Google's MapReduce and Sawzall

CSCI 334
Stephen Freund

Count Words

cow: 20
moo: 15
the: 45
purple: 3
wombat: 21

Reverse Link Map

X.html: A.html, C.html
Y.html: A.html, B.html
Z.html: C.html

Word Index

cow: A.html, B.html
moo: A.html
purple: A.html, C.html
wombat: B.html, C.html
Computations Over Data

- Word Count
- Reverse Link Map
- Word Index
- Links out of a domain
- Page Rank
- log file processing

But.... many terabytes or petabytes of data
- 1 terabyte = 1000 gigabytes
- 1 petabyte = 1000 terabytes

Computing Infrastructure

- Millions of computers
- Datacenters distributed around world

- Problems:
  - need to coordinate computers
  - machines fail constantly
  - network, failure, computer/data locations, etc. should be transparent to user running analyses.
MapReduce and Sawzall

- **MapReduce** (Dean and Ghemawat)
  - Map/reduce from FP
  - distributed computer management

- **Sawzall** (Pike et al.)
  - language for writing code to perform data analysis

- Papers up on web page
- **Cloud Compute Services:**
  - Hadoop, Amazon EC2, IBM SmartCloud, ...

**Summary**

- Page Rank: 24 separate map-reduce operations

- Sawzall/MapReduce execution model:
  - specify data set, map fn, reduce fn
  - most map/reduce functions < 50 lines of code
  - hides details of distributed system
  - fault tolerant, fast, flexible architecture
# of Queries for Each Latitude/Longitude

proto "querylog.proto"

queries_per_degree: table sum[lat: int][lon: int] of int;
log_record: QueryLogProto = input;
loc: Location = locationinfo(log_record.ip);
emit queries_per_degree[int(loc.lat)][int(loc.lon)] < 1;

map phase produces key-value pairs of form <(lat,lon),1>
reduce phase sums up values for each key

Page with Highest Page Rank

proto "document.proto"

max_pagerank_url:
  table maximum(1) [domain: string] of url: string
  weight pagerank: int;

doc: Document = input;
emit max_pagerank_url[domain(doc.url)] <- doc.url
  weight doc.pagerank;