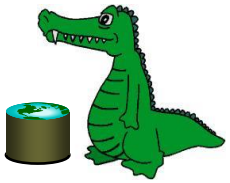
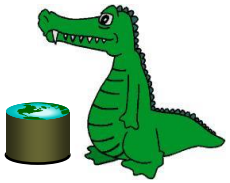


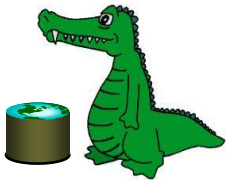
Logistics

- Lab 2 grades released
- Lab 3 published – quite hard
  - START NOW
  - Work with your partner
  - Multiple M/R jobs needs to be writen
  - Due coming Friday
- Lab 3 group finalize today
- Pop quiz 4



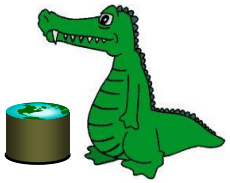
Review

- Large-scale data storage and processing
- Distributed File System
- Map-Reduce API
  - InputFormat
  - Map function
  - [Combiner]
  - Sorting & Shuffling
  - Reduce function
  - OutputFormat



Exercise 1: Host size

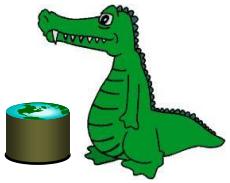
- Suppose we have a large web corpus
- Let's look at the metadata file
  - Lines of the form (URL, size, date, …)
- For each host, find the total number of bytes
  - i.e., the sum of the page sizes for all URLs from that host



Example 1: Host size (cont.)

- InputFormats: transform on-disk data representation on HDFS to in-memory key-value
  - Example: TextInputFormat, KeyValueTextInputFormat

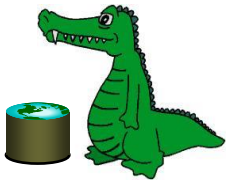
- Mapper: (position, "URL, size, data,…") -> (hostname, size)
- Mapper: (URL, "size, data,…") -> (hostname, size)
- Reducer: (hostname, list (size)) -> (hostname, totalsize)



Exercise 2: Distributed Grep

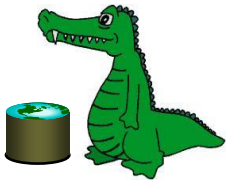
- Find all occurrences of the given pattern in a very large set of webpages

- InputFormat
  - webpages → (url+offset, single line)



Exercise 2: Distributed Grep (cont.)

- Input consists of (url+offset, single line)
- map(key=url+offset, val=line):
  - If contents matches regexp, emit (line, "1")
- reduce(key=line, values=uniq_counts):
  - Don't do anything; just emit line



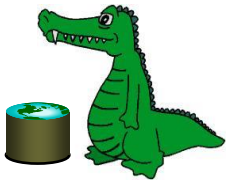
Exercise 3: Graph reversal

- Given a directed graph as an adjacency list:

src1: dest11, dest12, …

src2: dest21, dest22, …

- Construct the graph in which all the links are reversed



Exercise 3: Graph reversal (cont.)

- KayValueTextInputFormat
- Map
  - For each URL linking to target, …
  - Output <target, source> pairs
- Reduce
  - Concatenate list of all source URLs
  - Outputs: <target, ***list*** (source)> pairs