ThereminQ

a Cloud-native Toolbox for Quantum Workloads and Visualisations

Preface

Since the introduction of the current generation of Quantum Systems most vendors deliver a specific toolset both large and small to simulate, operate and work with these new, evolving and more then often data-heavy systems.

For me it quickly became paramount if not mandatory for developers, researchers and regular users of said systems to be able to work with a pre-set and pre-tuned environment. A system that is catered towards diverse scenarios and toolsets yet assure state, behaviour and accelerated adoption to achieve the ambitious business goals set for this emerging industry.

The requirement of Quality, Speed, Collaboration and timely deliverables are elements taken from the IT and manufacturing world. Therefore, elements of Agile, DevOPS and Lean manufacturing are integrated to form the DataOPS-style workflow.

It is that workflow -that way of doing things in ThereminQ- to package, integrate and distribute for Cloud-native consumption in the strange and bold “New World of Quantum”

DataOPS - not just DevOPS for Data

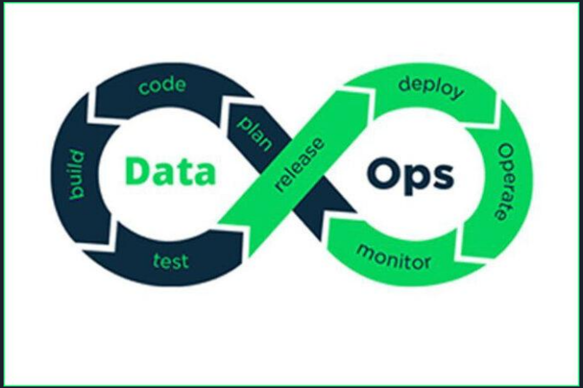
The burden of IT maintenance for DataOPS needs to be virtually non-existent due to the focus on highly specialized tools and services. Leveraging more then a decade of refinements that galvanized the Agile DevOPS revolution with an emphasis to automation and standardisation.

Build / Test

Releases are Build, code Tested

Release / Deploy

Make images available and deploy



Operate / Monitor

As environments are activated output is monitored and stored for future use and analysis.

Plan / Code

New releases and new code iterations are prepared

Introducing: ThereminQ

- Orchestrating Quantum Compute stacks with automated data processing and visualisations
- Enables the same ‘clean room’ operation across different environments and devices
- Created with an emphasis on Open Standards and Protocols
- Scales from an Intuitive Desktop experience to HPC Deployment and vv
- All-in-one packaging, intuitive Dockerfile layout
- More than 3 years of intensive development and testing with the Qrack QC simulator and QC data

“Every Time a Clean Room Experience”

01 Highly automated and guaranteed DTAP, CI/CD and Benchmark iterations through Docker Hub

02 Promoting Trust, Collaboration, Integration and Acceleration

03 Cloud native : same container image can be deployed in very diverse environments

04 No more: “It works on my Quantum Computer” - the same codebase and versioning everywhere

The Toolbox

Container images with workloads for CPU, Desktop GPU and DC MultiGPU Cluster

ThereminQ:QFT (QFT and QFT-Cosmos 14-32 qbits)

ThereminQ:Sycamore (Supremacy 14-32 qubits)

ThereminQ:T-NNd (Deep Tensor Neural Network (25 - 128+ qubits)

ThereminQ:Bonsai Visualisation (QFT, Sycamore, TNNd and Realtime)

ThereminQ:Controller VDI interface to develop, modify and test tasks

ThereminQ-Tensors: the TensorFlow Python-dev VDI interface

Spotlight on Workbench from Datacenter to Workstation

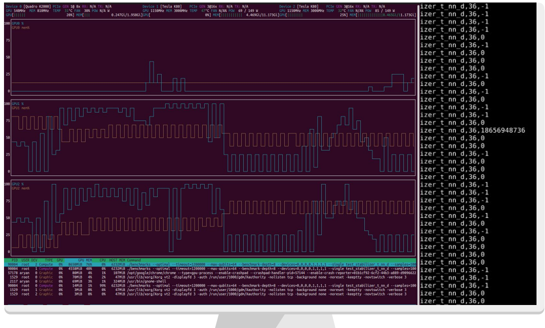
Diverse workloads

Diverse environments

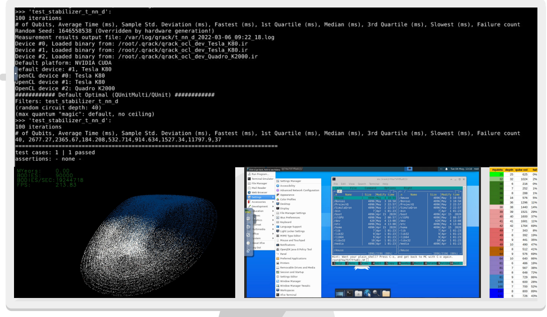
Same configuration and datasets

Always a ‘Clean Room’ experience

All the Things Cloud Native

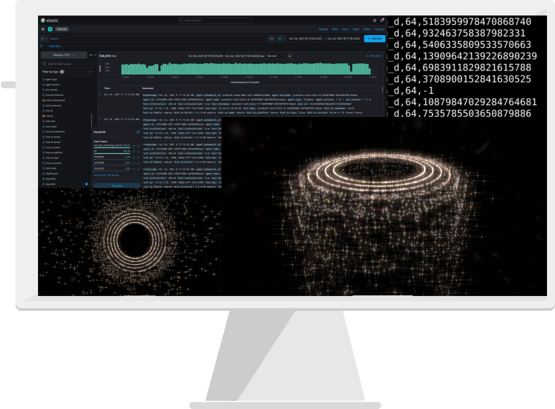


```
izer_t_nn_d,36,-1
izer_t_nn_d,36,-1
izer_t_nn_d,36,-1
izer_t_nn_d,36,-1
izer_t_nn_d,36,0
izer_t_nn_d,36,-1
izer_t_nn_d,36,0
izer_t_nn_d,36,0
izer_t_nn_d,36,0
izer_t_nn_d,36,-1
izer_t_nn_d,36,0
izer_t_nn_d,36,-1
izer_t_nn_d,36,0
izer_t_nn_d,36,-1
izer_t_nn_d,36,-1
izer_t_nn_d,36,0
izer_t_nn_d,36,18656948736
izer_t_nn_d,36,0
izer_t_nn_d,36,0
izer_t_nn_d,36,-1
izer_t_nn_d,36,0
izer_t_nn_d,36,0
izer_t_nn_d,36,0
izer_t_nn_d,36,-1
izer_t_nn_d,36,-1
izer_t_nn_d,36,-1
izer_t_nn_d,36,0
izer_t_nn_d,36,-1
izer_t_nn_d,36,0
izer_t_nn_d,36,-1
izer_t_nn_d,36,0
izer_t_nn_d,36,0
izer_t_nn_d,36,0
izer_t_nn_d,36,0
izer_t_nn_d,36,-1
izer_t_nn_d,36,-1
izer_t_nn_d,36,-1
izer_t_nn_d,36,0
izer_t_nn_d,36,0
```



```
>>> 'test_stabilizer_t_nn_d':
100 iterations
# of Qubits, Average Time (ms), Sample Std. Deviation (ms), Fastest (ms), 1st Quartile (ms), Median (ms), 3rd Quartile (ms), Slowest (ms), Failure count
Random Seed: 1646558538 (Overridden by hardware generation!)
Measurement results output file: /var/log/qrack/t_nn_d_2022-03-06_09:22_18.log
Device #0, Loaded binary from: /root/.qrack/qrack_ocl_dev_Tesla_K80.ir
Device #1, Loaded binary from: /root/.qrack/qrack_ocl_dev_Tesla_K80.ir
Device #2, Loaded binary from: /root/.qrack/qrack_ocl_dev_Quadro_K2000.ir
Default platform: NVIDIA CUDA
efault device: #1, Tesla K80
penCL device #0: Tesla K80
OpenCL device #1: Tesla K80
OpenCL device #2: Quadro K2000
############ Default Optimal (QUnitMulti/QUnit) ############
Filters: test_stabilizer_t_nn_d
(random circuit depth: 40)
(max quantum "magic": default, no ceiling)
>>> 'test_stabilizer_t_nn_d':
100 iterations
# of Qubits, Average Time (ms), Sample Std. Deviation (ms), Fastest (ms), 1st Quartile (ms), Median (ms), 3rd Quartile (ms), Slowest (ms), Failure count
40, 2677.27,2365.67,184.208,532.714,914.634,1527.34,11797.9,37
===============================================================================
test cases: 1 | 1 passed
assertions: - none -
```

```
MYears:     0.00
BODIES:     90000
BODIES/SEC: 19244718
FPS:        213.83
```



```
_d,64,5183959978470868740
_d,64,932463758387982331
_d,64,5406335809533570663
_d,64,13909642139226890239
_d,64,6983911829821615788
_d,64,3708900152841630525
_d,64,-1
_d,64,10879847029284764681
_d,64,7535785503650879886
```

Demo Links

ThereminQ Ecosystem
HPC/QC Workloads
VDI Workbench
Desktop Overlay
Elastic Interconnects

Thank you!

Any questions regarding this presentation and/or demo ?

email ablaauw@gmx.net