NUMA-Aware Data Structure Design & Benchmarking

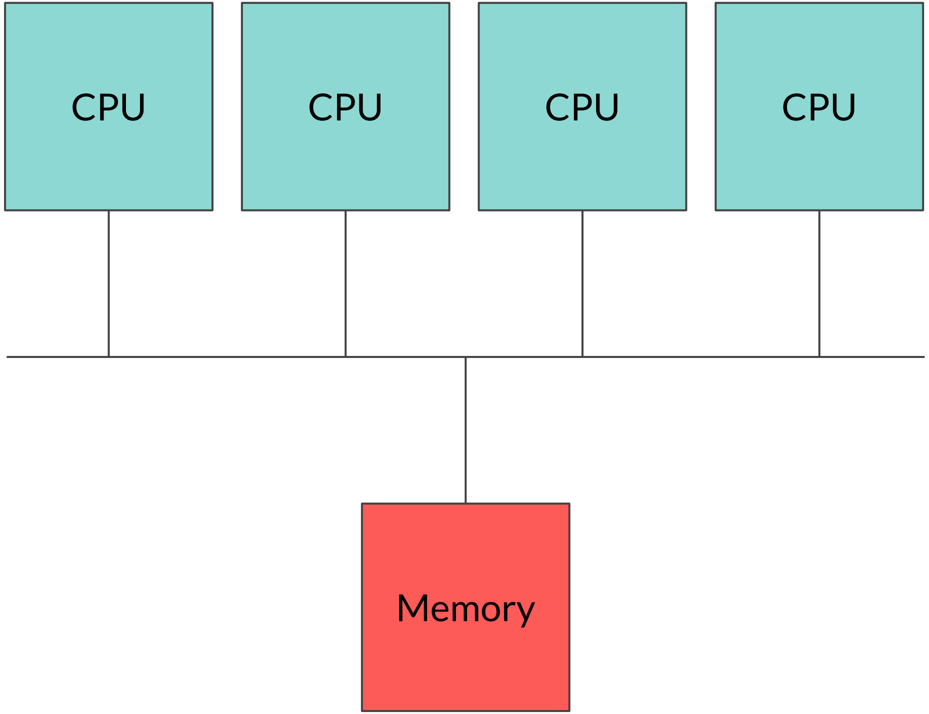
Presented by Yifan Kang
Mentored by Shangdi Yu and Prof. Julian Shun

What is NUMA

- Non-Uniform Memory Access
- What is Uniform Memory Access?

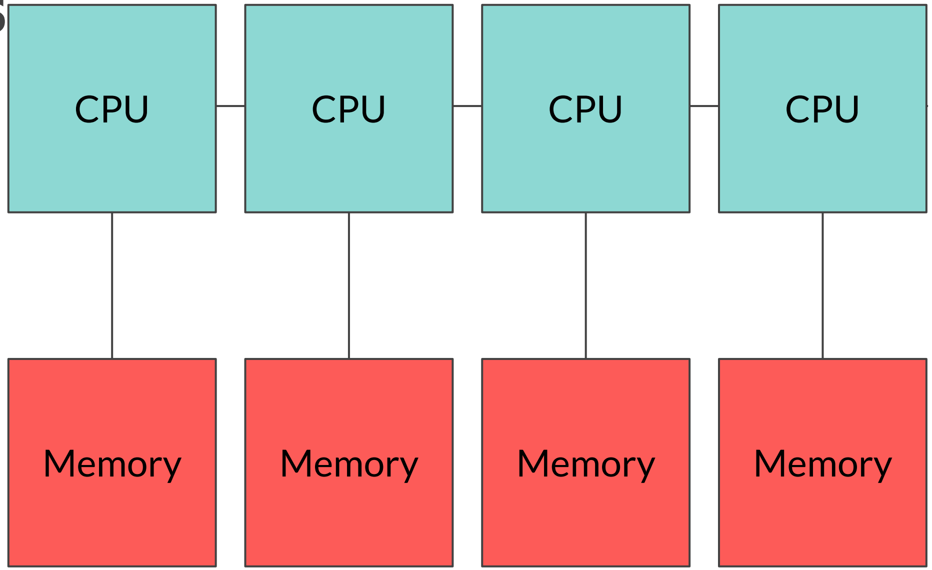
Uniform Memory Access

All the processors have the same access to all memory.



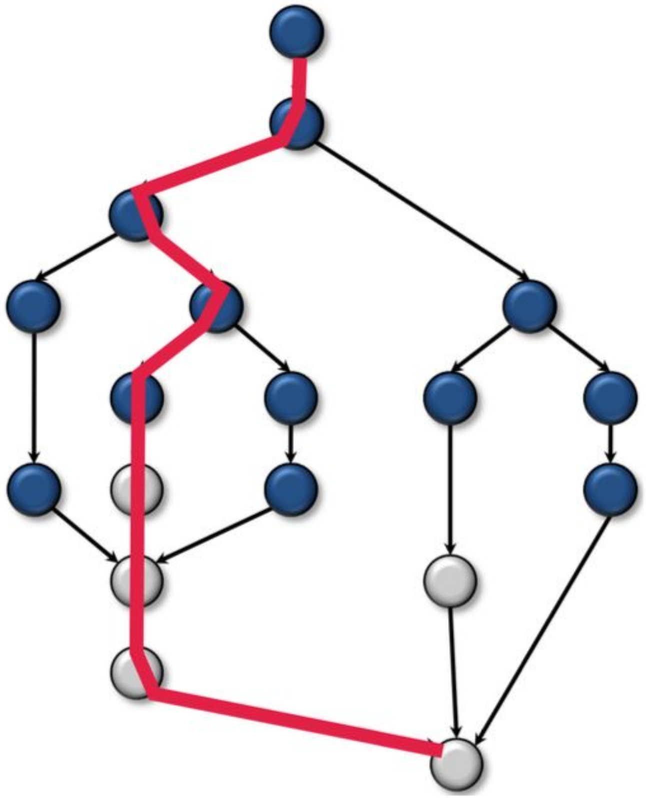
Non-Uniform Memory Access

Each processor has its own memory controller.



Parallel Programming Backgrounds

- Split a problem into smaller tasks
- Execute them in different processors **concurrently**
- Perform tasks more **efficiently**



Parallel Programming Backgrounds

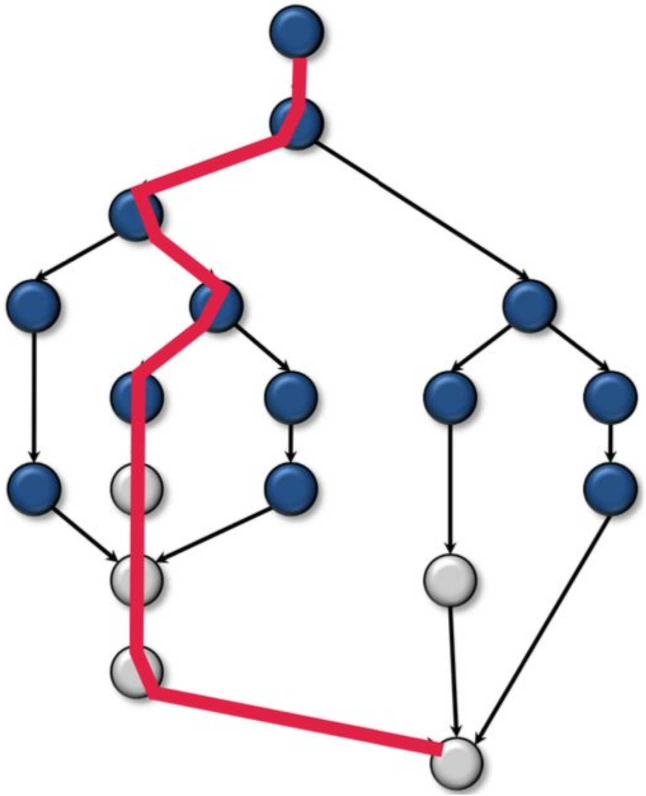
T_p = runtime with p processors

T_1 = work

T_∞ = span

Brent's Law:

$$T_p \leq T_\infty + \frac{T_1 - T_\infty}{p}$$



Example: Parallel prefix sum

Given an array $A_0, A_1, \ldots, A_{n-1}$, the prefix sum array S defined as:

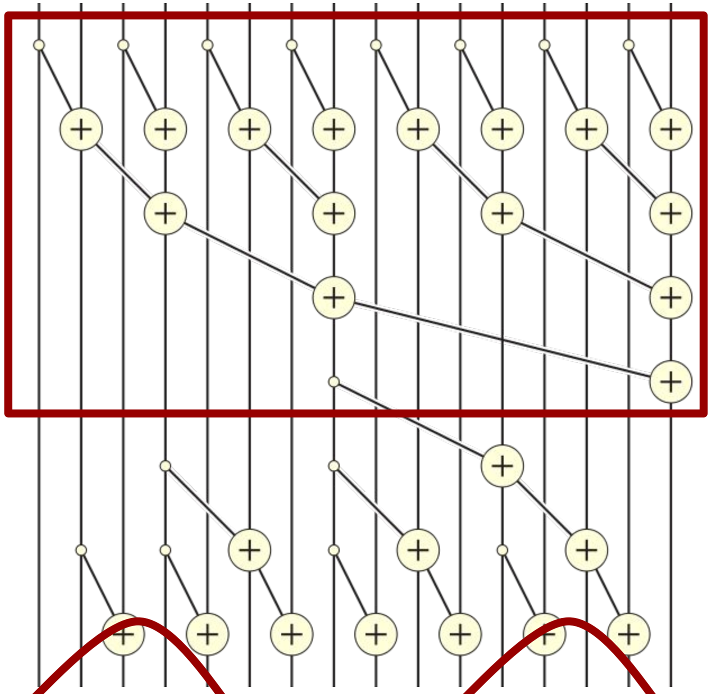
$$S_i = \sum_{k=0}^{i} A_k$$

`input`	6	4	16	10	16	14	2	8
`output`	6	10	26	36	52	66	68	76

Example: Parallel prefix sum

Core idea: Divide & Conquer

Step 1: Builds a tree of sums bottom-up

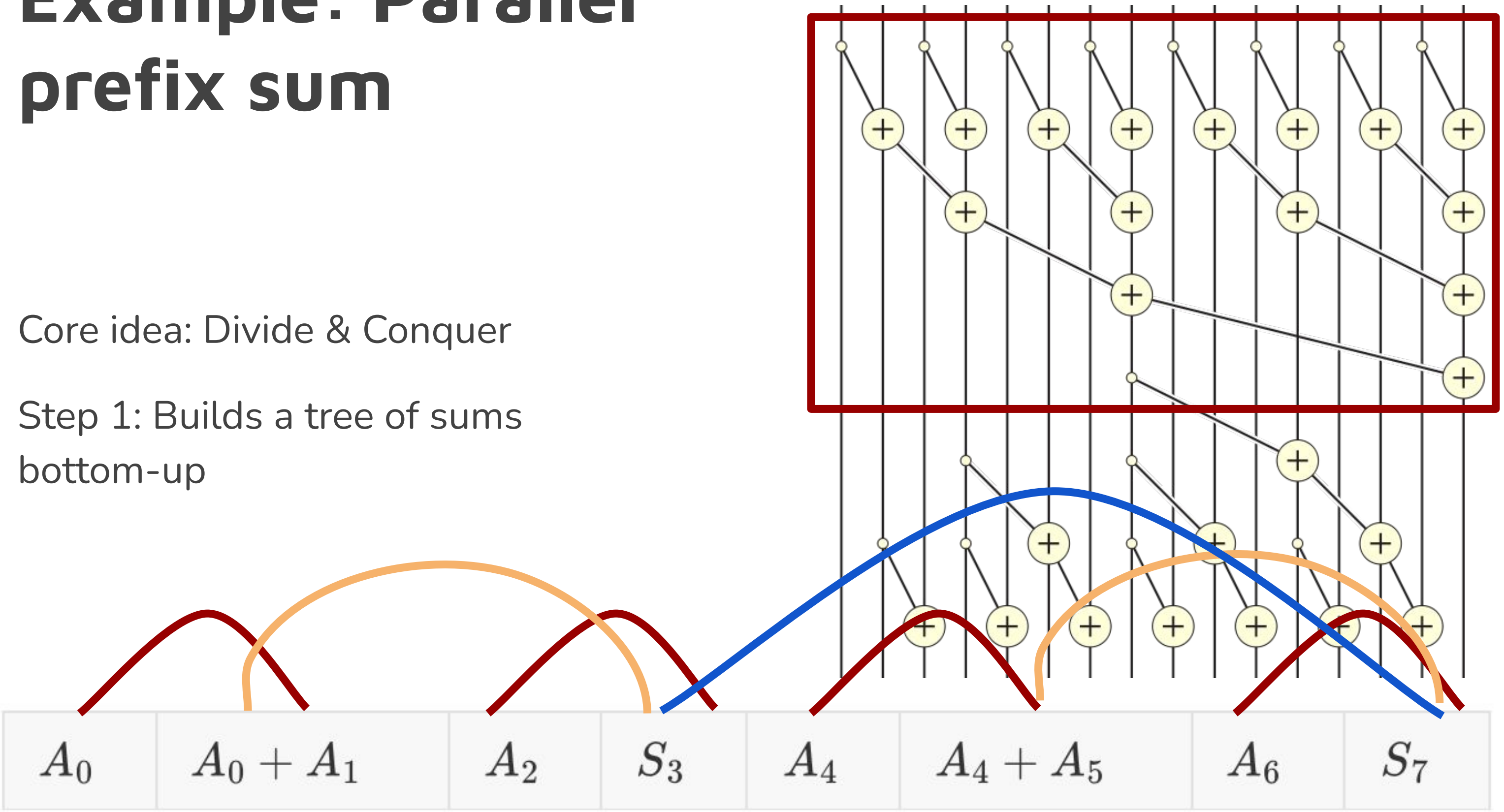


A_0	$A_0 + A_1$	A_2	$A_2 + A_3$	A_4	$A_4 + A_5$	A_6	$A_6 + A_7$

Example: Parallel prefix sum

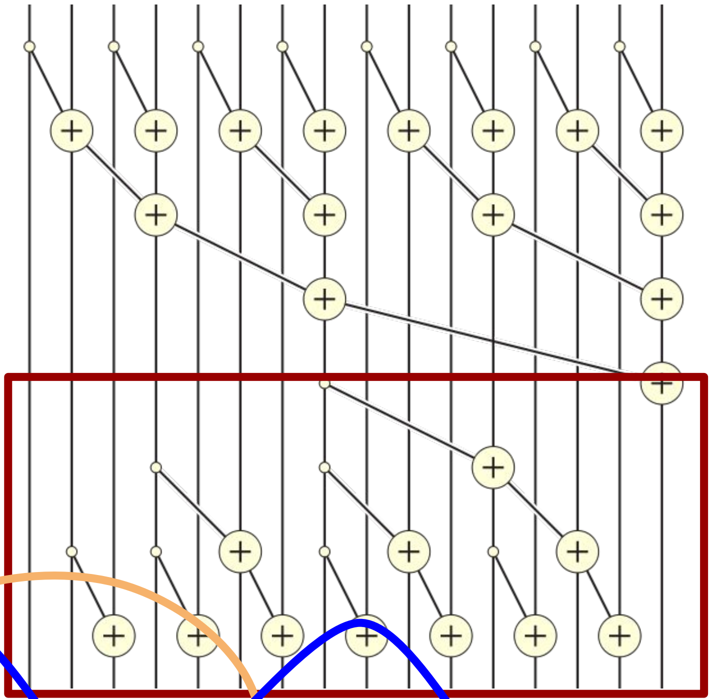
Core idea: Divide & Conquer

Step 1: Builds a tree of sums bottom-up



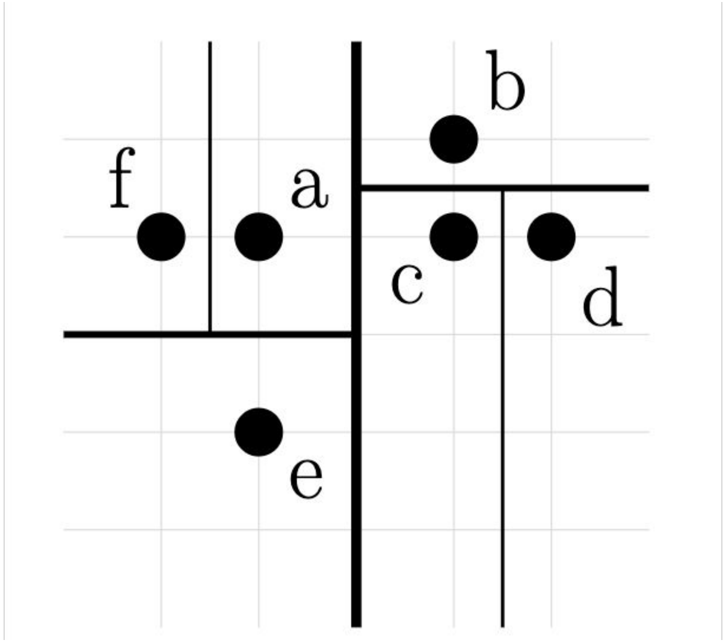
Example: Parallel prefix sum

Step 2: Traverses the tree top-down to compute prefixes

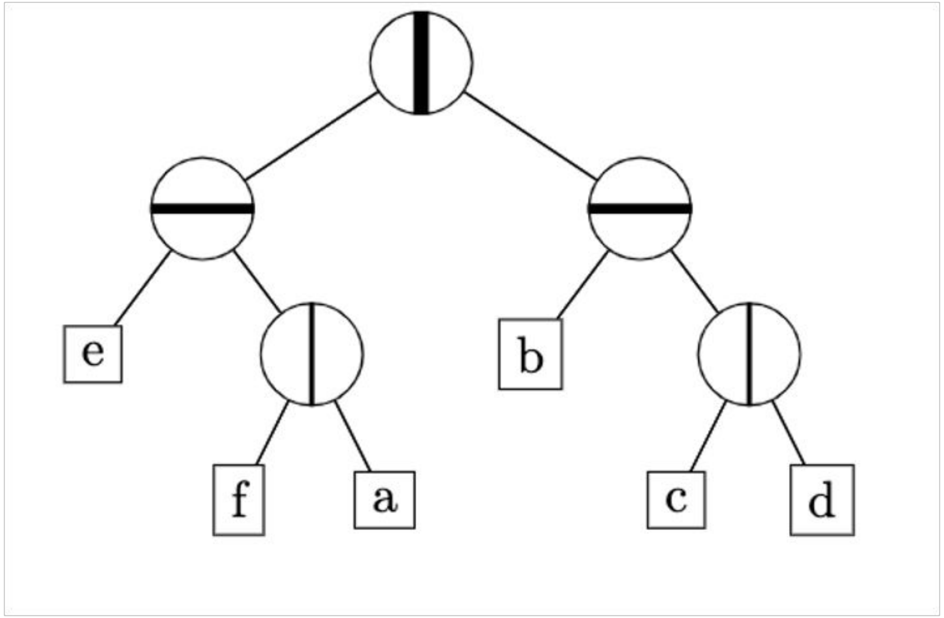


A_0	$A_0 + A_1$	A_2	S_3	A_4	$A_4 + A_5$	A_6	S_7

kd-tree



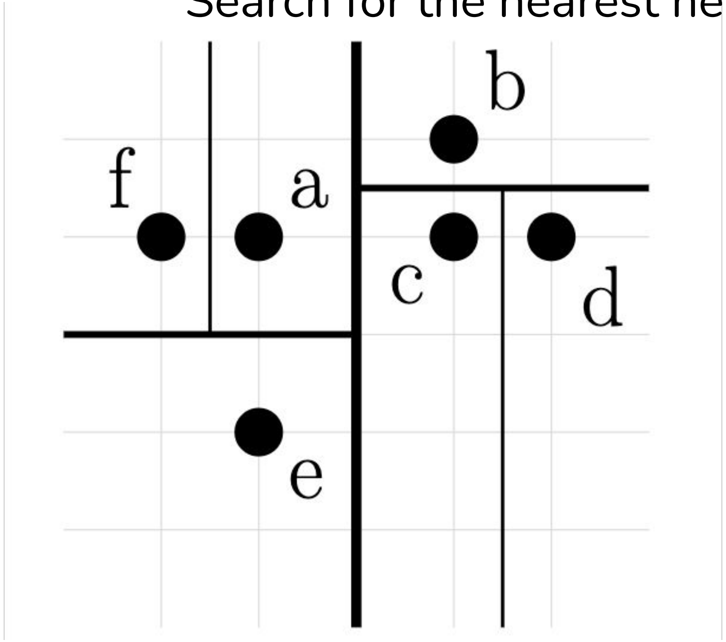
data set



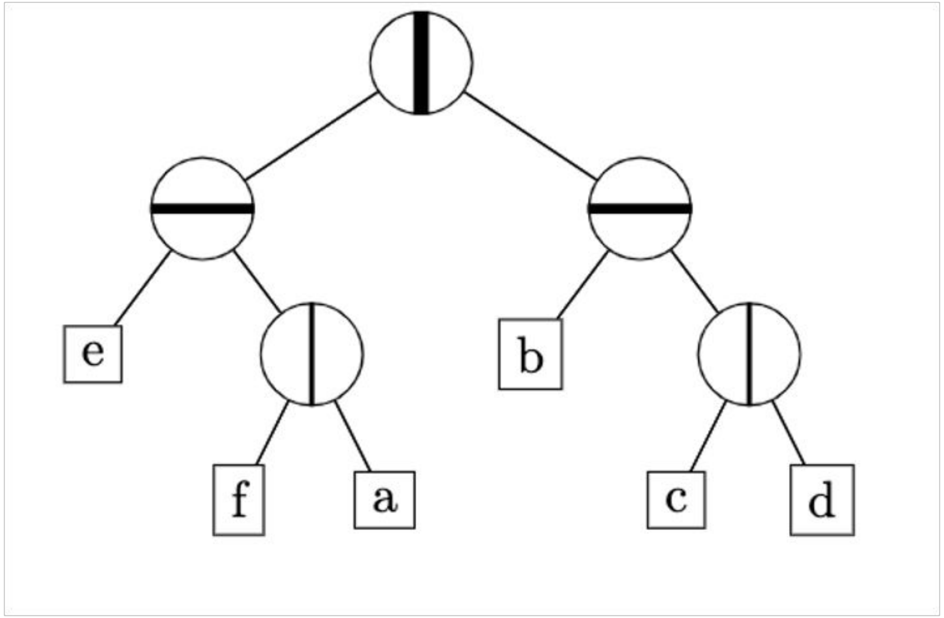
kd-tree

Nearest neighbor search

Search for the nearest neighbor of a.



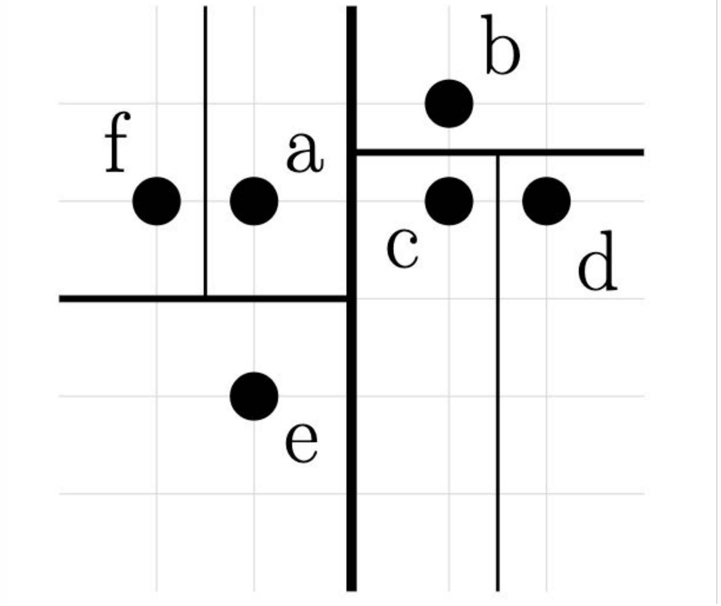
data set



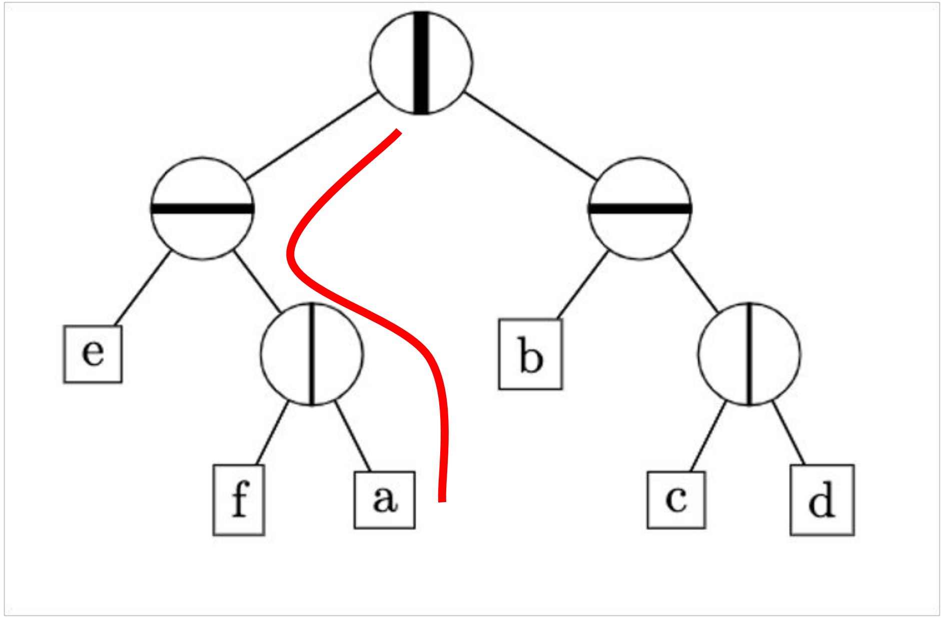
kd-tree

Nearest neighbor search

Step 1: Find the leaf node from the root.



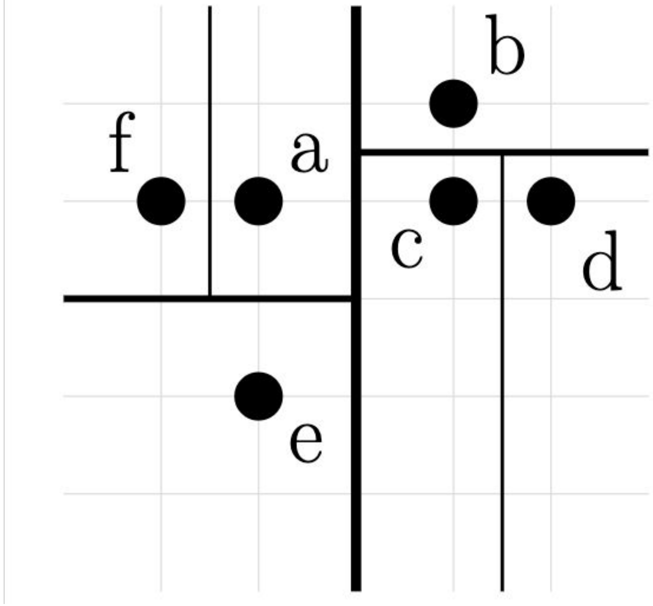
data set



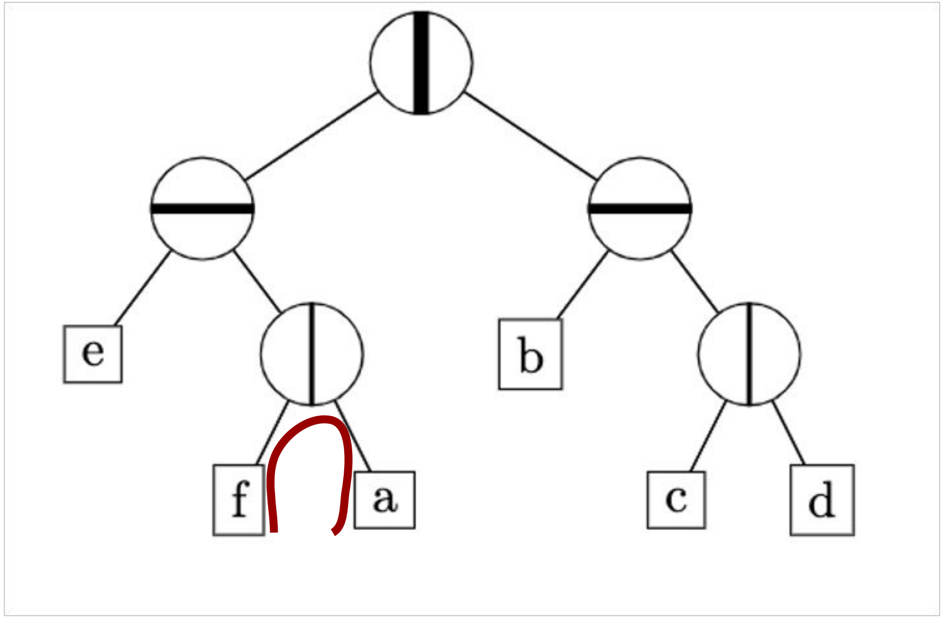
kd-tree

Nearest neighbor search

Step 2: Backtrack to find candidates.



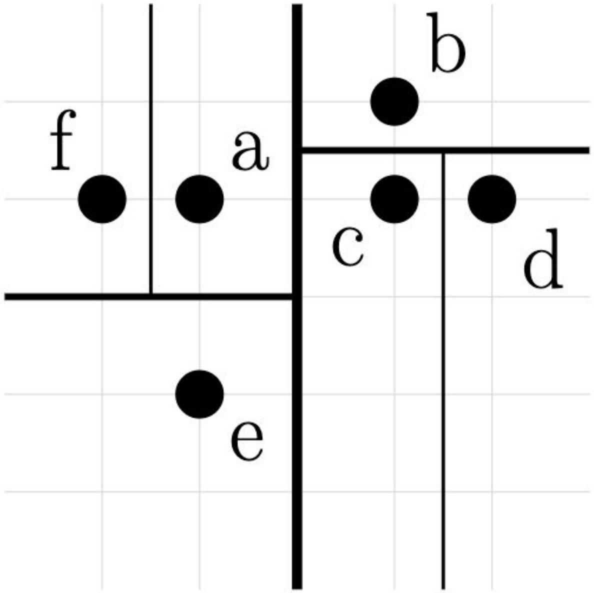
data set



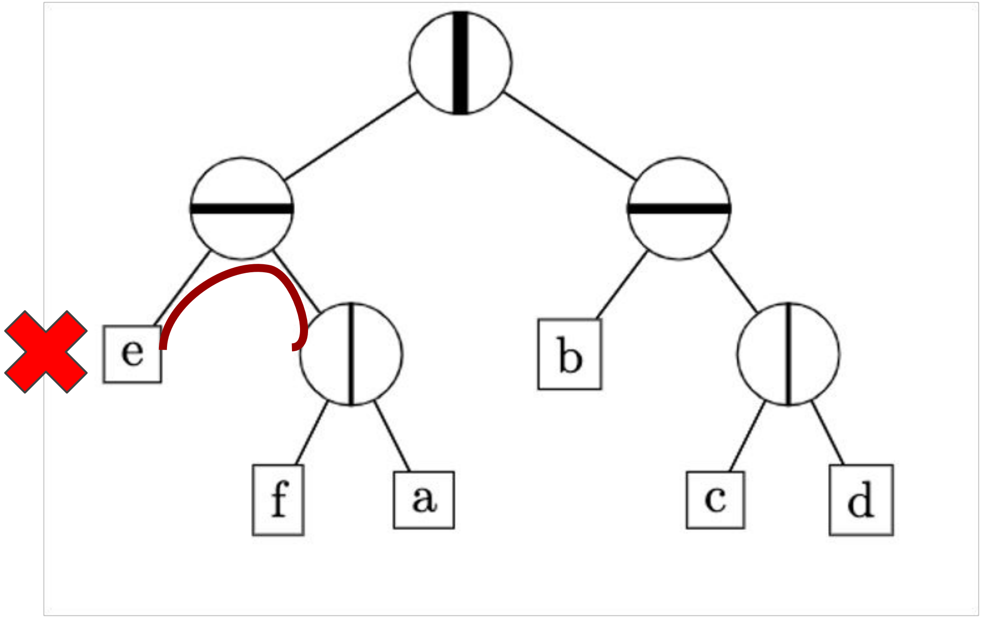
kd-tree

Nearest neighbor search

Step 2: Backtrack to find candidates.



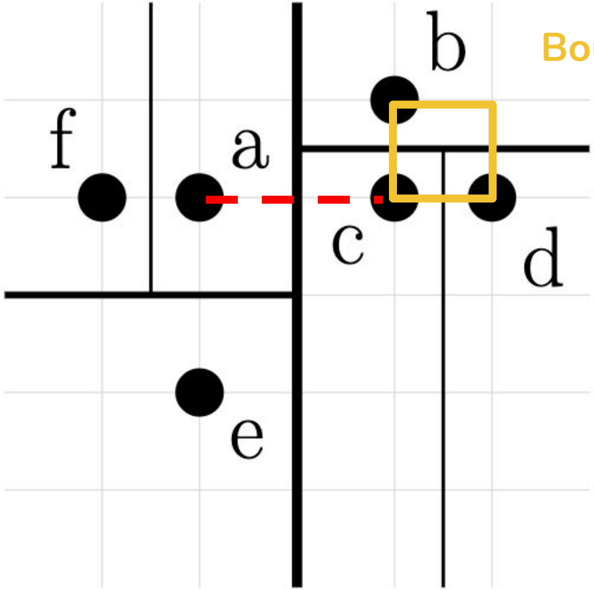
data set



kd-tree

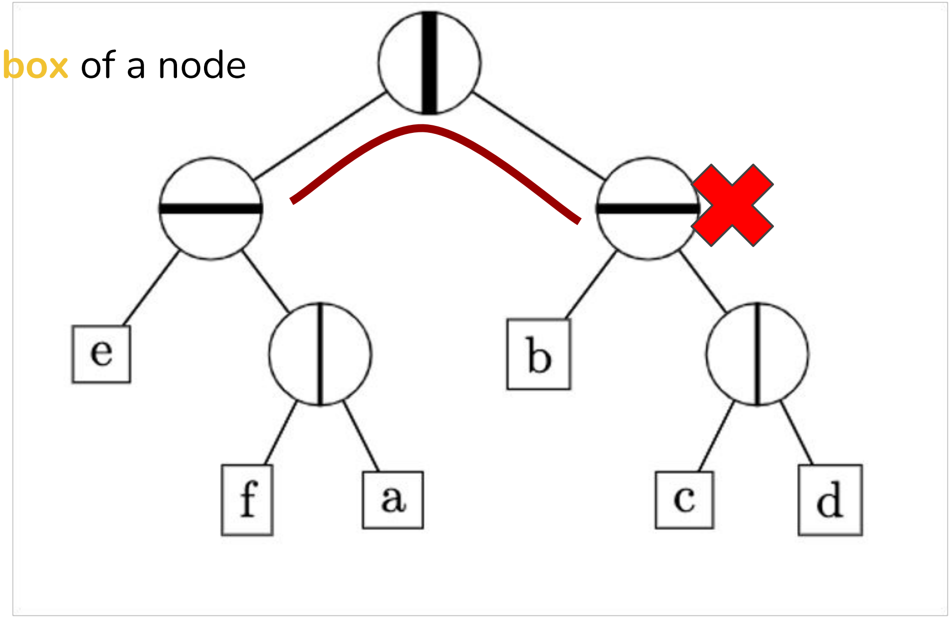
Nearest neighbor search

Step 2: Backtrack to find candidates.



Bounding box of a node

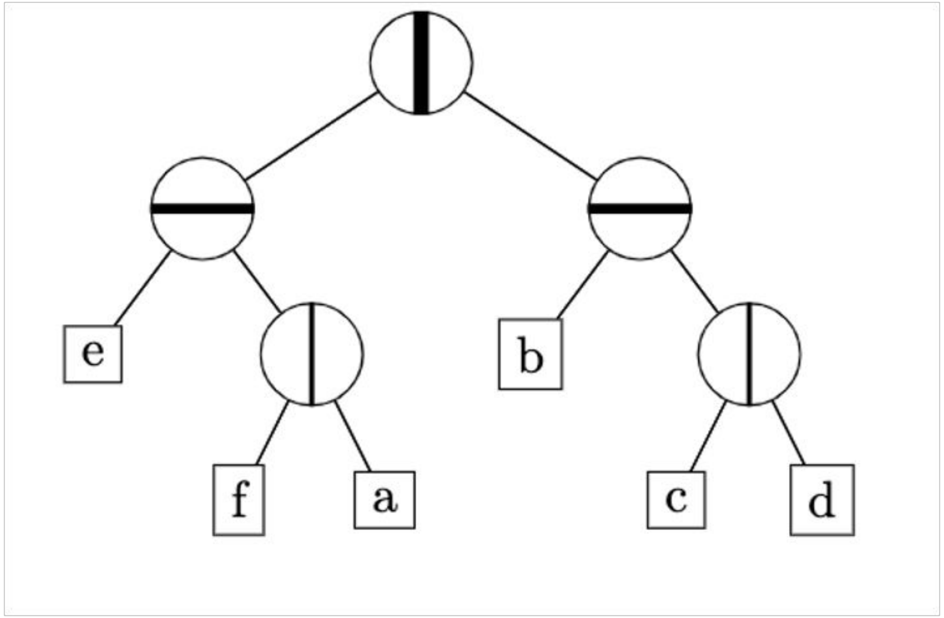
data set



kd-tree

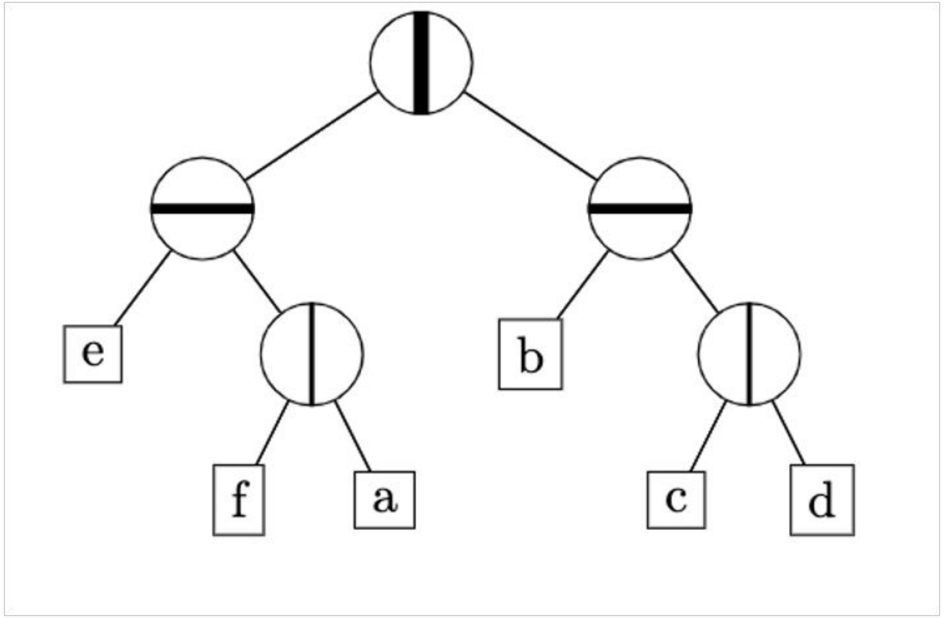
NUMA-aware kd-tree

How to make kd-tree NUMA-aware?



NUMA-aware kd-tree

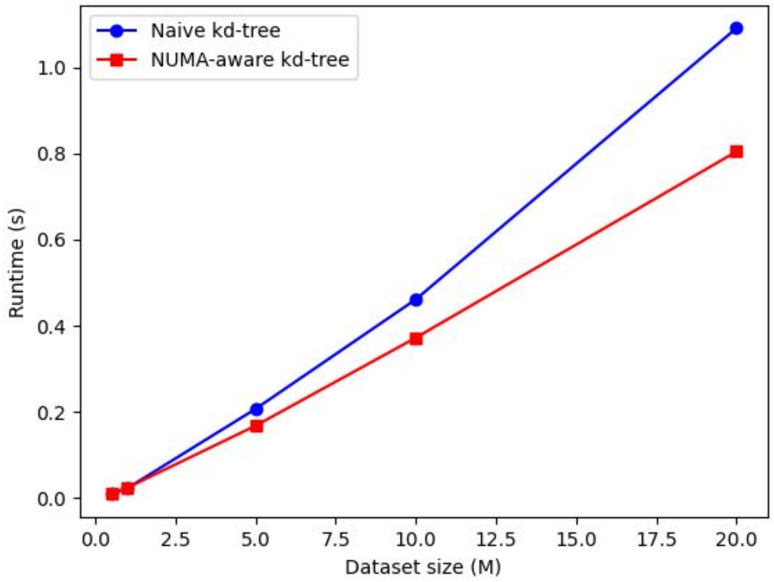
1. Split two parts into different NUMA nodes
2. Copy some subtrees in both nodes



Experiment Setup

- We performed our experiments in a virtual NUMA machine, c5.metal (96 vCPUs and 196 GiB), via Amazon Elastic Compute Cloud (Amazon EC2).
- Implemented with Parlaylib
- Random generated datasets

Runtime Comparison



Future Work

- Optimize dynamic kd-trees on NUMA machines.
- Real-world datasets
- Develop more NUMA data-structures such as interval trees and range trees.

Acknowledgements

- Shangdi Yu
- Prof. Julian Shun
- MIT PRIMES

Questions?

Image Sources

https://upload.wikimedia.org/wikipedia/commons/thumb/8/81/Prefix_sum_16.svg/300px-Prefix_sum_16.svg.png

https://www.cs.princeton.edu/courses/archive/fall13/cos326/lec/23-parallel-scan.pdf