CSE544: Principles of Database Systems

Parallel Databases Wrapup

Announcements

- Paper review (datalog) was due today
  - Will discuss the paper in class on Wednesday

- Homework 2 due on Sunday night

MapReduce Review

- What is the map function?
- What is the reduce function?
- What is a map task?
- What is a reduce task?
- What is a mapreduce job?

MapReduce Review

- What is the map function?
  - Takes (k,v) returns {(k1,v1),(k2,v2),…}
- What is the reduce function?
  - Takes (k,{v1,v2,…}) returns any result
- What is a map task?
  - A set of (k,v) pairs that are scheduled as a unit
- What is a reduce task?
  - A set of (k,{v1,…}) pairs scheduled as a unit
- What is a mapreduce job?
  - The entire map reduce program

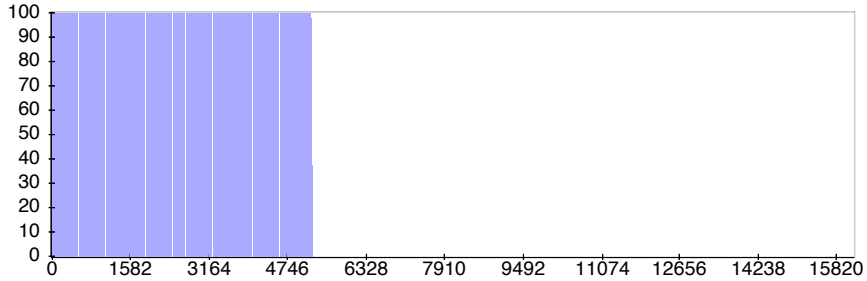
Anatomy of a Query Execution

- Running Part B of HW2
- 20 nodes = 1 master + 19 workers
- Using PARALLEL 50
- Let’s see what happened

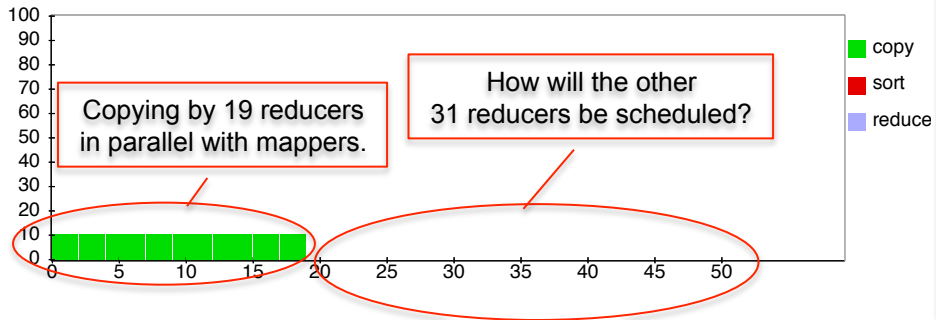
1h 16min

Only 19 reducers active, out of 50. Why?

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	33.17%	15816	10549	38	5229	0	0 / 0
reduce	4.17%	50	31	19	0	0	0 / 0



luce Completion Graph - close

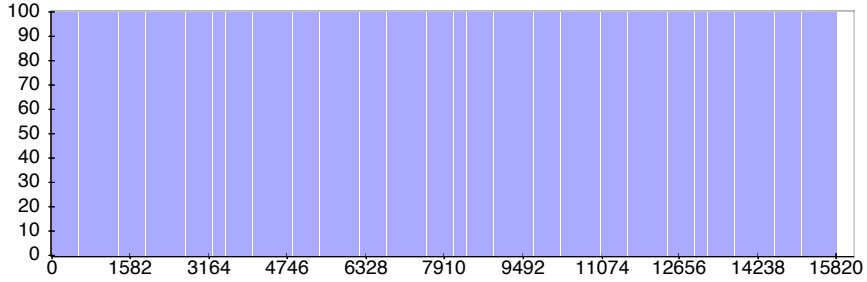


3h 50min

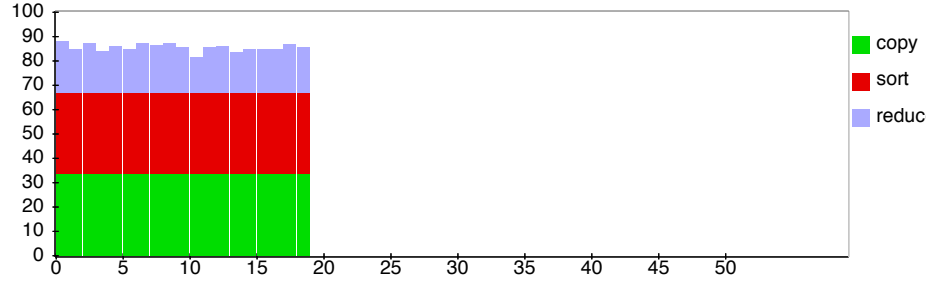
Some errors start to occur. Watch this…

Completed. Sorting, and the rest of Reduce may proceed now

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	15816	0	0	15816	0	0 / 18
reduce	32.42%	50	31	19	0	0	0 / 0



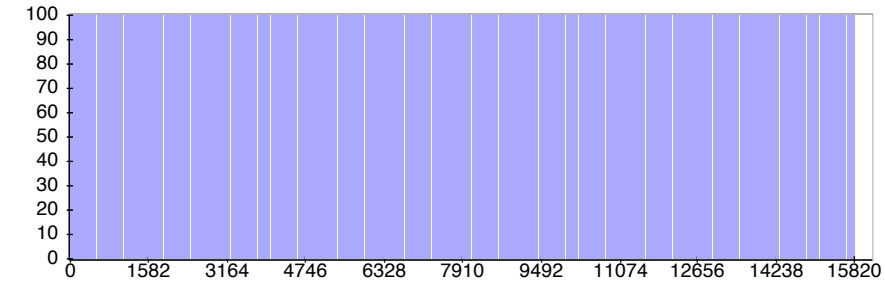
uce Completion Graph - close



3h 51min

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	15816	0	0	15816	0	0 / 18
reduce	37.72%	50	19	22	9	0	0 / 0

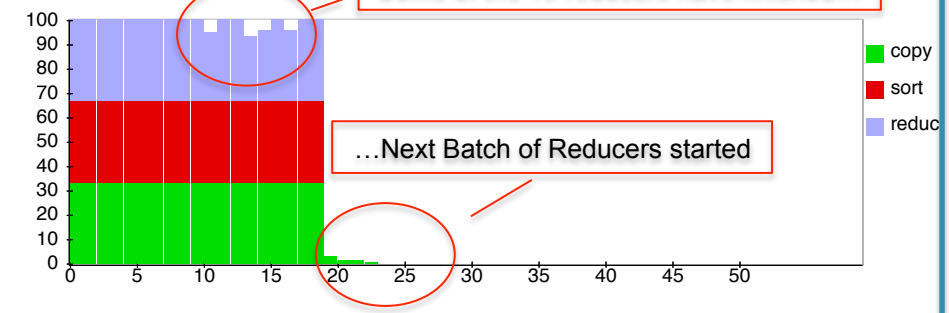
ɔ Completion Graph - close



duce Completion Graph - close

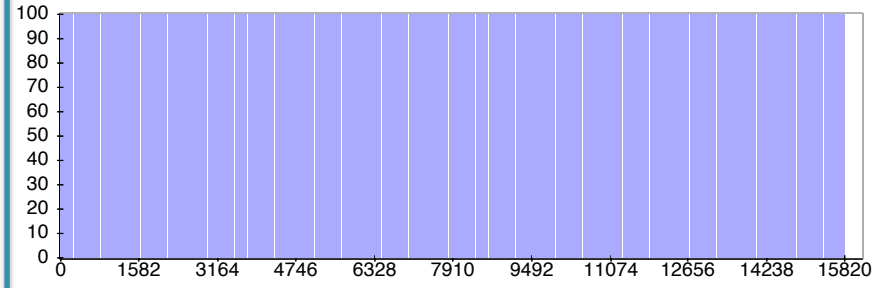
Some of the 19 reducers have finished…

…Next Batch of Reducers started



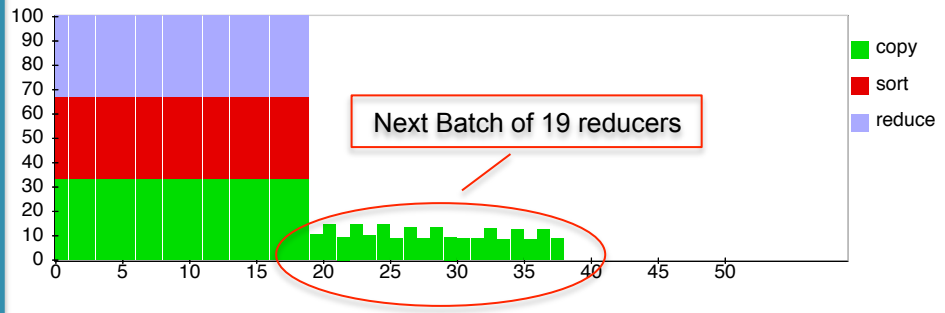
3h 52min

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	15816	0	0	15816	0	0 / 18
reduce	42.35%	50	11	20	19	0	0 / 0



ce Completion Graph - close

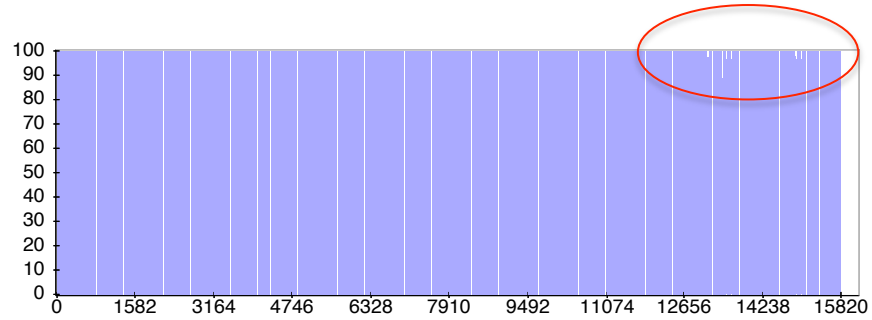
Next Batch of 19 reducers



4h 18min

Several servers failed: “fetch error”. Their map tasks need to be rerun. All reducers are waiting….

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	99.88%	15816	2638	30	13148	0	15 / 3337
reduce	48.42%	50	15	16	19	0	0 / 0

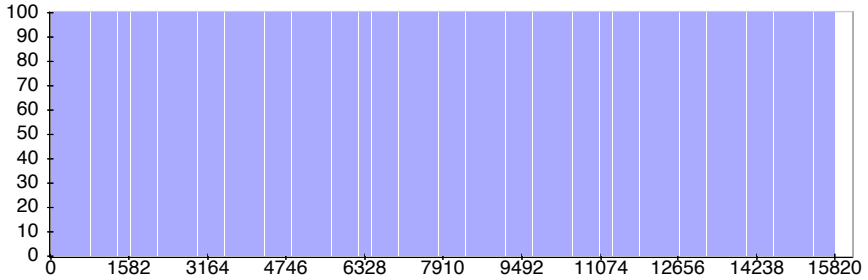


uce Completion Graph - close

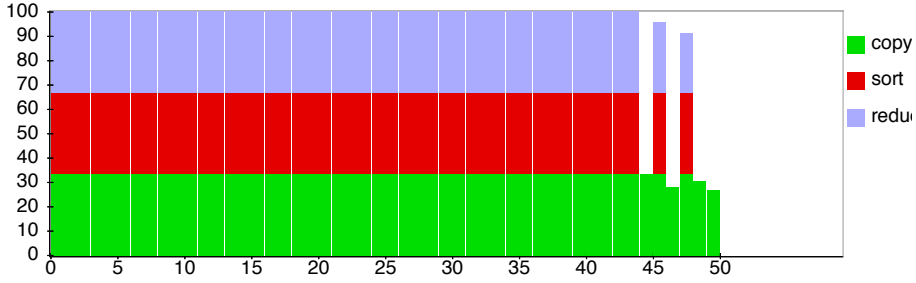
7h 10min

Mappers finished, reducers resumed.

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	15816	0	0	15816	0	26 / 5968
reduce	94.15%	50	0	6	44	0	0 / 8



uce Completion Graph - close



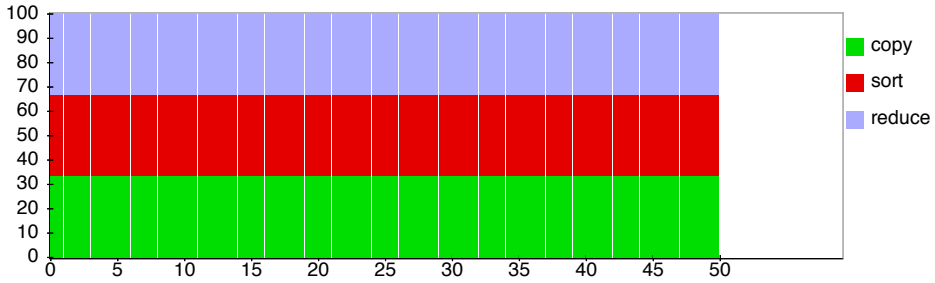
7h 20min

Success! 7hrs, 20mins.

Hadoop job_201203041905_0001 on ip-10-203-30-146

User: hadoop
Job Name: PigLatin:DefaultJobName
Job File: hdfs://10.203.30.146:9000/mnt/var/lib/hadoop/tmp/mapred/staging/hadoop/.staging/job_201203041905_0001/job.xml
Submit Host: ip-10-203-30-146.ec2.internal
Submit Host Address: 10.203.30.146
Job-ACLs: All users are allowed
Job Setup: Successful
Status: Succeeded
Started at: Sun Mar 04 19:08:29 UTC 2012
Finished at: Mon Mar 05 02:28:39 UTC 2012
Finished in: 7hrs, 20mins, 10sec
Job Cleanup: Successful
Black-listed TaskTrackers: 3

Kind	% Complete	Num Tasks	Pending	Running	Complete	Killed	Failed/Killed Task Attempts
map	100.00%	15816	0	0	15816	0	26 / 5968
reduce	100.00%	50	0	0	50	0	0 / 14



Doc(key, word)

MAP=GROUP BY, REDUCE=Aggregate

```
SELECT word, sum(1)
FROM Doc
GROUP BY word
```

Joins in MapReduce

- If MR is GROUP-BY plus AGGREGATE, then how do we compute R(A,B) ⋈ S(B,C) using MR?

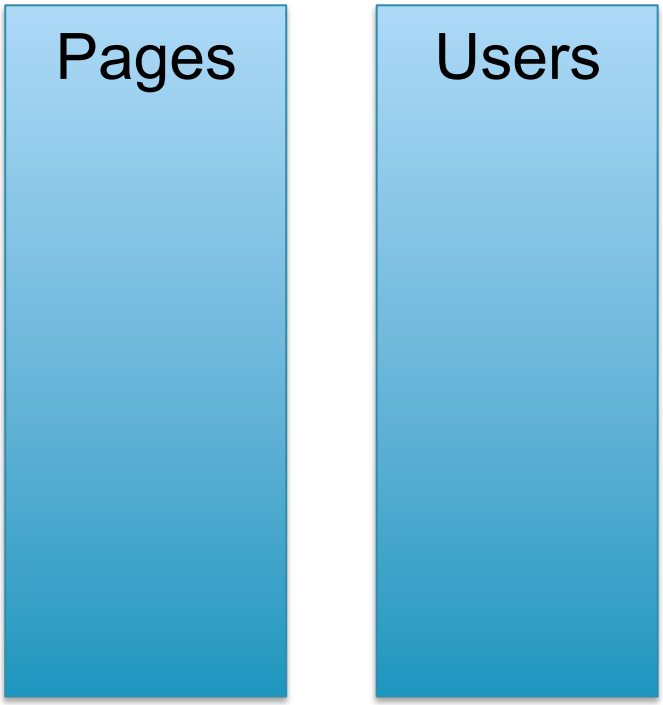
Joins in MapReduce

- If MR is GROUP-BY plus AGGREGATE, then how do we compute R(A,B) ⋈ S(B,C) using MR?

- Answer:
  - Map: group R by R.B, group S by S.B
    - Input = either a tuple R(a,b) or a tuple S(b,c)
    - Output = (b,R(a,b)) or (b,S(b,c)) respectively
  - Reduce:
    - Input = (b,{R(a1,b),R(a2,b),…,S(b,c1),S(b,c2),…})
    - Output = {R(a1,b),R(a2,b),…} × {S(b,c1),S(b,c2),…}
    - In practice: improve the reduce function (next…)

Hash Join in PigLatin

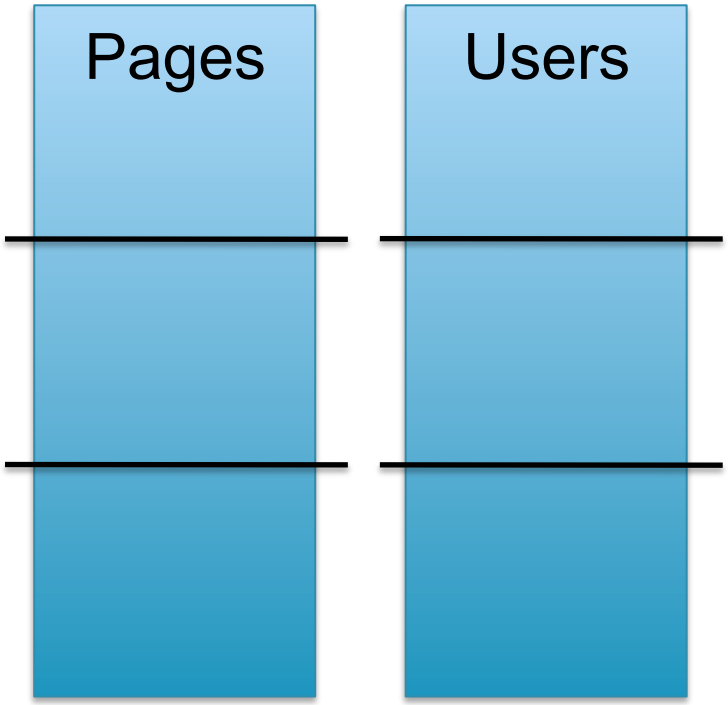
```
Users = load 'users' as (name, age);
Pages = load 'pages' as (user, url);
Jnd = join Users by name, Pages by user;
```



Credit: Alan Gates, Yahoo!

Hash Join in PigLatin

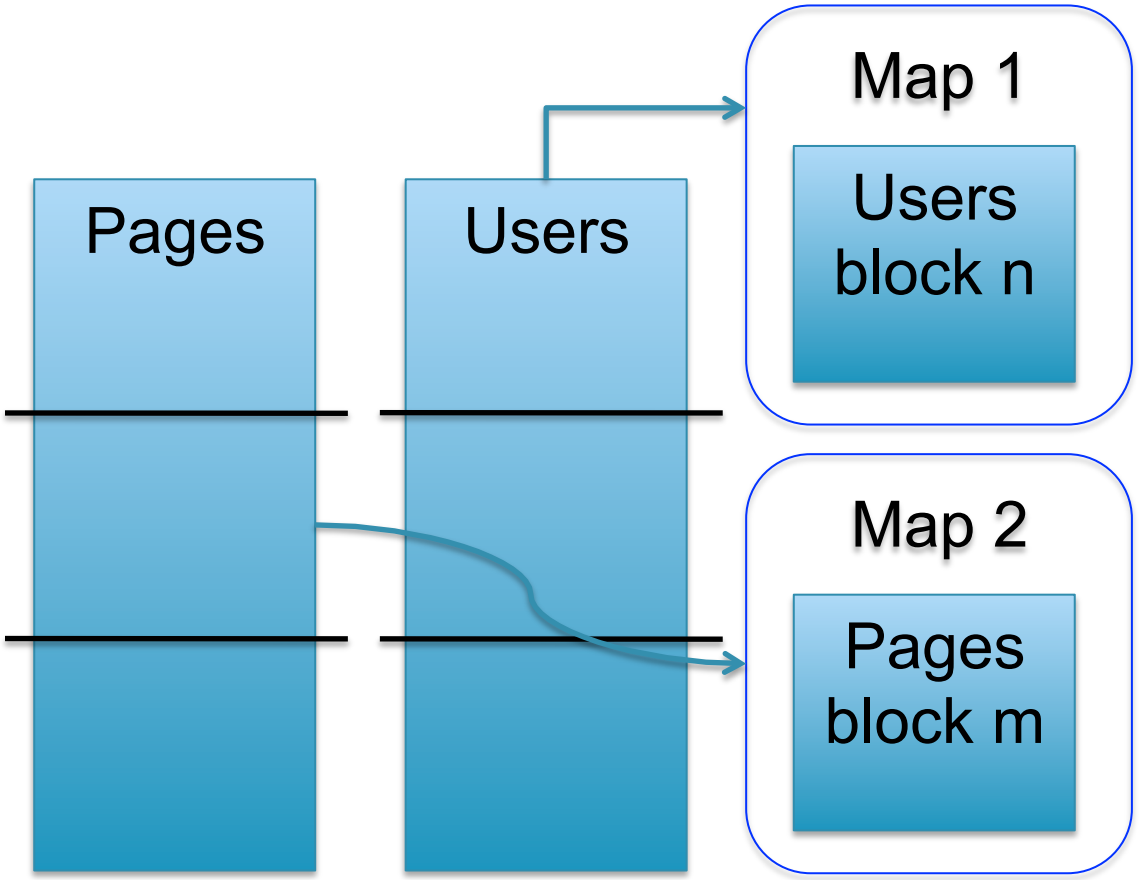
```
Users = load 'users' as (name, age);
Pages = load 'pages' as (user, url);
Jnd = join Users by name, Pages by user;
```



Credit: Alan Gates, Yahoo!

Hash Join in PigLatin

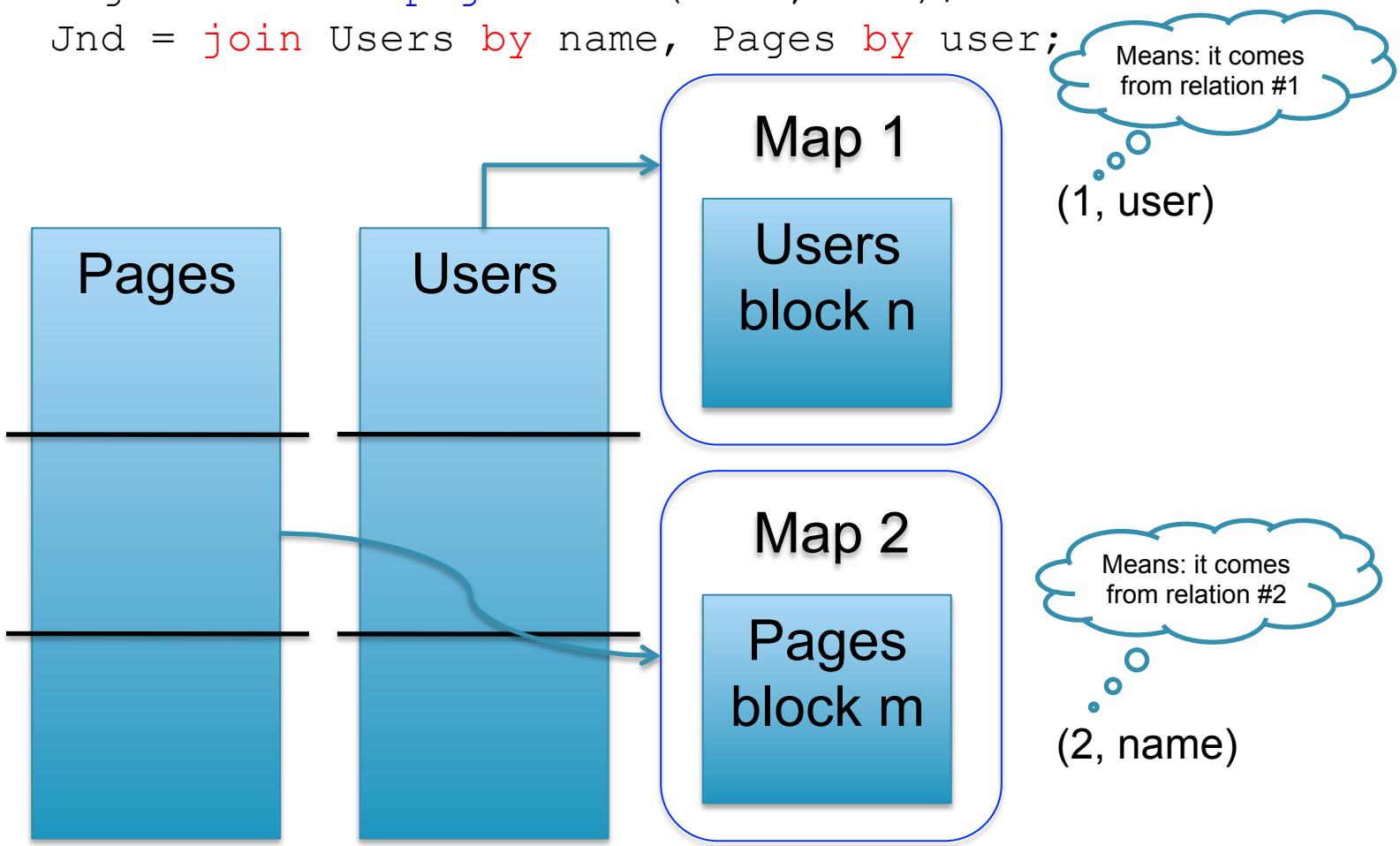
```
Users = load 'users' as (name, age);
Pages = load 'pages' as (user, url);
Jnd = join Users by name, Pages by user;
```



Credit: Alan Gates, Yahoo!

Hash Join in PigLatin

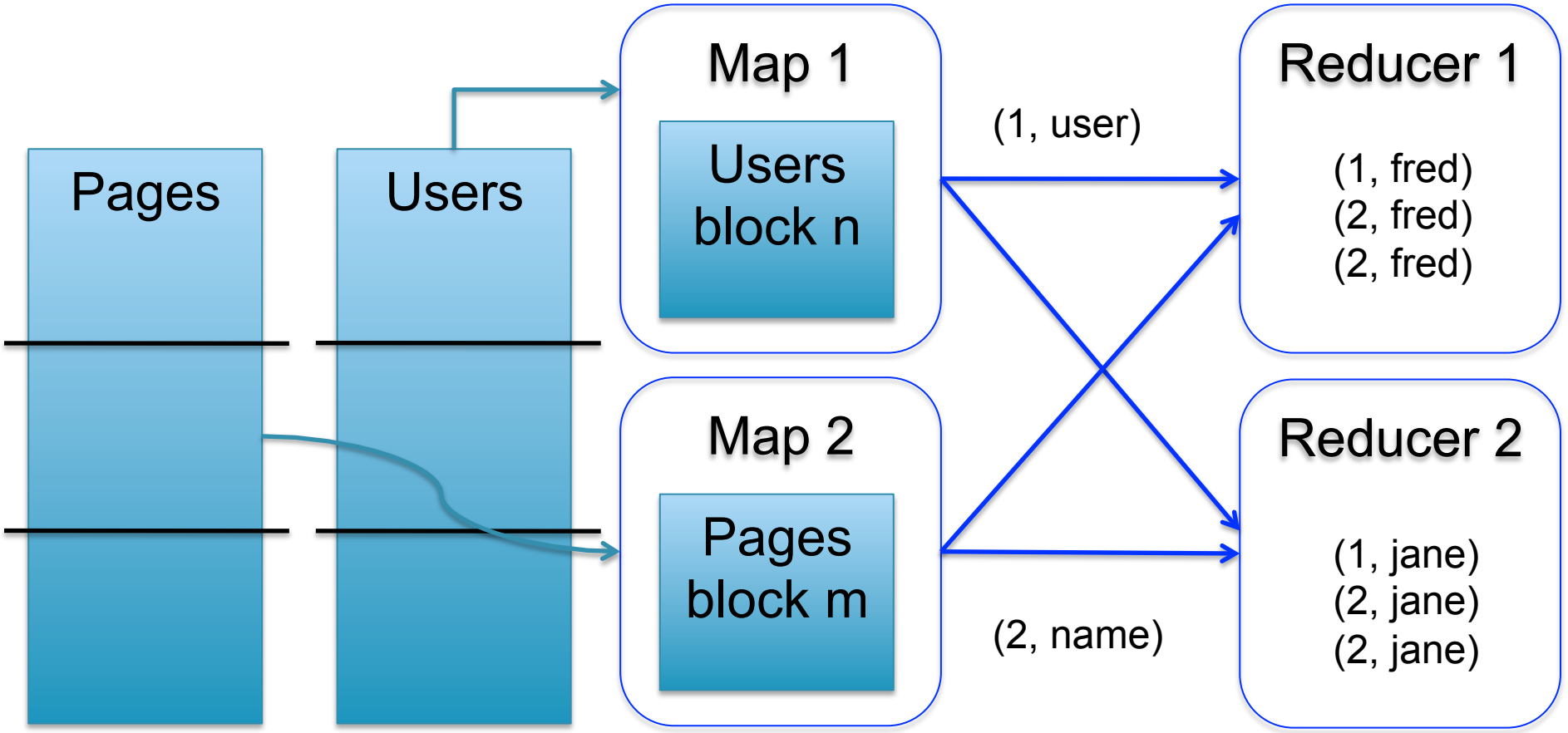
```
Users = load 'users' as (name, age);
Pages = load 'pages' as (user, url);
Jnd = join Users by name, Pages by user;
```



Credit: Alan Gates, Yahoo!

Hash Join in PigLatin

```
Users = load 'users' as (name, age);
Pages = load 'pages' as (user, url);
Jnd = join Users by name, Pages by user;
```



Credit: Alan Gates, Yahoo!

Hash Join in PigLatin

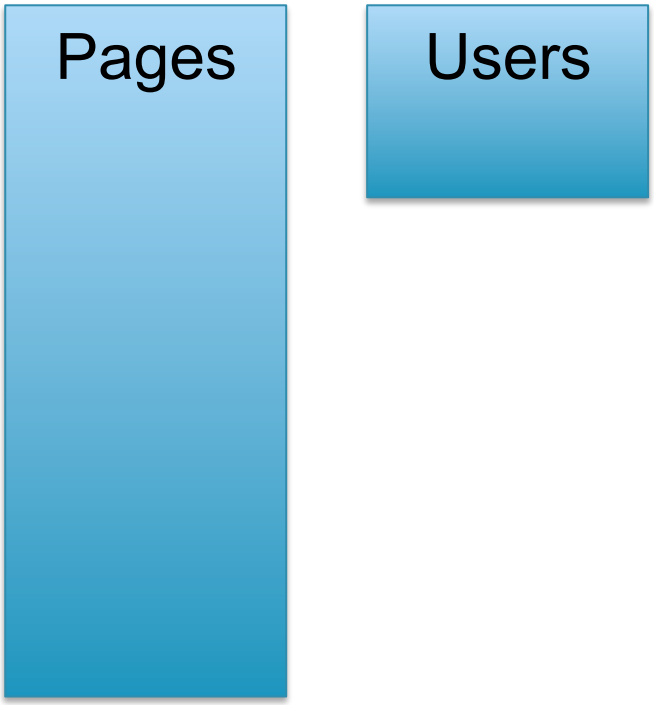
```
Users = load 'users' as (name, age);
Pages = load 'pages' as (user, url);
Jnd = join Users by name, Pages by user;
```

```
map(String usr, String value):
    // usr: either Users.name or Pages.user
    // value.relation is either 'Users' or 'Pages'
    if value.relation='Users':
     EmitIntermediate(usr, (1, value));
    else
     EmitIntermediate(usr, (2, value));
```

```
reduce(String usr, Iterator values):
    Users = empty;  Pages = empty;
    for each v in values:
      if v.type = 1: Users.insert(v)
      else Pages.insert(v);
    for v1 in Users, for v2 in Pages
      Emit(usr, v1,v2);
```

Broadcast Join in PigLatin

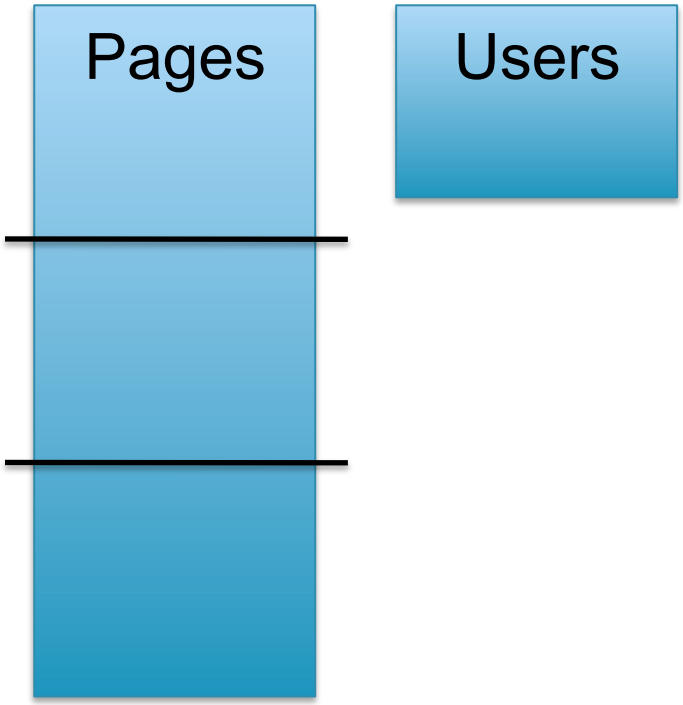
```
Users = load 'users' as (name, age);
Pages = load 'pages' as (user, url);
Jnd = join Pages by user, Users by name using "replicated";
```



Credit: Alan Gates, Yahoo!

Broadcast Join in PigLatin

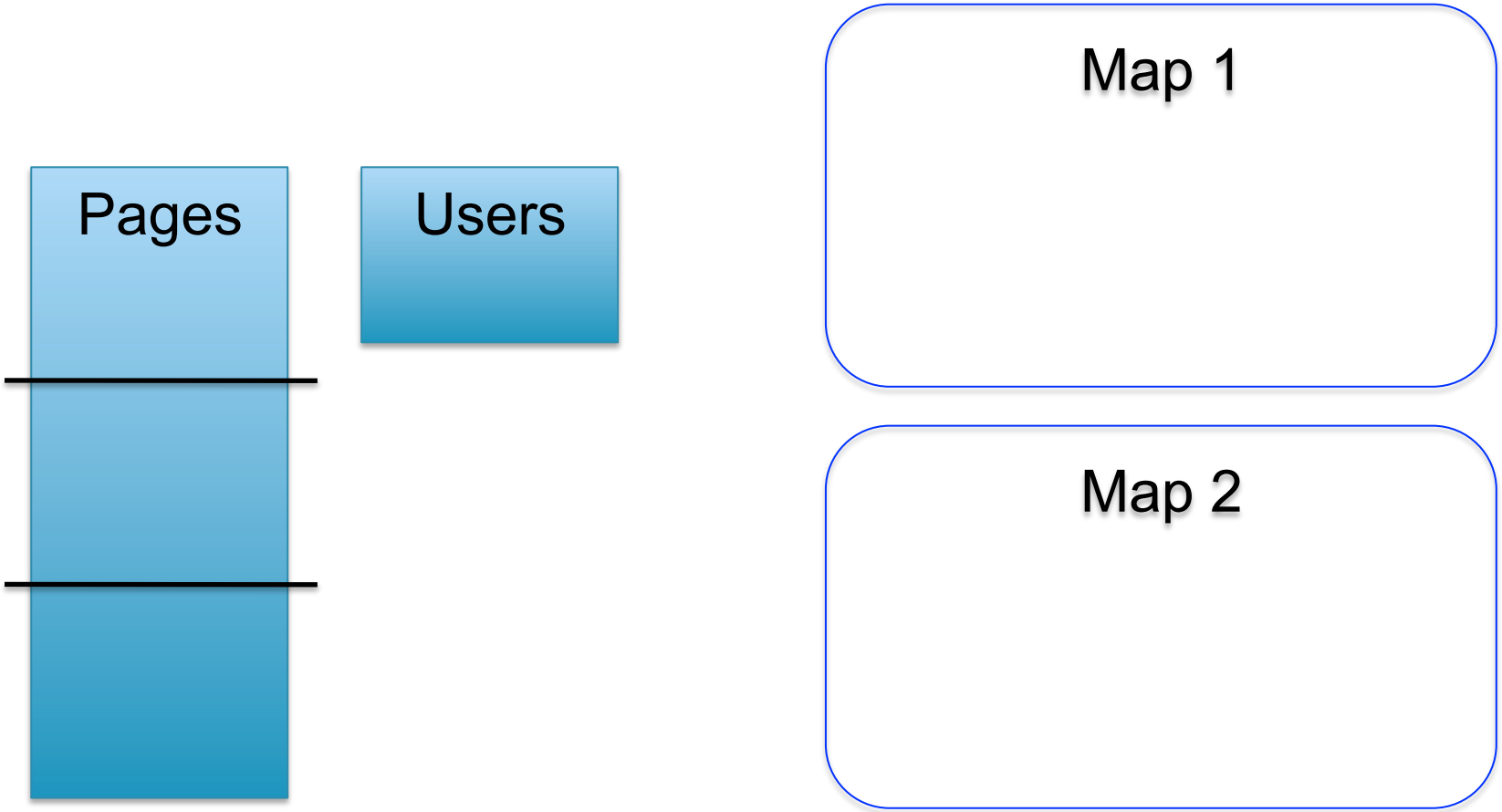
```
Users = load 'users' as (name, age);
Pages = load 'pages' as (user, url);
Jnd = join Pages by user, Users by name using "replicated";
```



Credit: Alan Gates, Yahoo!

Broadcast Join in PigLatin

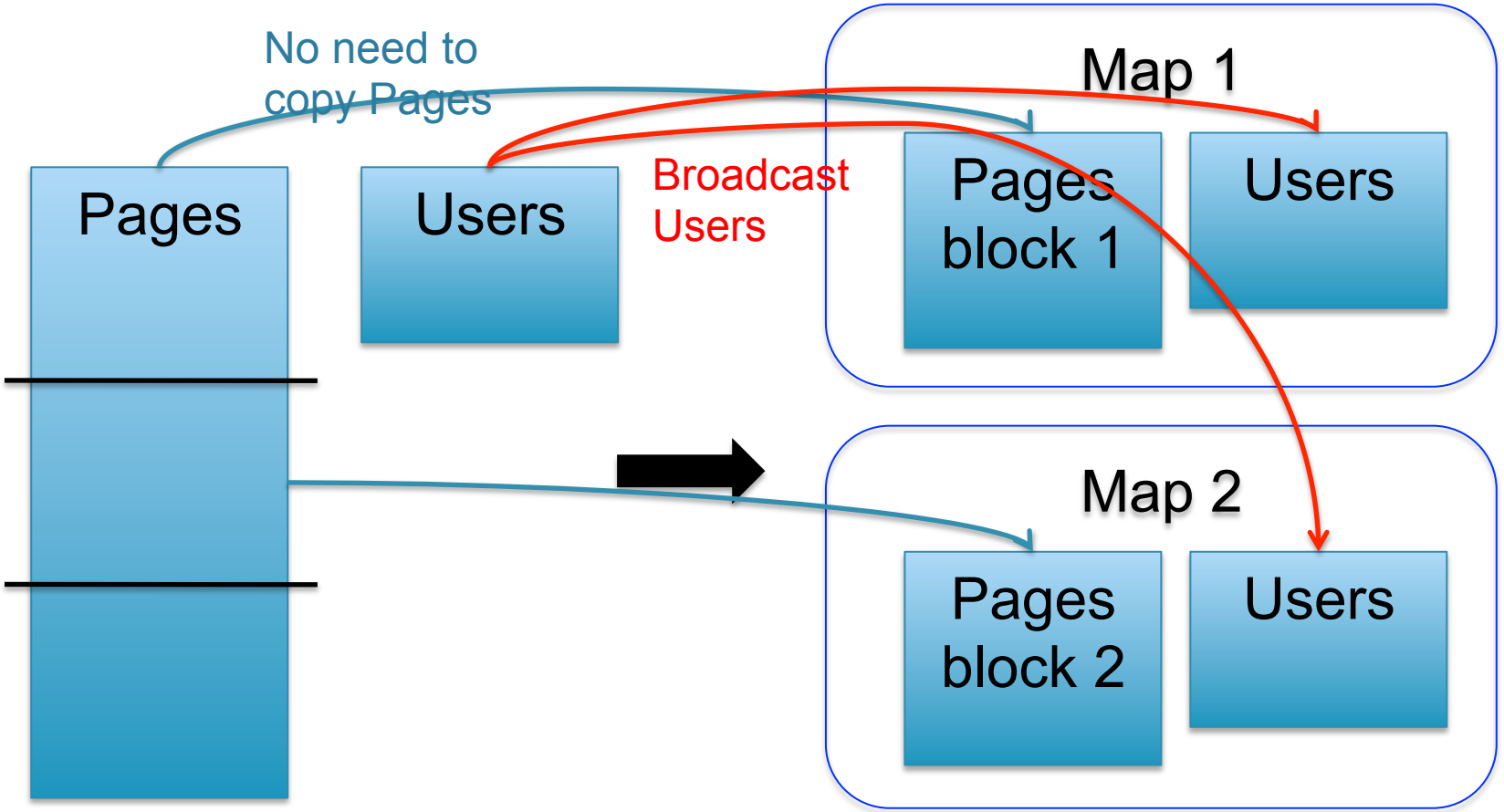
```
Users = load 'users' as (name, age);
Pages = load 'pages' as (user, url);
Jnd = join Pages by user, Users by name using "replicated";
```



Credit: Alan Gates, Yahoo!

Broadcast Join in PigLatin

```
Users = load 'users' as (name, age);
Pages = load 'pages' as (user, url);
Jnd = join Pages by user, Users by name using "replicated";
```



Credit: Alan Gates, Yahoo!

Parallel DBs v.s. Map-Reduce

Parallel DB

- Plusses

- Minuses

Map-Reduce

- Minuses

- Plusses

Parallel DBs v.s. Map-Reduce

Parallel DB

- Plusses
  - Efficient binary format
  - Indexes, physical tuning
  - Cost-based optimization
- Minuses
  - Difficult to import data
  - Lots of baggage: logging, transactions

Map-Reduce

- Minuses
  - Lots of time spent parsing!
  - Text files
  - “Optimizer is between your eyes and your keyboard”
- Plusses
  - Any data
  - Lightweight, easy to speedup