EVALUATE THE DASK DISTRIBUTED COMPUTING FRAMEWORK IN RESPECT TO VARIOUS SCIENTIFIC COMPUTING TASKS

EERO VAINIKKO

Course Coordinator

ARTJOM LIND

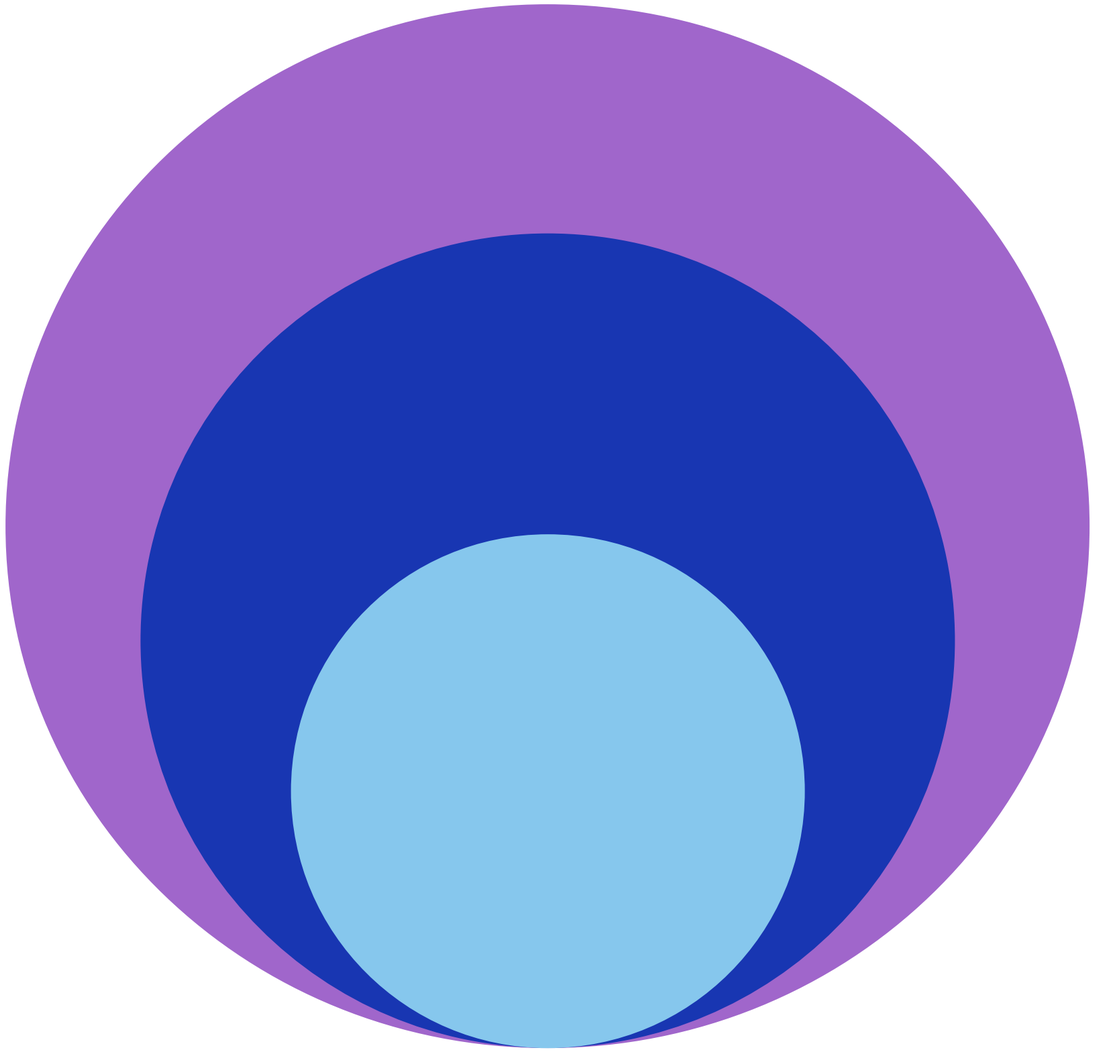
Topic Supervisor

JEYHUN ABBASOV

Student

Agenda

- Recap: Dask & Its Data Structures
- Recap: Experiment 1
- Experiment 2
- Discussion



Recap: Dask & Its Data Structures

Dask is a flexible library for *parallel computing*, *Python-based Big Data engine*.

Dask has *five main data structures*:

Dask Array	is used for the processing of large arrays, provides a distributed clone of the NumPy library
Dask DataFrame	is used to process a large amount of tabular data, parallel composition of Pandas Dataframe
Dask Bag	is a parallel collection of Python objects, like Spark's RDD. offers a programming abstraction similar to the PyToolz library
Dask Delayed	is used for processing arbitrary tasks that don't fit in above APIs.
Futures	is used for processing arbitrary tasks, similar to Delayed. but they operate in real-time ratherthan lazily.

Recap: Experiment 1

Comparison of Dask and other popular Python Frameworks such as Pandas(as a baseline), Modin(Ray), Vaex.

Expertiment 1 - Setup

Baseline

- Pandas

Dataset

- The r/place Parquet dataset - 12GB (22GB uncompressed)

Setup

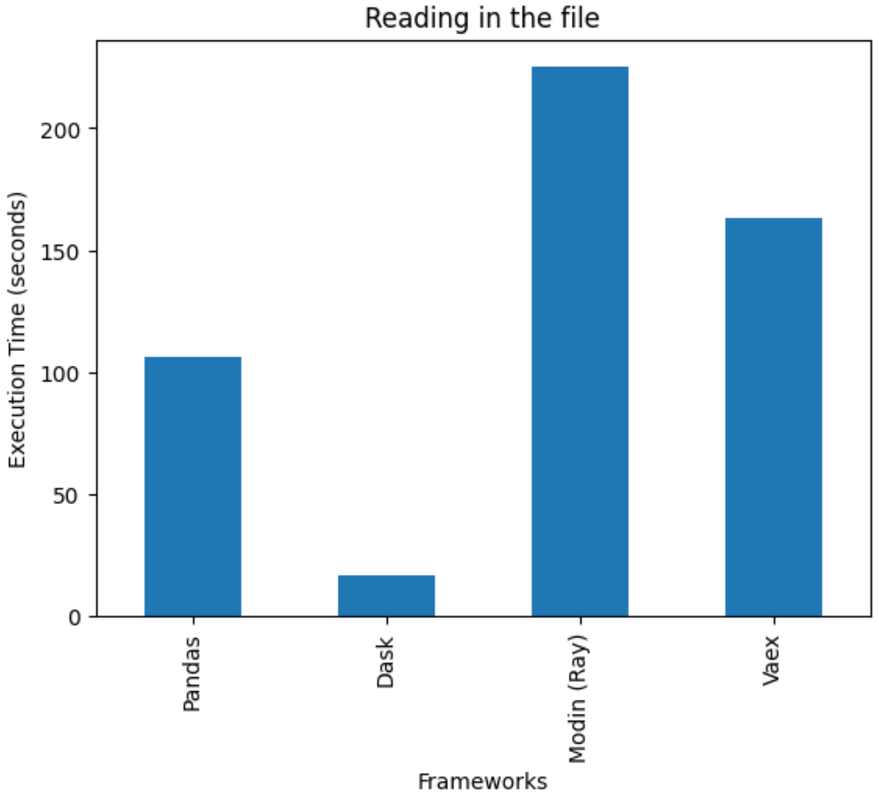
- Processor: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz
- Number of cores: 4
- Memory: 16.0 GB
- Hard Disk: 250 GB

Tests

- Test 1. Reading in the file. Speed comparison
- Test 2. Compute metrics of a column: (mean, std)
- Test 3. Finding the unique value of a column
- Test 4. Cumulative sum of a column
- Test 5. Groupby Aggregation

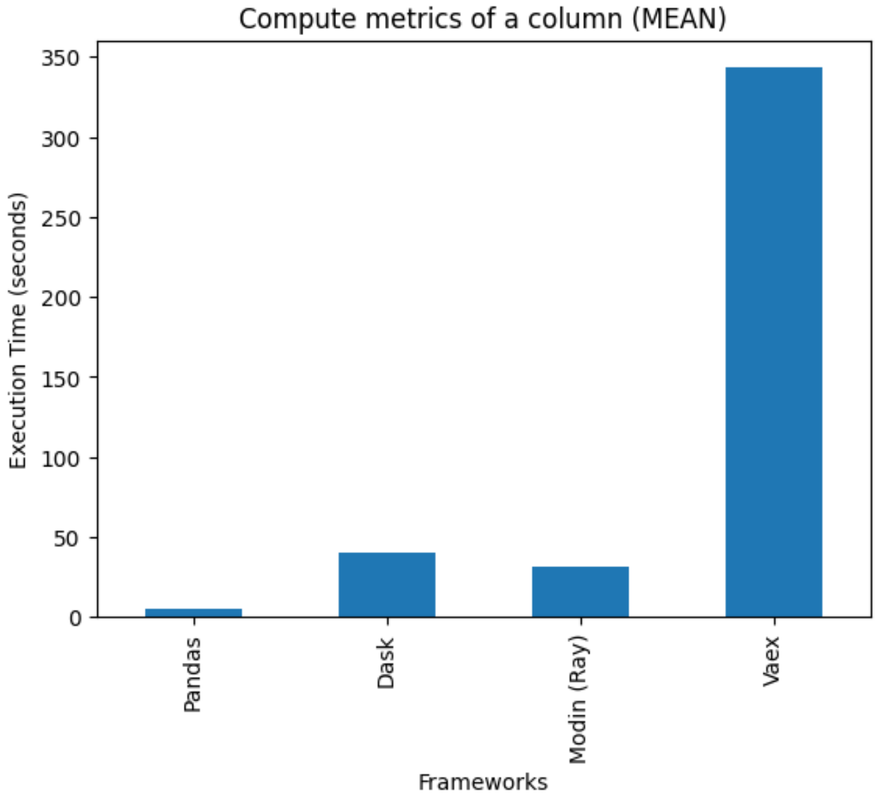
Recap: Experiment 1 - Results

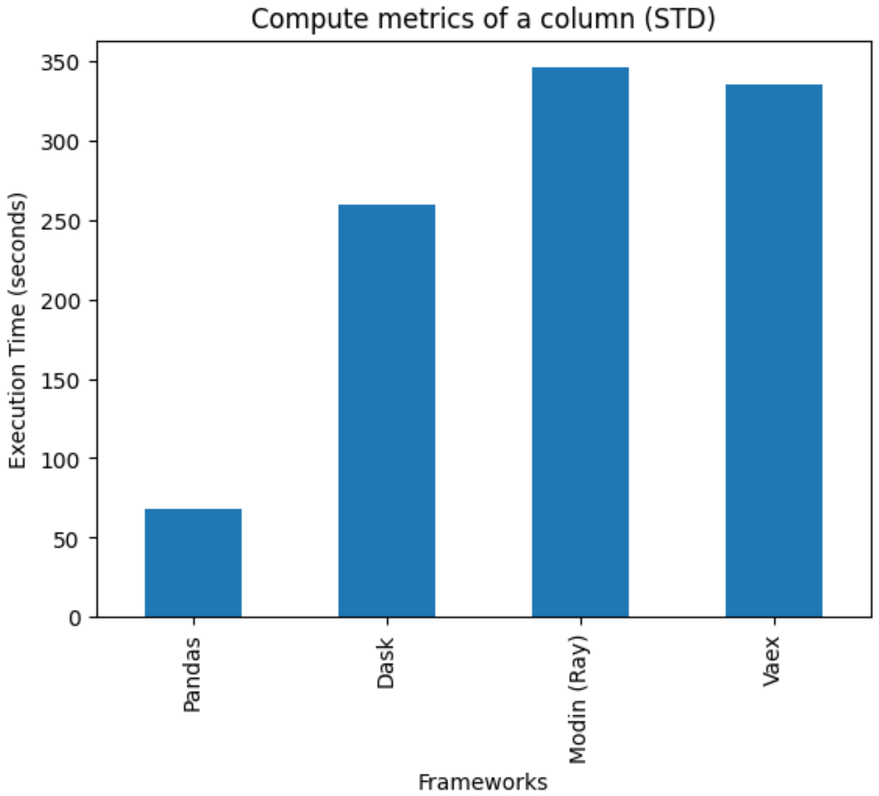
Test 1. Reading in the file. Speed comparison



Recap: Experiment 1 - Results

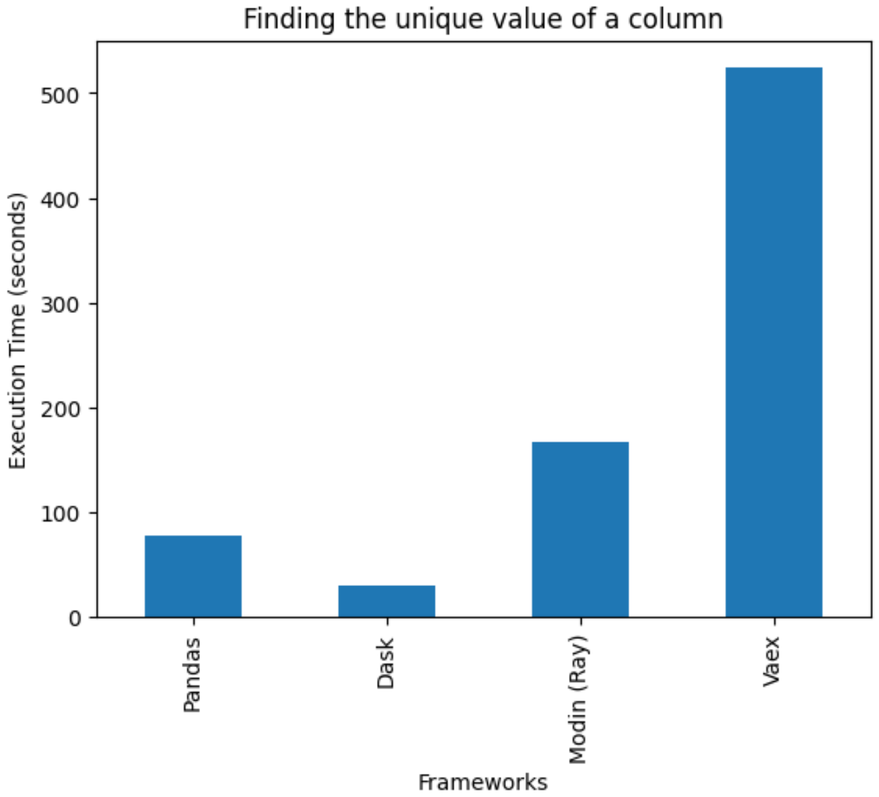
Test 2. Compute metrics of a column: (mean, std)



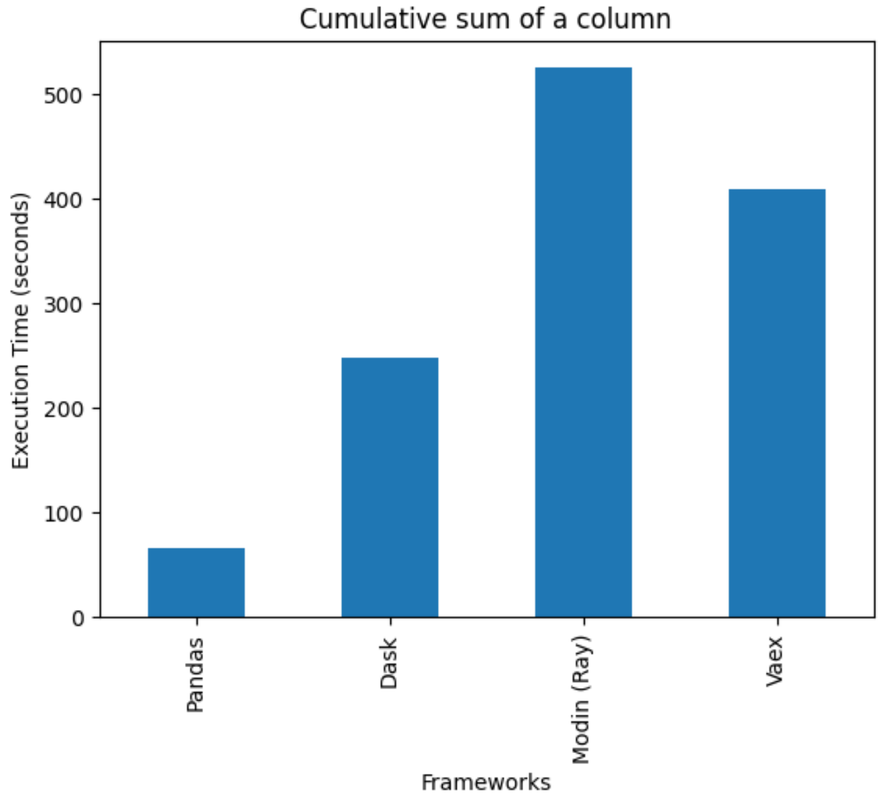


Recap: Experiment 1 - Results

Test 3. Finding the unique value of a column

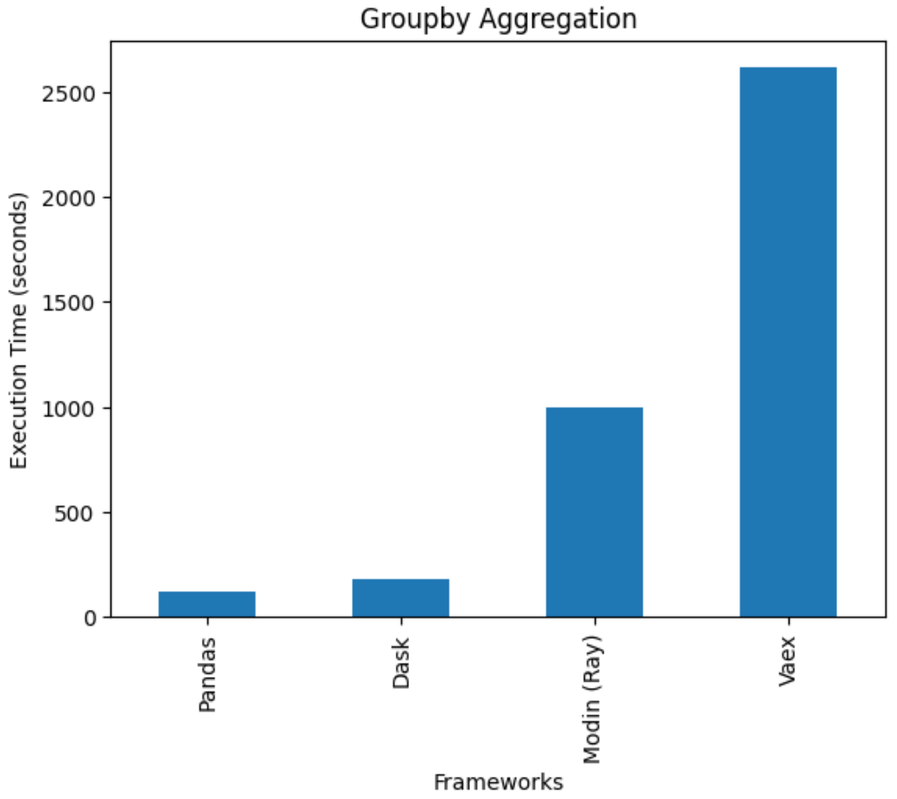


Test 4. Cumulative sum of a column



Recap: Experiment 1 - Results

Test 5. Groupby Aggregation



Experiment 2

Comparison of Apache Spark and Dask Big Data engines in processing neuroimaging pipelines application.

Expertiment 2 - Setup

Application

- Incrementation Application

Dataset

- BigBrain - a three-dimensional image of a human brain with voxel intensities ranging from 0 to 65,535 (Total data size: 81GB)
- Splitted: 30 blocks - 2.7GB, 125 blocks - 0,6GB, 750 blocks - 0.1GB

Setup

- Processor: Intel Xeon Gold 6130, c8-30gb-186, cloud instances with 8 VCPUs
- Memory: 30 GB at 2666MHz
- Hard Disk: CentOS 7.5.1804
- Programming Language: Python

Tests

- Test 1. Number of workers (1, 2, 4, 8)
- Test 2. Number of blocks (30, 125, 750)
- Test 3. Number of iterations (1, 10, 100)
- Test 4. Sleep delays (1, 4, 16, 64)

Experiment 2 - Application

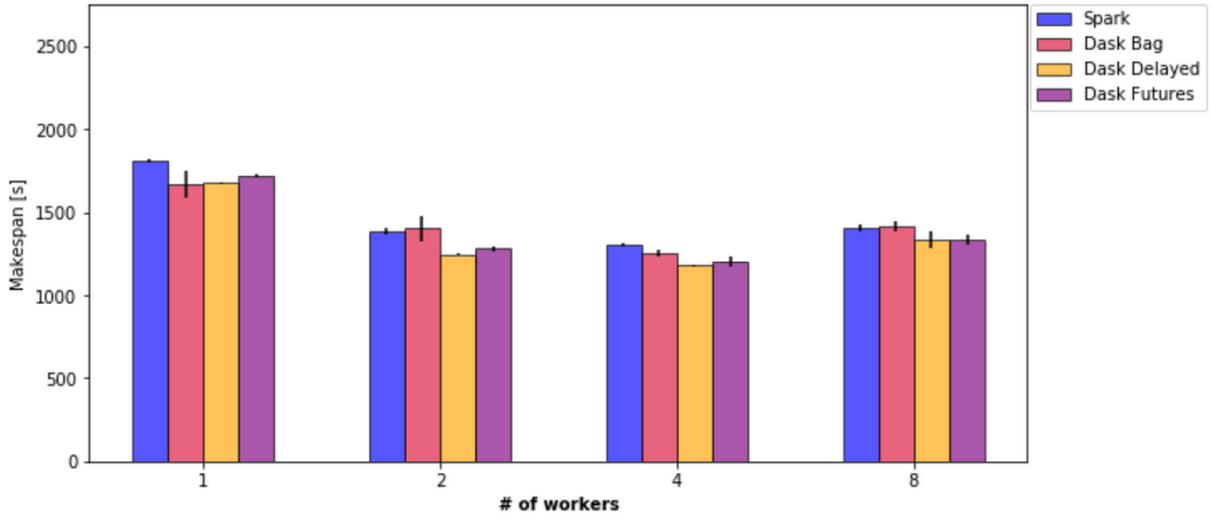
Incrementation Application - *reads blocks* of the BigBrain image from the shared file system, *increments the intensity value of each voxel by 1* to avoid caching effects, *sleeps* for a configurable amount of time to emulate more compute intensive processing, *repeats* his process for a specified number of iterations, and finally *writes the result* as a image back to the shared filesystem.

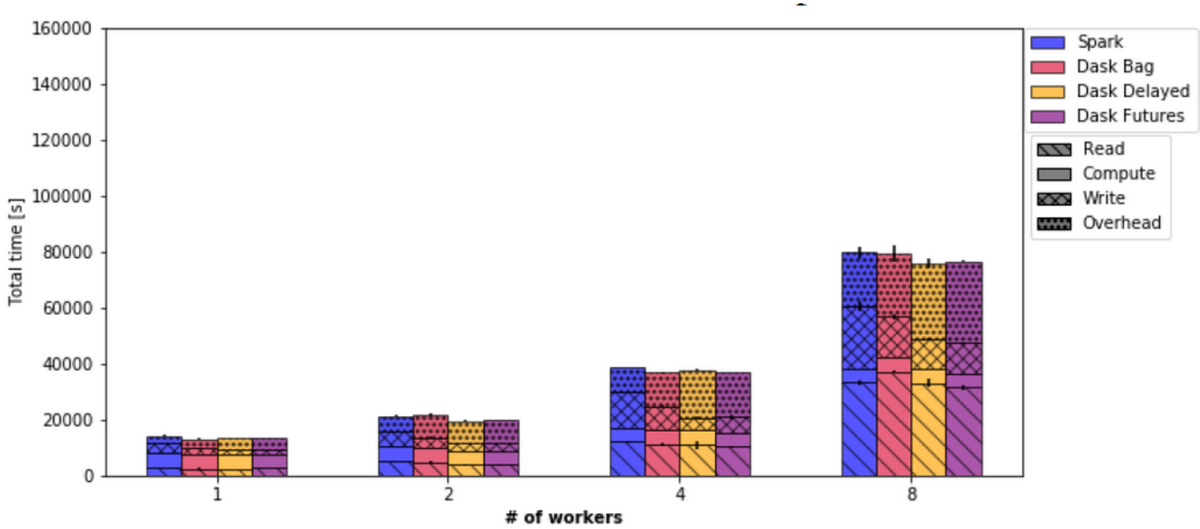
Algorithm Incrementation

Require: x, a sleep delay in float
Require: $file$, a file containing a BigBrain block
Require: fs, NFS mount to write image to
  Read $block$ from $file$
  for each $i \in iterations$ **do**
    for each $block \in image$ **do**
      $block \leftarrow block + 1$
      Sleep x
    end for
  end for
  Write $block$ to fs

Experiment 2 - Results

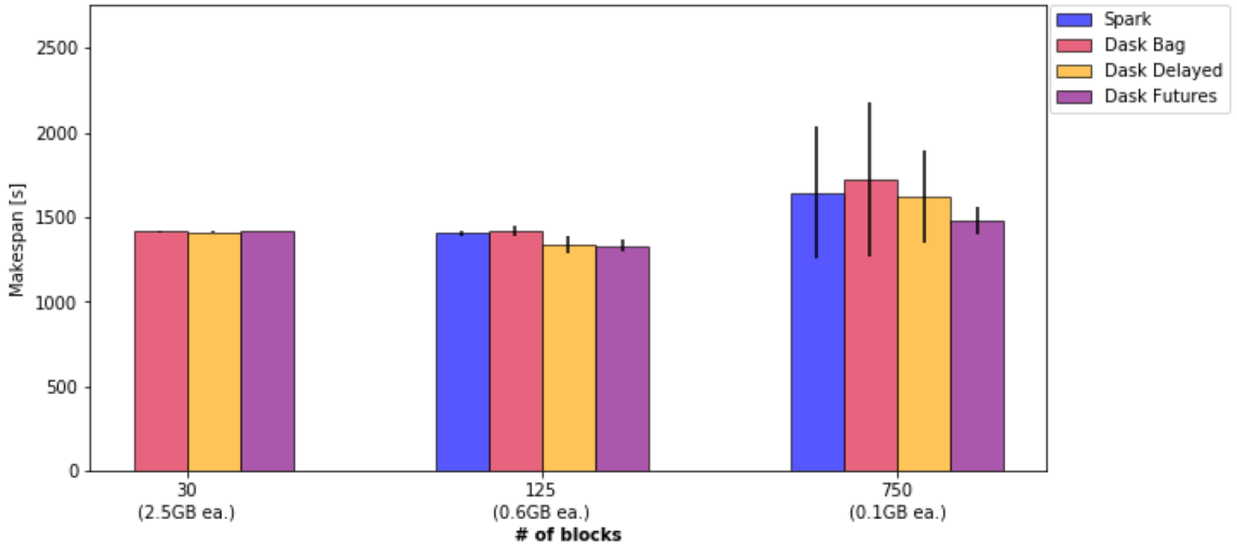
Test 1. Number of workers (1, 2, 4, 8)

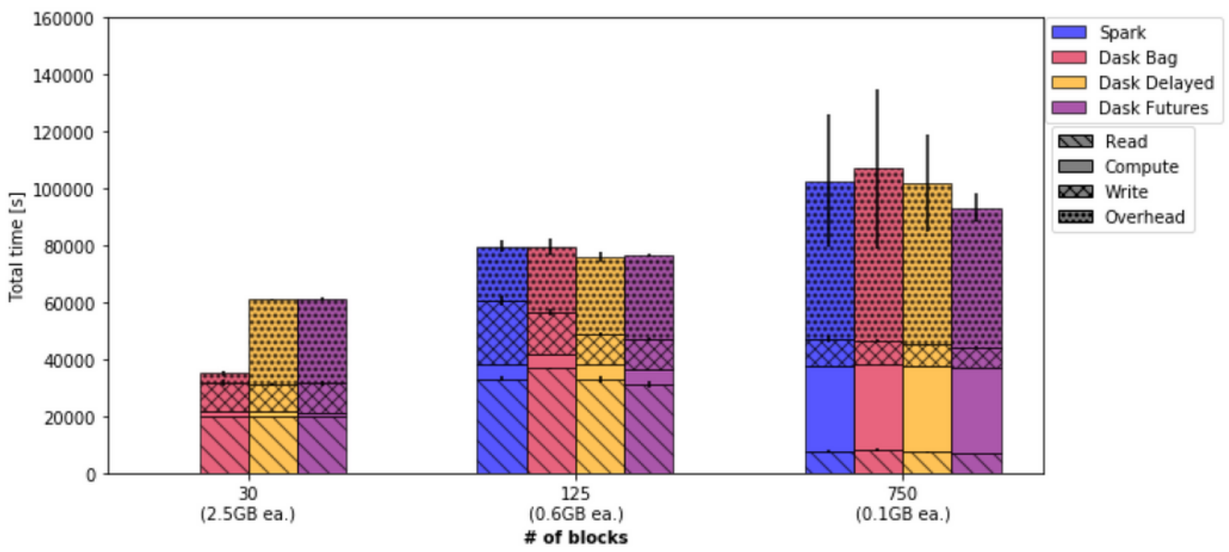




Experiment 2 - Results

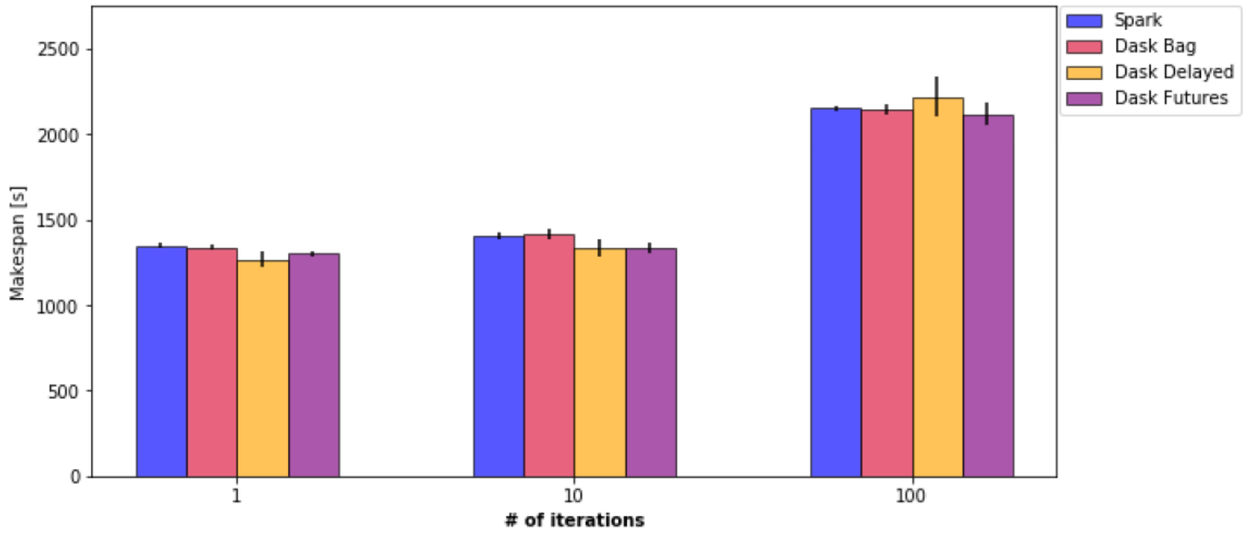
Test 2. Number of blocks (30, 125, 750)

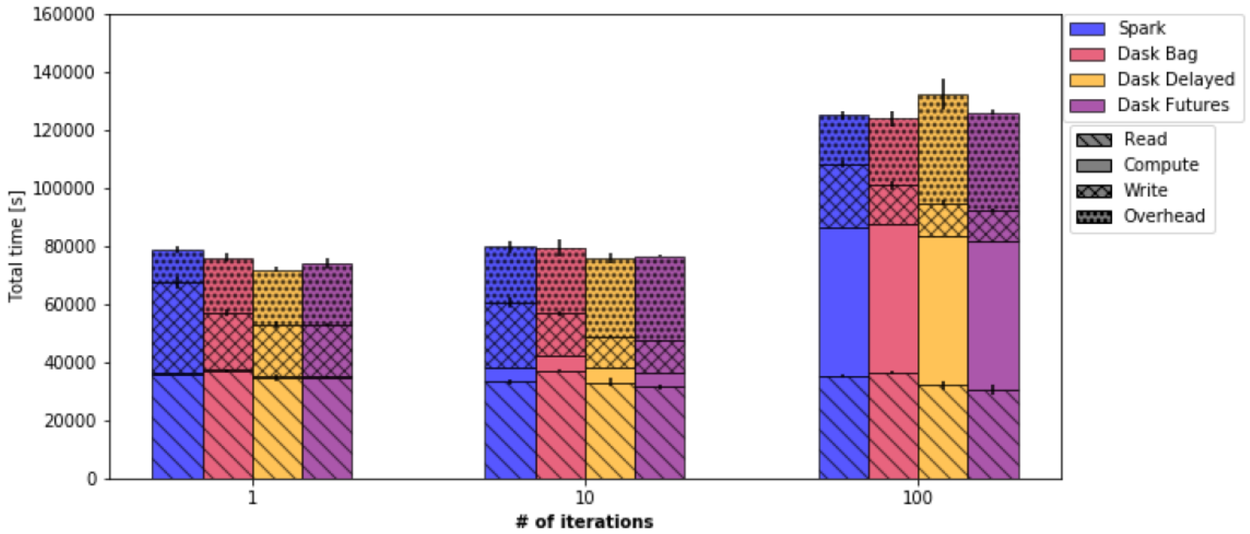




Experiment 2 - Results

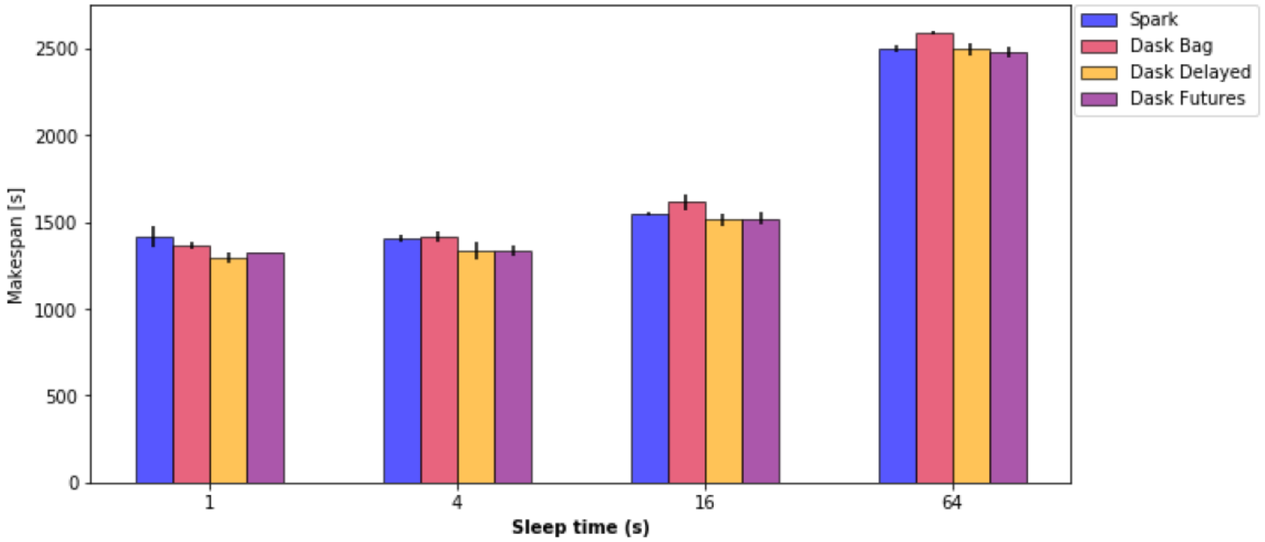
Test 3. Number of iterations (1, 10, 100) - Makespan

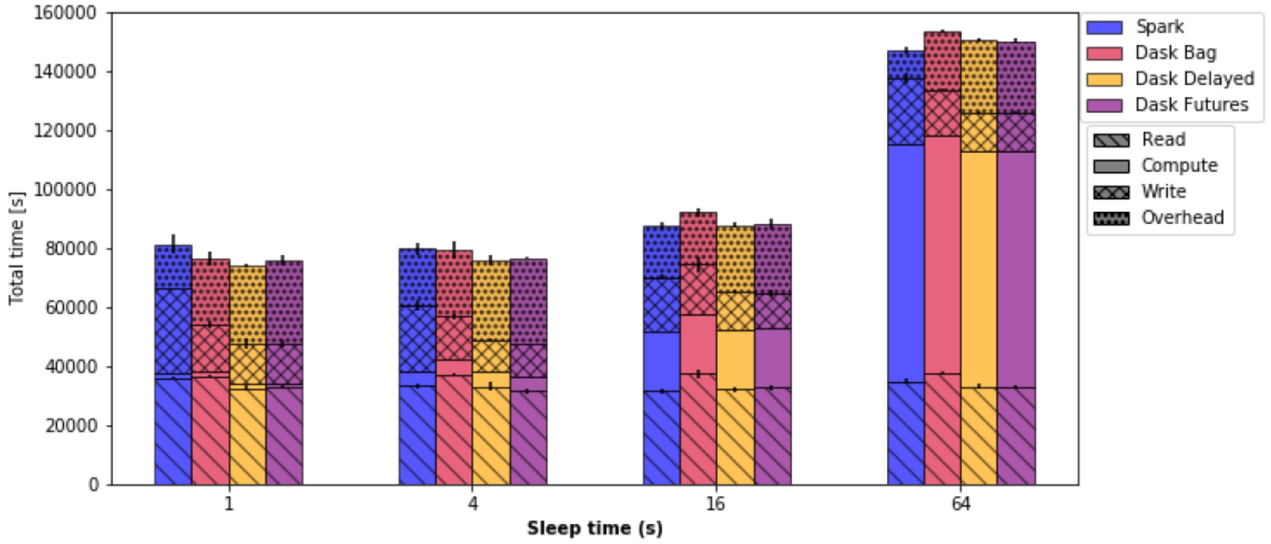




Experiment 2 - Results

Test 4. Sleep delays (1, 4, 16, 64)





Discussion

Presented a evaulation of Dask

Overall, the results show no substantial performance difference between the engines/frameworks

The exp2 results suggest that future research should focus on strategies to reduce the impact of data transfers on applications.

References

[1] A performance comparison of Dask and Apache Spark for data-intensive neuroimaging pipelines

M. Dugr´e, V. Hayot-Sasson and T. Glatard, ”A Performance Comparison of Dask and Apache Spark for Data-Intensive Neuroimaging Pipelines,” 2019 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS), 2019, pp. 40-49, doi: 10.1109/WORKS49585.2019.00010.

[2] Evaluate the Dask distributed computing framework in respect to various scientific computing tasks

- https://github.com/abbcyhn/ut-3-seminar