MAP-REDUCE

Curt Clifton
Rose-Hulman Institute of Technology

SVN Update *ErlangInClass*

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

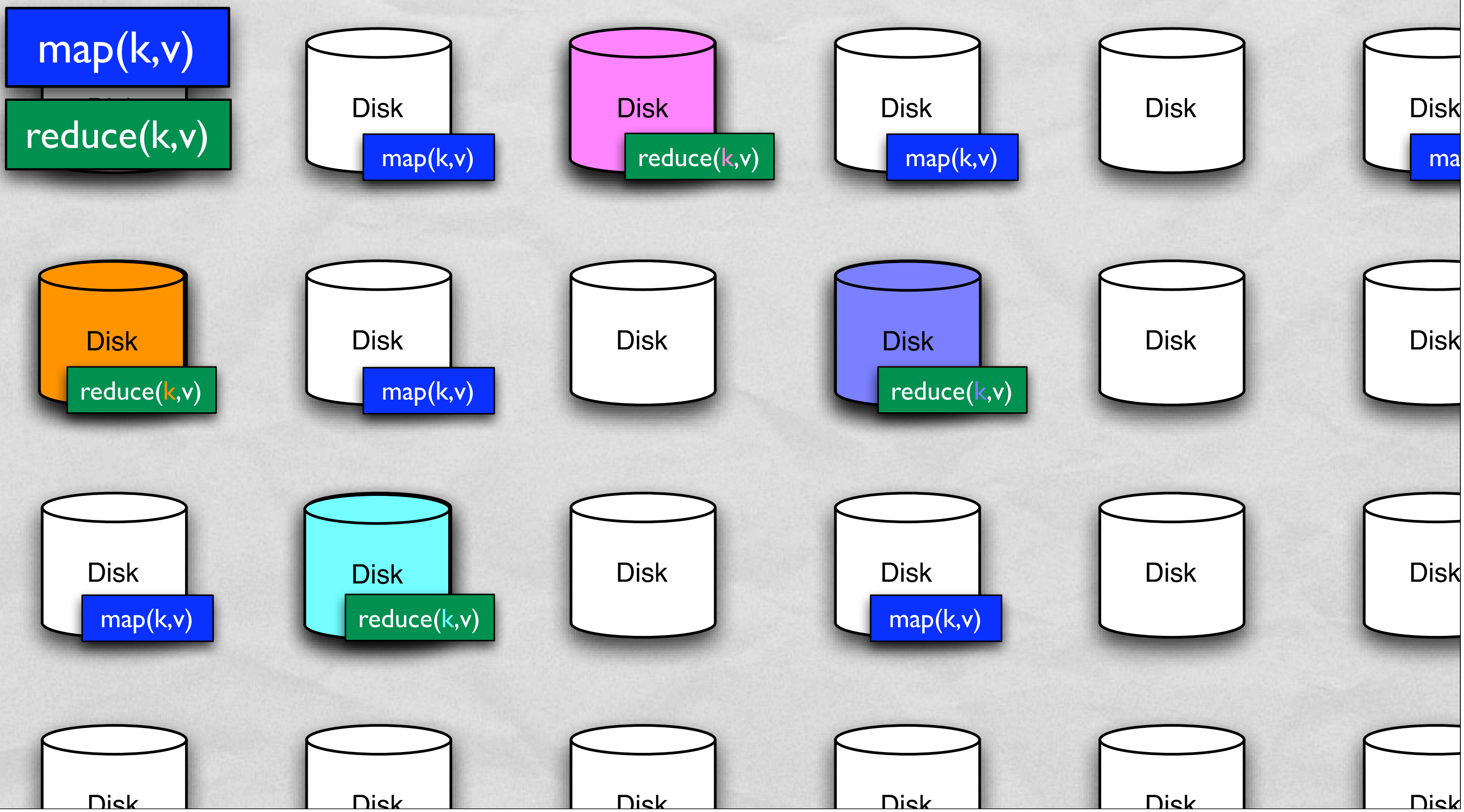
FUNCTIONS FTW

- 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

Q1

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)

Q2

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!

currentRank / len([targetURL])

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

Σ[partialRank]

map-reduce isn't statically typed!

FAULT TOLERANCE

- Google file system stores data in triplicate!

HADOOP

- Yahoo’s open source implementation of
  - Google File System
  - Map-Reduce
    - Includes several interfaces: Java, pipes (including bash, perl, and Python), and Pig

DEMO

PAC-MAN

DUE NEXT THURSDAY

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.