Massively Parallel Computation

• Computational Model
• Summing
• Sorting
• Minimum Spanning Tree

Computational Model

• Massively Parallel Computation (MPC) model.
  • $P$ processors each with space $S$.
  • Typically $S = N^\epsilon$ and $P = N^{1-\epsilon}$.
  • Synchronous computation in rounds.
  • Round = local computation + communication.
  • Communication into a processor is $< S$.
• Complexity model.
  • Rounds and space ($\implies$ communication)
  • Computation is free (!)
• Implementations.
  • Map Reduce
  • Bulk-Synchronous parallel

Philip Bille
Summing

- Given a list of $N$ integers $A_0, A_1, ..., A_{N-1}$ compute their sum.
- Input distributed arbitrarily among processors.

Massively Parallel Computation

- Computational Model
- Summing
- Sorting
- Minimum Spanning Tree

Sorting

- Given a list of $N$ integers $A_0, A_1, ..., A_{N-1}$ compute list $(A_0, \text{rank}(A_0)), (A_1, \text{rank}(A_1)), ..., (A_{N-1}, \text{rank}(A_{N-1}))$
- Input and output distributed arbitrarily among processors.
Sorting

• Goal. Sorting in $O(1)$ rounds whp. with $S = \tilde{\Theta}(\sqrt{N})$ and $P = \tilde{\Theta}(\sqrt{N})$.

• Idea.
  - Sample $\tilde{\Theta}(\sqrt{N})$ items and use sample to partition items into $\tilde{\Theta}(\sqrt{N})$ ranges.
  - Distribute items according to ranges and sort each range locally.

\[ \mathcal{X} = \tilde{\Theta}(\mathcal{P}) \]
\[ \mathcal{P} = \tilde{\Theta}(\mathcal{S}) \]

\begin{itemize}
  \item Sample
  \begin{itemize}
    \item Each processor samples its items with probability $2P \ln N/N$ and sends these to processor 0.
    \item Processor 0 broadcasts the set of samples to all processors.
    \item Let $X$ be the set of samples. $|X| \leq 4P \ln N$ whp.
  \end{itemize}
\end{itemize}

\begin{itemize}
  \item Lemma. Let $I$ be the sorted input. Consider a partition of $I$ into $P$ ranges of $N/P$ consecutive items. Then, all ranges contain at least one item from $X$ whp.
  \item Proof.
  \begin{itemize}
    \item $\Pr(\text{range contains no items}) = \left(1 - \frac{2P \ln N}{N}\right)^P \leq e^{-2 \ln N} = \frac{1}{N^2}$
    \item $\Rightarrow \Pr(\text{some range contain no items}) = P \cdot \frac{1}{N^2} < \frac{1}{N}$
    \item $\Rightarrow \Pr(\text{all ranges contain at least 1 item}) > 1 - \frac{1}{N}$
  \end{itemize}
\end{itemize}

\begin{itemize}
  \item Compute local histogram.
  \begin{itemize}
    \item Each processor counts number of items in ranges defined by $X$.
    \item Each histogram uses $O(|X|)$ space.
  \end{itemize}
\end{itemize}
• Compute global histogram.
  • Each processor sends count for range $i$ to processor $i \mod P$.
  • Processor $i$ sums counts for range $i \mod P$ and sends sum to processor 0.
  • Processor 0 constructs global count.
  • Each processor is responsible for counting $\lfloor \frac{|X'|}{P} \rfloor$ ranges and receives $O(P \log N)$ integers.

• Exchange.
  • Assign each range defined by $X'$ to a processor.
  • Each processor sends each of its items to processor assigned to corresponding range.
  • Each processor locally sorts its items.
  • Output sorted sequence.

• Select.
  • Processor 0 selects $X' \subseteq X$ such that each range defined by $X'$ contains $O(\frac{N}{P})$ items from $I$ and $|X'| = O(P)$.
  • Processor 0 broadcasts $X'$ to all machines.
  • Sampling lemma $\implies X'$ exists whp.

• Theorem. Sorting in $O(1)$ rounds whp. with $S = \tilde{\Theta}(\sqrt{N})$ and $P = \tilde{\Theta}(\sqrt{N})$. 

| 7, 42, 3 | 1, 18, 2 | 9, 10, 11 | 4, 51, 6 | 3, 24, 92 | 56, 19, 8 | 5, 22, 33 |
Massively Parallel Computation

- Computational Model
- Summing
- Sorting
  - Minimum Spanning Tree

Minimum Spanning Tree

- Let $G$ be a graph with $n$ nodes and $m$ edges.
- Goal. MST in $O(1/\varepsilon)$ rounds whp. for $S = \Theta(n^{1+\varepsilon})$ and $P = \Theta(m/S) = \Theta(m/n^{1+\varepsilon})$
- Idea.
  - Repeatedly filter edges not part of MST in rounds.
  - When all edges fit on one processor compute the MST directly.

Minimum Spanning Tree

- Minimum spanning tree. Given a connected, weighted, undirected graph compute the minimum spanning tree (MST).
- Input given as list of edges with weights. Output edges in MST.
- Input and output distributed arbitrarily among processors.

Minimum Spanning Tree

- Shuffle.
  - Let $m'$ be the current edges. Initially, $m' = m$.
  - Choose $k = 2m'/n^{1+\varepsilon}$ active processors.
  - Distribute edges among active processors randomly.
  - Let $E_i$ be the edges at processor $i$. $|E_i| = n^{1+\varepsilon}$ whp.
Filter. Active processor $i$:
- Computes a local minimum spanning forest of $G = (V, E)$.
- Discards all other edges in $E$.

Repeat.
- Repeat shuffle and filter step until remaining edges fit on a single machine.
- Then compute MST.

Correctness.
- Edges in $E_i$ that are not in the local minimum spanning forest are not in the MST.

Rounds.
- Total edges remaining after a round is $\leq k(n - 1) = \frac{2m'}{n^{1+\epsilon}}(n - 1) < \frac{2m'}{n^\epsilon}$.
- $\implies$ A round reduces edges by factor $n^\epsilon$.
- $\implies$ After $O(1/\epsilon)$ rounds the remaining edges is $< n^{1+\epsilon}$. 
• Theorem. MST in $O(1/\epsilon)$ rounds whp. for $S = \Theta(n^{1+\epsilon})$ and $P = \Theta(m/n^{1+\epsilon})$