Bringing Scaling Transparency To Proteomic Applications With Serverless Computing

Presented at: Sixth International Workshop on Serverless Computing (WoSC6) 2020

Gil Vernik, Pedro Garcia-López, Mariano Mirabelli

Tuesday, December 8, 2020

IN WHAT CONTEXT DID THIS WORK TAKE PLACE?

- *CloudButton Project*
- *In collaboration with URV CLOUDLAB*
- *In collaboration with IBM*

WHAT ARE THE AIMS OF THIS WORK?

- *Move Proteomics Work Queue application to Serverless.*
- *Demonstrate scaling transparency.*
- *Show cost/performance optimisations.*

WHAT SCALING TRANSPARENCY MEANS?

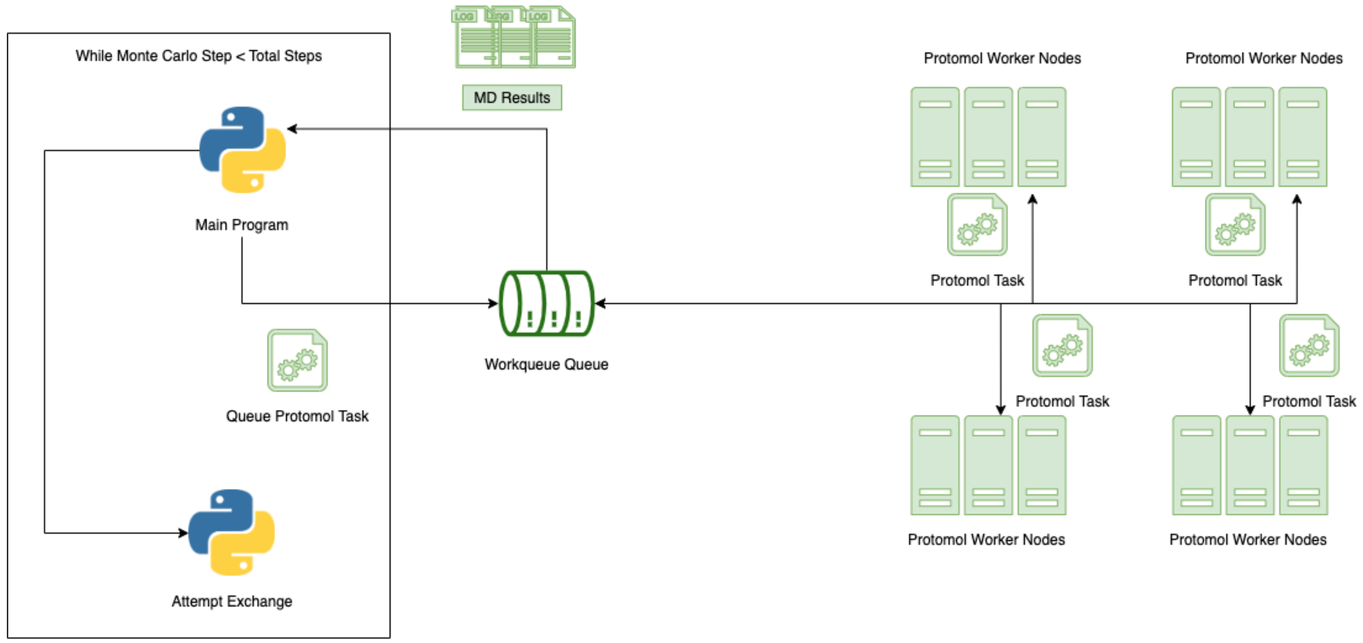
- *“ Scaling transparency means that applications can expand in scale without changes to the system structure or the application algorithms.”[1]*

1. Ansa Reference Manual[ANSA 1989] and the International Organisation for Standardization’s Reference Model for Open Distributed Processing(RM-ODP)[ISO 1992]

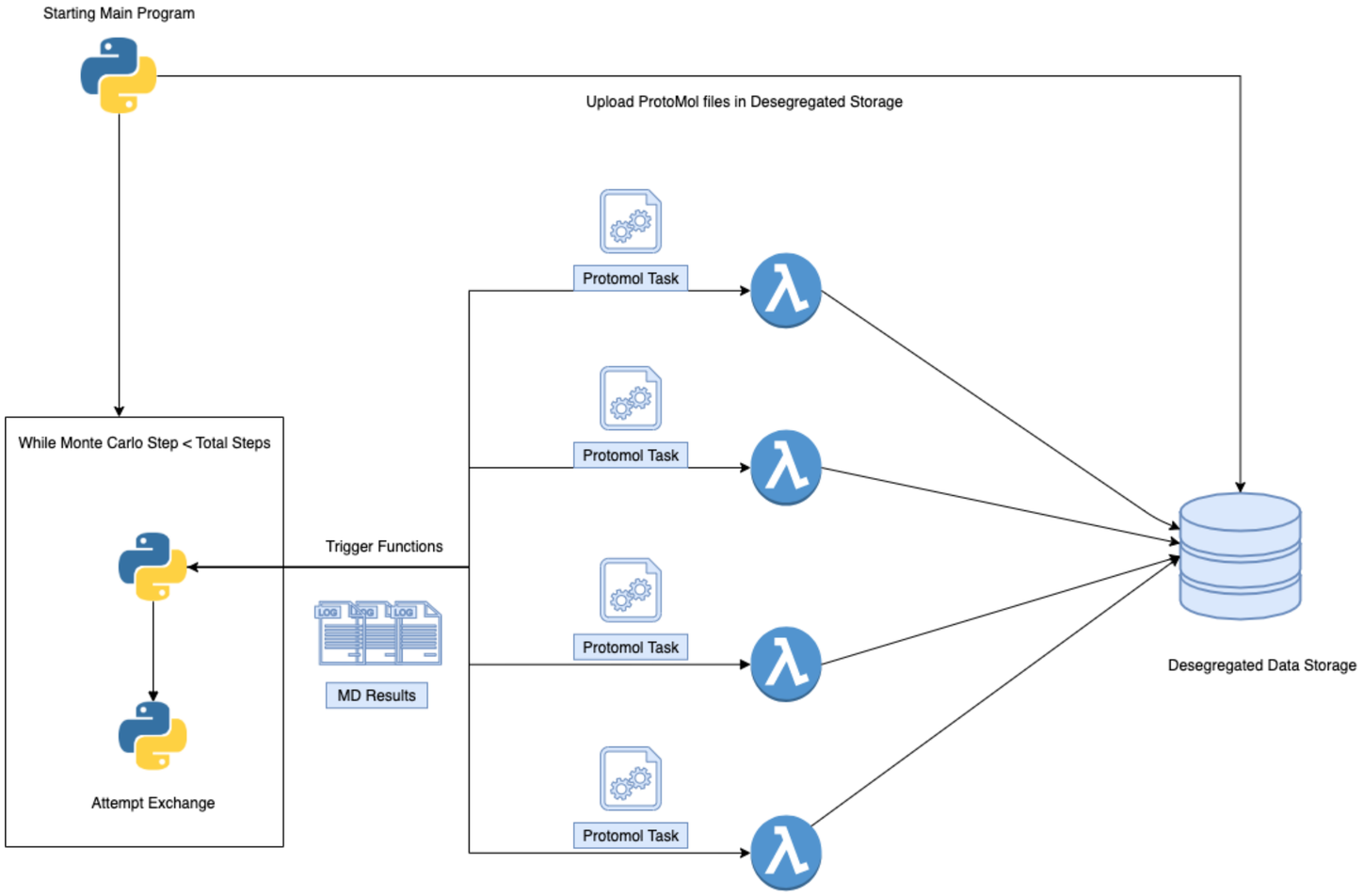
WHAT IS OUR WORK BASED ON?

- *Work Queue framework*
- *Replica Exchange algorithm*
- *ProtoMol framework*

REPLICA EXCHANGE WORK QUEUE ARCHITECTURE



REPLICA EXCHANGE SERVERLESS ARCHITECTURE



SERVERLESS PROTOTYPES

LITHOPS

https://github.com/lithops-cloud/lithops

COS Prototype

Redis Prototype

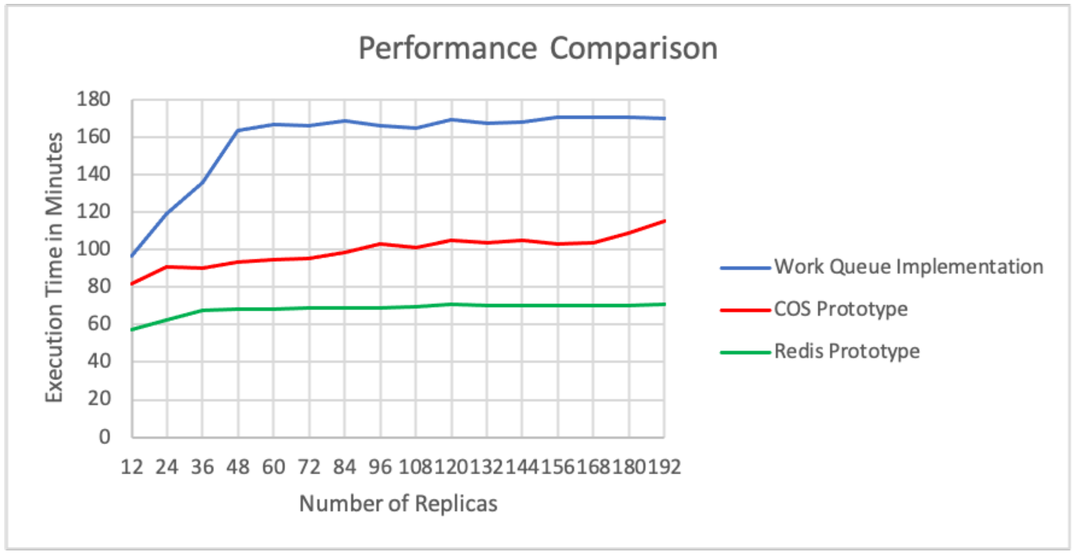
EXPERIMENT SETUP

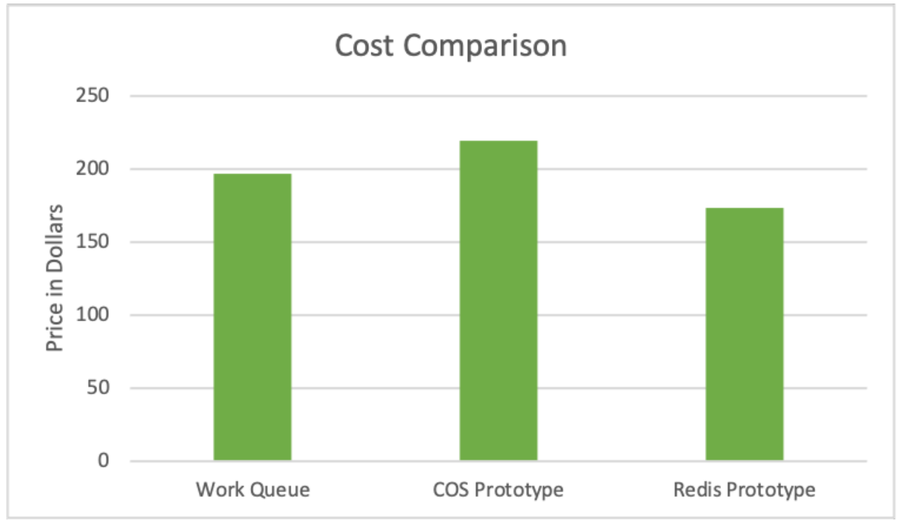
Replica Exchange Configuration and Replicas by Execution

Replica Exchange Values	
Default Monte Carlo Steps	100
Default MD Steps	10000
Default Boundary Conditions	Vacuum
Default Output Frequency	10000
Default Physical Temperature	300
Minimum Temperature	300
Maximum Temperature	400

Replicas Variation	
Initial Number of Replicas	12
Replicas Delta	12
Ending Number of Replicas	192

RESULTS OBTAINED





RESULTS OBTAINED

Why is Redis Prototype cheaper?

Implementation	Work Queue	COS	Redis
Avg Function Time(secons)	0	38	30
Experiment Total Time(hours)	42.19	26.55	18.21
Total Function Cost	0	213.63	166.46
Worker Nodes Cost(VMs)	188.139	0	0
Master Node Cost(VM)	8.43	5.31	3.46
Redis Node Cost(VM)	0	0	3.27
Total Price	196.57	218.94	173.38

RESULTS OBTAINED

Why Do Serverless Prototypes scale transparently?

```
# Start of serverless function code
def serverless_task_process(task,ibm_cos):
  # ProtoMol invocation inside serverless through task object.
  cmd = "./" + task.input_remote_execn_file
  subprocess.call(cmd, shell = True)
# End of serverless function code

#For each Monte Carlo Step
  # For each replica
    task = create_task(replica_id, local_temp_dir, bucket)
    tasks_list.append(task)
  # End For each replica
lithops.map(serverless_task_process, tasks_list)
result_list = lithops.get_result()
# End For Monte Carlo Step
```

How many functions do we launch?

Index	Réplicas	Functions Triggered
1	12	1200
2	24	2400
‘	replicas[index-1]+12	replicas[index]*Monte_Carlo_Steps
‘	‘	‘
‘	‘	‘
12	192	19200

HOW DOES LITHOPS FACILITATE ACCESS TO CLOUD RESOURCES?

Invoking Serverless Functions

```
from lithops.multiprocessing import Pool

pool_client = Pool()
activation_list = pool_client.map(serverless_task_process,task_list_iterdata)
```

Accessing COS

```
from lithops.multiprocessing.cloud_proxy import open as cloud_open

def read_from_remote_storage(filename):
    with cloud_open(filename, 'rb') as f:
        lines = f.readlines()
    return lines
```

Accessing Redis

```
  from lithops.multiprocessing import Manager

  shared_map = Manager().dict()
  res = shared_map['key']
```

MAIN CONCLUSIONS

- *Serverless prototypes reduce the total execution time of the Replica Exchange by around forty percent.*
- *Serverless prototypes scale transparently.*
- *Serverless solution can obtain a comparable or even cheaper cost than a serverful one using Work Queue over VM.*

THANK YOU!

- *Links to our work:*
  - *https://www.serverlesscomputing.org/wosc6/#p10*
  - *https://github.com/faas-prototypes/protomol*
- *Contact:*
  - *mmirabelli@uoc.edu*
  - *pedro.garcia@urv.cat*
  - *gilv@il.ibm.com*