RED HAT CLOUD PLATFORMS BUSINESS UNIT

Machine Learning benchmarking with OpenStack and Kubernetes

Choose the right ML infrastructure

Erwan Gallen
Product Manager Cloud Platforms

Red Hat

About your presenter

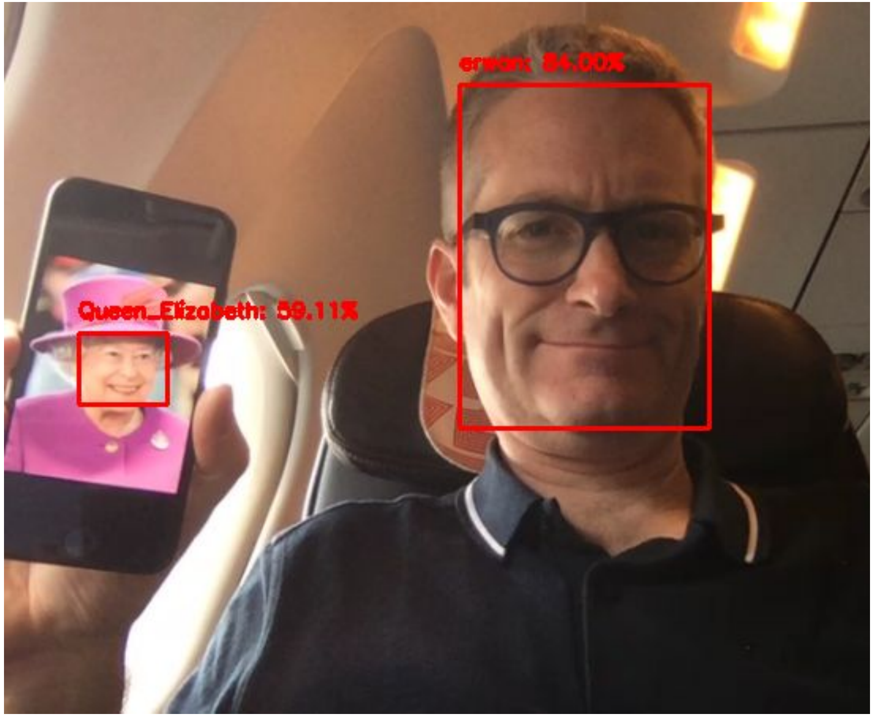
Erwan Gallen
IRC: egallen
Twitter: @egallen
https://egallen.com
https://erwan.com

Product Manager @ Red Hat
Cloud Platforms Business Unit
Hybrid Cloud Computing and AI

Agenda

- Why you need benchmarking for Machine Learning?
- MLPerf, “SPEC for Machine Learning”
- How to benchmark your OpenStack and Kubernetes ML full stack:
    - OpenStack and OpenShift prerequisites
    - Simple TensorFlow Benchmark
    - Thoth knowledge base

Face recognition



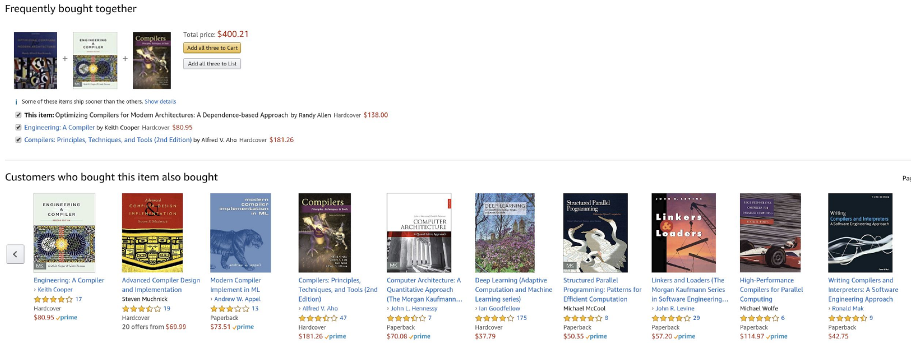
翻译

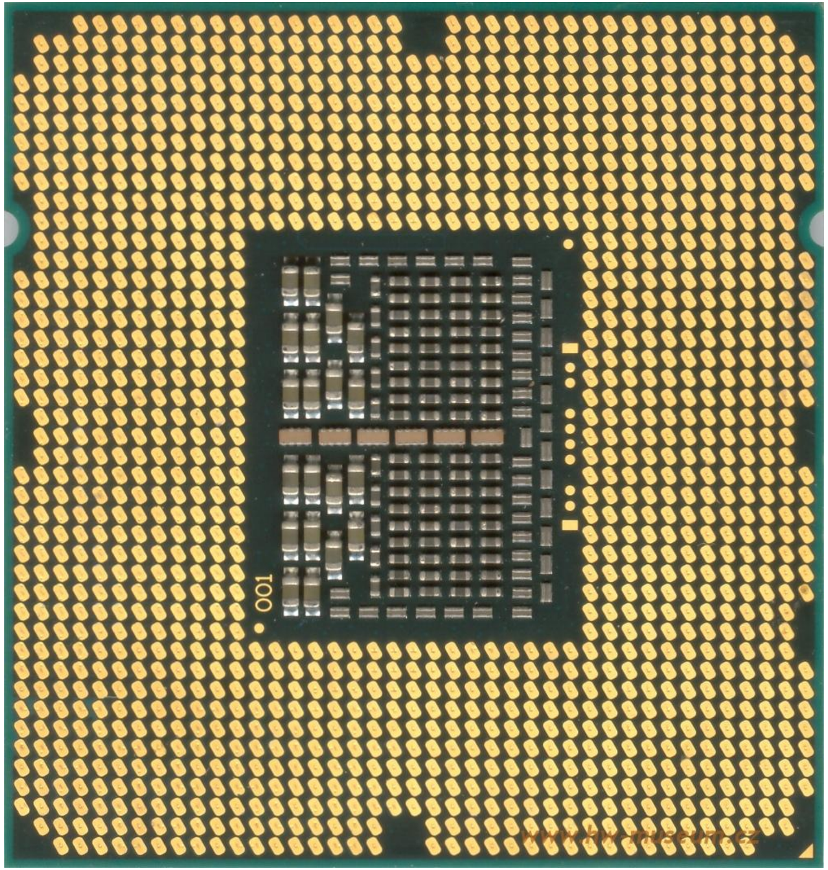
Fraud detection

Red Hat

Self Driving Car

Recommendation engine

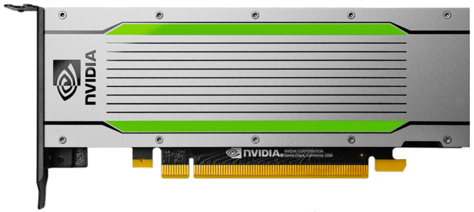




Hardware accelerators for Data Center AI/ML

GPU

NVIDIA Tesla V100
Volta architecture
Memory: 16 GB / 32 GB HBM2
AI/ML inferencing and training



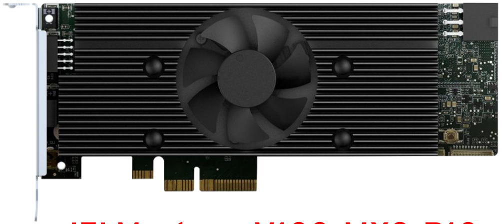
NVIDIA Tesla T4
Turing architecture (low power)
Memory: 16 Go GDDR6
AI/ML inferencing

FPGA

Intel FPGA PAC D5005
Intel Intel Stratix 10
Memory: 32 GB DDR4
AI/ML inferencing

Xilinx Alveo U50 DC Accelerator
UltraScale+ XCU50 (low power)
Memory: 8 GB HBM2
AI/ML inferencing

VPU



IEI Mustang-V100-MX8-R10
VPU Intel Myriad X (x8)
AI/ML inferencing

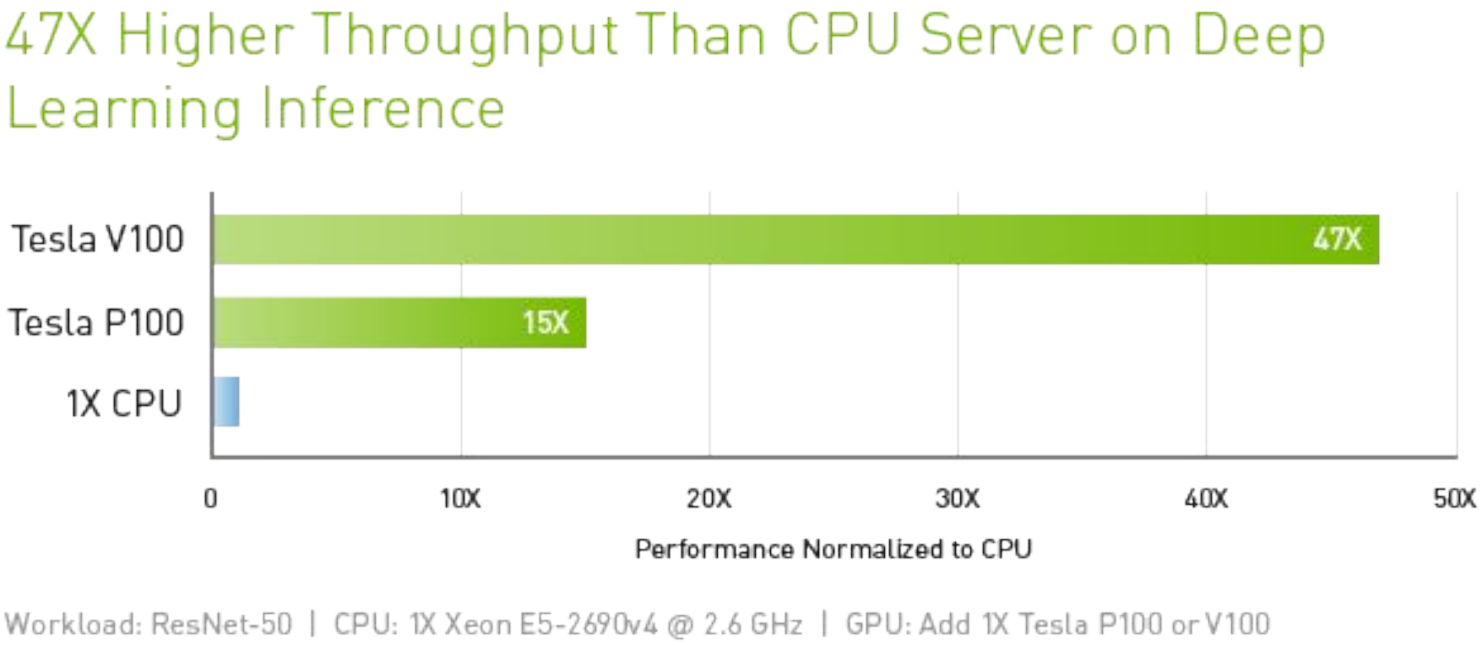
Historical break: explosion of software and hardware solutions

Red Hat

NVIDIA is leading Deep Learning computing

- CUDA cores
- Tensor Cores (Mixed precision Matrix math support)
- Access via frameworks and libraries (cuDNN, cuBlas, TensorRT) and C++
- NVLink/NVSwitch:
  - High speed connecting between GPUs for distributed algorithms
- Integrated Software Stack:
  - Driver: hardware certification, pre-built packages, and testing
  - Platform integration: OpenStack + vComputeServer, OpenShift + NVIDIA k8s-device-plugin

GPU versus CPU performance



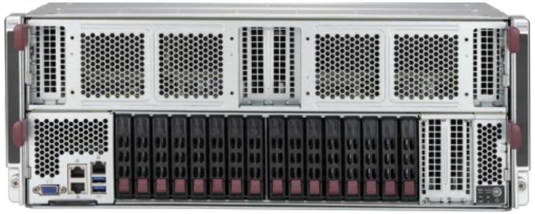
GPU accelerated servers

NVIDIA DGX-2 (16 x **V100** + NVSwitch)

NVIDIA DGX-1 (8 x **V100** + NVLink)

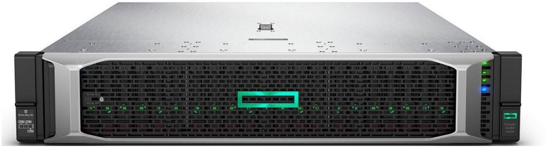
Dell EMC PowerEdge R940xa (8 x **V100**)

Dell EMC PowerEdge R740xd (3 x **V100**)

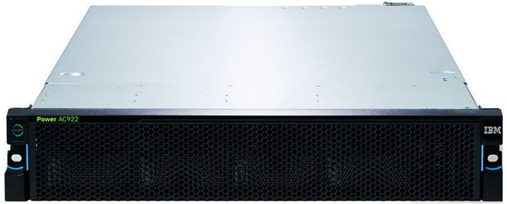


Supermicro SYS-4029GP-TVRT (8 x **V100**)

HPE Apollo 6500 Gen10 (8 x **V100**)



HPE ProLiant DL380 Gen10 (3 x **V100**)

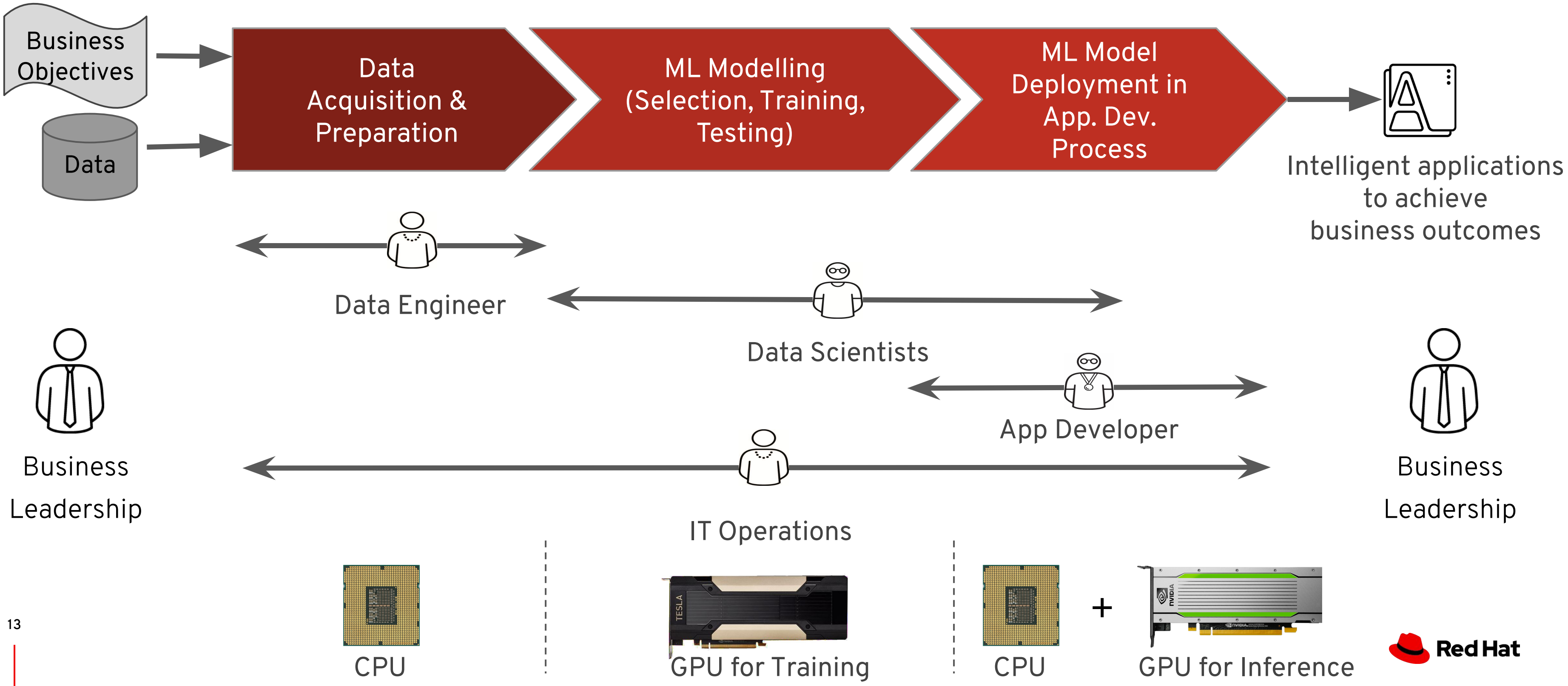


IBM Power System AC922 (6 x **V100**)

NVIDIA Tesla Qualified servers: https://www.nvidia.com/en-us/data-center/tesla/tesla-qualified-servers-catalog/

Machine Learning Benchmarking

Machine Learning Pipeline & Key Personas



Machine Learning benchmarking

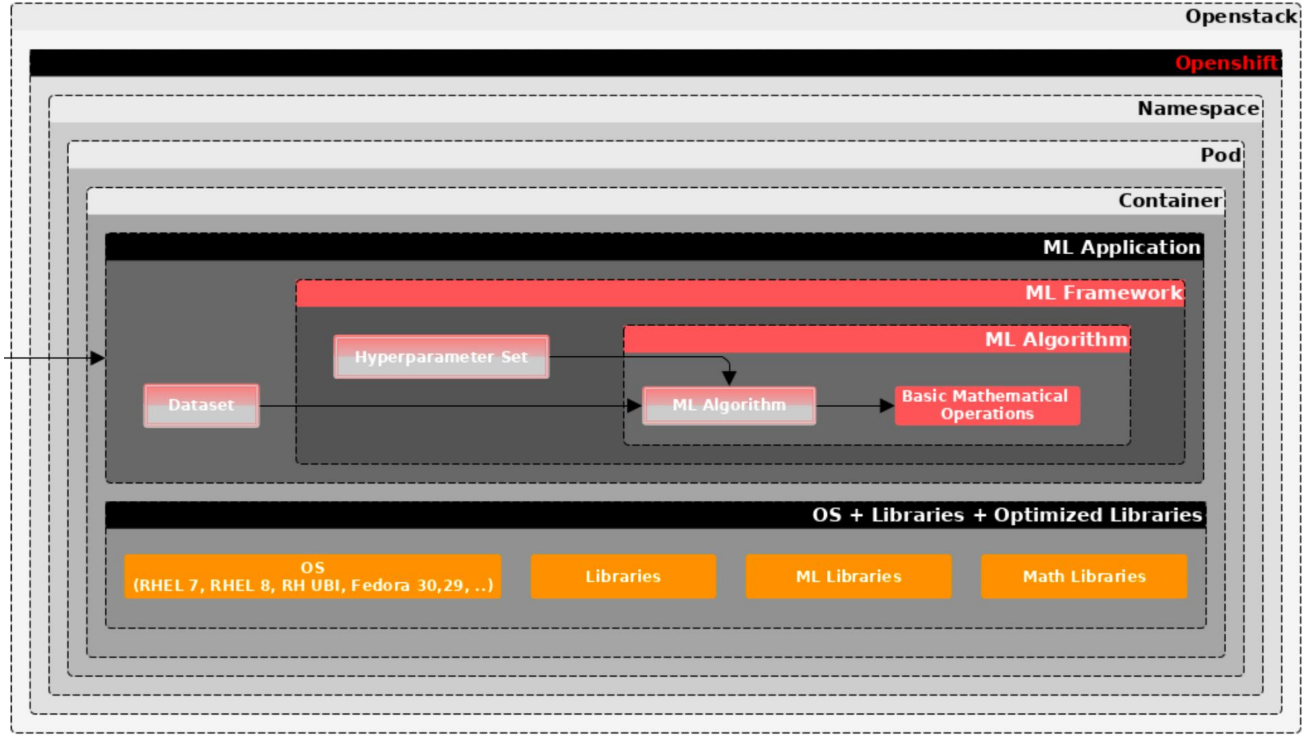
Machine learning training presents a number of unique challenges to benchmark:

- Some optimizations that improve training throughput actually increase time to solution
- Time to solution has high variance
- The software and hardware systems are so diverse that they cannot be fairly benchmarked with the same binary, code, or even hyperparameters.

Needs industry-standard performance benchmarks to drive design and enable competitive evaluation.

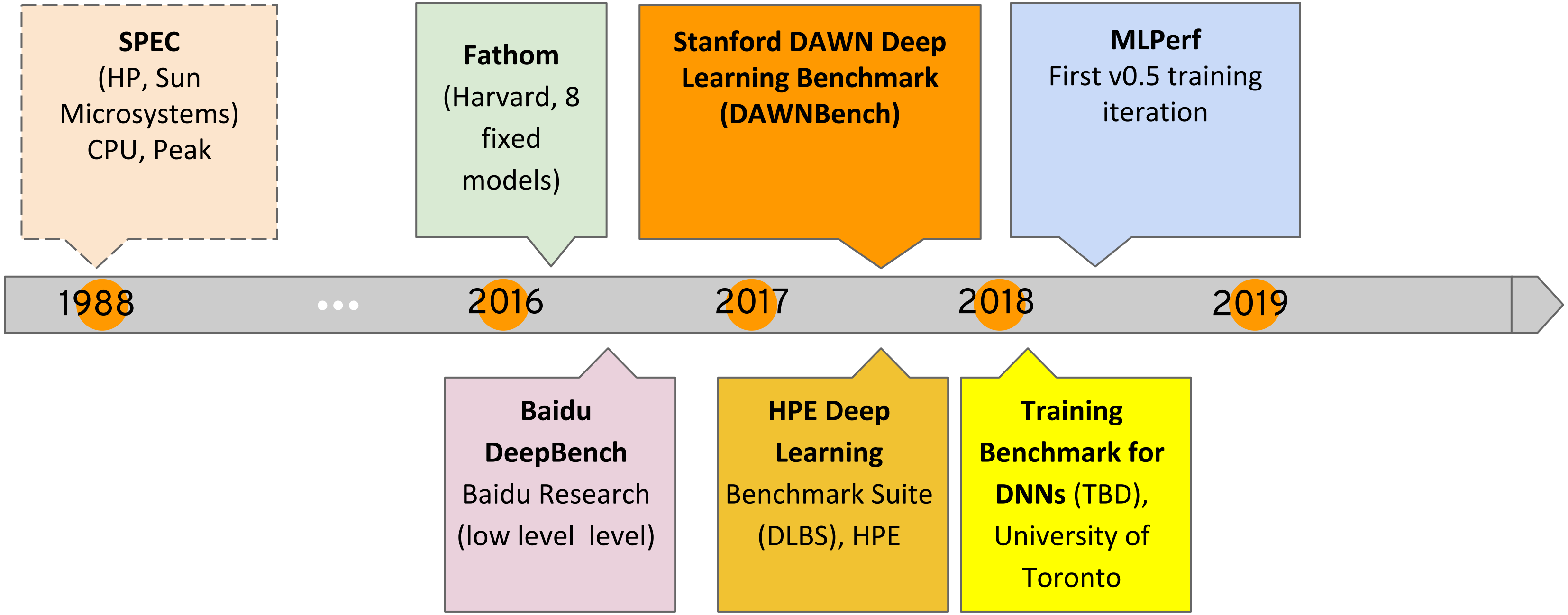
Source: Peter Mattson, arXiv:1910.01500v2 [cs.LG] 30 Oct 2019

Performance of the full Machine Learning stack



CPU	GPU	FPGA

Deep Learning Benchmark history



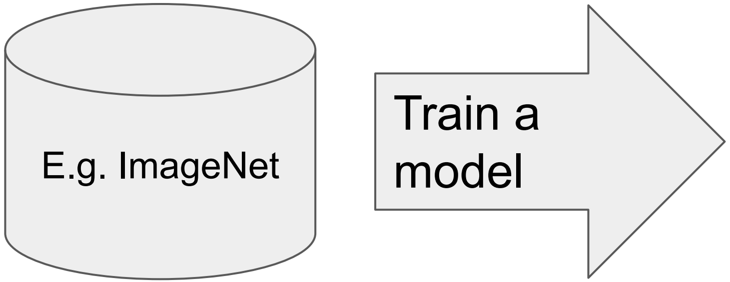
Source: https://www.anandtech.com/show/12673/titan-v-deep-learning-deep-dive/5

MLPerf

MLPerf training, do we specify the model?

Dataset

Target Quality



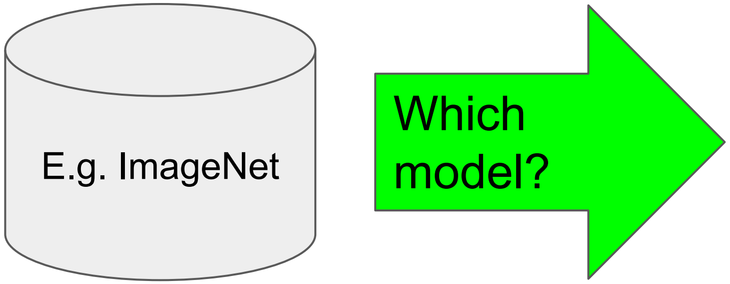
E.g. 75.9%

The goal of training in machine learning is to create a model that generalizes well to unseen data according to a given quality metric (e.g., accuracy).

MLPerf training, do we specify the model?

Dataset

Target Quality



E.g. 75.9%

Choice: two divisions for Training

- Closed division:
  - Model is specified
  - Fixed model parameters
  - Fixed data format
- Open division:
  - Model is not specified
  - Encourage innovations
  - Tricks and model adjustement welcomed

High Level: MLPerf

General MLPerf goals since 2018:

- Accelerate progress in ML via fair and useful measurement
- Serve both the commercial and research communities
- Enable fair comparison of competing systems yet encourage innovation to improve the state-of-the-art of ML
- Enforce replicability to ensure reliable results
- Keep benchmarking effort affordable so all can participate

HARVARD UNIVERSITY
Baidu 百度
Google
Berkeley UNIVERSITY OF CALIFORNIA

MLPerf Training

MLPerf training benchmark suite measures how fast a system can train ML models.

V0.6 results published 2019, July 10th

MLPerf Inference

MLPerf inference benchmark measures how fast a system can perform ML inference using a trained model.

V0.5 coming soon: 2019 mid November

Red Hat

High Level: MLPerf

Name: MLPerf
Founders: collaboration of companies and researchers from educational institutions.
Created: February 2018
Version: 0.6.0
Goal:

Measure system performance for both training and inference from mobile devices to cloud services. MLPerf can help people choose the right ML infrastructure for their applications

Metrics:

- **wall clock time** to train a model to a target quality (based on original publication result, less a small delta to allow for run-to-run variance);
- **power (a useful proxy for cost)**
- **cloud cost**

Schedule of submission rounds

Past and future submission schedule:

Submission round	Submission date	Results public
Training v0.5		2018, December 12nd
Training v0.6		2019, July 10th
Inference v0.5	2019, October 11st	2019, November 6th
Training v0.7	2020, February 21st [tentative]	
Inference v0.7 (~~v0.6~~)	2020, May [tentative]	
Training v0.8	2020, August [tentative]	
Inference v0.8	2020, November [tentative]	

Source: https://github.com/mlperf/policies/blob/master/submission_rules.adoc

MLPerf governance

The MLCommons mission is to **accelerate ML innovation** and increase its positive impact on society by creating public resources and supporting outreach activities.

More than 40 companies and 800 members involved.

Plan to create an **MLCommons** Foundation to host MLPerf

- Zurich foundation
- Target launch in February 2020
- Membership will be required for many MLPerf activities
- Become a founding member now and help set the direction

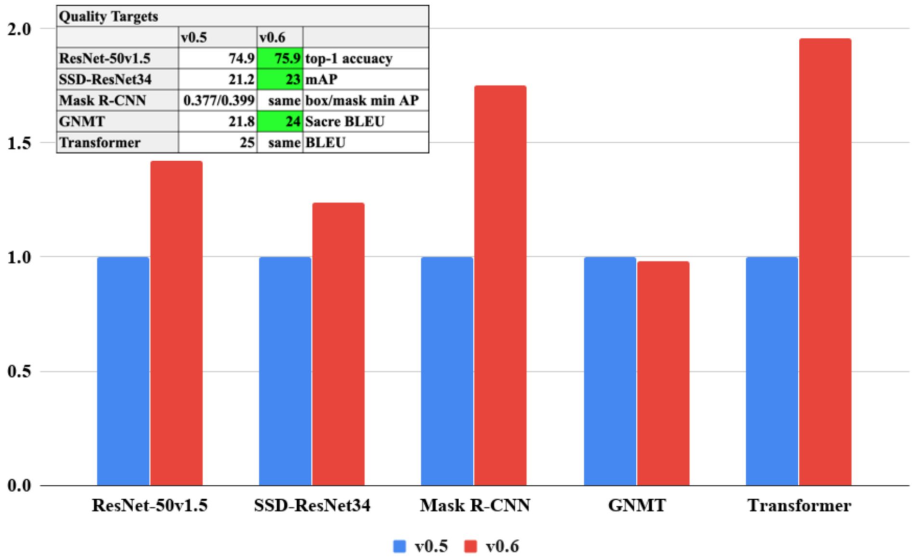
MLPerf choices for v0.6

- Mix of importance, availability of data, and readiness of code.
- Cutting but not bleeding edge models.
- Compare to v0.5, quality targets raised

Area	Problem	Dataset	Model
Vision	Image recognition	ImageNet	ResNet-50
	Object detection, light-weight	COCO	SSD w/Resnet34
	Object detection, heavy-weight	COCO	Mask R CNN
Language	Translation	WMT Eng.-German	NMT
	Translation	WMT Eng.-German	Transformer
Commerce	Recommendation	Movielens-20M	NCF
Reinforcement Learning	Go	Pro games	Mini go

Agile benchmark development

- Rapidly iterate the benchmark suite:
  - Remain relevant in the very fast moving ML field
  - Correct inevitable mistakes in the formulation
  - Scale problems to match faster hardware
- Like SPEC, have quarterly deadlines and then publish results for that quarter via searchable database



From MLPerf Training v0.5 to v0.6, Quality targets raised:
Image classification (ResNet-50) to 75.9%
Single Shot Detector (light-weight Object Detection) to 23%
Google Neural Machine Translation (GNMT) to 24 Sacre BLEU.

MLPerf Training v0.6 Results

https://mlperf.org/training-results-0-6

MLPerf Training v0.6 Results

July 10th, 2019

Any use of the MLPerf results and site must comply with the MLPerf Terms of Use.

You may wish to read the Training Overview to better understand the results.

To see the earlier MLPerf Training v0.5 results go here.

- Closed Division Times
- Open Division Times

Closed Division Times

								Benchmark results (minutes)									
								Image classification	Object detection, light-weight	Object detection, heavy-wt.	Translation , recurrent	Translation , non-recur.	Recom-mendation	Reinforce-ment Learning			
								ImageNet	COCO	COCO	WMT E-G	WMT E-G	MovieLens-20M	Go			
#	**Submitter**	**System**	**Processor**	**#**	**Accelerator**	**#**	**Software**	ResNet-50 v1.5	SSD w/ ResNet-34	Mask-R-CNN	NMT	Transformer	NCF	Mini Go	**Details**	**Code**	**Notes**
Available in cloud																	
0.6-1	Google	TPUv3.32			TPUv3	16	TensorFlow, TPU 1.14.1.dev	42.19	12.61	107.03	12.25	10.20	[1]		details	code	none
0.6-2	Google	TPUv3.128			TPUv3	64	TensorFlow, TPU 1.14.1.dev	11.22	3.89	57.46	4.62	3.85	[1]		details	code	none
0.6-3	Google	TPUv3.256			TPUv3	128	TensorFlow, TPU 1.14.1.dev	6.86	2.76	35.60	3.53	2.81	[1]		details	code	none
0.6-4	Google	TPUv3.512			TPUv3	256	TensorFlow, TPU 1.14.1.dev	3.85	1.79		2.51	1.58	[1]		details	code	none
0.6-5	Google	TPUv3.1024			TPUv3	512	TensorFlow, TPU 1.14.1.dev	2.27	1.34		2.11	1.05	[1]		details	code	none
0.6-6	Google	TPUv3.2048			TPUv3	1024	TensorFlow, TPU 1.14.1.dev	1.28	1.21			0.85	[1]		details	code	none
Available on-premise																	
0.6-7	Intel	32x 2S CLX 8260L	CLX 8260L	64			TensorFlow						[1]	14.43	details	code	none
0.6-8	NVIDIA	DGX-1			Tesla V100	8	MXNet, NGC19.05	115.22					[1]		details	code	none
0.6-9	NVIDIA	DGX-1			Tesla V100	8	PyTorch, NGC19.05		22.36	207.48	20.55	20.34	[1]		details	code	none
0.6-10	NVIDIA	DGX-1			Tesla V100	8	TensorFlow, NGC19.05						[1]	27.39	details	code	none
0.6-11	NVIDIA	3x DGX-1			Tesla V100	24	TensorFlow, NGC19.05						[1]	13.57	details	code	none
0.6-12	NVIDIA	24x DGX-1			Tesla V100	192	PyTorch, NGC19.05			22.03			[1]		details	code	none
0.6-13	NVIDIA	30x DGX-1			Tesla V100	240	PyTorch, NGC19.05		2.67				[1]		details	code	none
0.6-14	NVIDIA	48x DGX-1			Tesla V100	384	PyTorch, NGC19.05				1.99		[1]		details	code	none
0.6-15	NVIDIA	60x DGX-1			Tesla V100	480	PyTorch, NGC19.05					2.05	[1]		details	code	none
0.6-16	NVIDIA	130x DGX-1			Tesla V100	1040	MXNet, NGC19.05	1.69					[1]		details	code	none
0.6-17	NVIDIA	DGX-2			Tesla V100	16	MXNet, NGC19.05	57.87					[1]		details	code	none
0.6-18	NVIDIA	DGX-2			Tesla V100	16	PyTorch, NGC19.05		12.21	101.00	10.94	11.04	[1]		details	code	none
0.6-19	NVIDIA	DGX-2H			Tesla V100	16	MXNet, NGC19.05	52.74					[1]		details	code	none
0.6-20	NVIDIA	DGX-2H			Tesla V100	16	PyTorch, NGC19.05		11.41	95.20	9.87	9.80	[1]		details	code	none
0.6-21	NVIDIA	4x DGX-2H			Tesla V100	64	PyTorch, NGC19.05		4.78	32.72			[1]		details	code	none

Source: https://mlperf.org/training-results-0-6

MLPerf records at Scale and per accelerator

Record Type	Benchmark	Record
Max Scale (minutes To Train)	Object Detection (Heavy Weight) - Mask R-CNN	18.47 Mins
	Translation (Recurrent) - GNMT	1.8 Mins
	Reinforcement Learning - MiniGo	13.57 Mins
Per Accelerator (hours To Train)	Object Detection (Heavy Weight) - Mask R-CNN	25.39 Hrs
	Object Detection (Light Weight) - SSD	3.04 Hrs
	Translation (Recurrent) - GNMT	2.63 Hrs
	Translation (Non-recurrent) - Transformer	2.61 Hrs
	Reinforcement Learning - MiniGo	3.65 Hrs

Per Accelerator comparison using reported performance for MLPerf 0.6 NVIDIA DGX-2H (16 V100s) compared to other submissions at same scale except for MiniGo where NVIDIA DGX-1 (8 V100s) submission was used| MLPerf ID Max Scale: Mask R-CNN: 0.6-23, GNMT: 0.6-26, MiniGo: 0.6-11 | MLPerf ID Per Accelerator: Mask R-CNN, SSD, GNMT, Transformer: all use 0.6-20, MiniGo: 0.6-10

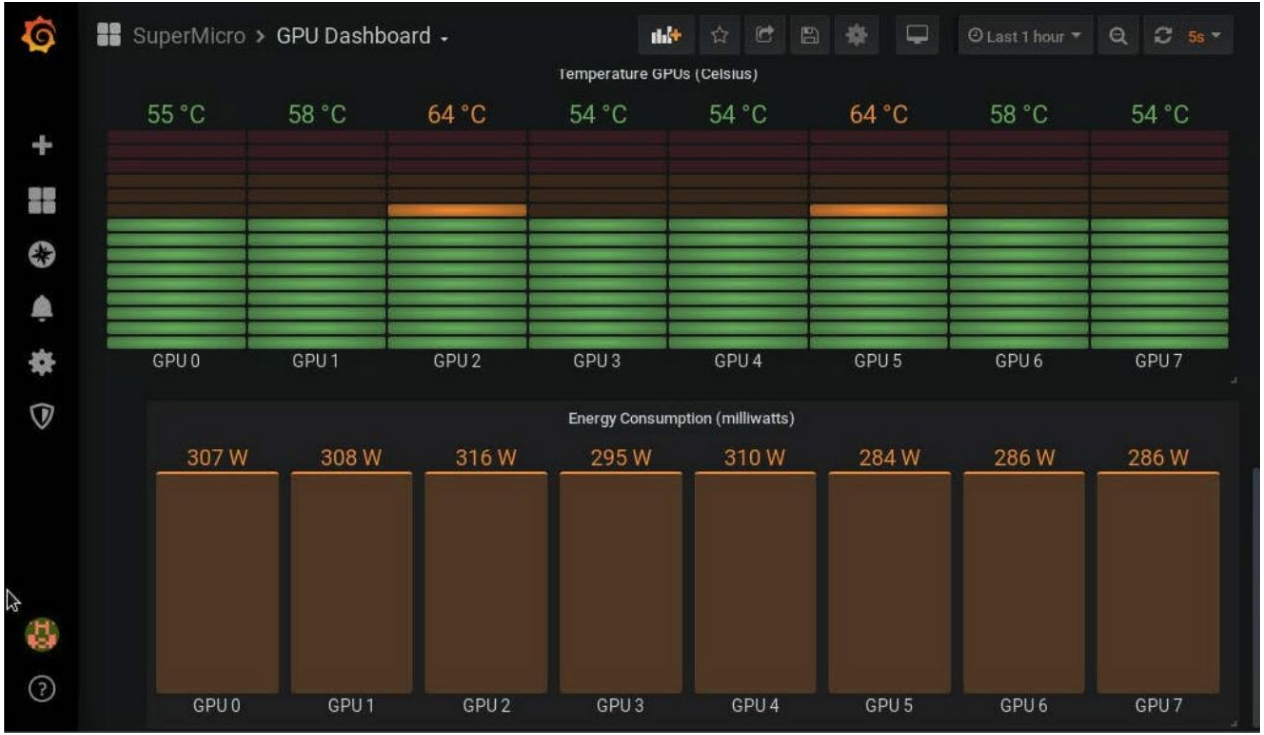
The Open Data Hub Project

- Open community: https://opendatahub.io
- AI end-to-end platform, Meta-Project that integrates best of open source AI projects
- Reference Architecture for OpenShift
- Red Hat's internal Data Science and AI platform
- OpenShift 3.11 or 4+, based on operator
- GPU performance benchmarks with MLPerf

Red Cloud Platform products provide value across the ML pipeline

Red Hat OpenShift | Red Hat OpenStack Platform | Red Hat Virtualization | Red Hat Ceph Storage

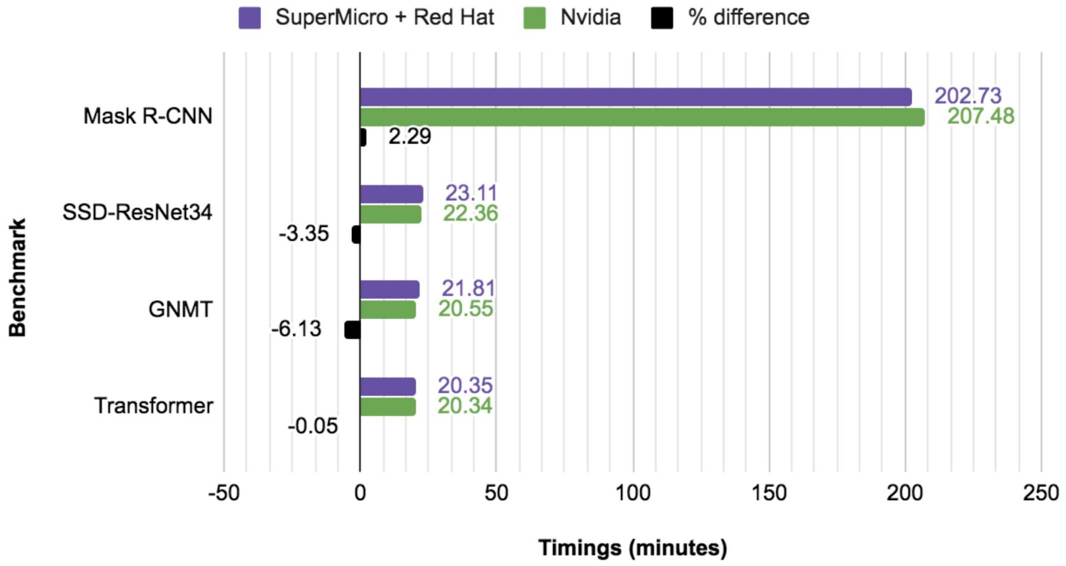
Red Hat and SuperMicro MLperf Training v0.6



Supermicro GPU Server
SYS-4029GP-TVRT
8 x Tesla V100 / server

Source Red Hat: https://www.redhat.com/en/resources/supermicro-deep-learning-openshift-reference-architecture (results not verified by MLPerf Consortium).
Diane Feddema, Mehnaz Mahbub, Reeann Zhang, David Gray, Mayur Shetty, Kevin Yu, George Chang, Jason Zhu

Red Hat and SuperMicro MLperf Training v0.6



Benchmark	Object Detection (heavy weight)	Object Detection (light weight)	Translation (Recurrent)	Translation (Non-Recurrent)
Data	COCO 2017	COCO 2017	WMT English-German	WMT English-German
Data Size	21GB	21GB	56GB	56GB
Model	Mask R-CNN	SSD-ResNet34	GNMT	Transformer
Framework	PyTorch	PyTorch	PyTorch	PyTorch

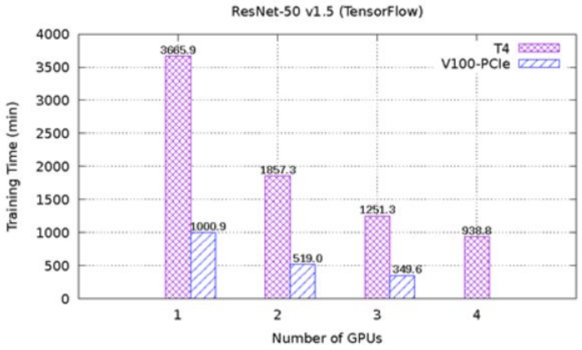
Benchmark results showing that MLPerf v0.6 on OpenShift was faster than the NVIDIA published timing for Mask R-CNN and only .05 to 6.13% slower for SDD-ResNet34, GMNT and Transformer.

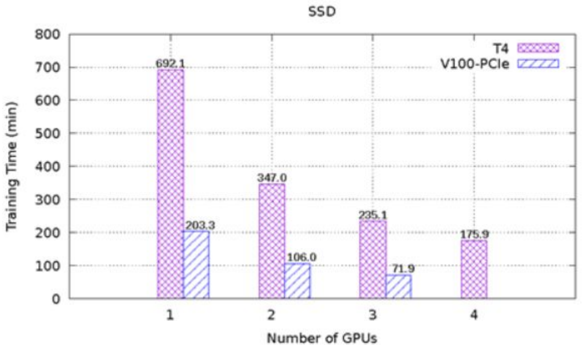
Source Red Hat: https://www.redhat.com/en/resources/supermicro-deep-learning-openshift-reference-architecture (results not verified by MLPerf Consortium).
Diane Feddema, Mehnaz Mahbub, Reeann Zhang, David Gray, Mayur Shetty, Kevin Yu, George Chang, Jason Zhu

Red Hat

Dell MLPerf NVIDIA V100 with NVIDIA T4

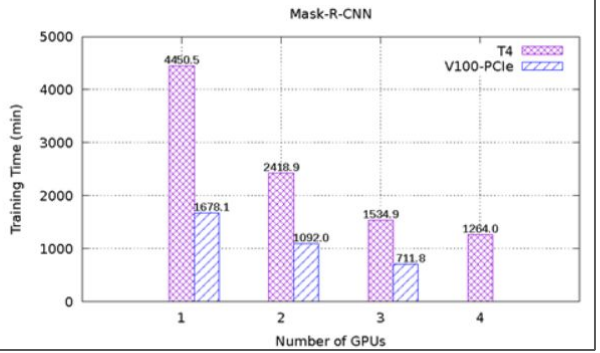
2.5 days

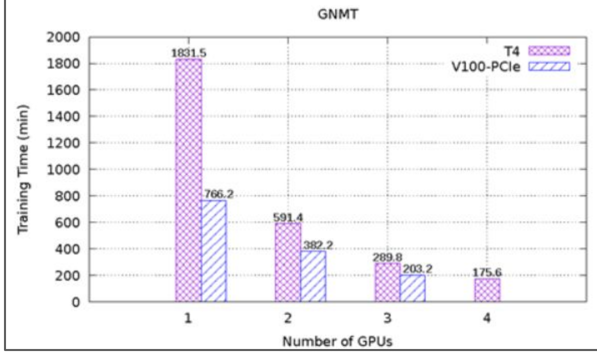




11.5 hours

3 days



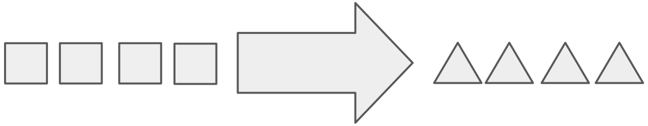


30 hours

Source Dell: https://www.dell.com/support/article/fr/fr/frbsdt1/sln316560/deep-learning-performance-on-t4-gpus-with-mlperf-benchmarks?lang=en

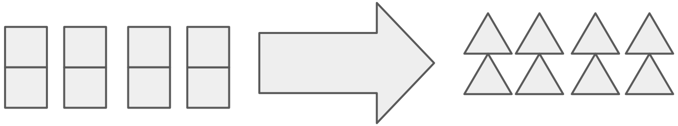
Red Hat

Inference metric: one metric for each scenario



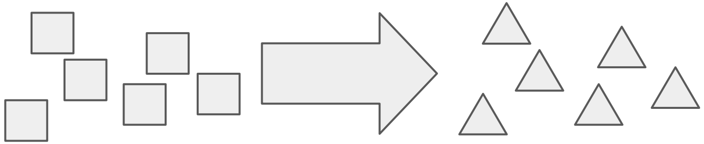
Single stream
e.g. cell phone augmented vision

Latency



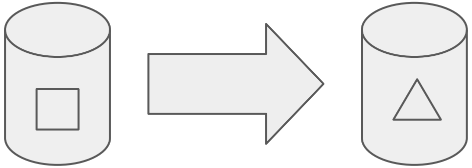
Multiple stream
e.g. multiple camera driving assistance

Number streams
subject to latency bound



Server
e.g. translation site

QPS
subject to latency bound



Offline
e.g. photo sorting

Throughput

Inference scenarios

Scenario	Query generation	Inferences per query	Latency constraint (ms)	Tail latency	Metric
Single stream	The LoadGen sends the next query as soon as the SUT completes the previous one	1	None	90%	90th percentile measured latency
Multiple stream	The LoadGen sends a new query every Latency Constraint, if the SUT has completed the prior query. Otherwise, the new query is dropped. Such an event is one overtime query.	Variable, see metric	Benchmark specific based on typical use	90%	Maximum number of inferences per query supported
Server	The LoadGen sends new queries to the SUT according to a Poisson distribution.	1	Benchmark specific based on typical use	90%	Maximum Poisson throughput parameter supported
Offline	The LoadGen sends all queries to the SUT at one time.	All	None	N/A	Measured throughput

Red Hat

Inference Models v0.5

Area	Task	Model	Dataset
Vision	Image classification	Resnet50-v1.5	ImageNet (224x224)
Vision	Image classification	MobileNets-v1 224	ImageNet (224x224)
Vision	Object detection	SSD-ResNet34	COCO (1200x1200)
Vision	Object detection	SSD-MobileNets-v1	COCO (300x300)
Language	Machine translation	GNMT	WMT16

Inference submitters

- Alibaba
- AMD
- Centaur
- Dell
- dividiti
- Facebook
- FCCL-FAI
- FuriosaAI
- Google
- Habana
- Hailo
- Inspur
- Intel
- MediaTek
- Microsoft
- ModelScope
- Nvidia
- PQLabs
- Qualcomm
- Samsung
- SuperMicro
- Tencent
- Xilinx

Running the inference reference benchmark app

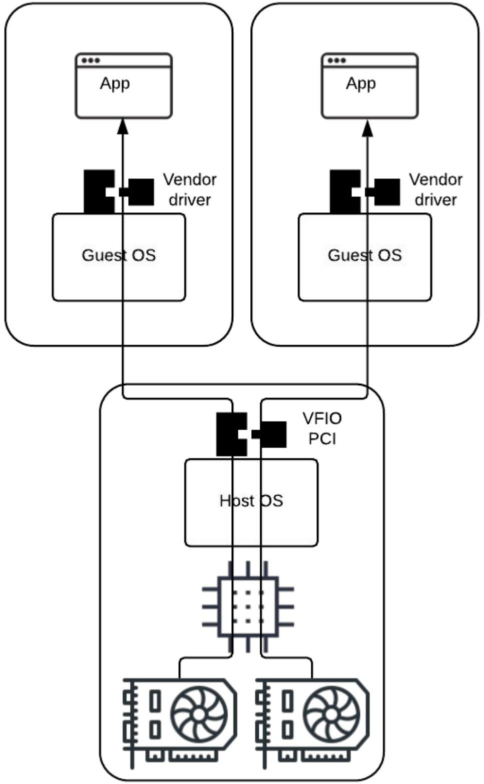
```
$ ./run_local.sh resnet50 gpu
…
TestScenario.SingleStream qps=163.51, mean=0.0061, time=60.040, queries=9817,
tiles=50.0:0.0059,80.0:0.0063,90.0:0.0066,95.0:0.0070,99.0:0.0083,99.9:0.0108
```

```
$ ./run_local.sh resnet50 cpu
…
TestScenario.SingleStream qps=10.18, mean=0.0981, time=100.568, queries=1024,
tiles=50.0:0.0961,80.0:0.1045,90.0:0.1076,95.0:0.1114,99.0:0.1275,99.9:0.1395
```

```
$ ./run_local.sh mobilenet cpu
…
Accuracy qps=48.12, mean=0.019353,  acc=87.50, queries=8,
t=80:0.0198,90:0.0278,95:0.0366,99:0.0436,99.9:0.0451
INFO:main:starting TestScenario.SingleStream
TestScenario.SingleStream qps=67.94, mean=0.014653, queries=683,
t=80:0.0154,90:0.0173,95:0.0191,99:0.0256,99.9:0.0627
```

OpenStack and Kubernetes prerequisites

Exposing GPUs to virtual machines with PCI Passthrough



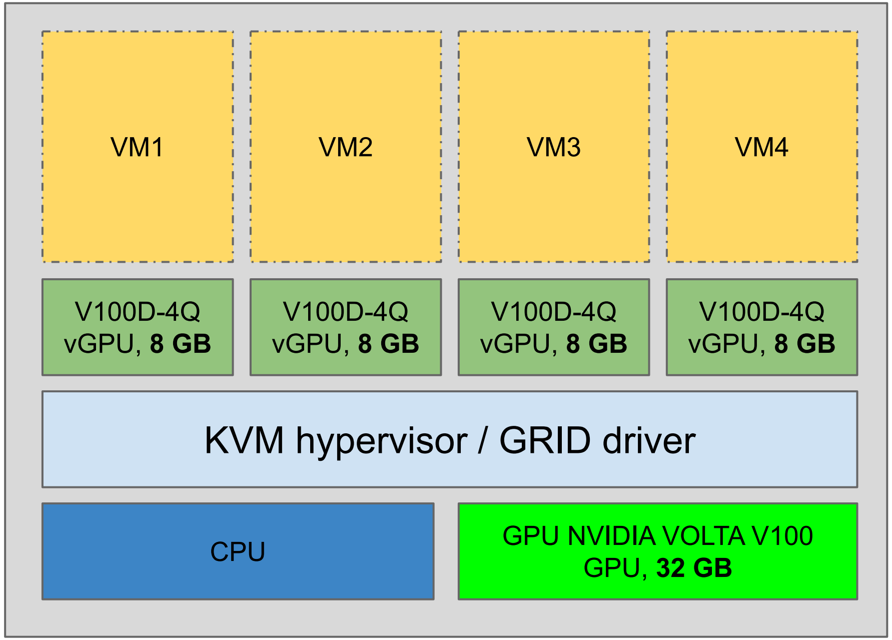
- 1-1 MAPPING OF HOST DEVICE TO GUEST
- IMPLEMENTED IN QEMU AS HOST DEVICE

Implemented in upstream OpenStack since Havana
Supported by Red Hat OpenStack Platform

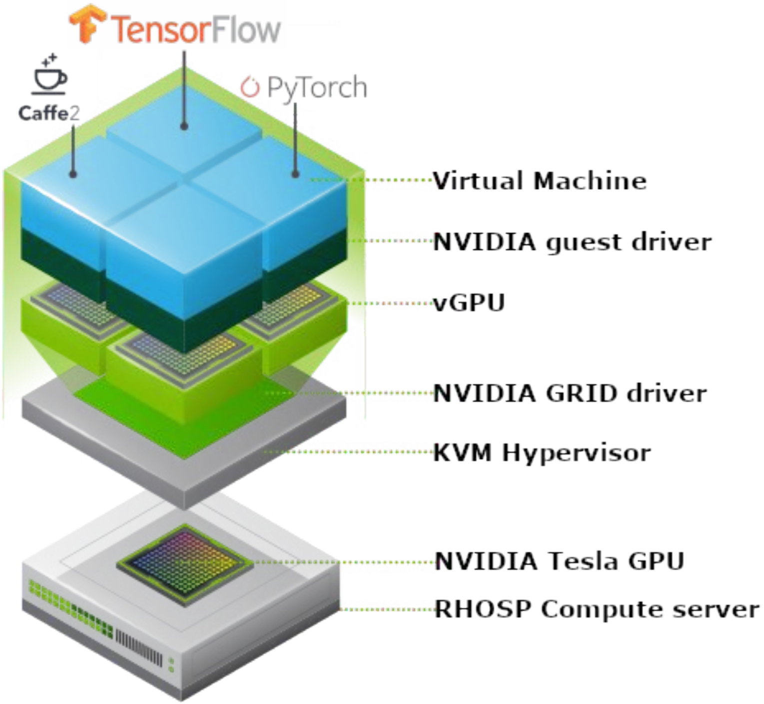
PROS:
- Full compatibility on the guest
- Maximum performance on the guest

CAVEATS:
- Device exposure to the guest
- PCI-E lanes limitations per CPU
- Capacity management challenges

NVIDIA vGPU with GRID driver



OpenStack Compute node



Source: NVIDIA software documentation

Enable GPUs with OpenShift

Modeling libs and frameworks

CUDA Toolkit

Universal Base Image

CRIO
(+ GPU runtime plugin)

CUDA driver for GPUs

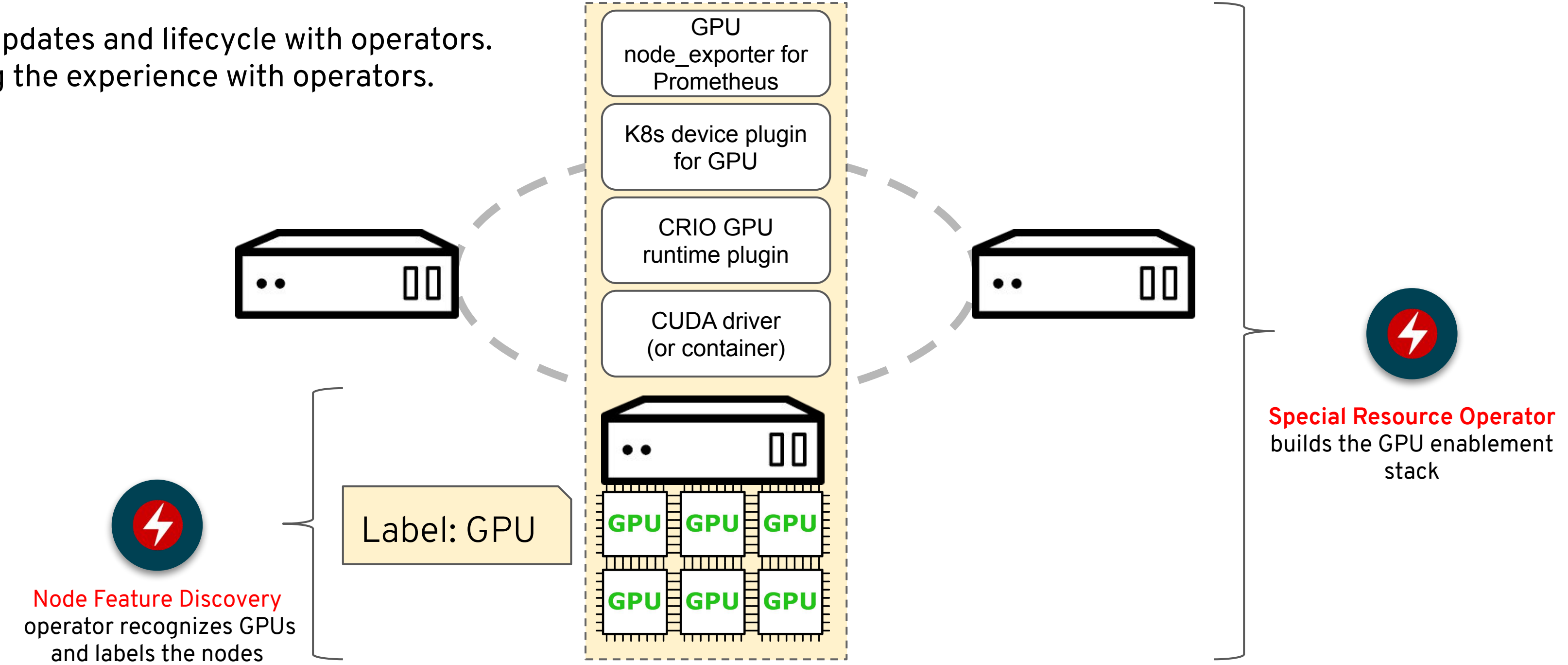
RHEL or RHELCoreOS

Machine Learning container

Enabled and supported in OCP 3.11 and OpenShift 4.2 (with BYO RHEL7 Workers)

Enable GPUs with OpenShift

Managing updates and lifecycle with operators.
Automating the experience with operators.



GPU supported in OpenShift 3.11 and OpenShift 4.2 with RHEL7 only on GPU nodes; NFD and GPU operator are in roadmap

OpenShift on OpenStack

OpenShift

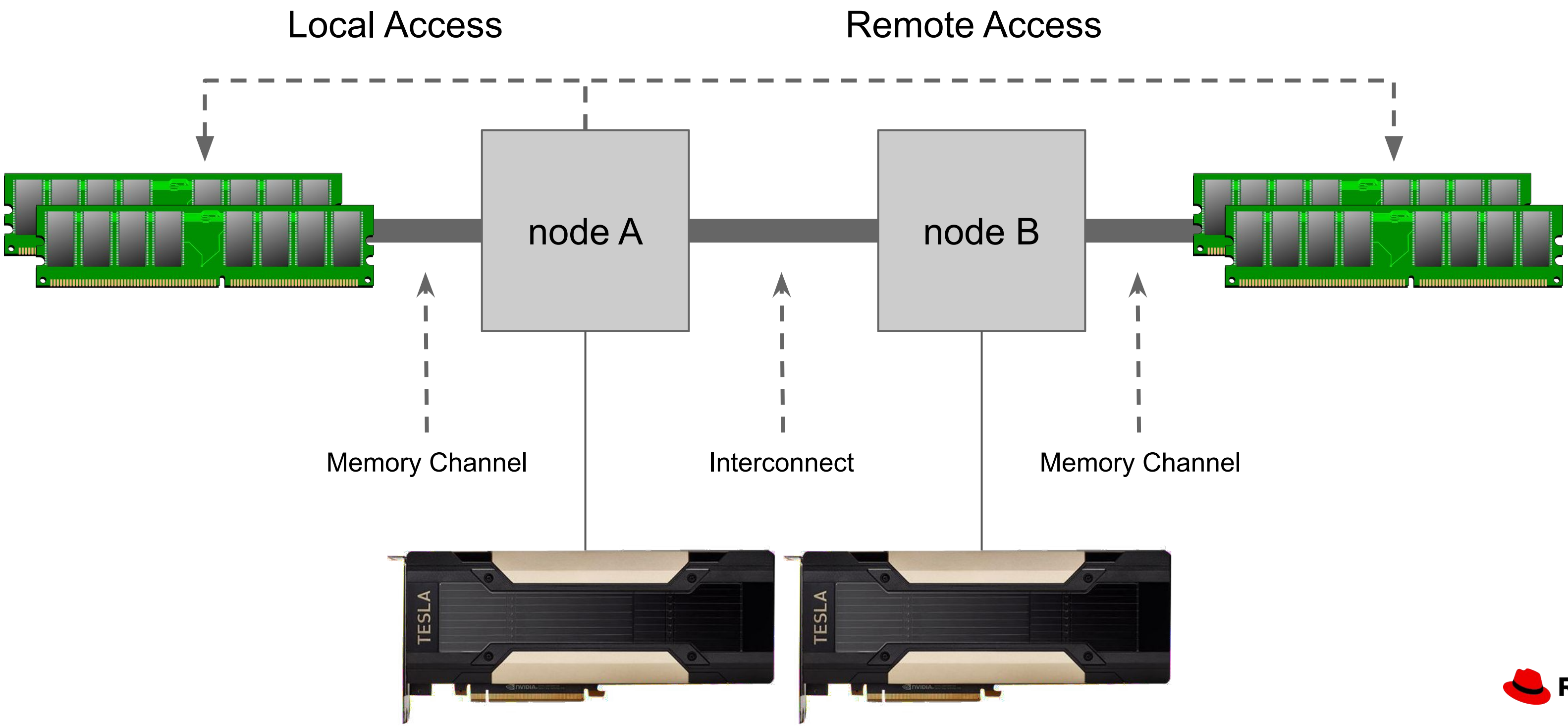
```
(overcloud) [stack@perflab-director ~]$ oc get nodes
NAME                          STATUS   ROLES    AGE     VERSION
perflab-x7szb-master-0        Ready    master   8d      v1.14.6+c07e432da
perflab-x7szb-master-1        Ready    master   8d      v1.14.6+c07e432da
perflab-x7szb-master-2        Ready    master   8d      v1.14.6+c07e432da
perflab-x7szb-worker-2jqns    Ready    worker   8d      v1.14.6+c07e432da
perflab-x7szb-worker-7gk2p    Ready    worker   8d      v1.14.6+c07e432da
perflab-x7szb-worker-gpu-rrstz Ready   worker   6d14h   v1.14.6+c07e432da
perflab-x7szb-worker-v6xwp    Ready    worker   8d      v1.14.6+c07e432da
```

OpenStack

```
(overcloud) [stack@perflab-director ~]$ openstack server list -c Name
-c Status -c Image -c Flavor
+--------------------------------+--------+-------+--------------+
| Name                           | Status | Image | Flavor       |
+--------------------------------+--------+-------+--------------+
| perflab-x7szb-worker-gpu-rrstz | ACTIVE | rhcos | m1-gpu.large |
| perflab-x7szb-worker-2jqns     | ACTIVE | rhcos | m1.large     |
| perflab-x7szb-worker-7gk2p     | ACTIVE | rhcos | m1.large     |
| perflab-x7szb-worker-v6xwp     | ACTIVE | rhcos | m1.large     |
| perflab-x7szb-master-0         | ACTIVE | rhcos | m1.large     |
| perflab-x7szb-master-2         | ACTIVE | rhcos | m1.large     |
| perflab-x7szb-master-1         | ACTIVE | rhcos | m1.large     |
+--------------------------------+--------+-------+--------------+
```


Red Hat

Take care of NUMA affinity



TensorFlow Benchmark

TensorFlow Benchmark

https://github.com/tensorflow/benchmarks

```
$ cat << EOF > tensorflow-benchmarks-gpu.yaml
apiVersion: v1
kind: Pod
metadata:
 name: tensorflow-benchmarks-gpu
spec:
 containers:
 - image: nvcr.io/nvidia/tensorflow:19.09-py3
   name: cudnn
   command: ["/bin/sh","-c"]
   args: ["git clone https://github.com/tensorflow/benchmarks.git;cd
benchmarks/scripts/tf_cnn_benchmarks;python3 tf_cnn_benchmarks.py
--num_gpus=1 --data_format=NHWC --batch_size=32 --model=resnet50
--variable_update=parameter_server"]
   resources:
    limits:
      nvidia.com/gpu: 1
    requests:
      nvidia.com/gpu: 1
 restartPolicy: Never
EOF
```

```
$ oc create -f
tensorflow-benchmarks-gpu.yaml
pod/tensorflow-benchmarks-gpu created
```

- Simple quick jobs
- Optional training dataset
- Can be added in the monitoring

CPU

```
$ oc logs tensorflow-benchmarks-cpu

Step              Img/sec                        total_loss
1 images/sec: 2.2 +/- 0.0 (jitter = 0.0)  8.108
10  images/sec: 2.2 +/- 0.0 (jitter = 0.0)  8.122
20  images/sec: 2.2 +/- 0.0 (jitter = 0.0)  7.983
...
--------------------------------------------------
total images/sec: 2.24
```

GPU

```
$ oc logs tensorflow-benchmarks-gpu

Step              Img/sec                        total_loss
1 images/sec: 327.4 +/- 0.0 (jitter = 0.0)  8.108
10  images/sec: 326.5 +/- 0.7 (jitter = 1.0)  8.122
20  images/sec: 327.2 +/- 0.4 (jitter = 0.6)  7.983
...
--------------------------------------------------
total images/sec: 325.03
```

TensorFlow

```
TensorFlow:   1.14
Model:        resnet50
Dataset:      imagenet
Mode:         training
Accelerator:  GPU
Adaptor:      1 x V100
```

Thoth

Thoth

```
import tensorflow as tf
from flask import Flask

application = Flask()
```

TensorFlow

59 releases

28 releases

Flask

Combinations of TensorFlow and Flask 59 * 28 = 1,652
+ Flask dependencies (click, itsdangerous, jinja2, ...) = 54,395,000
+ TensorFlow dependencies = 139,740,802,927,165,440,000

Thoth

- Open source project
- Latest versions are not always greatest choices.
- Create knowledge base
  - What packages in which versions should I use?
    - Application builds correctly
    - Application runs correctly
    - Application behaves and performs well
- Create an advanced Python resolver which uses knowledge base to resolve software stacks

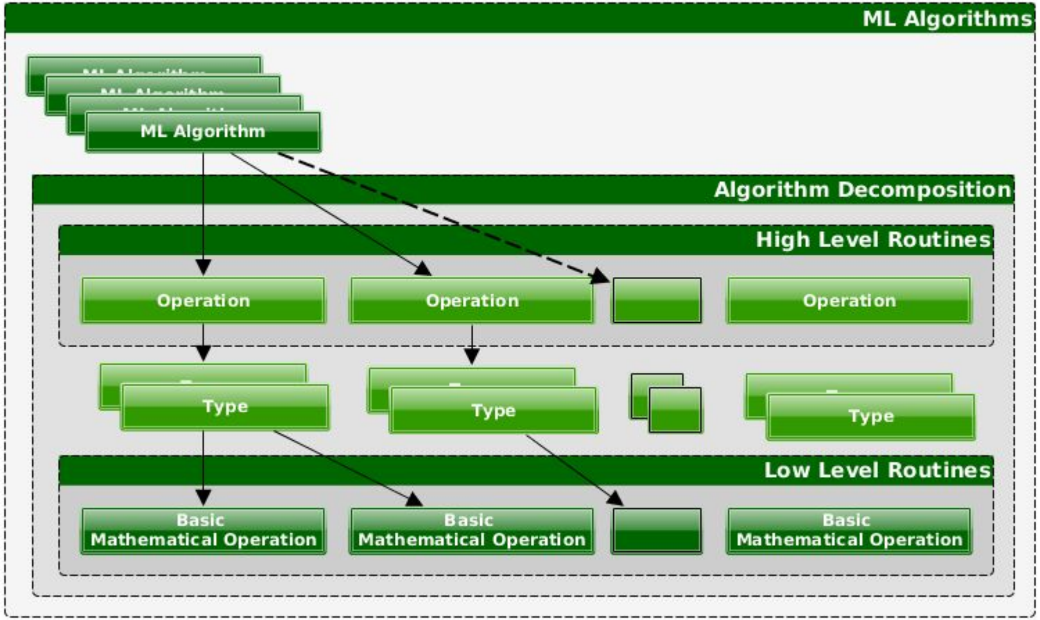
```
$ pip3 install thamos
$ cd ~/repositories/my-repo/
$ thamos config
$ thamos advise
```

Thoth PI

	Benchmark (High Level Test)	Thoth PI	Micro-benchmark (Low Level Test)
Goal	Measure system performance for both training and inference from mobile devices to cloud services.	Evaluate Performance Indicators that can be used to recommend AI software stacks.	Benchmark operations that are important to deep learning on different hardware platforms.
Metrics	• Time • FLOPS • Cost	• Time • FLOPS	• Time • FLOPS
Time requested for benchmarking	~hours, days	~minutes, (hours)	~seconds, minutes
Using ML Frameworks	Yes	Yes	No
Phase of ML workflow	Training/Inference	Training/Inference	Training/Inference

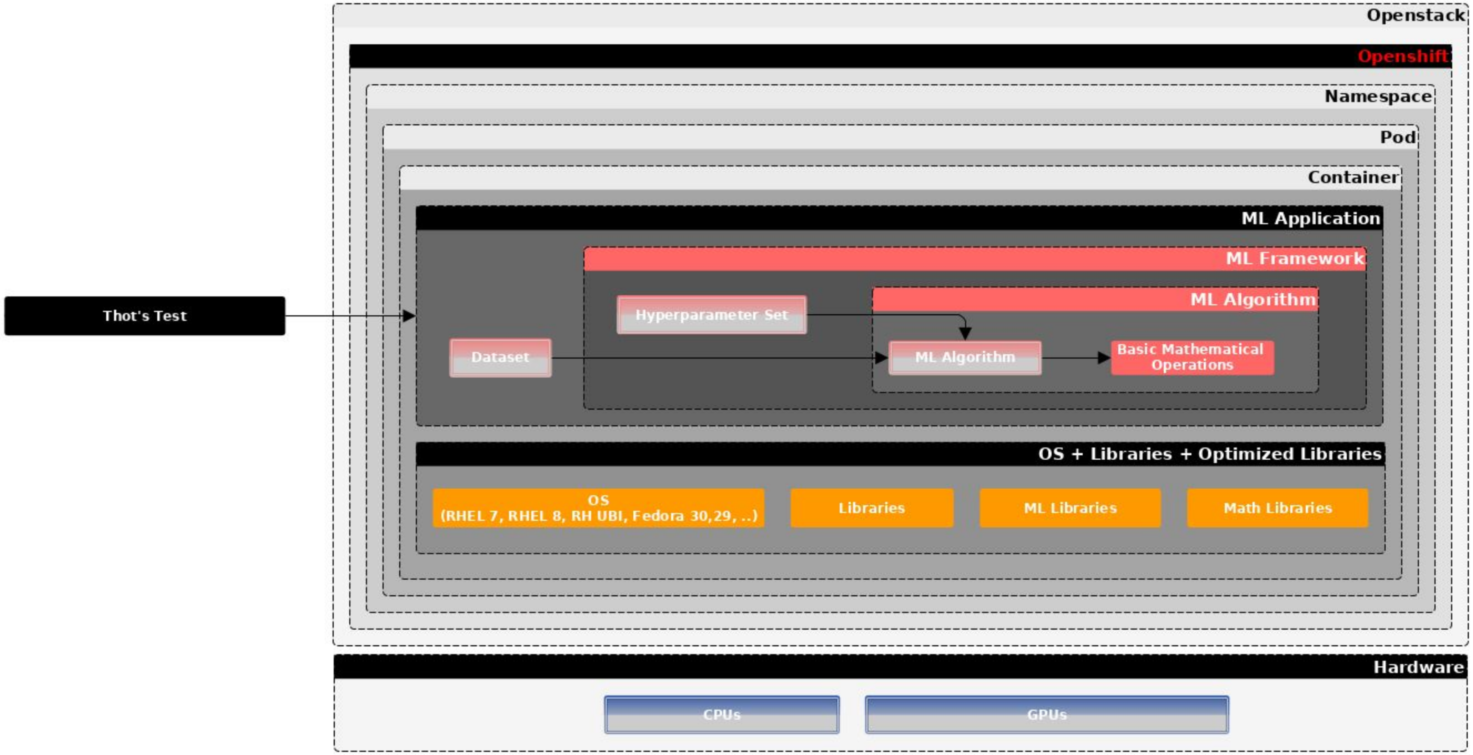
Thoth PI

Algorithm decomposition

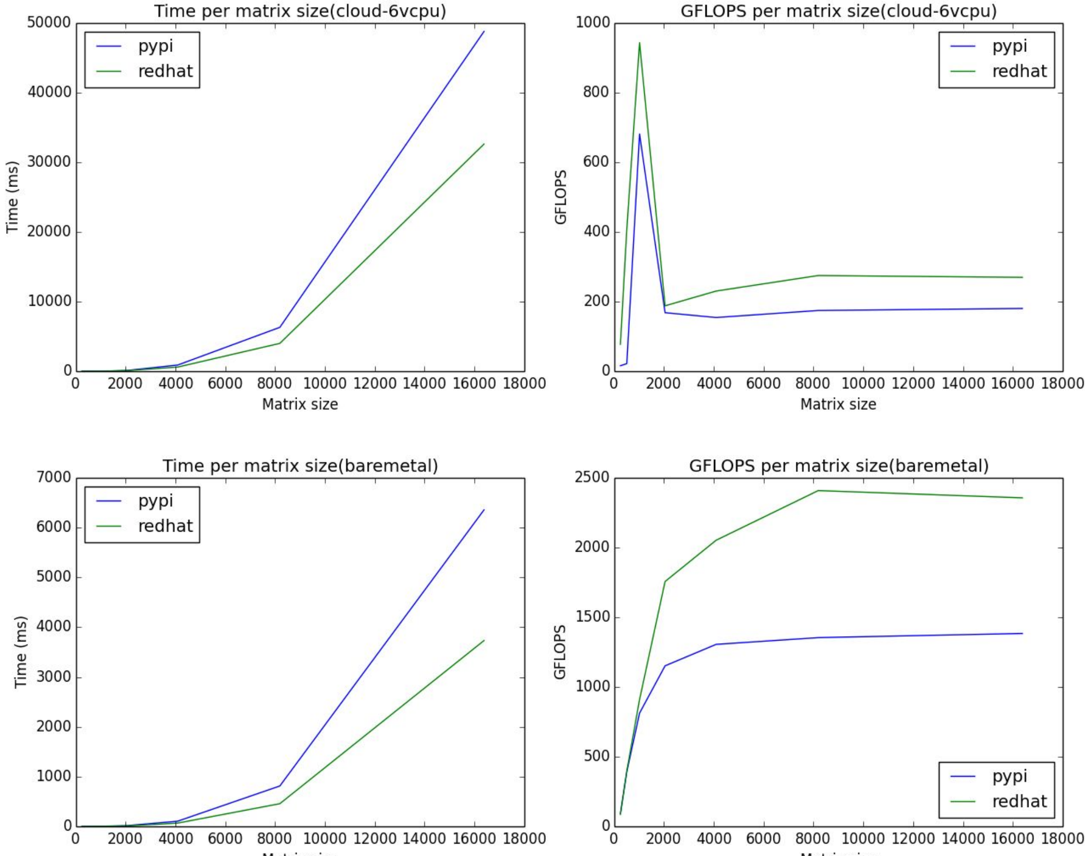


Source: [Machine Learning Frameworks *Overview, Example Applications, and Test Patterns*]

Thoth PI


Thot's Test
Openstack
Openshift
Namespace
Pod
Container
ML Application
ML Framework
ML Algorithm
Hyperparameter Set
Dataset
ML Algorithm
Basic Mathematical Operations
OS + Libraries + Optimized Libraries
OS
(RHEL 7, RHEL 8, RH UBI, Fedora 30,29, ..)
Libraries
ML Libraries
Math Libraries
Hardware
CPUs
GPUs

TensorFlow optimized



Red Hat

Key takeaways

- Benchmark your full Machine Learning stack
- NVIDIA with GPU hardware and software libraries is leading Deep Learning computing
- MLPerf is an agile industry standard
- CPU may be enough for simple inferencing on small datasets
- Take care of the NUMA affinity of your OpenStack compute nodes
- Use GPU certified servers and tested drivers for Kubernetes
- Compare with others with MLPerf
- Create quick benchmarks that can be added in your monitoring
- Drivers and libraries latest versions are not always the greatest choices
- Create your benchmarking knowledge base

Thank You

linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

facebook.com/redhatinc

twitter.com/RedHat