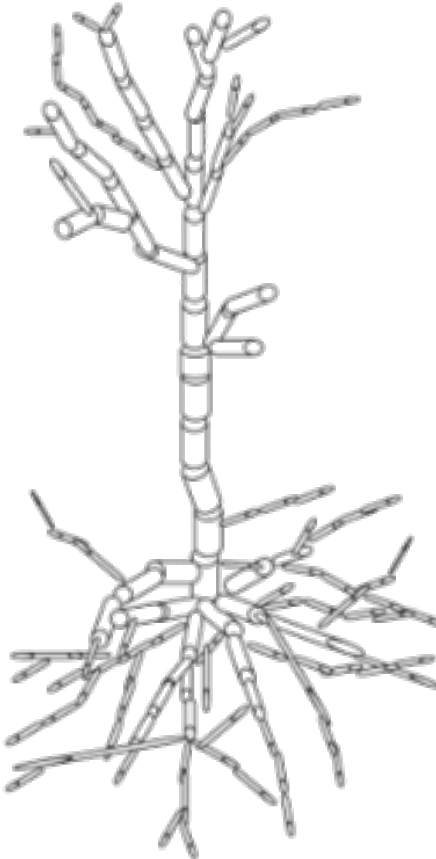


Enabling Breakthroughs Via Innovative Spatial Data Management

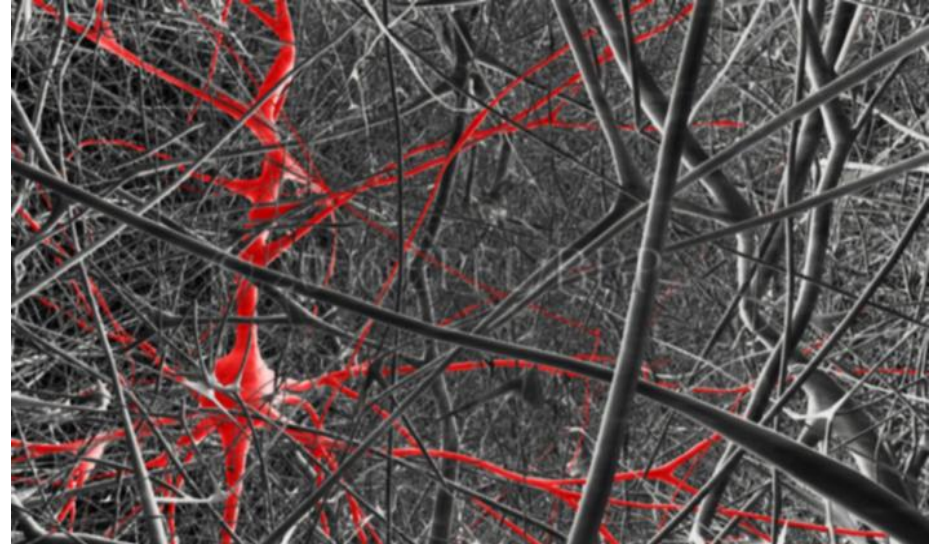
Farhan Tauheed, Sadegh Nobari, Laurynas Biveinis,
Thomas Heinis, Anastasia Ailamaki
Ecole Polytechnique Federale de Lausanne

Spatial Data Management Challenge

Model Visualization



Single Neuron
3D Model

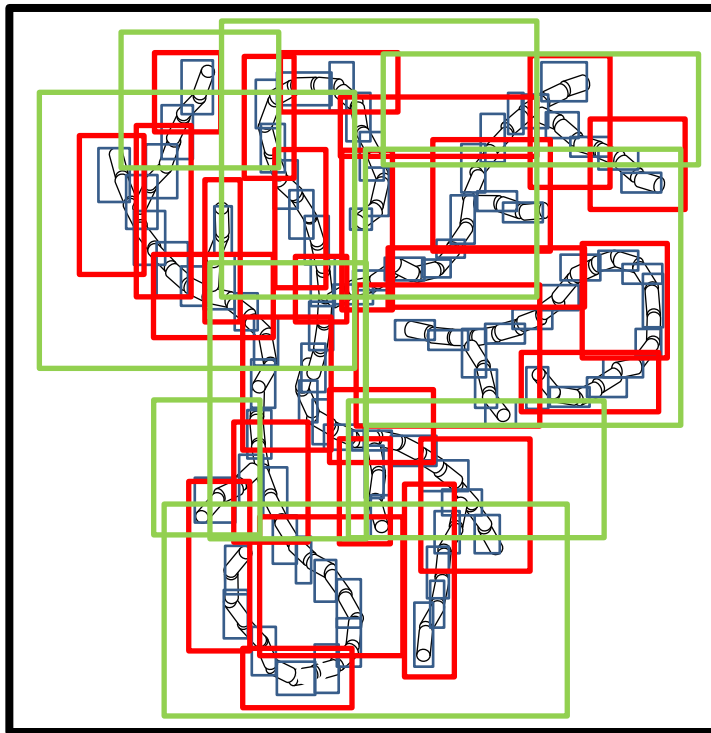
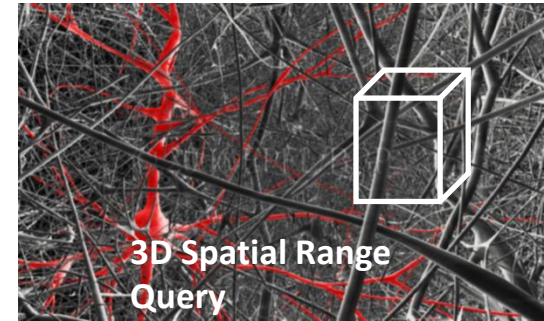


Model of 1 Million Neurons:

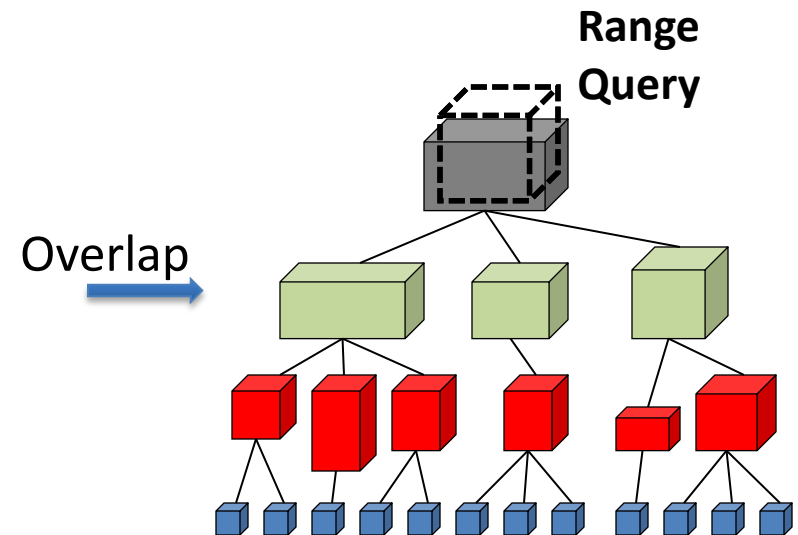
- Cylinder representation **270 GB**
- Mesh representation **5.8 TB**
- Will grow orders of magnitude!

Spatial Range Queries

State of the Art (R-Tree): Hierarchy of Minimum Bounding Rectangles (**MBR**)



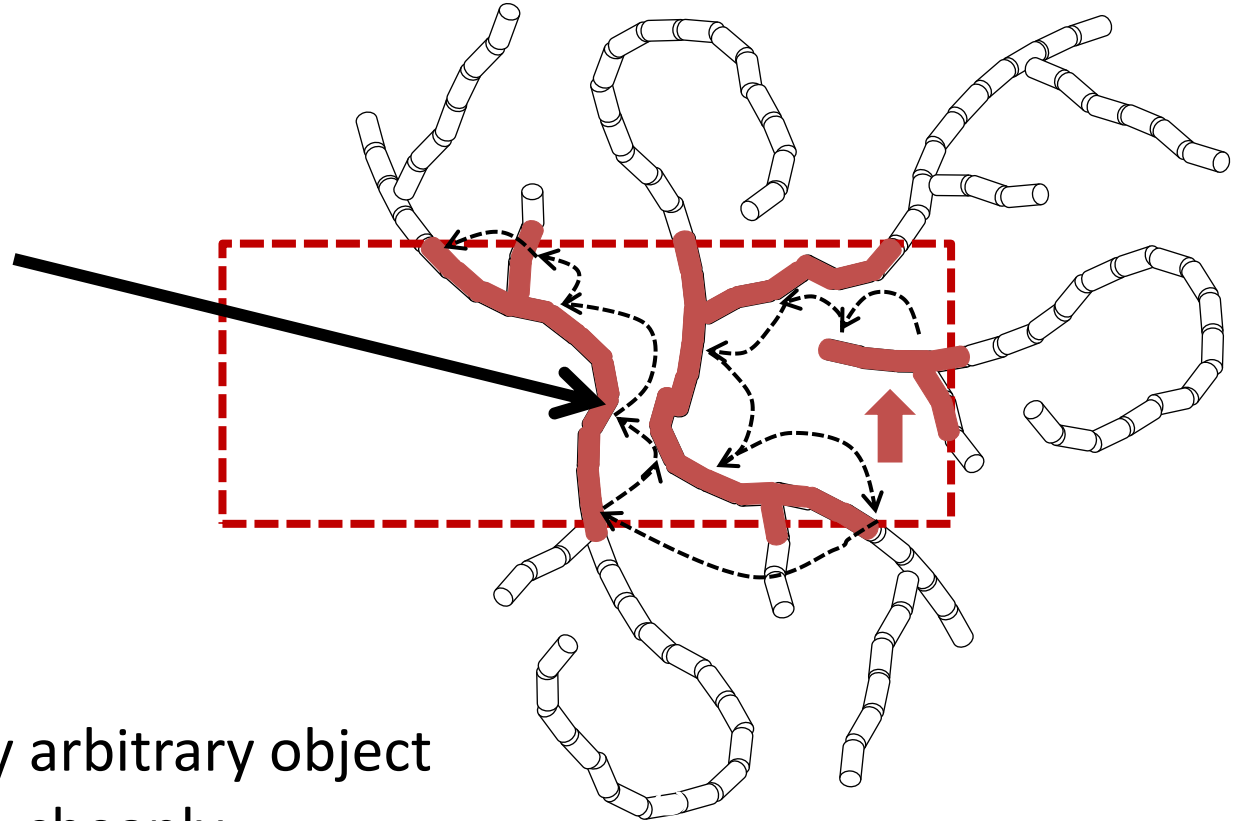
STEP2: Query Restriction



Overlap reduces performance, increases with data density

FLAT: A Two Phase Spatial Index

Add reachability information during index construction



Key Idea:

1) SEEDING: Find any arbitrary object inside the query region cheaply.

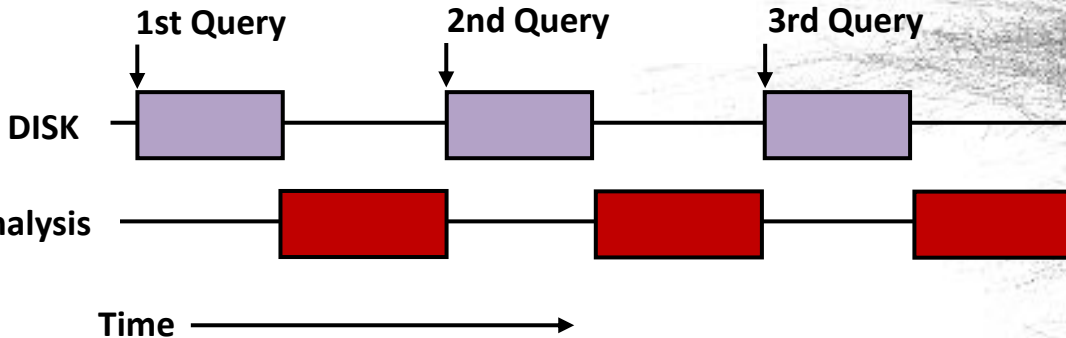
2) CRAWLING: Retrieve remaining

Querying overhead independent of data density
→ Speedup of 2-8x compared to current methods

Moving Range Queries

Model analysis: User issues series of consecutive queries

I/O expensive \rightarrow long wait



Use idle time \rightarrow predict next query & prefetch

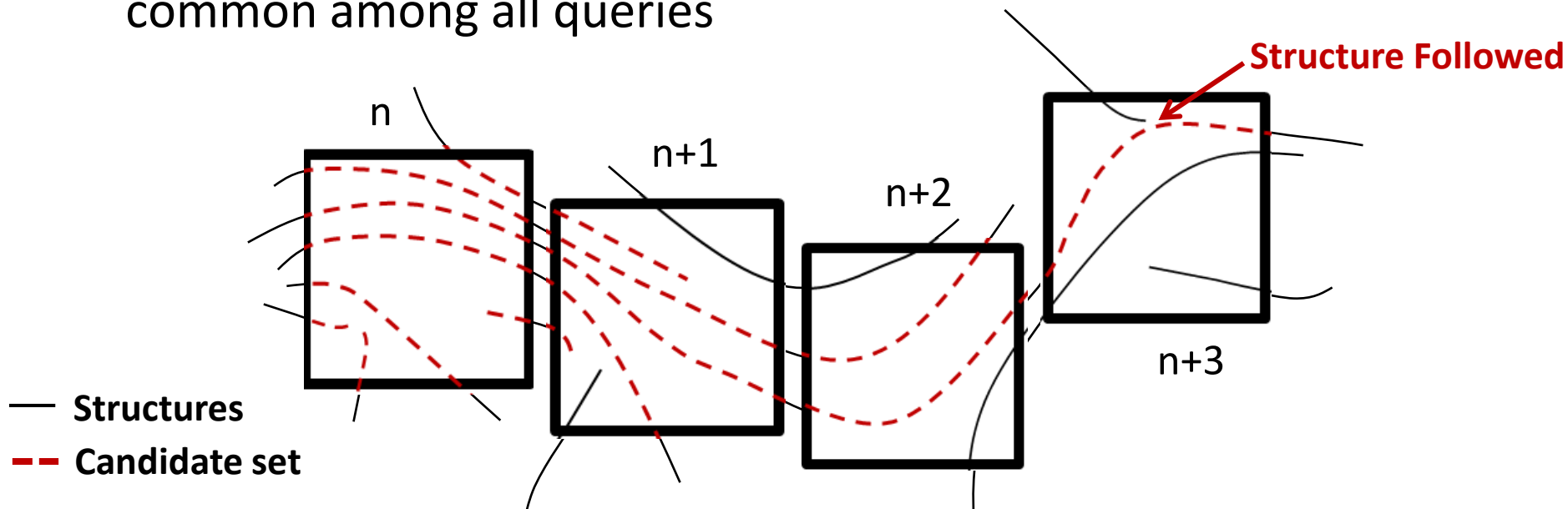
Series of spatial queries



Neuron Tissue Sample
(1692 neurons)

SCOUT: Content Aware Prefetching

- **Key Idea:** Use content of previous queries
- Structure followed can be any one of the many paths in the result
- **Insight:** Iterative candidate pruning, structure followed is common among all queries



SCOUT achieves between 72% and 91% prediction accuracy
→ Speedup of 4-15x compared to state of the art

Touch Detection – Spatial Join

Modeling synapses: place synapses where neurons intersect → **In-Memory Spatial Join**

State of the art join approaches:

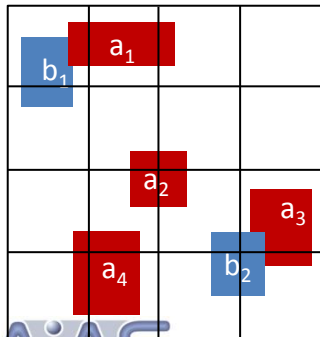
- Space-oriented Partitioning (**Partition Based Spatial Join**, Size Separation Spatial Join)
- Data-oriented Partitioning (**Synchronized R-Tree Traversal**, Indexed Nested Loop (R-Tree))

Dataset A



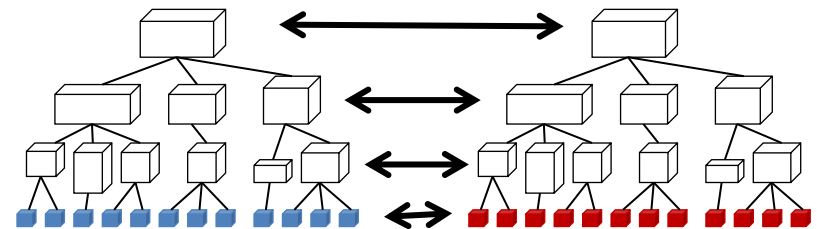
Dataset B

Space-oriented Partitioning: ex. **PBSM**



1	1	1	1	
1				
4	4	2	2	2 3
4	4	2	2	3

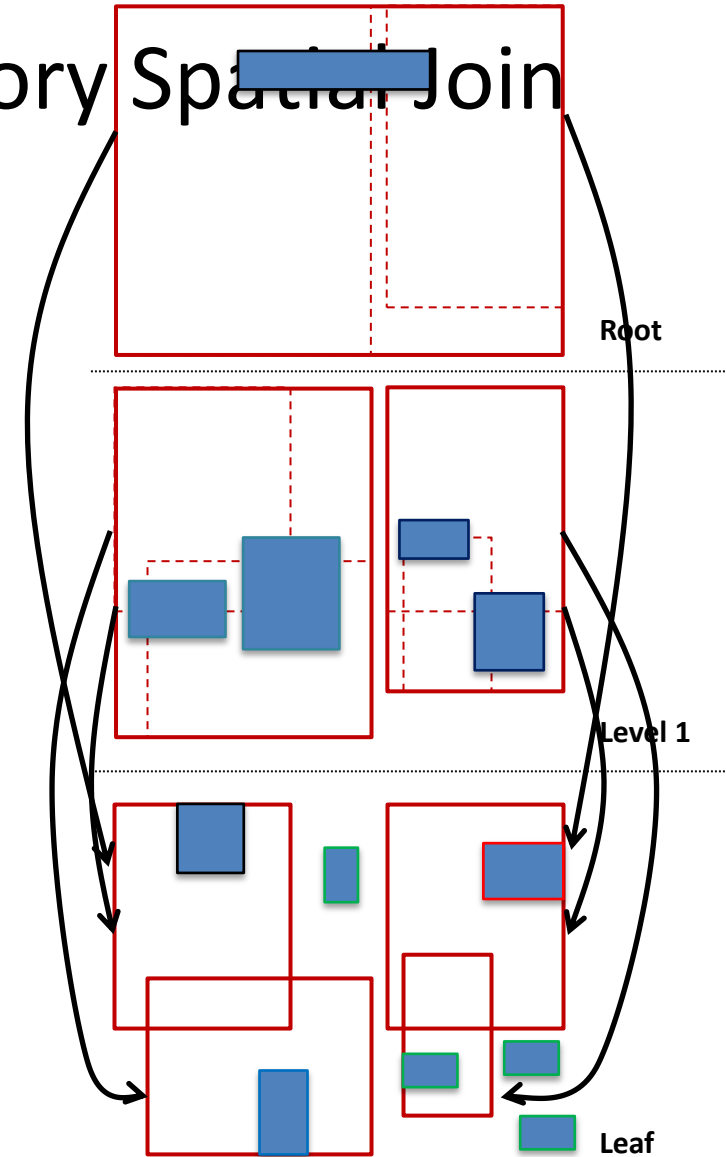
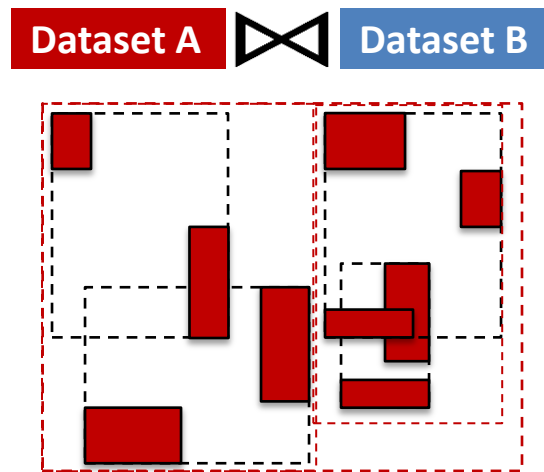
Data-oriented Partitioning: ex. **Sync R-Tree Join**



TOUCH: Efficient In-Memory Spatial Join

Key Idea: combine the best of both worlds:

- use data-oriented partitioning
- directly assign elements
- use filtering



Using TOUCH results in a one to two orders of magnitude speedup for the spatial join

Demo

- FLAT - Efficient Range Query Execution on Dense Data
- SCOUT – Content Aware Prefetching
- TOUCH - Efficient In-Memory Spatial Join

