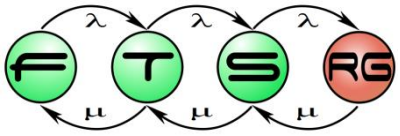
RHadoop (rmr2)

„Big Data” elemzési módszerek

Kocsis Imre
ikocsis@mit.bme.hu
2015. 10. 07.



Egy/A Big Data probléma

- „At rest Big Data”
  - Nincs update
  - „Mindent” elemzünk
- Elosztott tárolás
- „Computation to data”

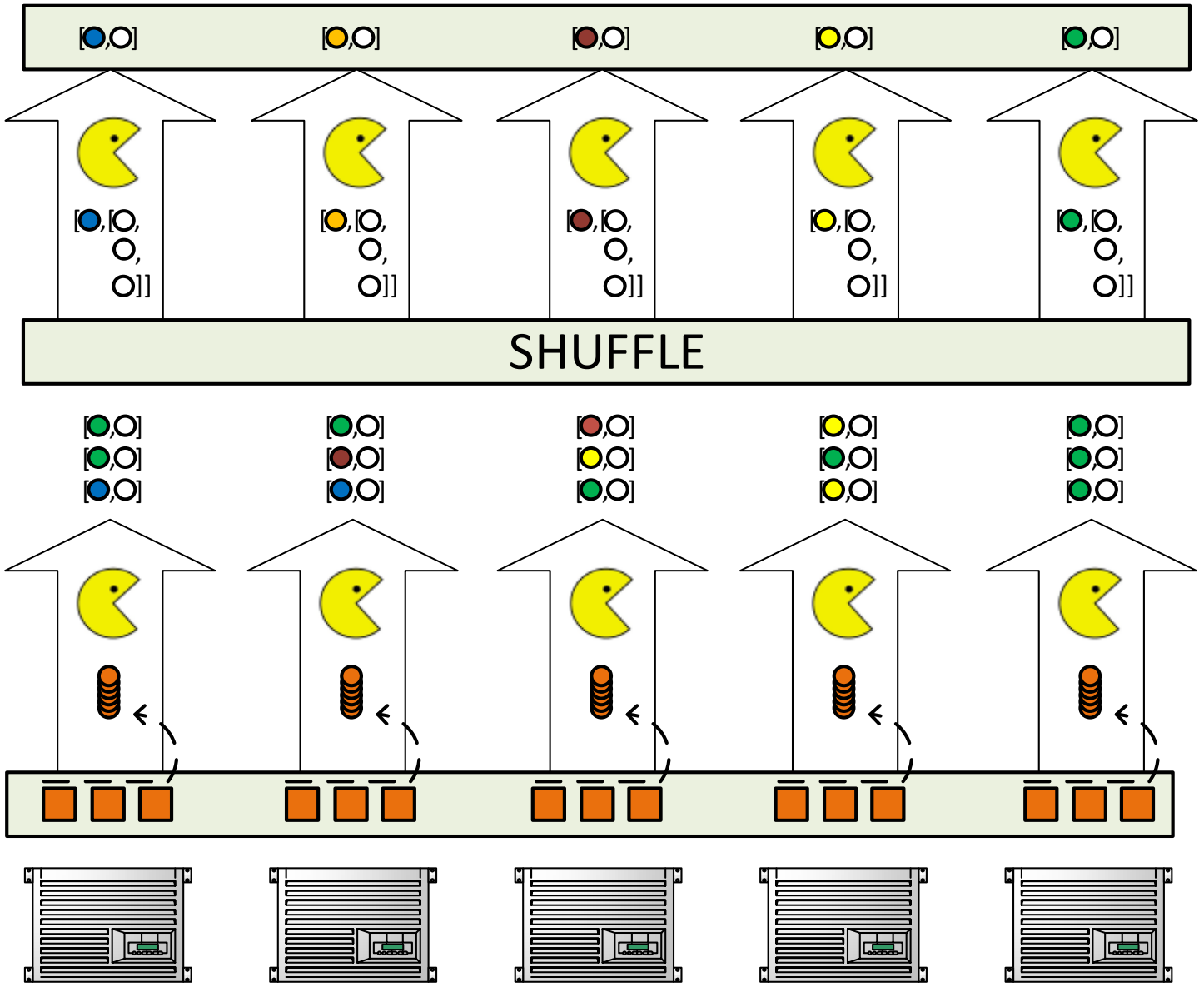
„Not true, but a very, very good lie!”
(T. Pratchett, Nightwatch)

MapReduce

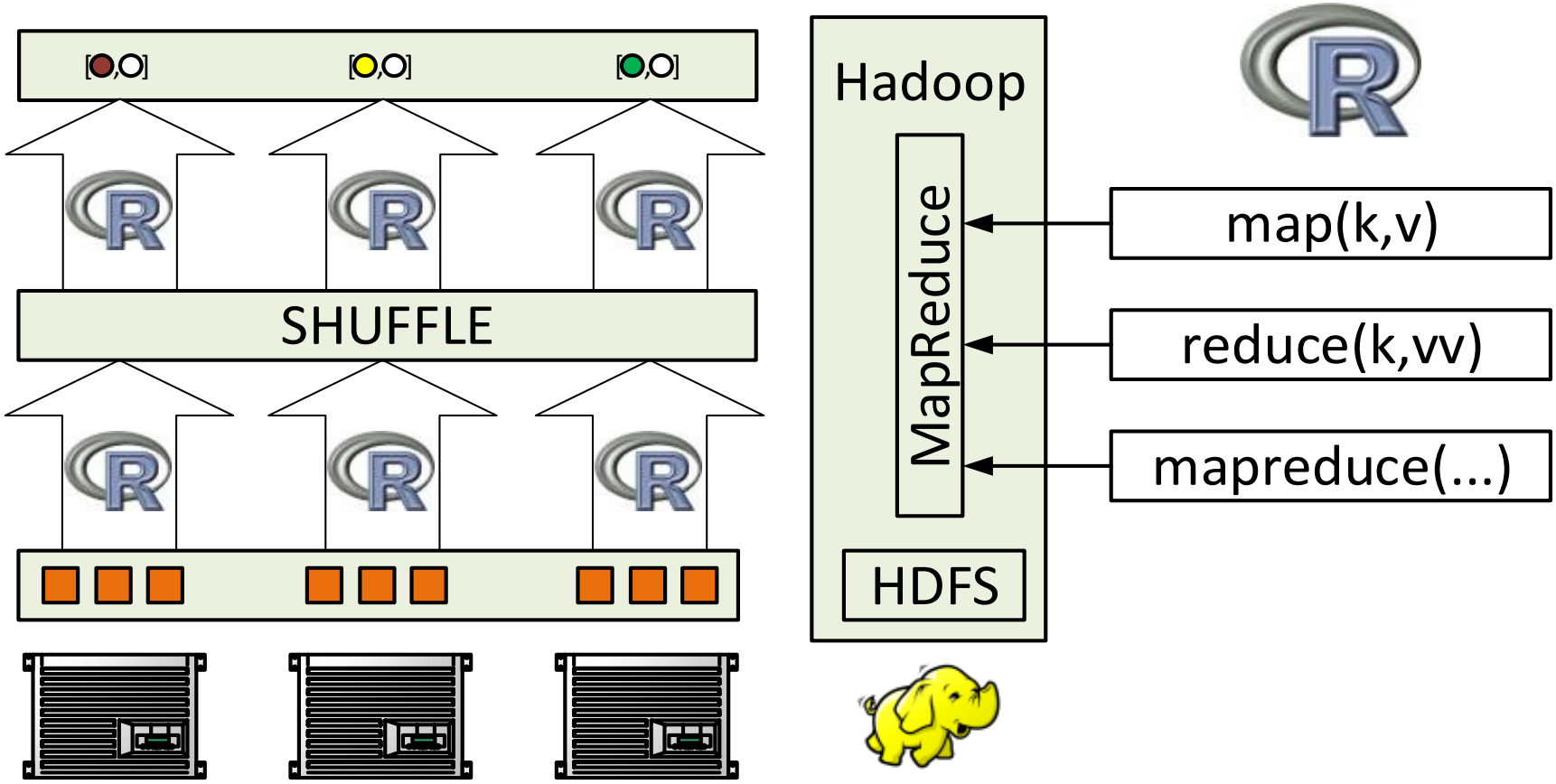
„Reduce”

„Map”

Distributed File System



RHadoop = Hadoop + R



RHadoop

- github.com/RevolutionAnalytics/RHadoop/

- *„The most mature [...] project for R and Hadoop is RHadoop.” (O’Reilly, R In a Nutshell, 2012)*

- `rmr(2)`: mapreduce
- `rhdfs`: HDFS állománykezelés
- `rhbase, plyrmr`

Local backend

```
rmr.options(backend="local")
```

- Helyi állományrendszer
- Szekvenciális végrehajtás
- Debug!

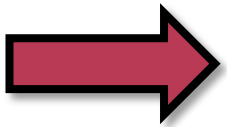
- Input/output itt is állományrendszer

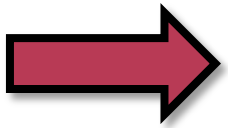
Szószámlálás

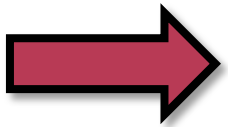
```
map(String input_key, String input_value):
  // input_key: document name
  // input_value: document contents
  for each word w in input_value:
    EmitIntermediate(w, "1");
```

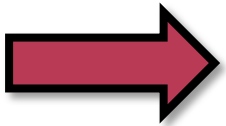
```
reduce(String output_key, Iterator intermediate_values):
  // output_key: a word
  // output_values: a list of counts
  int result = 0;
  for each v in intermediate_values:
    result += ParseInt(v);
  Emit(AsString(result));
```

rmr: mapreduce









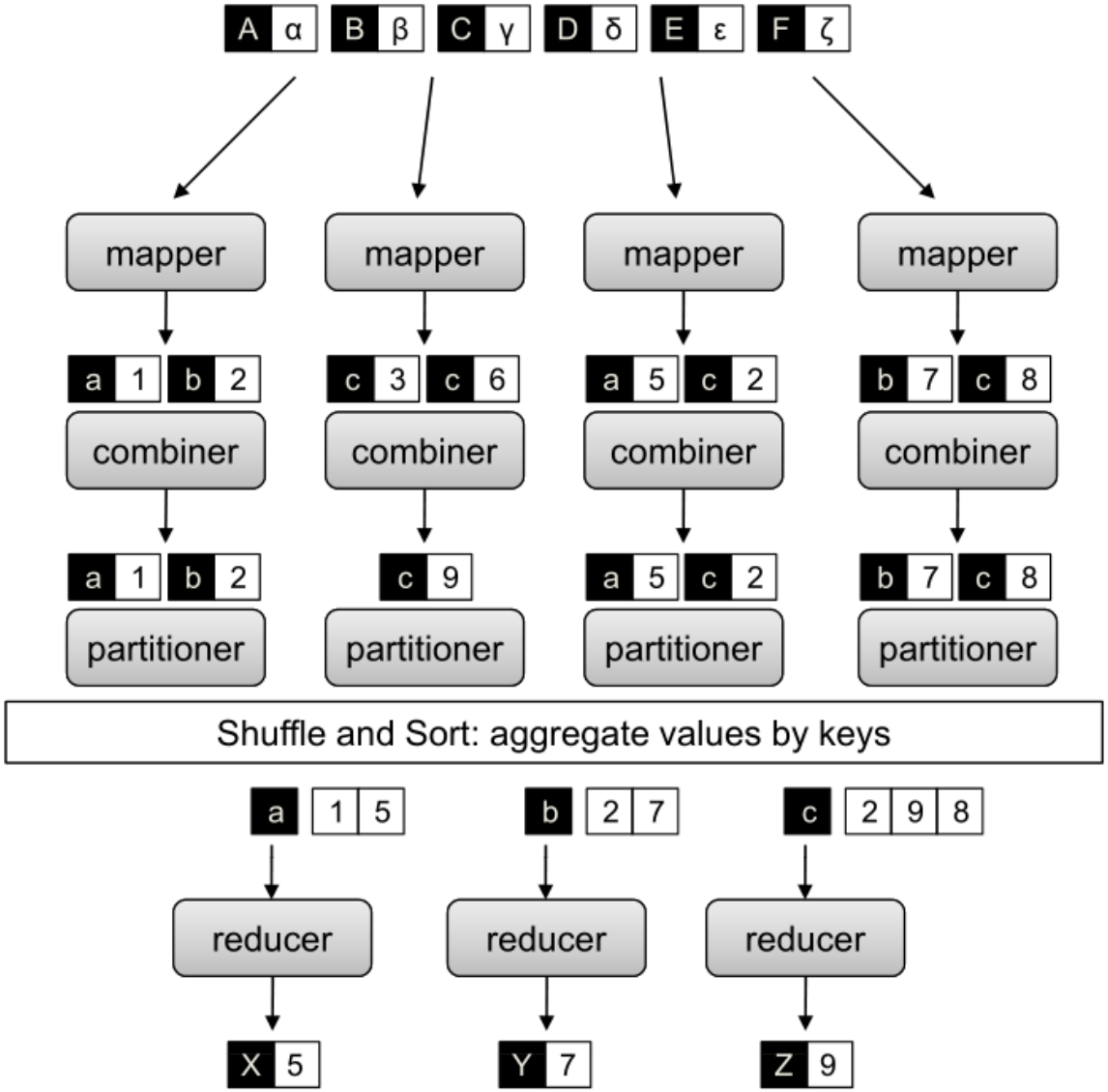
```
wordcount =
  function(
    input,
    output = NULL,
    pattern = " "){

    wc.map =
      function(., lines) {
        keyval(
          unlist(
            strsplit(
              x = lines,
              split = pattern)),
          1)}

    wc.reduce =
      function(word, counts ) {
        keyval(word, sum(counts))}

    mapreduce(
      input = input ,
      output = output,
      input.format = "text",
      map = wc.map,
      reduce = wc.reduce,
      combine = T)}
```

MapReduce: a teljes kép



Forrás: [1], p 30

Input/output format

- text
- json
- csv
- native (R sorosítás)
- sequence.typedbytes (Hadoop)
- pig.hive
- hbase

Előnyök

- Map és Reduce: R-ben
  - Csomagok!
  - MR algoritmus-prototipizálás
- + a vezérlés is: kényelem

- Hadoop Job: egy függvényhívás!
  - Pl. iteratív MapReduce teljesen R-ben
  - Map és Reduce: ~a hívó környezetben

Hogyan lehet ilyenem?

- Local backend, sandbox VM-ek
  - Cloudera, Hortonworks
- Saját Hadoop klaszter ☺
- Amazon Elastic MapReduce (EMR)
  - Bérelhető Hadoop klaszter
  - Erősen javasolt kipróbálni
- Saját felhő megoldás

Hátrányok?

- Nehézkes debug
- +1 hangolási réteg
- ~~MAHOUT-klón~~
- ~~Sok Hadoop funkc.~~
- Kevés példa

```
14/01/10 15:34:57 INFO streaming.StreamJob:  map 100%  reduce 0%
14/01/10 15:35:04 INFO streaming.StreamJob:  map 100%  reduce 100%
14/01/10 15:35:04 INFO streaming.StreamJob: To kill this job, run:
14/01/10 15:35:04 INFO streaming.StreamJob: UNDEF/bin/hadoop job  -Dmapred.job.tra
cker=hdfs://10.6.21.150:54311 -kill job_201401101345_0002
14/01/10 15:35:04 INFO streaming.StreamJob: Tracking URL: http://vm-large-10.vcl.i
ntra:50030/jobdetails.jsp?jobid=job_201401101345_0002
14/01/10 15:35:04 ERROR streaming.StreamJob: Job not successful. Error: NA
14/01/10 15:35:04 INFO streaming.StreamJob: killJob...
Streaming Command Failed!
Error in mr(map = map, reduce = reduce, combine = combine, vectorized.reduce,  :
  hadoop streaming failed with error code 1
>
```