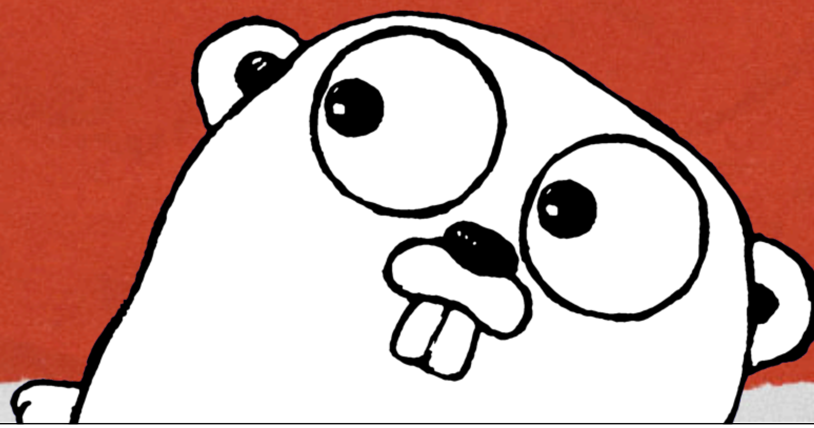
MAP-REDUCE

Curt Clifton
Rose-Hulman Institute of Technology



Schedule Overview

9	36 Fri Nov 5	• Map-reduce	MapReduce	• MapReduce [DeanGhemawat04]		Teaching Material Revisions
10	37 Mon Nov 8	• 4th Period: May the Forth Be With You • 4th Period: Jr. Raptor Wranglers • — • 5th Period: Team Amethyst • 5th Period: Honest Jim's Miracle Tonic		• Install software	• HW 15 Pair Programming Encouraged	
10	38 Tue Nov 9	• 4th Period: Team Bruce • 4th Period: Steak Jell • — • 5th Period: defn team-name (fn [] "Team Lambda")) • 5th Period: Black Perl		• Install software		
10	39 Thu Nov 11	• 4th Period: Discussion on Language Design • 4th Period: Course Evaluations • — • 5th Period: Deck the Halls – Scala la la la • 5th Period: Adjective Animal Productions		• Read this history of programming languages • Install software	• HW 16	
10	40 Fri Nov 12	• 5th Period: Discussion on Language Design • 5th Period: Course Evaluations • **Complete Team Performance Evaluations on ANGEL before 8am on Monday**	Language Design		• **Team Evals before 8am on Monday**	Final Presentation, Code Review, and Rubric

Two more HW

MORE CONCURRENCY IDIOMS IN GO

PARALLEL MERGE SORT

See GoConcurrency/parsort.go

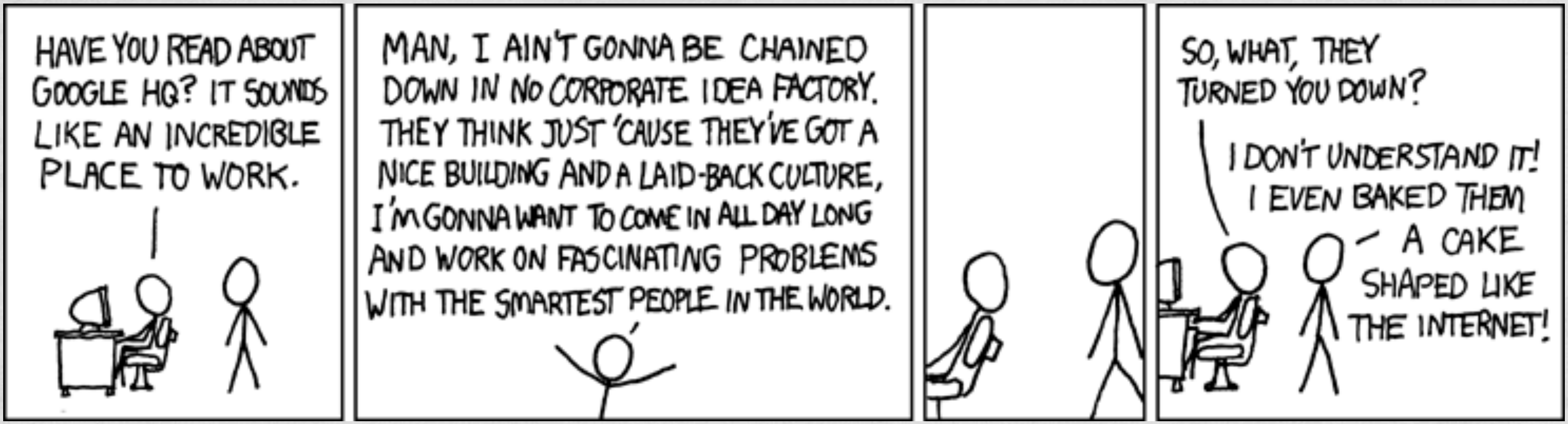
IDIOM: TIMEOUT

```go
func performWithTimeout(ev [] *script.Event, t *testing.T) {
    result := make(chan os.Error)
    timesUp := make(chan bool)

    go func() {
        result <- script.Perform(0, ev)
    }()

    go func() {
        time.Sleep(timeout)
        timesUp <- true
    }()

    select {
        case err := <- result:
            if err != nil {
                t.Errorf("Got error: %s", err)
            }
        case <- timesUp:
            t.Errorf("failed to receive expected events before timeout")
    }
}
```

Q1

GOOGLE'S MAP-REDUCE

- Described by Jeffrey Dean and Sanjay Ghemawat [OSDI 2004]
- Relies on the Google File System for storing massive data sets across thousands of commodity drives
- Open source version implemented by Yahoo!, et al



http://xkcd.com/192/

I hear once you've worked there for 256 days they teach you the secret of levitation.

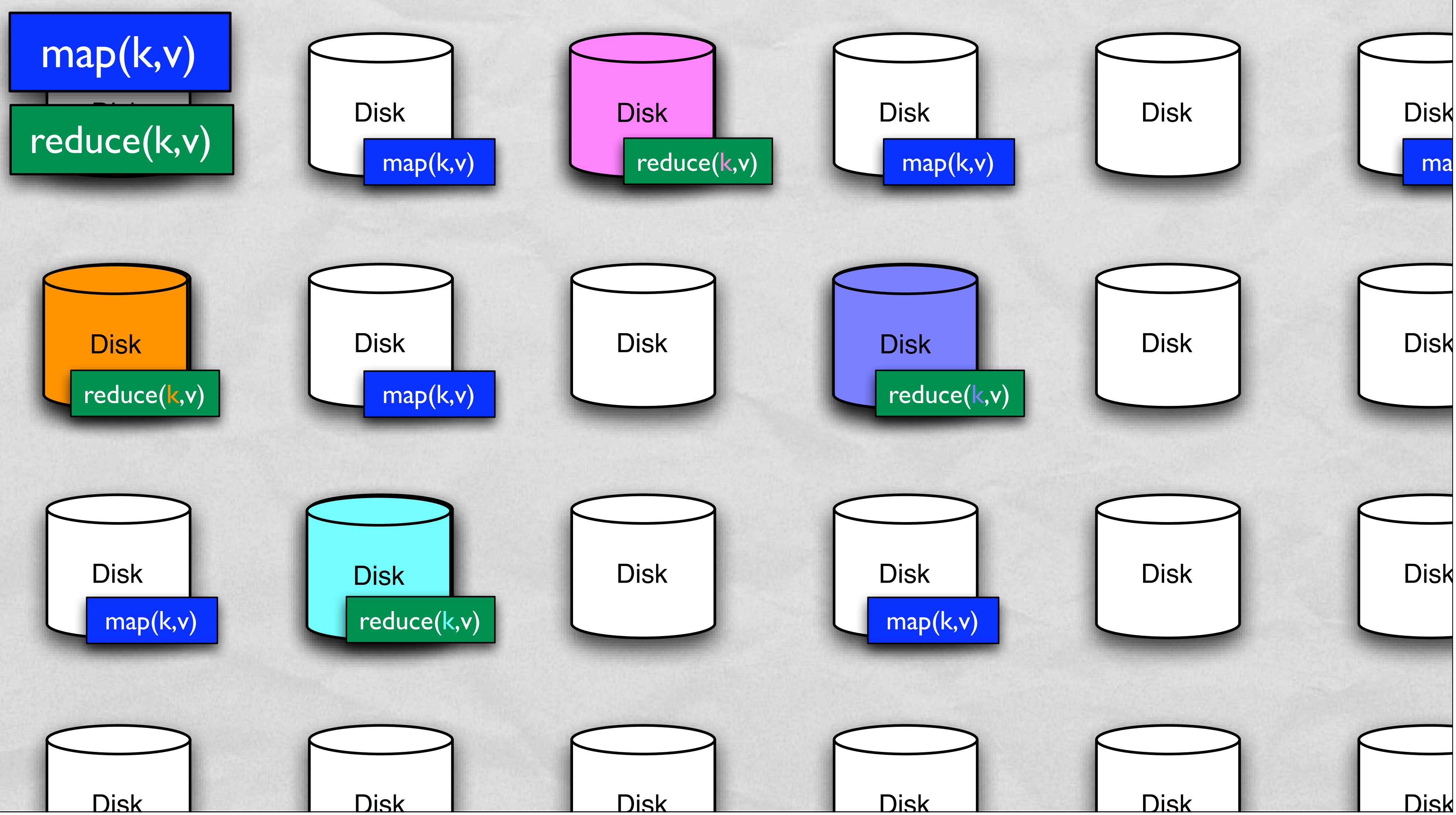
FUNCTIONS FTW

- Algorithms implemented by a pair of functions
  - *map*: processes a key/value pair, generates a set of new key/value pairs
  - *reduce*: gets a single key and a set of all associated values, processes the set into a single result for the key
- Automatically parallelized and distributed!

EXAMPLE: INDEXING

- map:
  - takes a (URL, textual contents) pair
  - emits a list of (word, URL) pairs
- reduce:
  - takes every URL for a given word
  - produces a (word, [URL]) pair

GOOGLE FILE SYSTEM



TYPES

- `map ::`
  ```
  (Key k1, Key k2, Value v1, Value v2)
     => k1 -> v1 -> [(k2, v2)]
  ```

- `reduce ::`
  ```
  (Key k2, Value v2, Value v3)
     => k2 -> [v2] -> v3
  ```

OTHER EXAMPLES

- Inverted Index
- Distributed Grep
- Count of URL Access Frequency
- Reverse Web-Link Graph

Q2

PAGE RANK: RANDOM WALK OF THE WEB

- Suppose user starts at a random page
- Surfs by either:
  - Clicking some link from the page at random, or
  - Entering a new random URL
- What is the probability that she arrives at a given page?

THE FORMULA

- Given a page *A*, and pages T_1–T_n that link **to** *A*, page rank of *A* is:

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_n)}{C(T_n)}\right)$$

- where:
  - $C(T_i)$ is the number of edges leaving page T_i
  - *d* represents the likelihood of a user clicking (rather than randomly entering a new URL)

Q3

PAGE RANK USING MAP-REDUCE

Multiple Passes!

- Phase 1:
  - map:: URL -> pageText -> [(URL, (1, [targetURL]))]

    PR$_{init}$
  - reduce is just identity function

PAGE RANK USING MAP-REDUCE

Repeat Phase 2 until it converges!

$$PR(A) = (1-d) + d\left(\frac{PR(T_1)}{C(T_1)} + \ldots + \frac{PR(T_n)}{C(T_n)}\right)$$

currentRank / len([targetURL])

- Phase 2:
  - map :: URL -> (currentRank, [targetURL]) -> (URL, [targetURL]) : [(targetURL, partialRank)]
  - reduce :: targetURL -> ([targetsTargets]) : [partialRank] -> (targetURL, (newRank, [targetsTargets]))

(1-d) + dΣ[partialRank]

map-reduce isn't statically typed!

DEMO
TIME PERMITTING

SANTA SIMULATOR

- Due Monday
  Can pair program this one

ACKNOWLEDGEMENTS

- Slides contain material © 2008 Google, Inc. and © Spinaker Labs, Inc., distributed under the Creative Commons Attribution 2.5 license.
- Original materials from the 2008 NSF Data-Intensive Scalable Computing in Education Workshop, Seattle, WA.