The Round Complexity of Distributed Sorting

by

Boaz Patt-Shamir
Marat Teplitsky
Tel Aviv University
Motivation

• Distributed sorting
• Infrastructure is more and more distributed
  – Cloud
  – Smartphones
CONGEST model

• Models congestion in a network
  – Bandwidth restriction: Message complexity in $O(\log n)$

• Abstract model
  – Removes complexity
Network

- Fully connected network (clique)
- Message in $O(\log n)$
- Synchronous rounds

Switch with 512 ports
Problem statement

• Number of nodes: $n$
  – Denoted as $V = \{v_1, \ldots, v_n\}$

• Input Values: max $n$ per node
  – Max $n^2$ in total

• Goal: Sort in $O(\log \log n)$ rounds w.h.p

• Definition
  – With high probability (w.h.p): $1 - n^{-O(1)}$
The algorithm – Overview

• Split the input values into \( n \) ranges
• Each node sorts one range
  – Send input values to the corresponding node
The algorithm – Partition phase

• Create a global order on the keys
  – The nodes are order by their id
  – Each node creates an arbitrarily local order
  – The global order is then $\sum_{l=1}^{i-1} a_i + k_{ij}$

• Partition them into $n$ disjoint ranges
The algorithm – Partition phase
The algorithm – Partition phase
The algorithm – Partition phase

Choose order of delimiters: 12, 6, 10

Choose order of delimiters: 12, 6, 10
The algorithm – Partition phase

Broadcast order of delimiters: 12, 6, 10

V1

V2

V3

V4

23 22 10 89

2 3 4

18 31

8 9

10

6

12

12

12
The algorithm – Partition phase

Send delimiters: 24,5,50
The algorithm – Partition phase

Broadcast delimiters: 24, 5, 50

\[ v_1 \]

\[ v_2 \]

\[ v_3 \]

\[ v_4 \]
The algorithm – Partition phase

\[ V_1 \]
23 22 10 89
1 2 3 4
[min, 4]

\[ V_2 \]
90 5 3
5 6 7
[5, 23]

\[ V_3 \]
18 31
8 9
[24, 49]

\[ V_4 \]
50 6 24
10 11 12
[50, max]

17-Apr-12

Seminar in Distributed Computing
Christoph Burkhalter
The algorithm – First stage

• Only nodes with max $2n \ln \ln n$ keys in their range are *active* nodes in this phase
• Keys are *active* if their destination node is active.
The algorithm – First stage

repeat

• for each active key pick intermediate destination (source node)

• for each final destination, pick one key and send it (intermediate node)

• Send all other received keys back

until all active key reached their destination
The algorithm – First stage

• Active nodes: Max 2 keys in the range

<table>
<thead>
<tr>
<th></th>
<th>$V_1$</th>
<th></th>
<th>$V_2$</th>
<th></th>
<th>$V_3$</th>
<th></th>
<th>$V_4$</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[min,4]</td>
<td></td>
<td>[5,23]</td>
<td></td>
<td>[24,49]</td>
<td></td>
<td>[50,max]</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>5, 6, 10, 18, 23</td>
<td></td>
<td>24, 31</td>
<td></td>
<td>50, 89, 90</td>
<td></td>
</tr>
</tbody>
</table>
The algorithm – First stage

* Active nodes
The algorithm – First stage

* Active nodes
The algorithm – First stage

* Active nodes

* * *
The algorithm – First stage

* Active nodes
The algorithm – First stage

* Active nodes
The algorithm – Cleanup stage

• Do the same for the nodes with more than $2 n \ln \ln n$ keys.

• Local sort the keys
Chernoff bound
Analysis

• Lemma 1.1:
  – With high probability, the number of non-selected segments is at most \( \frac{2n}{\ln n \ln \ln n} \).
Analysis

• Lemma 1.2:
  – With high probability, the number of ranges with more than $2n \ln \ln n$ keys is at most $\frac{2n}{\ln n \ln \ln n}$.
Analysis

• Lemma 2.1:
  – W.h.p., the number of keys remaining to the cleanup stage is at most \( \frac{4n^2}{\ln n} \).
Analysis

• Lemma 2.2:
  – In the cleanup stage, w.h.p., all ranges are of size $O(n)$. 
Analysis

• Lemma 3.1:
  – If there are more than \( n \) active keys with destination \( v_i \), then w.h.p. at least \( \frac{n}{9} \) keys will be delivered at \( v_i \) in one iteration.
Analysis

• Lemma 3.2:
  – If there are \textbf{at most} $n$ active keys with destination $v_i$, then w.h.p. all keys will be delivered in $O(\ln \ln n)$ iterations.
Related Work

• Concurrently to this paper, Lenzen and Wattenhofer proved the following:
  – Suppose there are $O(n)$ messages in each node and the number of messages destined to each node is $O(n)$, then routing all messages can be done in $O(1)$

• With this, the algorithm can be improved to work in $O(1)$
Q&A