Examples of Massively Parallel Non-Numerical Algorithms

Algorithm Engineering for Parallel Sorting and Graph Generation
Michael Axtmann, Sebastian Lamm and Peter Sanders
Institute of Theoretical Informatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Sorting Algorithms
One of the most fundamental non-numeric algorithms
- Load balancing with space-filling curves built down to sorting on the curve
- Sorting brings “similar” data together
- Used to build index data structures

Graph Generation
- Complex networks composed of billions of entities
- Need for algorithms capable of processing massive amounts of data
- Real-world datasets are often scarce or too small
- Graph generators provide scalable synthetic instances

Requirements
- Scale to largest available machines
- Performance guarantees with asymptotic analysis
- Robustness with low overhead
- Input size
- Duplicates, keys
- Distribution of input elements

Graph Models
- Erdos-Renyi Graphs \( G(n, m) \) and \( G(n, p) \)
- Random Hyperbolic Graphs \( RHG(n, γ, d) \)
- Random Geometric Graphs \( RGG(n, r) \)
- Random Delaunay Graphs \( RDG(n) \)

Zero Communication Generators
- Communication-free sampling algorithms
- Neighborhood queries using efficient recomputations

Asymptotic Analysis

<table>
<thead>
<tr>
<th>Algorithm</th>
<th># Messages ( O(\cdot) )</th>
<th>Comm. Vol.</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gather-merge</td>
<td>( \log_p n )</td>
<td>( n )</td>
<td>( \text{best case} )</td>
</tr>
<tr>
<td>FIS [1,2]</td>
<td>( \log_p )</td>
<td>( n/\sqrt{p} )</td>
<td>( \text{not robust} )</td>
</tr>
<tr>
<td>Bitonic</td>
<td>( \log^2 p )</td>
<td>( \log p )</td>
<td>( \text{not robust} )</td>
</tr>
<tr>
<td>HC quicksort</td>
<td>( \log p )</td>
<td>( \log p )</td>
<td>( \text{best case} )</td>
</tr>
<tr>
<td>RQuick [2]</td>
<td>( \log^2 p )</td>
<td>( \log p )</td>
<td>( \text{not robust} )</td>
</tr>
<tr>
<td>JanusSort [3]</td>
<td>( \log^2 p )</td>
<td>( \log p )</td>
<td>( \text{best case} )</td>
</tr>
<tr>
<td>HykSort</td>
<td>( \frac{k \log_p n}{p} )</td>
<td>( \frac{k \log_p n}{p} )</td>
<td>( \text{not robust} )</td>
</tr>
<tr>
<td>AMS-sort [1,2]</td>
<td>( \frac{\log p}{p} )</td>
<td>( \frac{n}{p} )</td>
<td>( \text{sampling cost} )</td>
</tr>
</tbody>
</table>

Experimental Results

- Running times of different algorithms on 262 144 cores
- Graphs of up to \( 2^{42} \) vertices and \( 2^{46} \) edges in less than 20 minutes

References