

PARALLEL QUERY ENGINE FOR INTERACTIVE SPATIOTEMPORAL ANALYSIS

❖ Mihir V. Sathe

Dept. of Computer Science

❖ Yao-Yi Chiang

Spatial Sciences Institute

❖ Craig A. Knoblock

Information Sciences Institute

❖ Aaron Harris

Dept. of Mechanical Engineering

UNIVERSITY OF SOUTHERN CALIFORNIA

What is this?

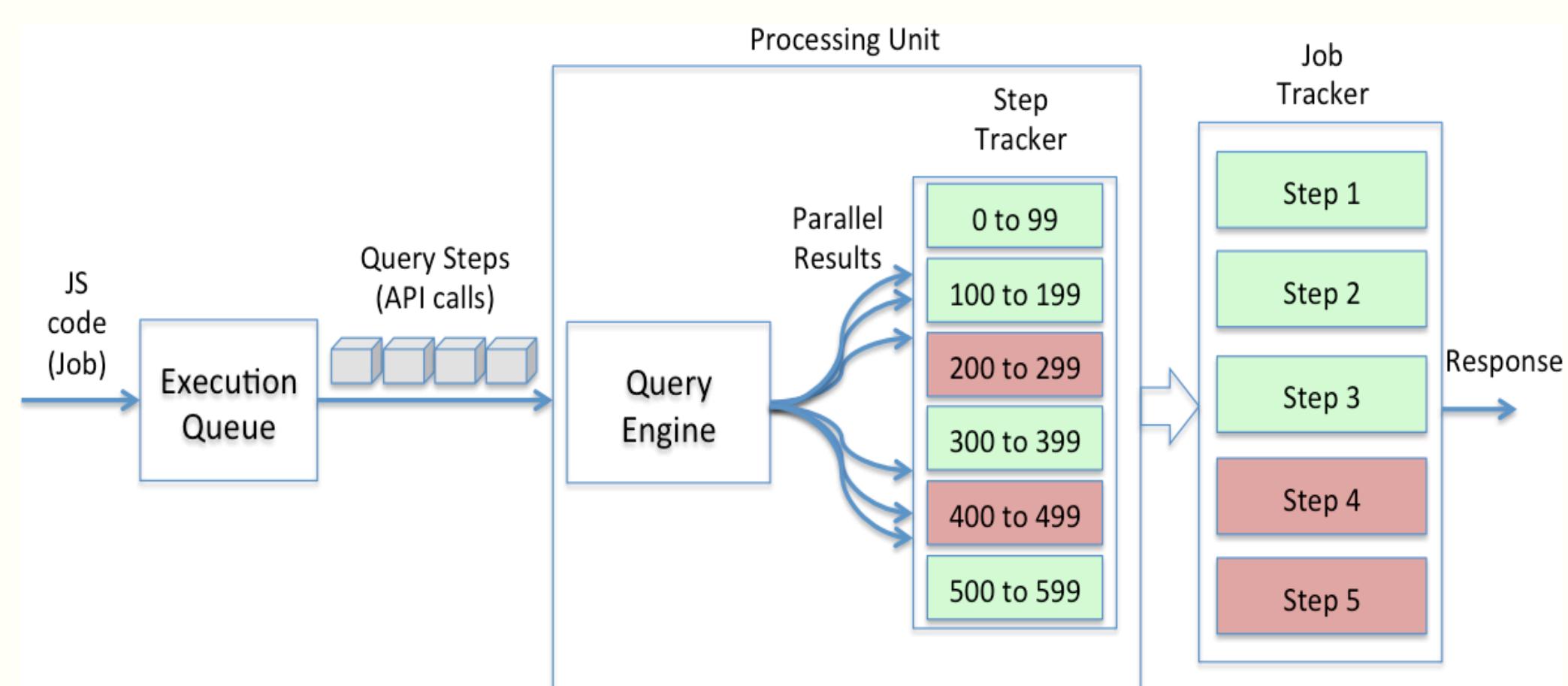
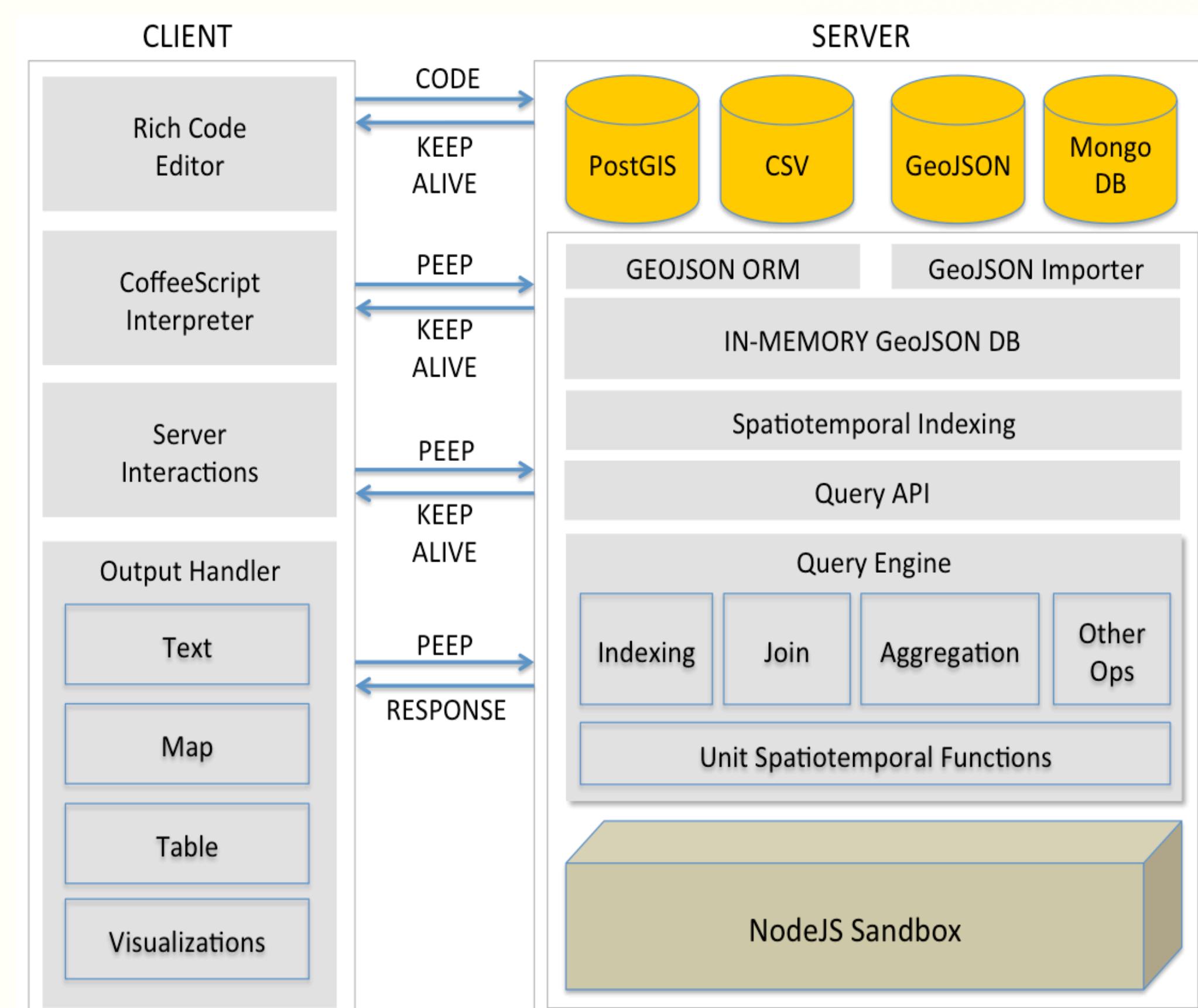
- Integrated scripting tool for flexible and fast prototyping
- Import, clean, transform, visualize and export data with simple scripts
- Go back and forth with your data for incremental research. Save and restore previous states.
- Machine learning, NLP, custom visualizations... anything is possible with an extension!



Why here?

- Highly optimized for spatiotemporal operations
- In built in-memory indexing with R-trees (space) and interval trees (time)
- Speedy execution of spatiotemporal operations like joins, bucketing and aggregations
- Visualize your geospatial data on maps with one line code. Create your own visualizations.

Architecture



Speed-Up

- Spatiotemporal operations execute in parallel on multiple cores
- This speed-up combines with in-memory spatiotemporal indexing to provide low response times
- Parallelization is optimized according to size of data and number of cores
- We have process level caching of data and indexes to avoid repeated loading of datasets making efficient use of available memory to save time

Experiment

- Identifying spatiotemporally co-occurring crime patterns in cities Compton and West Hollywood in Los Angeles county
- We joined the crimes with all cities in LA county to compare performance in spatial joins with postGIS
- We searched inside every crime's bounding box to find spatially nearby crimes and then filtered them temporally to get series of co-occurring crimes
- We then used apriori algorithm to find the set of crimes that occur most frequently together in both cities

Association Rule (Compton)	Support
Assault, Aggravated Assault, Vehicle/Boating Laws	23.86%
Theft/Larceny, Vehicle break-in/Theft, Assault	22.13%
Assault, Grand Theft Auto, Motor Vehicle Theft	20.61%

Result

- Our tool in parallel mode showed a major improvement over brute force and indexed postGIS spatial join
 - We also outperformed spatial R in executing the co-occurrence algorithm (see the charts)
 - We do not claim to be a replacement for these tools; but a smart and efficient way to perform advanced spatiotemporal research and prototyping on reasonable sized data
- Runtime Comparison of City \bowtie Crime

A bar chart comparing the runtime of different join methods. The Y-axis represents Runtime in milliseconds (0 to 100,000). The X-axis lists the methods: Brute Force, Brute force with MBR, PostGIS, Our Tool (Single Thread), and Our Tool (Parallel: 5 threads). The bars show decreasing runtime from left to right.

Method	Runtime (milliseconds)
Brute Force	87200
Brute force with MBR	15060
PostGIS	4207
Our Tool (Single Thread)	4755
Our Tool (Parallel: 5 threads)	891

Runtime Comparison for Series Extraction

A bar chart comparing the runtime of Our Tool and R for series extraction in Compton and West Hollywood. The Y-axis represents Runtime in milliseconds (0 to 35,000). The X-axis lists the cities: Compton and West Hollywood. For each city, there are two bars: a blue bar for Our Tool and a red bar for R.

City	Our Tool (ms)	R (ms)
Compton	21255	32601
West Hollywood	7254	11952
- MORE INFORMATION AND UPDATES ON MIHIRSATHE.COM/MAGMA.