Institute of Theoretical Informatics, Algorithmics II

Examples of Massively Parallel Non-Numerical Algorithms

Algorithm Engineering for Parallel Sorting and Graph Generation

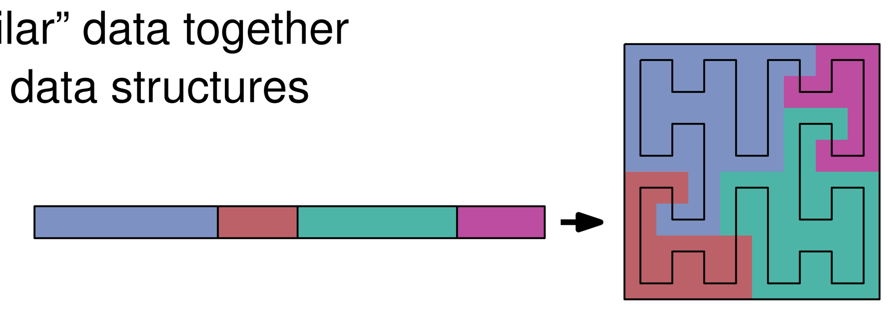
Michael Axtmann, Sebastian Lamm and Peter Sanders

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 buils down to sorting on the curve
- Sorting brings "similar" data together
- Used to build index data structures

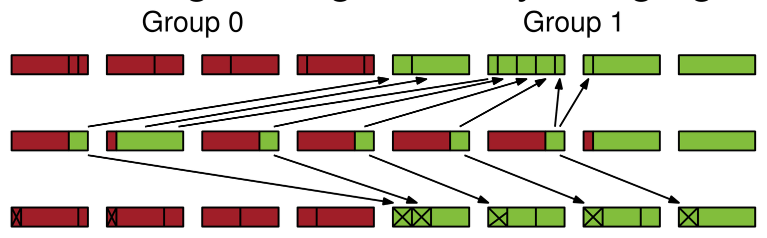


Requirements

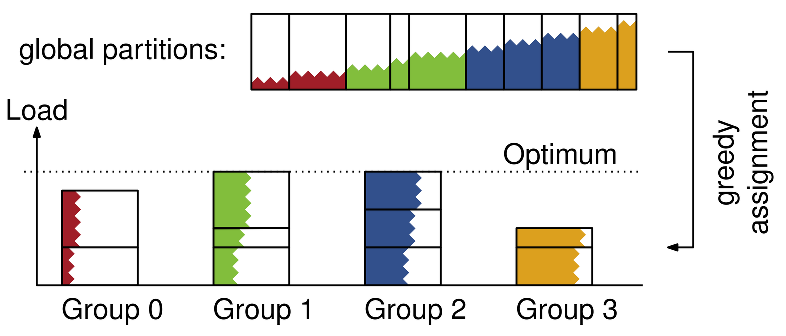
- Scale to largest available machines
- Performance guarantees with asymptotic analysis
- Robustness with low overhead
  - Input size
  - Duplicates keys
  - Distribution of input elements

Ideas

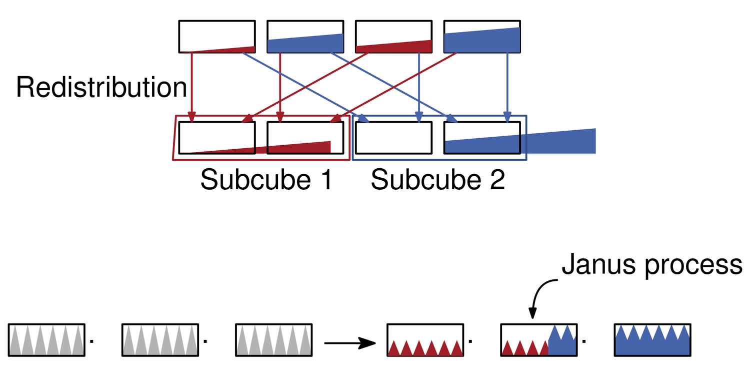
Message assignment by merging



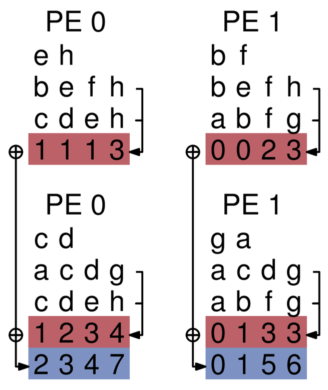
Optimal overpartition



Avoid data imbalances



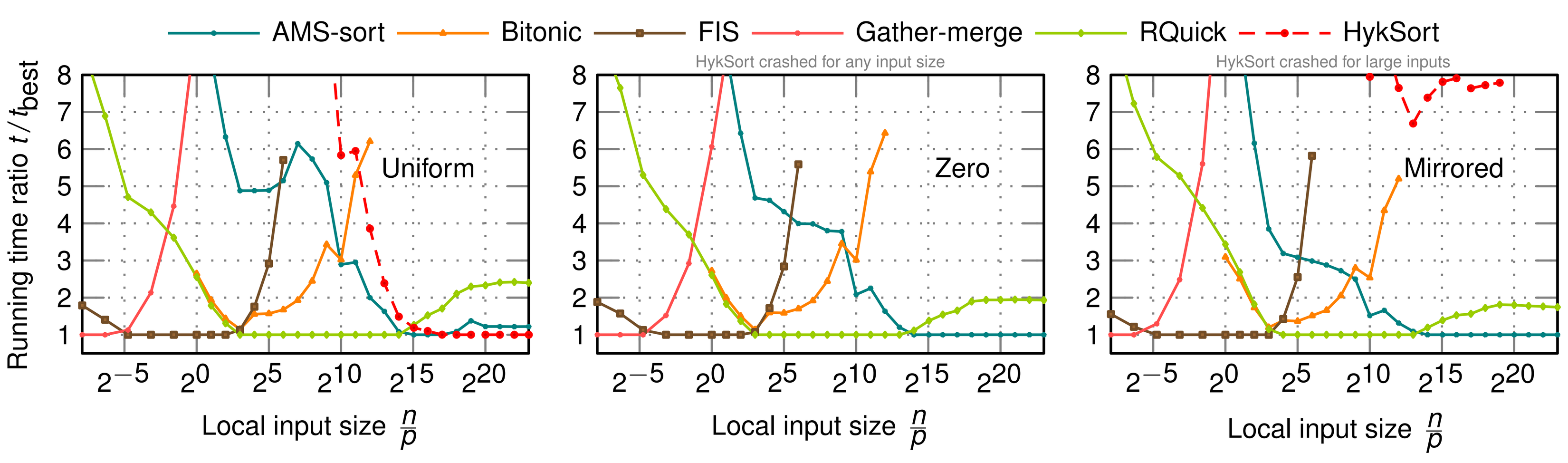
Trade of startups and work



Asymptotic Analysis Implicit $\mathcal{O}(\cdot)$

Algorithm	# Messages	Comm. Vol.	Remarks
Gather-merge	$\log p$	n	
FIS [1,2] NEW	$\log p$	$n/\sqrt{p}$	robust
Bitonic	$\log^2 p$	$\frac{n}{p}\log^2 p$	
HC quicksort	$\log^2 p$	$(p+)\frac{n}{p}\log p$	best case
RQuick [2] NEW	$\log^2 p$	$\frac{n}{p}\log p$	robust, $p = 2^k$
JanusSort [3] NEW	$\log^2 p$	$\frac{n}{p}\log p$	robust
HykSort	$\geq k \log_k p$	$\geq \frac{n}{p}\log_k p$	not robust
AMS-sort [1,2] NEW	$\leq k \log_k p$	$\leq \frac{n}{p}\log_k p$	robust
Sample sort	$\geq p$	$\geq n/p$	+sampling cost

Experimental Results



Running times of different algorithms on 262 144 cores

References

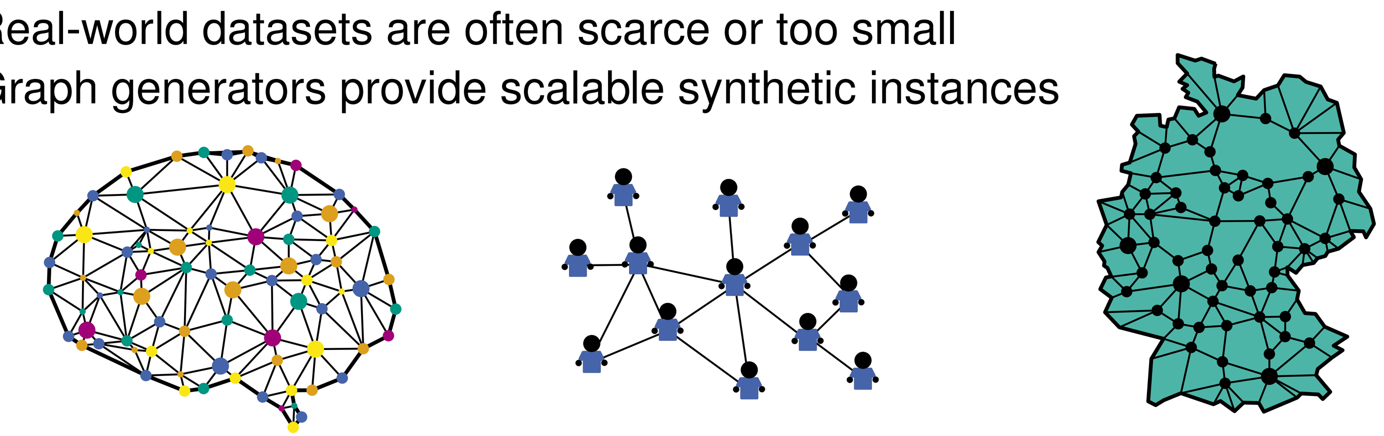
Wiebigke, A. and Axtmann, M., 2018. Lightweight MPI Communicators with Applications to Perfectly Balanced Quicksort. To appear at IPDPS 2018.

Axtmann, M. and Sanders, P., 2017. Robust massively parallel sorting. ALENEX'17.

Axtmann, M., Bingmann, T., Sanders, P., and Schulz, C., 2015. Practical massively parallel sorting. SPAA'15.

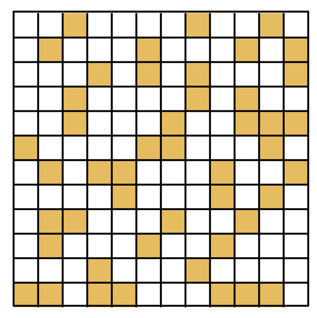
Graph Generation

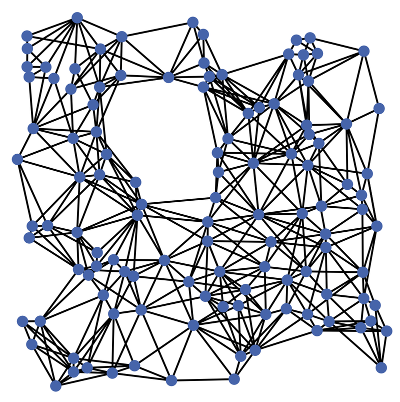
- Complex networks composed of billions of entities
- Need for algorithms capable of processing massive amounts of data
- Real-world datasets are often scarce or too small
- Graph generators provide scalable synthetic instances



Graph Models

Erdos-Renyi Graphs $G(n,m)$ and $G(n,p)$

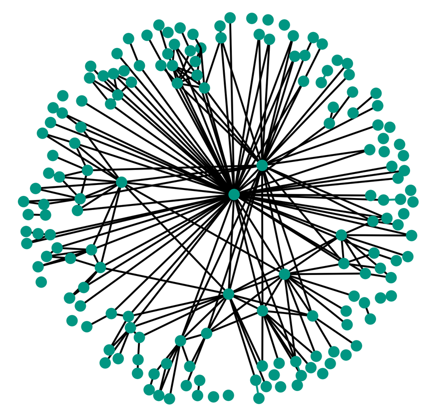




Random Geometric Graphs $RGG(n,r)$

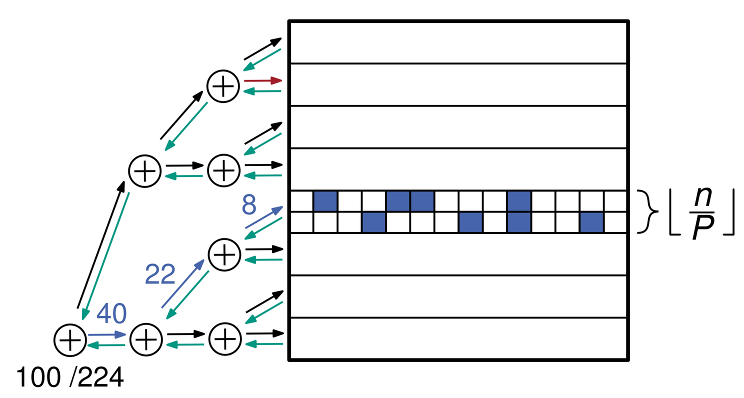
Random Delaunay Graphs $RDG(n)$

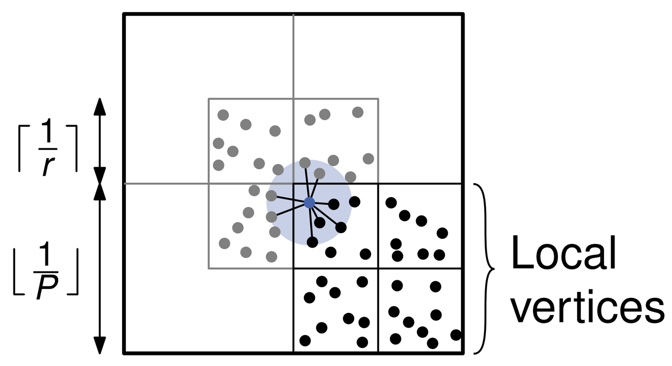
Random Hyperbolic Graphs $RHG(n, \gamma, \bar{d})$



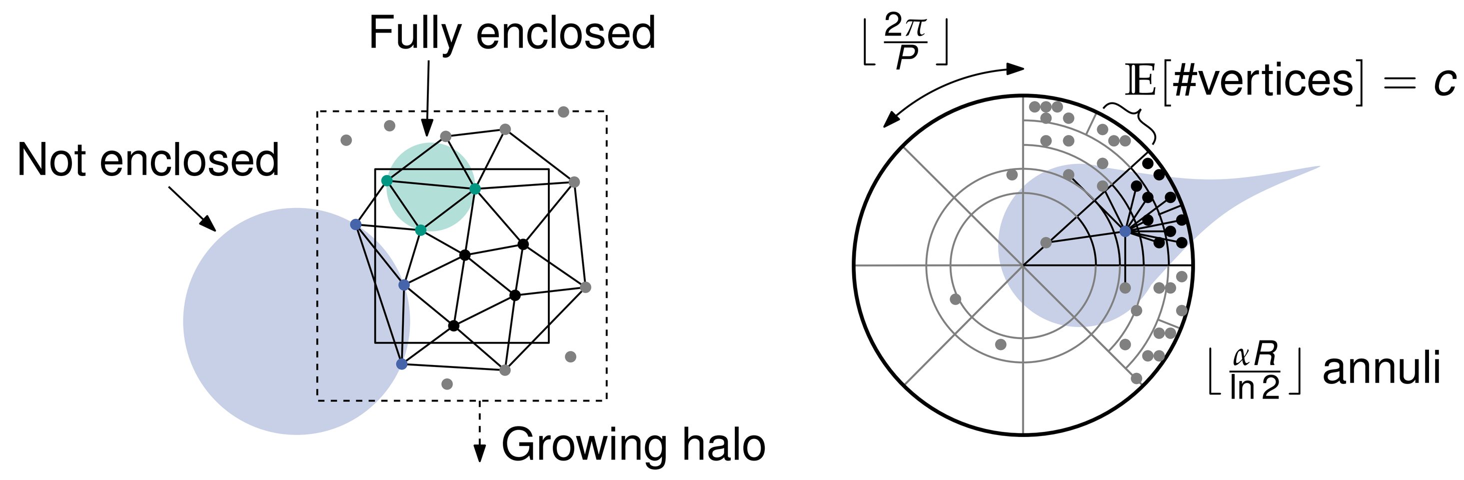
Zero Communication Generators

- Communication-free sampling algorithms

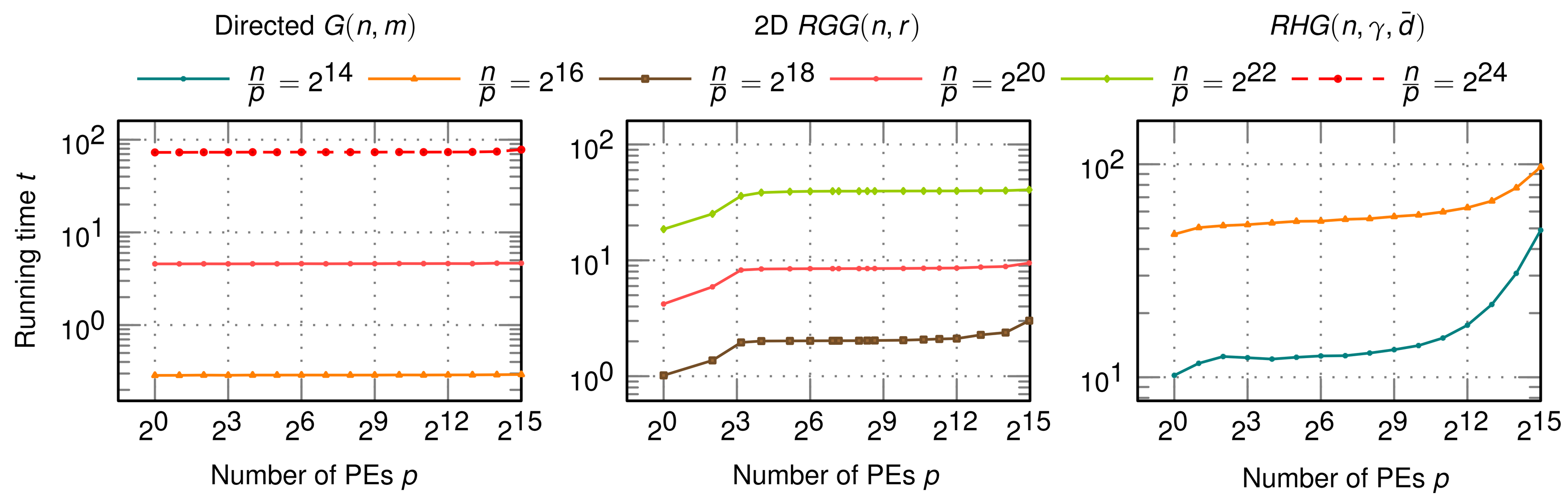




- Neighborhood queries using efficient recomputations



Experimental Results



Graphs of up to 2^{42} vertices and 2^{46} edges in less than 20 minutes

References

Funke, D., Lamm, S., Sanders, P., Schulz, C., Strash, D. and von Looz, M., 2017. Communication-free massively distributed graph generation. To appear at IPDPS 2018.

Sanders, P., Lamm, S., Hübschle-Schneider, L., Schrade, E. and Dachsbacher, C., 2017. Efficient random sampling-parallel, vectorized, In: Transactions on Mathematical Software

Lamm, S., Sanders, P., Schulz, C. and Strash, D., 2017. Communication efficient algorithms for generating massive networks (Master thesis, Karlsruher Institut für Technologie (KIT)).