

# Data analysis using Lithops

Dawid Białka, Kamil Burkiewicz

# What is Lithops



Main features:

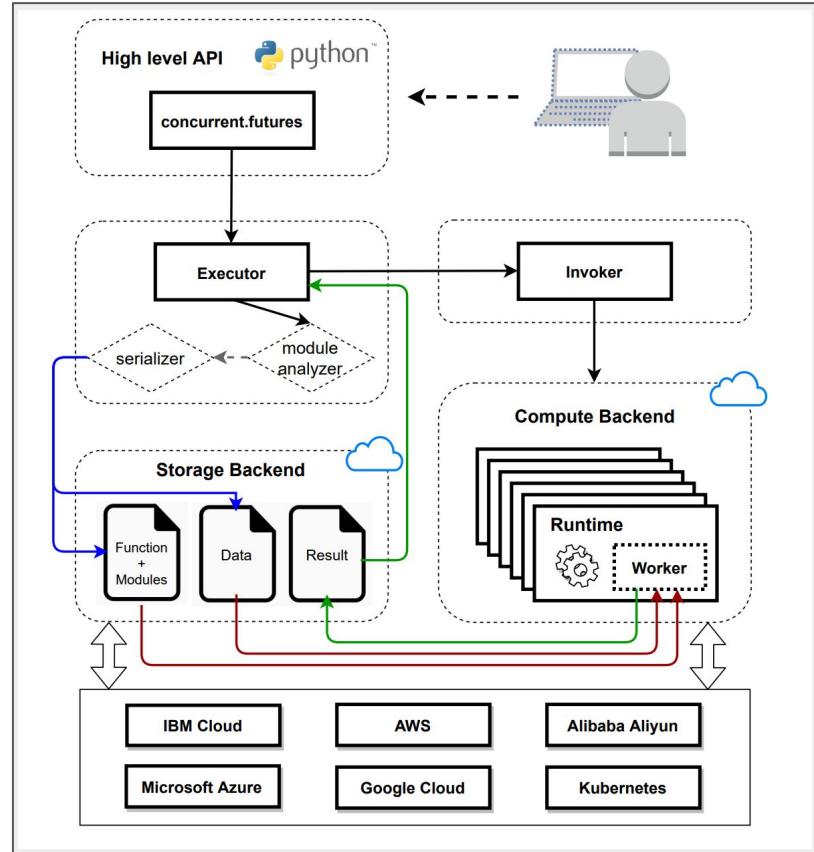
- distributed computing framework
- cloud-agnostic
- suited for highly-parallel jobs

IBM Cloud	aws	Microsoft Azure	Google Cloud	Alibaba Cloud	kubernetes	OPENSIFT
Cloud Functions Code Engine VPC Gen2 ---	AWS Lambda AWS Batch AWS EC2 ---	Functions Container APPs ---	Cloud Functions Cloud Run ---	Functions Compute ---	Batch/Job - OpenWhisk Knative ---	OpenStack Swift - Ceph MinIO - Redis - Infinispan
Cloud Object Storage ---	AWS S3 ---	Blob Storage ---	Cloud Storage ---	Object Storage Service ---		

# Lithops Architecture

## Components:

- Storage:
  - provides abstraction for storage
- Compute:
  - allows running distributed computations
  - Various kinds of Executors:
    - Localhost
    - Serverless
    - Standalone



# Dataset

## AWS OpenAQ

Global, aggregated physical air quality data from public data sources provided by government, research-grade and other sources.

Source: <https://registry.opendata.aws/openaq/>

- ~ 50 GB
- uploaded to s3 in the region where computations were performed for data locality
- divided into files ~ 25 MB each

	date	parameter	value	unit	averagingPeriod	location	city	country	coordinates	attribution	sourceName	sourceType	mobile
0	{'utc': '2020-10-31T06:30:00.000Z', 'local': ...}	pm25	100.000000	µg/m³	{'value': 1, 'unit': 'hours'}	US Diplomatic Post: Kabul	Kabul	AF	{'latitude': 34.535812, 'longitude': 69.190514}	[{'name': 'EPA AirNow DOS', 'url': 'http://air...']}	StateAir_Kabul	government	False
1	{'utc': '2020-10-31T07:30:00.000Z', 'local': ...}	pm25	45.000000	µg/m³	{'value': 1, 'unit': 'hours'}	US Diplomatic Post: Kabul	Kabul	AF	{'latitude': 34.535812, 'longitude': 69.190514}	[{'name': 'EPA AirNow DOS', 'url': 'http://air...']}	StateAir_Kabul	government	False
2	{'utc': '2020-10-31T08:30:00.000Z', 'local': ...}	pm25	46.000000	µg/m³	{'value': 1, 'unit': 'hours'}	US Diplomatic Post: Kabul	Kabul	AF	{'latitude': 34.535812, 'longitude': 69.190514}	[{'name': 'EPA AirNow DOS', 'url': 'http://air...']}	StateAir_Kabul	government	False
3	{'utc': '2020-10-31T09:30:00.000Z', 'local': ...}	pm25	48.000000	µg/m³	{'value': 1, 'unit': 'hours'}	US Diplomatic Post: Kabul	Kabul	AF	{'latitude': 34.535812, 'longitude': 69.190514}	[{'name': 'EPA AirNow DOS', 'url': 'http://air...']}	StateAir_Kabul	government	False
4	{'utc': '2020-10-31T10:30:00.000Z', 'local': ...}	pm25	39.000000	µg/m³	{'value': 1, 'unit': 'hours'}	US Diplomatic Post: Kabul	Kabul	AF	{'latitude': 34.535812, 'longitude': 69.190514}	[{'name': 'EPA AirNow DOS', 'url': 'http://air...']}	StateAir_Kabul	government	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
4995	{'utc': '2020-10-31T12:00:00.000Z', 'local': ...}	so2	6.000000	µg/m³	{'unit': 'hours', 'value': 1}	GR0020A KENTPIKH MAKEDONIA	GR	{'latitude': 40.67354965, 'longitude': 22.8934...}	[{'name': 'EEA', 'url': 'http://www.eea.europa...']}	EEA Greece	government	False	
4996	{'utc': '2020-11-01T04:00:00.000Z', 'local': ...}	pm10	12.500000	µg/m³	{'unit': 'hours', 'value': 1}	FR05090 Seine-Maritime	FR	{'latitude': 49.5146953797856, 'longitude': 0...}	[{'name': 'EEA', 'url': 'http://www.eea.europa...']}	EEA France	government	False	
4997	{'utc': '2020-11-01T04:00:00.000Z', 'local': ...}	o3	44.113987	µg/m³	{'unit': 'hours', 'value': 1}	FI00357 Lapland	FI	{'latitude': 68.477009999634, 'longitude': 28...}	[{'name': 'EEA', 'url': 'http://www.eea.europa...']}	EEA Finland	government	False	
4998	{'utc': '2020-11-01T06:00:00.000Z', 'local': ...}	pm25	2.137920	µg/m³	{'value': 24, 'unit': 'hours'}	Heerlen-Jamboreepad	Heerlen	NL	{'latitude': 50.9003, 'longitude': 5.986850000...}	[{'name': 'RIVM', 'url': 'http://www.lml.rivm...']}	Netherlands	government	False
4999	{'utc': '2020-11-01T06:00:00.000Z', 'local': ...}	no2	0.005300	ppm	{'unit': 'hours', 'value': 1}	Renwu	高雄市	TW	{'latitude': 22.689056, 'longitude': 120.332631}	[{'name': 'http://opendata.epa.gov.tw/', 'url': ...}	Taiwan	government	False

5000 rows x 13 columns

AWS Services Search [Alt+S] Global

Resource Groups & Tag Editor

## Amazon S3

Buckets

- Access Points
- Object Lambda Access Points
- Multi-Region Access Points
- Batch Operations
- Access analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

- Dashboards
- AWS Organizations settings

Feature spotlight: 3

▶ AWS Marketplace for S3

Amazon S3 > Buckets > input-bucket-lithops

### input-bucket-lithops Info

Objects Properties Metrics Management Access Points

To enable sorting in the table below, use the search to reduce the size of the list to 999 objects or fewer.

#### Objects (999+)

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

Copy S3 URI  Copy URL  Download  Open  Delete  Actions  Create folder  Upload

Find objects by prefix

Name	Type	Last modified	Size	Storage class
df0	-	January 9, 2023, 01:32:53 (UTC+01:00)	19.0 MB	Standard
df1	-	January 9, 2023, 01:32:53 (UTC+01:00)	24.1 MB	Standard
df10	-	January 9, 2023, 01:32:55 (UTC+01:00)	24.7 MB	Standard
df100	-	January 9, 2023, 01:38:14 (UTC+01:00)	26.0 MB	Standard
df1000	-	January 9, 2023, 02:20:54 (UTC+01:00)	25.7 MB	Standard
df1001	-	January 9, 2023, 02:20:54 (UTC+01:00)	25.6 MB	Standard
df1002	-	January 9, 2023, 02:20:57 (UTC+01:00)	25.9 MB	Standard
df1003	-	January 9, 2023, 02:20:57 (UTC+01:00)	25.7 MB	Standard
df1004	-	January 9, 2023, 02:20:59 (UTC+01:00)	25.7 MB	Standard
df1005	-	January 9, 2023, 02:21:03 (UTC+01:00)	25.9 MB	Standard
df1006	-	January 9, 2023, 02:21:03 (UTC+01:00)	25.7 MB	Standard
df1007	-	January 9, 2023, 02:21:04 (UTC+01:00)	25.7 MB	Standard
df1008	-	January 9, 2023, 02:21:04 (UTC+01:00)	25.8 MB	Standard
df1009	-	January 9, 2023, 02:21:06 (UTC+01:00)	25.6 MB	Standard
df101	-	January 9, 2023, 01:38:16 (UTC+01:00)	25.9 MB	Standard
df1010	-	January 9, 2023, 02:21:09 (UTC+01:00)	25.8 MB	Standard
df1011	-	January 9, 2023, 02:21:23 (UTC+01:00)	25.6 MB	Standard
df1012	-	January 9, 2023, 02:21:24 (UTC+01:00)	25.7 MB	Standard
df1013	-	January 9, 2023, 02:21:24 (UTC+01:00)	26.0 MB	Standard
df1014	-	January 9, 2023, 02:21:28 (UTC+01:00)	25.7 MB	Standard
df1015	-	January 9, 2023, 02:21:32 (UTC+01:00)	25.8 MB	Standard
df1016	-	January 9, 2023, 02:21:34 (UTC+01:00)	25.7 MB	Standard
df1017	-	January 9, 2023, 02:21:36 (UTC+01:00)	25.8 MB	Standard
df1018	-	January 9, 2023, 02:21:37 (UTC+01:00)	25.9 MB	Standard

# Computation of average levels of air pollution with map-reduce model

```
✓ [25] def day_average_map(x):
    if isinstance(x, str):
        df = pd.DataFrame.loads_recursive(json.loads(download_bytes(os.path.basename(x), input_bucket).decode(encoding='utf-8'))))
    else:
        df = x
    df_to_process = preprocess(df)
    df_sum = df_to_process.groupby(['country', 'city', 'date', 'parameter', 'latitude', 'longitude'])['value'].agg(['sum','count'])
    df_to_process = df_to_process.drop(columns=['value'], axis=1)
    df_merged = pd.merge(df_sum, df_to_process, on=['country', 'city', 'date', 'parameter', 'latitude', 'longitude']).drop_duplicates()
    return df_merged.to_json()

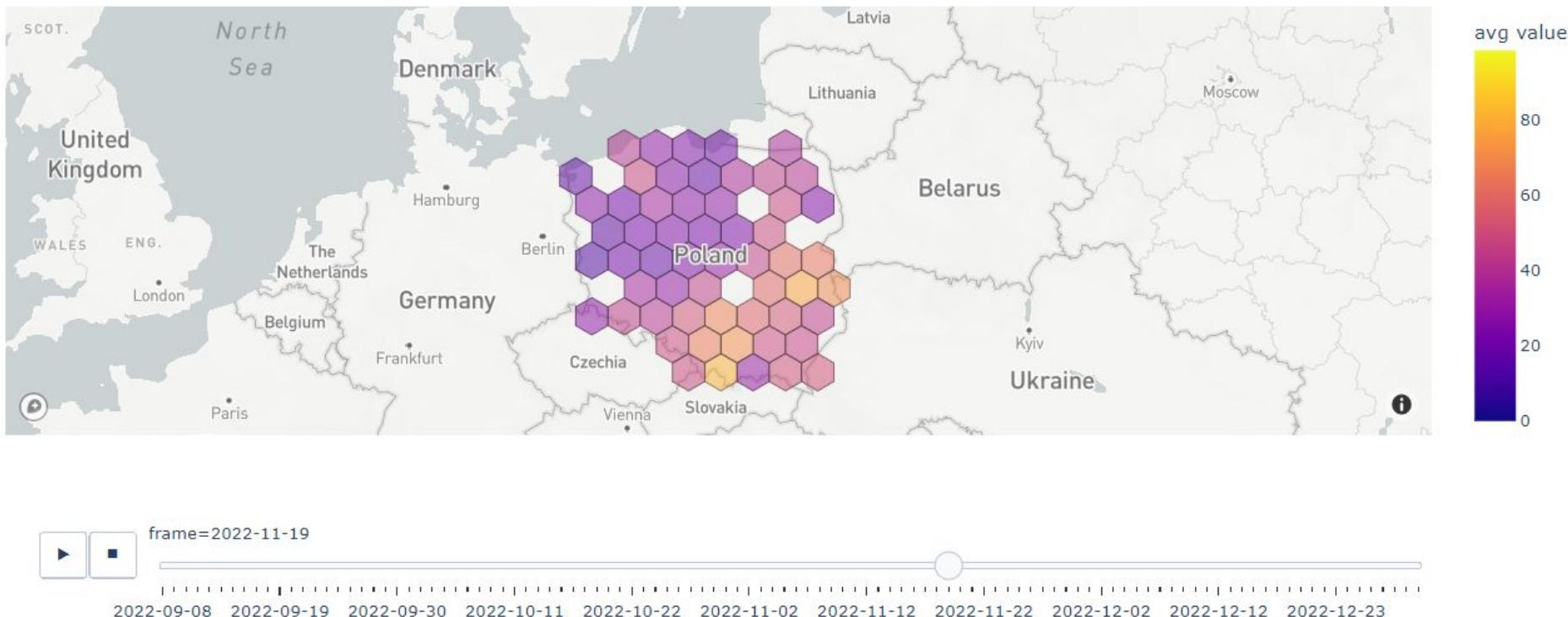
✓ [26] def day_average_reduce(results):
    dfs = [pd.read_json(result) for result in results]
    df_concatenated = pd.concat(dfs)
    df_concatenated.reset_index(drop=True, inplace=True)

    df_avg = df_concatenated.groupby(['country', 'city', 'date', 'parameter', 'latitude', 'longitude']).sum(['count', 'sum'])
    df_avg['average'] = df_avg.apply(lambda row: row['sum'] / row['count'], axis=1)

    df_avg = df_avg.drop(columns=['count', 'sum'], axis=1).dropna()
    df_concatenated = df_concatenated.drop(columns=['count', 'sum'], axis=1).dropna()

    df_merged = pd.merge(df_avg, df_concatenated, on=['country', 'city', 'date', 'parameter', 'latitude', 'longitude']).drop_duplicates()
    return df_merged.to_json()
```

# Map of average concentration of PM10 in Poland based on localization and time



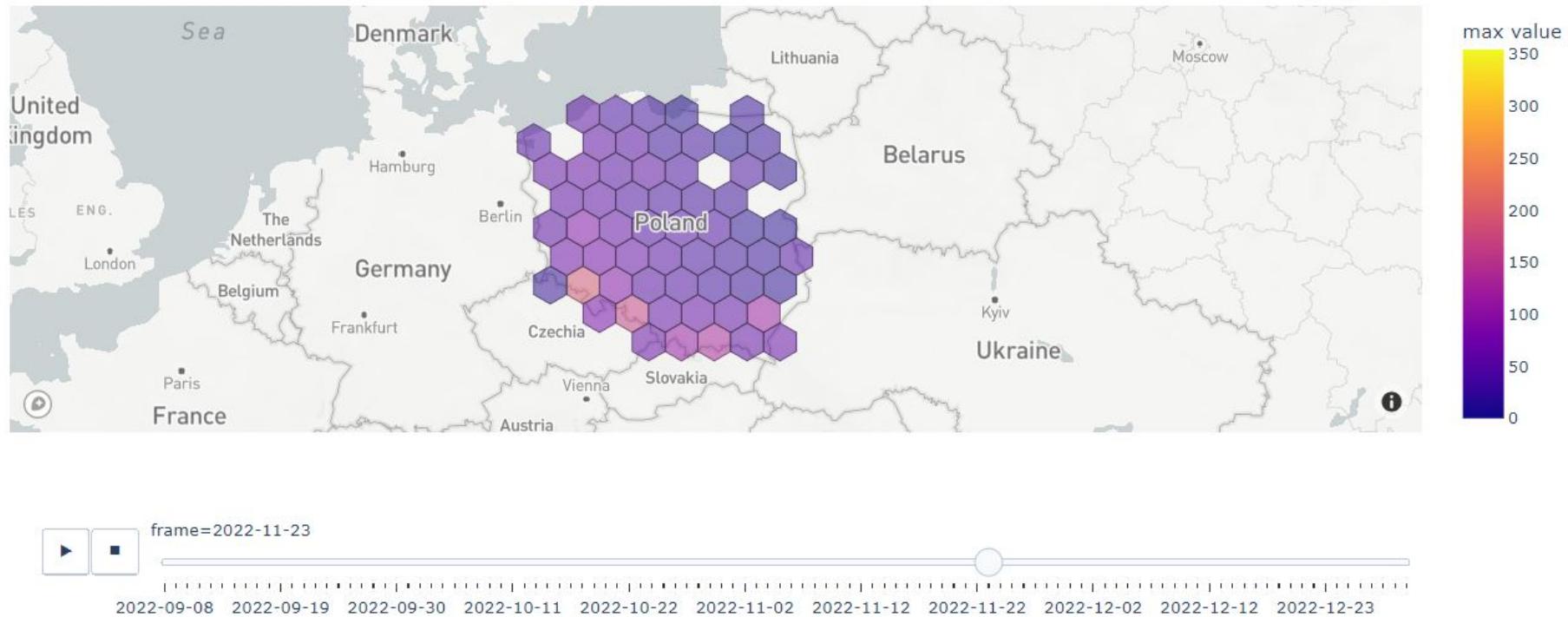
# Computation of maximum levels of air pollution with map-reduce model

```
[ ] def day_max_map(x):
    df = pd.DataFrame.loads_recursive(json.loads(download_bytes(os.path.basename(x), input_bucket).decode(encoding='utf-8'))))
    df_to_process = preprocess(df)
    df_sum = df_to_process.groupby(['country', 'city', 'date', 'parameter', 'latitude', 'longitude'])['value'].agg(['max'])
    df_to_process = df_to_process.drop(columns=['value'], axis=1)

    df_merged = pd.merge(df_sum, df_to_process, on=['country', 'city', 'date', 'parameter', 'latitude', 'longitude']).drop_duplicates()
    return df_merged.to_json()

def day_max_reduce(results):
    dfs = [pd.read_json(result) for result in results]
    df_concatenated = pd.concat(dfs)
    df_concatenated.reset_index(drop=True, inplace=True)
    return df_concatenated.to_json()
```

# Map of maximum concentration of PM10 in Poland based on localization and time



# Measurements

Compute backend: AWS Lambda

- 2048 MB memory
- cpu type: Intel(R) Xeon(R) Processor @ 2.50GHz model 63
- timeout: 900s
- vCPUs: 2 (scaled to 1.15)\*
- lithops invokes lambdas asynchronously by design

Storage: S3

\*lambda documentation states: At 1769 MB, a function has the equivalent of one vCPU, source:  
<https://docs.aws.amazon.com/lambda/latest/dg/configuration-function-common.html>

# Warmup

To avoid cold start there was a warmup.

990 Lambdas processing whole dataset in about a minute

## ▼ Warmup

```
✓ [21] fexec = lithops.ServerlessExecutor(config=config)
fexec.map(map_function=day_average_map, map_iterdata=objs, chunksize=2)
result = fexec.get_result()

1 min
```

```
2023-01-16 02:10:12,323 [INFO] config.py:131 -- Lithops v2.7.1
2023-01-16 02:10:12,339 [INFO] aws_s3.py:60 -- S3 client created - Region: us-east-1
2023-01-16 02:10:12,422 [INFO] aws_lambda.py:94 -- AWS Lambda client created - Region: us-east-1
2023-01-16 02:10:12,427 [INFO] invokers.py:108 -- ExecutorID cc0069-1 | JobID M000 - Selected Runtime: lithops-default-runtime-v38 - 1024MB
2023-01-16 02:10:12,711 [INFO] invokers.py:172 -- ExecutorID cc0069-1 | JobID M000 - Starting function invocation: day_average_map() - Total: 1979 activations
2023-01-16 02:10:14,648 [INFO] invokers.py:208 -- ExecutorID cc0069-1 | JobID M000 - View execution logs at /tmp/lithops/logs/cc0069-1-M000.log
2023-01-16 02:10:14,970 [INFO] wait.py:97 -- ExecutorID cc0069-1 - Getting results from 1979 function activations
```

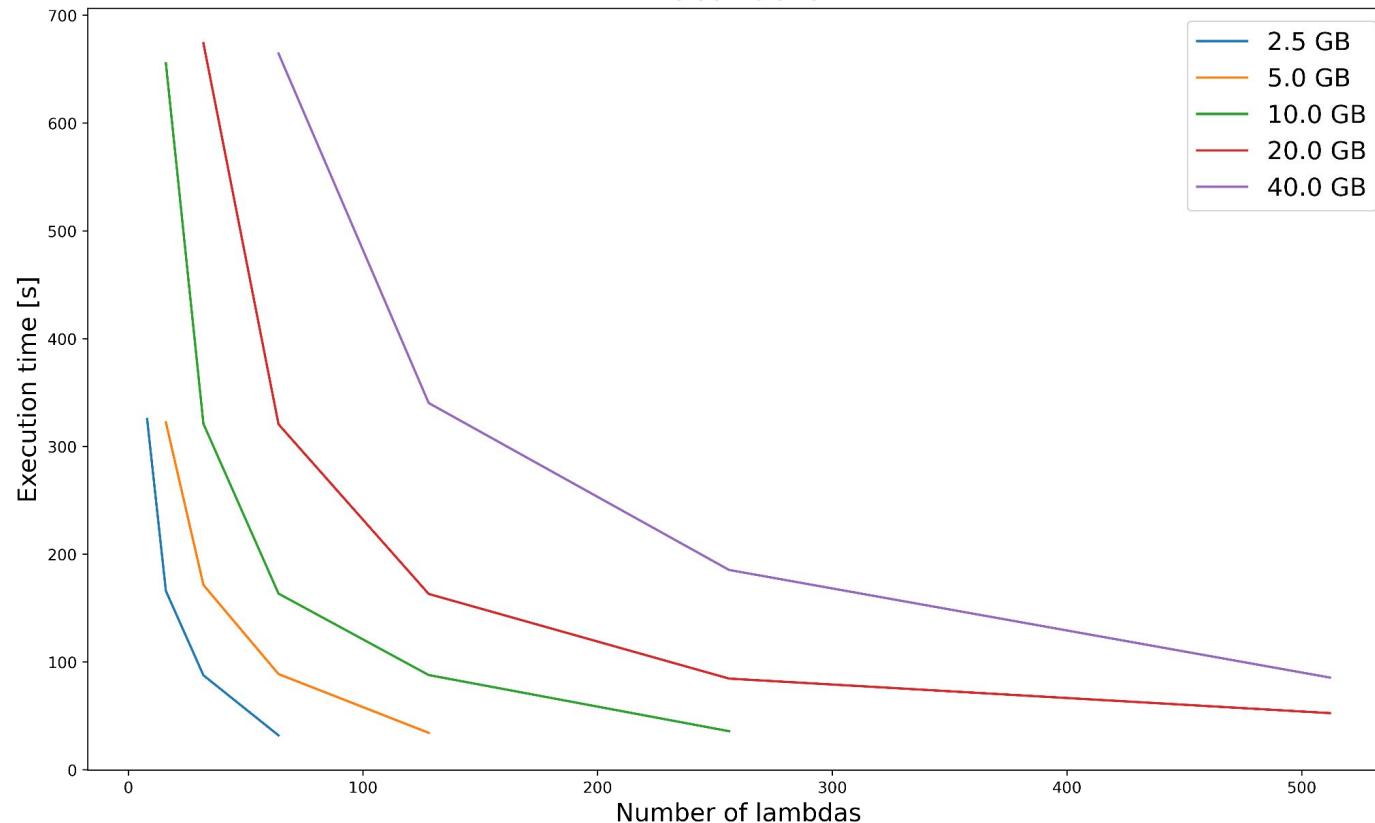
100%  1979/1979

2023-01-16 02:11:22,127 [INFO] executors.py:609 -- ExecutorID cc0069-1 - Cleaning temporary data

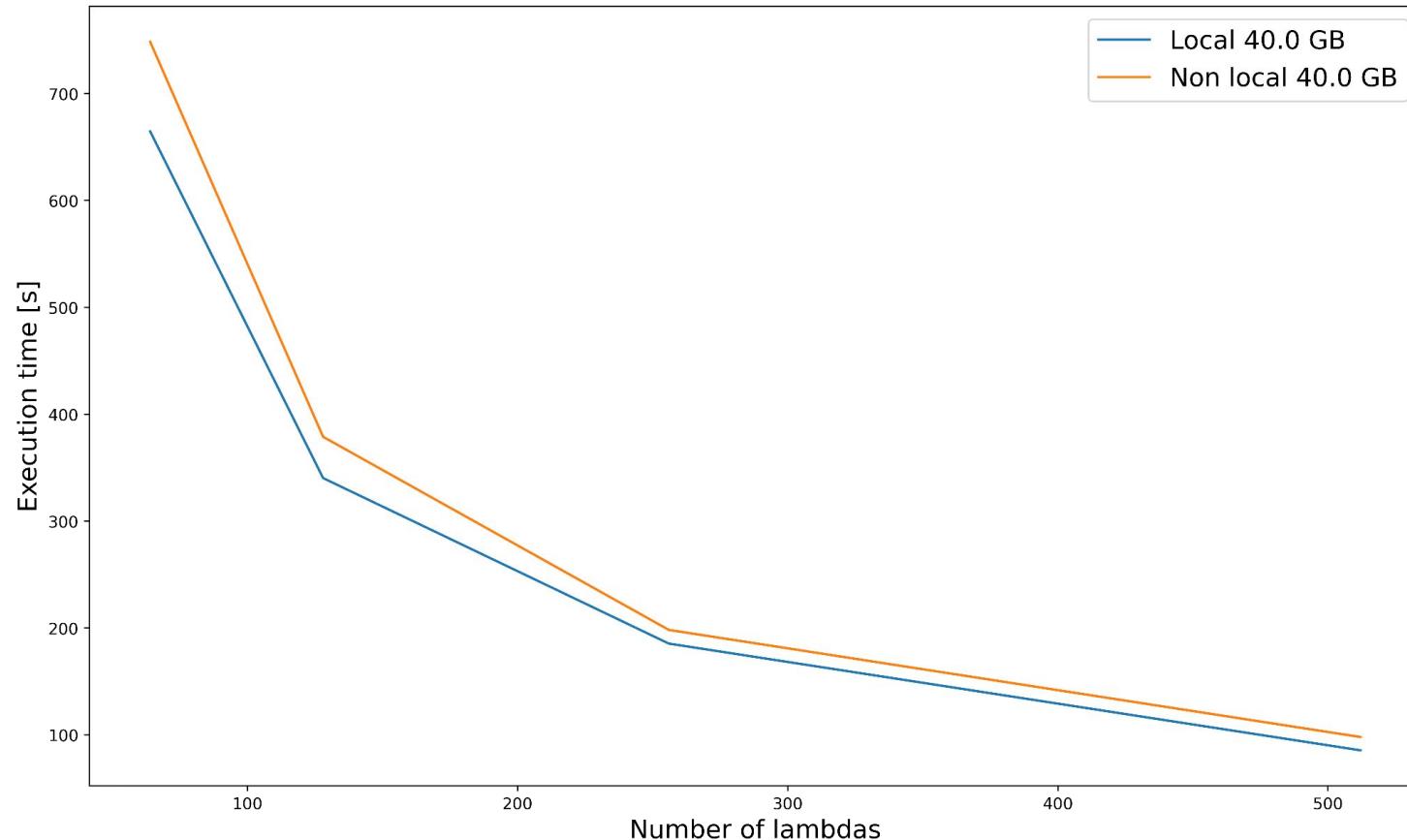


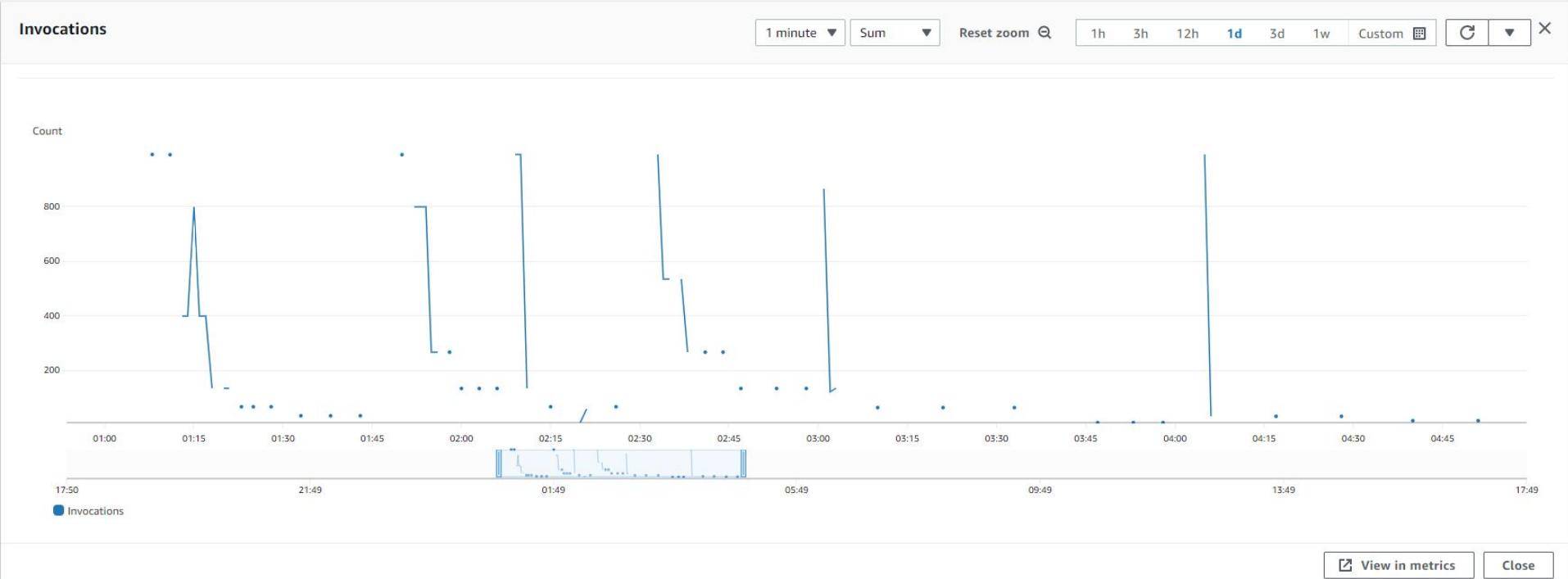
# Scalability

Local data



# Scalability





Duration 1 minute (multiple) Reset zoom 

1h

3h

12h

**1d**

3d

1w

Custom 

Milliseconds

700k

600k

500k

400k

300k

200k

100k

0

01:00 01:15 01:30 01:45 02:00 02:15 02:30 02:45 03:00 03:15 03:30 03:45 04:00 04:15 04:30 04:45 05:00

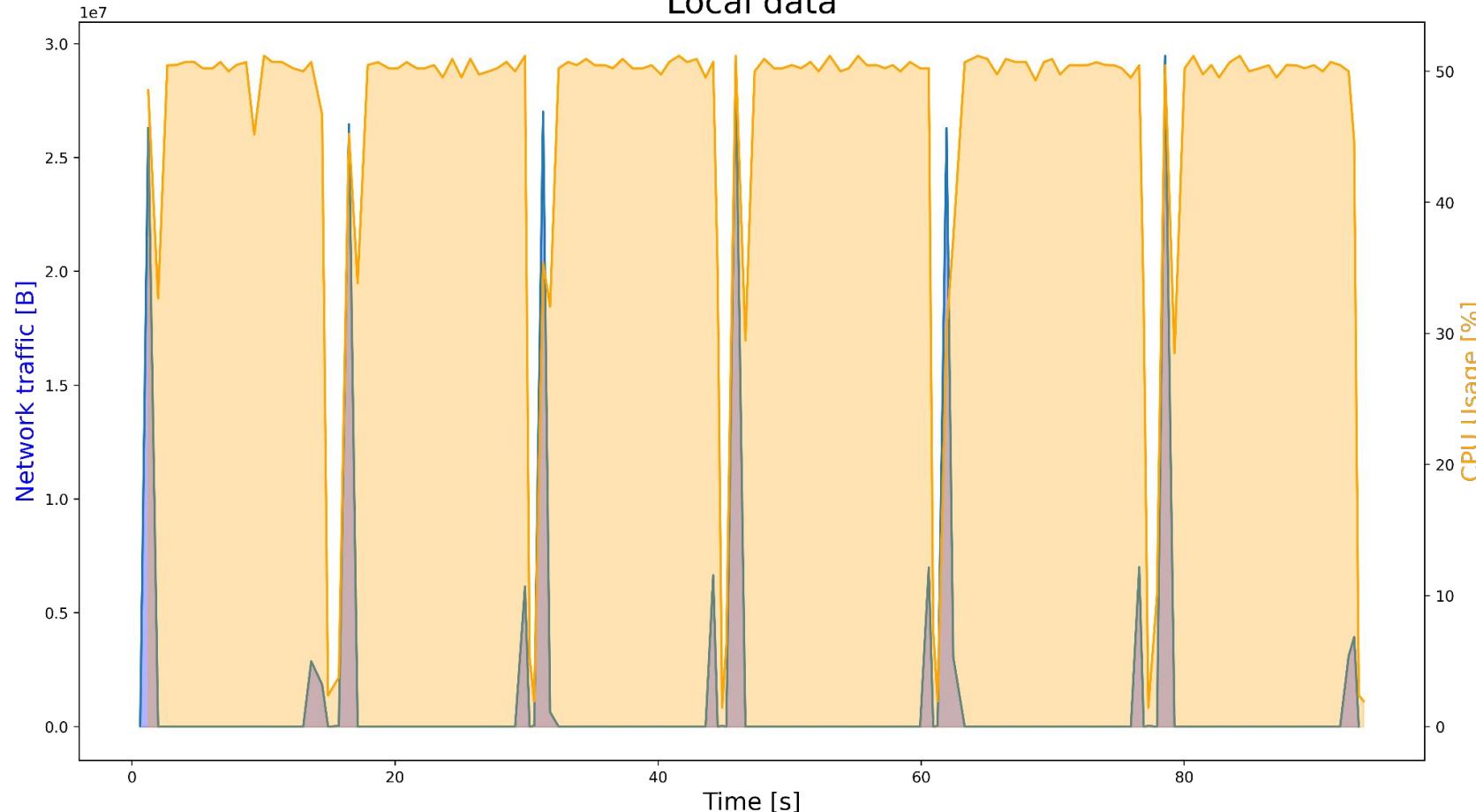
17:52 21:51 01:51 05:51 09:51 13:51 17:51

 Duration minimum  Duration average  Duration maximum View in metrics

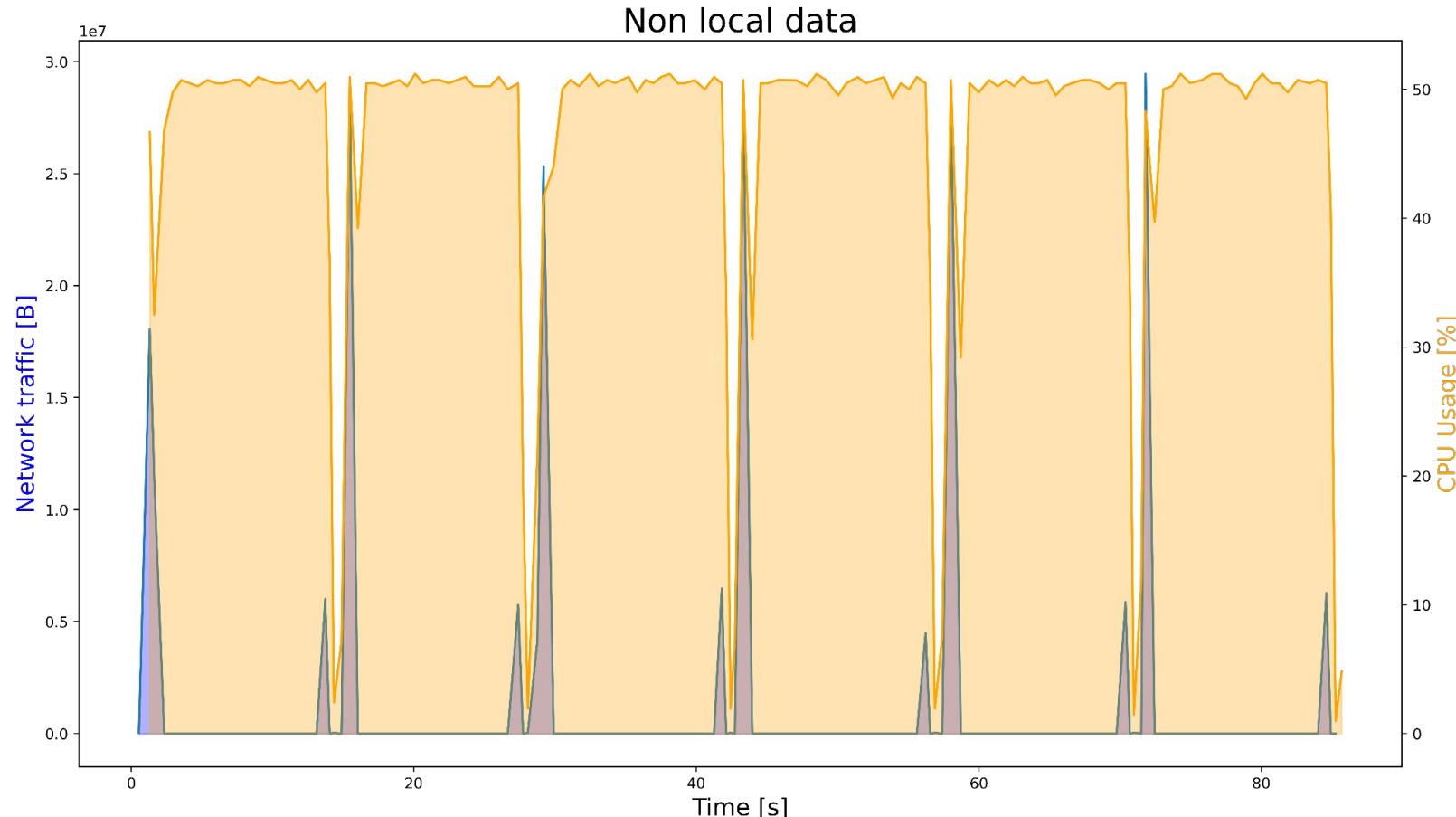
Close

# CPU and network usage

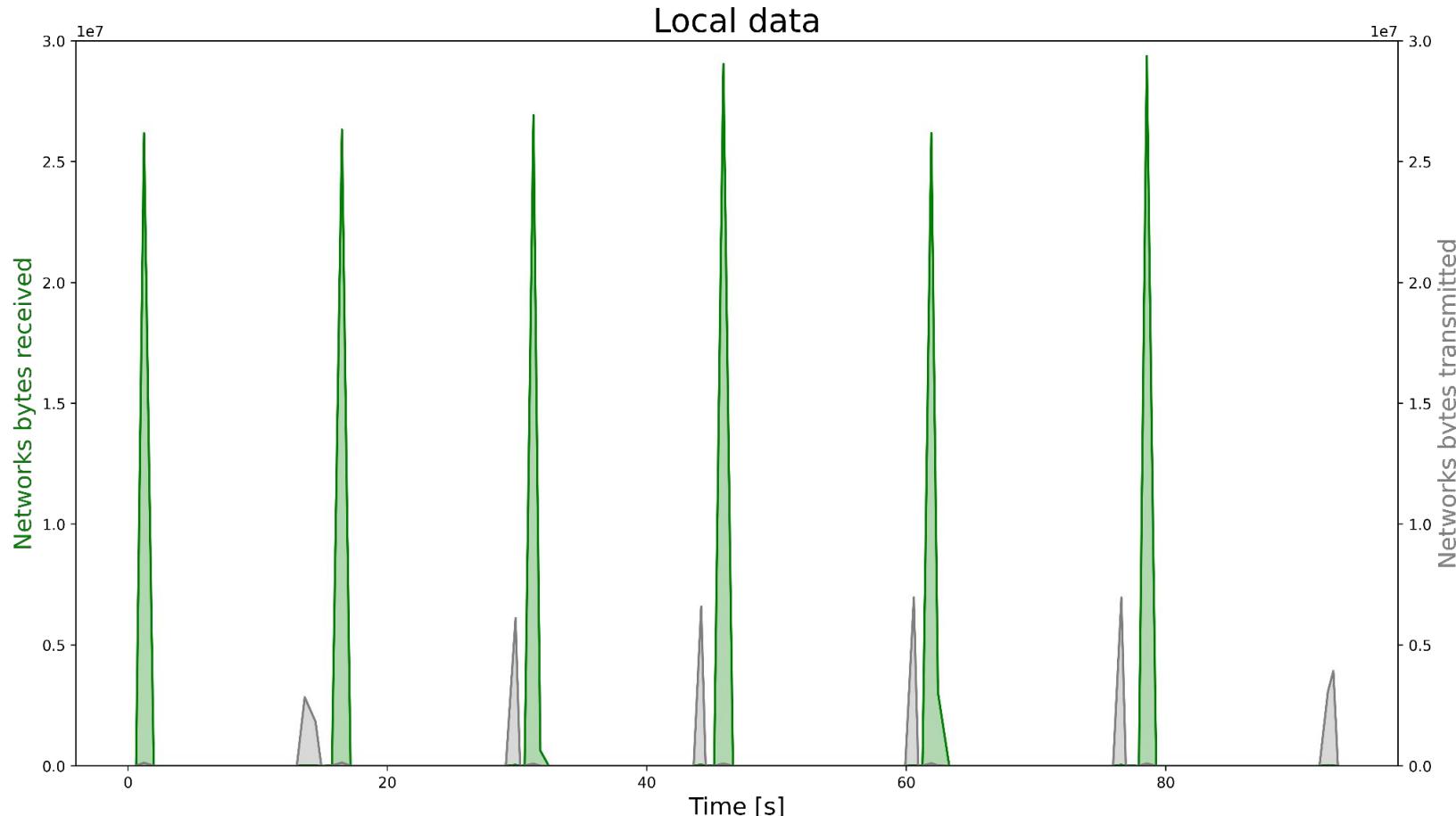
Local data



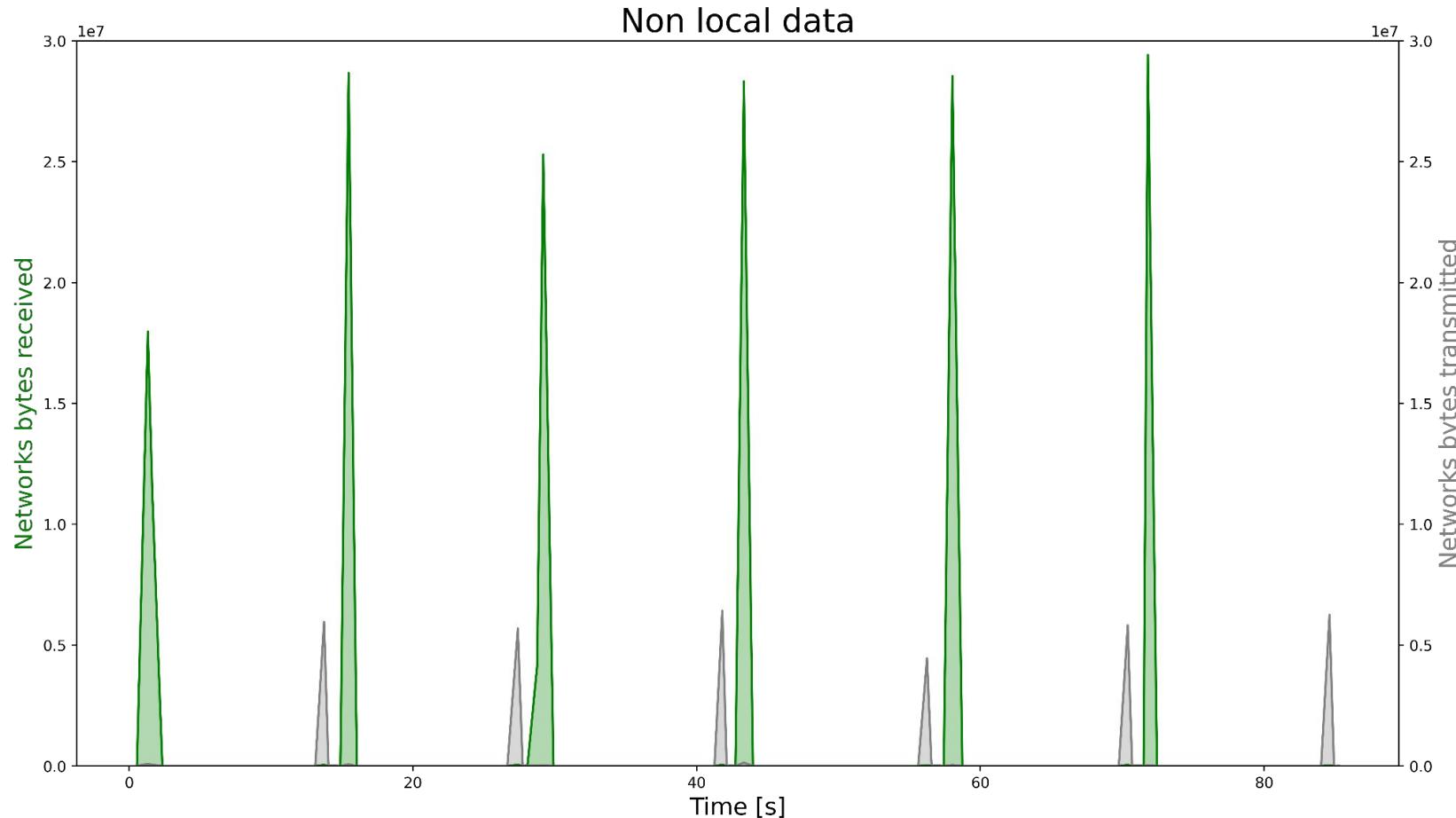
# CPU and network usage



# Network transmission

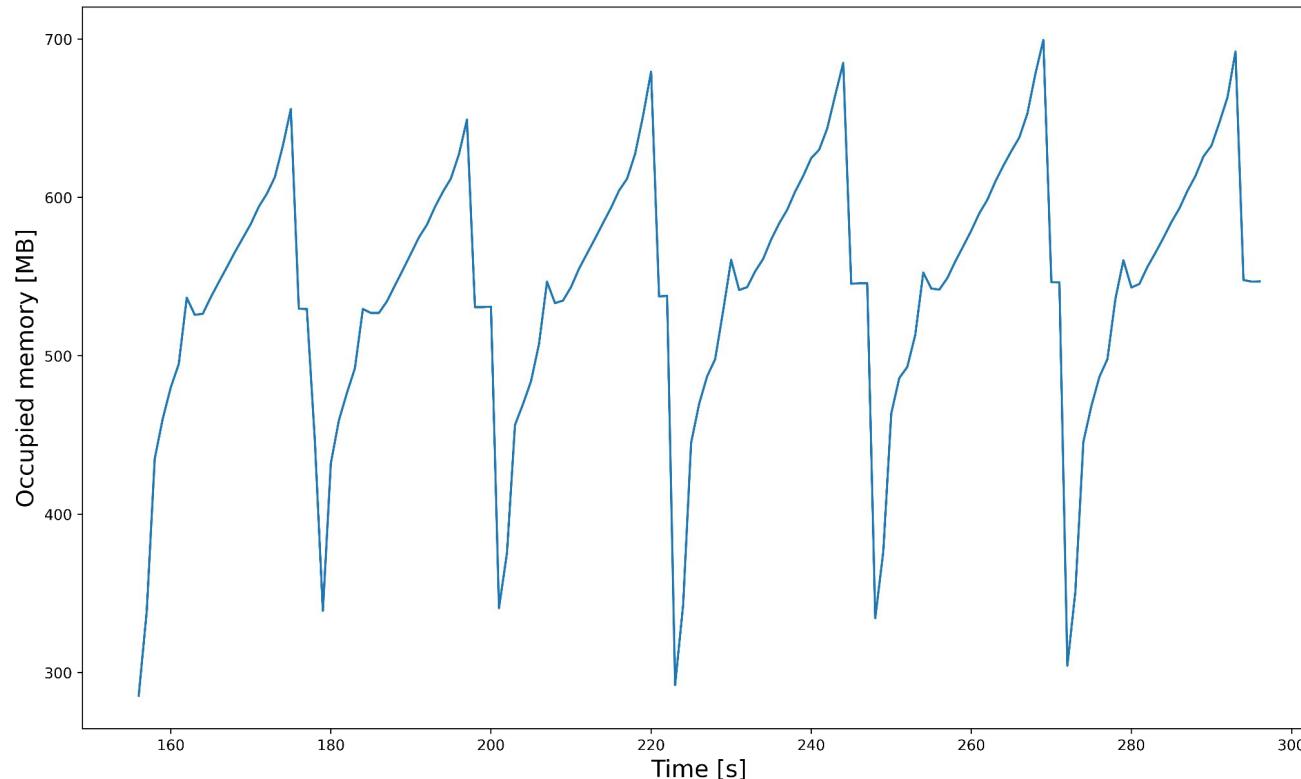


# Network transmission



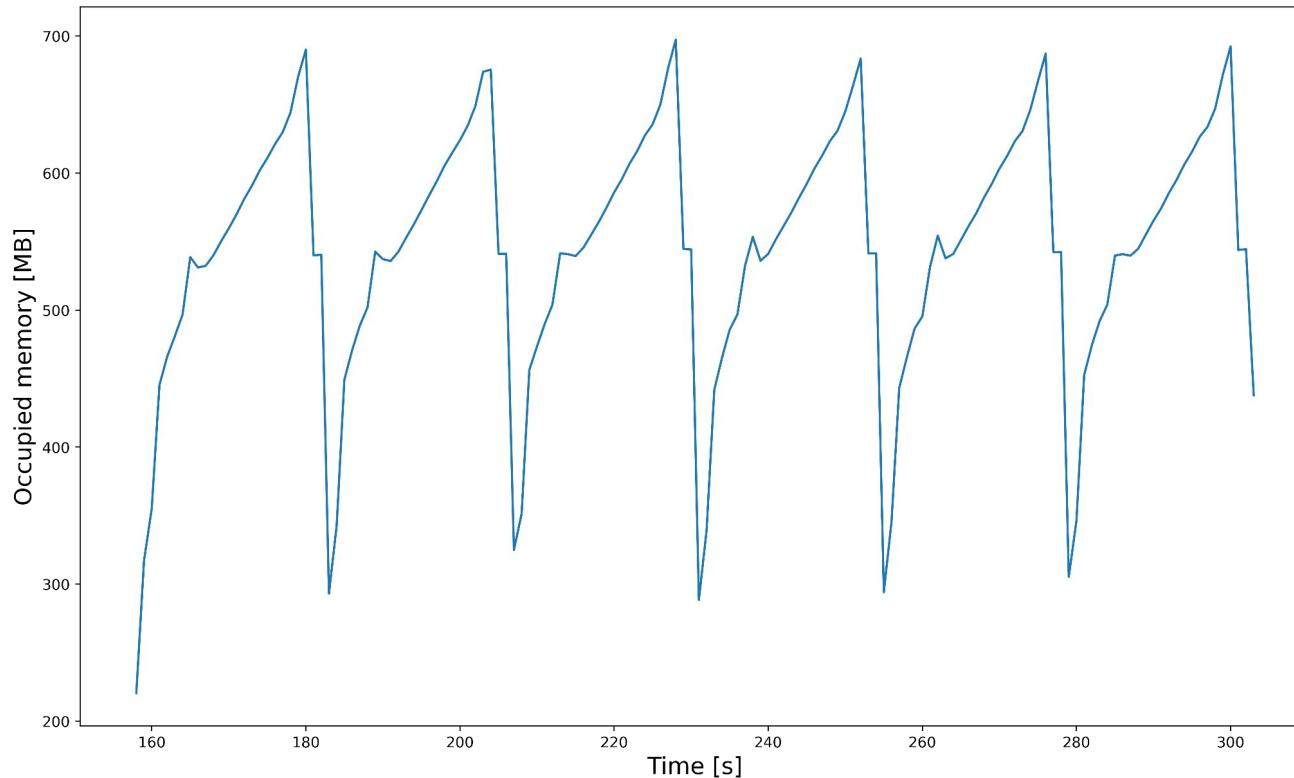
# Occupied memory

Local data

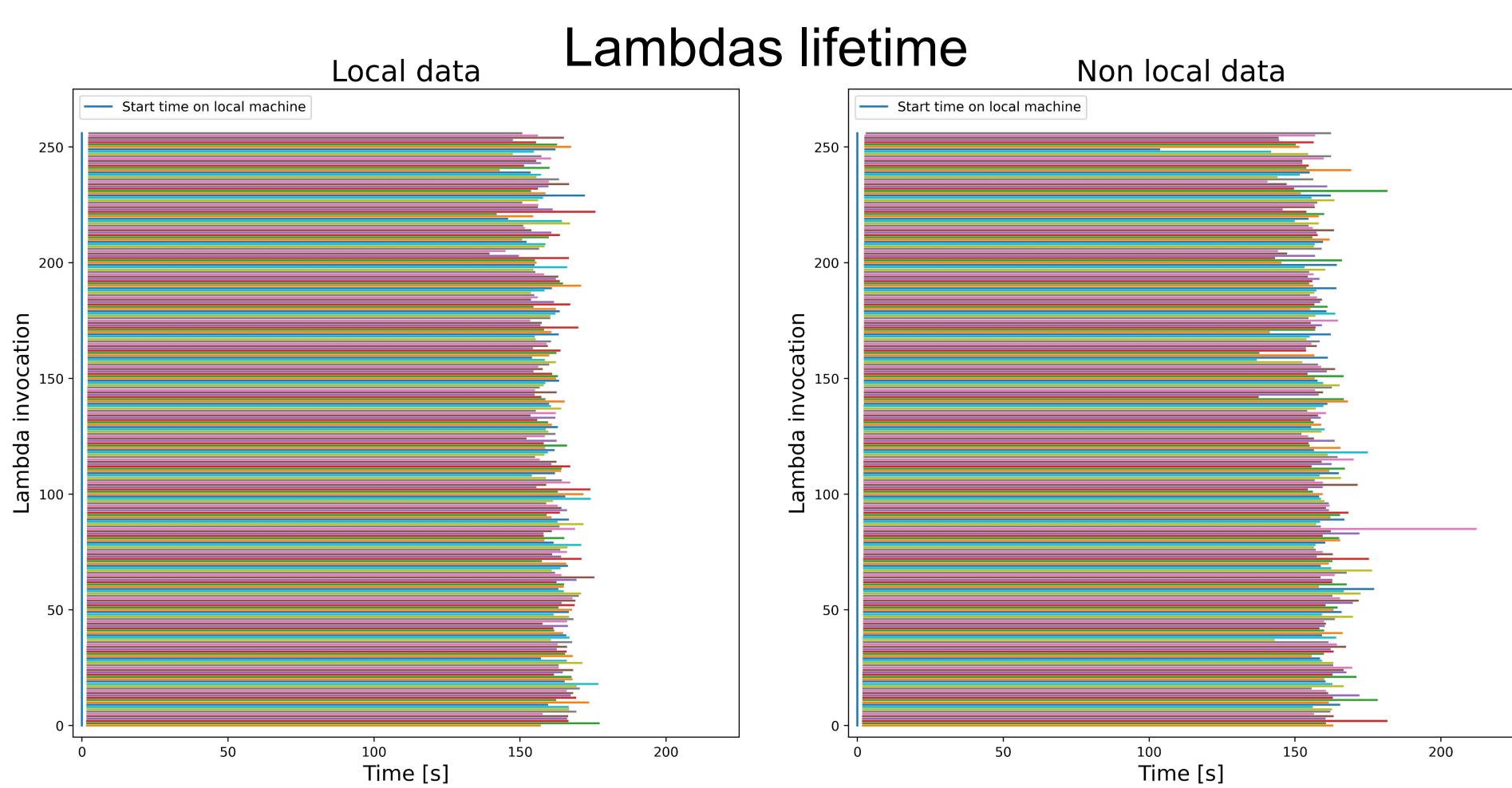


# Occupied memory

Non local data



# Lambdas lifetime



# Cost

Cost and usage <a href="#">Info</a>		
Current month costs		
\$65.70		
Forecasted month end costs		
\$2.10		Up 483% over last month
Last month costs		
\$0.36		
Costs shown are unblended. <a href="#">Learn more</a>		
<a href="#">Go to AWS Cost Management</a>		

First day of testing

Second day of testing

Dear Kamil Burkiewicz,

This is a friendly notification that you have exceeded 100% of your available credits.

Email : 303737@student.agh.edu.pl

Account Type : ALLv1-30213-a034N00000iZJqTQAW ( 1-[ACLLA]-Lab - LabLearner - Associate )

Expires : 11-03-2022 PDT

Credits Remaining : -\$24.42 out of \$100

Your account is now deactivated.

Regards,

Vocareum Support

31 min

```

import time
fexec = lithops.ServerlessExecutor(config=config)
start_time = time.time()
print(start_time)
fexec.map(map_function=day_average_map, map_iterdata=objs, chunksize=len(objs))
result = fexec.get_result()
print(time.time() - start_time)

```

```

2023-01-17 20:58:35,279 [INFO] config.py:131 -- Lithops v2.7.1
2023-01-17 20:58:35,300 [INFO] aws_s3.py:60 -- S3 client created - Region: us-east-1
2023-01-17 20:58:35,537 [INFO] aws_lambda.py:94 -- AWS Lambda client created - Region: us-east-1
2023-01-17 20:58:35,549 [INFO] invokers.py:108 -- ExecutorID 7b5093-5 | JobID M000 - Selected Runtime: lithops-default-runtime-v38 - 1024MB
2023-01-17 20:58:35,746 [INFO] invokers.py:116 -- Runtime lithops-default-runtime-v38 with 1024MB is not yet deployed
1673989115.5475168
2023-01-17 20:58:35,748 [INFO] aws_lambda.py:395 -- Deploying runtime: lithops-default-runtime-v38 - Memory: 1024 Timeout: 900
2023-01-17 20:58:35,944 [INFO] aws_lambda.py:194 -- Creating default lambda layer for runtime lithops-default-runtime-v38
2023-01-17 21:00:19,016 [INFO] invokers.py:172 -- ExecutorID 7b5093-5 | JobID M000 - Starting function invocation: day_average_map() - Total: 1099 activations
2023-01-17 21:00:19,027 [INFO] invokers.py:208 -- ExecutorID 7b5093-5 | JobID M000 - View execution logs at /tmp/lithops/logs/7b5093-5-M000.log
2023-01-17 21:00:19,047 [INFO] wait.py:97 -- ExecutorID 7b5093-5 - Getting results from 1099 function activations

```

2% 19/1099

```
2023-01-17 21:30:15,902 [INFO] executors.py:609 -- ExecutorID 7b5093-5 - Cleaning temporary data
```

---

```
----- TimeoutError Traceback (most recent call last)
```

```
<ipython-input-48-27a15cdfb44b> in <module>
      4 print(start_time)
      5 fexec.map(map_function=day_average_map, map_iterdata=objs, chunksize=len(objs))
--> 6 result = fexec.get_result()
      7 print(time.time() - start_time)
```

▼ 14 frames

```
/usr/local/lib/python3.8/dist-packages/lithops/monitor.py in _future_timeout_checker()
  106         if current_time > fut_timeout:
  107             msg = 'The function did not run as expected.'
--> 108             raise TimeoutError('HANDLER', msg)
  109     except TimeoutError:
  110         # generate fake TimeoutError call status
```

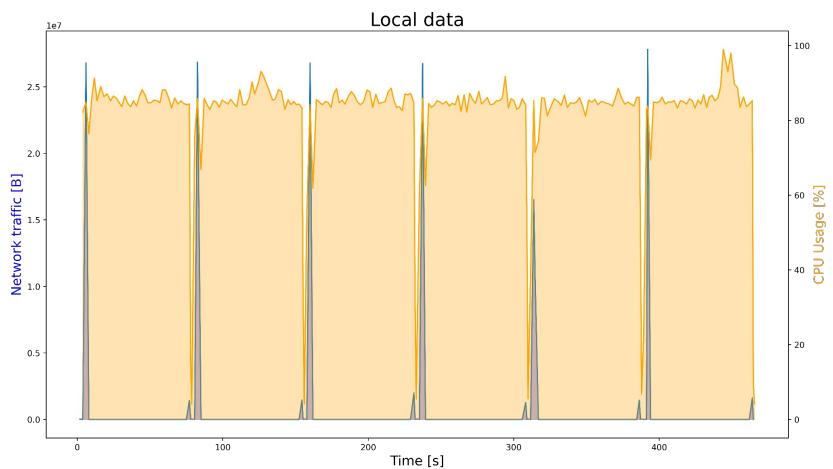
TimeoutError: The function did not run as expected.

SEARCH STACK OVERFLOW

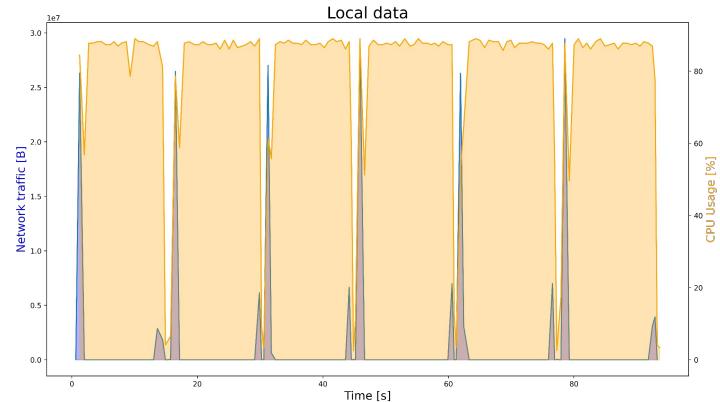
[ ]

# Sources

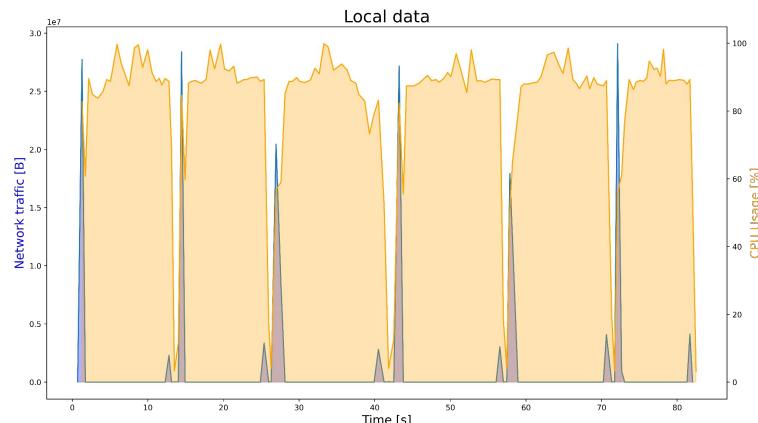
- J. Sampe, M. Sanchez-Artigas, G. Vernik, I. Yehekzel and P. Garcia-Lopez, "Outsourcing Data Processing Jobs with Lithops," in IEEE Transactions on Cloud Computing, doi: [10.1109/TCC.2021.3129000](https://doi.org/10.1109/TCC.2021.3129000)



Scaled 1024 MB



Scaled 2048 MB



Scaled 3538 MB

$$\text{scaled} = \text{utilization} \cdot \frac{\text{number of pCPU}}{\text{estimated number of vCPU}}$$



Search or jump to...

Pull requests Issues Codespaces Marketplace Explore



lithops-cloud / lithops Public

Edit Pins Watch 11 Fork 81 Star 251

Code Issues 3 Pull requests Discussions Projects Wiki Security Insights

## [AWS] Added session token as optional #1026

Merged

JosepSampe merged 1 commit into [lithops-cloud:master](#) from [Kamilbur:aws-session-token](#) yesterday

Edit

Code



Conversation 1

Commits 1

Checks 0

Files changed 17

+47 -3



Kamilbur commented 2 days ago

I've added an option to pass a session token from AWS credentials. Session tokens are included f.e. in credentials for student accounts.

Tested it with [lithops test](#) with credentials from my AWS Academy account.

Also included info about those changes in docs.

### Developer's Certificate of Origin 1.1

By making a contribution to this project, I certify that:

- (a) The contribution was created in whole or in part by me and I have the right to submit it under the Apache License 2.0; or
- (b) The contribution is based upon previous work that, to the best of my knowledge, is covered under an appropriate open source license and I have the right under that license to submit that work with modifications, whether created in whole or in part by me, under the same open source license (unless I am permitted to submit under a different license), as indicated in the file; or
- (c) The contribution was provided directly to me by some other person who certified (a), (b) or (c) and I have not modified it.
- (d) I understand and agree that this project and the contribution are public and that a record of the contribution (including all personal information I submit with it, including my sign-off) is maintained indefinitely and may be redistributed consistent with this project or the open source license(s) involved.

[AWS] Added session token as optional

### Reviewers

JosepSampe

### Assignees

No one assigned

### Labels

None yet

### Projects

None yet

### Milestone

No milestone

### Development

Successfully merging this pull request may close these issues.

None yet

### Notifications

Customize

Unsubscribe

You're receiving notifications because you were mentioned.

### 2 participants

 Allow edits by maintainers

JosepSampe reviewed yesterday

View changes

JosepSampe left a comment

(Member)

Cool! Thanks @Kamilbur

Memory	vCPUs
128 - 3008 MB	2
3009 - 5307 MB	3
5308 - 7076 MB	4
7077 - 8845 MB	5
8846+ MB	6