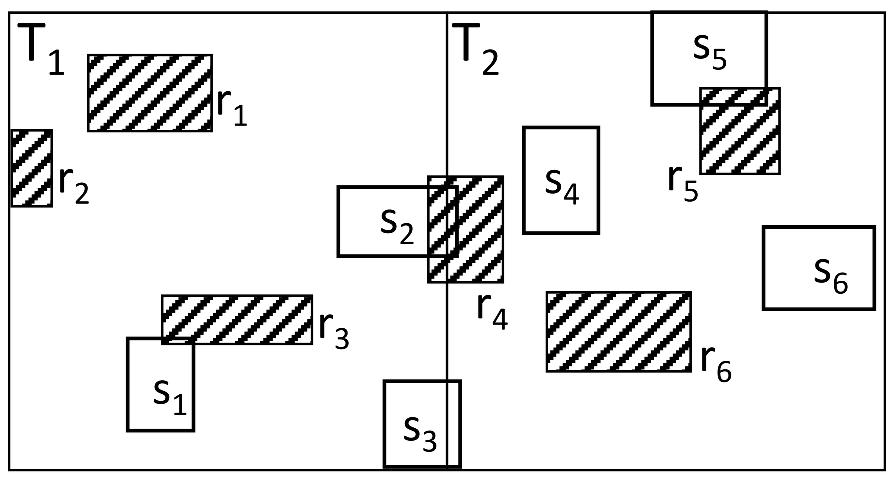
P28

PARALLEL IN-MEMORY EVALUATION OF SPATIAL JOINS

Dimitrios Tsitsigkos[1,3], Panagiotis Bouros[2], Nikos Mamoulis[3] and Manolis Terrovitis[1]
[1]Athena RC, Greece [2]Johannes Gutenberg University Mainz, Germany [3]University of Ioannina, Greece
{dtsitsigkos,mter}@imis.athena-innovation.gr, bouros@uni-mainz.de, nikos@cs.uoi.gr

Tuning PBSM

Partition-Based Spatial join (PBMS) [1]



Advantages

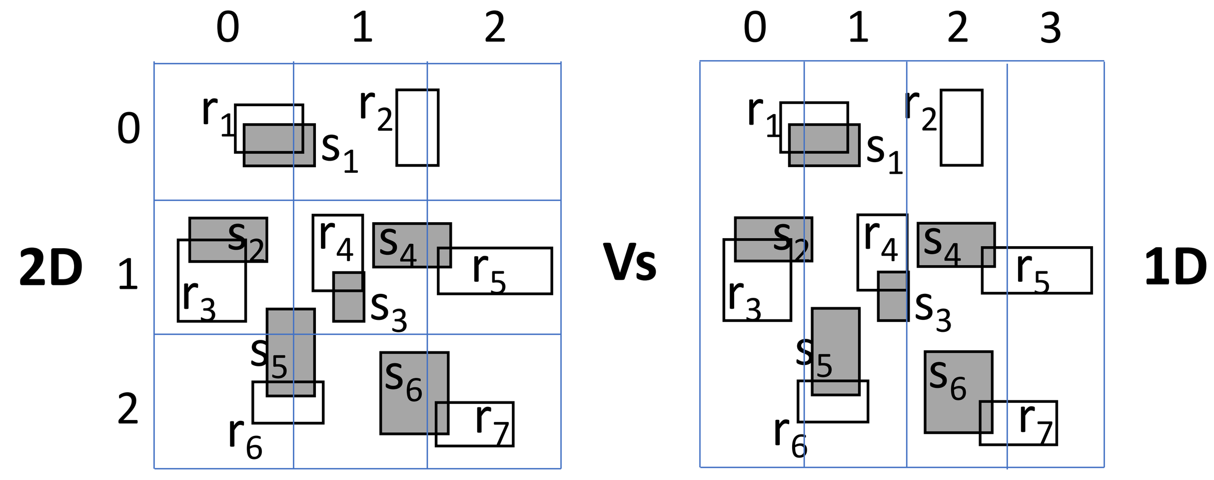
- ✓ Multi-assignment, single-join (MASJ)
- ✓ One independent join task per partition
- ✓ Suitable for dynamic data, no preprocessing
- ✓ Simple, easy to implement
- ✓ Adopted by all distributed spatial DMS

Challenges

- ❑ *What's next?* [2]
- ❑ Type and number of partitions
- ❑ In-memory evaluation
- ❑ Parallel processing on multi-core CPUs

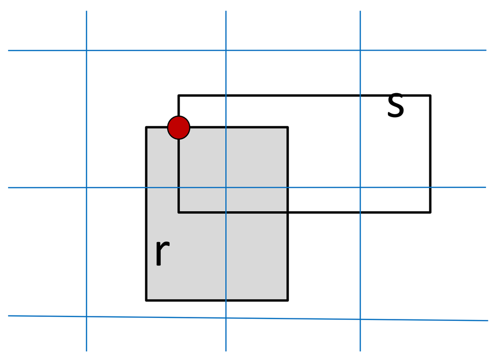
2D Versus 1D partitioning

- ❑ Traditionally a 2D grid splits space into tiles
- ❑ 1D partitioning into stripes



Duplicated results elimination

- ❑ Duplication test by *reference point* [3]



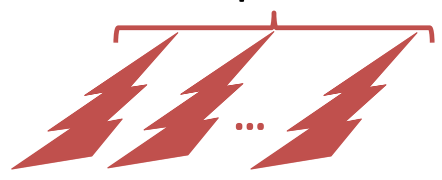
Sweeping axis

- ❑ Compute histogram statistics
- ❑ Divide x- and y-projections into buckets
- ❑ *Estimate* intersections per axis

$$I_T^x = \sum_{i=0}^{k} \{H_R^x[i] \cdot H_S^x[i]\} \quad I_T^y = \sum_{i=0}^{k} \{H_R^y[i] \cdot H_S^y[i]\}$$

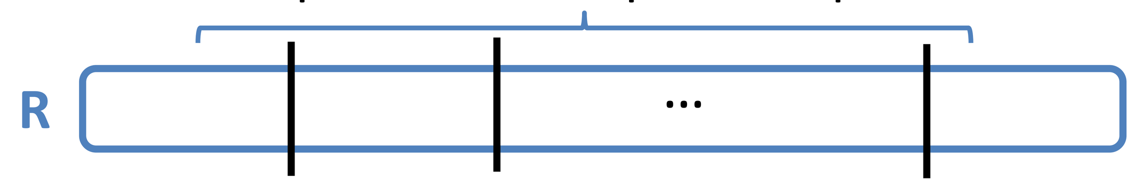
Parallel Processing

Initiate *m* parallel threads

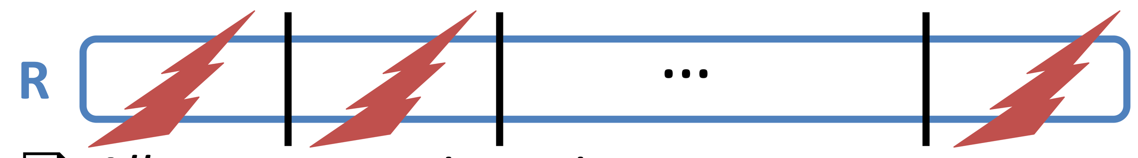


① **Partitioning phase**

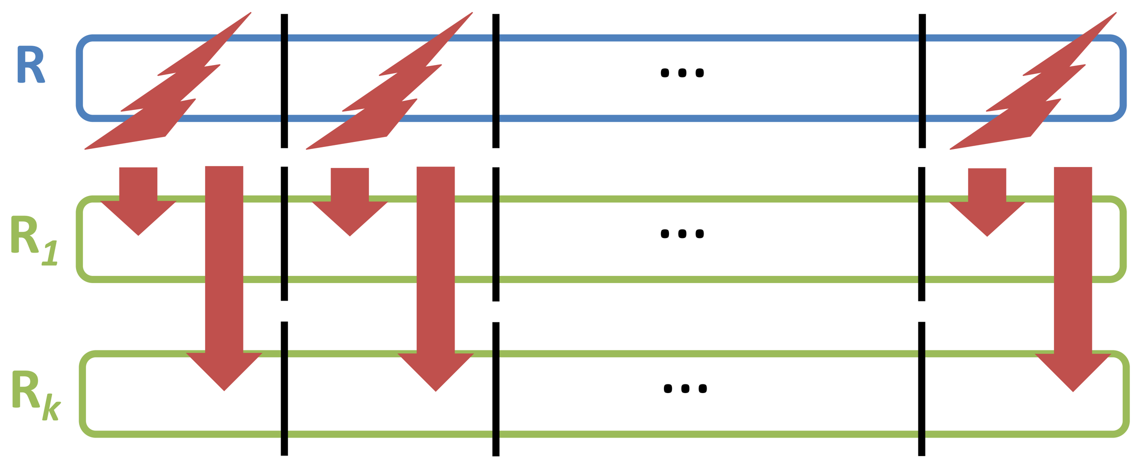
- ❑ Divide inputs into *m* equi-sized parts



- ❑ First pass: *compute* partitions size



- ❑ *Allocate* space in main memory
- ❑ Split *logically* every partition into *m* parts
- ❑ Second pass: *fill* *k* partitions



② **Joining phase**

- ❑ *Consume* tasks in round-robin manner

Experiments

Setup

- ❑ All data in main memory
- ❑ *Plane-sweep* join [4]
- ❑ OpenMP multi-threading
- ❑ Focus on filtering phase

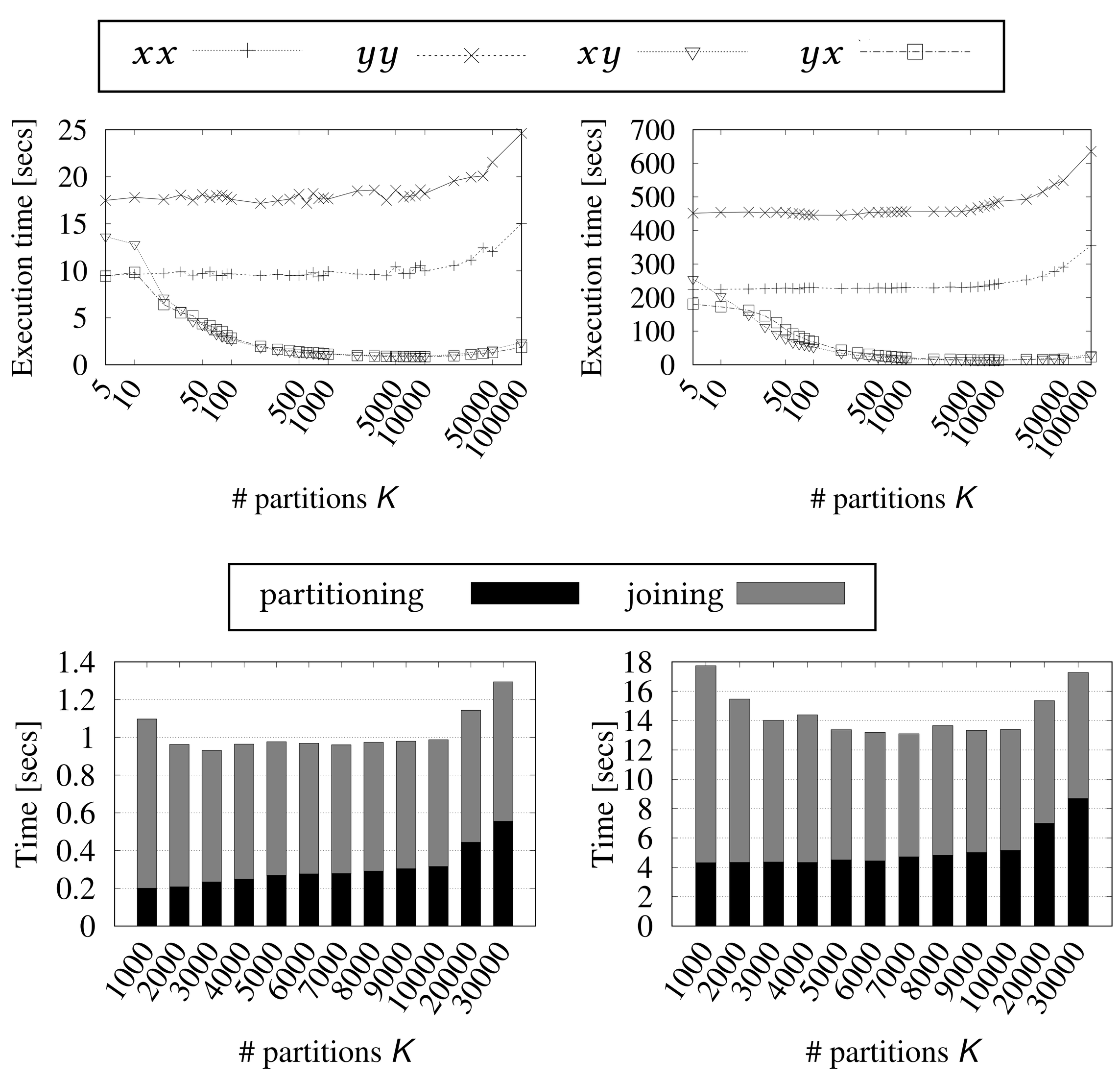
Datasets

source	dataset	alias	cardinality	avg. x-extent	avg. y-extent
Tiger 2015	AREAWATER	$T2$	2.3M	0.000007230	0.000022958
	EDGES	$T4$	70M	0.000006103	0.00001982
	LINEARWATER	$T5$	5.8M	0.000022243	0.000073195
	ROADS	$T8$	20M	0.000012538	0.000040672
OSM	Buildings	$O3$	115M	0.00000056	0.000000782
	Lakes	$O5$	8.4M	0.000021017	0.000028236
	Parks	$O6$	10M	0.000016544	0.000022294
	Roads	$O9$	72M	0.000010549	0.000016281

Selecting sweeping axis

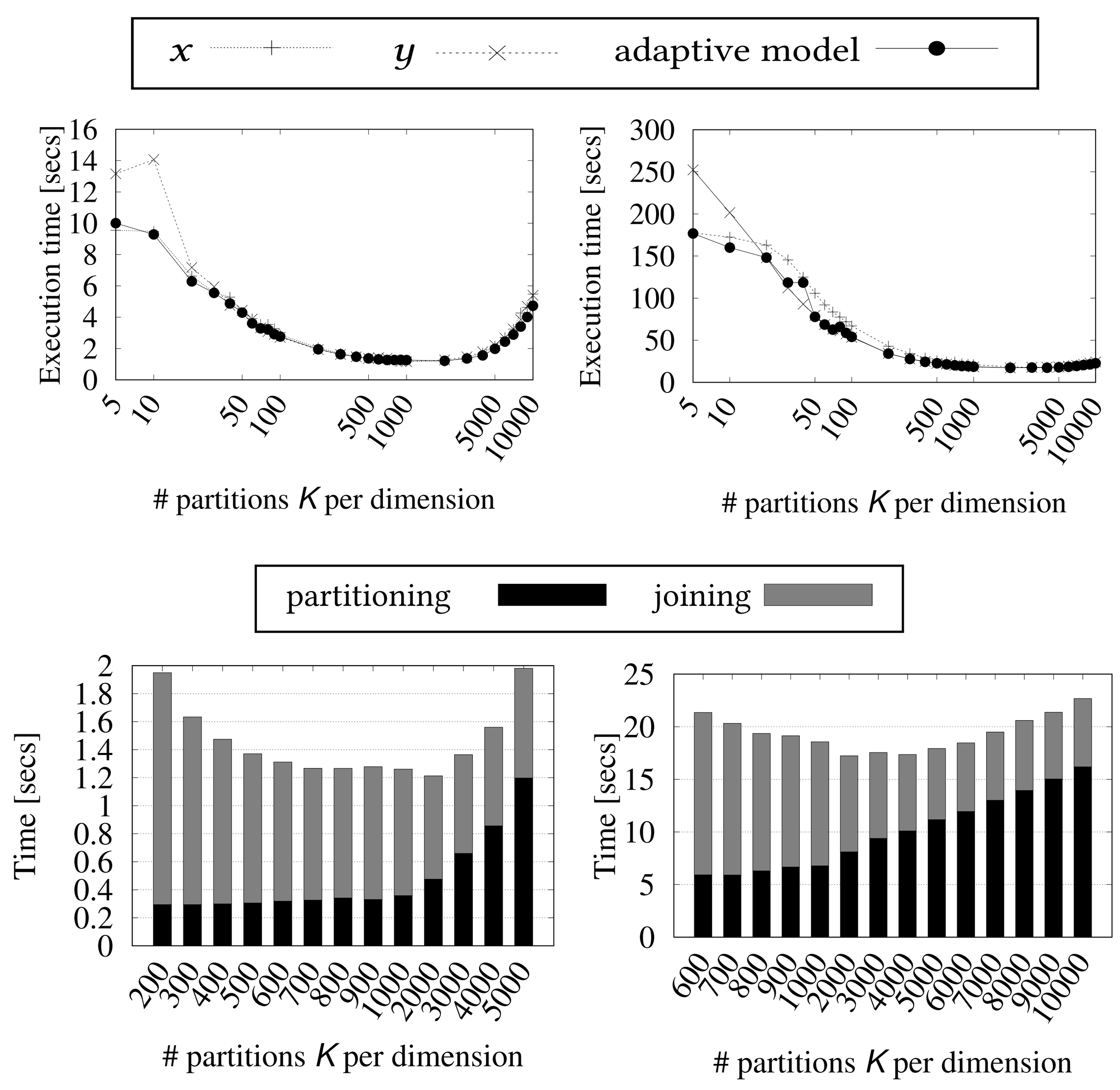
query	sweeping axis		adaptive model	
	x	y	I^x	I^y
$T2 \bowtie T5$	8.94s	16.96s	8,376	19,232
$T2 \bowtie T8$	24.52s	40.72s	8,895	18,660
$O5 \bowtie O6$	24.92s	66.06s	2,692	12,279
$O6 \bowtie O9$	216.88s	444.19s	3,989	11,510
$T4 \bowtie T8$	674.50s	1,360.92s	8,135	19,406
$O9 \bowtie O3$	926.14s	1,681.30s	4,535	11,529

Tuning 1D partitioning



$T2 \bowtie T5$ $O6 \bowtie O9$

Tuning 2D partitioning



$T2 \bowtie T5$ $O6 \bowtie O9$

1D Vs 2D partitioning

query	1D		2D	
	K	speedup	$K \times K$	speedup
$T2 \bowtie T5$	3000	9.6x	1000 × 1000	8.16x
$T2 \bowtie T8$	7000	10.67x	2000 × 2000	8.98x
$O5 \bowtie O6$	3000	8.62x	1000 × 1000	6.82x
$O6 \bowtie O9$	7000	16.56x	2000 × 2000	12.58x

Parallel processing (1D partitioning)

# threads	queries			
	$O5 \bowtie O6$	$O6 \bowtie O9$	$T4 \bowtie T8$	$O9 \bowtie O3$
1	2.98s	14.4s	20.1s	43.0s
5	0.75s	3.32s	4.34s	10.6s
10	0.46s	1.91s	2.47s	6.11s
15	0.38s	1.45s	1.85s	4.54s
20	0.32s	1.21s	1.64s	3.54s
25	0.29s	1.07s	1.42s	3.09s
30	0.28s	0.99s	1.36s	2.89s
35	0.27s	0.96s	1.27s	2.72s
40	0.27s	0.91s	1.21s	2.72s

References

[1] J. M. Patel and D. J. DeWitt.. *Partition Based Spatial-Merge Join*. In ACM SIGMOD, 1996.
[2] P. Bouros and N. Mamoulis. *Spatial Joins: What's Next?*. SIGSPATIAL Special 11(1), 2019.
[3] J.-P. Dittrich and B. Seeger. *Data Redundancy and Duplicate Detection in Spatial Join Processing*. IEEE ICDE, 2000.
[4] T. Brinkhoff, H.-P. Kriegel and B. Seeger. *Efficient Processing of Spatial Joins Using R-tree*. In ACM SIGMOD, 1993.

27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL), Chicago, IL, USA, November 5-8, 2019