Analysis of Geospatial Data Loading

Aske Wachs and **Eleni Tzirita Zacharatou**

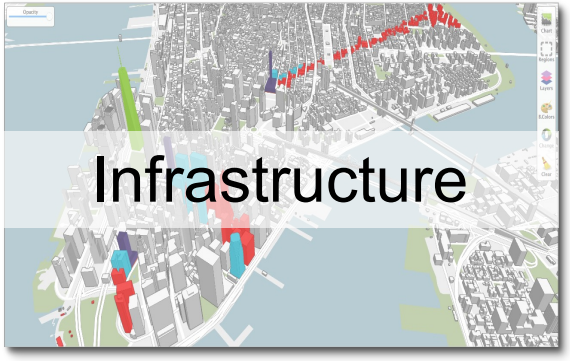
IT University of Copenhagen

DBTest@SIGMOD
June 9, 2024

IT UNIVERSITY OF COPENHAGEN

DASYA
Data-Intensive Systems and Applications

Spatial Data is Ubiquitous



Diverse Spatial Data Landscape

Multiple libraries & systems

GDAL/OGR

DuckDB

ArcGIS

GeoPandas

ORACLE SPATIAL

Multiple file formats

GeoJSON

GeoParquet

Geospatial data in Parquet

FGB

Impact of system and file format on loading performance

Benchmarks for Spatial Data Processing

- Analyzing spatial data efficiently is critical
  - First step: loading spatial data
  - Quantify & compare efficiency → benchmarks

- Existing benchmarks
  - Spatial joins
  - Topological relations (e.g., intersections)
  - Exploratory analytics workloads
  - Writing spatial data (anecdotal)

Suprio Ray, Bogdan Simion, and Angela Demke Brown. ”Jackpine: A benchmark to evaluate spatial database performance”, ICDE 2011.

Yaming Zhang and Ahmed Eldawy. “Evaluating computational geometry libraries for big spatial data exploration”, GeoRich 2020.

Varun Pandey, Alexander van Renen, Andreas Kipf, and Alfons Kemper. “How good are modern spatial libraries”, Data Sci. Eng. 6, 2 (2021).

Chris Holmes. ”Performance Explorations of GeoParquet (and DuckDB)", August 2023.

No benchmarks for loading spatial data

No lower-level metrics

Outline

- Introduction
- Micro-Architecture of an OoO Processor Core
- Setup & Methodology
- Comparative Analysis
- Micro-Architectural Behavior
- Conclusion

Micro-Architecture of an OoO Processor Core

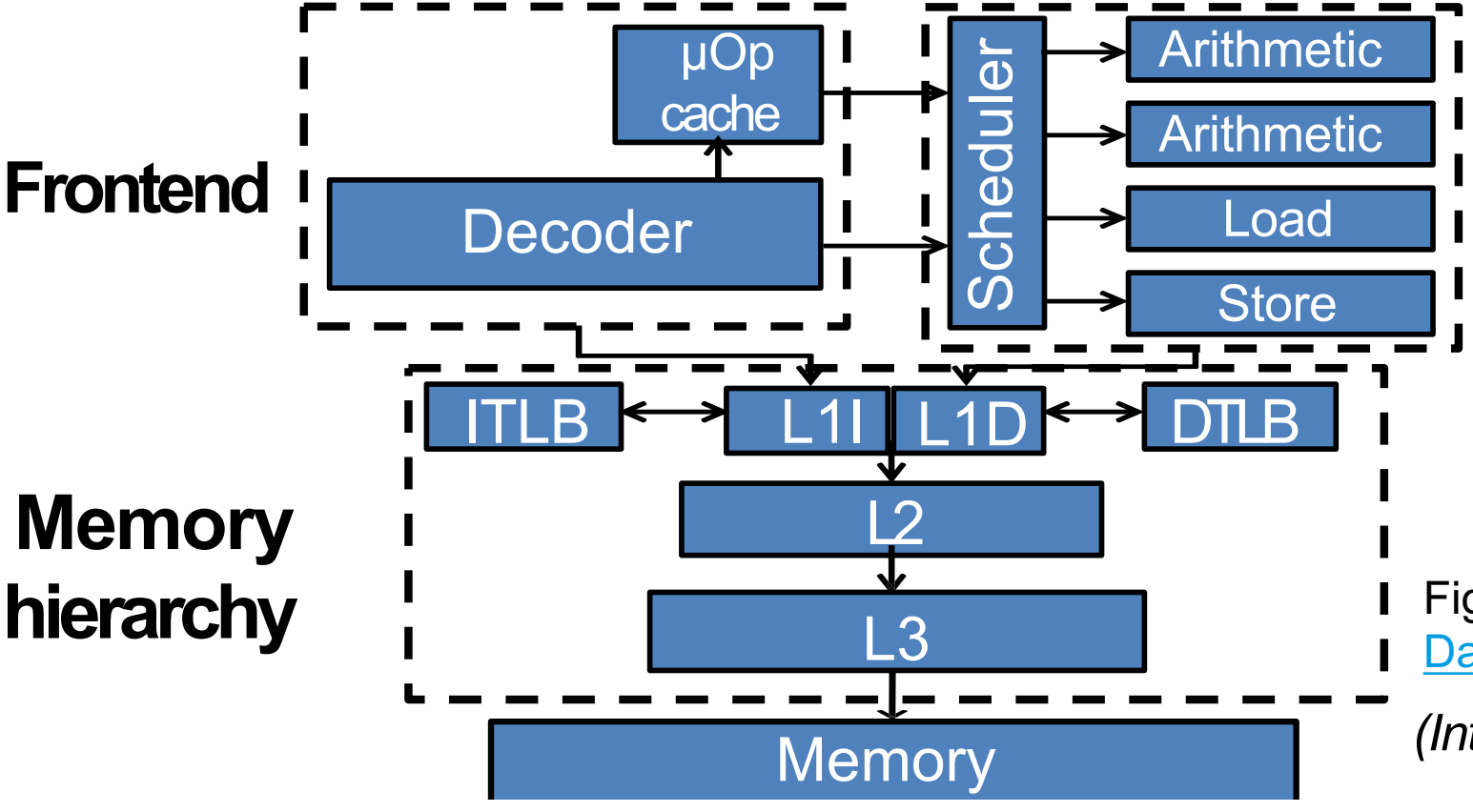


Figure from Utku's Sirin DaMoN 2017 talk

(Intel, SWopt. man., 2016)

Delays in fetching, decoding, etc. an instruction cause *frontend* stalls, rest cause *backend* stalls

Experimental Setup

Software: DuckDB

single- and multi-threaded

GDAL/OGR

GDAL

GeoPandas

File formats:

Shapefile

GeoJSON

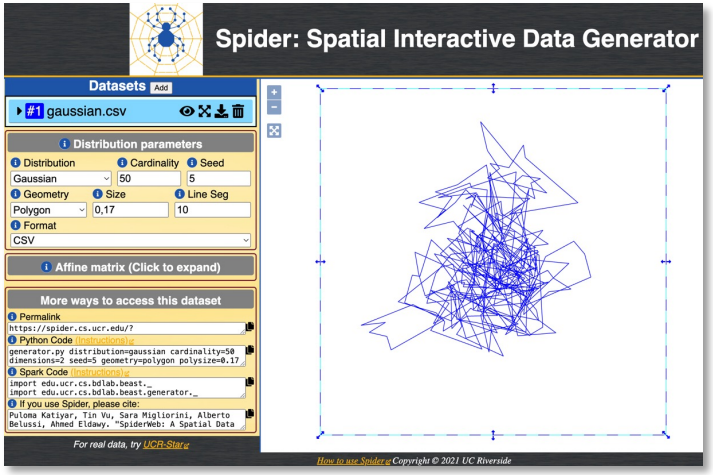
GeoParquet

Geospatial data in Parquet

Hardware: AMD Ryzen 5 5600G, 6-core, 12-thread CPU

Datasets

Synthetic Data Generation



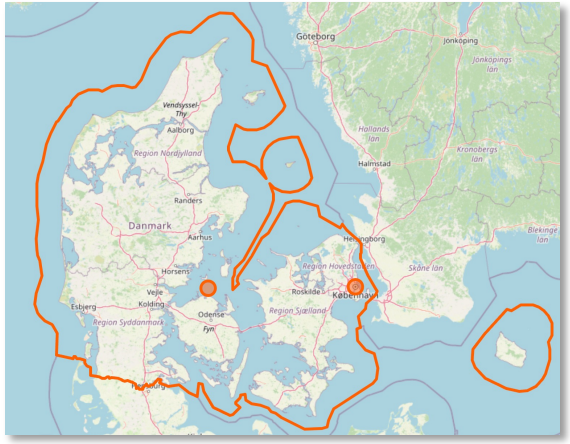
Puloma Katiyar, Tin Vu, Sara Migliorini, Alberto Belussi, Ahmed Eldawy. "SpiderWeb: A Spatial Data Generator on the Web", SIGSPATIAL 2020.

Polygons
(max. 10 line segments)
Gaussian Distribution

File sizes

# of Polygons	GeoParquet Size	Shapefile Size	CSV Size	GeoJSON Size
1 M	122 MB	202 MB	295 MB	410 MB
2 M	244 MB	404 MB	591 MB	820 MB
4 M	488 MB	808 MB	1.2 GB	1.7 GB
8 M	975 MB	1.6 GB	2.4 GB	3.3 GB

OpenStreetMap (OSM)

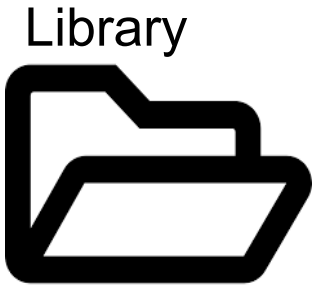


Denmark

6,799,943 Geometries

Methodology

- *Perf stat* tool for hardware counters
- Python script for each library



run.py $ *filepath*

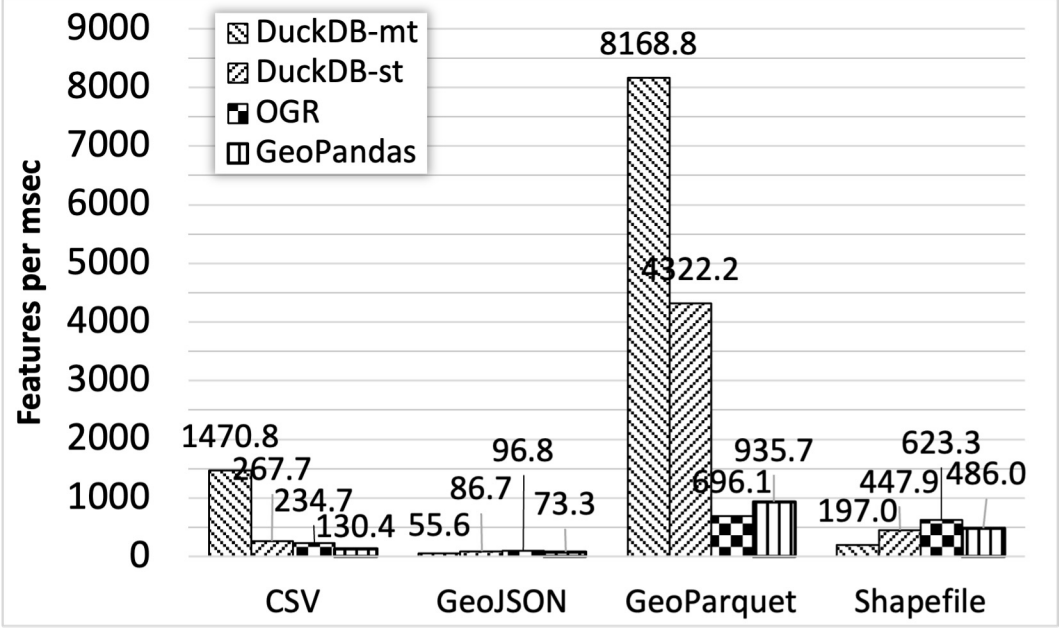
- Runs: (1) Copy file to main memory, (2) Load file X 10

Outline

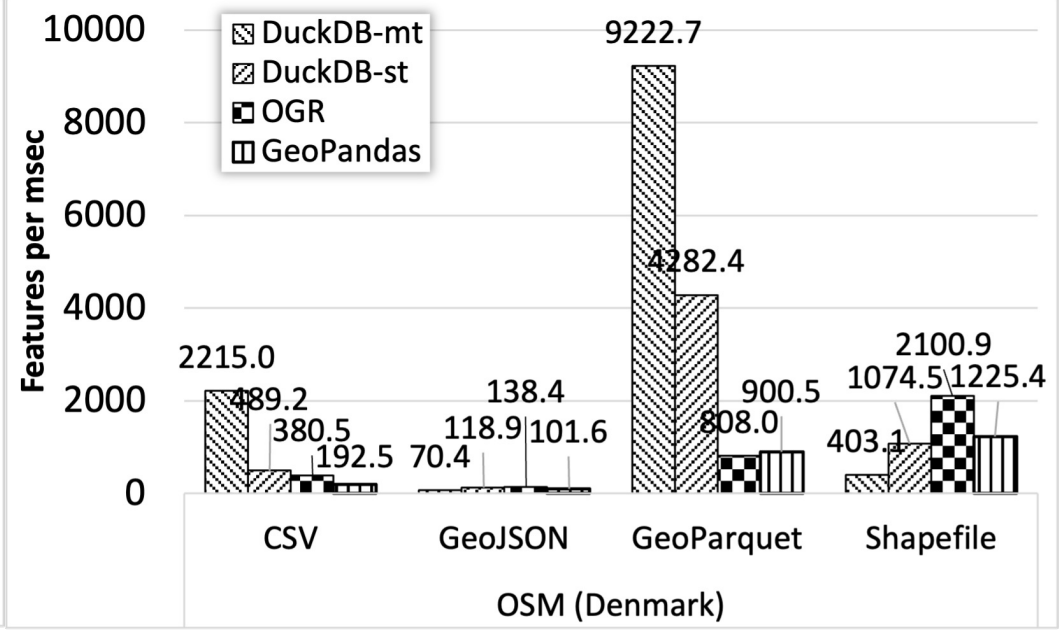
- Introduction
- Micro-Architecture of an OoO Processor Core
- Setup & Methodology
- Comparative Analysis
- Micro-Architectural Behavior
- Conclusion

Loading Throughput

Synthetic Data, 8M polygons



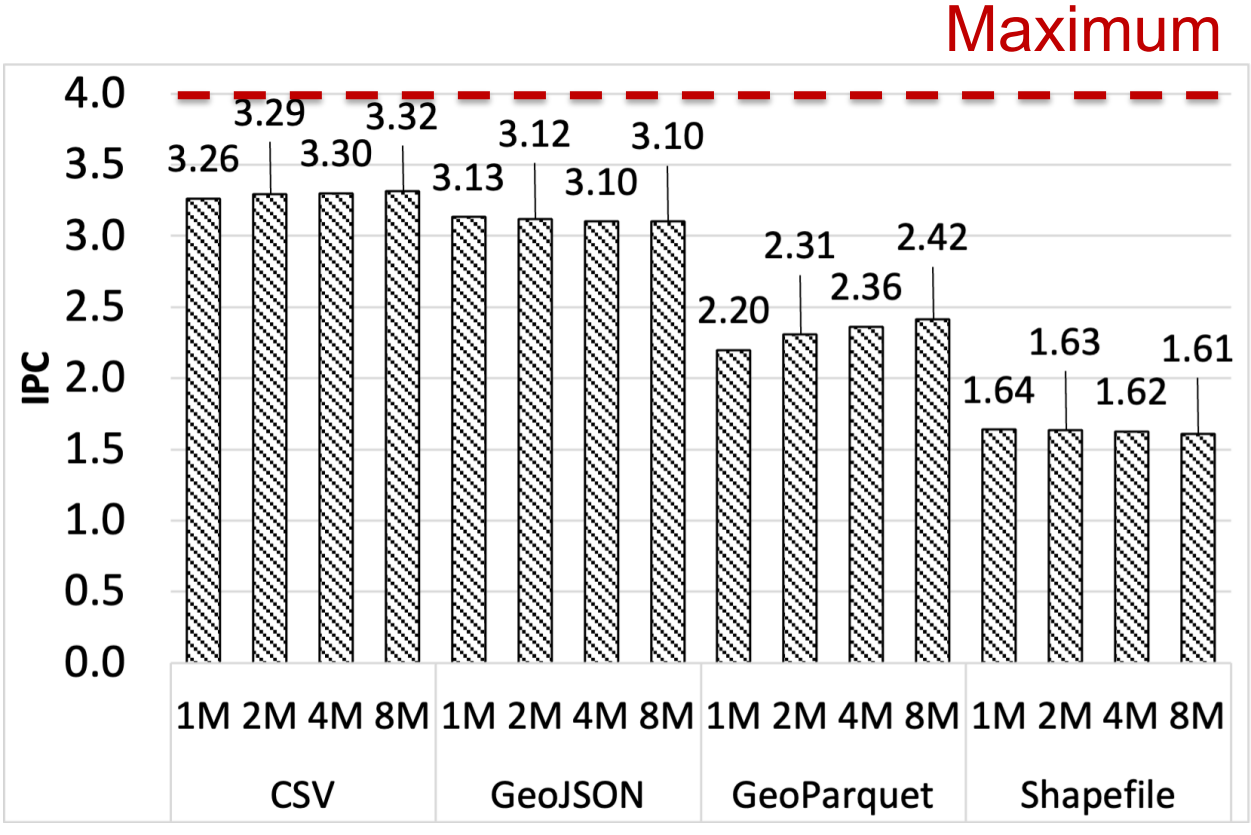
Real Data, ~6.8M geometries



GeoParquet provides the highest throughput in most cases

Instructions per Cycle in DuckDB

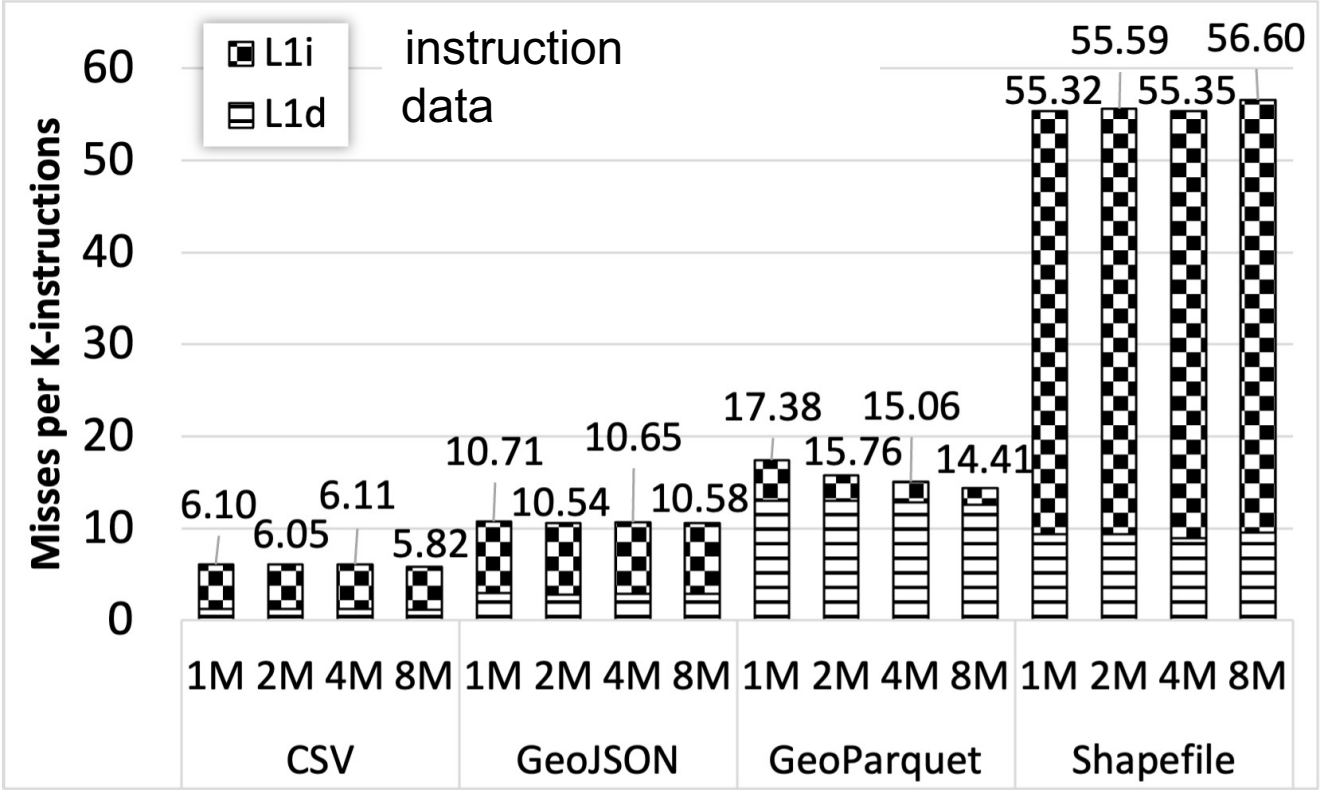
DuckDB-st



High IPC for CSV and GeoJSON, lower for GeoParquet and Shapefile

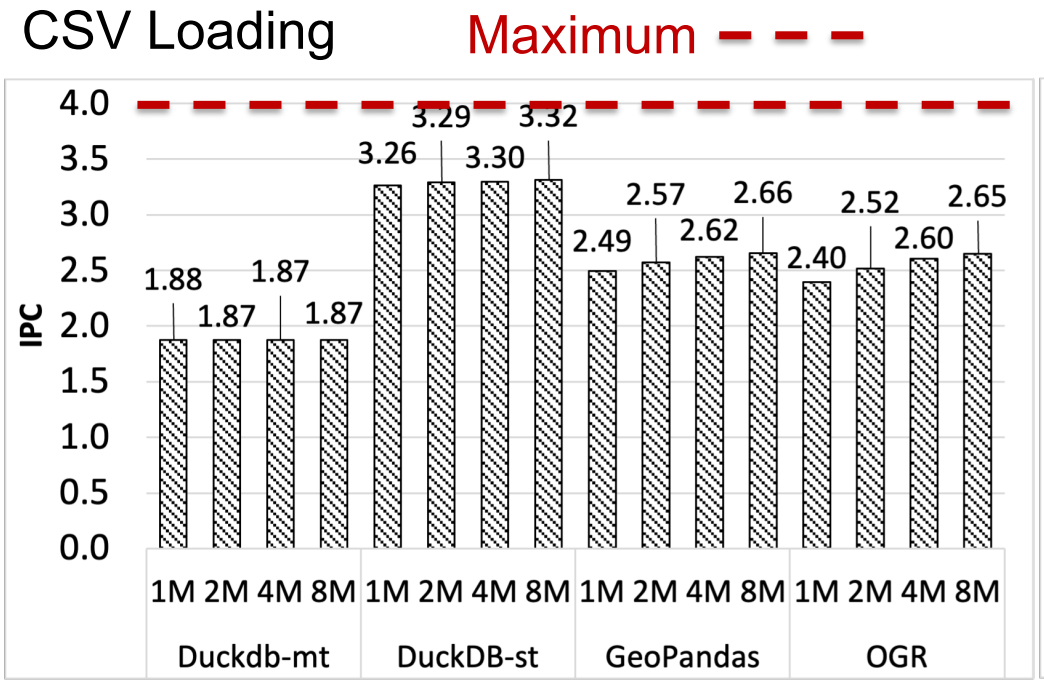
L1 Cache Misses in DuckDB

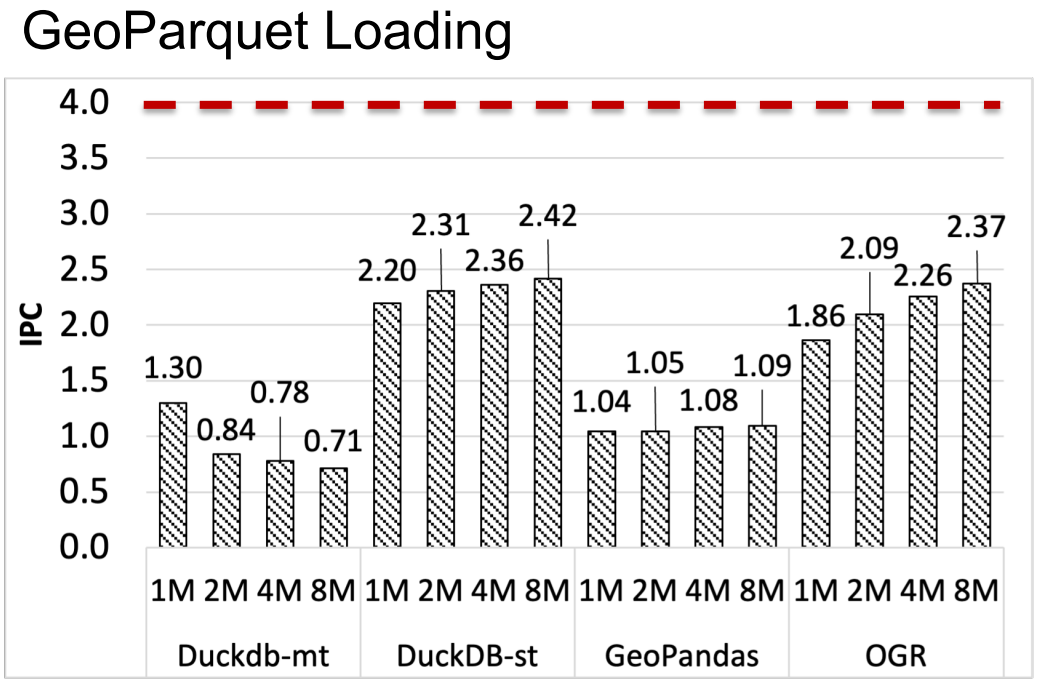
DuckDB-st



Instruction misses dominate, except for GeoParquet

Instructions per Cycle: Impact of File Format



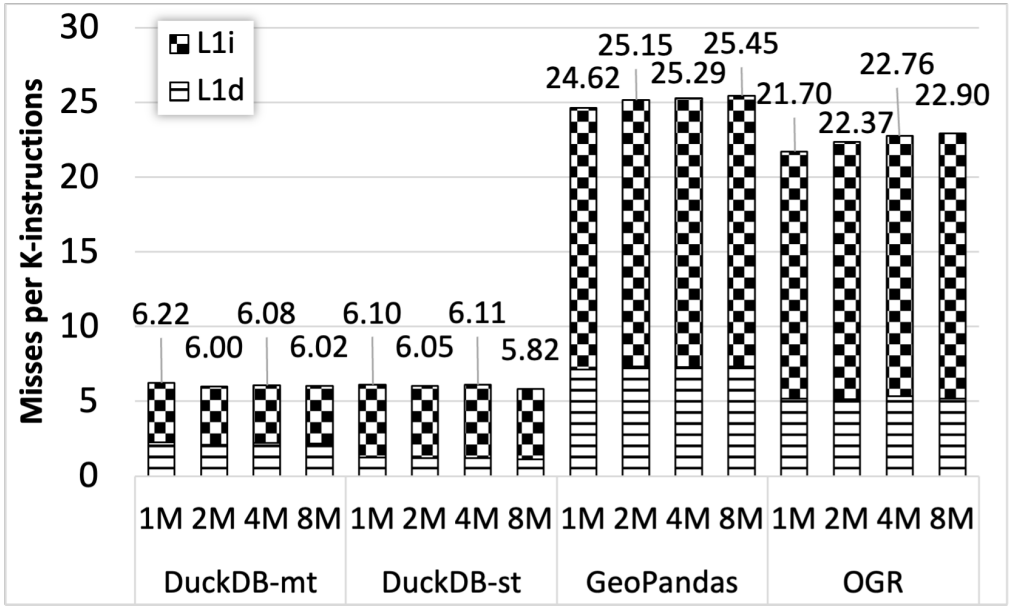


DuckDB's IPC drops in the multi-threaded setup

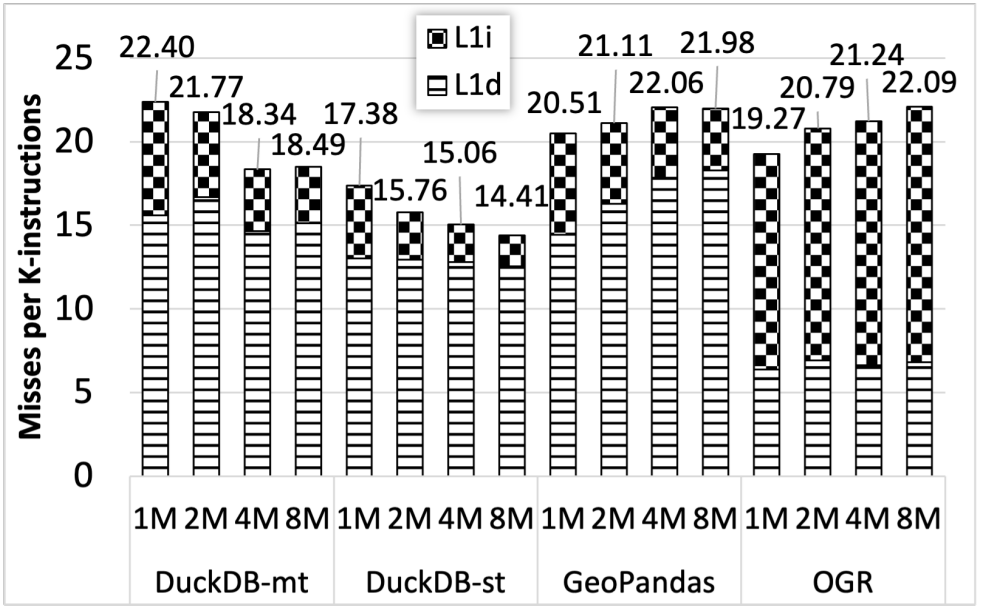
Loading GeoParquet exhibits lower IPC in all systems

L1 Cache Misses: Impact of File Format

CSV Loading



GeoParquet Loading



Few misses in DuckDB

Instruction misses dominate

Data misses dominate except for OGR

Conclusion

- Analysis of geospatial data loading:
  - Impact of system and file format
  - Lower-level metrics
- Main findings:
  - Denser files correlate with higher loading throughput
  - Winner: Loading GeoParquet with DuckDB
  - L1 instruction misses dominate except for GeoParquet
- Future directions: Evaluate impact of
  - different tuning options
  - loading optimizations

Eleni Tzirita Zacharatou
elza@itu.dk

heltzi.github.io

Thank you!

DASYA
Data-Intensive Systems and Applications