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

Introduction

The recent rise in data sharing and improved data collection strategies have brought neuroimaging to the Big Data era [1],[2]. Big Data frameworks can bring significant performance improvements to data-intensive neuroimaging pipelines [3].

We perform a benchmark of Dask [4] and Apache Spark [5], two Big Data engines, on neuroimaging dataset.

DASK

APACHE Spark

- Python based
- GIL limitation
- Thread based scheduler

- In-memory computing
- Data locality
- Lazy evaluation
- Fault tolerant

- Java based
- 2GB data block limitation
- Process based scheduler

Dataset

- BigBrain [6], a 81GB 3-D image of a human brain annex
- 39GB of anatomical MRI data from the Consortium for Reliability and Reproducibility (CoRR) dataset [7].

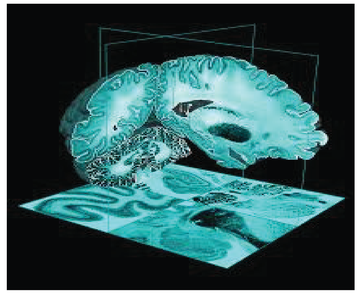


Figure 1: 3-D brain human annex

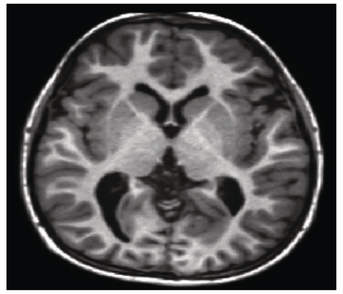


Figure 2: Anatomical MRI image

Infrastructure

- A scheduler node with a 2TB Network File System (NFS)
- Up to 8 workers with 30GB memory and 8 VCPUs each
- 10 Gbit/s network bandwidth
- Dask and Spark default configuration

Applications:

- Voxels Incrementation
- **Histogram (Python & NumPy)**
- BIDS app [8] example

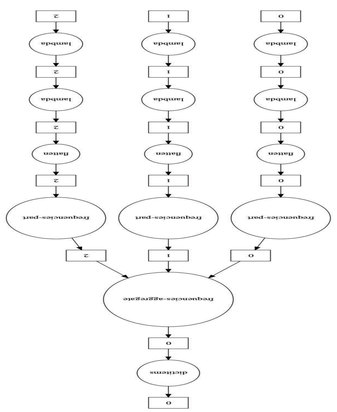


Figure 3: Histogram application flow

Results

Figure 4: Bar and Gantt charts representing the execution time of each task for the Histogram Python application.

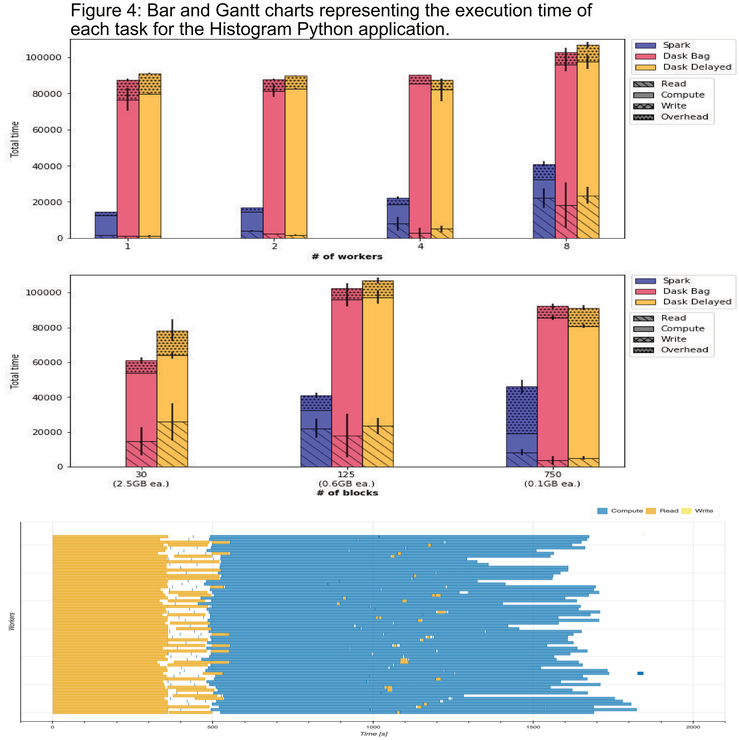
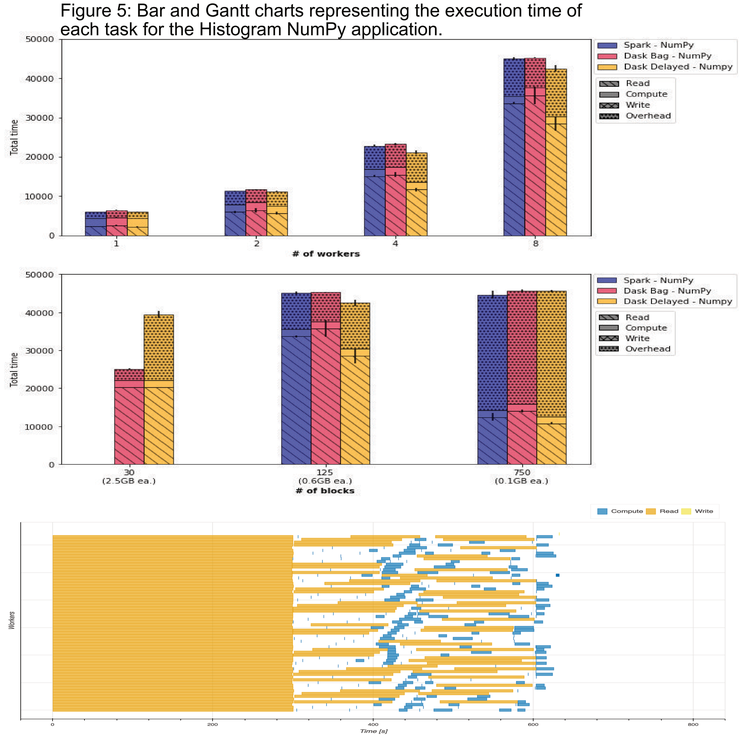


Figure 5: Bar and Gantt charts representing the execution time of each task for the Histogram NumPy application.



Discussion

Overhead / data transfer compensation:

- Almost exact compensation between differences in engine overheads and in data transfer times
- Saturated bandwidths give room for extra engine overhead as long as it desynchronizes data transfers
- Interesting idea to explore in data transfer optimization.

Effect of the share file system:

- Performance of NFS degrades substantially when accessed by concurrent tasks
- Limitations imposed by network and disk bandwidths.

Effect of Python's Global Interpreter Lock (GIL):

- Dask reduces the impact of Python's GIL on parallelism by using external C-based extensions
- Issues remain in Dask to parallelize Python functions
- Dask easily parallelize functions that release the GIL
- Need to be cautious to properly configure the workers depending on the type of functions executed in an application. i.e. multi-threading vs. multi-process workers.

Conclusion

- Our results show no substantial performance difference between the engines.
- Differences in engine overheads do not impact performance due to their impact on data transfer time: higher overheads are almost exactly compensated by a lower transfer time when data transfers saturate the bandwidth.
- These results suggest that future research should focus on strategies to reduce the impact of data transfers on applications.

References

[1] F. Alfaro-Almagro, M. Jenkinson, N. K. Bangerter, J. L. Andersson, L. Griffanti, G. Douaud, S. N. Sotiropoulos, S. Jbabdi, M. Hernandez-Fernandez, E. Vallee, D. Vidaurre, M. Webster, P. McCarthy, C. Rorden, A. Daducci, D. C. Alexander, H. Zhang, I. Dragonu, P. M. Matthews, K. L. Miller, and S. M. Smith, "Image processing and Quality Control for the first 10,000 brain imaging datasets from UK Biobank," Neuroimage, vol. 166, pp. 400 – 424, 2018.

[2] J. D. Van Horn and A. W. Toga, "Human neuroimaging as a Big Data science," Brain imaging and behavior, vol. 8, no. 2, pp. 323–331, 2014.

[3] V. Hayot-Sasson, S. T. Brown, and T. Glatard, "Performance Evaluation of Big Data Processing Strategies for Neuroimaging," in 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), Larnaca, Cyprus, 2019.

[4] Matthew Rocklin, "Dask: Parallel Computation with Blocked algorithms and Task Scheduling," in Proceedings of the 14th Python in Science Conference, Kathryn Huff and James Bergstra, Eds., 2015, pp. 126–132.

[5] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin, A. Ghodsi, J. Gonzalez, S. Shenker, and I. Stoica, "Apache Spark: A Unified Engine for Big Data Processing," Commun. ACM, vol. 59, no. 11, pp. 56–65, Oct. 2016.

[6] K. Amunts, C. Lepage, L. Borgeat, H. Mohlberg, T. Dickscheid, M.-É. Rousseau, S. Bludau, P.-L. Bazin, L. B. Lewis, A.-M. Oros-Peusquens, N. J. Shah, T. Lippert, K. Zilles, and A. C. Evans, "BigBrain: An Ultrahigh-Resolution 3D Human Brain Model," Science, vol. 340, no. 6139, pp. 1472–1475, 2013.

[7] X.-N. Zuo, J. S. Anderson, P. Bellec, R. M. Birn, B. B. Biswal, J. Blautzik, J. C. Breitner, R. L. Buckner, V. D. Calhoun, F. X. Castellanos et al., "An open science resource for establishing reliability and reproducibility in functional connectomics," Scientific data, vol. 1, p. 140049, 2014.

[8] K. J. Gorgolewski, F. Alfaro-Almagro, T. Auer, P. Bellec, M. Capotă, M. M. Chakravarty, N. Wv. Churchill, A. L. Cohen, R. C. Craddock, G. A. Devenyi et al., "BIDS apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods," PLoS computational biology, vol. 13, no. 3, p. e1005209, 2017.

Pre-print available on arXiv.org

Code available on GitHub