DATA+AI SUMMIT 2022

From bits to Data Frames

Data engineering with Arrow and Rust

Jorge C Leitao

Data Scientist, Munin Data

ORGANIZED BY databricks

Background

- PhD in Physics (Max Planck Institute, Germany)
- Data scientist (Teradata, Denmark)
- Co-founder and consultant (Munin Data, Denmark)
- Open source as an hobby
  - Python: Django, Keras, LIME
  - **Rust: Arrow**

Munin Data

solve high-impact business challenges in the realm of Analytics and Big Data

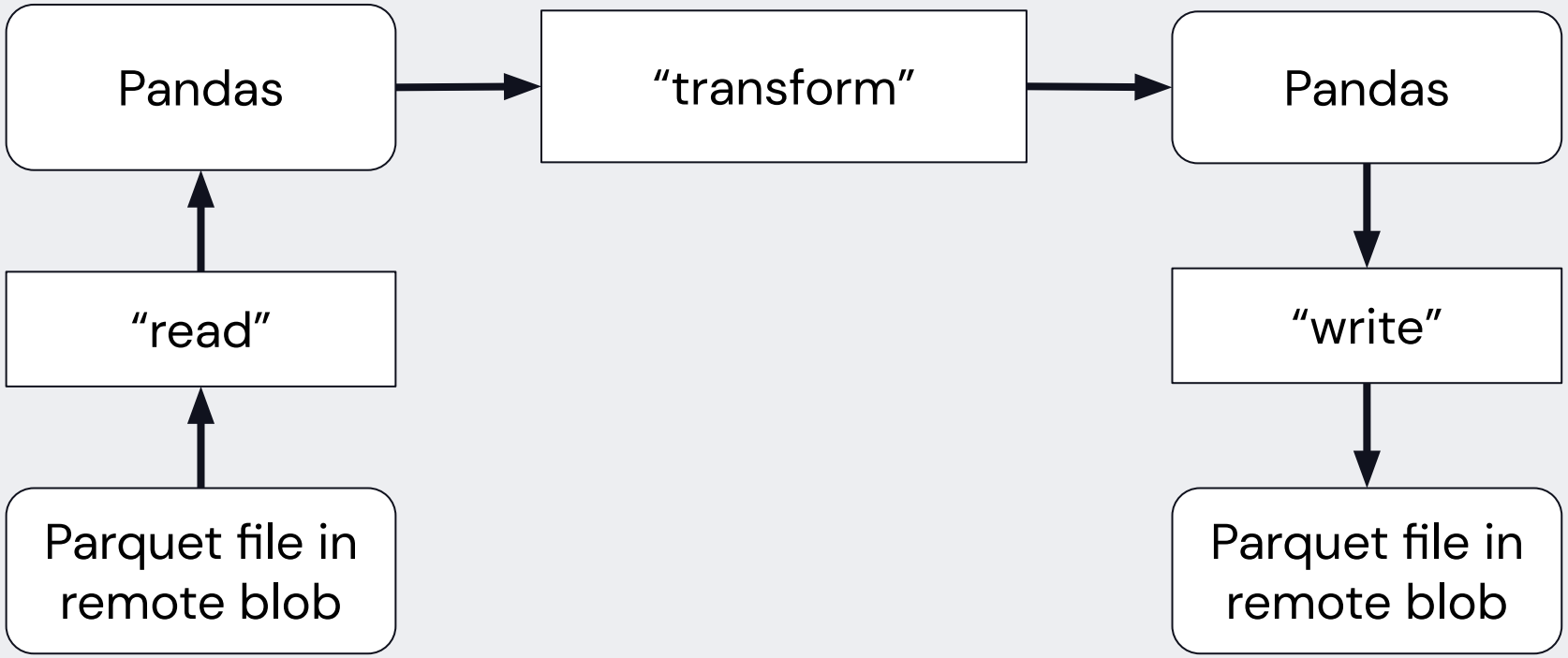
- Operates in Denmark, Europe
- Large enterprises in Bio-tech and pharma
- Experts on
  - Data lakes, cloud infrastructure and analytics workloads
  - Open source stacks
  - DevOps and GxP in Pharma

MUNIN DATA

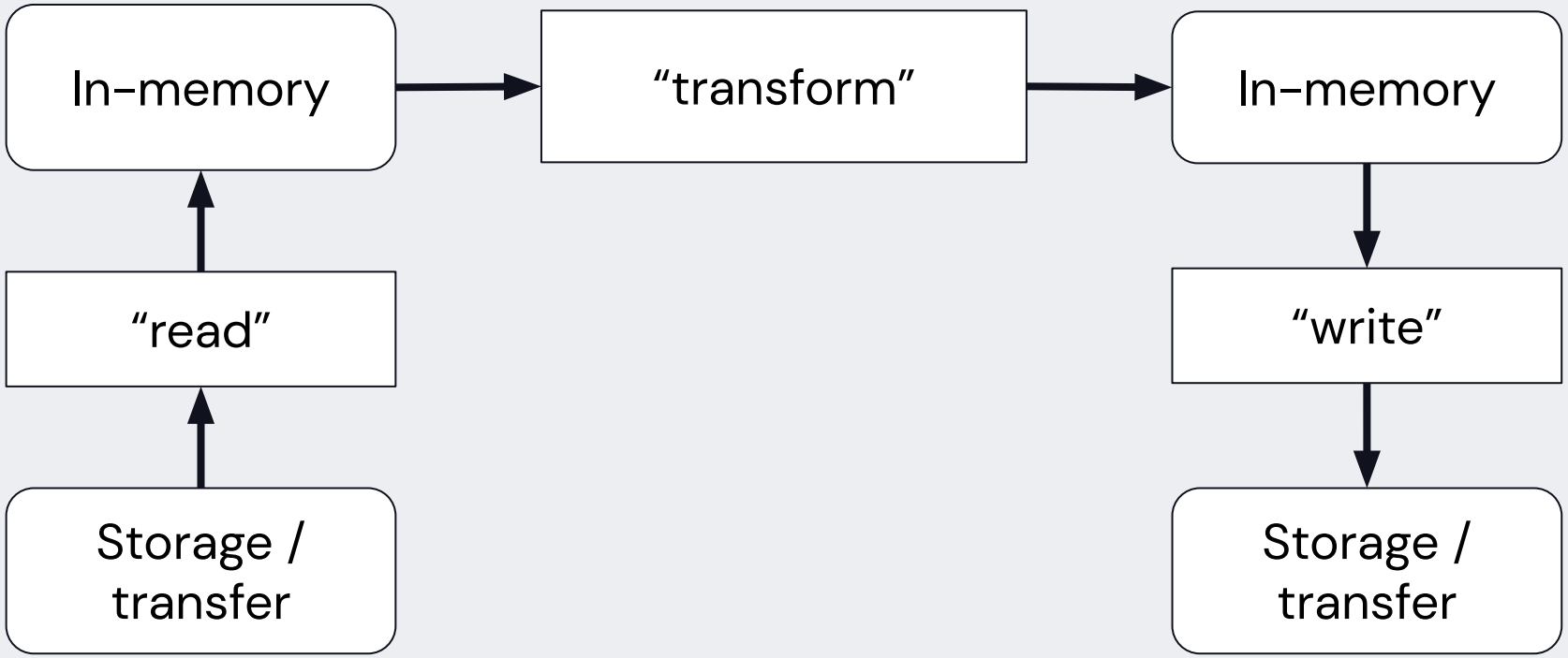
Outline

- **Anatomy of Analytics Workloads**
- Arrow and Rust for Analytics
- Demo and Benchmarks

A simple ETL


Pandas
“transform”
Pandas
“read”
“write”
Parquet file in remote blob
Parquet file in remote blob

Analytics workload



Information is both stored and used

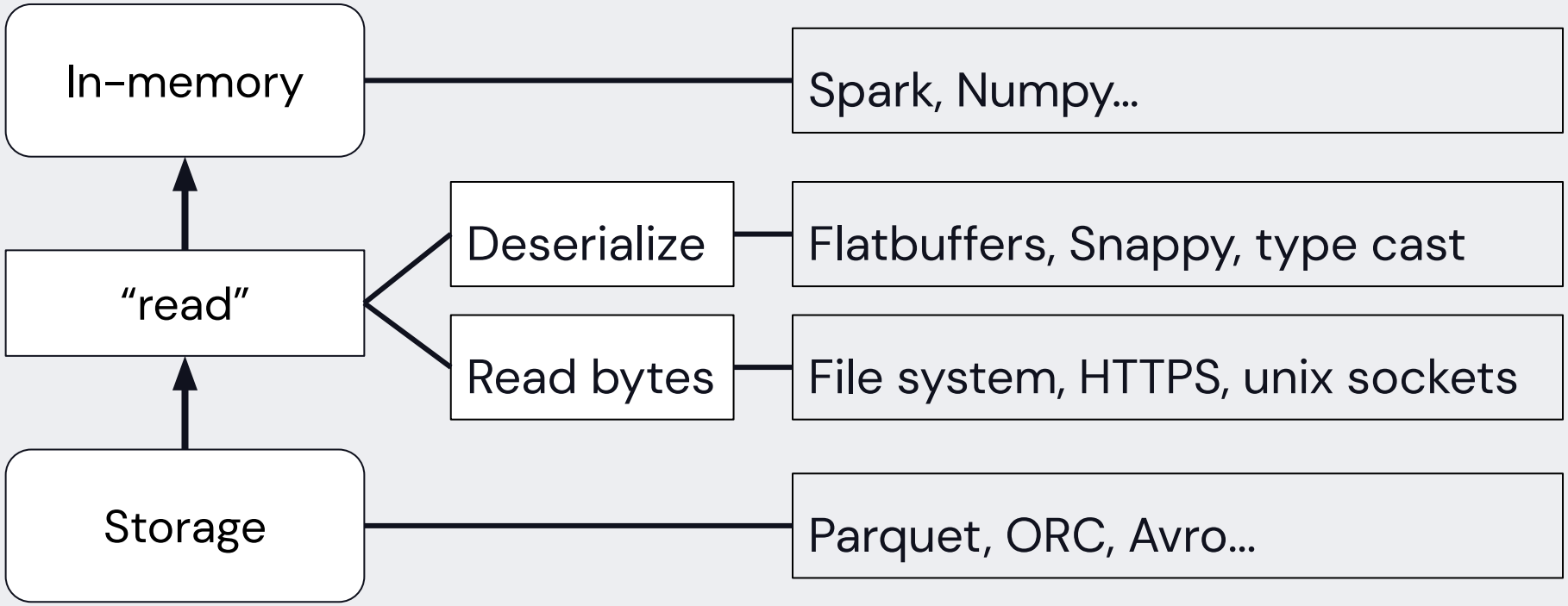
Storage formats are optimized to

- save space
- be cross-language compatible
- long-term storage

In-memory formats are optimized to

- hit fast instruction sets
- be cache friendly
- be parallelizable

“Read” uses IO and CPU



CPUs sleep and run

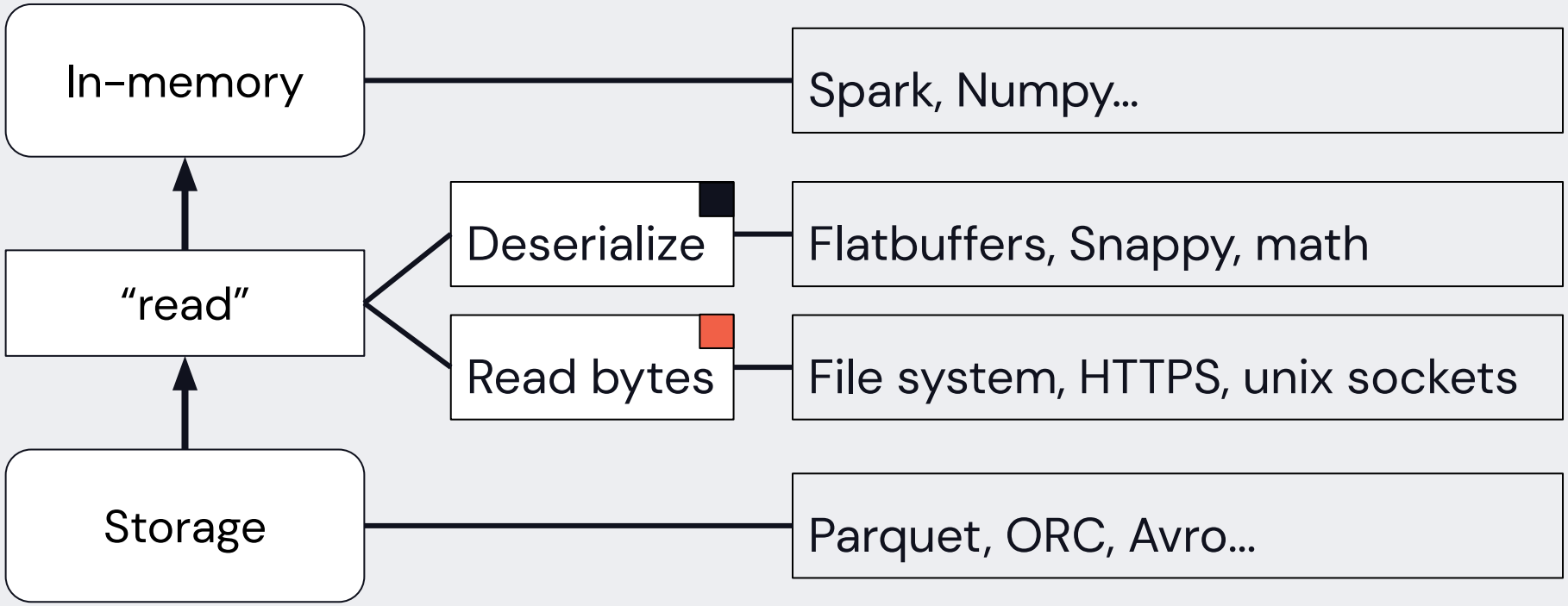
IO-bounded (e.g. read)

- CPU sleeps
- Use of external resources
- Single-thread “concurrentable”

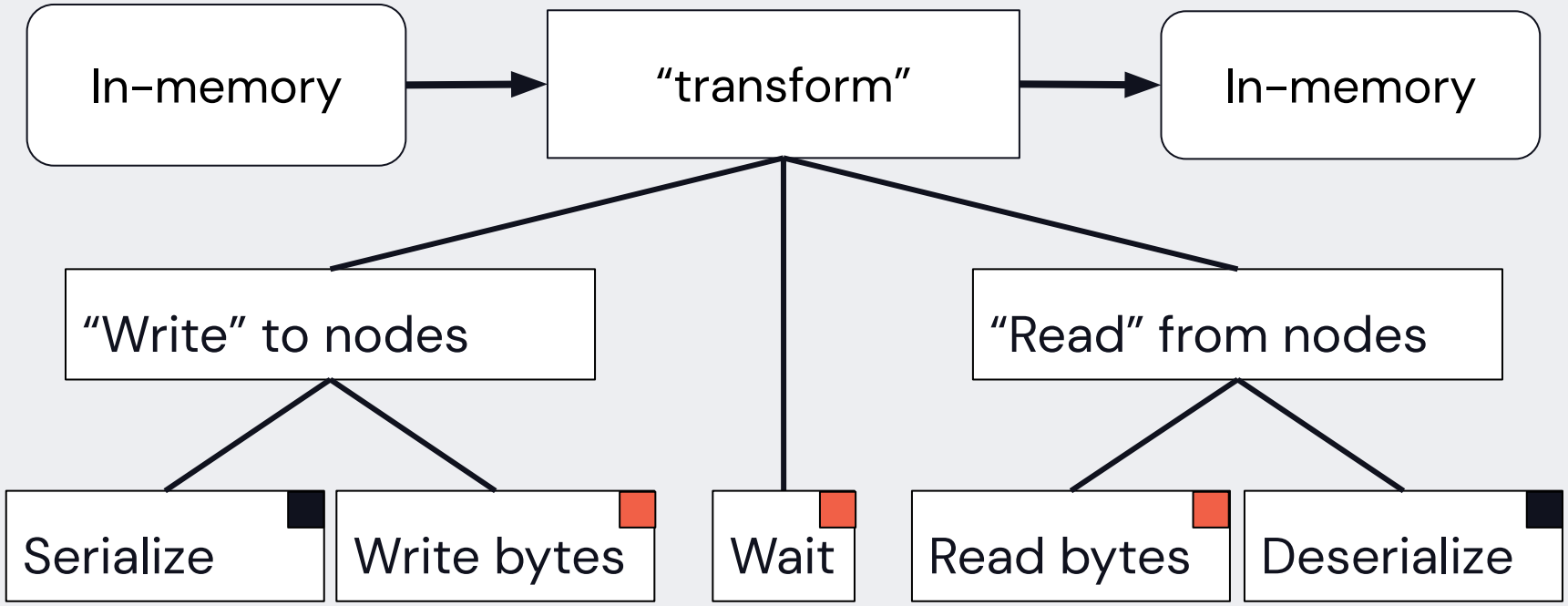
CPU-bounded (e.g. deserialize)

- CPU runs
- Primarily CPU and RAM
- Multi-core “parallelizable”

“Read” uses IO and CPU



“Transform” uses IO and CPU



In summary

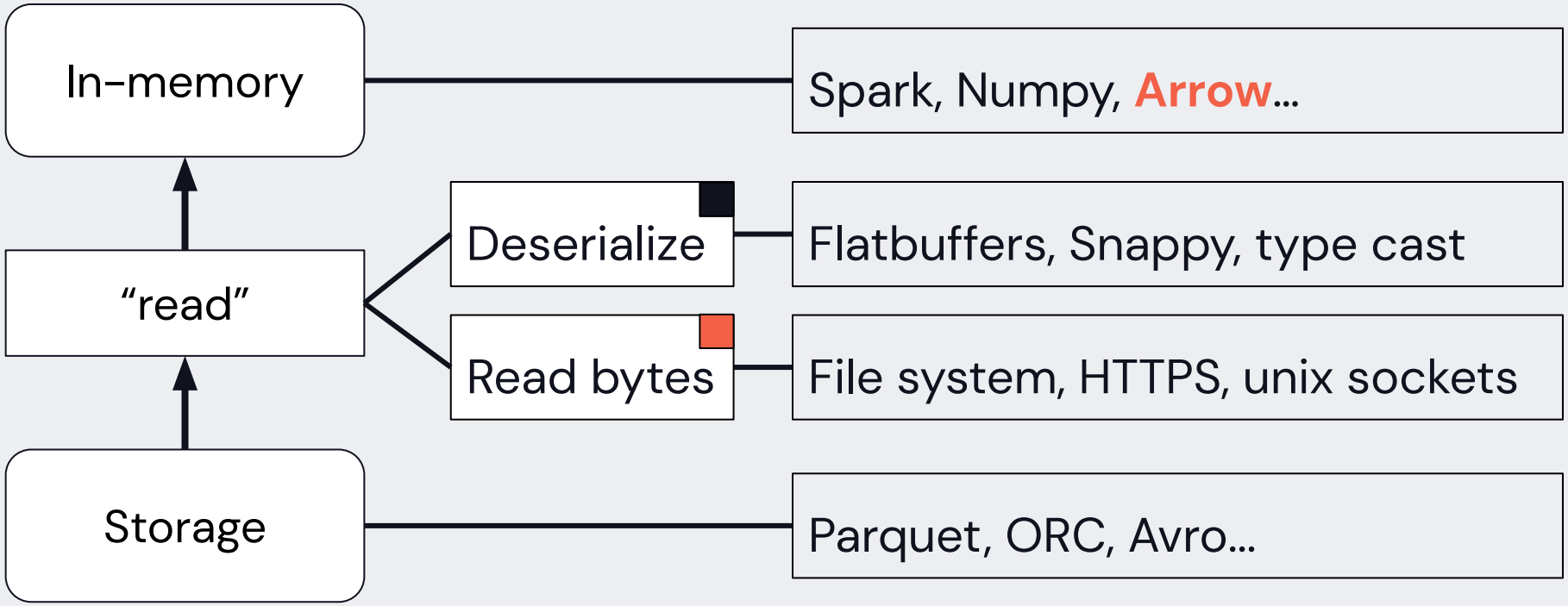
- Storage vs compute induce different formats
- Analytics is a mix of CPU- and IO-bounded tasks
- Control over CPU and IO seems quite important…

Outline

- Anatomy of Analytics Workloads
- **Arrow and Rust for Analytics**
- Demo and Benchmarks

Apache Arrow

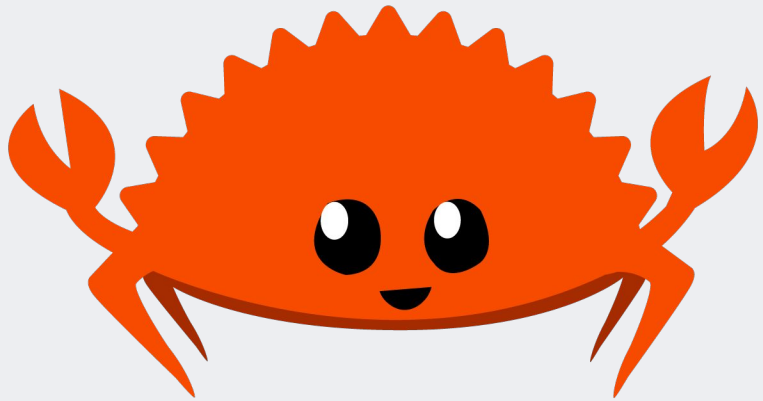
In-memory columnar format (and IPC specification)



APACHE ARROW

Rust Programming Language

- Easy to develop
- Easy to parallelize
- Easy to hit modern instruction sets
- Easy to package and distribute



The most loved language in SO survey for 7(!) consecutive years

Arrow with Rust

Arrow2 - Rust Library

- Complete Arrow specification
- Interoperability with Parquet, Avro, ODBC, CSV, JSON, etc.
- Fast
- Safe and sound (memory, data races, etc.)
- Complete separation between IO- and CPU-bounded APIs

https://github.com/jorgecarleitao/arrow2

https://github.com/jorgecarleitao/parquet2

Demo

- Example of simple math
- Write a parquet file (fast)

Who uses arrow2

- Databend, https://databend.rs/
- Materialize, https://materialize.com/
- Graphana SDK, https://github.com/grafana/grafana-plugin-sdk-rust
- Polars, https://pola.rs

Polars

Blazingly fast DataFrame API

- Rust Native with Apache Arrow
- API in Python, Node.js and Rust
- Fast

https://github.com/pola-rs/polars/

https://databricks.com/dataaisummit/session/polars-blazingly-fast-dataframes-rust-and-python

Benchmarks

H2O.ai (groupby 50Gb)

basic questions

Input table: 1,000,000,000 rows x 9 columns (50 GB)

Polars	0.8.8	2021-06-30	143s
data.table	1.14.1	2021-06-30	155s
DataFrames.jl	1.1.1	2021-05-15	200s
ClickHouse	21.3.2.5	2021-05-12	256s
cuDF*	0.19.2	2021-05-31	492s
spark	3.1.2	2021-05-31	568s
(py)datatable	1.0.0a0	2021-06-30	730s
dplyr	1.0.7	2021-06-20	internal error
pandas	1.2.5	2021-06-30	out of memory
dask	2021.04.1	2021-05-09	out of memory
Arrow	4.0.1	2021-05-31	internal error
DuckDB*	0.2.7	2021-06-15	out of memory
Modin		see README	pending

https://h2oai.github.io/db-benchmark/

Benchmarks

H2O.ai (join 5Gb)

basic questions

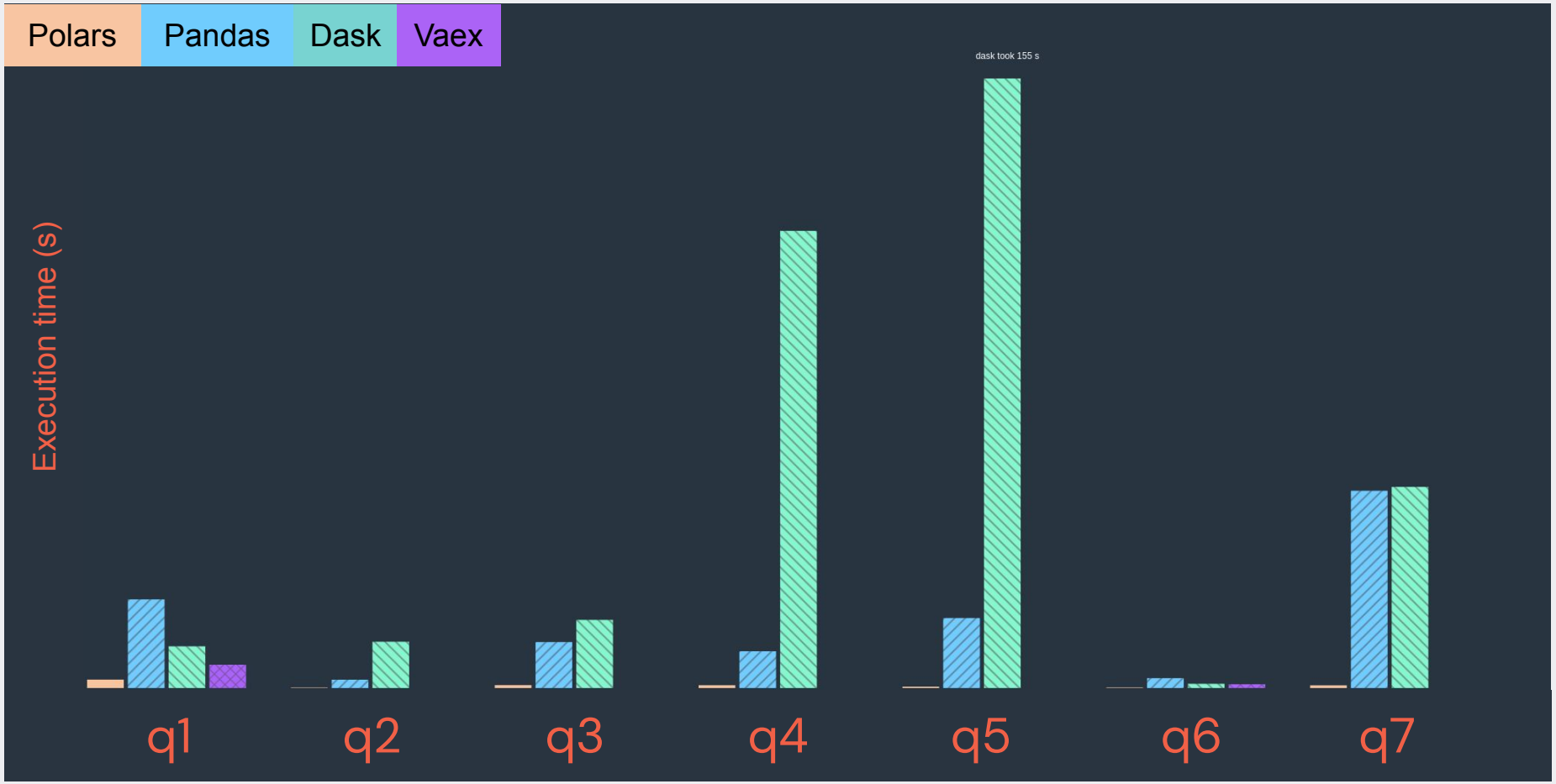
Input table: 100,000,000 rows x 7 columns (5 GB)

Polars	0.8.8	2021-06-30	43s
data.table	1.14.1	2021-06-30	92s
ClickHouse	21.3.2.5	2021-05-12	159s
spark	3.1.2	2021-05-31	332s
DataFrames.jl	1.1.1	2021-06-03	349s
dplyr	1.0.7	2021-06-20	370s
(py)datatable	1.0.0a0	2021-06-30	500s
pandas	1.2.5	2021-06-30	628s
DuckDB	0.2.7	2021-06-15	630s
dask	2021.04.1	2021-05-09	internal error
cuDF*	0.19.2	2021-05-31	internal error
Arrow	4.0.1	2021-05-31	not yet implemented
Modin		see README	pending

https://h2oai.github.io/db-benchmark/

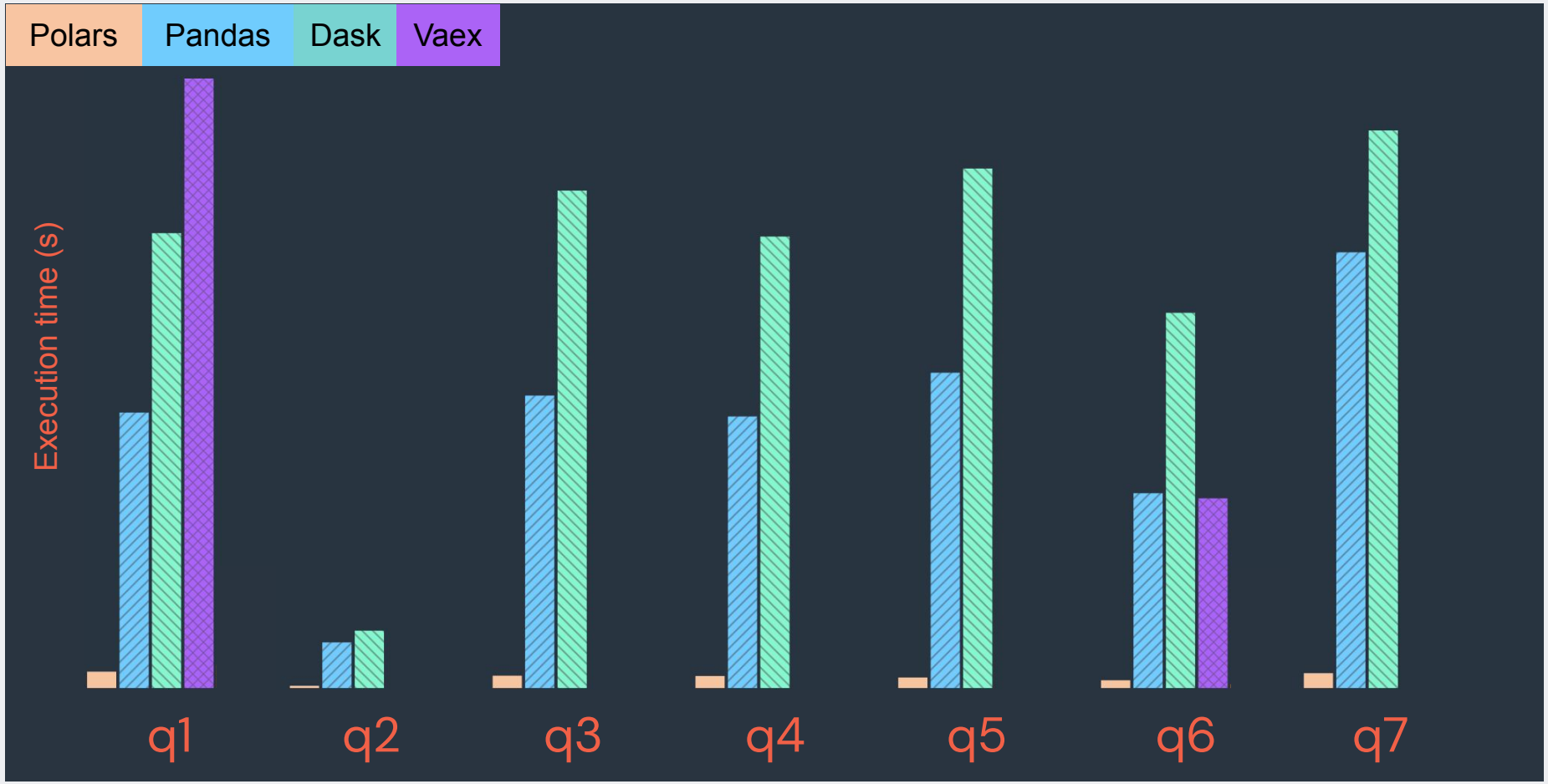
Benchmarks

TPCH SF-10 (“transform”)



Benchmarks

TPCH SF-10 (“read” + “transform”) from parquet



In summary

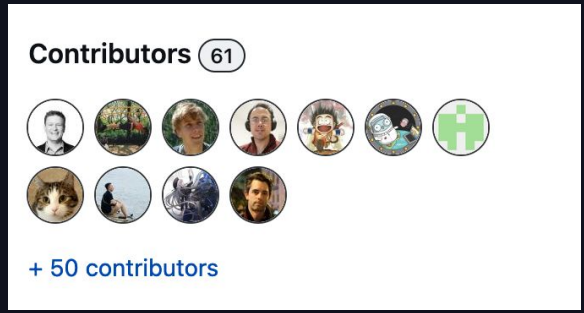
- Information is stored and used in analytics
- Arrow with Rust are very suited for analytics workloads
- Impressive developments that will change the analytics landscape

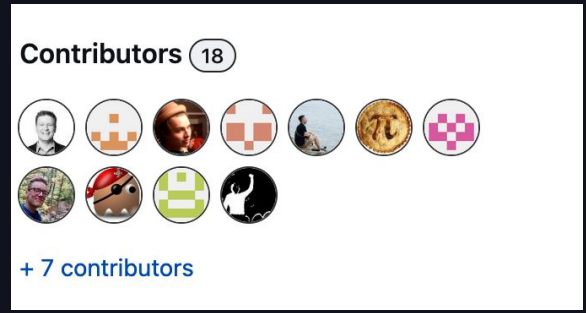
And more...

join us to learn and have fun :)

- https://github.com/jorgecarleitao/arrow2
- https://github.com/jorgecarleitao/parquet2
- https://pola.rs
- https://databend.rs/
- https://materialize.com/
- https://github.com/DataEngineeringLabs/arrow2-convert
- https://databricks.com/dataaisummit/session/polars-blazingly-fast-dataframes-rust-and-python

DATA+AI
SUMMIT 2022





Thank you

Jorge Leitao
Data scientist