CSE P 590

Building Data Analysis Pipelines

Fall 2024

Big data

Today

- Big data characteristics and challenges
- Big data processing
- In-class 5

Big data: characteristics and challenges

Big data

Characteristics

What do we mean by big data?

Big data

Characteristics

- *Volume*: data sets are (too) big → distributed analysis
- *Variety*: data formats: structured, semi-structured, unstructured
- *Velocity*: data changes rapidly → real-time analysis

- *Variability*: meaning of data changes
- *Veracity*: noisy data (tradeoff between noisy and useful)
- *Value*: informed decision making (value of collected data)

Big data: variety of data formats

Structured

Semi-structured

Unstructured

What are differences, examples, and challenges?

Big data: variety of data formats

Structured

- Rigid schema
- Examples: Relational databases, parquet, protobufs

Semi-structured

- Flexible schema
- Examples: json, xml, log files

Unstructured

- No schema
- Examples: commit/review messages, audio, video

Big data: distributed data

Distributed File Systems

- Datasets stored across multiple nodes
- HDFS (Hadoop Distributed File System), S3, etc.

What are the advantages and challenges of distributed data?

Big data: distributed data

Distributed File Systems

- Datasets stored across multiple nodes
- HDFS (Hadoop Distributed File System), S3, etc.

Advantages

- High fault tolerance
- Data replication for better performance

Challenges

- Data locality: move data vs. move computation

Big data: challenges

Compute bound

- Compute-intensive simulations
- Real-time processing
- High-volume data

Memory bound

- Data exceeds memory on a single machine

I/O bound

- Different data sources (structured, semi-structured, unstructured)
- Data sharing among different processes

Big data processing

Compute bound: Rcpp

Optimize runtime

- Loops or recursive functions
- Custom functions

Libraries

- Advanced data structures
- HPC libraries (e.g., simulations)

```
library(Rcpp)

cppFunction('int add(int x, int y, int z) {
  int sum = x + y + z;
  return sum;
}')

add(1, 2, 3)
```

https://adv-r.hadley.nz/rcpp.html

Memory bound: sparklyr (a Spark DSL)

Distributed data storage

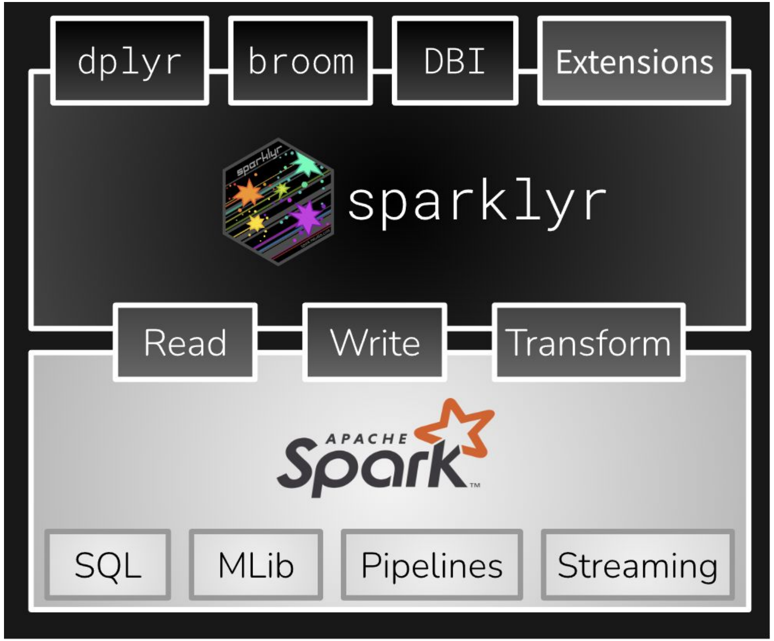
- A dplyr backend
- Supports SQL (like dbplyr)
- Lazy evaluation

Distributed data processing

- Support for many data formats

Modeling/ML

- Support for many common model types



https://spark.posit.co

I/O bound: arrow

Reduce serialization costs

- Backend for dplyr
- Efficient columnar data format
- Zero-copy data sharing (between R and Python)

https://adv-r.hadley.nz/rcpp.html

Live demo: Spark

In-class 5