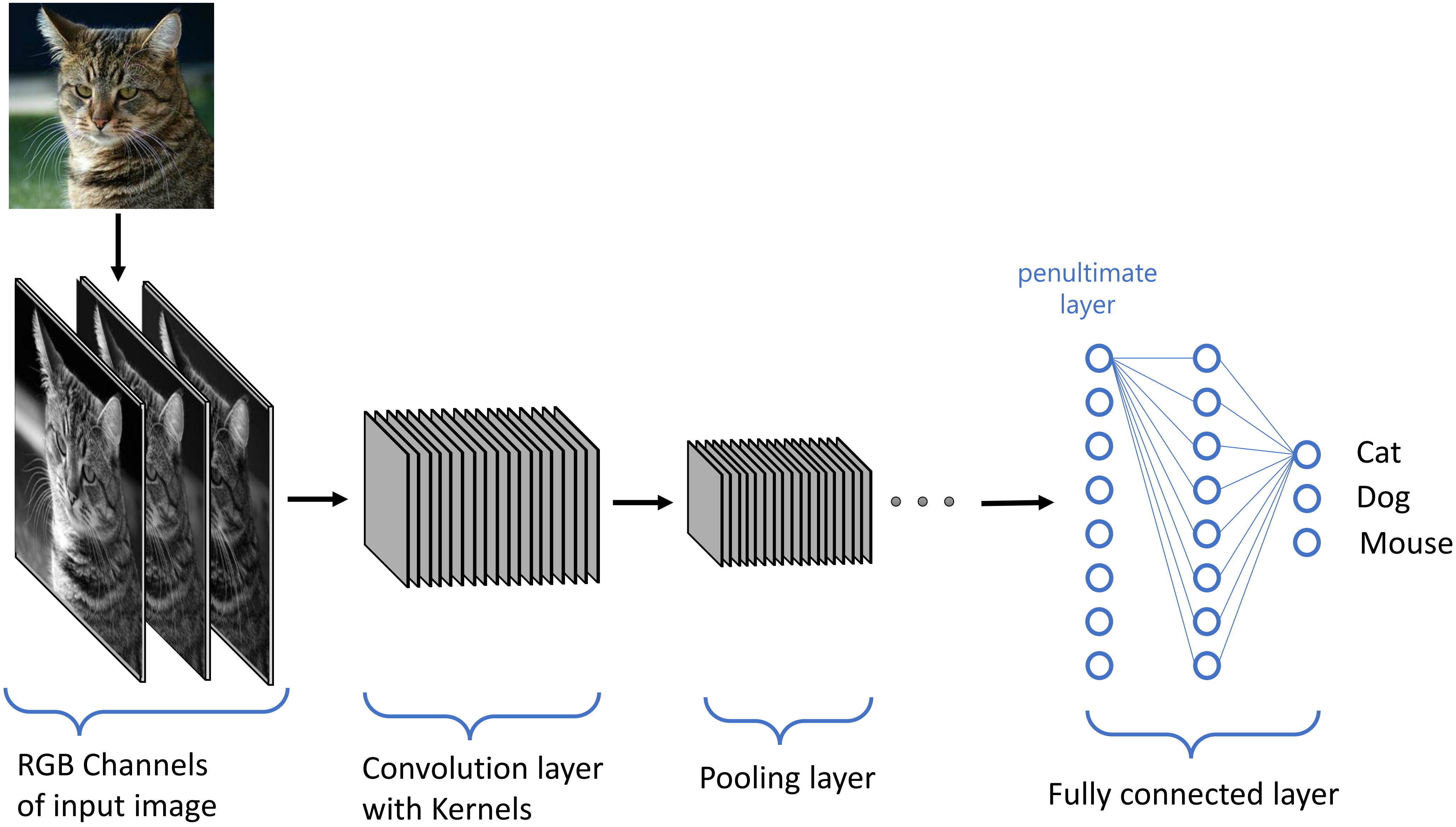
Mathew Salvaris

Distributed Deep Learning

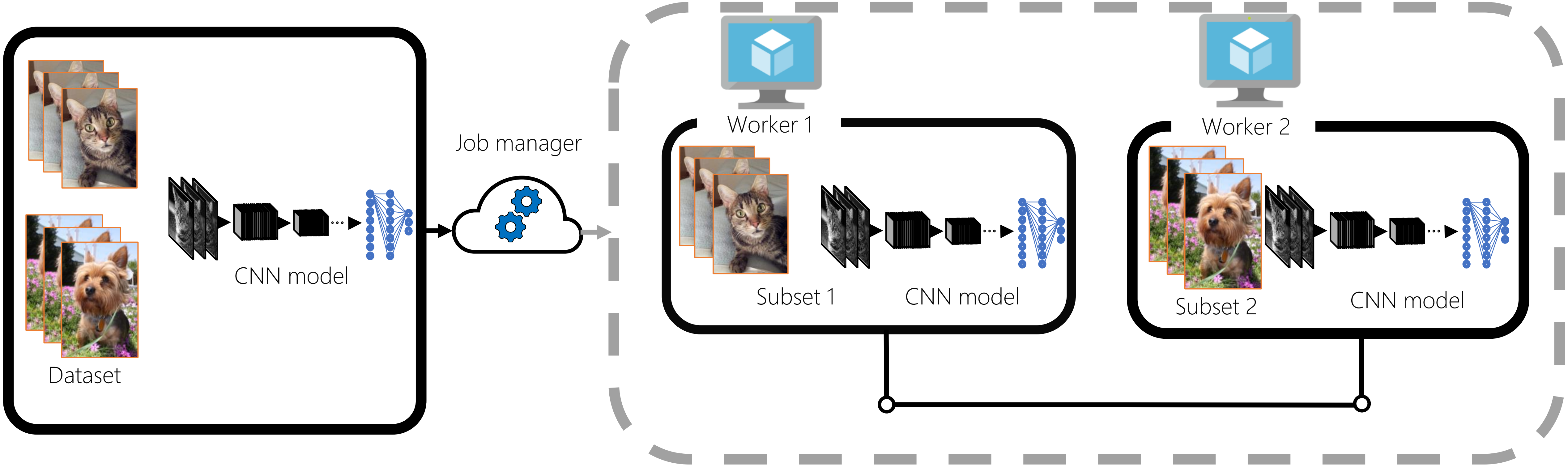
What will be covered

- Overview of Distributed Training
- What affects distributed training
  - Network
  - Model
  - Data location
  - Data format

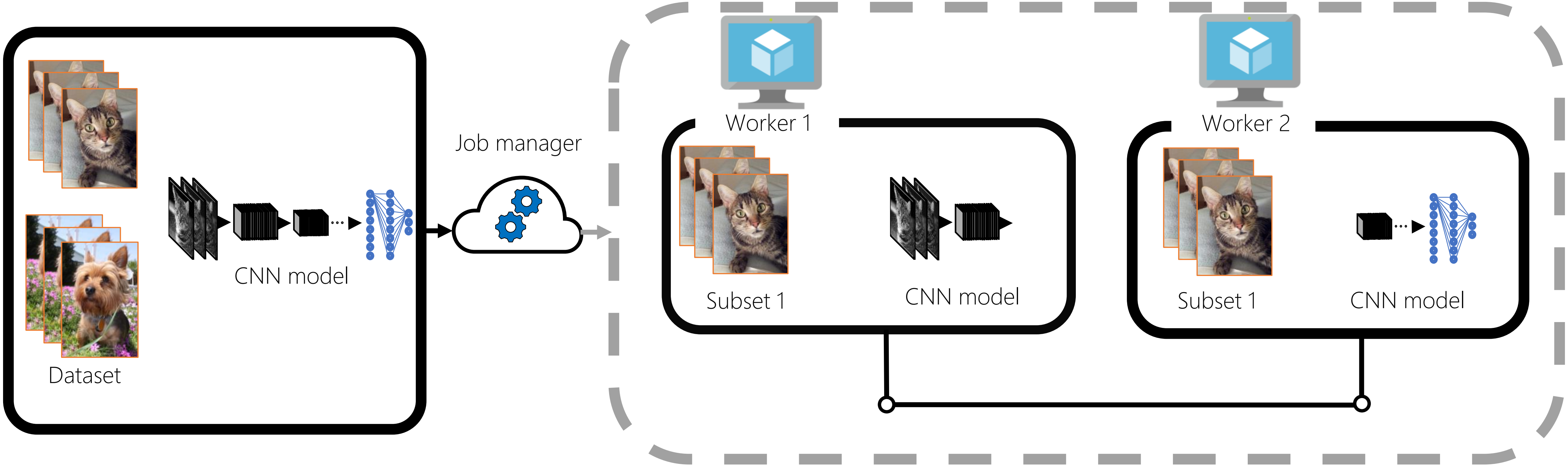
Deep Learning Model (CNN)



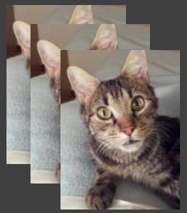
Distributed training mode: Data parallelism



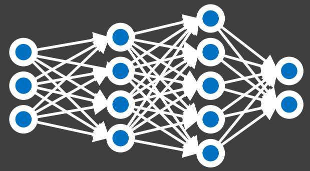
Distributed training mode: Model parallelism



Data parallelism vs model parallelism

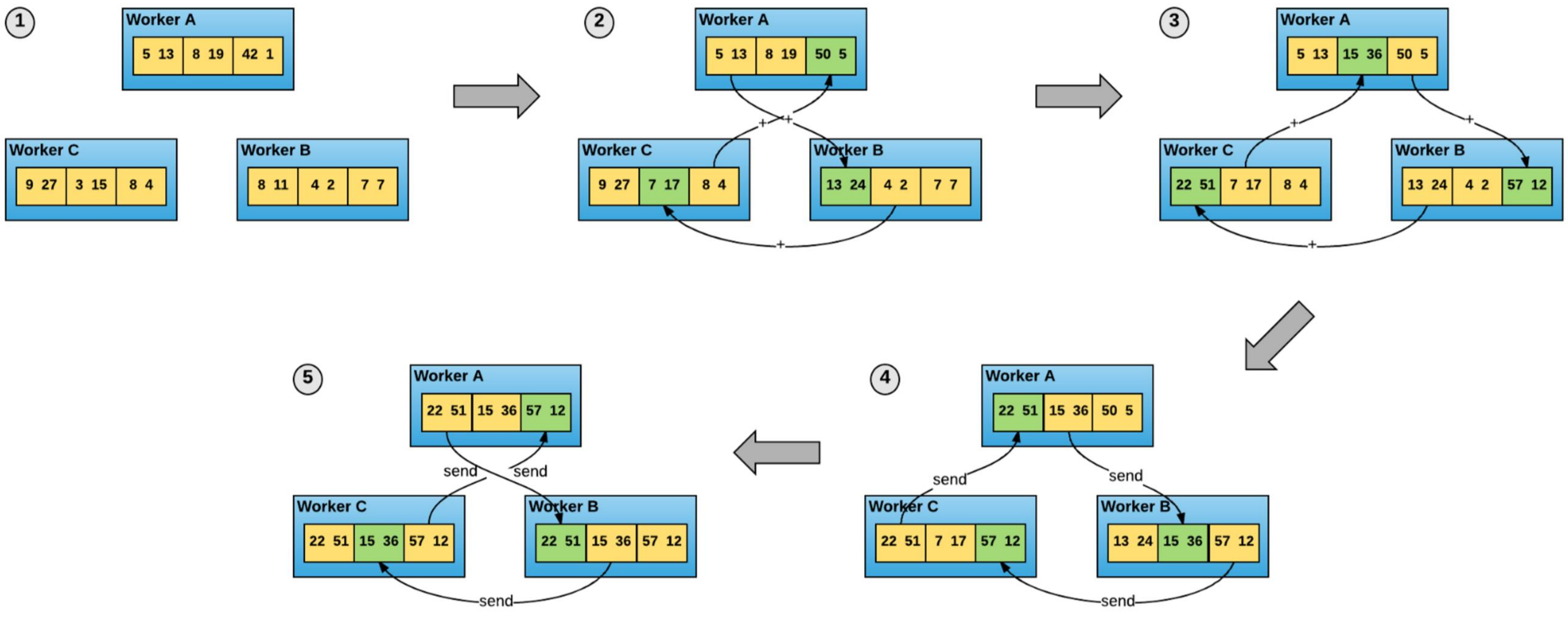
Data parallelism 

- Easier implementation
- Stronger fault tolerance
- Higher cluster utilization

Model parallelism 

- Better scalability of large models
- Less memory on each GPU

Horovod: Ring All Reduce



Effects of Network, Model and Precision

Clusters of 8 nodes using **K80**, **P40**, **P100** and **V100** (4 GPUs per node+Infiniband)

Two MPI configurations **OpenMPI+NCCL** and **IntelMPI**

Setup

Experiments

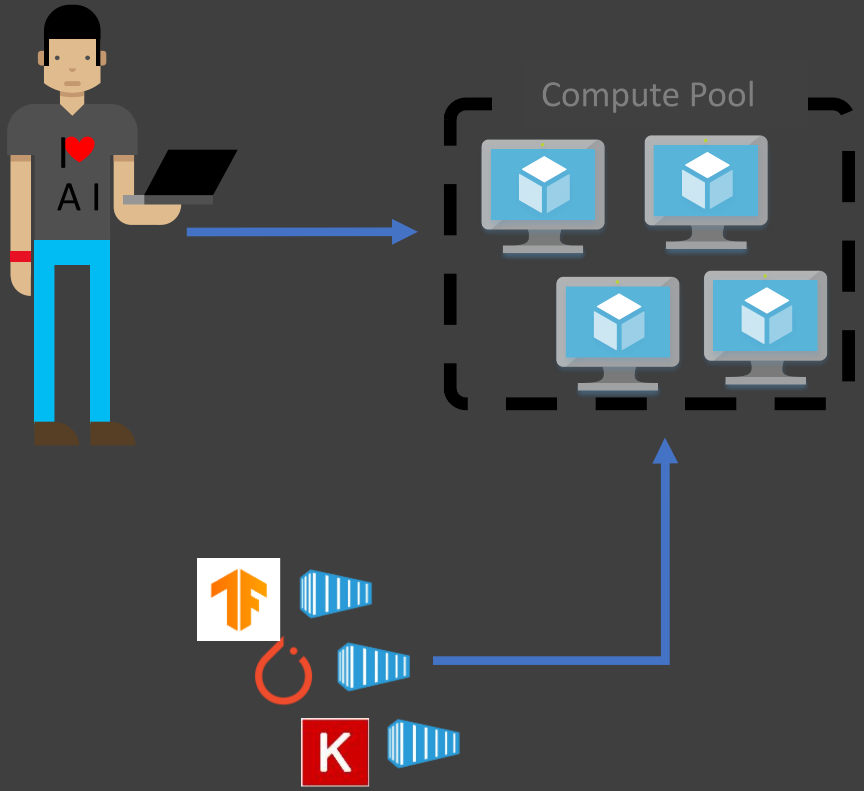
345 experiments across many different models including **ResNet50**, **MobileNet V2** etc.

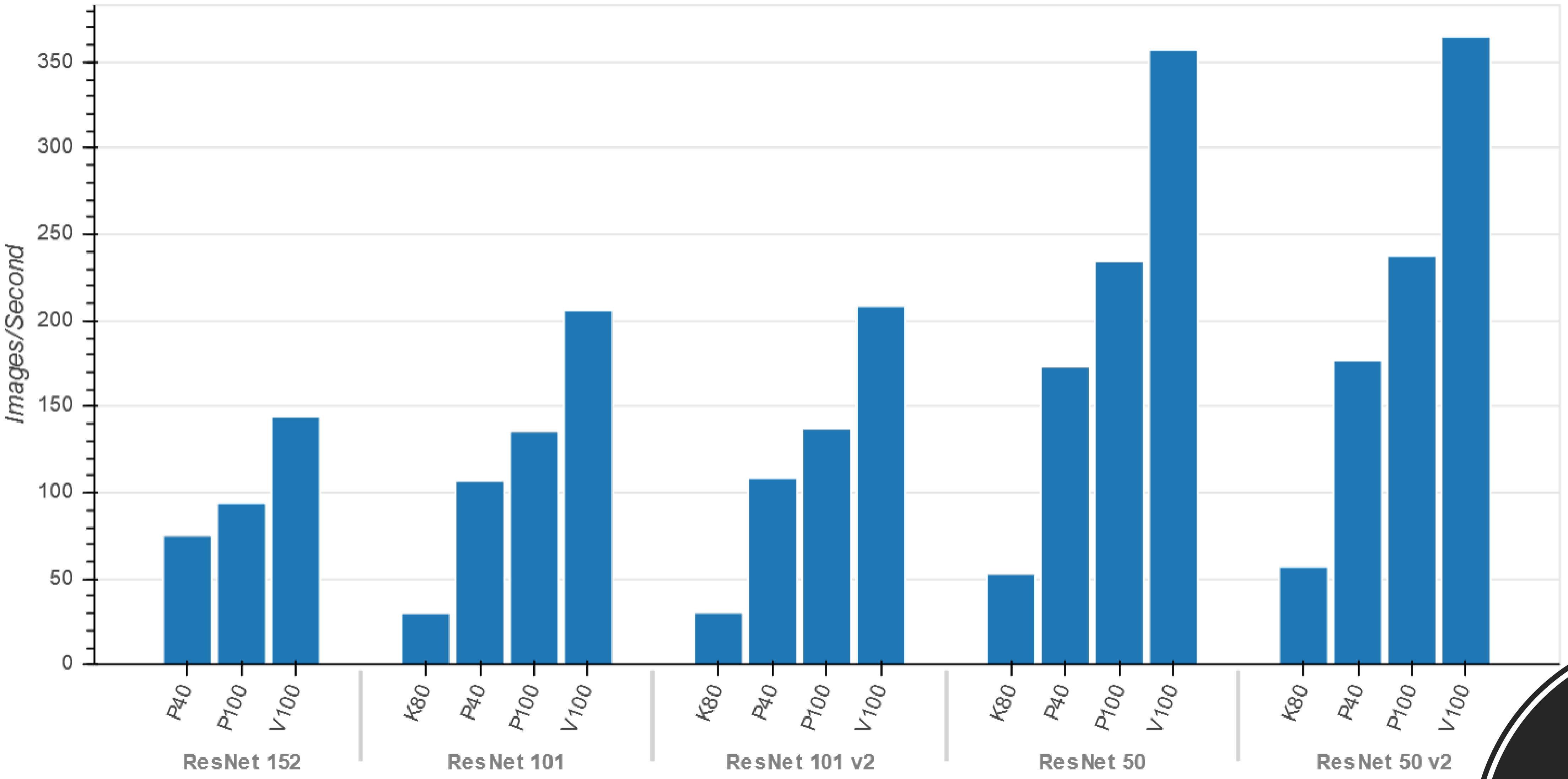
Using **synthetic** data

Batch size remains **64** across all models and GPUs

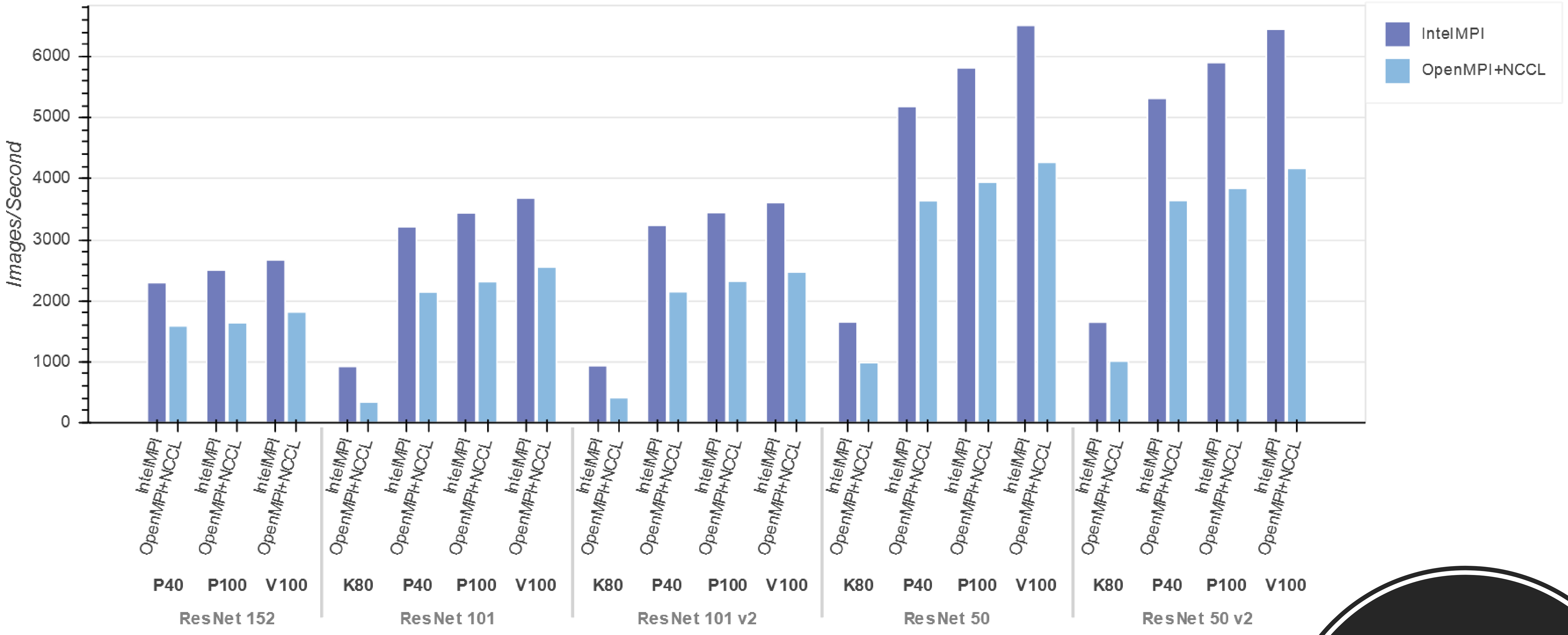
Use the benchmarking scripts from TensorFlow

Distributed training with synthetic data



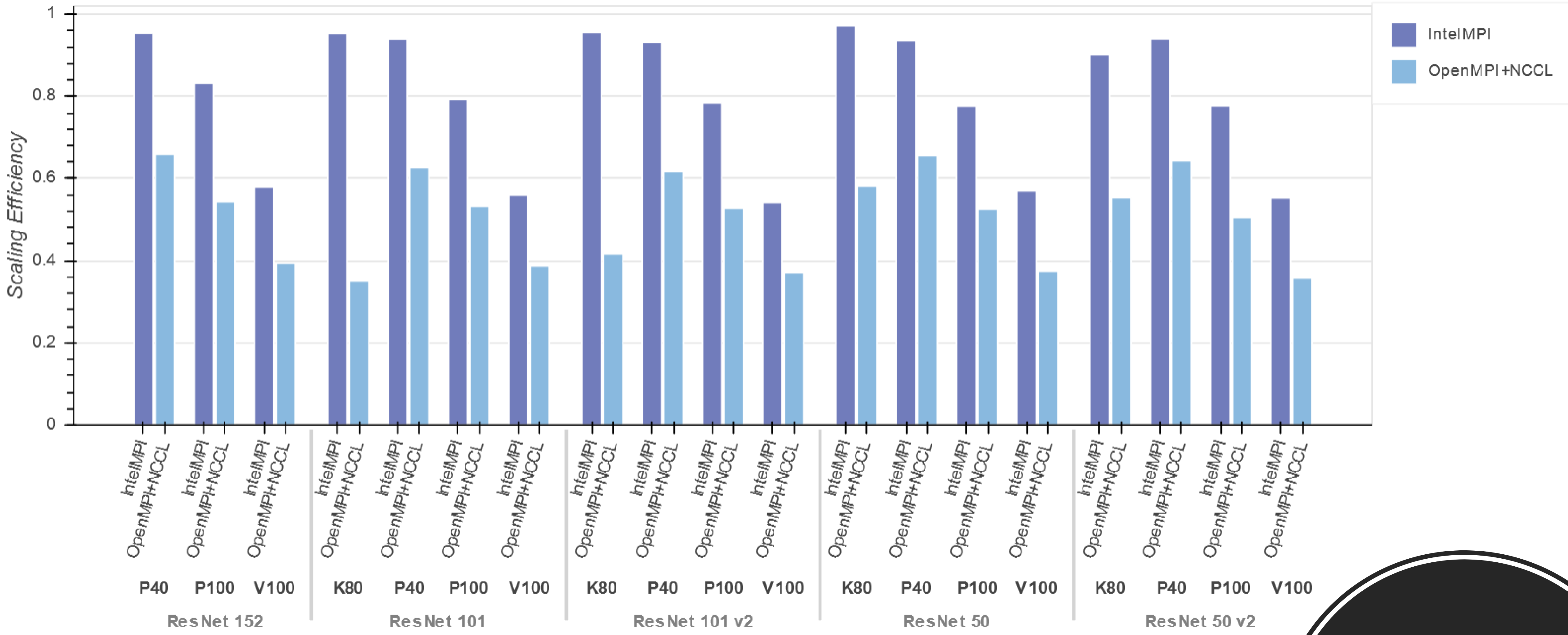

Images/Second
0
50
100
150
200
250
300
350
P40
P100
V100
K80
P40
P100
V100
K80
P40
P100
V100
K80
P40
P100
V100
K80
P40
P100
V100
ResNet 152
ResNet 101
ResNet 101 v2
ResNet 50
ResNet 50 v2

Single GPU



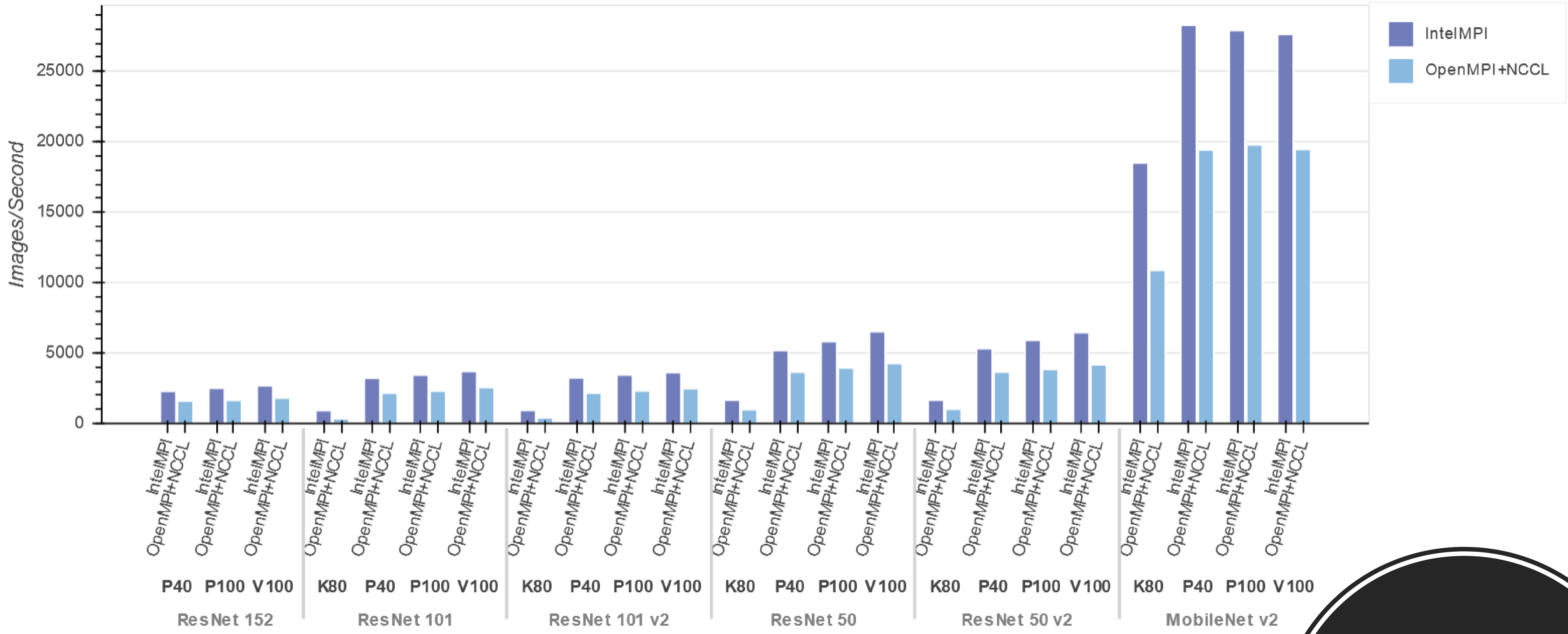
Images/Second
0
1000
2000
3000
4000
5000
6000
IntelMPI
OpenMPI+NCCL
P40
P100
V100
ResNet 152
K80
P40
P100
V100
ResNet 101
K80
P40
P100
V100
ResNet 101 v2
K80
P40
P100
V100
ResNet 50
K80
P40
P100
V100
ResNet 50 v2

32 GPUs

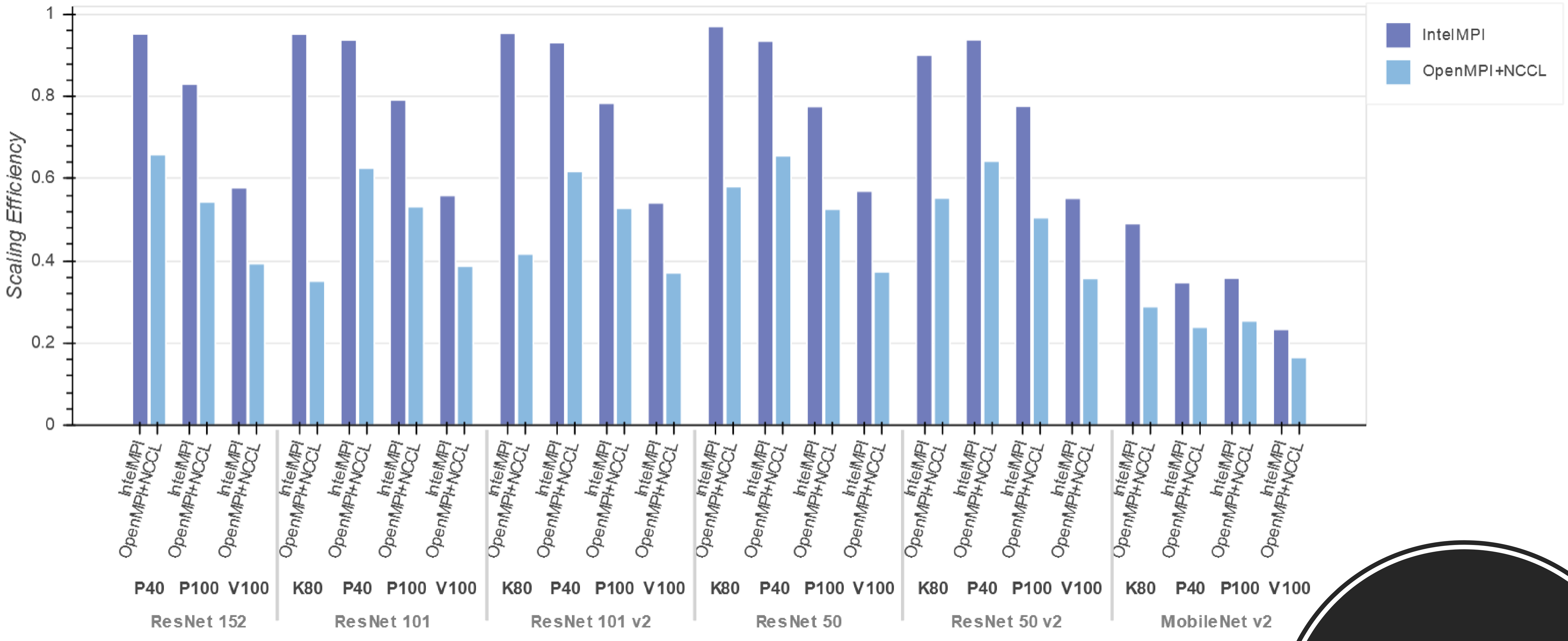


Scaling Efficiency
1
0.8
0.6
0.4
0.2
0
IntelMPI
OpenMPI+NCCL
P40
P100
V100
K80
ResNet 152
ResNet 101
ResNet 101 v2
ResNet 50
ResNet 50 v2

32 GPUs

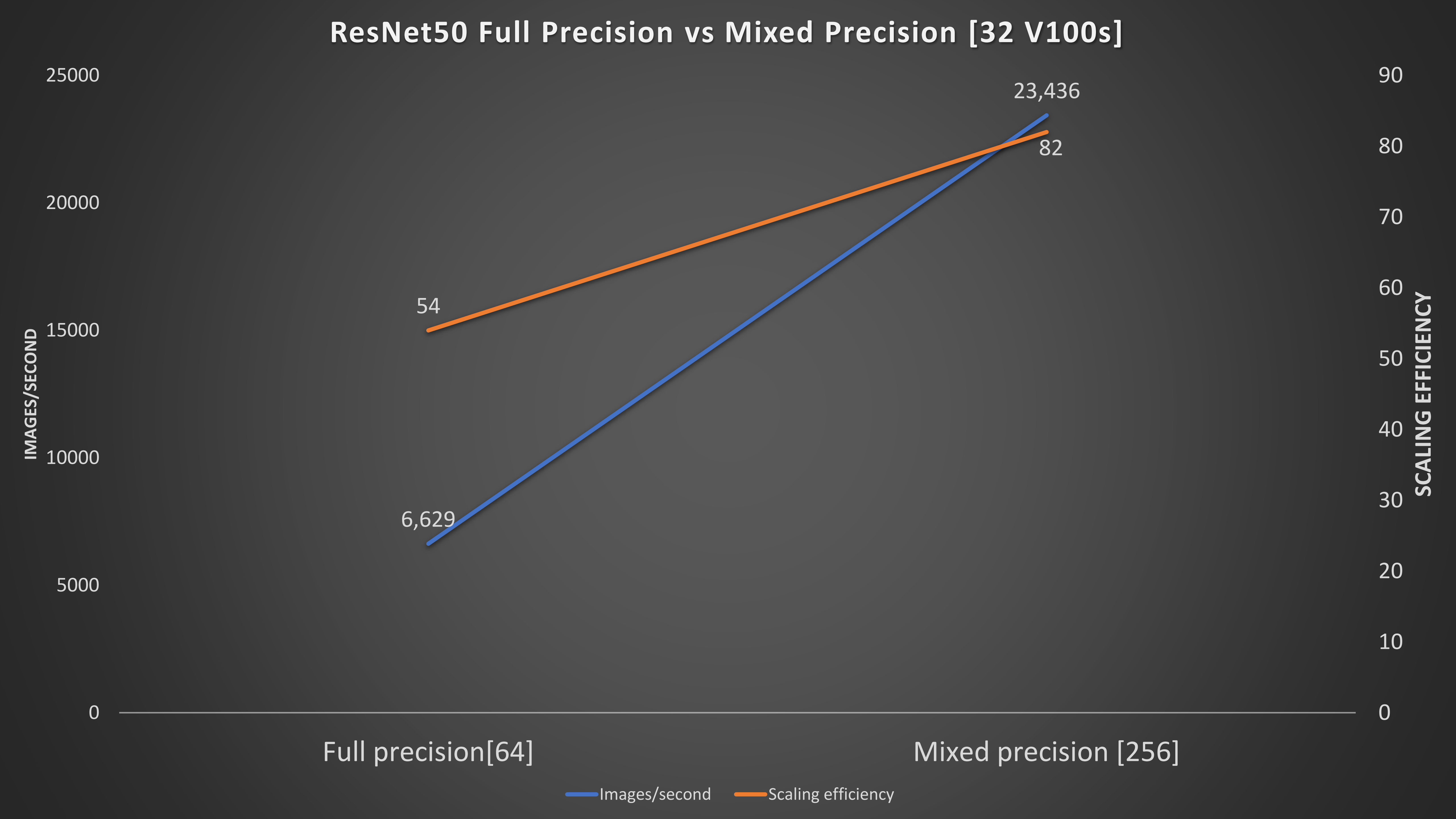

Images/Second
0
5000
10000
15000
20000
25000
IntelMPI
OpenMPI+NCCL
P40 P100 V100
ResNet 152
K80 P40 P100 V100
ResNet 101
K80 P40 P100 V100
ResNet 101 v2
K80 P40 P100 V100
ResNet 50
K80 P40 P100 V100
ResNet 50 v2
K80 P40 P100 V100
MobileNet v2

MobileNet


Scaling Efficiency
1
0.8
0.6
0.4
0.2
0
IntelMPI
OpenMPI+NCCL
P40 P100 V100
ResNet 152
K80 P40 P100 V100
ResNet 101
K80 P40 P100 V100
ResNet 101 v2
K80 P40 P100 V100
ResNet 50
K80 P40 P100 V100
ResNet 50 v2
K80 P40 P100 V100
MobileNet v2

MobileNet


GPU
K80
P40
P100
V100
Batch Execution
Data Transfer
Time it takes to process batch on GPU
Time it takes to transfer weights between GPUs


ResNet50 Full Precision vs Mixed Precision [32 V100s]
IMAGES/SECOND
SCALING EFFICIENCY
25000
20000
15000
10000
5000
0
90
80
70
60
50
40
30
20
10
0
23,436
82
54
6,629
Full precision[64]
Mixed precision [256]
Images/second
Scaling efficiency

Effects of Storage

Using **ResNet50 across three frameworks [PyTorch, TensorFlow, Keras]**

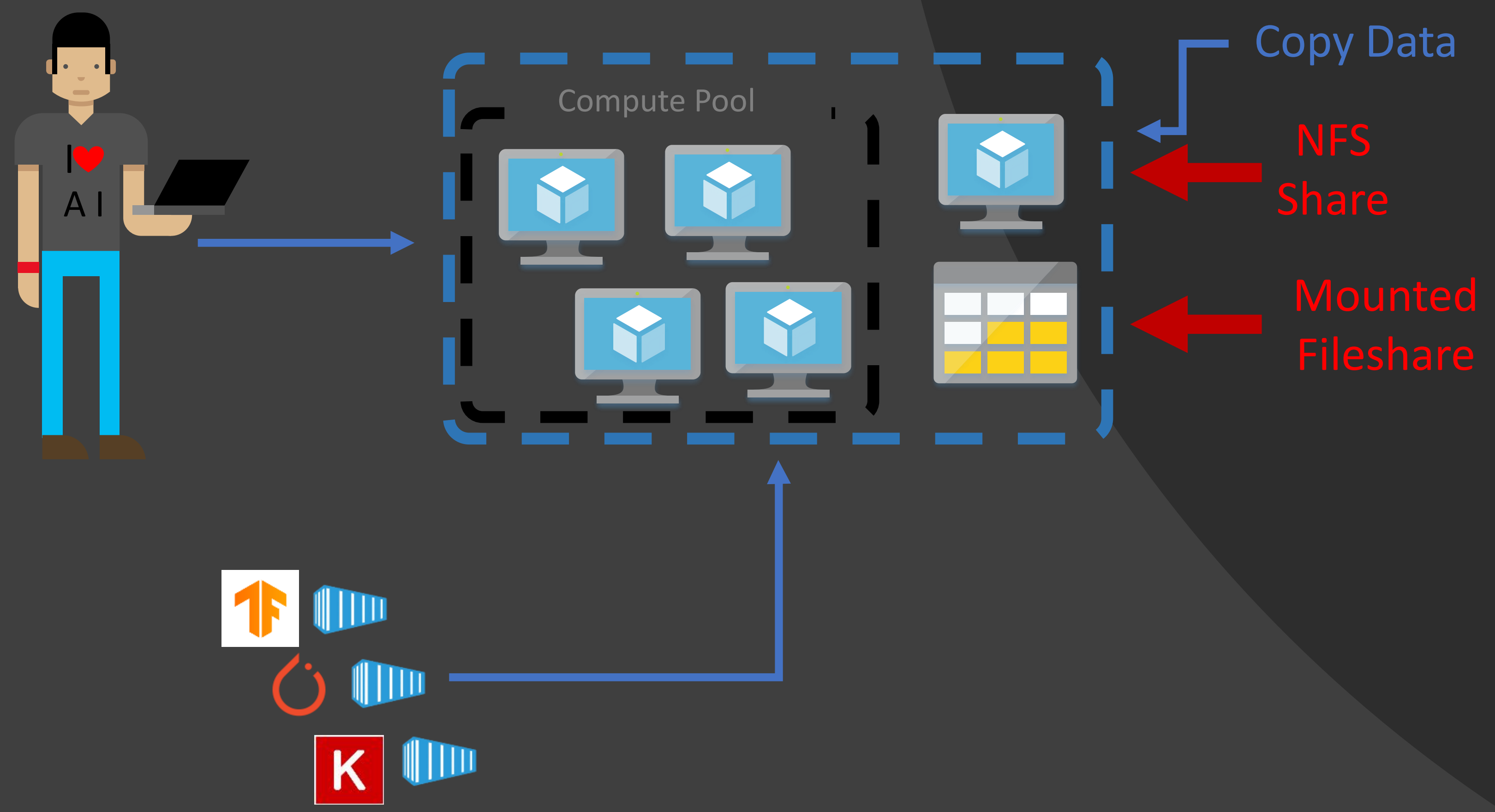
Using **real and synthetic** data. Real data on local, NFS and Blob storage

Batch size remains **64** across all configurations

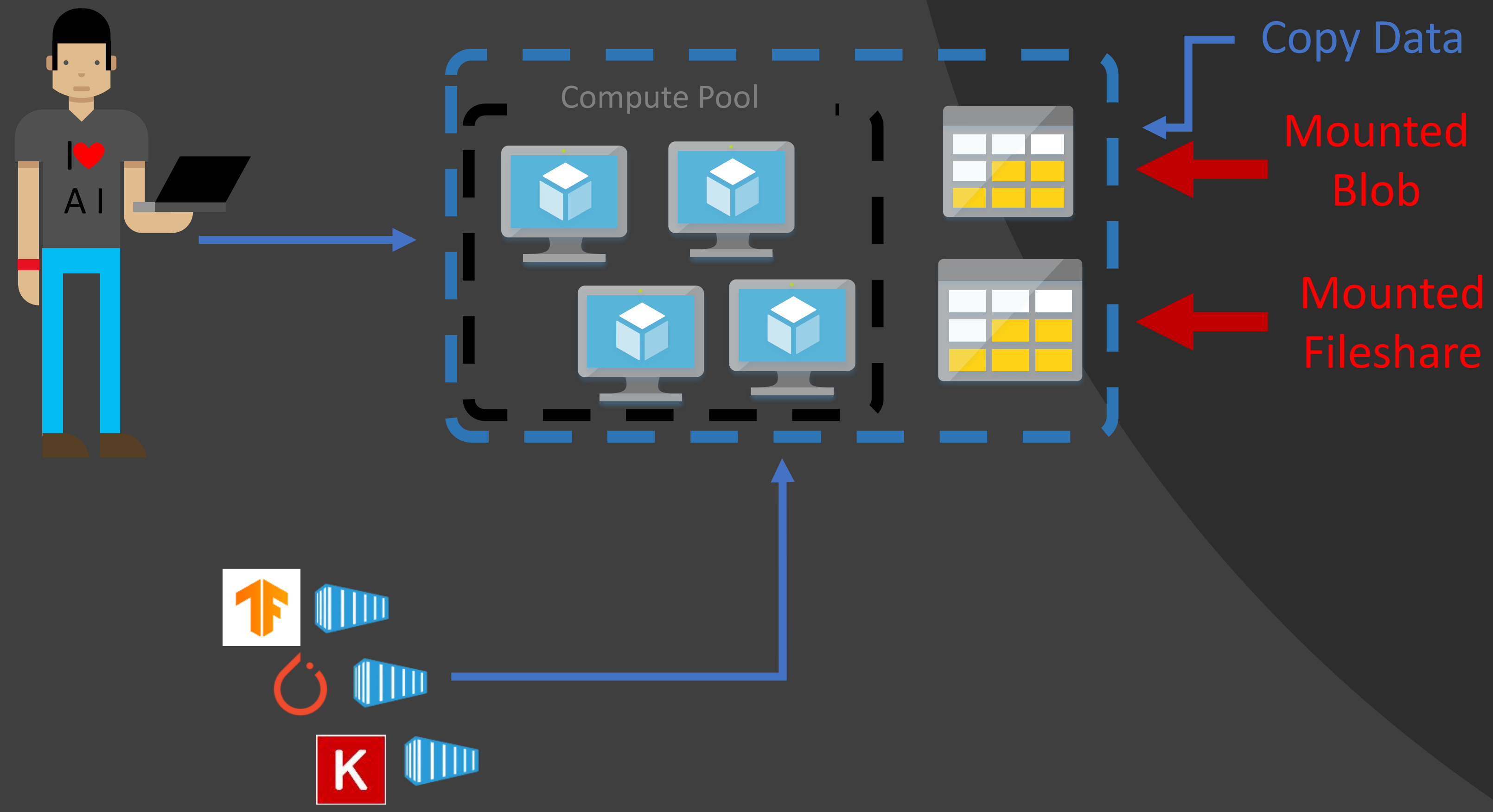
Uses **V100** GPUs

Experiments

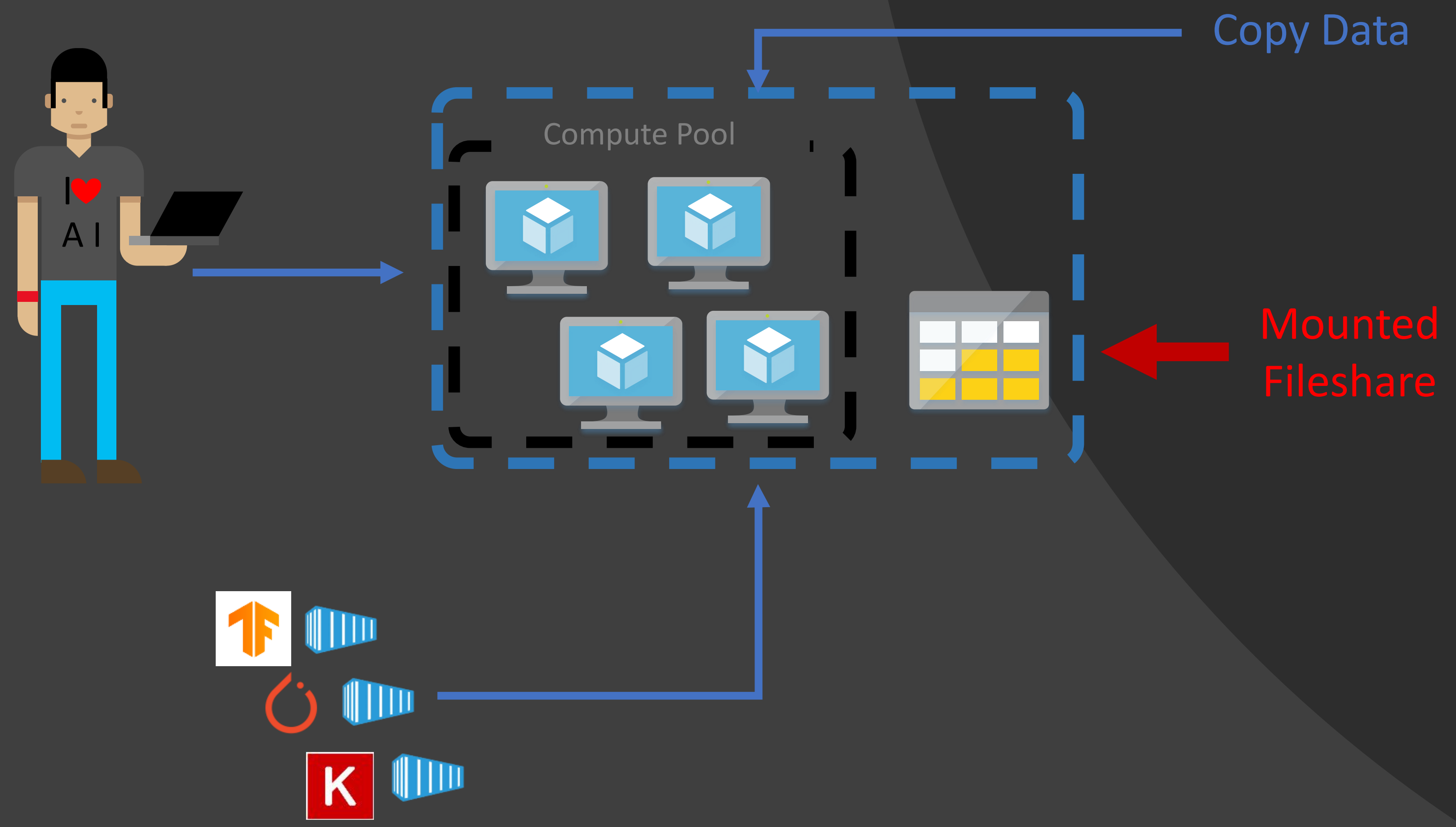
Distributed training with NFS

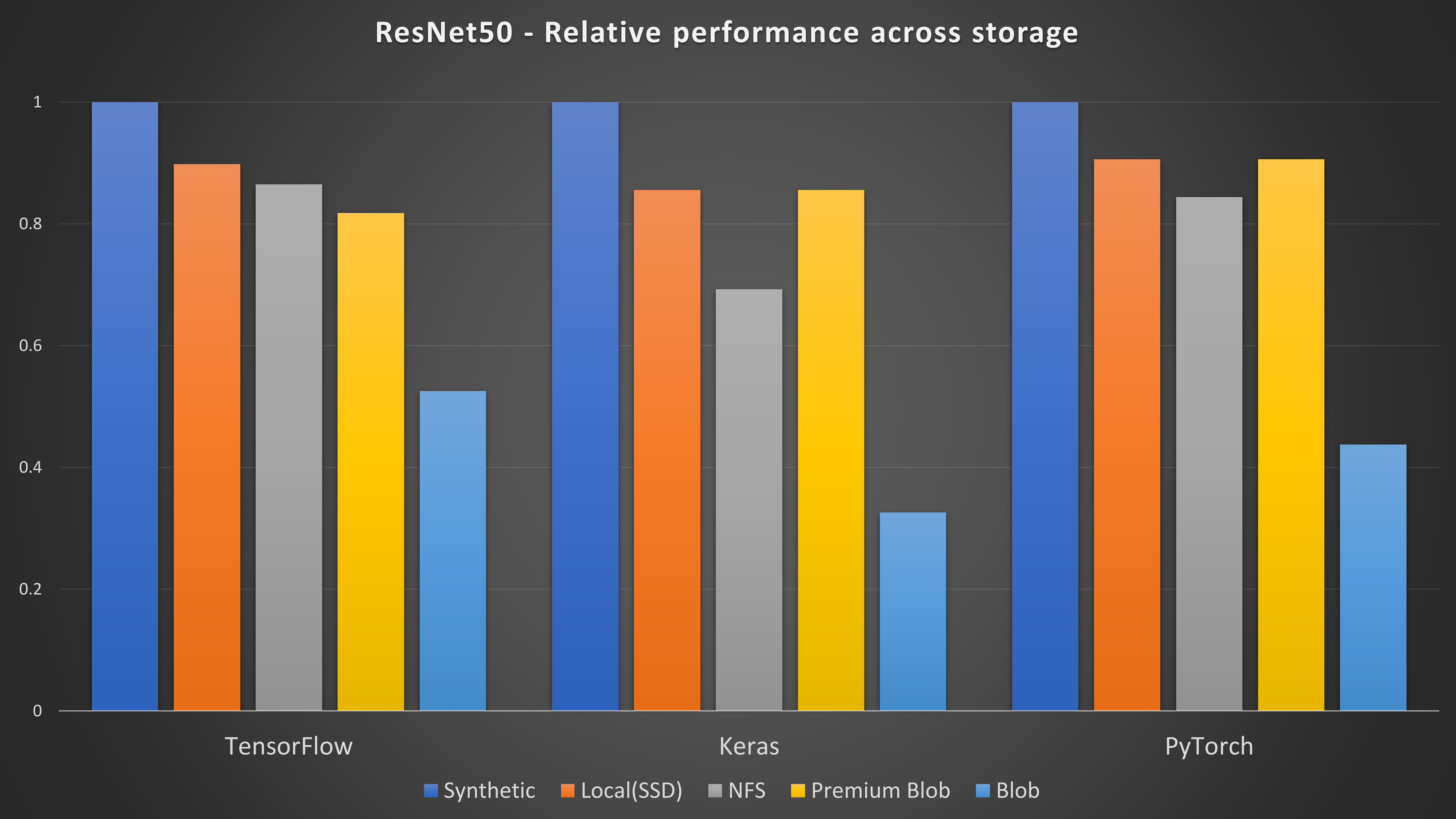


Distributed training with blob storage



Distributed training with local storage




ResNet50 - Relative performance across storage
1
0.8
0.6
0.4
0.2
0
TensorFlow
Keras
PyTorch
Synthetic
Local(SSD)
NFS
Premium Blob
Blob

Data Loaders and Preprocessors

Keras Data Loader

Simple with no parameters for buffering and parallelizing

PyTorch Data Loader

Specify number of workers with *num_workers*

TensorFlow

Highly configurable

Many options : buffer, shuffle, cache and shard

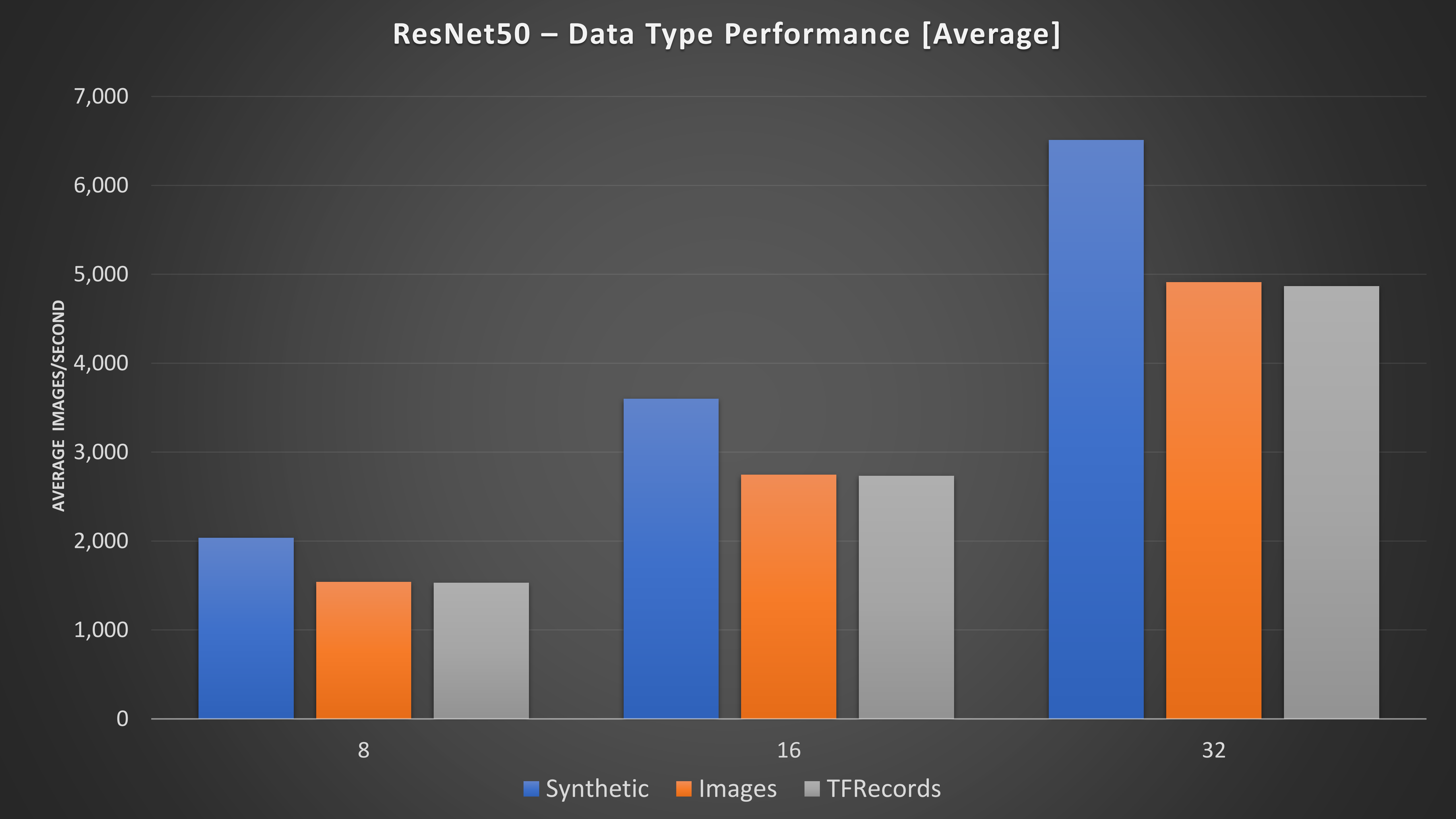
Daunting and easy to get wrong

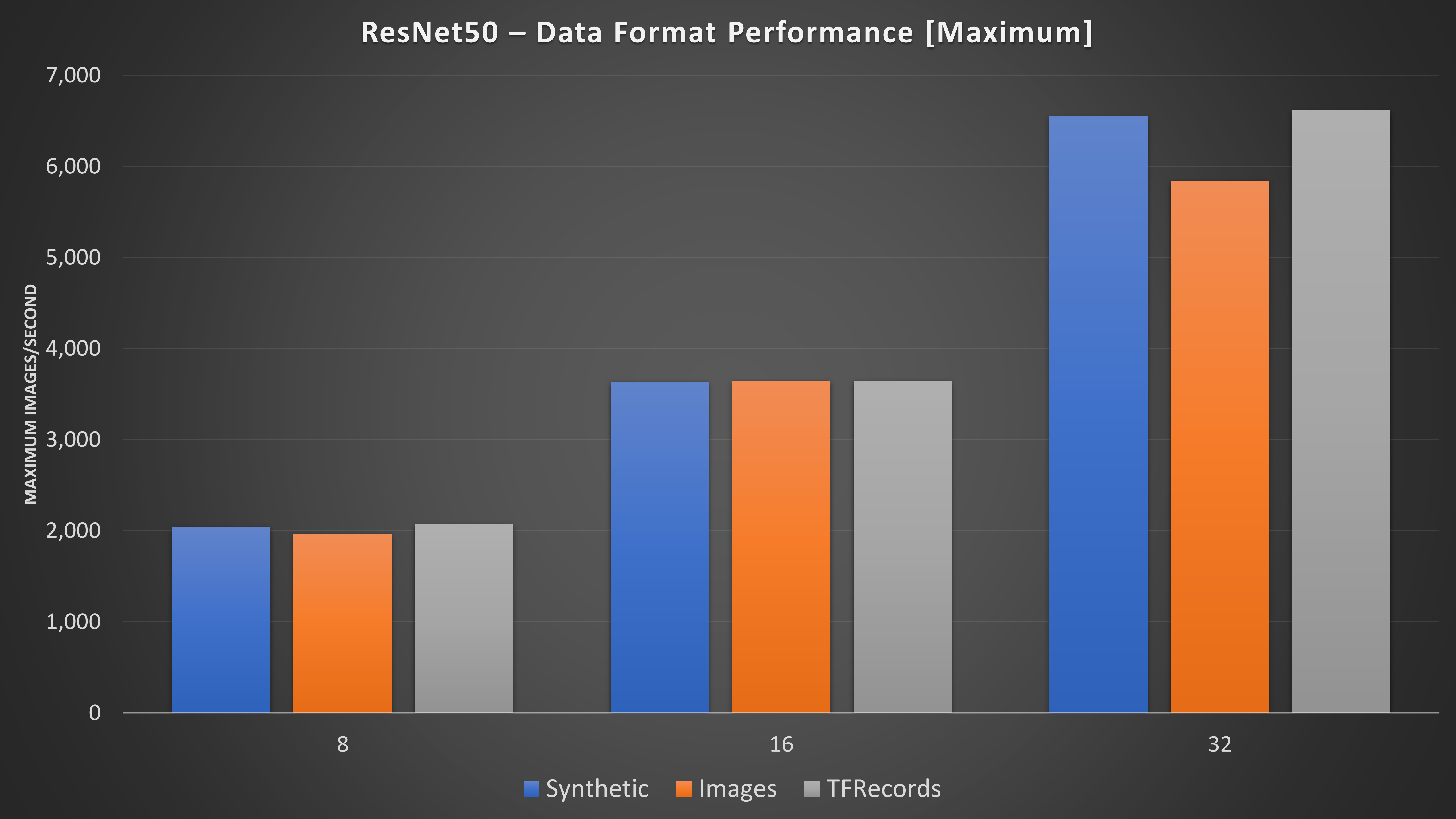
https://www.tensorflow.org/guide/performance/datasets

Effects of Data Type

TensorFlow Records

- Binary data format created for TensorFlow – Recommended format for TensorFlow
- Can aggregate number of examples to smaller number of TFRecords – efficient for transferring and reading in the cloud
- Have to export data to format - Has to be tailored to use case


ResNet50 – Data Type Performance [Average]
AVERAGE IMAGES/SECOND
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
8
16
32
Synthetic
Images
TFRecords


ResNet50 – Data Format Performance [Maximum]
MAXIMUM IMAGES/SECOND
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
8
16
32
Synthetic
Images
TFRecords

Things not discussed

- Asynchronous distributed training
- Tradeoff between batch size and other parameters
- Optimization of TensorFlow pipeline
- Other data formats such as Parquet (Petastorm)
- Transform libraries [albumentations]
- Distributed file systems BeeGFs and other storage GlusterFS, Lustre etc.
- Models other than CNN

Summary

Do try to use enhanced networking wherever possible especially for the latest GPUs

Training small models using distributed training is not recommended

Do use TFRecords or other columnar or row based data formats

Not all data loaders are equal

Thanks
&
Questions?