MapReduce: Exercises

Exercise (See ex. 2.3.1.(b) of [LRU14])

Let S be a set of N integers represented by pairs (i, x_i), with $0 \leq i < N$, where i is the key of the pair and x_i is an arbitrary integer.

1. Design a 2-round MR algorithm to compute the arithmetic mean of the integers in S, using $O\left(\sqrt{N}\right)$ local space and $O(N)$ aggregate space.
2. Show how to modify the algorithm of the previous point to reduce the local space bound to $O\left(N^{1/4}\right)$, at the expense of an increased number of rounds.
3. Can you attain $M_L = O(M)$, for any $M < \sqrt{N}$?

input: pairs (i, x_i), $i \in [0, N)$ key, $x_i \in \mathbb{R}$

output: $(0, \frac{1}{N} \sum_{i=0}^{N-1} x_i)$

① 2-round solution, $M_L = O(\sqrt{N})$, $M_A = O(N)$

Round 1:

- Map: $\forall$ input pair $(i, x_i) \rightarrow (i \bmod \sqrt{N}, x_i)$ — l (parameter)
- Reduce: $\forall$ key $j \in [0, \sqrt{N})$, let L_j be the set of values with key j $\Rightarrow$ emit pair $(0, \sum_{x \in L_j} x)$ — odd $|L_j|$ as value

Round 2:

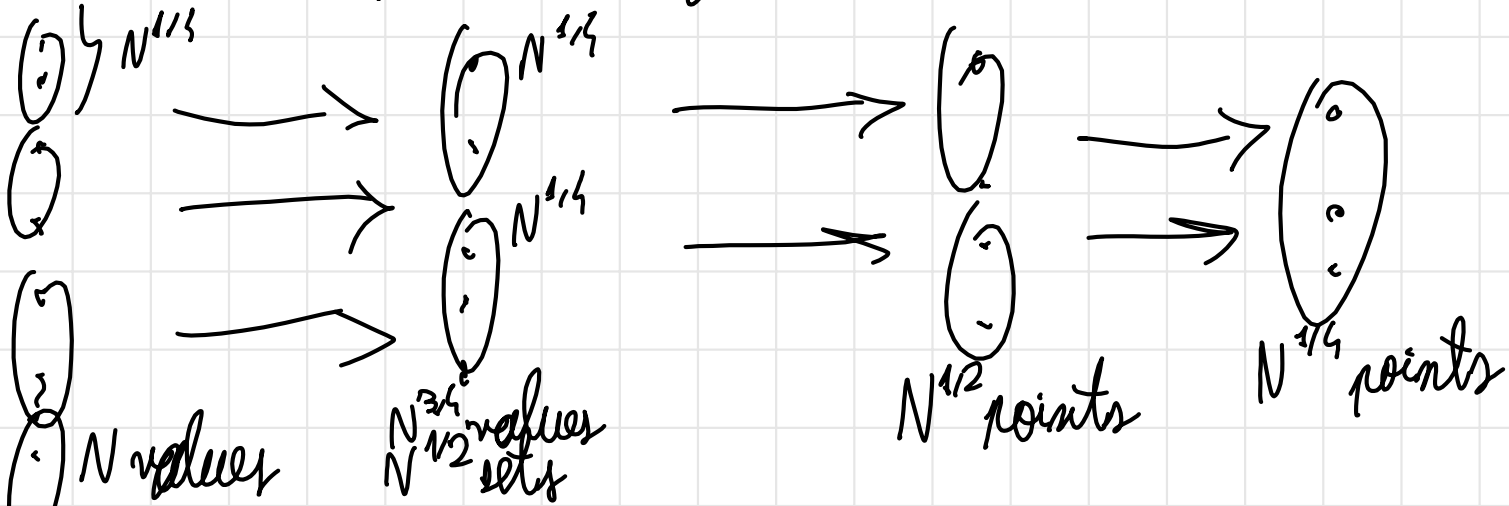
- Map: empty
- Reduce: let L_0 be the set of values with key 0 $\Rightarrow$ $\Rightarrow$ emit $(0, \frac{1}{N} \sum_{y \in L_0} y)$

we assume N is known ⇒ possible in Spark
we can modify the algorithm for the case we don't know N

$M_L = \max\{O(1), O(\sqrt{N}), O(1), O(\sqrt{N})\} = O(\sqrt{N})$
$M_A = \max\{O(N), O(N), O(\sqrt{N}), O(\sqrt{N})\} = O(N)$

② goal: $M_L = O(\sqrt[4]{N})$, $M_A = O(N)$

we could use $\ell = N^{3/4}$ ⇒ $N^{1/4}$ el. for partition ⇒
⇒ $N^{3/4}$ partial sums
on top of this, repeat partitioning



ROUND 1:
- Map: $(i, x_i) \rightarrow (i \bmod N^{3/4}, x_i)$
- Reduce: $\forall$ key $j \in [0, N^{3/4})$, L_j: list of values with key $j \Rightarrow$
  $\Rightarrow$ note $|L_j| = N^{1/4}$, emit $(j, \sum_{x \in L_j} x)$

ROUND 2:
- Map: $(j, y_j) \rightarrow (j \bmod N^{1/2}, y_j)$
- Reduce: $\forall$ key k, ... $\Rightarrow$ emit $(k, \sum_{z \in L_k} z)$ $(k \in [0, N^{1/2}))$

ROUND 3:
- Map: (k, z_k) $k \in [0, N^{1/2}[\rightarrow$ emit $(z \bmod N^{1/4}, z_k)$
- Reduce: $\forall$ key $h \in [0, N^{1/4})$, let $L_h = \ldots . \Rightarrow$
  $\Rightarrow$ emit $(0, \sum_{z \in L_h} z)$

ROUND 4:
- Map: empty
- Reduce: let L_0 = list of values with key $0 \Rightarrow$ emit $(0, \frac{1}{N} \sum_{z \in L_0} z)$

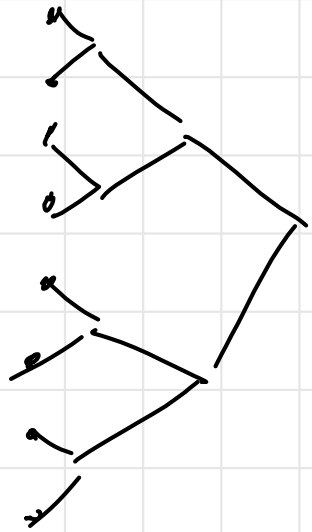
$$M_L = \max\{O(N^{1/4}), O(1)\} = O(N^{1/4})$$

all Reduce all Map

$$M_L = \max\{O(N), O(N^{3/4}), O(N^{1/2}), O(N^{1/4})\} = O(N)$$

(3)

N points



in general, $\log_2 N$ rounds,
$M_L = O(1)$ by sending 2 el. per time

sending M el. per time:

$$N \to \frac{N}{M} \to \frac{N}{M^2} \to \dots \to \frac{N}{M^i} \to \dots \to M \Rightarrow$$

$$\Rightarrow i = \frac{\log_2 N/M}{\log_2 M} + 1 = \frac{\log N}{\log M} = \log_M N$$

Exercise (See ex. 2.3.1.(d) of [LRU14])

Let S be a set of N integers represented by pairs (i, x_i), with $0 \leq i < N$, where i is the key of the pair and x_i is an arbitrary integer. We want to design an efficient MR algorithm to compute the number D of distinct integers in S.

1. Design a simple 2-round MR algorithm to compute D and analyze its local and aggregate space requirements. Under what circumstances is the local space proportional to N, thus not meeting the design goals of MR algorithms?
2. Design a better MR algorithm to compute D in $O(1)$ rounds, using $O\left(\sqrt{N}\right)$ local space and $O(N)$ aggregate space.

① problem: $O(\sqrt{N})$ impossible

ROUND 1:
- Map: $(i, x_i) \rightarrow (x_i, 1)$
- Reduce: $\forall$ key x, let L_x: values with key $x \Rightarrow$ emit $(0, x_i)$

ROUND 2:
- Map: empty
- Reduce: let $L_0 = \{$list of distinct elements$\} \Rightarrow$ emit $(0, |L_0|)$

There could be one very frequent element or all different elements

② part 1: remove duplicates → 2 rounds
part 2: count number of distinct elements → // (1)

PART 1:

ROUND 1:
- Map: $(i, x_i) \to (i \bmod \sqrt{N}, x_i)$
- Reduce: $\forall$ key $j \in [0, \sqrt{N})$ let L_j = list of values with key j
  $\forall$ distinct value x in L_j, emit (x, j)

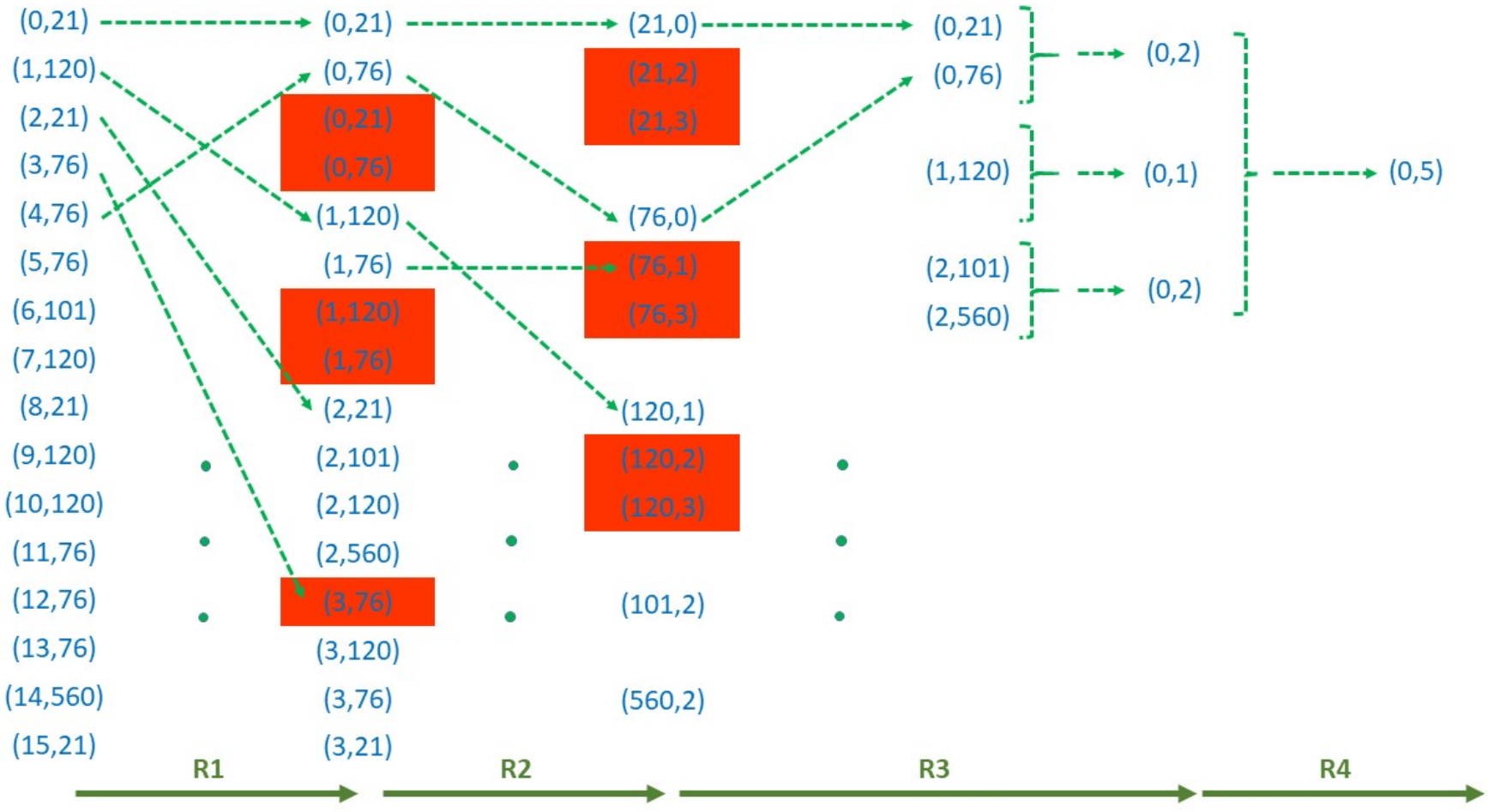
ROUND 2:
- Map: empty
- Reduce: $\forall$ integer x, $L_x = \{$list of values in $[0, \sqrt{N})\} \Rightarrow$
  $\Rightarrow$ emit (x, j'), $j' \in L_x$ (take whatever from L_x)

PART 2:

ROUND 1:
- Map: $(x, j') \Rightarrow (j', x)$ (cannot have $> \sqrt{N}$ different keys)
- Reduce: $\forall j' \in [0, \sqrt{N}-1)$, $L_{j'} = \ldots \Rightarrow$ emit $(\odot, |L_{j'}|)$

ROUND 2:
- Map: empty
- Reduce: $L_\odot = \{\text{partial counts}\} \Rightarrow$ emit $(\odot, \sum_{x \in L_\odot} x)$

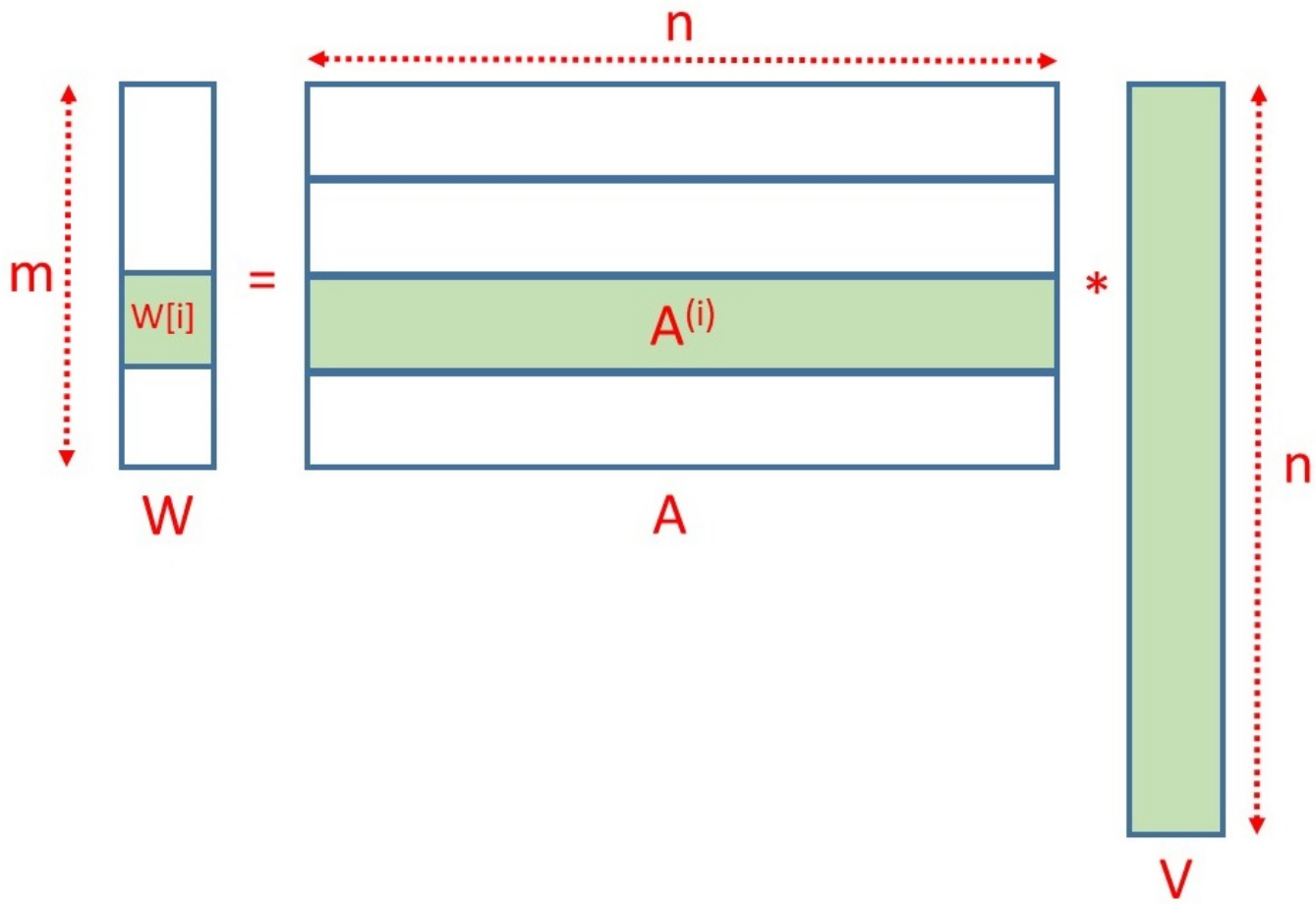
Example ($N = 16$)


(0,21)
(1,120)
(2,21)
(3,76)
(4,76)
(5,76)
(6,101)
(7,120)
(8,21)
(9,120)
(10,120)
(11,76)
(12,76)
(13,76)
(14,560)
(15,21)
(0,21)
(0,76)
(0,21)
(0,76)
(1,120)
(1,76)
(1,120)
(1,76)
(2,21)
(2,101)
(2,120)
(2,560)
(3,76)
(3,120)
(3,76)
(3,21)
(21,0)
(21,2)
(21,3)
(76,0)
(76,1)
(76,3)
(120,1)
(120,2)
(120,3)
(101,2)
(560,2)
(0,21)
(0,76)
(1,120)
(2,101)
(2,560)
(0,2)
(0,1)
(0,2)
(0,5)
R1
R2
R3
R4

Exercise

Design an $O(1)$-round MR algorithm for computing a matrix-vector product $W = A \cdot V$, where A is an $m \times n$ matrix, V is an n-vector, and $m \leq \sqrt{n}$. Your algorithm must use $o(n)$ local space and linear (i.e., $O(mn)$) aggregate space.

How would your solution change if m were larger?



Exercise

Design a 2-round MR algorithm for the word count problem, using random partitioning, and analyze its local and aggregate space requirements.