

# Final Project Step 4

Group 6: Dae Hwang & Pratham Choksi

Creation of S3 bucket:

The screenshot shows the AWS S3 console interface. The top navigation bar includes 'Amazon S3', 'Buckets', and the selected bucket name 'team6-s3-cosmical-data'. Below the navigation is a toolbar with actions like 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. The main area is titled 'Objects (9)' and contains a table listing nine objects. The columns are 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. The objects listed are: '100MB/' (Folder), '10MB/' (Folder), '25MB/' (Folder), '50MB/' (Folder), '75MB/' (Folder), 'Anomaly Detection/' (Folder), 'demo/' (Folder), 'payload.json' (json file, last modified August 5, 2025, 15:03:04 UTC-04:00, size 14.6 KB, Standard storage class), and 'result-partition-100MB/' (Folder). A search bar at the top says 'Find objects by prefix'.

Evidence of repository cloning:

The screenshot shows the VS Code Explorer sidebar. The tree view displays a cloned repository structure under the 'AI-FOR-ASTRONOMY' folder. The visible files and folders include: 'assignment\_submissions', 'aws', 'code' (selected item, indicated by a green dot), 'data', 'papers', '.gitignore', 'batch\_size\_analysis.ipynb', 'LICENSE', 'README.md', and 'Step4.ipynb'. The 'code' folder is currently expanded.

## Documentation of Step Function configuration:

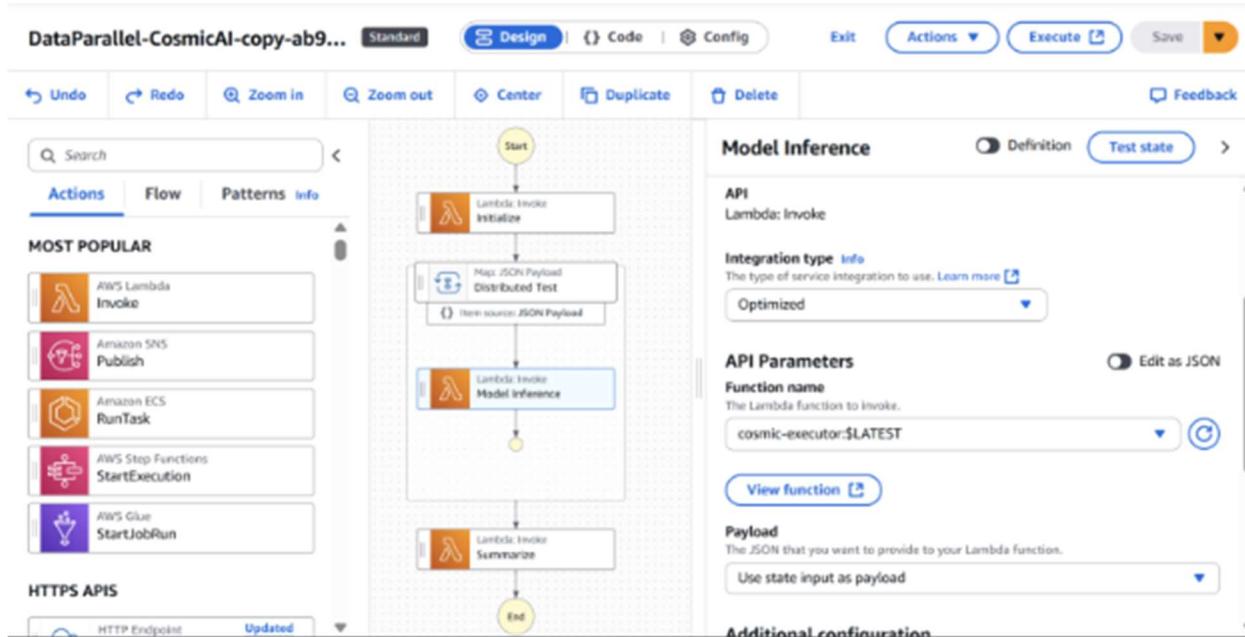
- Executions:

Executions (0/23)							View details	Stop execution	Redrive	<b>Start execution</b>
<input type="text"/> Filter executions by property or value		All	Last 15 months	Local timezone	23 matches	< 1 >				
	Name	Status	Start Time (local)	End Time (local)	Duration		Version			
<input type="checkbox"/>	<a href="#">8b198b5a-3a00-4e99-9079-86f7ee3bdbc9</a>	Succeeded	Jul 24, 2025, 00:10:03	Jul 24, 2025, 00:11:02	00:00:59.142		-			
<input type="checkbox"/>	<a href="#">42cd6ba1-cf26-489c-a881-130a1fa95ef3</a>	Succeeded	Jul 24, 2025, 00:07:03	Jul 24, 2025, 00:08:06	00:01:05.042		-			
<input type="checkbox"/>	<a href="#">9b5e98d5-eb8b-4cba-b48a-74b1da79bfc0</a>	Succeeded	Jul 24, 2025, 00:03:39	Jul 24, 2025, 00:04:31	00:00:51.503		-			
<input type="checkbox"/>	<a href="#">ceb16928-060d-4a4c-a0c9-7d1adda40ec0</a>	Succeeded	Jul 23, 2025, 23:51:03	Jul 23, 2025, 23:52:05	00:01:01.970		-			
<input type="checkbox"/>	<a href="#">3475a5e6-7fcc-4016-b7fb-c4a360b0b0a</a>	Succeeded	Jul 23, 2025, 23:49:23	Jul 23, 2025, 23:50:18	00:00:54.921		-			
<input type="checkbox"/>	<a href="#">092d8ee3-dca6-47bb-b98a-2107b1f6016b</a>	Succeeded	Jul 23, 2025, 23:48:03	Jul 23, 2025, 23:48:46	00:00:42.250		-			
<input type="checkbox"/>	<a href="#">7f9802fb-1ec9-4ef5-b65d-fb4b21e2ecd1</a>	Succeeded	Jul 23, 2025, 23:46:31	Jul 23, 2025, 23:47:09	00:00:37.731		-			
<input type="checkbox"/>	<a href="#">75mb-256-8gb</a>	Succeeded	Jul 23, 2025, 23:44:25	Jul 23, 2025, 23:45:10	00:00:44.604		-			
<input type="checkbox"/>	<a href="#">619620da-3f57-4242-a840-0d497ec67f34</a>	Succeeded	Jul 23, 2025, 23:41:16	Jul 23, 2025, 23:42:13	00:00:57.197		-			
<input type="checkbox"/>	<a href="#">01c18276-f4fe-48df-811f-fcd93000ba75</a>	Succeeded	Jul 22, 2025, 00:04:29	Jul 22, 2025, 00:05:23	00:00:53.784		-			
<input type="checkbox"/>	<a href="#">5f39c394-c5d4-4c7c-8ab1-df8832d739c8</a>	Succeeded	Jul 21, 2025, 23:45:23	Jul 21, 2025, 23:46:16	00:00:53.061		-			
	-									

- Logs

Log events		Actions	<b>Start tailing</b>	Create metric filter				
<input type="text"/> Filter events - press enter to search		Clear	1m	30m	1h	12h	Custom	UTC timezone
Display								
Timestamp		Message						
2025-07-24T03:46:32.619Z		END RequestId: 380c5542-70a9-4347-9744-15c124d32636						
2025-07-24T03:46:32.619Z		REPORT RequestId: 380c5542-70a9-4347-9744-15c124d32636 Duration: 708.71 ms Billed Duration: 709 ms Memory Size: 128 MB Max Memory: 128 MB						
2025-07-24T03:48:04.091Z		START RequestId: f89c1440-6ec2-4d0a-a335-e22783494c61 Version: \$LATEST						
2025-07-24T03:48:04.739Z		END RequestId: f89c1440-6ec2-4d0a-a335-e22783494c61						
2025-07-24T03:48:04.739Z		REPORT RequestId: f89c1440-6ec2-4d0a-a335-e22783494c61 Duration: 737.95 ms Billed Duration: 738 ms Memory Size: 128 MB Max Memory: 128 MB						
2025-07-24T03:49:23.498Z		START RequestId: 16f95440-9a17-4bad-8a7a-15d46705a9fc Version: \$LATEST						
2025-07-24T03:49:24.719Z		END RequestId: 16f95440-9a17-4bad-8a7a-15d46705a9fc						
2025-07-24T03:49:24.719Z		REPORT RequestId: 16f95440-9a17-4bad-8a7a-15d46705a9fc Duration: 1220.08 ms Billed Duration: 1221 ms Memory Size: 128 MB Max Memory: 128 MB						
2025-07-24T03:51:03.228Z		START RequestId: 1d5f2446-341b-4607-923e-a42017f0cded Version: \$LATEST						
2025-07-24T03:51:04.448Z		END RequestId: 1d5f2446-341b-4607-923e-a42017f0cded						
2025-07-24T03:51:04.448Z		REPORT RequestId: 1d5f2446-341b-4607-923e-a42017f0cded Duration: 1210.67 ms Billed Duration: 1211 ms Memory Size: 128 MB Max Memory: 128 MB						
		No newer events at this moment. Auto retry paused. <a href="#">Resume</a>						
		<a href="#">Back to top</a>						

- Step function:



Experiments:

## 1. Baseline (Benchmark for different batch sizes)

### A. 32 Batch Size, 100 MB Partition, 1 File Limit

Json Input

```
{
  "bucket": "cosmicai-data",
  "file_limit": "1",
  "batch_size": 32,
  "object_type": "folder",
  "S3_object_name": "Anomaly Detection",
  "script": "/tmp/Anomaly Detection/Inference/inference.py",
  "result_path": "results/batchtest-32/",
  "data_bucket": "cosmicai-data",
  "data_prefix": "100MB"
}
```

```
[{"total_cpu_time (seconds)": 29.713294564998964, "total_cpu_memory (MB)": 71518.209396, "total_process_memory (MB)": 2737.90234375, "library_overhead_memory (MB)": 2676.611415863037, "total_image_memory (MB)": 0.626220703125, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 0.18596233641305826, "num_batches": 160, "batch_size": 32, "device": "cpu", "throughput_bps": 28248300.71145055, "sample_persec": 172.07785521107792, "result_path": "results/batchtest-32/", "data_path": "100MB/1.pt"}]
```

## B. 64 Batch Size, 100 MB Partition, 1 File Limit

Json Input

```
{
  "bucket": "cosmicai-data",
  "file_limit": "1",
  "batch_size": 64,
  "object_type": "folder",
  "S3_object_name": "Anomaly Detection",
  "script": "/tmp/Anomaly Detection/Inference/inference.py",
  "result_path": "results/batchtest-64/",
  "data_bucket": "cosmicai-data",
  "data_prefix": "100MB"
}
```

```
[{"total_cpu_time (seconds)": 22.514245414000403, "total_cpu_memory (MB)": 71483.247204, "total_process_memory (MB)": 1681.69921875, "library_overhead_memory (MB)": 1619.782070159912, "total_image_memory (MB)": 1.25244140625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 0.28181335937727864, "num_batches": 80, "batch_size": 64, "device": "cpu", "throughput_bps": 37280844.39721232, "sample_persec": 227.10066031440252, "result_path": "results/batchtest-64/", "data_path": "100MB/1.pt"}]
```

## C. 128 Batch Size, 100 MB Partition, 1 File Limit

Json Input

```
{
  "bucket": "cosmicai-data",
  "file_limit": "1",
```

```

"batch_size": 128,
"object_type": "folder",
"S3_object_name": "Anomaly Detection",
"script": "/tmp/Anomaly Detection/Inference/inference.py",
"result_path": "results/batchtest-128/",
"data_bucket": "cosmicai-data",
"data_prefix": "100MB"
}

```

```
[{"total_cpu_time (seconds)": 25.759323667000533, "total_cpu_memory (MB)": 71466.756972, "total_process_memory (MB)": 2138.39453125, "library_overhead_memory (MB)": 2075.224941253662, "total_image_memory (MB)": 2.5048828125, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 0.6448647426904104, "num_batches": 40, "batch_size": 128, "device": "cpu", "throughput_bps": 32584321.34517822, "sample_persec": 198.4912362644872, "result_path": "results/batchtest-128/", "data_path": "100MB/1.pt"}]
```

#### D. 256 Batch Size, 100 MB Partition, 1 File Limit

Json Input

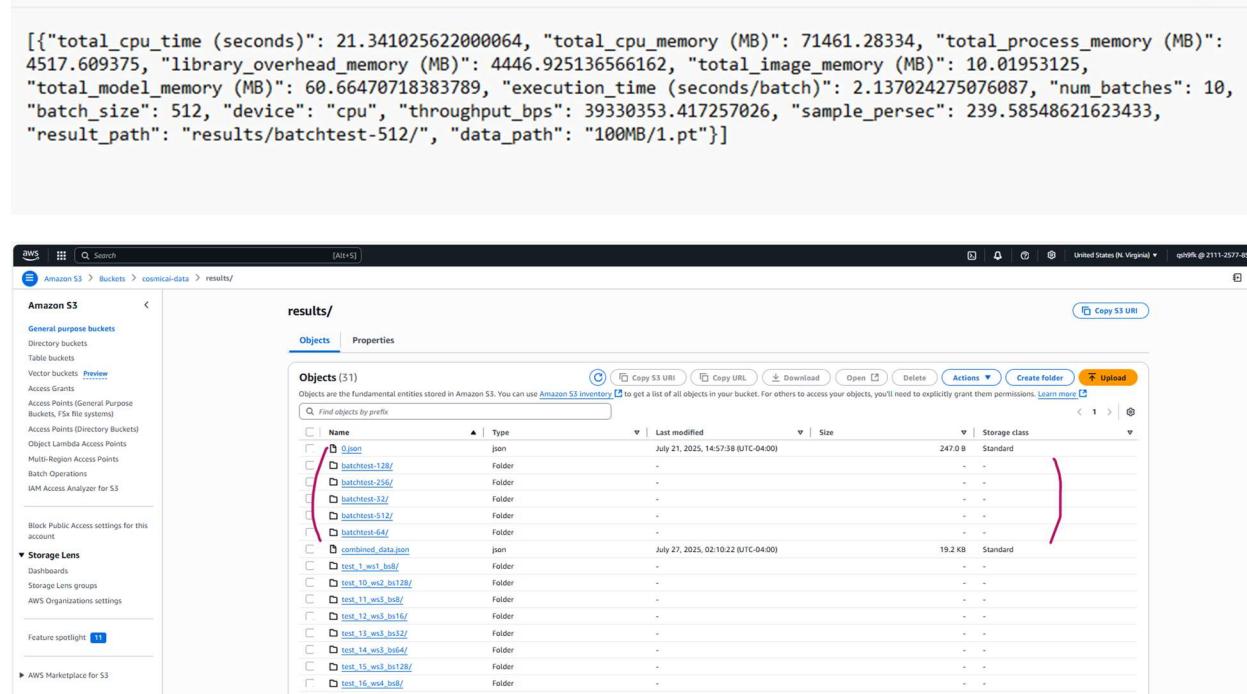
```
{
"bucket": "cosmicai-data",
"file_limit": "1",
"batch_size": 256,
"object_type": "folder",
"S3_object_name": "Anomaly Detection",
"script": "/tmp/Anomaly Detection/Inference/inference.py",
"result_path": "results/batchtest-256/",
"data_bucket": "cosmicai-data",
"data_prefix": "100MB"
}
```

```
[{"total_cpu_time (seconds)": 20.604544096999888, "total_cpu_memory (MB)": 71460.480308, "total_process_memory (MB)": 4361.40234375, "library_overhead_memory (MB)": 4295.727870941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.0316376469454276, "num_batches": 20, "batch_size": 256, "device": "cpu", "throughput_bps": 40736163.63694322, "sample_persec": 248.14914496188612, "result_path": "results/batchtest-256/", "data_path": "100MB/1.pt"}]
```

#### E. 512 Batch Size, 100 MB Partition, 1 File Limit

## Json Input

```
{  
  "bucket": "cosmicai-data",  
  "file_limit": "1",  
  "batch_size": 512,  
  "object_type": "folder",  
  "S3_object_name": "Anomaly Detection",  
  "script": "/tmp/Anomaly Detection/Inference/inference.py",  
  "result_path": "results/batchtest-512/",  
  "data_bucket": "cosmicai-data",  
  "data_prefix": "100MB"  
}
```



The screenshot shows the AWS S3 console with the path `Amazon S3 > Buckets > cosmicai-data > results/`. The `combined_data.json` file is highlighted with a red circle.

Name	Type	Last modified	Size	Storage class
combined_data.json	json	July 21, 2025, 14:57:38 (UTC-04:00)	247.0 B	Standard
test_1.ws1.bs1/	Folder	-	-	-
test_10.ws2.bs128/	Folder	-	-	-
test_11.ws3.bs8/	Folder	-	-	-
test_12.ws5.bs16/	Folder	-	-	-
test_13.ws1.bs32/	Folder	-	-	-
test_14.ws3.bs64/	Folder	-	-	-
test_15.ws5.bs128/	Folder	-	-	-
test_16.ws6.bs8/	Folder	-	-	-

## 2. Test Various Data Sizes

### A. 10 MB

```
{
```

```

"bucket": "cosmicai-data",
"file_limit": "105",
"batch_size": 256,
"object_type": "folder",
"S3_object_name": "Anomaly Detection",
"script": "/tmp/Anomaly Detection/Inference/inference.py",
"result_path": "result-partition-10MB/step4_1GB/",
"data_bucket": "cosmicai-data",
"data_prefix": "10MB"
}

```

```

[{"total_cpu_time (seconds)": 3.10212384999996, "total_cpu_memory (MB)": 7146.579476, "total_process_memory (MB)": 1182.38671875, "library_overhead_memory (MB)": 1116.712245941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.5540972712328747, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 27041396.171207063, "sample_persec": 164.72585386943874, "result_path": "result-partition-10MB/step4_1GB/", "data_path": "10MB/1.pt"}, {"total_cpu_time (seconds)": 3.135295963999995, "total_cpu_memory (MB)": 7146.579476, "total_process_memory (MB)": 1158.4140625, "library_overhead_memory (MB)": 1092.739589691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.5707157862700563, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 26755292.311536346, "sample_persec": 162.98301846696117, "result_path": "result-partition-10MB/step4_1GB/", "data_path": "10MB/10.pt"}, {"total_cpu_time (seconds)": 3.699574815999997, "total_cpu_memory (MB)": 7146.579476, "total_process_memory (MB)": 1317.4296875, "library_overhead_memory (MB)": 1251.755214691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.8534073442191765, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 22674432.650262713, "sample_persec": 138.12398056933912, "result_path": "result-partition-10MB/step4_1GB/", "data_path": "10MB/1007.pt"}, {"total_cpu_time (seconds)": 3.155868450999997, "total_cpu_memory (MB)": 7146.579476, "total_process_memory (MB)": 1179.8359375, "library_overhead_memory (MB)": 1114.161464691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.5810221594050868, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 26580879.81247101, "sample_persec": 161.9205641597893, "result_path": "result-partition-10MB/step4_1GB/", "data_path": "10MB/1089.pt"}, {"total_cpu_time (seconds)": 11.096149436000019, "total_cpu_memory (MB)": 7146.579476, "total_process_memory (MB)": 1174.3046875, "library_overhead_memory (MB)": 1108.630214691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 5.55893200707633}

```

result-partition-10MB/

[Copy S3 URI](#)

**Objects** | [Properties](#)

**Objects (2)**

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

<input type="checkbox"/> Name	Type	Last modified	Size	Storage class
<input type="checkbox"/> <a href="#">100MB/</a>	Folder	-	-	-
<input type="checkbox"/> <a href="#">step4_1GB/</a>	Folder	-	-	-

## B. 25 MB

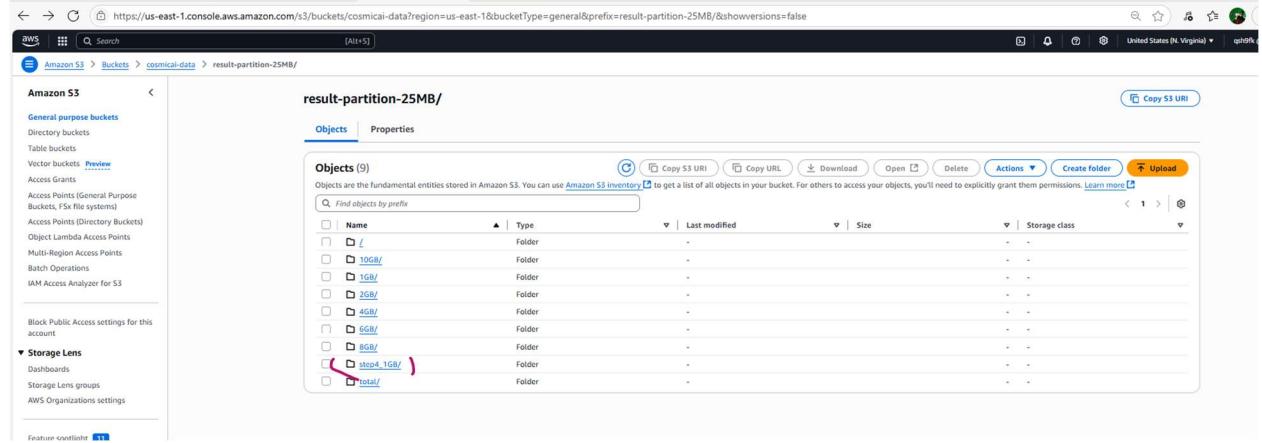
```

{
"bucket": "cosmicai-data",
"file_limit": "42",

```

```
"batch_size": 256,  
"object_type": "folder",  
"S3_object_name": "Anomaly Detection",  
"script": "/tmp/Anomaly Detection/Inference/inference.py",  
"result_path": "result-partition-25MB/step4_1GB/",  
"data_bucket": "cosmicai-data",  
"data_prefix": "25MB"  
}
```

1



C. 50 MB

```
{  
  "bucket": "cosmicai-data",  
  "file_limit": "21",  
  "batch_size": 256,  
  "object_type": "folder".
```

```

"S3_object_name": "Anomaly Detection",
"script": "/tmp/Anomaly Detection/Inference/inference.py",
"result_path": "result-partition-50MB/step4_1GB/",
"data_bucket": "cosmicai-data",
"data_prefix": "50MB"
}

```

```

[{"total_cpu_time (seconds)": 11.529824701999928, "total_cpu_memory (MB)": 35725.870728, "total_process_memory (MB)": 2629.4296875, "library_overhead_memory (MB)": 2563.755214691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.1547868246134514, "num_batches": 10, "batch_size": 256, "device": "cpu", "throughput_bps": 36391963.524581484, "sample_persec": 221.68593764974102, "result_path": "result-partition-50MB/step4_1GB/", "data_path": "50MB/1.pt"}, {"total_cpu_time (seconds)": 15.38578035900002, "total_cpu_memory (MB)": 35725.870728, "total_process_memory (MB)": 2546.25, "library_overhead_memory (MB)": 2480.575527191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.540985826253523, "num_batches": 10, "batch_size": 256, "device": "cpu", "throughput_bps": 27271477.312787466, "sample_persec": 166.12742027770142, "result_path": "result-partition-50MB/step4_1GB/", "data_path": "50MB/10.pt"}, {"total_cpu_time (seconds)": 10.46793000599968, "total_cpu_memory (MB)": 35725.870728, "total_process_memory (MB)": 2573.484375, "library_overhead_memory (MB)": 2507.809902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.048431174309856, "num_batches": 10, "batch_size": 256, "device": "cpu", "throughput_bps": 40083661.21663971, "sample_persec": 244.17434951656745, "result_path": "result-partition-50MB/step4_1GB/", "data_path": "50MB/108.pt"}, {"total_cpu_time (seconds)": 10.27132054000012, "total_cpu_memory (MB)": 35725.870736, "total_process_memory (MB)": 2616.78125, "library_overhead_memory (MB)": 2551.106777191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.028739459405322, "num_batches": 10, "batch_size": 256, "device": "cpu", "throughput_bps": 40850926.45740754, "sample_persec": 248.8482362171512, "result_path": "result-partition-50MB/step4_1GB/", "data_path": "50MB/109.pt"}, {"total_cpu_time (seconds)": 10.98844437100073, "total_cpu_memory (MB)": 35725.870744, "total_process_memory (MB)": 2615.18046875, "library_overhead_memory (MB)": 2579.80599591162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.028739459405322, "num_batches": 10, "batch_size": 256, "device": "cpu", "throughput_bps": 40850926.45740754, "sample_persec": 248.8482362171512, "result_path": "result-partition-50MB/step4_1GB/", "data_path": "50MB/110.pt"}]

```

result-partition-50MB/

[Copy S3 URI](#)

[Objects](#) | [Properties](#)

Objects (8)

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">10GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">1GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">2GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">4GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">6GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">8GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">step4_1GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">total/</a>	Folder	-	-	-

## D. 75 MB

{

```

"bucket": "cosmicai-data",
"file_limit": "11",
"batch_size": 256,
"object_type": "folder",
"S3_object_name": "Anomaly Detection",

```

```

"script": "/tmp/Anomaly Detection/Inference/inference.py",
"result_path": "result-partition-100MB/step4_1GB/",
"data_bucket": "cosmical-data",
"data_prefix": "100MB"
}

```

```

[{"total_cpu_time (seconds)": 16.7831817380002, "total_cpu_memory (MB)": 53586.178516, "total_process_memory (MB)": 3453.90625, "library_overhead_memory (MB)": 3388.231777191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.120629766543559, "num_batches": 15, "batch_size": 256, "device": "cpu", "throughput_bps": 37501199.10666007, "sample_persec": 228.4429770142548, "result_path": "result-partition-75MB/step4_1GB/", "data_path": "75MB/1.pt"}, {"total_cpu_time (seconds)": 17.12113890600002, "total_cpu_memory (MB)": 53586.178508, "total_process_memory (MB)": 3467.6953125, "library_overhead_memory (MB)": 3402.020839691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.1431955033740233, "num_batches": 15, "batch_size": 256, "device": "cpu", "throughput_bps": 36760956.3508321, "sample_persec": 223.93370096754447, "result_path": "result-partition-75MB/step4_1GB/", "data_path": "75MB/10.pt"}, {"total_cpu_time (seconds)": 19.045730841000076, "total_cpu_memory (MB)": 53586.178508, "total_process_memory (MB)": 3465.86328125, "library_overhead_memory (MB)": 3400.188808441162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.2717024244381898, "num_batches": 15, "batch_size": 256, "device": "cpu", "throughput_bps": 33046221.49994383, "sample_persec": 201.3049555308469, "result_path": "result-partition-75MB/step4_1GB/", "data_path": "75MB/108.pt"}, {"total_cpu_time (seconds)": 16.67243293799993, "total_cpu_memory (MB)": 53586.178524, "total_process_memory (MB)": 3468.03515625, "library_overhead_memory (MB)": 3402.360683441162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.1132349588231565, "num_batches": 15, "batch_size": 256, "device": "cpu", "throughput_bps": 37750305.68966879, "sample_persec": 229.96843914271922, "result_path": "result-partition-75MB/step4_1GB/", "data_path": "75MB/109.pt"}, {"total_cpu_time (seconds)": 15.641507431000058, "total_cpu_memory (MB)": 53586.178508, "total_process_memory (MB)": 3510.28515625, "library_overhead_memory (MB)": 3444.610683441162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.0000000000000002}

```

**result-partition-75MB/**

[Copy S3 URI](#)

[Objects](#) [Properties](#)

**Objects (8)**

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">10GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">1GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">2GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">4GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">6GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">8GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">step4_1GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">total/</a>	Folder	-	-	-

## E. 100 MB

```

{
"bucket": "cosmical-data",
"file_limit": "11",
"batch_size": 256,
"object_type": "folder",

```

```
"S3_object_name": "Anomaly Detection",
"script": "/tmp/Anomaly Detection/Inference/inference.py",
"result_path": "result-partition-100MB/step4_1GB/",
"data_bucket": "cosmicai-data",
"data_prefix": "100MB"
}
```

}

```
partition-100MB/step4_1GB/", "data_path": "100MB/108.pt"}, {"total_cpu_time (seconds)": 25.58494865899992, "total_cpu_memory (MB)": 71460.462876, "total_process_memory (MB)": 4210.02734375, "library_overhead_memory (MB)": 4144.352870941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.2809987984948132, "num_batches": 20, "batch_size": 256, "device": "cpu", "throughput_bps": 32806400.793958403, "sample_persec": 199.84405941738794, "result_path": "result-partition-100MB/step4_1GB/", "data_path": "100MB/100.pt"}, {"total_cpu_time (seconds)": 21.292850470000104, "total_cpu_memory (MB)": 71460.480308, "total_process_memory (MB)": 4447.7421875, "library_overhead_memory (MB)": 4382.067714691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.0661000822061464, "num_batches": 20, "batch_size": 256, "device": "cpu", "throughput_bps": 39419338.485590555, "sample_persec": 240.12754925432841, "result_path": "result-partition-100MB/step4_1GB/", "data_path": "100MB/101.pt"}, {"total_cpu_time (seconds)": 23.273928576999957, "total_cpu_memory (MB)": 71460.480316, "total_process_memory (MB)": 4446.5078125, "library_overhead_memory (MB)": 4380.833339691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.165289598225677, "num_batches": 20, "batch_size": 256, "device": "cpu", "throughput_bps": 36063962.18081861, "sample_persec": 219.6878787817898, "result_path": "result-partition-100MB/step4_1GB/", "data_path": "100MB/102.pt"}, {"total_cpu_time (seconds)": 21.94074792600015, "total_cpu_memory (MB)": 71460.480316, "total_process_memory (MB)": 4293.51171875, "library_overhead_memory (MB)": 4227.837245941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.098539305067755, "num_batches": 20, "batch_size": 256, "device": "cpu", "throughput_bps": 38255308.47129219, "sample_persec": 233.0367231438364, "result_path": "result-partition-100MB/step4_1GB/", "data_path": "100MB/103.pt"}, {"total_cpu_time (seconds)": 22.543164456000266, "total_cpu_memory (MB)": 71460.480308, "total_process_memory (MB)": 4359.51171875, "library_overhead_memory (MB)": 4293.837245941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)":
```

## result-partition-100MB/

[Objects](#) | [Properties](#)

[Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

### Objects (13)

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](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	<a href="#">0.1GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">100MB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">10GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">1GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">200MB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">2GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">400MB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">4GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">6GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">8GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">step4_1GB/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">tmp/</a>	Folder	-	-	-
<input type="checkbox"/>	<a href="#">total/</a>	Folder	-	-	-

### 3. 10 MB Partition Comparison

#### A. 2 GB Data Size

```
{
  "bucket": "cosmicai-data",
  "file_limit": "204",
  "batch_size": 256,
  "object_type": "folder",
  "S3_object_name": "Anomaly Detection",
  "script": "/tmp/Anomaly Detection/Inference/inference.py",
  "result_path": "result-partition-10MB/2GB",
  "data_bucket": "cosmicai-data",
  "data_prefix": "10MB"
}
```

[{"total\_cpu\_time (seconds)": 9.192125545000005, "total\_cpu\_memory (MB)": 7146.562068, "total\_process\_memory (MB)": 1157.02734375, "library\_overhead\_memory (MB)": 1091.352870941162, "total\_image\_memory (MB)": 5.009765625, "total\_model\_memory (MB)": 60.66470718383789, "execution\_time (seconds/batch)": 4.605057024500081, "num\_batches": 2, "batch\_size": 256, "device": "cpu", "throughput\_bps": 9125828.361387981, "sample\_persec": 55.5910597063108, "result\_path": "result-partition-10MB/2GB", "data\_path": "10MB/1.pt"}, {"total\_cpu\_time (seconds)": 9.098548529000015, "total\_cpu\_memory (MB)": 7146.579444, "total\_process\_memory (MB)": 1122.98828125, "library\_overhead\_memory (MB)": 1057.313808441162, "total\_image\_memory (MB)": 5.009765625, "total\_model\_memory (MB)": 60.66470718383789, "execution\_time (seconds/batch)": 4.558176953863021, "num\_batches": 2, "batch\_size": 256, "device": "cpu", "throughput\_bps": 9219685.94579923, "sample\_persec": 56.162804256730524, "result\_path": "result-partition-10MB/2GB", "data\_path": "10MB/10.pt"}, {"total\_cpu\_time (seconds)": 5.097674576999999, "total\_cpu\_memory (MB)": 7146.579476, "total\_process\_memory (MB)": 1168.93359375, "library\_overhead\_memory (MB)": 1103.259120941162, "total\_image\_memory (MB)": 5.009765625, "total\_model\_memory (MB)": 60.66470718383789, "execution\_time (seconds/batch)": 2.5337860894559685, "num\_batches": 2, "batch\_size": 256, "device": "cpu", "throughput\_bps": 16585835.787354575, "sample\_persec": 101.03457472803713, "result\_path": "result-partition-10MB/2GB", "data\_path": "10MB/100.pt"}, {"total\_cpu\_time (seconds)": 3.058965350000005, "total\_cpu\_memory (MB)": 7146.579476, "total\_process\_memory (MB)": 1169.203125, "library\_overhead\_memory (MB)": 1103.528652191162, "total\_image\_memory (MB)": 5.009765625, "total\_model\_memory (MB)": 60.66470718383789, "execution\_time (seconds/batch)": 1.5324757917888245, "num\_batches": 2, "batch\_size": 256, "device": "cpu", "throughput\_bps": 27422919.3213, "sample\_persec": 167.04994713326818, "result\_path": "result-partition-10MB/2GB", "data\_path": "10MB/1089.pt"}, {"total\_cpu\_time (seconds)": 2.962993206000002, "total\_cpu\_memory (MB)": 7146.579484, "total\_process\_memory (MB)": 1158.49609375, "library\_overhead\_memory (MB)": 1092.821620941162, "total\_image\_memory (MB)": 5.009765625, "total\_model\_memory (MB)": 60.66470718383789, "execution\_time (seconds/batch)": 1.4843507255107642, "num\_batches": 2, "batch\_size": 256, "device": "cpu", "throughput\_bps": 28312014.996010624, "sample\_persec": 172.46597828953838, "result\_path": "result-partition-10MB/2GB", "data\_path": "10MB/109.pt"}, {"total\_cpu\_time (seconds)": 3.3117704809999997, "total\_cpu\_memory (MB)": 7146.579476, "total\_process\_memory (MB)": 1332.70703125, "library\_overhead\_memory (MB)": 1267.032558441162, "total\_image\_memory (MB)": 5.009765625, "total\_model\_memory (MB)": 60.66470718383789, "execution\_time (seconds/batch)": 1.659125720422699, "num\_batches": 2, "batch\_size": 256, "device": "cpu", "throughput\_bps": 25329581.407063723, "sample\_persec": 154.298132352971, "result\_path": "result-partition-10MB/2GB", "data\_path": "10MB/1090.pt"}, {"total\_cpu\_time (seconds)": 2.9554528500000017, "total\_cpu\_memory (MB)": 7146.579476, "total\_process\_memory (MB)": 1134.53515625, "library\_overhead\_memory (MB)": 1068.800683441162, "total\_image\_memory (MB)": 5.009765625, "total\_model\_memory (MB)": 60.66470718383789, "execution\_time (seconds/batch)": 1.480861237575342475, "num\_batches": 2, "batch\_size": 256, "device": "cpu", "throughput\_bps": 28383386.322674695, "sample\_persec": 177.9007451429989, "result\_path": "result-partition-10MB/2GB", "data\_path": "10MB/1090.pt"}]

## B. 4 GB Data Size

```
{
  "bucket": "cosmicai-data",
  "file_limit": "816",
  "batch_size": 256,
  "object_type": "folder",
  "S3_object_name": "Anomaly Detection",
  "script": "/tmp/Anomaly Detection/Inference/inference.py",
  "result_path": "result-partition-10MB/6GB",
  "data_bucket": "cosmicai-data",
  "data_prefix": "10MB"
}
```

{

```

"bucket": "cosmicai-data",
"file_limit": "408",
"batch_size": 256,
"object_type": "folder",
"S3_object_name": "Anomaly Detection",
"script": "/tmp/Anomaly Detection/Inference/inference.py",
"result_path": "result-partition-10MB/4GB",
"data_bucket": "cosmicai-data",
"data_prefix": "10MB"
}

```

```

[{"total_cpu_time (seconds)": 9.19212545000005, "total_cpu_memory (MB)": 7146.562068, "total_process_memory (MB)": 1157.02734375, "library_overhead_memory (MB)": 1091.352870941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 4.605057024500981, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 9125828.361387981, "sample_persec": 55.5910597063108, "result_path": "result-partition-10MB/2GB", "data_path": "10MB/1.pt"}, {"total_cpu_time (seconds)": 9.098548529000015, "total_cpu_memory (MB)": 7146.579444, "total_process_memory (MB)": 1122.98828125, "library_overhead_memory (MB)": 1057.3138808441162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 4.558176953863021, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 9219685.945799924, "sample_persec": 56.162804250730524, "result_path": "result-partition-10MB/2GB", "data_path": "10MB/10.pt"}, {"total_cpu_time (seconds)": 5.057674576999999, "total_cpu_memory (MB)": 7146.579476, "total_process_memory (MB)": 1168.93359375, "library_overhead_memory (MB)": 1183.259120941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.5337860894559685, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 16585835.787354575, "sample_persec": 101.03457472803713, "result_path": "result-partition-10MB/2GB", "data_path": "10MB/1007.pt"}, {"total_cpu_time (seconds)": 3.058965350000005, "total_cpu_memory (MB)": 7146.579476, "total_process_memory (MB)": 1169.203125, "library_overhead_memory (MB)": 1183.528652191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.5324757917808245, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 27422919.3213973, "sample_persec": 167.04994713326818, "result_path": "result-partition-10MB/2GB", "data_path": "10MB/1089.pt"}, {"total_cpu_time (seconds)": 2.962903206000002, "total_cpu_memory (MB)": 7146.579484, "total_process_memory (MB)": 1158.49609375, "library_overhead_memory (MB)": 1092.821620941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.4843507255107642, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 28312014.996010624, "sample_persec": 172.46597828953838, "result_path": "result-partition-10MB/2GB", "data_path": "10MB/109.pt"}, {"total_cpu_time (seconds)": 3.311770488999997, "total_cpu_memory (MB)": 7146.579476, "total_process_memory (MB)": 1332.70703125, "library_overhead_memory (MB)": 1267.032558441162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.659125720422699, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 25329581.407063723, "sample_persec": 154.298132352971, "result_path": "result-partition-10MB/2GB", "data_path": "10MB/1090.pt"}, {"total_cpu_time (seconds)": 2.9554528500000017, "total_cpu_memory (MB)": 7146.579476, "total_process_memory (MB)": 1134.53515625, "library_overhead_memory (MB)": 1068.860683441162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.4806182975342475, "num_batches": 2, "batch_size": 256, "device": "cpu", "throughput_bps": 28383386.322674695, "sample_persec": 172.9807451429989, "result_path": "result-partition-10MB/6GB", "data_path": "10MB/6.pt"}]

```

```

{
"bucket": "cosmicai-data",
"file_limit": "612",
"batch_size": 256,
"object_type": "folder",
"S3_object_name": "Anomaly Detection",
"script": "/tmp/Anomaly Detection/Inference/inference.py",
"result_path": "result-partition-10MB/6GB",
"data_bucket": "cosmicai-data",
"data_prefix": "10MB"
}

```

## F. 8 GB data size

```

{
"bucket": "cosmicai-data",
"file_limit": "816",
"batch_size": 256,
"object_type": "folder",
"S3_object_name": "Anomaly Detection",
"script": "/tmp/Anomaly Detection/Inference/inference.py",
}
```

```

    "result_path": "result-partition-10MB/8GB",
    "data_bucket": "cosmicaidata",
    "data_prefix": "10MB"
}

```

#### G. 12 GB data size

```

{
    "bucket": "cosmicaidata",
    "file_limit": "1230",
    "batch_size": 256,
    "object_type": "folder",
    "S3_object_name": "Anomaly Detection",
    "script": "/tmp/Anomaly Detection/Inference/inference.py",
    "result_path": "results/10MB/12GB/",
    "data_bucket": "cosmicaidata",
    "data_prefix": "10MB"
}

```

The screenshot shows the AWS S3 console interface. At the top, there's a header with 'result-partition-10MB/' and a 'Copy S3 URI' button. Below the header, there are tabs for 'Objects' and 'Properties'. The 'Objects' tab is selected, showing a table with the following data:

Name	Type	Last modified	Size	Storage class
100MB/	Folder	-	-	-
2GB/	Folder	-	-	-
4GB/	Folder	-	-	-
6GB/	Folder	-	-	-
8GB/	Folder	-	-	-
step4.1GB/	Folder	-	-	-

## 4. 10 MB Partition Comparison

### A. 2 GB

```

{
    "bucket": "cosmicaidata",
    "file_limit": "82",
}

```

```

"batch_size": 256,
"object_type": "folder",
"S3_object_name": "Anomaly Detection",
"script": "/tmp/Anomaly Detection/Inference/inference.py",
"result_path": "result-partition-25MB/2GB",
"data_bucket": "cosmicai-data",
"data_prefix": "25MB"
}

[{"total_cpu_time (seconds)": 5.29553965899999, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1759.26171875, "library_overhead_memory (MB)": 1693.587245941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.0607653777026584, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 39617582.62794655, "sample_persec": 241.33517682715976, "result_path": "result-partition-25MB/2GB", "data_path": "25MB/1.pt"}, {"total_cpu_time (seconds)": 8.429967472999971, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.6886319820719817, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 24886985.705692142, "sample_persec": 151.6020084411071, "result_path": "result-partition-25MB/2GB", "data_path": "25MB/10.pt"}, {"total_cpu_time (seconds)": 7.914526914999993, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.6886319820719817, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 24886985.705692142, "sample_persec": 161.47522318458135, "result_path": "result-partition-25MB/2GB/1", "data_path": "25MB/1.pt"}, {"total_cpu_time (seconds)": 6.971471624000015, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1759.26171875, "library_overhead_memory (MB)": 17864.672524, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.7929526380970326, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 30093571.5319781, "sample_persec": 183.31854003397964, "result_path": "result-partition-25MB/2GB/1", "data_path": "25MB/10.pt"}, {"total_cpu_time (seconds)": 6.44168906599984, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.5817014713489765, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 32556618.165834915, "sample_persec": 198.31882410961813, "result_path": "result-partition-25MB/2GB/1", "data_path": "25MB/100.pt"}, {"total_cpu_time (seconds)": 6.7690159230800085, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.7118436248638528, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 30993645.514578566, "sample_persec": 188.88144684806632, "result_path": "result-partition-25MB/2GB/1", "data_path": "25MB/1000.pt"}, {"total_cpu_time (seconds)": 7.53930611699999, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.0094418872488223, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 3.170765086447571, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 265807772.637980875, "sample_persec": 169.51162085305228, "result_path": "result-partition-25MB/2GB/1", "data_path": "25MB/11.pt"}, {"total_cpu_time (seconds)": 6.926193985000017, "total_cpu_memory (MB)": 17864.669452, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.0094418872488223, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.7748132396870178, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 30290298.027221583, "sample_persec": 184.51692268044334, "result_path": "result-partition-25MB/2GB/1", "data_path": "25MB/110.pt"}, {"total_cpu_time (seconds)": 6.487147201999998, "total_cpu_memory (MB)": 17864.672532, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.00919692820307, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.59891692820307, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 32340329.80403457, "sample_persec": 197.0049330167977, "result_path": "result-partition-25MB/2GB/1", "data_path": "25MB/1100.pt"}]

```

## B. 4 GB

```

{
"bucket": "cosmicai-data",
"file_limit": "164",
"batch_size": 256,
"object_type": "folder",
"S3_object_name": "Anomaly Detection",
"script": "/tmp/Anomaly Detection/Inference/inference.py",
"result_path": "result-partition-25MB/4GB",
"data_bucket": "cosmicai-data",
"data_prefix": "25MB"
}

[{"total_cpu_time (seconds)": 5.264722919999992, "total_cpu_memory (MB)": 17865.562932, "total_process_memory (MB)": 1759.8515625, "library_overhead_memory (MB)": 1694.177089691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.0545923846009373, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 39849481.76531963, "sample_persec": 242.7478177711966, "result_path": "result-partition-25MB/4GB", "data_path": "25MB/1.pt"}, {"total_cpu_time (seconds)": 5.42377566800036, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1767.3046825, "library_overhead_memory (MB)": 1701.630214691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.0864527159687083, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 38680891.8439948, "sample_persec": 235.62921444928602, "result_path": "result-partition-25MB/4GB", "data_path": "25MB/10.pt"}, {"total_cpu_time (seconds)": 5.19559088499998, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.081488680062597, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 40379715.1553437, "sample_persec": 245.97779699892694, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/1.pt"}, {"total_cpu_time (seconds)": 5.480414256000002, "total_cpu_memory (MB)": 17864.672492, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.195596321652583, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 38281135.366786025, "sample_persec": 233.19485072359908, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/10.pt"}, {"total_cpu_time (seconds)": 5.66419276999999, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.269222764970262, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 37839078.44459751, "sample_persec": 225.6279145016905, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/100.pt"}, {"total_cpu_time (seconds)": 5.7735715795931233, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.7735715795931233, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 30303858.25208446, "sample_persec": 184.5995263894034, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/19.pt"}, {"total_cpu_time (seconds)": 5.22553827593907876, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.7735715795931233, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 137.38930067804506, "sample_persec": 137.38930067804506, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/19.pt"}, {"total_cpu_time (seconds)": 5.6595073458278495, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.7735715795931233, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 31603567.52984903, "sample_persec": 151.56185873446047, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/191.pt"}, {"total_cpu_time (seconds)": 5.1252832424600993, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 2.7735715795931233, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 3.1252832424600993, "sample_persec": 151.56185873446047, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/192.pt"}, {"total_cpu_time (seconds)": 5.6923946850000023, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 3.726366262582894, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 30303858.25208446, "sample_persec": 184.5995263894034, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/193.pt"}, {"total_cpu_time (seconds)": 5.7530370861000023, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 3.016862191574344, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 27860046.18796946, "sample_persec": 184.41351009623196, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/193.pt"}, {"total_cpu_time (seconds)": 5.7530370861000023, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 3.016862191574344, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 27860046.18796946, "sample_persec": 184.41351009623196, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/194.pt"}, {"total_cpu_time (seconds)": 5.7569001481000015, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 3.016862191574344, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 27860046.18796946, "sample_persec": 184.41351009623196, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/194.pt"}, {"total_cpu_time (seconds)": 5.7569001481000015, "total_cpu_memory (MB)": 17864.672524, "total_process_memory (MB)": 1764.234375, "library_overhead_memory (MB)": 1698.559902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 3.016862191574344, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 27860046.18796946, "sample_persec": 184.41351009623196, "result_path": "result-partition-25MB/4GB/1", "data_path": "25MB/194.pt"}]

```

## C. 6 GB

```
{

```

D. 8 GB

```

{
  "bucket": "cosmicai-data",
  "file_limit": "328",
  "batch_size": 256,
  "object_type": "folder",
  "S3_object_name": "Anomaly Detection",
  "script": "/tmp/Anomaly Detection/Inference/inference.py",
  "result_path": "result-partition-25MB/8GB",
  "data_bucket": "cosmicai-data",
  "data_prefix": "25MB"
}

total_cpu_time (seconds)": 6.146913607000029, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1758.37109375, "library_overhead_memory (MB)": 1692.696620941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 13086638021915, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 34130377.196303256, "sample_persec": 207.90921781373817, "result_partition": "result-partition-25MB/8GB", "data_path": "25MB/1.pt"}, {"total_cpu_time (seconds)": 5.56559601700004, "total_cpu_memory (MB)": 5.562956, "total_process_memory (MB)": 1689.47265625, "library_overhead_memory (MB)": 1623.798183441162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.1148611739843515, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 37695204.43027255, "sample_persec": 229.62500262260752, "result_path": "result-partition-25MB/8GB", "data_path": "25MB/10.pt"}, {"total_cpu_time (seconds)": 7.33882392999997, "total_cpu_memory (MB)": 17864.672524, "execution_time (seconds/batch)": 2.940123514992167, "num_batches": 512, "device": "cpu", "throughput_bps": 28587207.160316844, "sample_persec": 174.14234381284626, "result_path": "result-partition-25MB/8GB/1", "data_path": "25MB/1.pt"}, {"total_cpu_time (seconds)": 5.64223341999989, "total_cpu_memory (MB)": 17864.672524, "execution_time (seconds/batch)": 2.260425282503908, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 37183233.01838874, "sample_persec": 50604908862536, "result_path": "result-partition-25MB/8GB/1", "data_path": "25MB/10.pt"}, {"total_cpu_time (seconds)": 5.550951276000012, "total_cpu_memory (MB)": 17864.672532, "execution_time (seconds/batch)": 2.223855284281695, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 37794689.51692516, "sample_persec": 230.23080846080143, "result_path": "result-partition-25MB/8GB/1", "data_path": "25MB/108.pt"}, {"total_cpu_time (seconds)": 9.074381654000005, "total_cpu_memory (MB)": 17864.672532, "execution_time (seconds/batch)": 3.6354330257026626, "num_batches": 512, "device": "cpu", "throughput_bps": 23119644.731662933, "sample_persec": 140.83604246870695, "result_path": "result-partition-25MB/8GB/1", "data_path": "25MB/19.pt"}, {"total_cpu_time (seconds)": 22.492726758000053, "total_cpu_memory (MB)": 17864.672524, "execution_time (seconds/batch)": 9.011170657352134, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 9327203.099228779, "sample_persec": 18256832352309, "result_path": "result-partition-25MB/8GB/1", "data_path": "25MB/108.pt"}, {"total_cpu_time (seconds)": 22.492726758000053, "total_cpu_memory (MB)": 17864.672524, "execution_time (seconds/batch)": 7.42273923202945, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 9327203.099228779, "sample_persec": 18256832352309, "result_path": "result-partition-25MB/8GB/1", "data_path": "25MB/108.pt"}]
```

## E. 10 GB

```
{  
    "bucket": "cosmicai-data",  
    "file_limit": "410",  
    "batch_size": 256,  
    "object_type": "folder",  
    "S3_object_name": "Anomaly Detection",  
    "script": "/tmp/Anomaly Detection/Inference/inference.py",  
    "result_path": "result-partition-25MB/10GB",  
    "data_bucket": "cosmicai-data",  
    "data_prefix": "25MB"  
}  
  
[{"total_cpu_time (seconds)": 21.48255563999999, "total_cpu_memory (MB)": 17865.479884, "total_process_memory (MB)": 1749.9689375, "library_overhead_memory (MB)": 1684.286464691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 4.303234932582158, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 9765899.528702447, "sample_persec": 59.49012870798274, "result_path": "result-partition-25MB/10GB", "data_path": "25MB/1.pt"}, {"total_cpu_time (seconds)": 17865.562956, "total_process_memory (MB)": 1755.69921875, "library_overhead_memory (MB)": 1690.024745941162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.0720254930328605, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 39201455.817162946, "sample_persec": 238.80029128388733, "result_path": "result-partition-25MB/10GB", "data_path": "25MB/10.pt"}, {"total_cpu_time (seconds)": 5.52192527300002, "total_cpu_memory (MB)": 17864.672524, "execution_time (seconds/batch)": 2.2122267134397577, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 37993357.3215523, "sample_persec": 231.4410168223215, "result_path": "result-partition-25MB/10GB/1", "data_path": "25MB/1.pt"}, {"total_cpu_time (seconds)": 5.36077667900007, "total_cpu_memory (MB)": 17864.672524, "execution_time (seconds/batch)": 2.1476664003505506, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 39135463.49017754, "sample_persec": 238.39829124133496, "result_path": "result-partition-25MB/10GB/1", "data_path": "25MB/10.pt"}, {"total_cpu_time (seconds)": 5.367419394000011, "total_cpu_memory (MB)": 17864.672524, "execution_time (seconds/batch)": 2.1503276445446056, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 39087029.464208057, "sample_persec": 238.1032496600912, "result_path": "result-partition-25MB/10GB/1", "data_path": "25MB/108.pt"}, {"total_cpu_time (seconds)": 5.397473535000002, "total_cpu_memory (MB)": 17864.672524, "execution_time (seconds/batch)": 2.162368114178405, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 38869385.58189705, "sample_persec": 236.77744628348594, "result_path": "result-partition-25MB/10GB/1", "data_path": "25MB/19.pt"}, {"total_cpu_time (seconds)": 6.717689496000001, "total_cpu_memory (MB)": 17864.672524, "execution_time (seconds/batch)": 2.6912809248450706, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 31230452.095906157, "sample_persec": 190.24398206570513, "result_path": "result-partition-25MB/10GB/1", "data_path": "25MB/190.pt"}, {"total_cpu_time (seconds)": 5.568077457999988, "total_cpu_memory (MB)": 17864.672492, "execution_time (seconds/batch)": 2.230716477696396, "num_batches": 3, "batch_size": 512, "device": "cpu", "throughput_bps": 37678441.36194135, "sample_persec": 229.52266911513982, "result_path": "result-partition-25MB/10GB/1", "data_path": "25MB/191.pt"}, {"total_cpu_time (seconds)": 6.945861355999955, "total_cpu_memory (MB)": 17864.672524, "execution_time (seconds/batch)": 2.7826924994303583, "num_batches": 3}
```

## F. 12 GB

```
{  
    "bucket": "cosmicai-data",  
    "file_limit": "492",  
    "batch_size": 256,  
    "object_type": "folder",  
    "S3_object_name": "Anomaly Detection",  
    "script": "/tmp/Anomaly Detection/Inference/inference.py",  
    "result_path": "result-partition-25MB/12GB",  
    "data_bucket": "cosmicai-data",  
    "data_prefix": "25MB"  
}
```

```
[{"total_cpu_time (seconds)": 5.24723246299981, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1738.578125, "library_overhead_memory (MB)": 1672.903652191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.0510888188794953, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 39982310.95712764, "sample_persec": 243.5569624581362, "result_path": "result-partition-25MB/12GB", "data_path": "25MB/1.pt"}, {"total_cpu_time (seconds)": 5.309632620999971, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1717.5078125, "library_overhead_memory (MB)": 1651.833339691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.0635883810453777, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 39512428.63211291, "sample_persec": 240.6946188603696, "result_path": "result-partition-25MB/12GB", "data_path": "25MB/10.pt"}, {"total_cpu_time (seconds)": 5.444923967999975, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1693.609375, "library_overhead_memory (MB)": 1627.934902191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.090688951549246, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 38530653.73051706, "sample_persec": 234.71402126289632, "result_path": "result-partition-25MB/12GB", "data_path": "25MB/108.pt"}, {"total_cpu_time (seconds)": 6.277321760000005, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1764.5078125, "library_overhead_memory (MB)": 1698.833339691162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.2574290849452279, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 33421336.044434313, "sample_persec": 203.59001001726554, "result_path": "result-partition-25MB/12GB", "data_path": "25MB/19.pt"}, {"total_cpu_time (seconds)": 5.570221528000038, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1732.95703125, "library_overhead_memory (MB)": 1667.282558441162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.1157877239186305, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 37663938.3093488, "sample_persec": 229.43432205987327, "result_path": "result-partition-25MB/12GB", "data_path": "25MB/190.pt"}, {"total_cpu_time (seconds)": 5.771450335999998, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1762.76953125, "library_overhead_memory (MB)": 1697.095058441162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.156096467931142, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 36350738.16565197, "sample_persec": 221.43480851396177, "result_path": "result-partition-25MB/12GB", "data_path": "25MB/191.pt"}, {"total_cpu_time (seconds)": 5.800999019000016, "total_cpu_memory (MB)": 17865.562948, "total_process_memory (MB)": 1686.21875, "library_overhead_memory (MB)": 1620.544277191162, "total_image_memory (MB)": 5.009765625, "total_model_memory (MB)": 60.66470718383789, "execution_time (seconds/batch)": 1.16201545294523, "num_batches": 5, "batch_size": 256, "device": "cpu", "throughput_bps": 36165577.56911412, "sample_persec": 220.30688090347297, }
```

## 5. Performance Measurement of AWS Lamda

Partition (MB)	Lambda/File Limit	Avg Runtime (s)	Memory (MB)	Data Size (GB)	Cost (\$)	Throughput (GB/s)	Batch Size	Total Execution Time (s)	Performance Efficiency (GB/s)/\$
<b>Different Batch Sizes @ 100 MB Partition &amp; 1 File Limit</b>									
100	1	225.48	2738.23	0.10	0.0101	0.0004	32	163.2	0.04
100	1	22.51	1681.7	0.10	0.0006	0.0043	64	65.9	7.04
100	1	25.76	2138.4	0.10	0.0009	0.0038	128	45.7	4.23
100	1	20.60	4361.4	0.10	0.0015	0.0047	256	29.3	3.24
100	1	21.34	4517.6	0.10	0.0016	0.0046	512	37.5	2.92
<b>Different Partitions @ 1 MB Data Size</b>									
10	105	3.33	1186.8	1.03	0.0068	0.3080	256	28.4	45.61
25	42	6.35	1720.11	1.03	0.0075	0.1615	256	25.8	21.62
50	21	10.92	2570	1.03	0.0096	0.0939	256	26.7	9.79
75	14	17.05	3473	1.03	0.0135	0.0601	256	33.3	4.46
100	11	24.07	4314	1.07	0.0186	0.0446	256	39.7	2.40
<b>Different Data Sizes @ 10 MB Partition</b>									
10	105	3.33	1186.8	1.03	0.0068	0.3080	256	28.4	45.61
10	204	3.09	1190.26	1.99	0.0122	0.6441	256	32.1	52.68
10	408	2.75	1180.918	3.98	0.0216	1.4482	256	41.8	67.11
10	612	2.80	1186.84	5.98	0.0331	2.1350	256	74.4	64.50
10	816	3.60	1187.79	7.97	0.0568	2.2127	256	95.2	38.94
<b>Different Data Size @ 25 MB Partition</b>									
25	42	6.35	1720.11	1.03	0.0075	0.1615	256	25.8	21.62
25	82	5.99	1740.11	2.00	0.0139	0.3340	256	23.8	23.98
25	164	6.60	1739.01	4.00	0.0306	0.6067	256	33.5	19.80
25	246	6.38	1724.74	6.01	0.0440	0.9419	256	34.7	21.39
25	328	6.30	1737.70	8.01	0.0584	1.2718	256	48.2	21.77
25	410	6.46	1730.85	10.01	0.0746	1.5496	256	53.6	20.76
25	492	6.39	1729.52	12.01	0.0885	1.8792	256	64.4	21.22

### **A. Batch Size Scaling**

Several batch sizes were tested to see how they affect the speed and cost of the AWS Lambda inference process. As the batch size increased from 32 to 512, the total time to complete the task became shorter, and the system was able to process more data per second. However, this also led to higher memory use, which increased the cost. To better understand the trade-off between speed and cost, a new measurement was created called performance efficiency. This value looks at how much data can be processed for each dollar spent.

After comparing the results, batch size 256 was chosen for the rest of the testing because it provided the best balance between speed and cost. While a smaller batch size like 64 gave the highest efficiency in one test, batch size 256 performed well across different tests and data sizes. It gave fast results without being too expensive to run. That's why it was used as the standard for comparing partition sizes and larger datasets in the rest of the project.

### **B. Partition Size Scaling**

The impact of partition size on performance was tested by keeping the data size at 1 GB and using the same batch size of 256. The results showed that smaller partition sizes worked better overall, mostly because they allowed more Lambda functions to run at the same time. The 10 MB partition performed the best, giving the most data processed for each dollar spent, thanks to 105 functions running in parallel. As the partition size got larger, up to 100 MB, the system didn't perform as efficiently, even though it could still process more data per second. This happened because larger partitions reduced the number of requests but increased how long each one took and how much memory was needed. Partition sizes around 25 MB seemed to offer a good middle ground, with a nice balance between speed and cost for medium-sized jobs.

### **C. Scaling of Larger Data Sizes**

To see how well the system handles bigger jobs, tests were run using larger datasets ranging from 1 GB to 12 GB. Two partition sizes were used for this: 10 MB and 25 MB. The 10 MB partition gave the best overall performance in earlier tests, so it was used again here. However, AWS Step Functions have a limit of 1,000 parallel Lambda executions per workflow, which means we couldn't use 10 MB partitions for the 10 GB and 12 GB test since it would require over 1,000 files. For that reason, 25 MB partitions were tested on the larger datasets to make sure they could still give good performance while staying within system limits.

With 10 MB partitions, the system scaled very well. As the data size increased, the system continued to process data faster. At 4 GB, it reached a peak performance efficiency of 67.11 GB per second per dollar, meaning it processed a lot of data quickly while using less money. Even at 7.97 GB, performance efficiency stayed high, showing that the system handles large workloads without slowing down much. Interestingly, the 4 GB test with 10 MB partitions did not show much increase in execution time compared to smaller sizes, which shows that the system can manage growth smoothly.

The 25 MB partitions also performed well. Though not quite as efficient as the 10 MB setup, the 4 GB, 6 GB, and 8 GB tests all delivered solid results. The system was able to keep up with larger data without much slowdown, and the performance efficiency stayed above 21 GB per second per dollar. This confirms that 25 MB partitions are a good backup option when smaller ones aren't allowed, and they still give reliable performance at scale.

Performance efficiency, in this case, is a way to measure how much useful work the system does for each dollar spent. It's like asking, "How much processing time can I buy with a small amount of money?" Higher efficiency means you're getting more done at a lower cost, which is especially important when working with large datasets and cloud resources.

## 6. Performance Measurement of Local Computer

Data Size (GB)	Total CPU Time (s)	Memory (MB)	Throughput (GB/s)	Batch Size
<b>Different Batch Sizes @ 0.1 GB Data Size</b>				
0.1	58.2933463	103444.049	0.0144	32
0.1	42.5918925	13404.4516	0.0197	64
0.1	41.0491573	103385.602	0.0204	128
0.1	40.8175005	103378.588	0.0206	256
0.1	42.4505605	103379.021	0.0198	512
<b>Different Partitions @ 256 Batch Size</b>				
1	457.0583523	1036506.2	0.0184	256
2	929.0256521	2119715.09	0.0151	256
4	2273.194203	4228769.87	0.0185	256
6	2758.645132	6347169.12	0.0187	256

### A. Batch Size Comparison (Local vs AWS Lambda)

Testing on a local computer helped give a basic idea of how the system performs without using cloud services. Different batch sizes were tried using a small dataset of 0.1 GB. The best result came from batch size 256, which finished the task the fastest, processing about 0.0206 GB of data per second.

As the batch size got bigger, the time it took went down a little, but the change was not very large.

When larger datasets were used, from 1 GB to 6 GB, the time to finish the job increased a lot, while the speed stayed almost the same at around 0.018 GB per second. During testing, my laptop could not run the 8 GB dataset due to memory limitations, showing that the computer could not keep up as the data got too large.

Overall, these tests show that running jobs on a local computer works fine for smaller amounts of data or early testing. But for larger data, it becomes slower and less effective. That's why using cloud systems is more useful when working with big datasets.