Lecture 20

Massively Parallel Algorithms: Sorting, Counting Distinct Elements

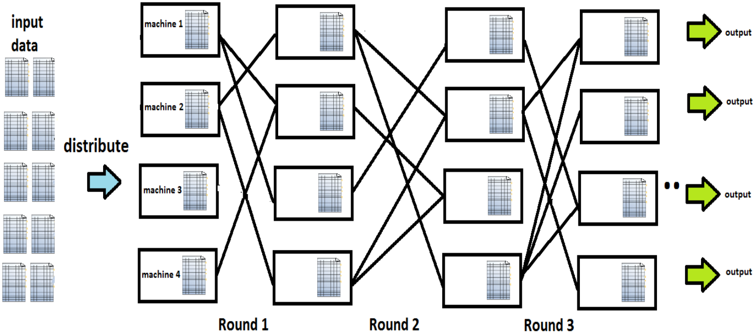
Course: Algorithms for Big Data

Instructor: Hossein Jowhari

Department of Computer Science and Statistics
Faculty of Mathematics
K. N. Toosi University of Technology

Spring 2021

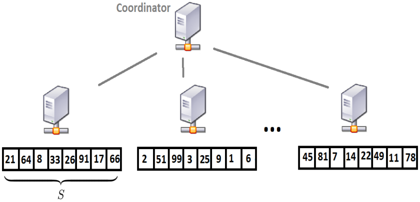
MPC: recap



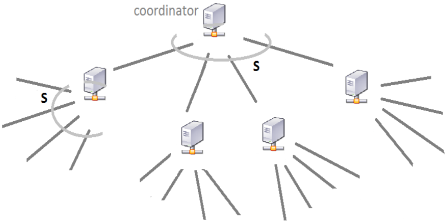
- N input data size, M machines
- $S = o(N)$ memory size per machine
- Each machine communicates at most S words in each round

Sum of N integers

Each machines computes its local sum and sends it up to the coordinator.



- $\sqrt{N} < S \leq N$
- $M = \frac{N}{S}$
- $\#$ rounds $= 1$



- $2 \leq S \leq \sqrt{N}$
- $M = \frac{N}{S}$
- $\#$ rounds $= \log_S N$

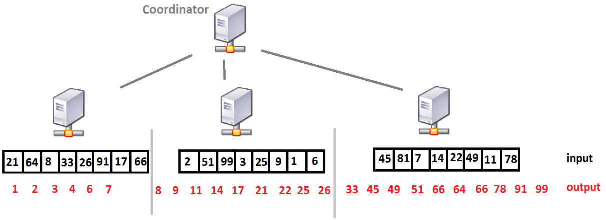
Sorting N integers

Input Data: N integers $\{a_1, a_2, \dots, a_N\}$

Assumption: The input integers are distinct (no repetitions).

The input is partitioned among the machines. Each machine gets $S = O(N^{2/3})$ elements. $M = O(N^{1/3})$

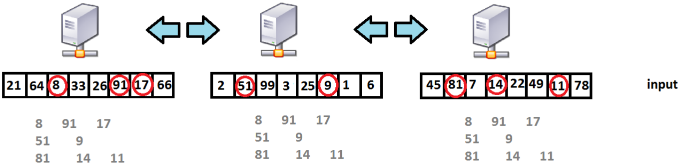
Output: The rank of each element is known by some machine.



A parallel algorithm inspired by quicksort

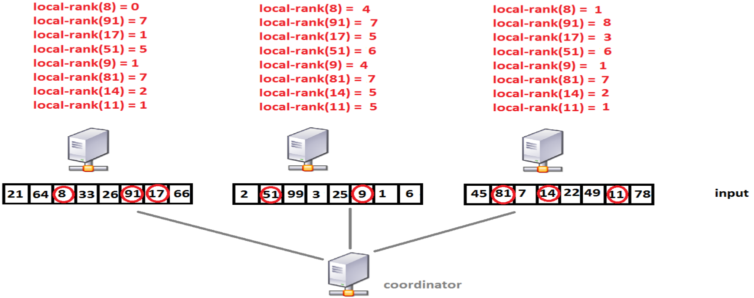
Stage 1: selecting potential pivots

- The machines select a random subset X of the elements. Each element is picked with probability $p = \frac{N^{1/3} \log N}{N}$. We call these elements potentially pivot elements.
- The machines communicate their selected elements. At the end of this stage, the machines all know the random subset X.



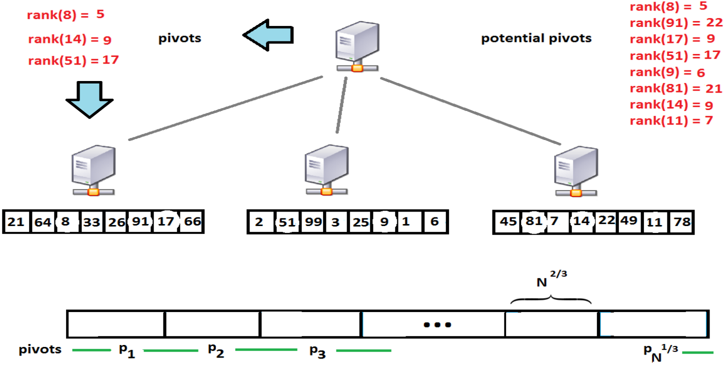
Stage 2: local ranks for the potential pivots

- For each $x \in X$, the machines compute $\text{local-rank}(x)$: how many elements in their input is smaller than x.
- For each $x \in X$, the machines send $(x, \text{local-rank}(x))$ to the coordinator.



Stage 3: computing the pivots

- Having received the local ranks, the coordinator computes the (global) rank of each potential pivot $x \in X$.
- For each $N^{2/3}$ length interval in $\{1, \ldots, N\}$, the coordinator selects an element from the potential pivots X that has a rank with that interval.
- The coordinators sends the selected pivots to all machines.


rank(8) = 5
rank(14) = 9
rank(51) = 17
pivots
potential pivots
rank(8) = 5
rank(91) = 22
rank(17) = 9
rank(51) = 17
rank(9) = 6
rank(81) = 21
rank(14) = 9
rank(11) = 7
21 64 8 33 26 91 17 66
2 51 99 3 25 9 1 6
45 81 7 14 22 49 11 78
N 2/3
• • •
pivots
p1
p2
p3
pN 1/3

Lemma: With high probability, there exists a potential pivot from each $N^{2/3}$ length interval.

Proof: Each element is picked probability $p = \frac{N^{1/3} \log N}{N}$. In expectation, number of elements selected from a rank interval is $\log N$.

Using Chernoff bound, with high probability there is at least one elements from each interval. □

Let $p_1, p_2, \ldots, p_{N^{1/3}}$ be the selected pivots.

Consider the real intervals:

$$I_1 = (-\infty, p_1], \;\; I_2 = (p_1, p_2], \ldots \;\; , \;\; I_{N^{1/3}+1} = (p_{N^{1/3}}, +\infty)$$

Machine M_i will be responsible for the elements in the interval I_i

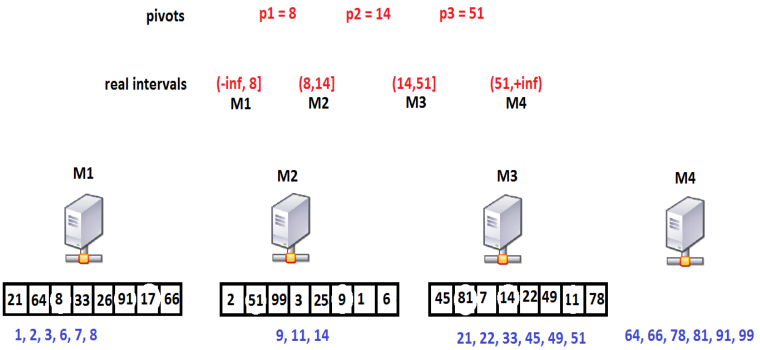
Stage 4: sending the elements to the responsible machines

- Each machine, for each element y in its local memory, sends y to the machine M_j where

$$y \in I_j = (p_{j-1}, p_j)$$

- Each machine locally sorts the received elements.

Let S_i be the sorted lists owned by machine M_i. The final list $S_1, S_2, \ldots, S_M$ is sorted in the increasing order.


pivots
p1 = 8
p2 = 14
p3 = 51
real intervals
(-inf, 8]
M1
(8,14]
M2
(14,51]
M3
(51,+inf)
M4
M1
M2
M3
M4
21 64 8 33 26 91 17 66
2 51 99 3 25 9 1 6
45 81 7 14 22 49 11 78
1, 2, 3, 6, 7, 8
9, 11, 14
21, 22, 33, 45, 49, 51
64, 66, 78, 81, 91, 99

Round complexity of each stage

- Stage 1: each machine selects $O(\log N)$ potential pivots in expectation. With high probability (Chernoff bound), number of selected pivots is $O(\log N)$.

  The potential pivots are broadcasted to other machines. There are $O(N^{1/3})$ machines. This can be done in one round. Recall the communication limit is $S = O(N^{2/3})$.

- Stage 2: Each machine sends $O(N^{1/3} \log N)$ words to the coordinator. This can be done in 2 rounds using a broadcast tree of depth 2.

- Stage 3: The coordinator sends $N^{1/3}$ number of pivots to all machines. This can be done in 1 round.

- Stage 4: The machines sends the elements to their responsible machines. Each machine has $O(N^{2/3})$ elements. Each real interval has at most $O(N^{2/3})$ elements in it. This stage can also be done in $O(1)$ round.

[Theorem] There is a $O(1)$ round MPC algorithm for sorting N numbers where each machine has $O(N^{2/3})$ space.

Question: What about the general case when $S = n^{\epsilon}$?

[Theorem] There is a $O(1/\epsilon)$ round MPC algorithm for sorting N numbers where each machine has $O(N^{\epsilon})$ space.

See Parallel Algorithms (Chapter 6) by Mohsen Ghaffari.

Question: What if the numbers are not distinct?

One idea is to perturb the numbers so that all numbers become distinct. We can do this by adding a small random ϵ to all numbers. (With high probability all numbers are distinct now.) We sort the perturbed numbers and then scrap the added noise.

Question: A MPC algorithm for counting distinct elements?