NumS Update
08/31/2021

Since our last update, we have focused heavily on improving the NumS open source project. Please find below some major updates.

1 Open source and applications updates

1. We have added fallback support to NumPy for functions that aren't yet available.
2. Implementation of multinomial logistic regression.
3. Implementation of various matrix inversion algorithms.
4. Implemented L-BFGS with line search with backtracking and strong wolfe conditions [3].
5. Added algorithm for automatic block partitioning.
6. We're exploring and modeling climatology data made available through the Amazon Sustainable Data Initiative (ASDI) using NumS and Modin.

2 Core optimizations

We've made several optimizations to element-wise and reduction operations – these are the optimizations that helped NumS outperform other libraries. Element-wise operations scale perfectly. See Figure 1 for scaling results of QR decomposition and logistic regression.

3 HPC library comparisons

We have preliminary results comparing NumS' dgemm to SLATE [2] and ScaLAPACK [1]. These results show that NumS' RPC overhead has minimal impact on dgemm for datasets that fit comfortably into memory. See Figure 2 for these results.

4 Updated distributed systems comparison

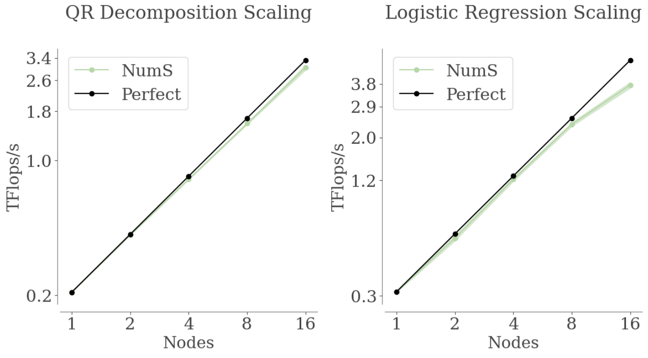
We've updated comparisons to Dask and Spark, updating to the latest versions for each library, and comparing both NumS and Spark tuned for optimal performance. See Figure 3 for comparison to Dask, and Figure 4 for comparison to Spark.

5 GPU benchmarks

On the GPU, we've performed data-parallel and model-parallel GPU-based benchmarks for the multi-layered perceptron. See Figure 5 for results.

6 Future design plans

Based on benchmarks that show significant performance improvements for custom fused operations for NumS, and cost of I/O when transmitting large data structures using Ray's RPCs, we've created specs to add support for operator fusion and mutable arrays. We're currently working on integrating these improvements.



(a) QR Decomposition Scaling. **(b)** Logistic Regression Scaling.

Figure 1. Scaling of QR decomposition and logistic regression in TFlops/s.

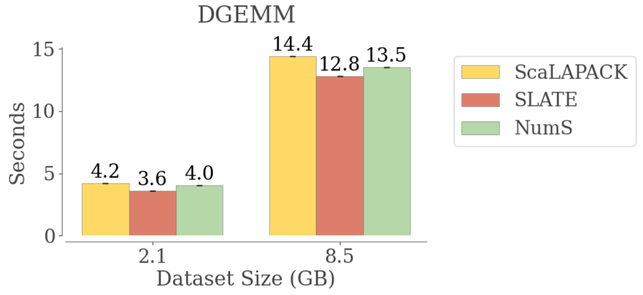
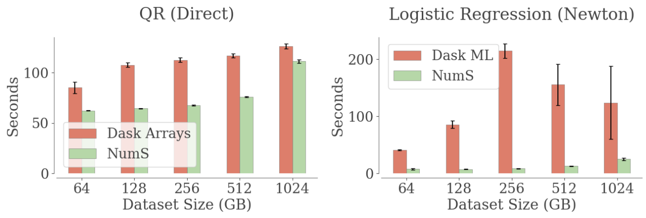


Figure 2. SLATE vs. ScaLAPACK vs. NumS on 2.1 GB dataset (2 nodes) and 8.5 GB dataset (4 nodes).

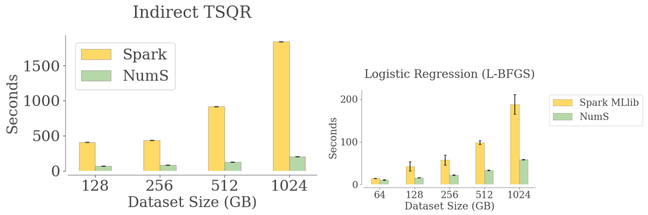


(a) QR Decomposition. **(b)** Logistic Regression.

Figure 3. Comparison between Ray/NumS to Dask Arrays and Dask ML.

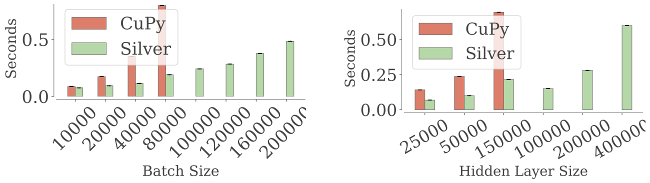
References

[1] L Susan Blackford, Jaeyoung Choi, Andy Cleary, Eduardo D'Azevedo, James Demmel, Inderjit Dhillon, Jack Dongarra, Sven Hammarling,



(a) QR Decomposition. **(b)** Logistic Regression.

Figure 4. Comparison between Ray/NumS to Spark MLlib.



(a) Data Parallel **(b)** Model Parallel

Figure 5. Data parallelism and model parallelism for training a multi-layer perceptron.

Greg Henry, Antoine Petitet, et al. 1997. *ScaLAPACK users' guide*. Vol. 4. Siam.

[2] Mark Gates, Jakub Kurzak, Ali Charara, Asim YarKhan, and Jack Dongarra. 2019. SLATE: Design of a Modern Distributed and Accelerated Linear Algebra Library. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (Denver, Colorado) *(SC '19)*. Association for Computing Machinery, New York, NY, USA, Article 26, 18 pages. https://doi.org/10.1145/3295500.3356223

[3] Dong C. Liu and Jorge Nocedal. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Math. Program.* 45, 1–3 (Aug. 1989), 503–528.