

DS 5110 Big Data Systems Term Project

Team 1: Harold Haugen, George Shoriz, Zack Lisman, Dan Anthony

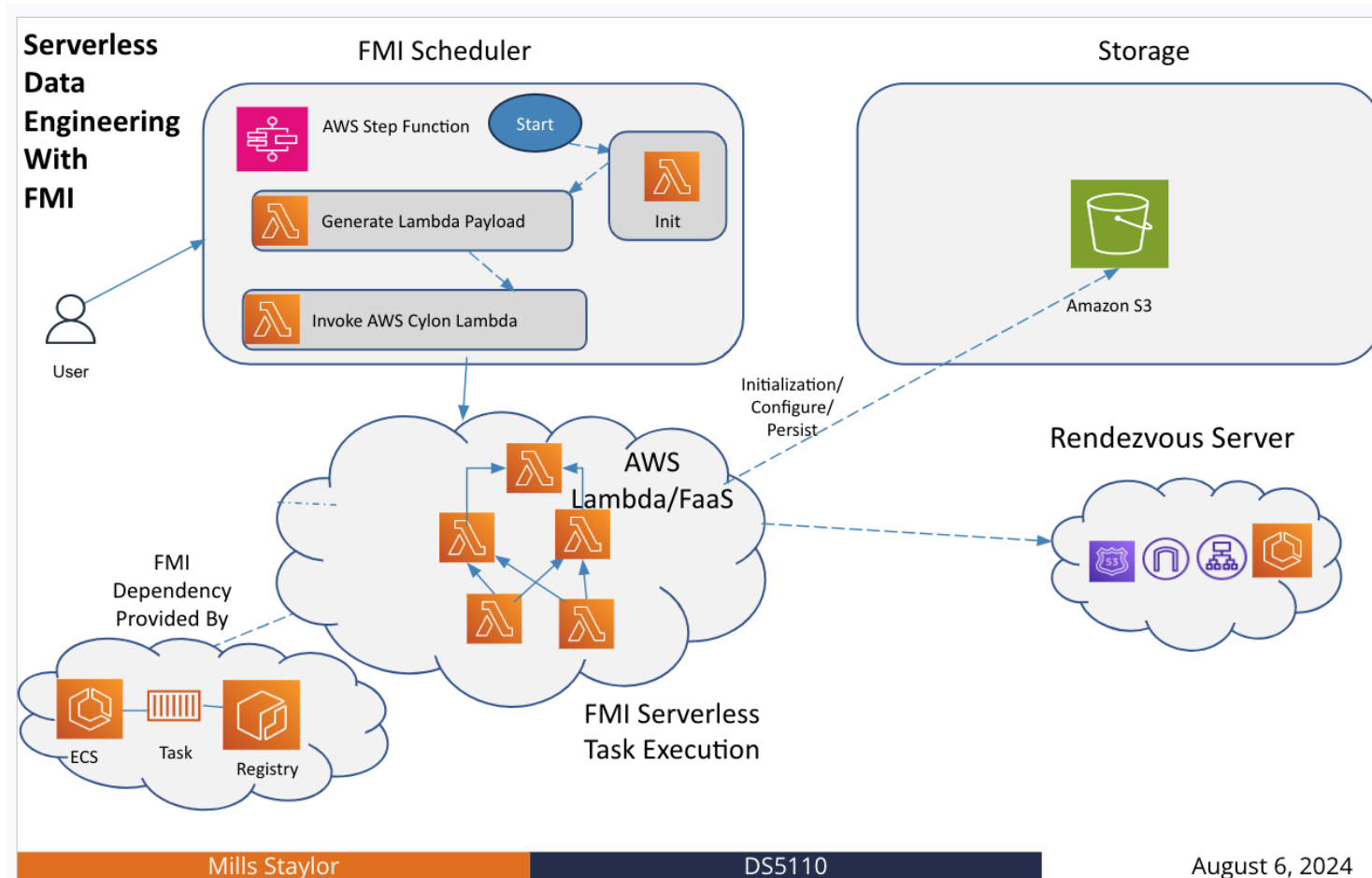
11.24.2024

Overview of Project

- ▶ 1: Step Function (Midterm Presentation)
 - ▶ Facilitates orchestration of FMI lambda functions
 - ▶ Measure execution time and cost
- ▶ 2: Rendezvous Server
 - ▶ Establishes communication between AWS lambda functions
- ▶ 3: AI for Astronomy Inference: Run Inference.py file
 - ▶ resized_images
- ▶ 4: Integrate w/CosmicAI
- ▶ 5: Review Data Partitioning Code/Results

Recap of Overall FMI/Lambda Structure

- Our understanding of the overall design of the AWS Step Function



Building the State Machine & S3 Bucket

- ▶ Building the State Machine (Cosmic_AI_Team_One)
 - Initiated build by making a copy of the 'cosmicai' state machine.
 - Set the IAM roles based on the MyStateMachine-e5ydt2afc design.

IAM role ARN

[arn:aws:iam::211125778552:role/service-role/StepFunctions-MyStateMachine-e5ydt2afc-role-z0ymrut89](#)

StepFunctions-MyStateMachine-e5ydt2afc-role-z0ymrut89 [Info](#)

Permissions policies (6) [Info](#) [Refresh](#) [Simulate](#) [Remove](#)

You can attach up to 10 managed policies.

Filter by Type All types

<input type="checkbox"/>	Policy name Info	Type	Attached entities
<input type="checkbox"/>	AmazonS3FullAccess	AWS managed	13
<input type="checkbox"/>	AWSLambda_FullAccess	AWS managed	4
<input type="checkbox"/>	AWSStepFunctionsFullAccess	AWS managed	2
<input type="checkbox"/>	cloudwatchaccess	Customer inline	0
<input type="checkbox"/>	LambdaInvokeScopedAccessPolicy-2cc7913...	Customer managed	1
<input type="checkbox"/>	XRayAccessPolicy-fcb083fb-726c-4c08-bf1...	Customer managed	1

Building the State Machine & S3 Bucket

► S3 Bucket (team-one-s3-cosmic)

- Established our own S3 bucket to house the inference related .py files, data and directories for outputs.
- Mimicked the directory structure after the 'cosmicai2' S3 bucket.
 - Directories include: Results, Scripts and the payload.json file

team-one-s3-cosmic Info

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (3) Info



Copy S3 URI

Copy URL

Download

Open

Delete

Actions

Create folder

Upload

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

Find objects by prefix

< 1 > ⚙

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	payload.json	json	November 18, 2024, 20:27:14 (UTC-05:00)	551.0 B	Standard
<input type="checkbox"/>	results/	Folder	-	-	-
<input type="checkbox"/>	scripts/	Folder	-	-	-

Experimental Design & Trials

► Changes to FMI Code for experimentation

- For the FMI run, we utilized the 'inference.py' model from the 'cosmicai2' S3 bucket.
- We performed an assessment of the code to identify parameters for experimentation and review.
 - We executed code runs with changes to Batch Size and World_Size.
- Because the FMI run for the 'inference.py' model is CPU based, changes to World_Size does not distribute the workload, therefore we focused our performance assessment on the various Batch Size runs.

Four inference .py files we created with 'batch_size' set to 256, 512, 1024 and 2048.

Note: 'prj_dir' is directed to the '/scripts/Anomaly Detection/' in our 'team-one-s3-cosmic' S3 bucket

```
187 # Pathes and other inference hyperparameters can be adjusted below
188 if __name__ == '__main__':
189     prj_dir = '/tmp/scripts/Anomaly Detection/' #adjust based on your system's directory
190     parser = argparse.ArgumentParser()
191     parser.add_argument('--batch_size', type=int, default=256)
192     parser.add_argument('--data_path', type=str, default=f'{prj_dir}Inference/resized_inference.pt')
193     parser.add_argument(
194         '--model_path',
195         type=str,
196         default=f'{prj_dir}Fine_Tune_Model/Mixed_Inception_z_VITAE_Base_Img_Full_New_Full.pt',
197     )
198     parser.add_argument('--device', type=str, default='cpu') # To run on GPU, put cuda, and on CPU put cpu
199
200     parser.add_argument('--plot_path', type=str, default=f'{prj_dir}Plots/')
201
202     parser.add_argument('--rank', type=int, **environ_or_required('RANK', required=False))
203     parser.add_argument('--world_size', type=int, **environ_or_required('WORLD_SIZE', required=False))
204     args = parser.parse_args()
205
206     engine(args)
```

Experimental Design & Trials

- ▶ Updates to the Lambda function code for Payload initialization.
 - We modified:
 - ‘bucket’ to point to our ‘team-one-s3-cosmic’ S3 bucket.
 - ‘world_size’ to understand if it could be run without partitioning. This required a change from CPU to Cuda in order to process.
 - ‘script’ to point to our various .py modified code files.

Payload Code Changes

Payload
The JSON that you want to provide to your Lambda function.

```
5  "Parameters": {
6      "Payload": {
7          "bucket": "team-one-s3-cosmic",
8          "world_size": "1",
9          "object_type": "folder",
10         "S3_object_name": "scripts/Anomaly Detection",
11         "data_path": "/tmp/scripts/Anomaly Detection/Inference/resized_inference.pt",
12         "script": "/tmp/scripts/Anomaly Detection/Inference/inference.py"
13     },
14 }
```

Successful Executions

<input type="checkbox"/>	Cosmic_Trial_Two_Thurs0907	✔ Succeeded	Nov 21, 2024, 21:07:27	Nov 21, 2024, 21:07:36	00:00:08.739
<input type="checkbox"/>	Cosmic_Trial_Two_Thurs0901	✔ Succeeded	Nov 21, 2024, 21:01:25	Nov 21, 2024, 21:01:34	00:00:08.829
<input type="checkbox"/>	Cosmic_Trial_Two_Thurs0834	✔ Succeeded	Nov 21, 2024, 20:34:51	Nov 21, 2024, 20:35:03	00:00:12.802
<input type="checkbox"/>	Cosmic_Trial_Two_Thurs0828	✔ Succeeded	Nov 21, 2024, 20:28:54	Nov 21, 2024, 20:29:05	00:00:11.347
<input type="checkbox"/>	Cosmic_Trial_Two_Thurs0650	✔ Succeeded	Nov 21, 2024, 18:50:42	Nov 21, 2024, 18:50:55	00:00:13.123

Step Function Demo

Retrieval of Output

- ▶ Capture of Inference.py performance from .json file outputs into our S3 Bucket.
- ▶ Use of 'CloudWatch/ Logs Insights' to retrieve Lambda function performance metrics.
- ▶ Designed a Retrieval Notebook (located on GitHub) to isolate our Lambdas Function's executions.

Inference.py Performance - Json Outputs

Key Metrics:

```
execution_info = {
    'total_cpu_time (seconds)': total_time,
    'total_cpu_memory (MB)': total_cpu_memory,

    # Average execution time per batch
    'execution_time (seconds/batch)': avg_time_batch,
    'num_batches': num_batches, # Number of batches
    'batch_size': batch_size, # Batch size
    'device': device, # Selected device

    # Throughput in bits per second (using total_time for all batches)
    'throughput_bps': total_data_bits / total_time,

    # Number of samples processed per second
    'sample_persec': num_samples / total_time,
    'cpu_info': get_cpu_info(),
    'ram_info (GB)': get_ram_info(),
    'avg_profile': str(avg),
    'self_cpu_memory (MB)': avg.self_cpu_memory_usage/1e6
}
```

CloudWatch

Logs Insights [Info](#)

/aws/lambda/cosmic-executor ✕

Key Metrics (definitions from ChatGPT):

2. **RequestId: f623ffe3-78ed-4d0e-a9de-ac00b9e9d71c :**
 - A unique identifier assigned to this specific Lambda invocation. It can be used for tracking and troubleshooting the execution.
3. **Duration: 6434.80 ms :**
 - The actual time (in milliseconds) it took for the Lambda function to run from start to finish.
4. **Billed Duration: 7381 ms :**
 - The duration AWS charges for. Lambda billing is rounded up to the nearest 100 ms, with an additional charge for the initialization phase if it was a cold start.
5. **Memory Size: 10240 MB :**
 - The memory size configured for the Lambda function. In this case, the function was allocated **10,240 MB (10 GB)** of memory.
6. **Max Memory Used: 645 MB :**
 - The peak amount of memory used by the function during execution. Here, it used **645 MB**, which is far below the allocated limit.
7. **Init Duration: 945.56 ms :**
 - The time it took to initialize the Lambda function during a **cold start** (i.e., when the function was started for the first time or after being inactive for a while). Initialization includes setting up the runtime environment, loading code, and initializing dependencies.

Results

► JSON Results:

```
"total_cpu_time (seconds)": 4.879611039999991,  
"total_cpu_memory (MB)": 14315.891824,  
"execution_time (seconds/batch)": 1.2199027599999976,  
"num_batches": 4,  
"batch_size": 256,  
"device": "cpu",  
"throughput_bps": 34449434.3139285,  
"sample_persec": 209.85279187334612,
```

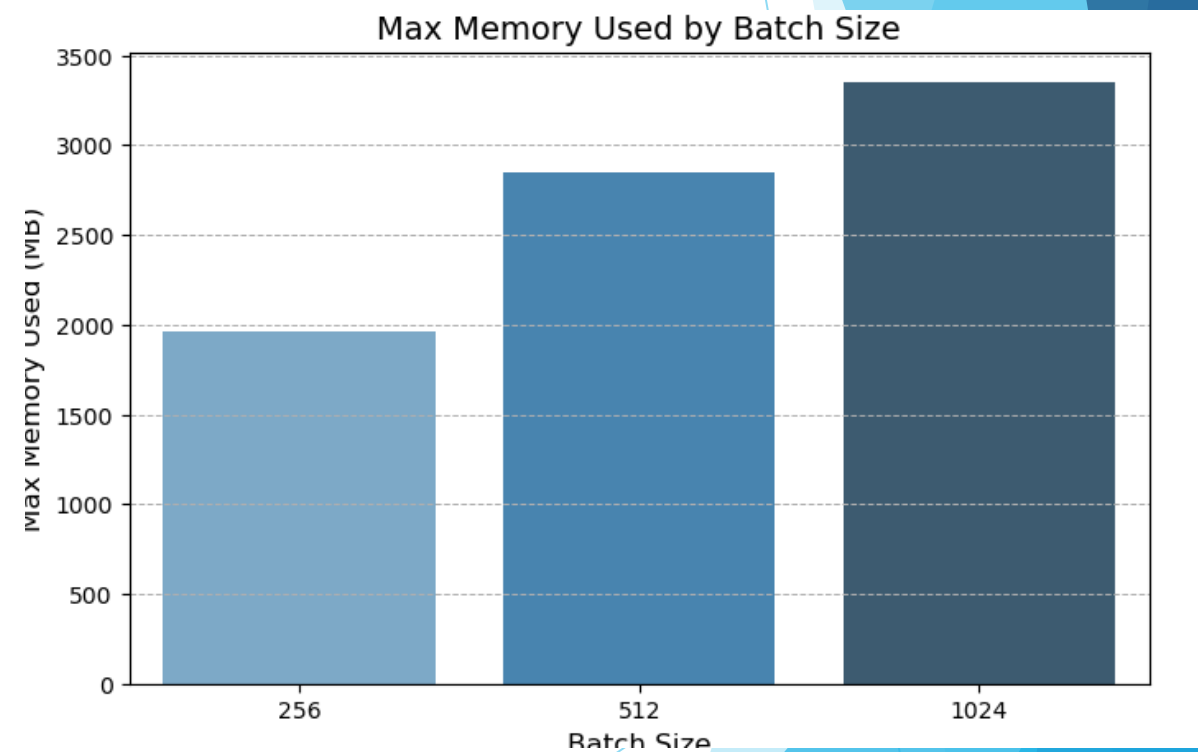
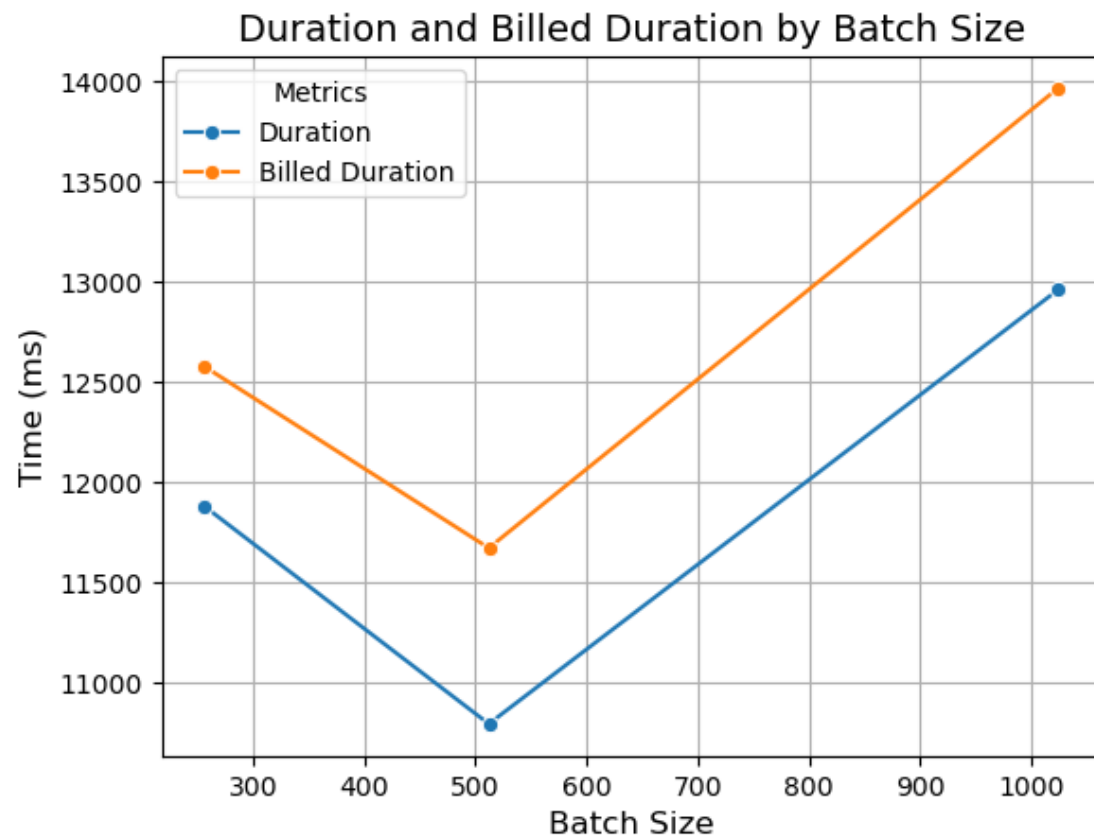
```
"total_cpu_time (seconds)": 5.806345376000015,  
"total_cpu_memory (MB)": 14320.256568,  
"execution_time (seconds/batch)": 2.9031726880000077,  
"num_batches": 2,  
"batch_size": 512,  
"device": "cpu",  
"throughput_bps": 6684057.66840576
```

```
"total_cpu_time (seconds)": 7.460655896000001,  
"total_cpu_memory (MB)": 14330.321692,  
"execution_time (seconds/batch)": 7.460655896000001,  
"num_batches": 1,  
"batch_size": 1024,  
"device": "cpu",
```

► CloudWatch Data:

	@timestamp		@message	inference_run
2645	2024-11-21 19:01:03.245	REPORT RequestId: f623ffe3-78ed-4d0e-a9de-ac00b9e9d71c\tDuration: 6434.80 ms\tBilled Duration: 7381 ms\tMemory Size: 10240 MB\tMax Memory Used: 645 MB\tInit Duration: 945.56 ms\t\n	received: {'S3_BUCKET': 'team-one-s3-cosmic', 'S3_OBJECT_NAME': 'scripts/Anomaly Detection', 'SCRIPT': '/tmp/scripts/Anomaly Detection/Inference/inference_batch2048.py', 'S3_OBJECT_TYPE': 'folder', 'WORLD_SIZE': '1', 'RANK': '0', 'data_path': '/tmp/scripts/Anomaly Detection/Inference/resized_in...	
2708	2024-11-21 18:52:57.067	REPORT RequestId: b9aa5c74-5ff2-4631-9e25-1d4e8bf749db\tDuration: 11879.83 ms\tBilled Duration: 12578 ms\tMemory Size: 10240 MB\tMax Memory Used: 1960 MB\tInit Duration: 698.09 ms\t\n	received: {'S3_BUCKET': 'team-one-s3-cosmic', 'S3_OBJECT_NAME': 'scripts/Anomaly Detection', 'SCRIPT': '/tmp/scripts/Anomaly Detection/Inference/inference_batch256.py', 'S3_OBJECT_TYPE': 'folder', 'WORLD_SIZE': '1', 'RANK': '0', 'data_path': '/tmp/scripts/Anomaly Detection/Inference/resized_inf...	
2767	2024-11-21 18:36:05.587	REPORT RequestId: 6196cb66-f681-4f5f-8641-71cc5f8f7b04\tDuration: 12960.20 ms\tBilled Duration: 13967 ms\tMemory Size: 10240 MB\tMax Memory Used: 3350 MB\tInit Duration: 1006.32 ms\t\n	received: {'S3_BUCKET': 'team-one-s3-cosmic', 'S3_OBJECT_NAME': 'scripts/Anomaly Detection', 'SCRIPT': '/tmp/scripts/Anomaly Detection/Inference/inference_batch1024.py', 'S3_OBJECT_TYPE': 'folder', 'WORLD_SIZE': '1', 'RANK': '0', 'data_path': '/tmp/scripts/Anomaly Detection/Inference/resized_in...	
2924	2024-11-21 17:32:32.226	REPORT RequestId: 394fd31e-ddb1-4746-aa70-7e3ac3c572eb\tDuration: 10792.45 ms\tBilled Duration: 11671 ms\tMemory Size: 10240 MB\tMax Memory Used: 2848 MB\tInit Duration: 877.64 ms\t\n	received: {'S3_BUCKET': 'team-one-s3-cosmic', 'S3_OBJECT_NAME': 'scripts/Anomaly Detection', 'SCRIPT': '/tmp/scripts/Anomaly Detection/Inference/inference.py', 'S3_OBJECT_TYPE': 'folder', 'WORLD_SIZE': '1', 'RANK': '0', 'data_path': '/tmp/scripts/Anomaly Detection/Inference/resized_inference.pt...	

Results(Continued)



Partitioning: Setup

- ▶ Input for Payload:
 - Change Bucket
 - Data Prefix: partitioned data files
 - File Limit: number of images (multiply by partition size to make total data)

Start execution

Name

7aea2eb7-85bf-4920-945d-2a9790037a7b

Must be 1-80 characters. Can use alphanumeric characters, dashes, or underscores.

Input - optional















Enter input values for this execution in JSON format

Format JSON

Export

Import

```
1 {  
2   "bucket": "cosmicai-data",  
3   "file_limit": "11",  
4   "batch_size": 512,  
5   "object_type": "folder",  
6   "S3_object_name": "Anomaly Detection",  
7   "script": "/tmp/Anomaly Detection/Inference/inference.py",  
8   "result_path": "result-partition-100MB/1GB/1",  
9   "data_bucket": "cosmicai-data",  
10  "data_prefix": "100MB"
```

<input type="checkbox"/>	Name	Type
<input type="checkbox"/>	 100MB/	Folder
<input type="checkbox"/>	 10MB/	Folder
<input type="checkbox"/>	 25MB/	Folder
<input type="checkbox"/>	 50MB/	Folder
<input type="checkbox"/>	 75MB/	Folder
<input type="checkbox"/>	 Anomaly Detection/	Folder
<input type="checkbox"/>	 demo/	Folder
<input type="checkbox"/>	 payload.json	json
<input type="checkbox"/>	 result-partition-100MB/	Folder
<input type="checkbox"/>	 result-partition-10MB/	Folder
<input type="checkbox"/>	 result-partition-25MB/	Folder
<input type="checkbox"/>	 result-partition-50MB/	Folder
<input type="checkbox"/>	 result-partition-75MB/	Folder
<input type="checkbox"/>	 team 1/	Folder

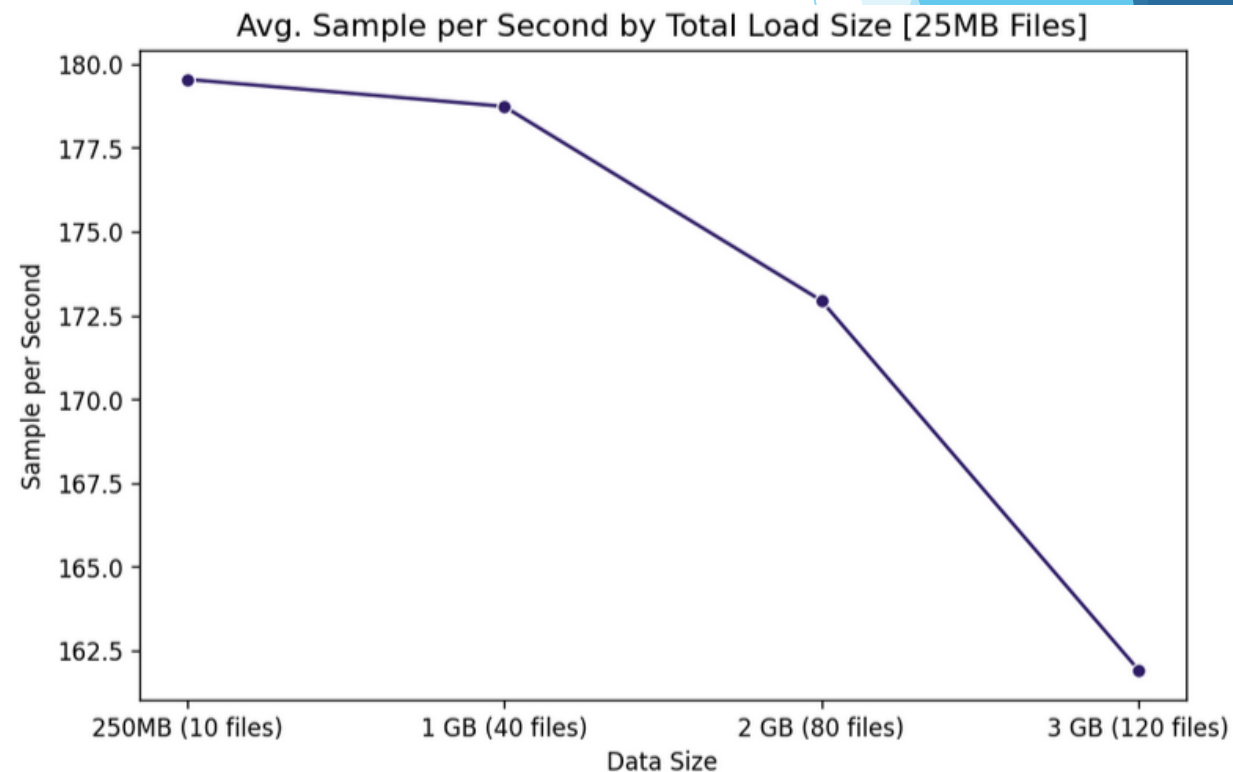
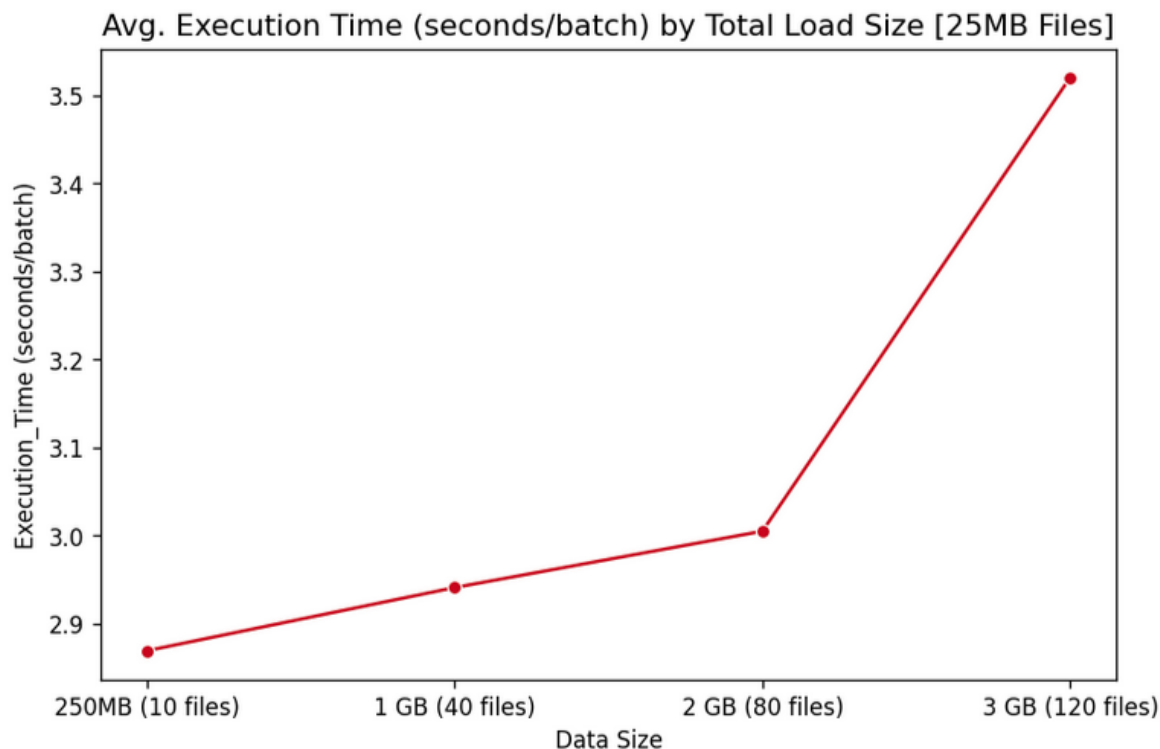
<https://github.com/UVA-MLSys/AI-for-Astronomy>

Parallel Performance

► Using the DataParallel-CosmicAI state machine, we:

- Edited the Payload json to run 4 size iterations for parallel execution (250MB, 1GB, 2GB, 3GB). ●
- We modified and used our `inference_data_parallel.py` file for execution.
- We pointed the Payload to data sources and result outputs located on our 'team-one-cosmic-data' S3 bucket.

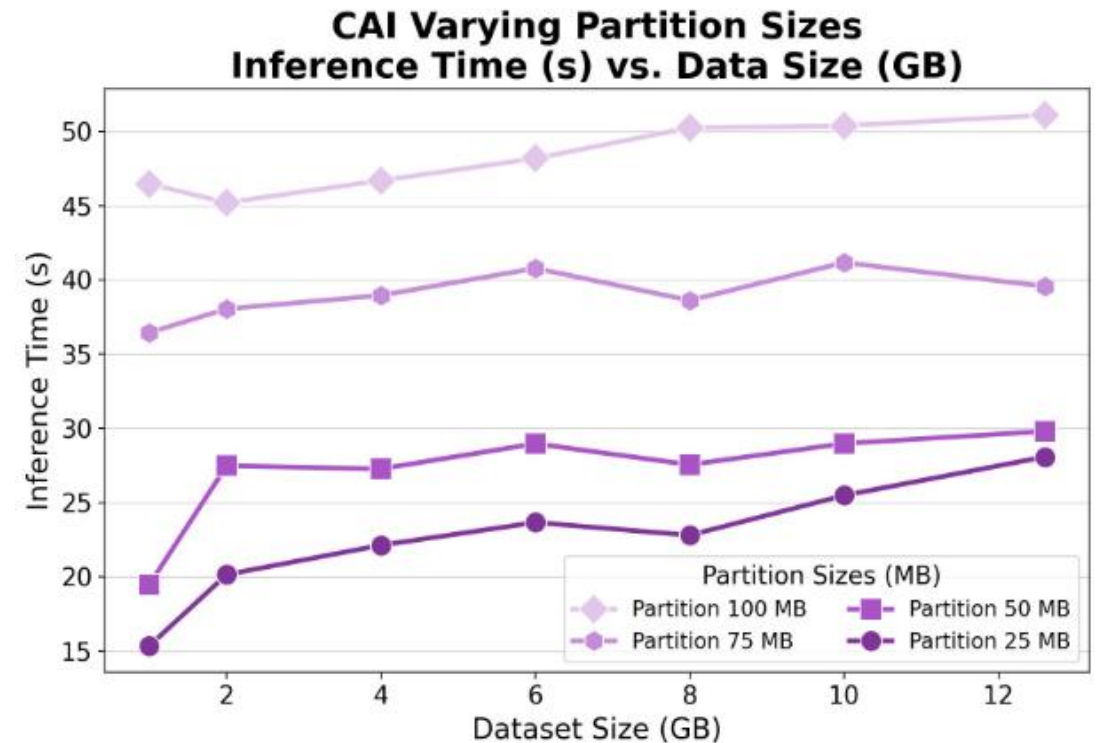
<input type="checkbox"/>	Team_1_25mb_FL120	✓ Succeeded
<input type="checkbox"/>	Team_1_25mb_FL80	✓ Succeeded
<input type="checkbox"/>	Team_1_25mb_FL40	✓ Succeeded
<input type="checkbox"/>	Team_1_25mb_FL10	✓ Succeeded



Partitioning: Inference Time

► Results for Inference Time

- For a given Partition size, the time is relatively steady with a slight increase in inference time (<5 seconds) as dataset size increases from 1GB to 12.6GB
- For smaller sizes (25MB & 50MB), the inference time increases by 5-10 seconds with dataset size increasing from 1GB to 12.6 GB
- Inference Time varies from ~15-25 seconds for 25MB to ~50 seconds for 100MB



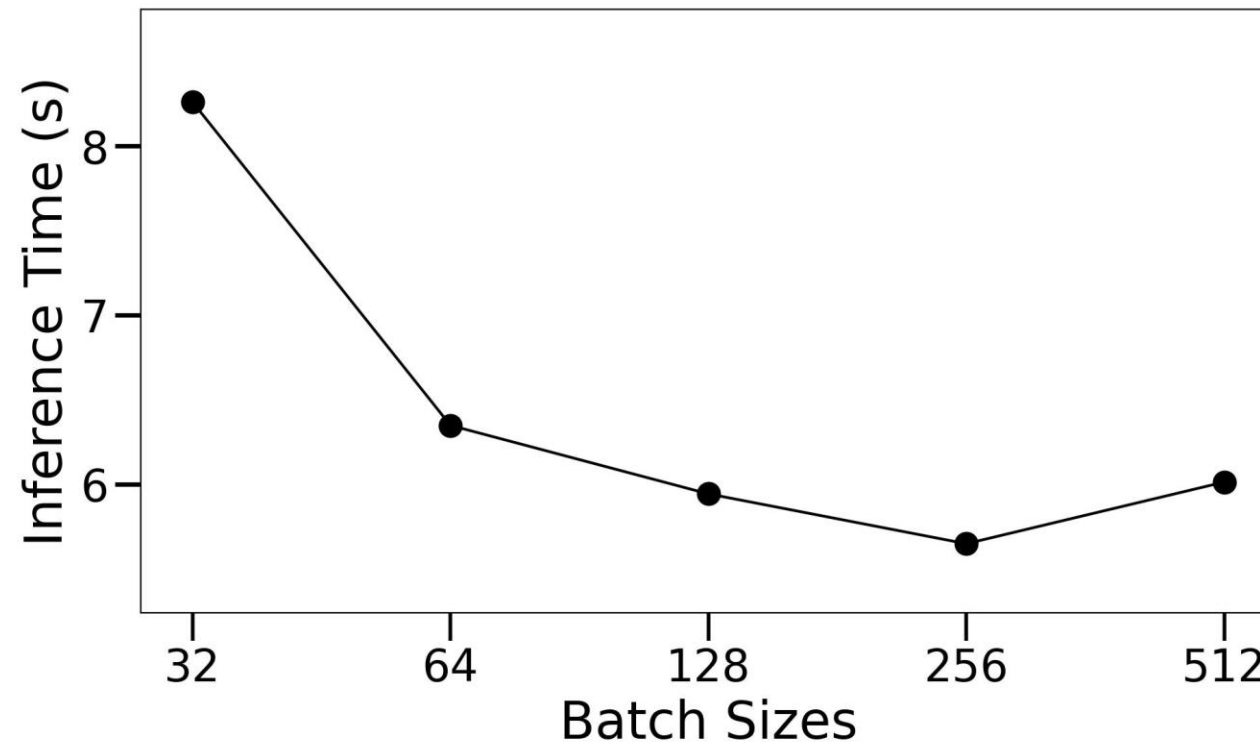
Partitioning: Cost Estimate

- ▶ AWS cost for Lambda function:
 - \$0.00001667 per GB-second of computation time
- ▶ Cost = requests x duration x memory (GB) x 0.00001667
 - Requests: number of times Lambda function is called
 - number of concurrent jobs = data / partition size

Partition	Requests	Duration (s)	Memory	Cost (\$)
25MB	517	6.55	2.8GB	0.16
50MB	259	11.8	4.0GB	0.20
75MB	173	17.6	5.9GB	0.30
100MB	130	25	7.0GB	0.38

Partitioning: Batch Size

- ▶ Batch size is varied with 1GB of data



<https://github.com/UVA-MLSys/AI-for-Astronomy>

Thank You!

Q & A

Appendix - Parallel Results

