Serverless Machine Learning on Modern Hardware

IBM Research

#Res6SAIS

Serverless Computing

- No need to setup/manage a cluster
- Automatic, dynamic and fine-grained scaling
- Sub-second billing
- AWS Lambda, Google Cloud Functions, Azure Functions, Databricks Serverless

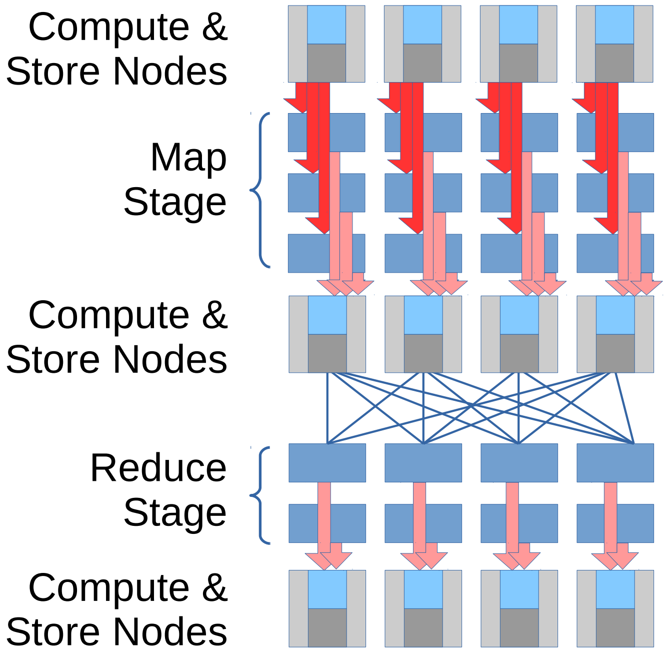
Challenge: Performance

- **Container startup:** may have to dynamically spin up containers per function call
  - Takes several 200-300 milliseconds for a “cold” container
- **Storage:** input data needs to be fetched from remote storage (e.g., S3 object store)
  - As opposed to compute-local storage, e.g., HDFS
- **Data sharing:** intermediate needs to be temporarily stored on remote storage (e.g. S3, Redis)
  - Becomes problematic as workloads get more complex
  - Affects operations like shuffle, broadcast, etc.,

Challenge: Performance

- **Container startup:** may have to dynamically spin up containers per function call
  - Takes several 200-300 milliseconds for a “cold” container
- **Storage**: input data needs to be fetched from remote storage (e.g., S3 object store)
  - As opposed to compute-local storage, e.g., HDFS
- **Data sharing:** intermediate needs to be temporarily stored on remote storage (e.g. S3, Redis)
  - Becomes problematic as workloads get more complex
  - Affects operations like shuffle, broadcast, etc.,

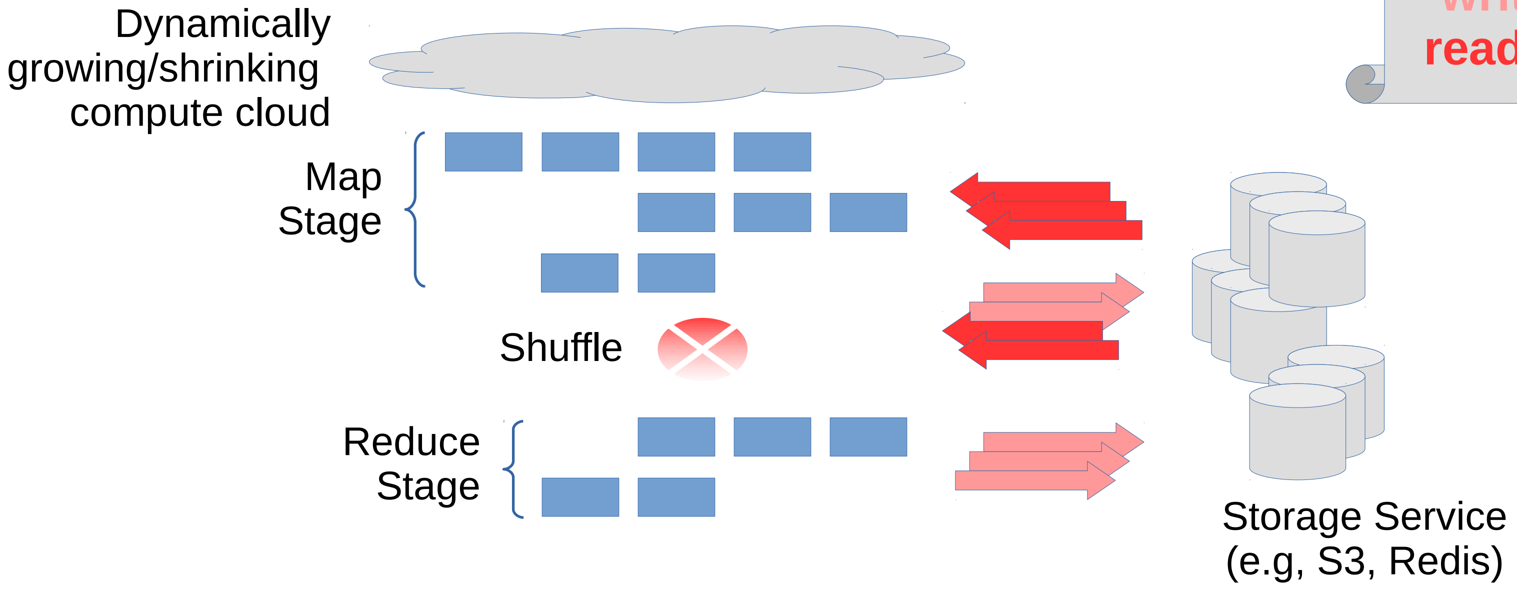
Example: MapReduce (Cluster)



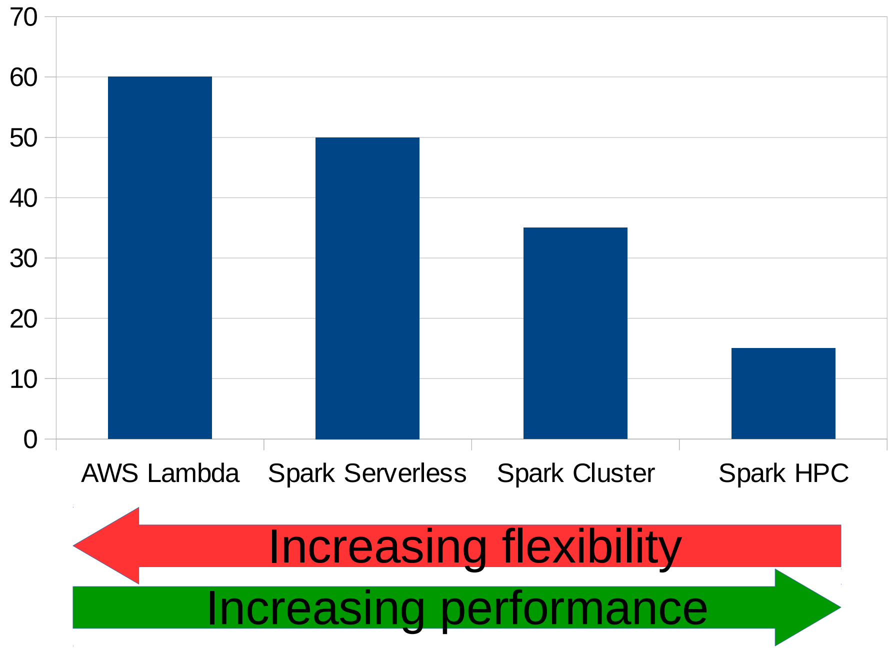
data is mostly written and read locally

Serverless MapReduce

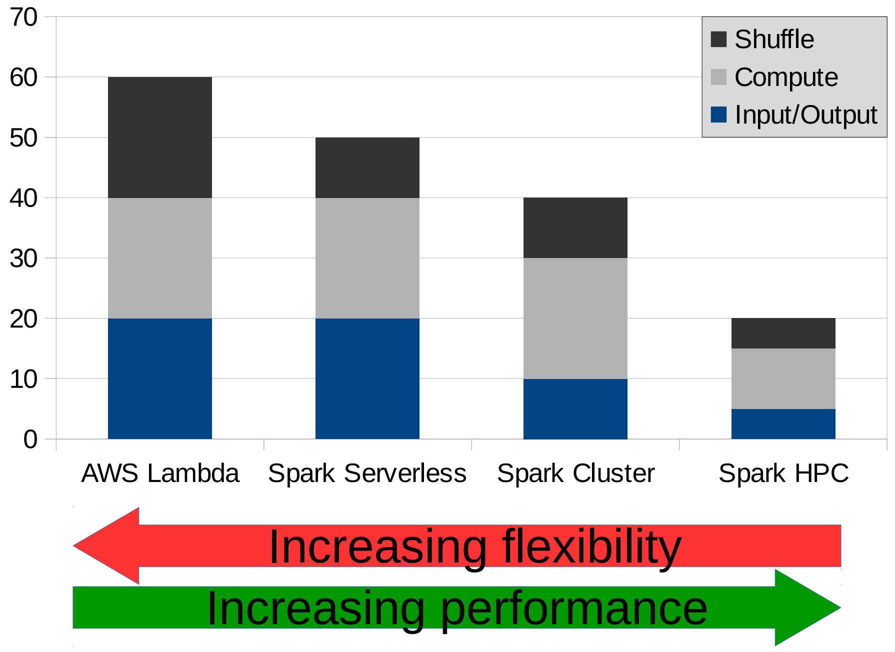
data is exclusively written and **read** remotely


Dynamically growing/shrinking compute cloud
Map Stage
Shuffle
Reduce Stage
Storage Service (e.g, S3, Redis)

Sorting 100GB

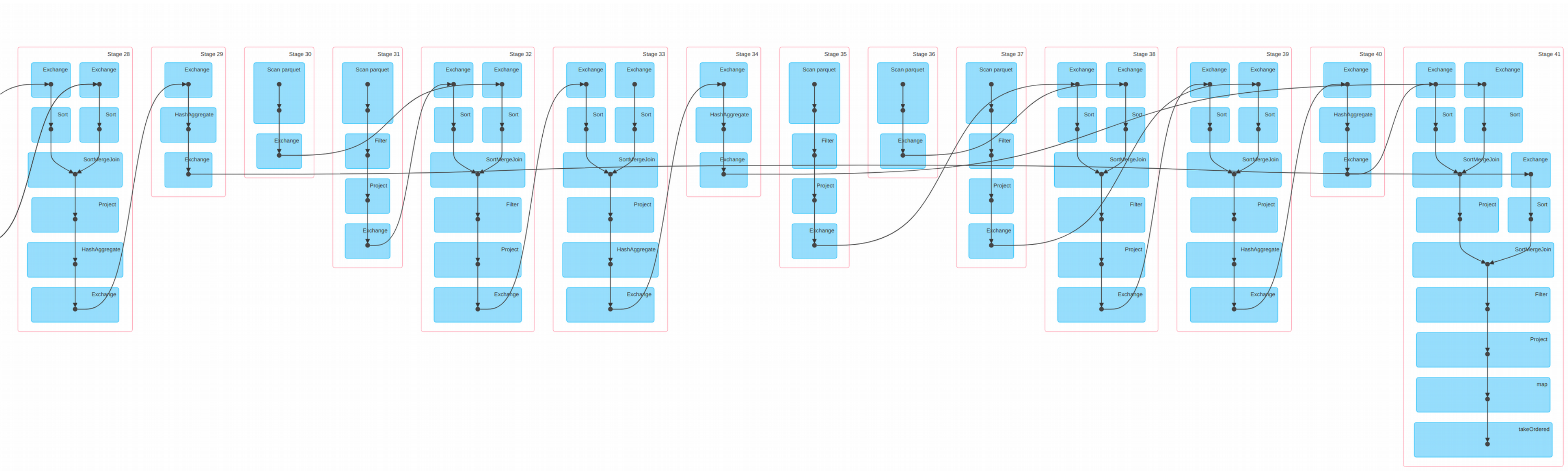


Is I/O a problem?



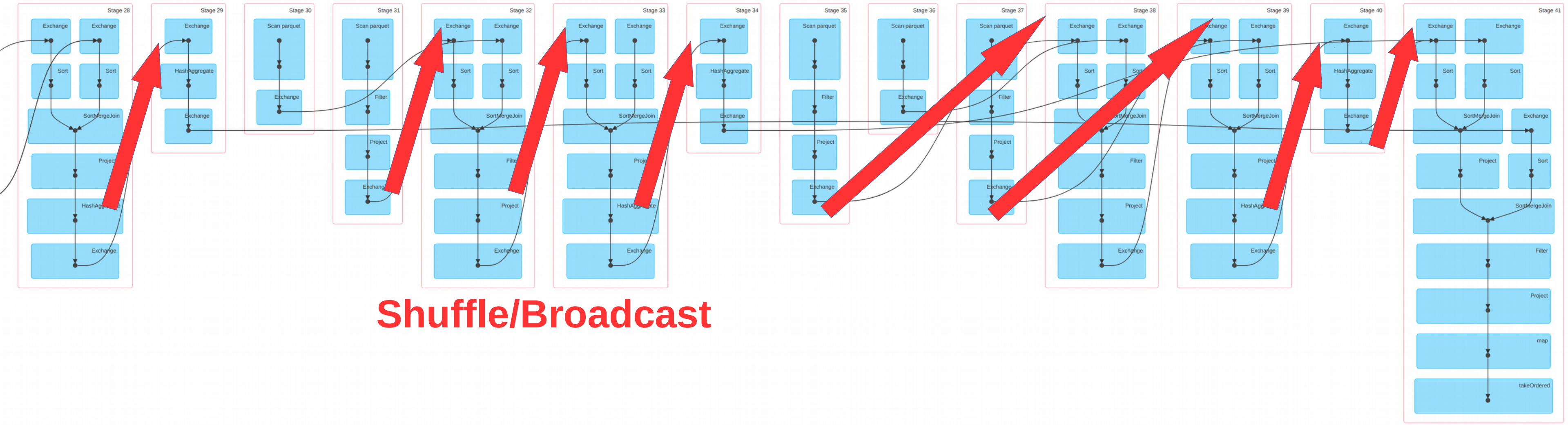
What about other workloads?

Example: SQL, Query 77 / TPC-DS benchmark



What about other workloads?

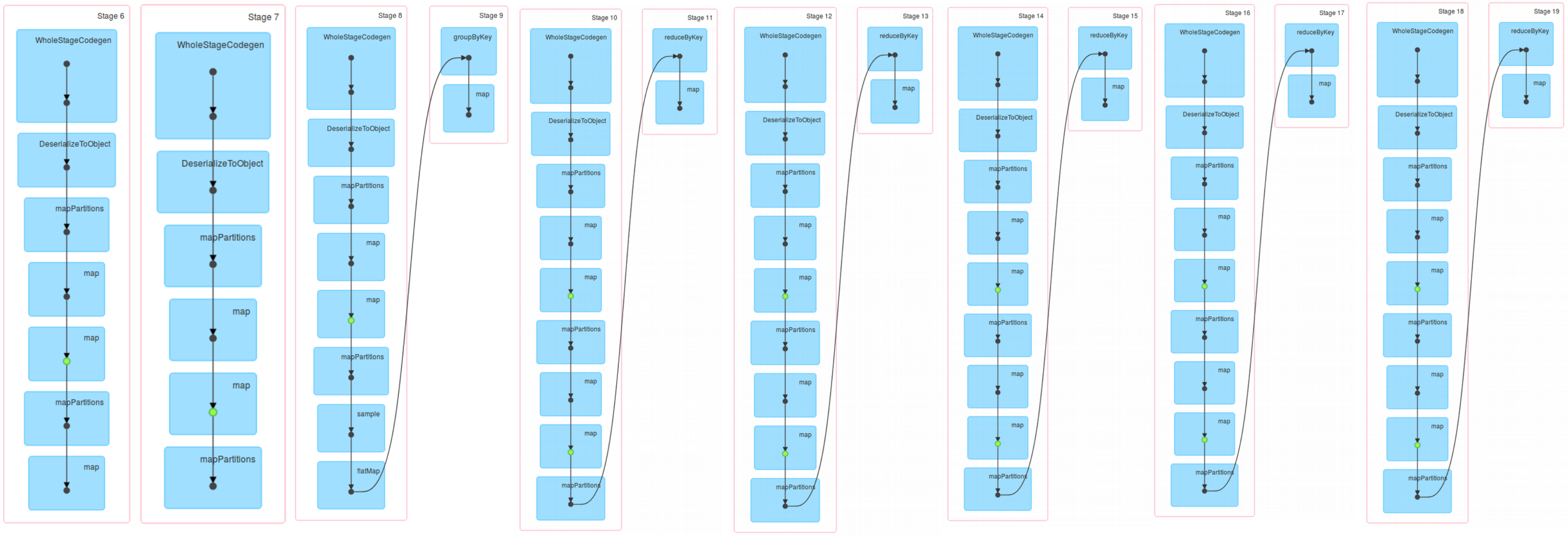
Example: SQL, Query 77 / TPC-DS benchmark



Shuffle/Broadcast

What about other workloads?

Example: RandomForest



SPARK+AI SUMMIT 2018

Workloads and Frameworks

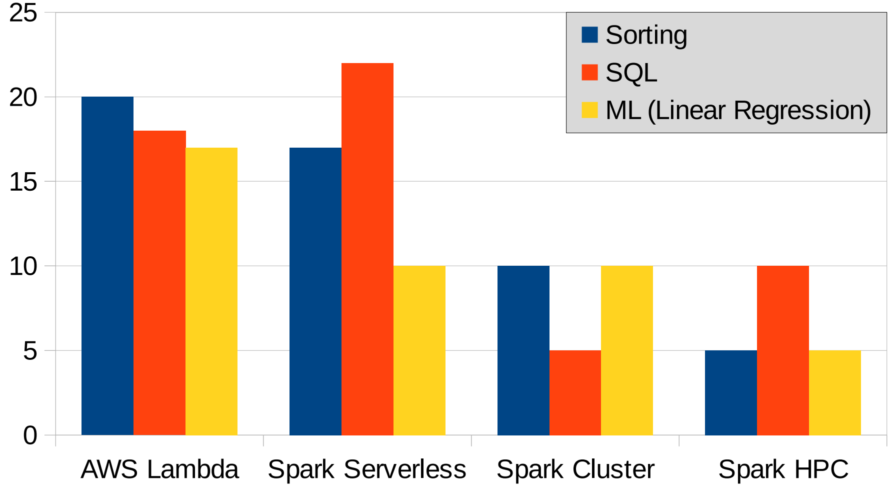
	Microservices	Workflows	MapReduce	SQL	ML
AWS λ, Google CF, Azure F					
AWS λ + AWS StepFunction					
PyWren					
Databricks Serverless					

Serverless frameworks not designed to run arbitrary workloads

Challenge #2: Workloads

- Serverless originally designed for simple use cases
  - E.g., image post-processing triggered by an upload
- What about more complex workloads?
  - MapReduce: Can be implemented on top of most frameworks
  - SQL? Databricks
  - Machine Learning?

Serverless: Different Workloads



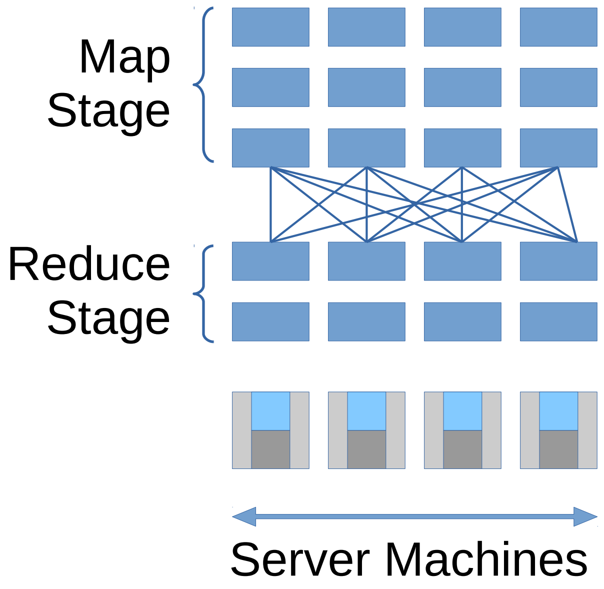
Serverless frameworks not designed to run arbitrary workloads

Backup

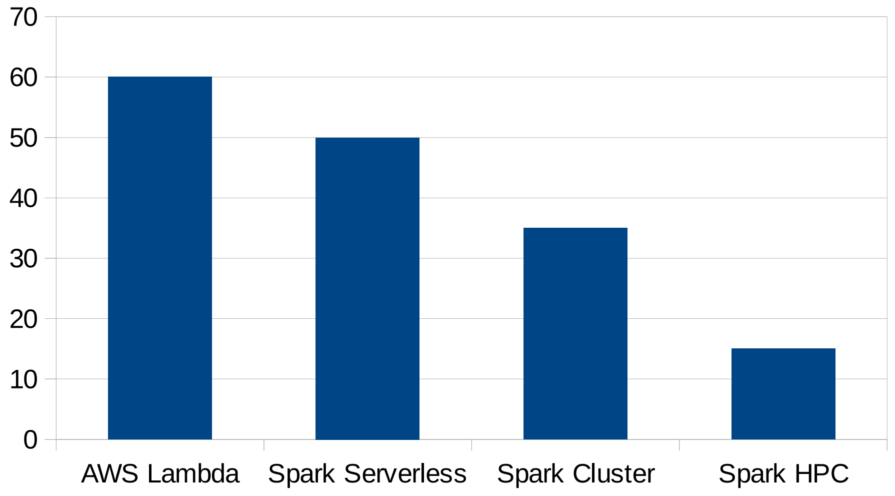
Template Tite

- Template List
- Template List
  - Template item

Example: MapReduce

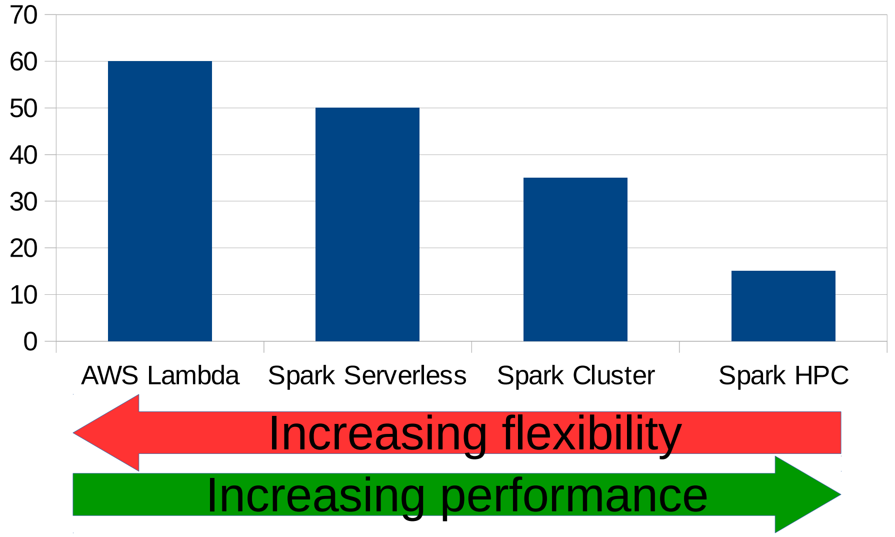


Sorting 100GB



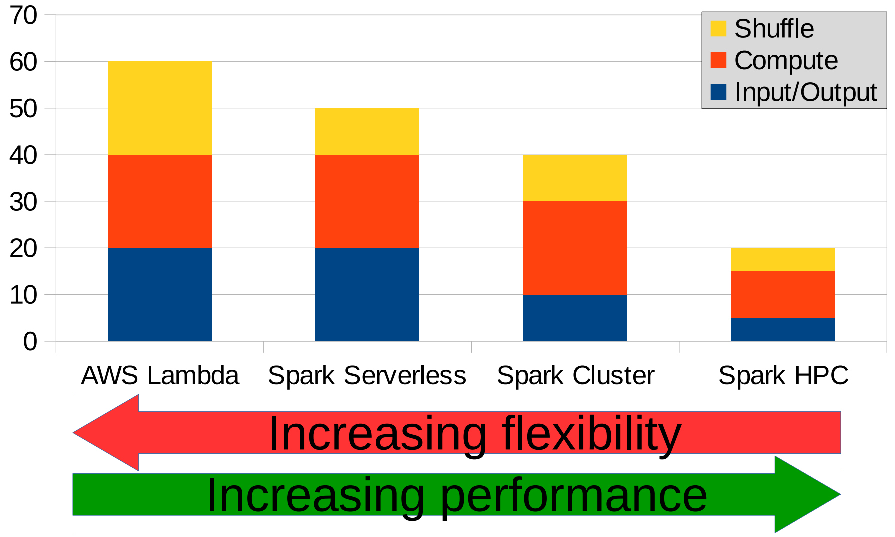
Serverless execution is 3-4x slower than an optimized cluster configuration

Sorting 100GB



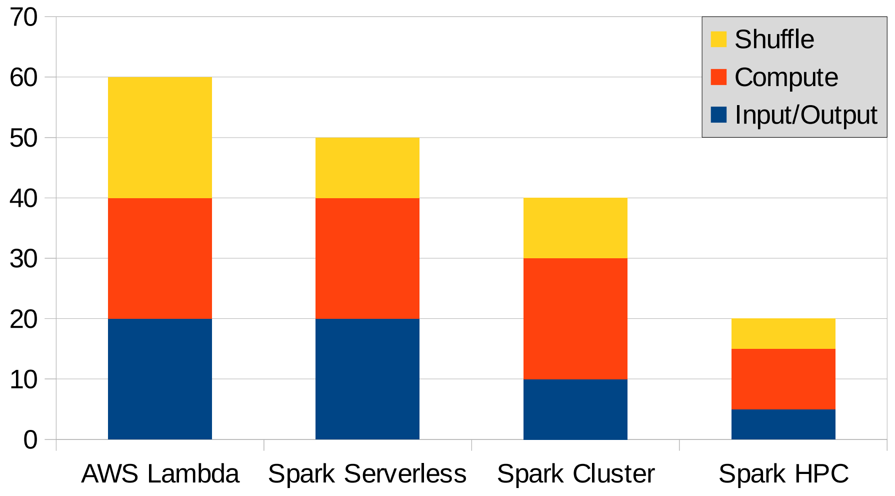
Serverless execution is 3-4x slower than an optimized cluster configuration

Sorting: Is I/O a problem?



With serverless, substantial amount of time is spent on reading from remote storage

Sorting: Is I/O a problem?



With serverless, substantial amount of time is spent on reading from remote storage

What about other workloads?

Example: CoCoa...