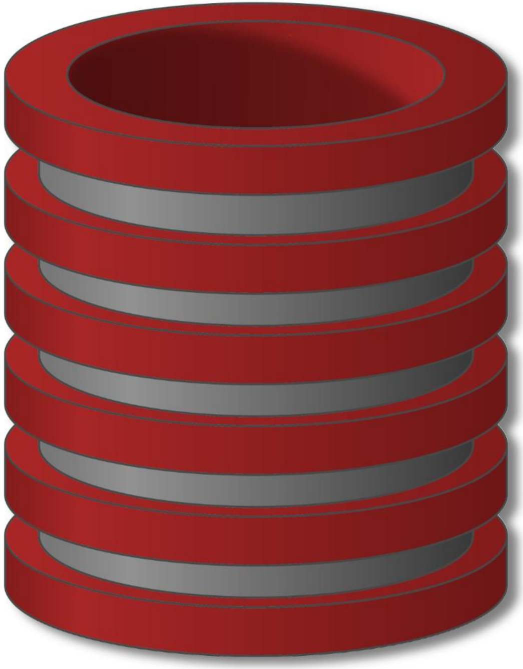
  Neo4j, FlockDB, Pregel, …
- RDF "triple stores" can map to graph databases

NoSQL Systems

- "NoSQL" = "Not Only SQL"

  Not every data management/analysis problem is best solved *exclusively* using a traditional DBMS

- Current incarnations
  - MapReduce framework ✓
  - Key-value stores ✓
  - Document stores ✓
  - Graph database systems ✓



NoSQL Systems

Overview
(as of November 2011)

Jennifer Widom