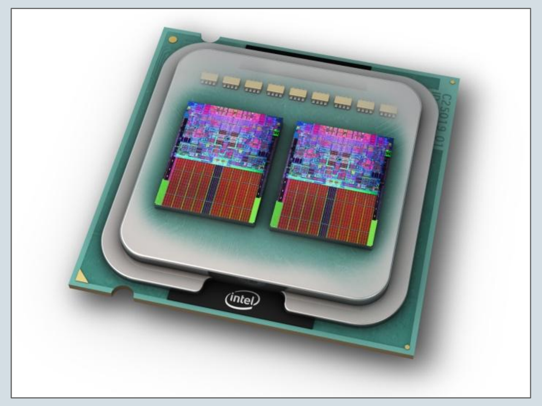
MADALGO
CENTER FOR MASSIVE DATA ALGORITHMICS

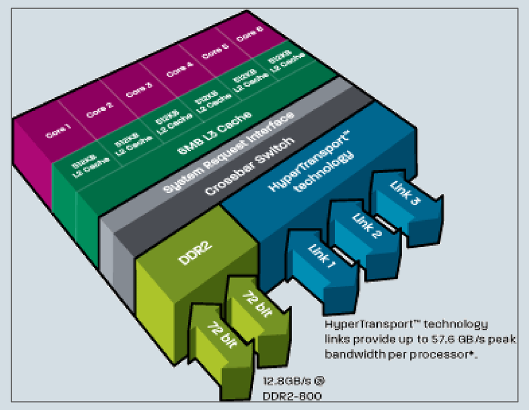

Nodari Sitchinava
Aarhus University

Parallel External Memory Model for Private-cache Chip Multiprocessors

Motivation

Parallel processors are becoming common place. Each core of a multi-core processor consists of a CPU and a private cache. Inter-processor communication is performed by writing and reading to/from shared memory (higher level cache or the main memory).

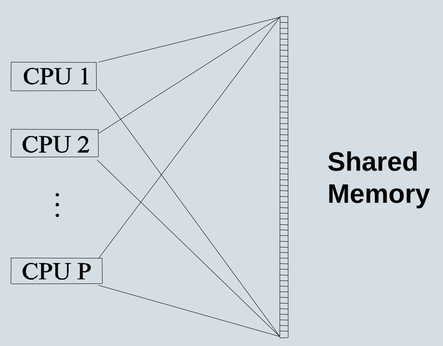

Intel Quad-core processor


AMD 6-core processor

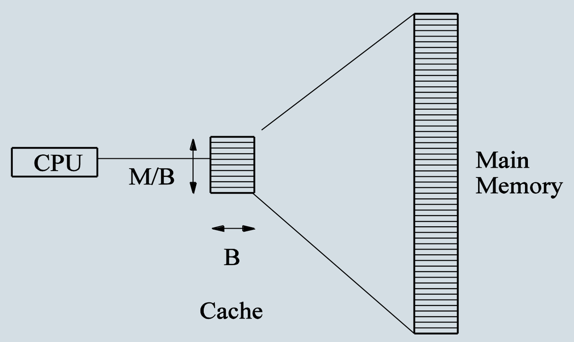
We need new models of computation which model parallelism while taking into account the latencies of memory hierarchies.

PEM Model

The existing *parallel random access (PRAM)* model does not have a notion of caches and, therefore, does not account for spatial locality. On the other hand, the existing *external memory (EM)* model, while explicitly modeling cache access, is not a parallel model.

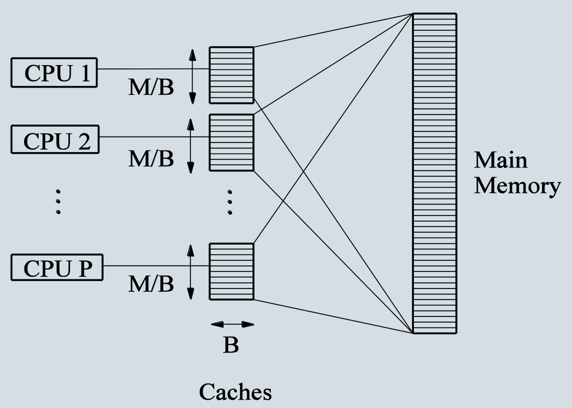


PRAM model



EM model

We combine the two models to obtain the *parallel external memory (PEM)* model – a parallel model that explicitly counts cache accesses.



PEM model

There are three complexity metrics in the PEM model:

- Space – amount of memory used
- Parallel time – max. time spent by a CPU for computing
- Parallel I/O – max. number of transfers between the main memory and the cache of a CPU

Fundamental Problems

Prefix sums: Given a sequence A = {a_1, …,a_n} compute B = {b_1, …,b_n}, s.t. $b_i = a_1+\ldots+a_i$.

A: {2, 4, 1, 9, 3, 2, 7, 1, 1, 8} → B: {2, 6, 7, 16, 19, 21, 28, 29, 30, 38}

Prefix sums is a basic building block for solutions to many problems in parallel models. Our PEM prefix sums solution is optimal in all three complexity metrics.

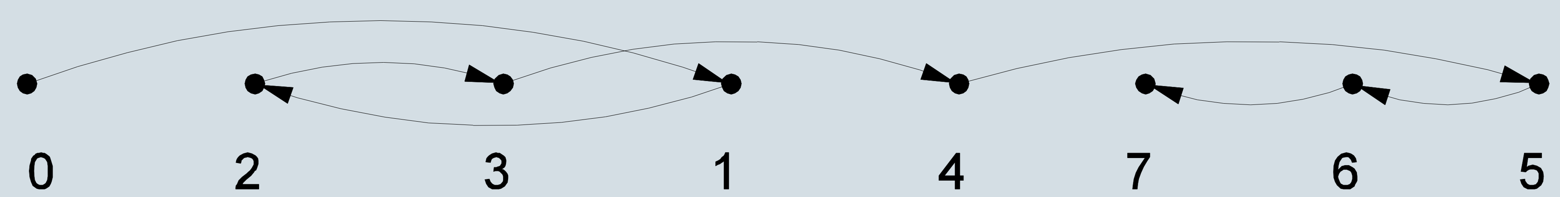
Sorting is a fundamental problems in computer science and a building block for solutions to many problems.

{3, 5, 1, 7, 2, 8, 4, 6, 9, 10} → {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

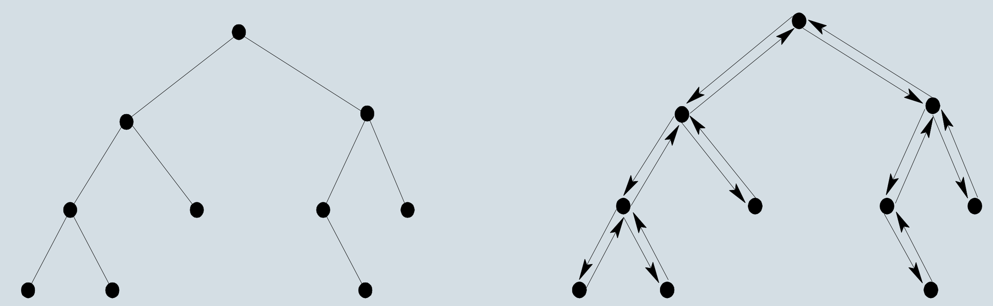
We develop an optimal sorting algorithm in all three complexity metrics in the PEM model.

Problems on Trees

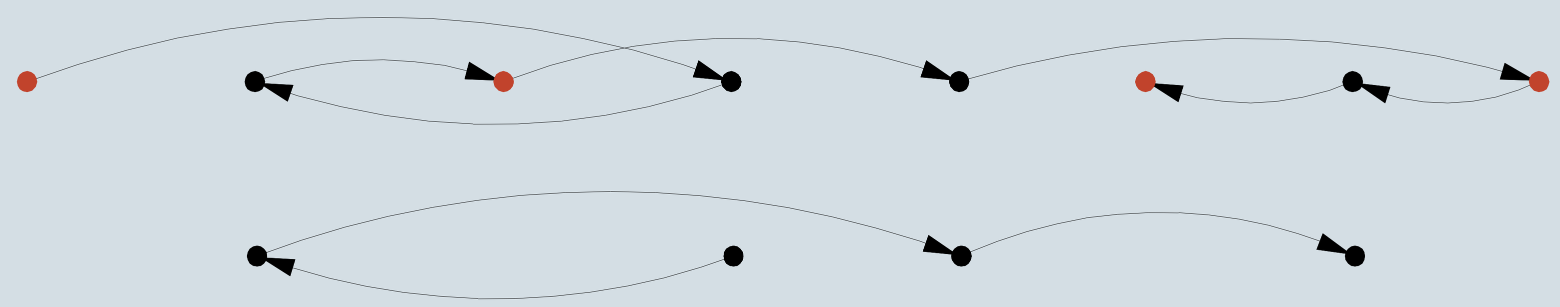
List ranking: Given a linked list of *N* elements assign each element its rank, the distance from the head of the list to the element.



List ranking is a linchpin to solving problems on trees:

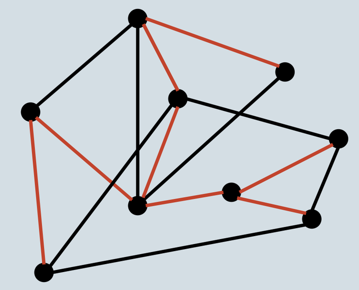


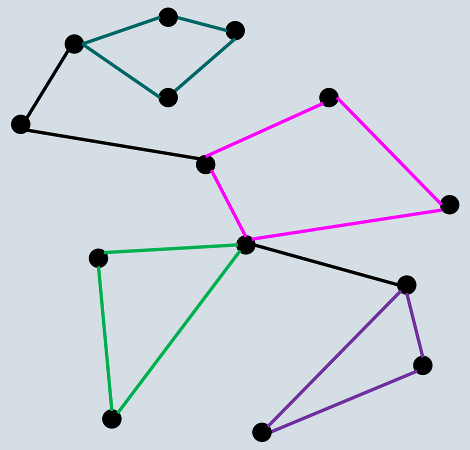
Solution: We solve the list ranking problem by identifying a maximal independent set via *deterministic coin tossing*, bridging the set out and recursively solving the problem on the remaining list. All steps are completed in parallel and I/O efficiently, resulting in optimal sorting complexity in all three complexity metrics in the PEM model.

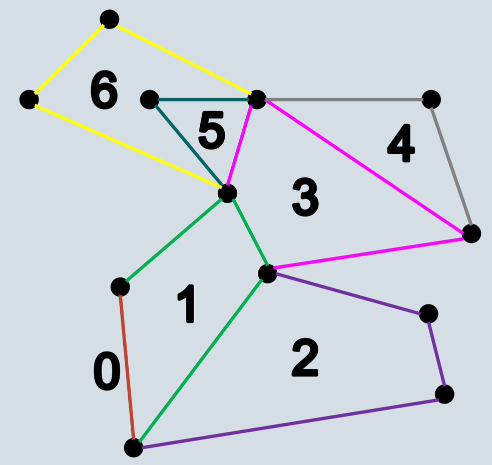


Problems on Graphs

Using list ranking and solutions on trees, we can solve problems on graphs efficiently:


Minimum spanning tree

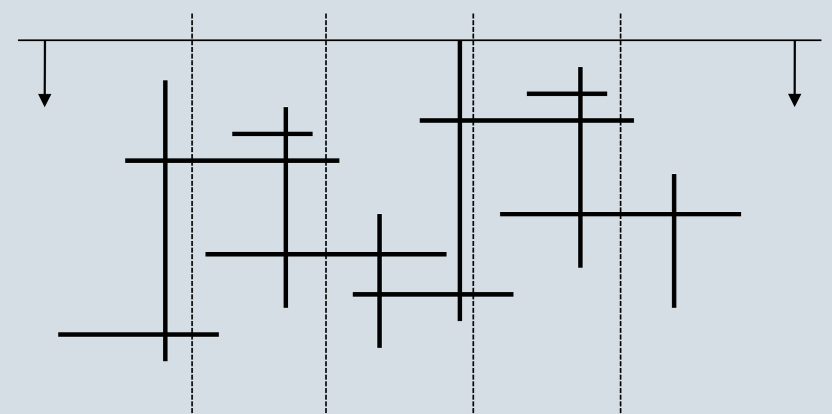

Bi-connected components



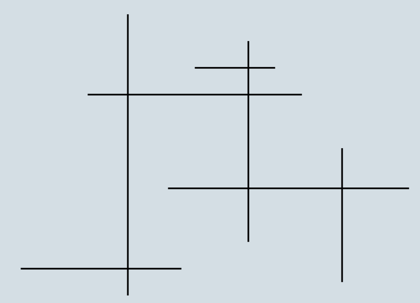
Ear decomposition

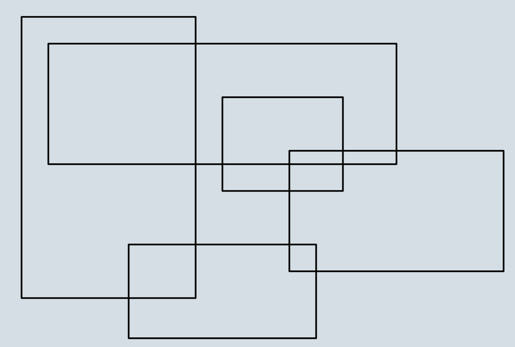
Orthogonal Geometric Problems

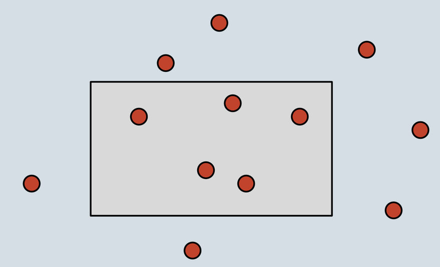
We parallelize the *distribution sweeping* technique.



Using distribution sweeping, we obtain efficient solutions to orthogonal geometric problems:


Line segment intersections


Rectangle intersections


Range query

References

[1] L. Arge. M. T. Goodrich, M. Nelson, N. Sitchinava, *Fundamental algorithms for private-cache chip multiprocessors*, SPAA, 2008.
[2] L. Arge, M.T. Goodrich, N. Sitchinava, *Parallel external memory graph algorithms*, IPDPS, 2010.
[3] D. Ajwani, N. Sitchinava, N. Zeh, *Geometric algorithms for private-cache chip multiprocessors*. Manuscript.

MADALGO – Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation