
Comparing Data Systems

Dask and Ray

Elisabeth Waldron, Chaitali Harge, Lingzhen Zhu, Fadumo Hussein

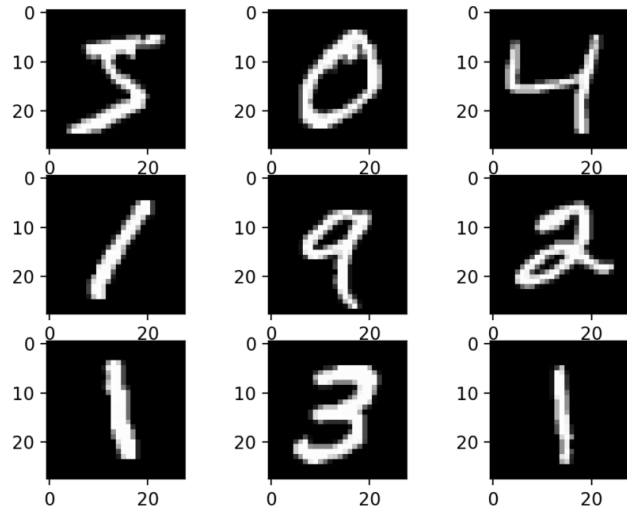
Objective

- Motivation: :
 - Understand performance of distributed computing systems.
 - Focus on Ray and Dask for K-Means clustering and CNN modeling in digit recognition.
- Problem Statement::
 - Identify efficient distributed computing system for large-scale data processing.
 - Factors: execution time and scalability.

Data Set

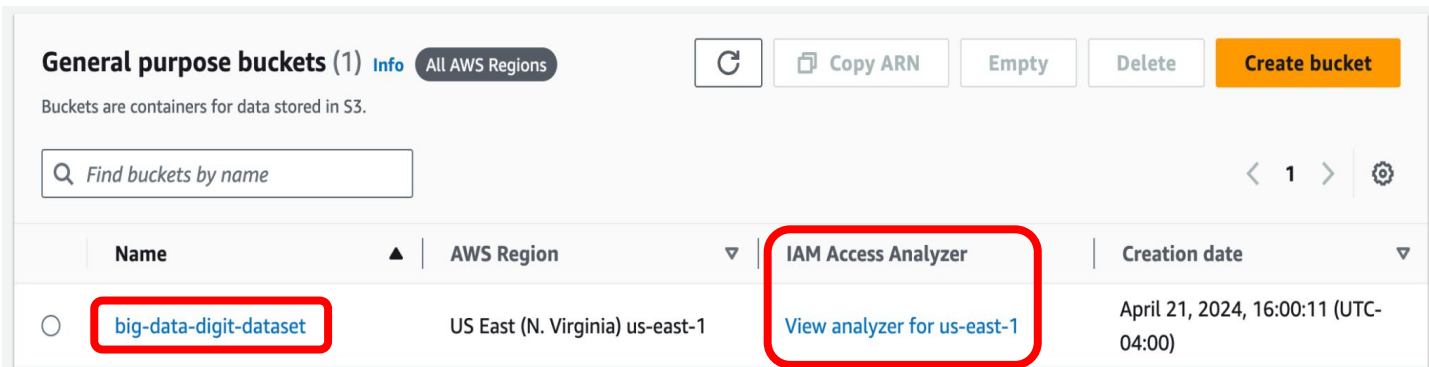
The dataset consists of 1797 8x8 images, each depicting a handwritten digit.


We plan to utilize the scikit-learn digit dataset and implement various algorithms including K-Means and CNN to assess the performance of these distributed computing systems.




Preprocessing

- Created 4 EC2 instances
- Created a bucket in AWS S3 to upload the data
- Attached an IAM role to the EC2 instances with permissions to access AWS S3
- Defined function to read data from S3 into Dask and Ray



General purpose buckets (1) [Info](#) **All AWS Regions**  [Copy ARN](#) [Empty](#) [Delete](#) [Create bucket](#)

Buckets are containers for data stored in S3.

< 1 > 

	Name ▲	AWS Region ▼	IAM Access Analyzer	Creation date ▼
<input type="radio"/>	big-data-digit-dataset	US East (N. Virginia) us-east-1	View analyzer for us-east-1	April 21, 2024, 16:00:11 (UTC-04:00)

Why AWS S3 ?

- Amazon S3 provides remote access from anywhere with internet connectivity, facilitating easy data sharing and collaboration.
- Its virtually unlimited scalability ensures seamless storage expansion to accommodate growing data volumes.
- High durability and redundancy via data replication ensure data integrity and availability, reducing the risk of data loss.
- Cost-effective pricing models allow for efficient management of storage expenses, aligning with budget constraints and optimizing resource allocation.

Ray

Ray is an open-source distributed computing framework designed for scaling and automating the deployment of AI applications. Here are some pros of working with Ray:

- Scalability: Efficiently utilize resources across multiple machines or clusters.
- Flexibility: Supports a wide range of use cases, from simple parallelism to complex algorithms.
- High Performance: Optimized for task scheduling and communication overhead.
- Fault Tolerance: : Handle failures gracefully, ensuring seamless recovery.
- Ease of Use: Simple and intuitive API abstracts complexities of distributed computing.
- Compatibility: : Integrates seamlessly with TensorFlow, PyTorch, and other ML frameworks.
- Cost Efficiency: Reduces infrastructure costs through dynamic scaling and resource optimization.

Ray with 4-Workers

	Host / Worker Process name	State	ID	IP / PID	Actions	CPU [?]	Memory [?]	GPU [?]	GRAM	Object Store Memory
>	ip-172-31-31-167	ALIVE	192c5...	172.31.31.167 (Head)	Log	46.1%	1.35GB/7.63GB(17.7%)	N/A	N/A	1.79MB/2.03GB(0.1%)
>	ip-172-31-26-250	ALIVE	61281...	172.31.26.250	Log	2.8%	511.49MB/7.63GB(6.5%)	N/A	N/A	0.0000B/2.20GB(0.0%)
>	ip-172-31-31-233	ALIVE	a9274...	172.31.31.233	Log	3.7%	534.85MB/7.63GB(6.8%)	N/A	N/A	0.0000B/2.20GB(0.0%)
>	ip-172-31-26-225	ALIVE	c7e11...	172.31.26.225	Log	4.1%	534.59MB/7.63GB(6.8%)	N/A	N/A	0.0000B/2.20GB(0.0%)

>	ip-172-31-31-167	ALIVE	192c5...	172.31.31.167 (Head)	Log	7.8%	1.54GB/7.63GB(20.2%)	N/A	N/A	3.20MB/2.03GB(0.2%)
>	ip-172-31-26-250	ALIVE	61281...	172.31.26.250	Log	39.1%	748.60MB/7.63GB(9.6%)	N/A	N/A	359.51KB/2.20GB(0.0%)
>	ip-172-31-31-233	ALIVE	a9274...	172.31.31.233	Log	75.7%	772.32MB/7.63GB(9.9%)	N/A	N/A	359.51KB/2.20GB(0.0%)
>	ip-172-31-26-225	ALIVE	c7e11...	172.31.26.225	Log	100%	1018.16MB/7.63GB(13.0%)	N/A	N/A	719.01KB/2.20GB(0.0%)

Ray with 3-Workers

>	ip-172-31-31-167	ALIVE	192c5...	172.31.31.167 (Head)	Log	<div><div>96.7%</div></div>	<div><div>1.60GB/7.63GB(21.0%)</div></div>	N/A	N/A	<div><div>3.20MB/2.03GB(0.2%)</div></div>
>	ip-172-31-26-250	ALIVE	61281...	172.31.26.250	Log	<div><div>20.8%</div></div>	<div><div>1012.23MB/7.63GB(13.0%)</div></div>	N/A	N/A	<div><div>719.01KB/2.20GB(0.0%)</div></div>
>	ip-172-31-26-225	ALIVE	c7e11...	172.31.26.225	Log	<div><div>14.8%</div></div>	<div><div>1004.66MB/7.63GB(12.9%)</div></div>	N/A	N/A	<div><div>719.01KB/2.20GB(0.0%)</div></div>
>	ip-172-31-31-233	DEAD	a9274...	172.31.31.233		<div><div>3.9%</div></div>	<div><div>537.10MB/7.63GB(6.9%)</div></div>	N/A	N/A	<div><div>0.0000B/2.20GB(0.0%)</div></div>

>	ip-172-31-31-167	ALIVE	192c5...	172.31.31.167 (Head)	Log	<div><div>99.2%</div></div>	<div><div>2.45GB/7.63GB(32.1%)</div></div>	N/A	N/A	<div><div>5.89MB/2.03GB(0.3%)</div></div>
>	ip-172-31-26-250	ALIVE	61281...	172.31.26.250	Log	<div><div>5.6%</div></div>	<div><div>1.01GB/7.63GB(13.3%)</div></div>	N/A	N/A	<div><div>719.01KB/2.20GB(0.0%)</div></div>
>	ip-172-31-26-225	ALIVE	c7e11...	172.31.26.225	Log	<div><div>99.8%</div></div>	<div><div>1.28GB/7.63GB(16.8%)</div></div>	N/A	N/A	<div><div>1.40MB/2.20GB(0.1%)</div></div>
>	ip-172-31-31-233	DEAD	a9274...	172.31.31.233		<div><div>3.9%</div></div>	<div><div>537.10MB/7.63GB(6.9%)</div></div>	N/A	N/A	<div><div>0.0000B/2.20GB(0.0%)</div></div>

Ray with 2-Workers

	Host / Worker Process name	State	ID	IP / PID	Actions	CPU [?]	Memory [?]	GPU [?]	GRAM	Object Store Memory
>	ip-172-31-31-167	ALIVE	192c5...	172.31.31.167 (Head)	Log	97.1%	3.12GB/7.63GB(40.9%)	N/A	N/A	5.89MB/2.03GB(0.3%)
>	ip-172-31-26-225	ALIVE	c7e11...	172.31.26.225	Log	99.4%	1.51GB/7.63GB(19.8%)	N/A	N/A	719.01KB/2.20GB(0.0%)
>	ip-172-31-26-250	DEAD	61281...	172.31.26.250		15.3%	1.99GB/7.63GB(26.1%)	N/A	N/A	9.60MB/2.03GB(0.5%)
>		DEAD	a9274...			0%		N/A	N/A	1.40MB/2.20GB(0.1%)

>	ip-172-31-31-167	ALIVE	192c5...	172.31.31.167 (Head)	Log	72.2%	2.28GB/7.63GB(29.9%)	N/A	N/A	5.89MB/2.03GB(0.3%)
>	ip-172-31-26-225	ALIVE	c7e11...	172.31.26.225	Log	55.5%	1.01GB/7.63GB(13.2%)	N/A	N/A	719.01KB/2.20GB(0.0%)
>	ip-172-31-26-250	DEAD	61281...	172.31.26.250		15.3%	1.99GB/7.63GB(26.1%)	N/A	N/A	6.40MB/2.03GB(0.3%)
>		DEAD	a9274...			0%		N/A	N/A	719.01KB/2.20GB(0.0%)

Evaluation of Ray

- The KNN model showed relatively consistent times across different worker counts maybe due to KNN computations are less computationally intensive.
- The ResNet model demonstrated significant variability in execution times, highlighting Ray's effectiveness in handling more computationally intensive tasks through parallel processing.
- With an increase in workers, there was a more balanced and effective distribution of workload, leading to better utilization of computational resources.

Dask

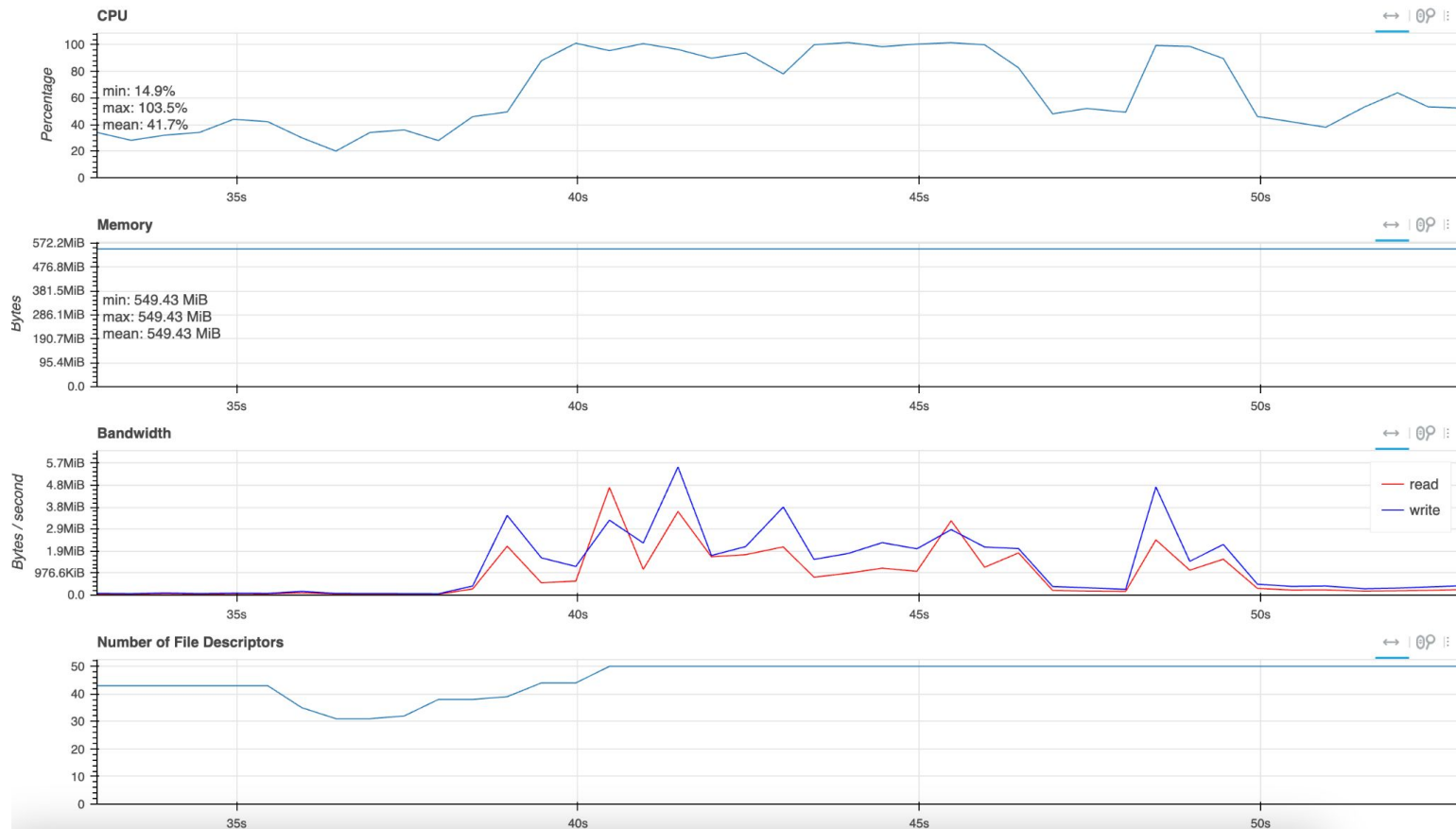
Advantages

- Dask's parallel processing capabilities offer significant advantages due to its lazy evaluation strategy, which defers computation until necessary, optimizing memory usage. Additionally, Dask distributes tasks across multiple workers, enabling efficient parallelization and scalability across distributed computing environments.

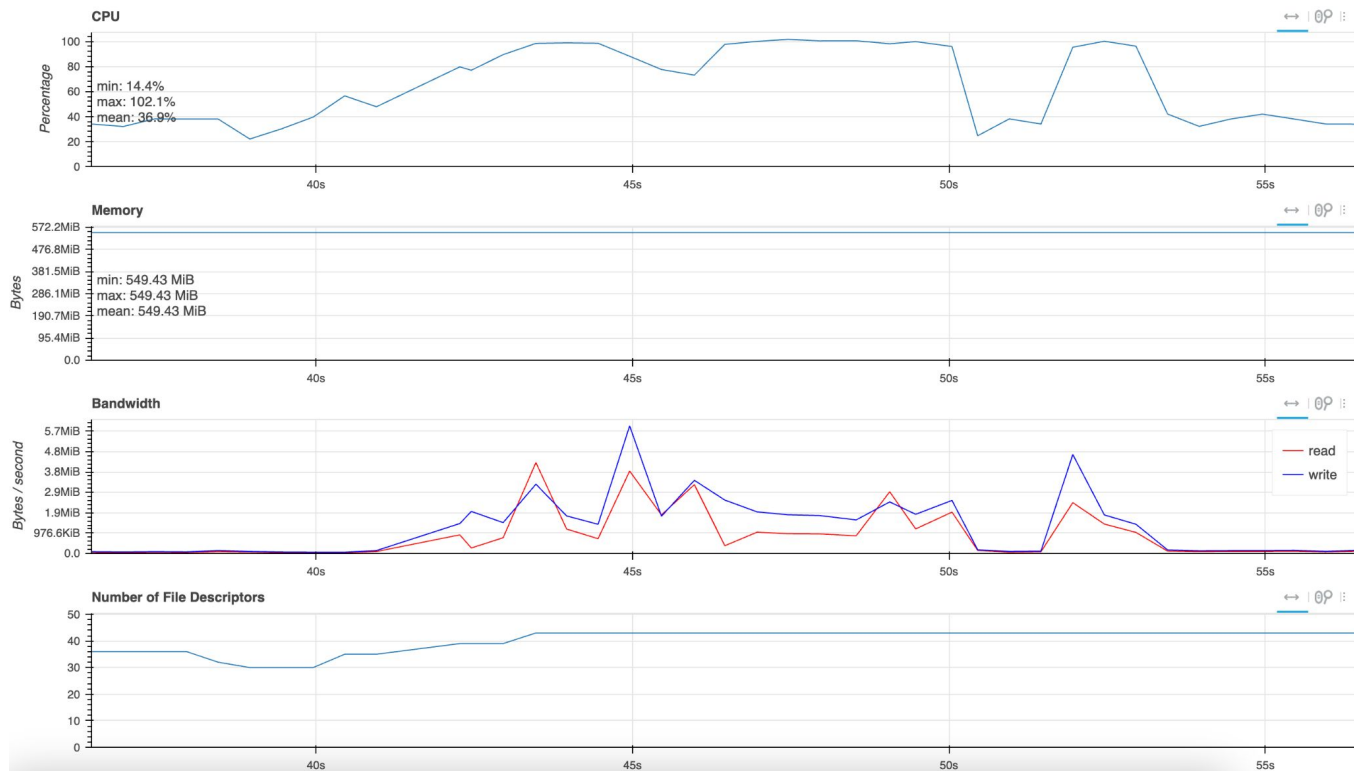
Disadvantages

- While Dask offers impressive parallel processing capabilities, it also comes with some drawbacks, such as increased computational and overhead costs. These costs arise from the need to manage task scheduling, data movement between workers, and potential bottlenecks, which can impact overall performance and scalability, particularly in complex or heterogeneous computing environments.

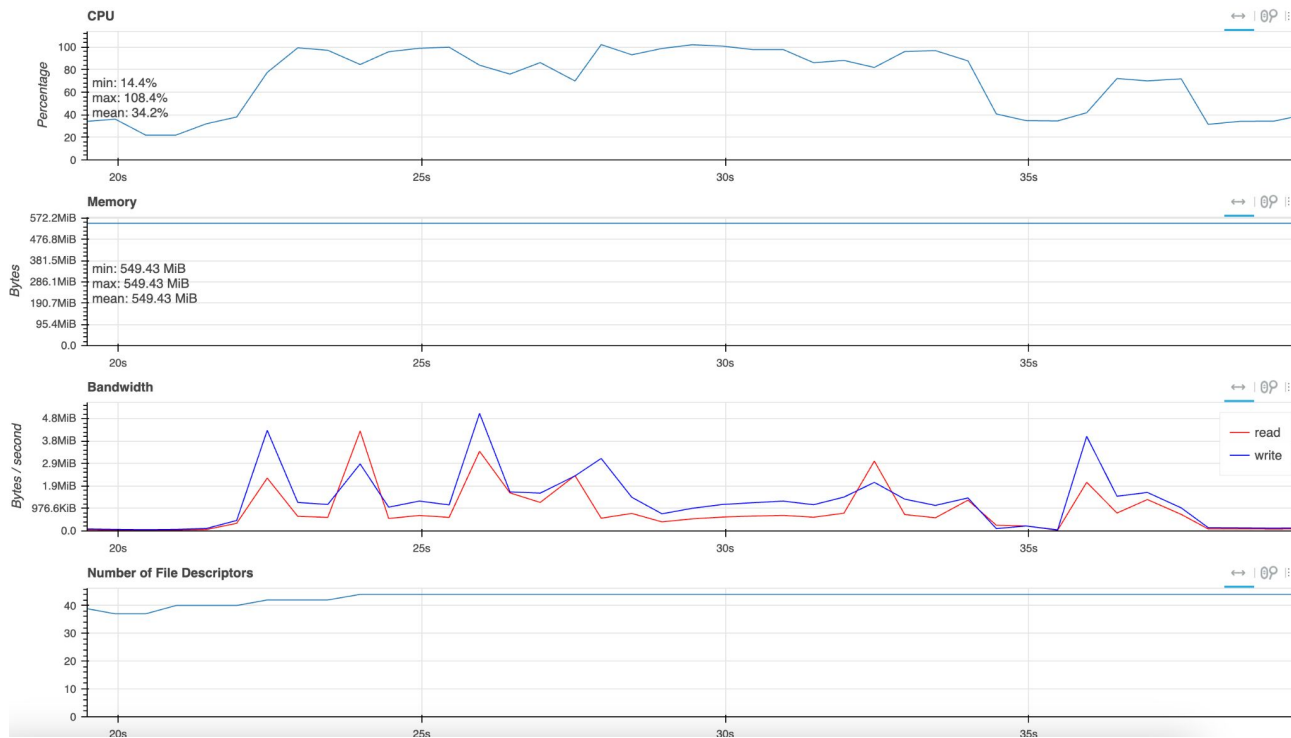
Dask with 6-Workers



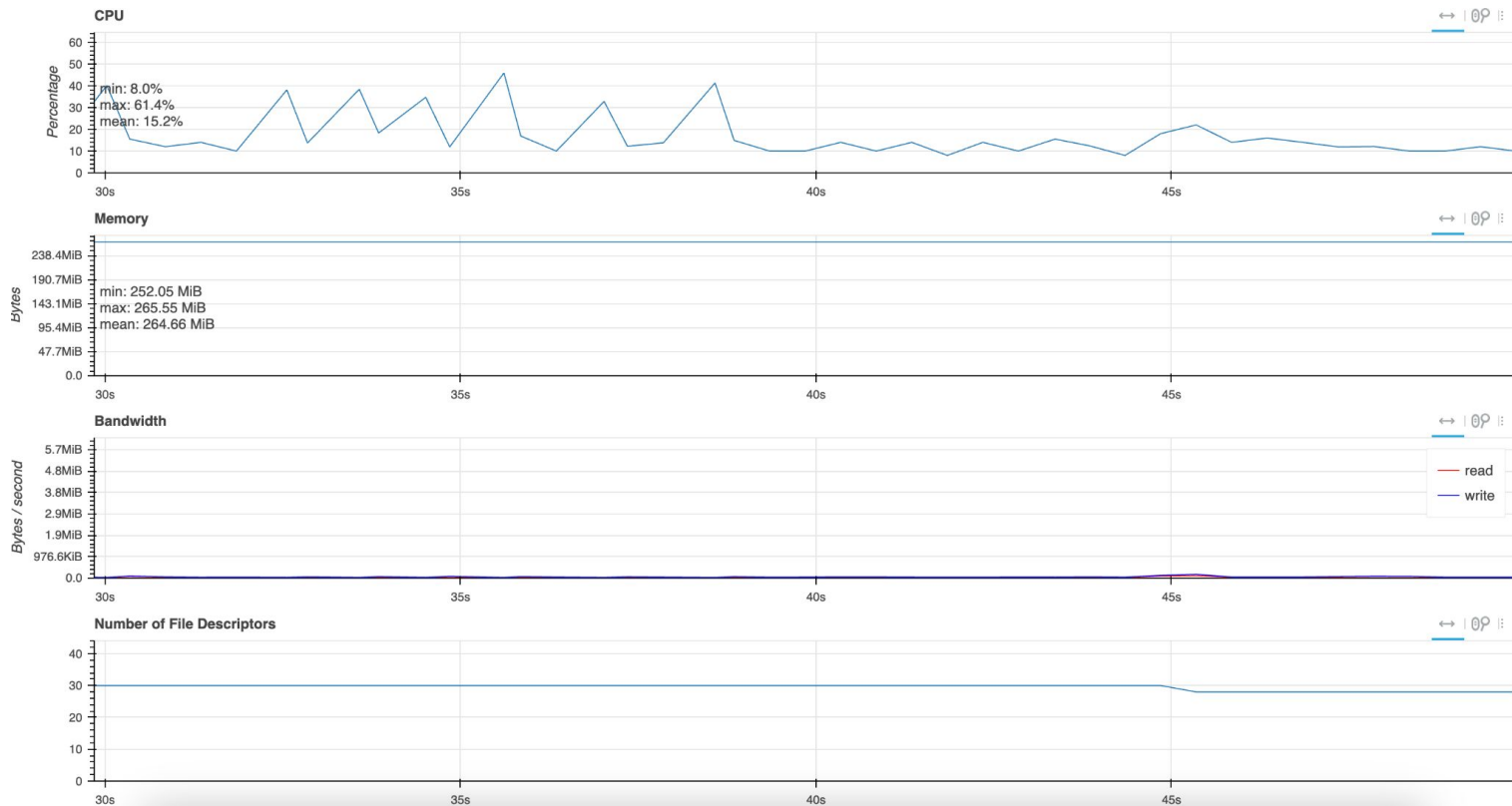
Dask with 4-Workers



Dask with 2-Workers



Dask with 1-Workers



Run Time Performance

	Dask			Ray		
	2	4	6	2	3	4
KNN for Classification	3.78 s	2.73 s	2.69 s	3.25 s	3.19 s	3.42 s
ResNet18	15.6 s	13.2 s	13.1 s	30.6 s	25.7 s	13.1 s

Conclusion

- Enhanced computational efficiency, especially with multiple workers on Dask
- Ray's performance metrics consistent but lacked efficiency compared to Dask, especially with complex models like ResNet18
- Integration with Amazon S3 introduced significant overhead, leading to operational instability and frequent worker crashes
- Challenges in managing deployment and associated costs need careful consideration for real-world applications
- Resource utilization metrics (CPU and memory usage) provided insights into efficiency of resource management across different worker setups

Resources

Dataset:

- MNIST Digit Dataset

Digits Classification Exercise:

- https://scikit-learn.org/stable/auto_examples/exercises/plot_digits_classification_exercise.html#sphx-glr-auto-examples-exercises-plot-digits-classification-exercise-py

AWS S3:

- <https://aws.amazon.com/s3/getting-started/>

Thank You!

