HIGH-PERFORMANCE INPUT PIPELINES FOR SCALABLE DEEP LEARNING

Miroslav Klivansky
Pure Storage

school of ai

QUESTION ON EVERYONE'S MIND:

WHY IS A STORAGE COMPANY HERE?

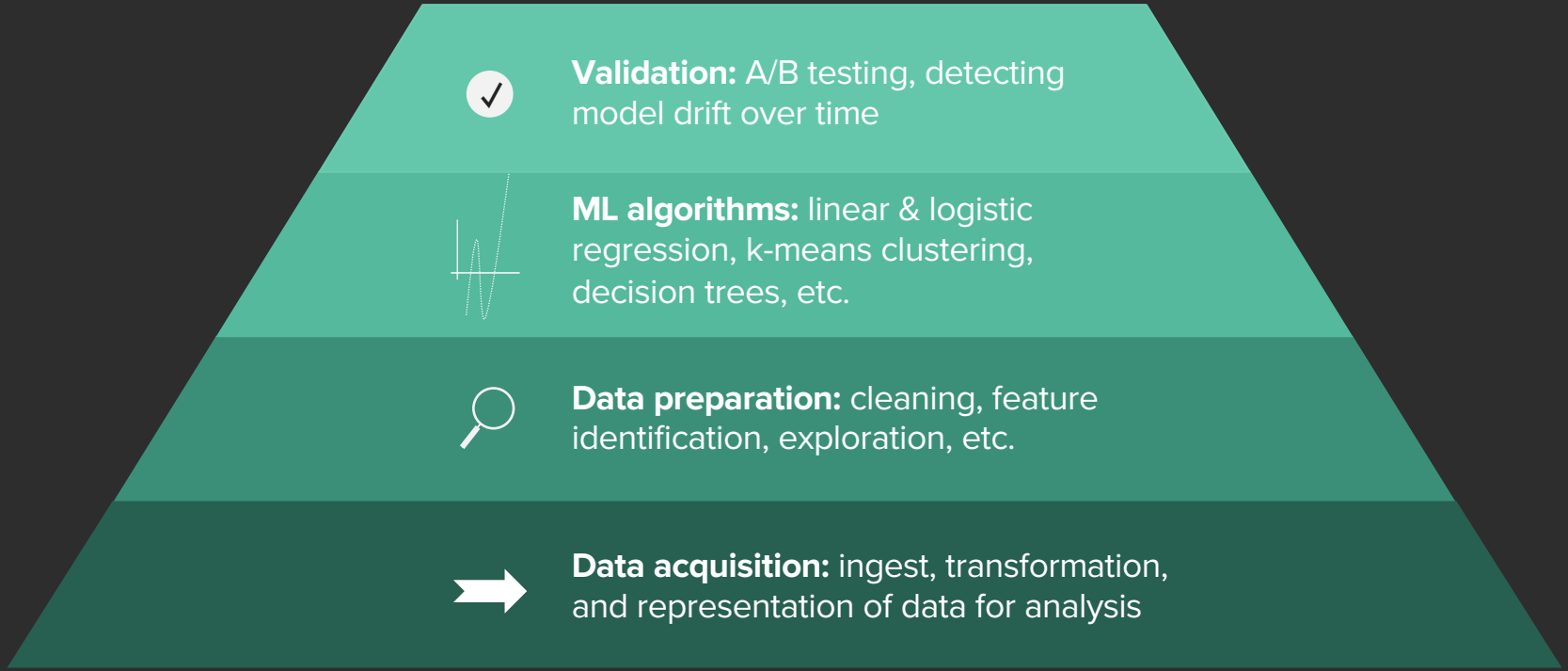
"We don't have better algorithms, we just have more data"

PETER NORVIG
Engineering Director, Google

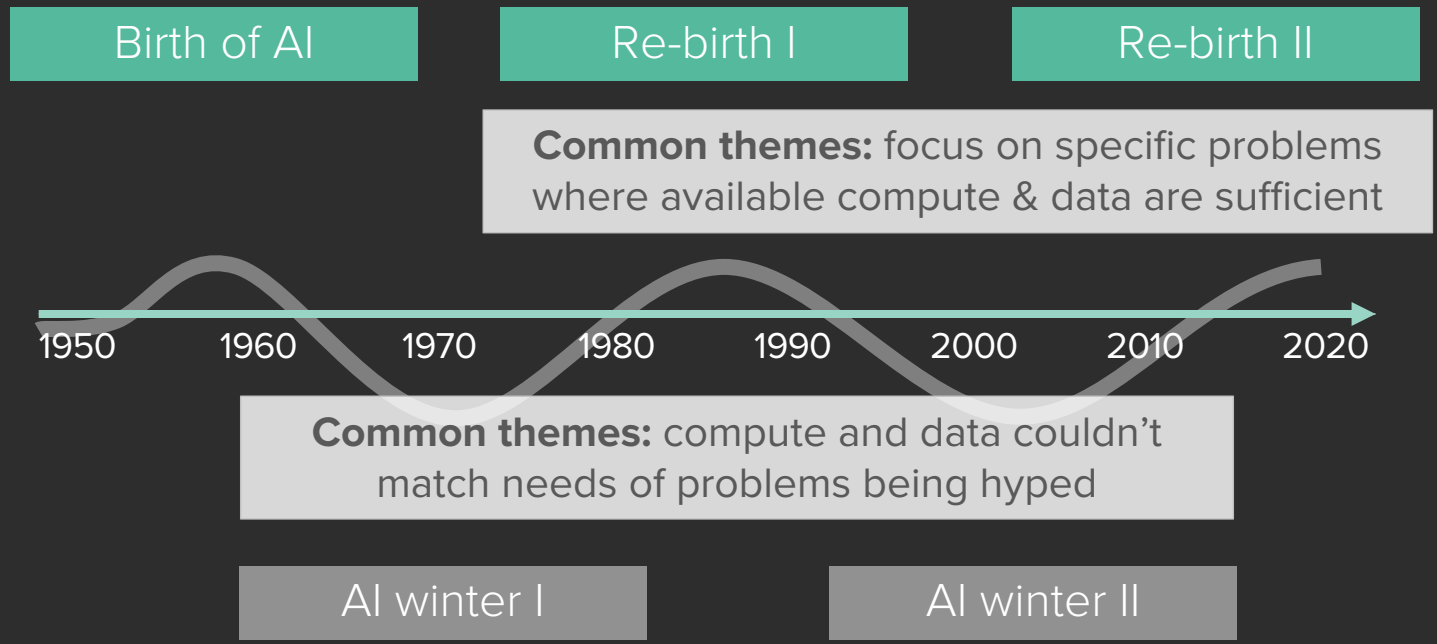
PURESTORAGE®

The AI “Hierarchy of Needs”

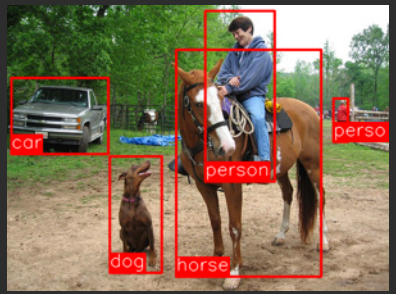
credit: Monica Rogati

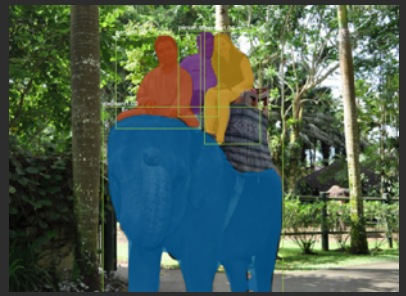


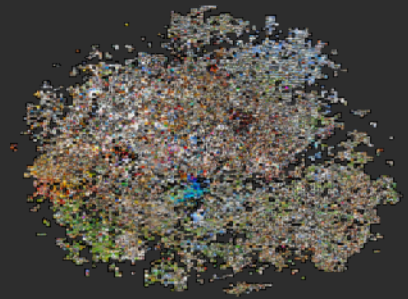
THIS IS NOT THE FIRST AI HYPE WAVE

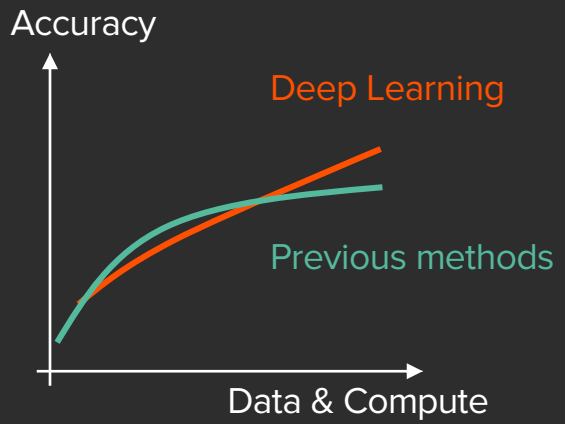


DEEP LEARNING = MASSIVE DATA & COMPUTE









STATE-OF-THE-ART RESULTS ACROSS VISION, SPEECH, LANGUAGE, AND MORE

PURESTORAGE®

Sources: https://arxiv.org/abs/1506.01497; https://arxiv.org/abs/1703.06870; https://shubhangdesai.github.io/blog/Neural-Style.html; https://cs.stanford.edu/people/karpathy/cnnembed/

THE INTUITION BEHIND DEEP LEARNING

deep
neural net

"dog"

Pr{dog} = 0.903
Pr{cat} = 0.072

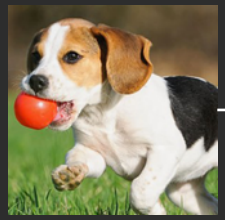
Primitives

Rough shapes

Macro features

PURESTORAGE®

TRAINING A DEEP NEURAL NETWORK



evaluate → compute gradients → apply gradients

Primitives

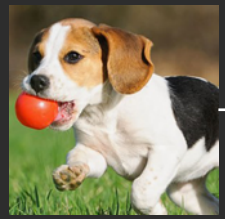
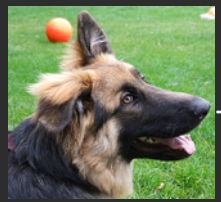
Rough shapes

Macro features

Pr{dog} = 0.903
Pr{cat} = 0.072
⋮

PURESTORAGE®

DISTRIBUTED TRAINING


evaluate
evaluate
evaluate

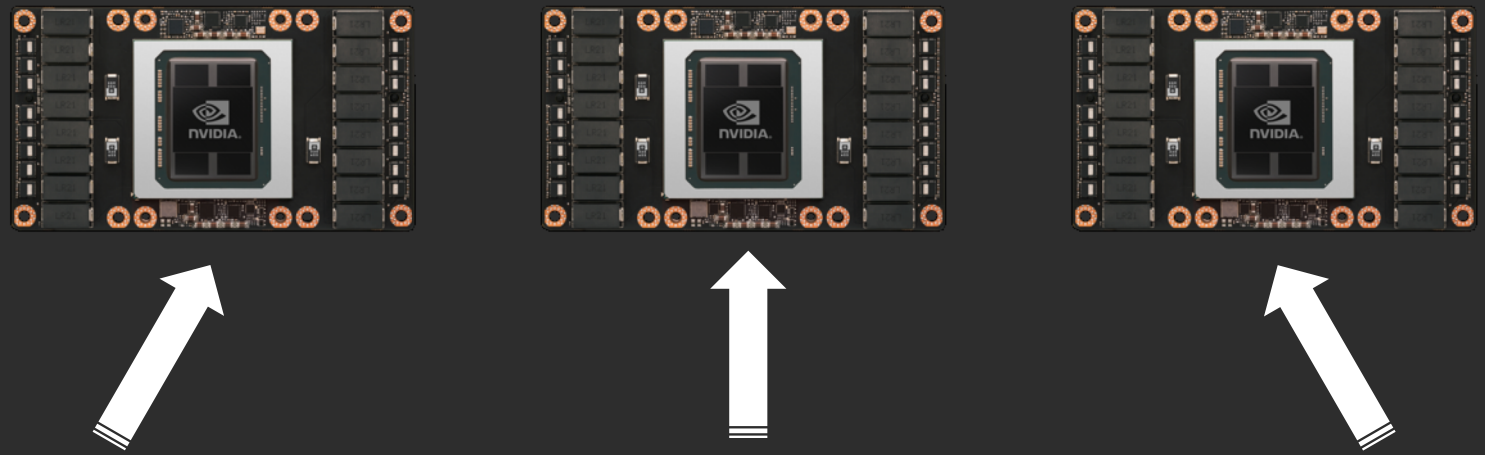
compute gradients
compute gradients
compute gradients

merge gradients

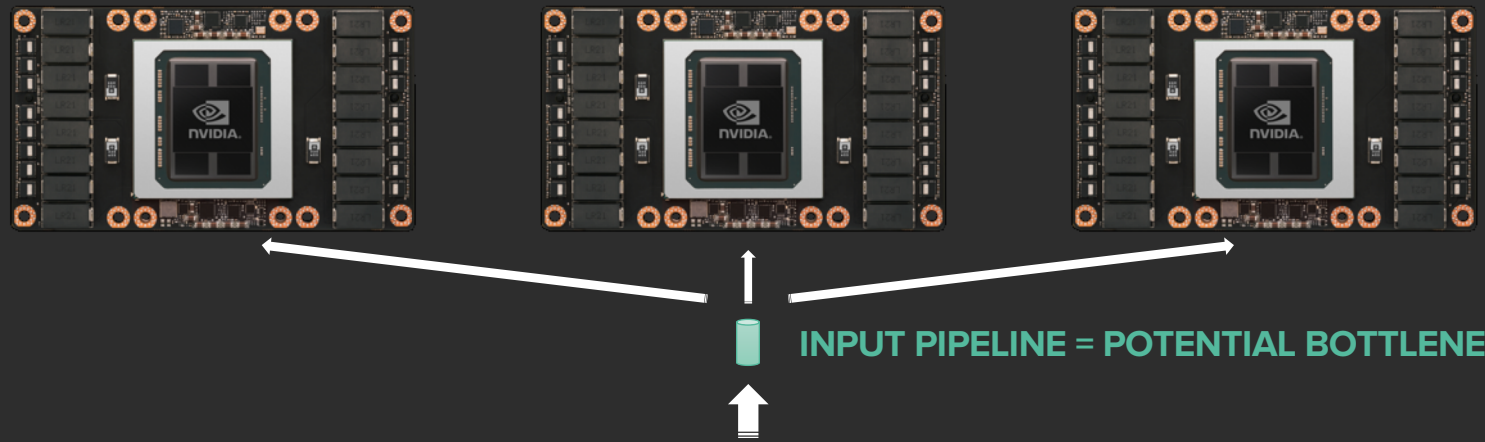
apply gradients
apply gradients
apply gradients

GPUs

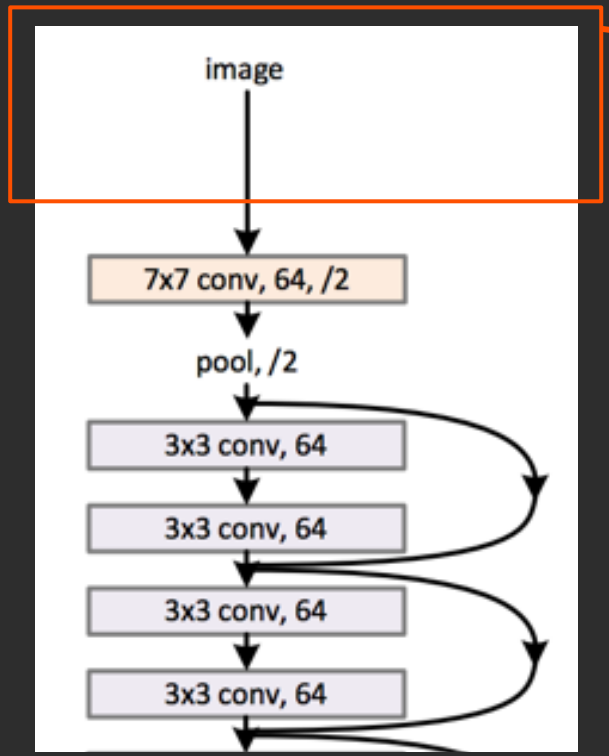
MORE, FASTER GPUs + MORE DATA

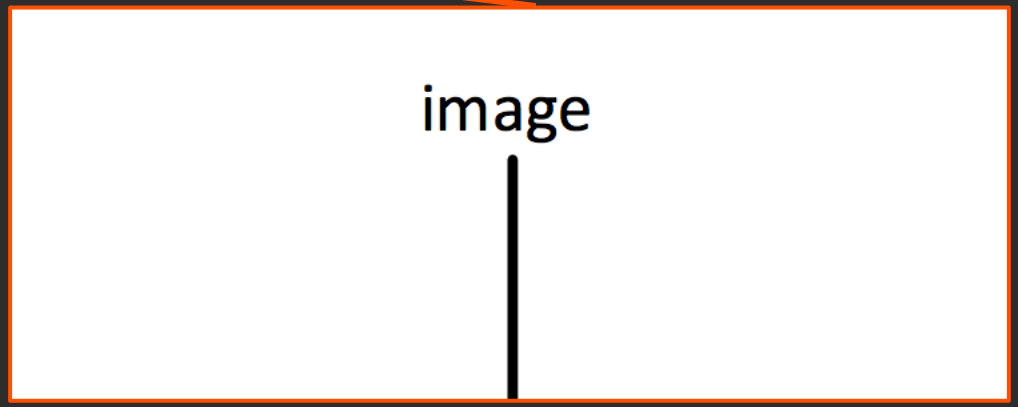


CAN WE KEEP GPUs FED WITH DATA?



INPUT PIPELINES



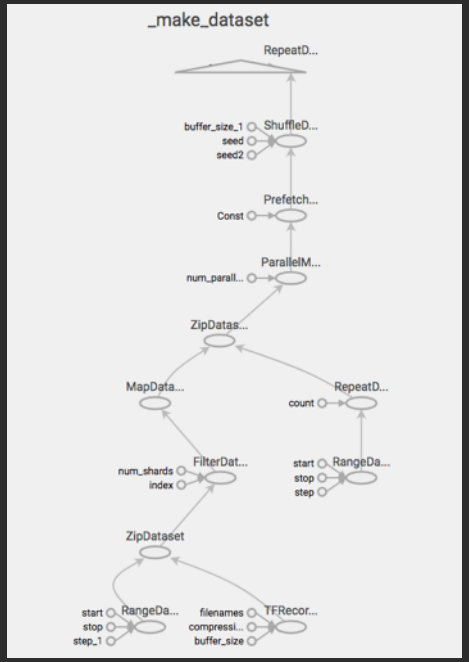


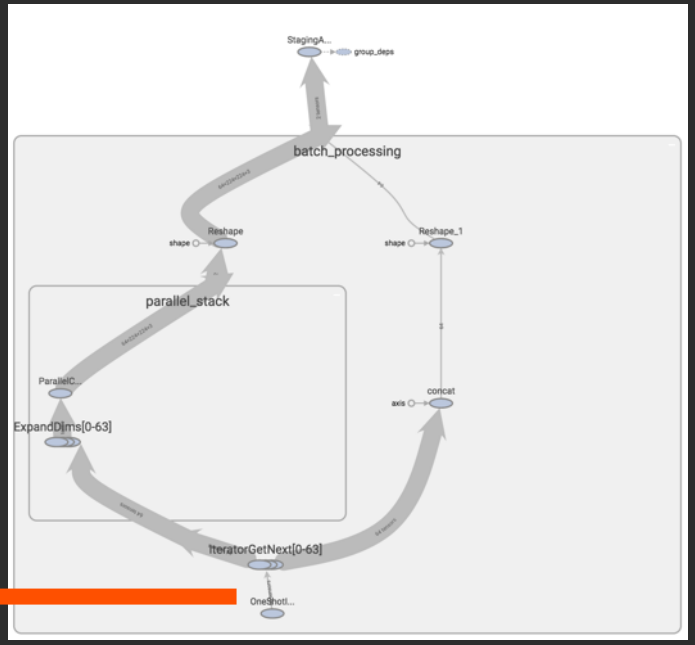
CAN IT BE THAT SIMPLE?

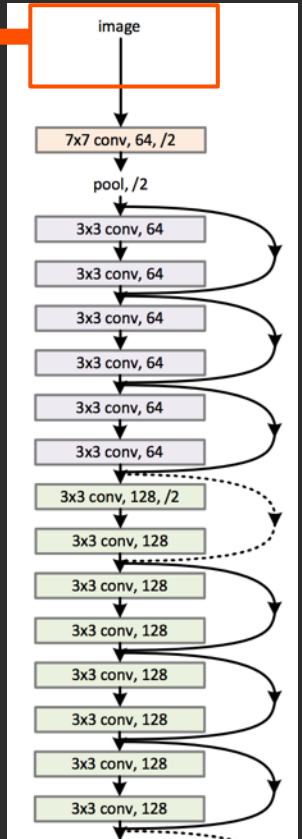
PURESTORAGE®

Source: K. He, X. Zhang, S. Ren, J. Sun, *Deep Residual Learning for Image Recognition*, CVPR 2015

REAL INPUT PIPELINES





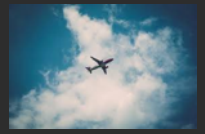


CAN YOU SPOT THE BOTTLENECK?

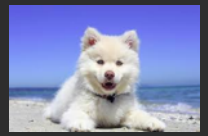
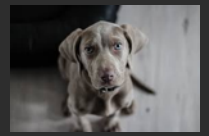
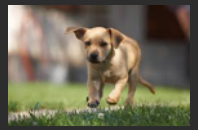
Source: K. He, X. Zhang, S. Ren, J. Sun, *Deep Residual Learning for Image Recognition*, CVPR 2015

FROM IMAGES TO TENSORS

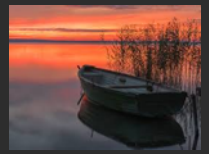
PLANE

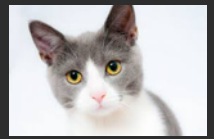
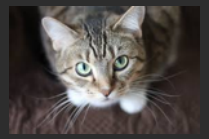
DOG

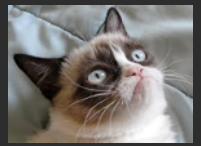





BOAT

CAT

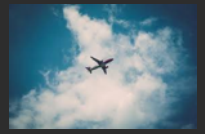


1. Enumerate

PURESTORAGE®

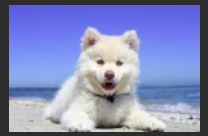
FROM IMAGES TO TENSORS

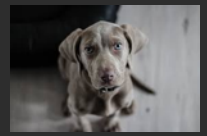

PLANE

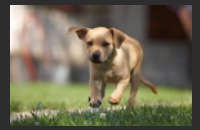

PLANE


PLANE


PLANE


DOG

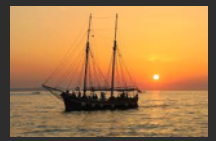

DOG


DOG

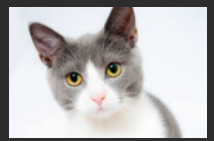

DOG


BOAT

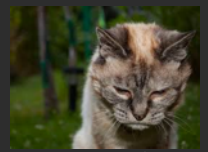

BOAT

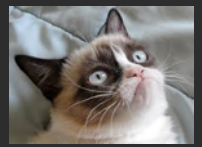

BOAT


BOAT


CAT


CAT


CAT

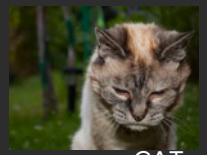

CAT

1. Enumerate
2. Associate labels

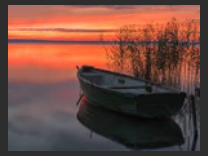
PURESTORAGE®

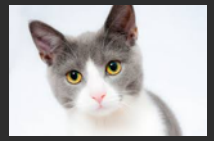
FROM IMAGES TO TENSORS


BOAT


CAT


BOAT


BOAT


CAT

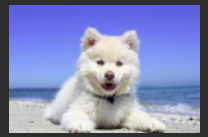

PLANE

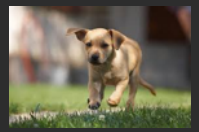

PLANE


CAT

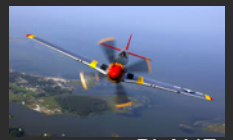

PLANE

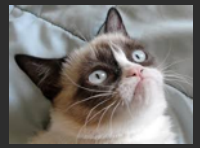

DOG

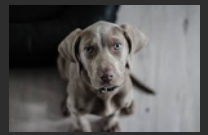

DOG


DOG


BOAT


PLANE


CAT


DOG

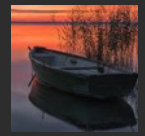
1. Enumerate
2. Associate labels
3. Shuffle

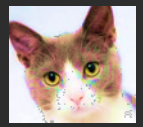
FROM IMAGES TO TENSORS


BOAT


CAT

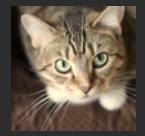

BOAT


BOAT

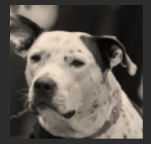

CAT


PLANE

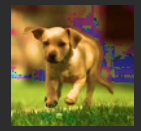

PLANE


CAT

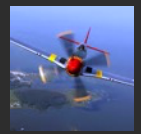

PLANE

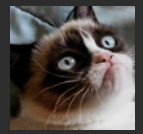

DOG

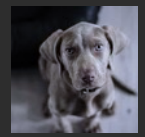

DOG


DOG


BOAT


PLANE


CAT


DOG

1. Enumerate
2. Associate labels
3. Shuffle
4. Read, crop, distort

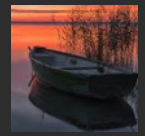
PURESTORAGE®

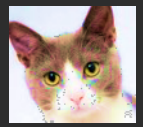
FROM IMAGES TO TENSORS


BOAT


CAT

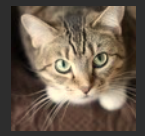

BOAT


BOAT

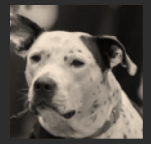

CAT


PLANE

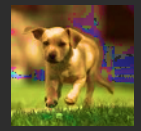

PLANE


CAT

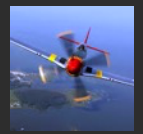

PLANE

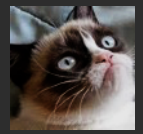

DOG

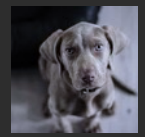

DOG


DOG

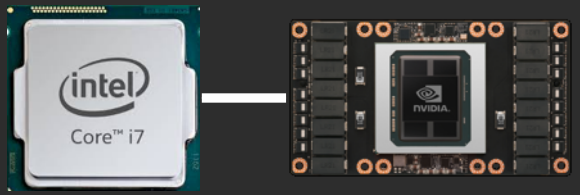

BOAT


PLANE


CAT


DOG

1. Enumerate
2. Associate labels
3. Shuffle
4. Read, crop, distort
5. Copy to GPU



FROM IMAGES TO TENSORS

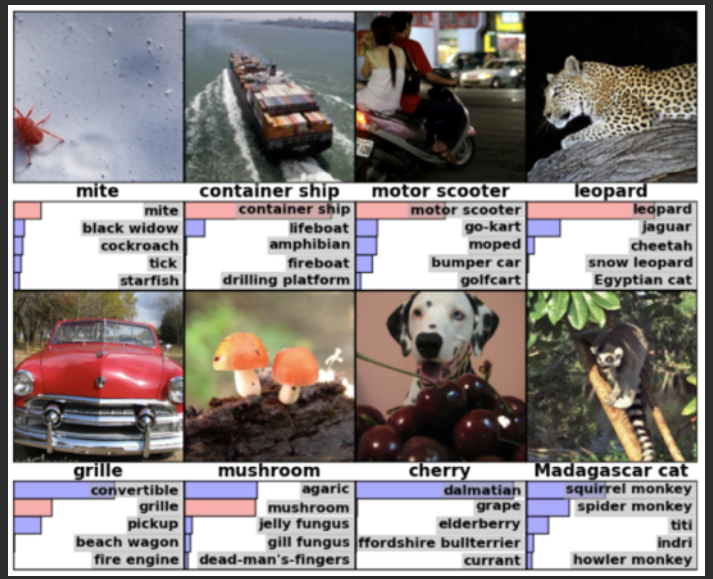
Other domains (NLP, speech, etc.)
will follow a similar(ish) flow

ANY OF THESE STEPS CAN BE A POTENTIAL BOTTLENECK

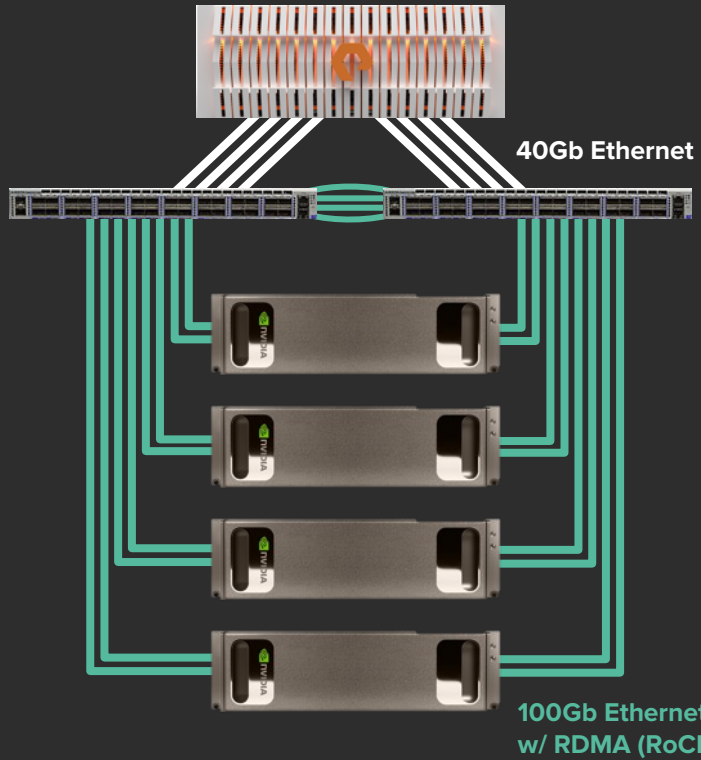
1. Enumerate
2. Associate labels
3. Shuffle
4. Read, crop, distort
5. Copy to GPU

EVALUATION METHODOLOGY

1.3M images, 1000 categories



HARDWARE STACK



AIRI

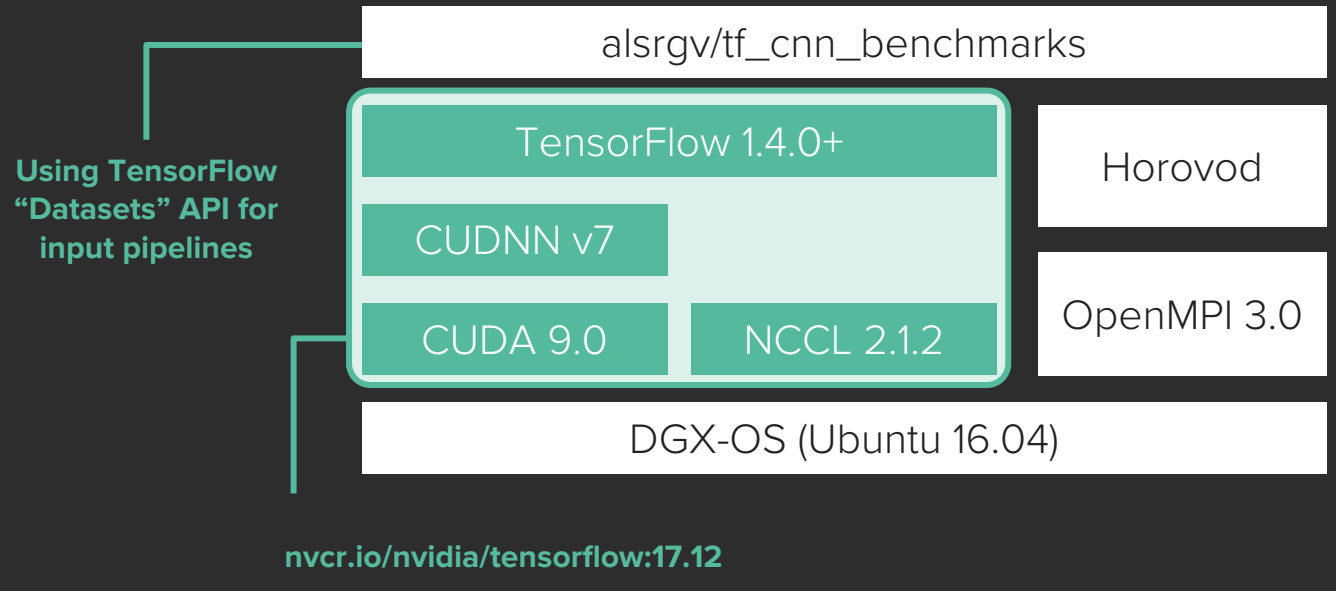
Pure Storage FlashBlade: 15x17TB
179T usable before data reduction

Arista DCS-7060CX2-32S
32x 100Gb/s QSFP100 ports

4x NVIDIA DGX-1, each with
8x Tesla V100 GPUs (SXM2)
2x Intel E5-2698 v4 @ 2.20GHz
4x Mellanox MT27700 100Gb/s VPI adapters
512GB DDR4-2400

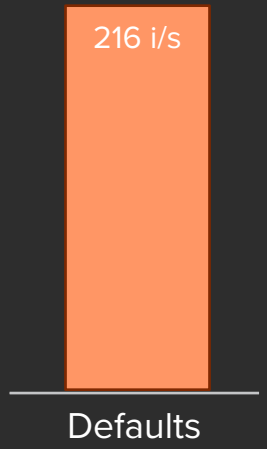
PURESTORAGE®

SOFTWARE STACK

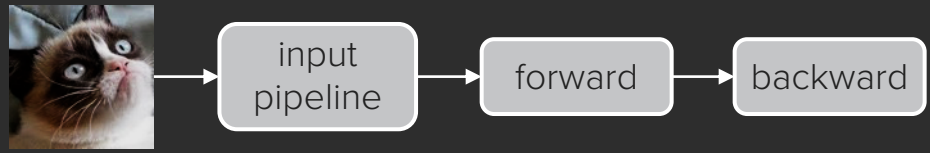


TRAINING WITH 1 GPU

Images per second when training Inception3 (batch size = 64)



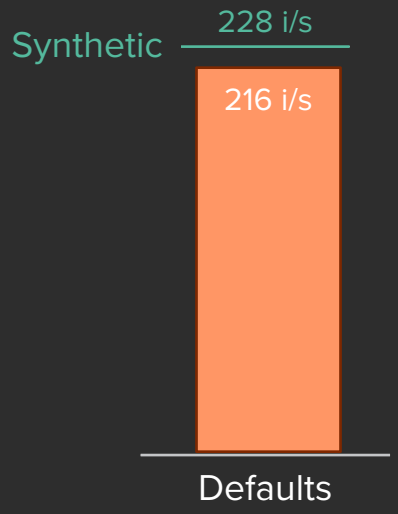
"Default" training pipeline



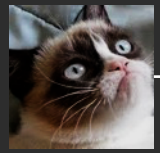
How do we know what good looks like?

TRAINING WITH 1 GPU

Images per second when training Inception3 (batch size = 64)

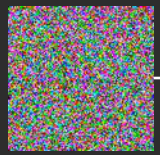


"Default" training pipeline



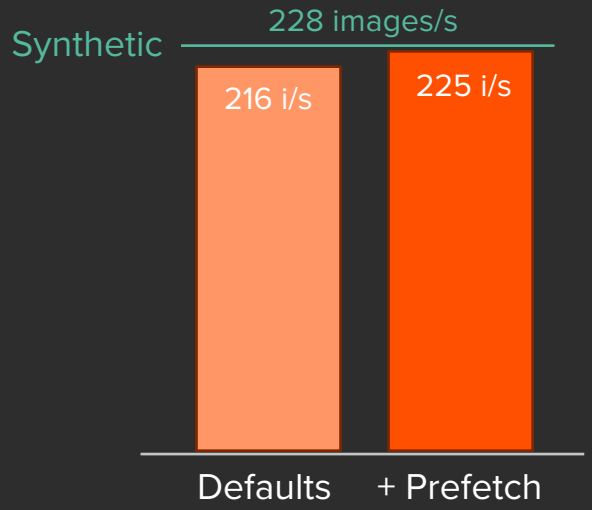
How do we know what good looks like?

Replace the input pipeline with synthetic data

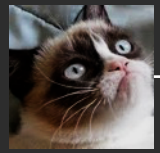


TRAINING WITH 1 GPU

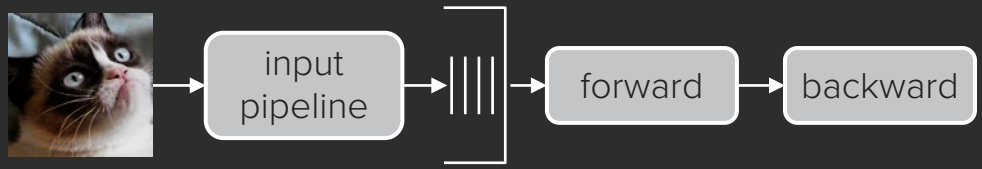
Images per second when training Inception3 (batch size = 64)



"Default" training pipeline



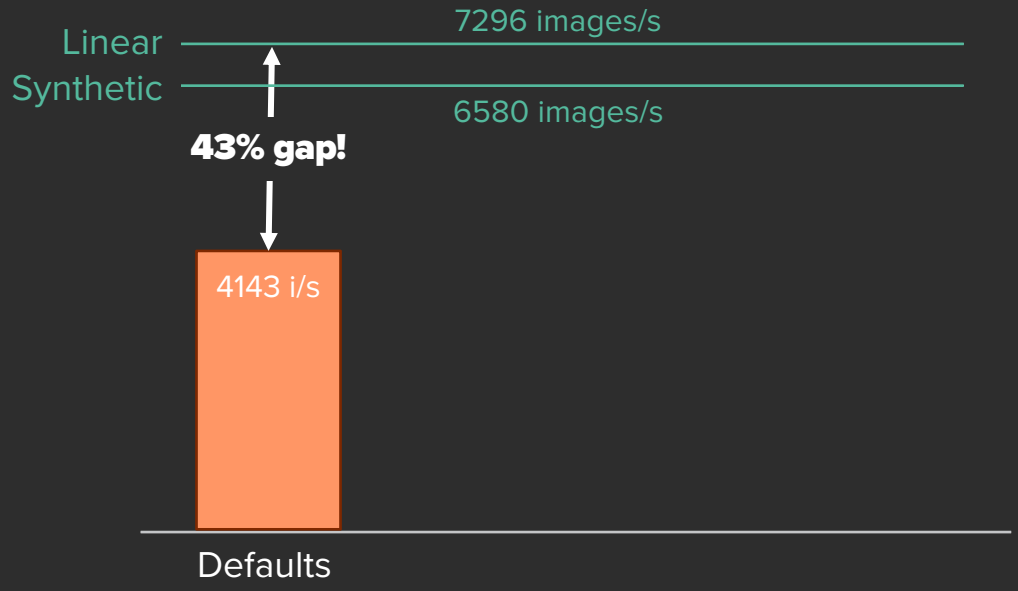
Adding a prefetch queue improves scheduler behavior



SHOULD WE CARE ABOUT ~1.3%?

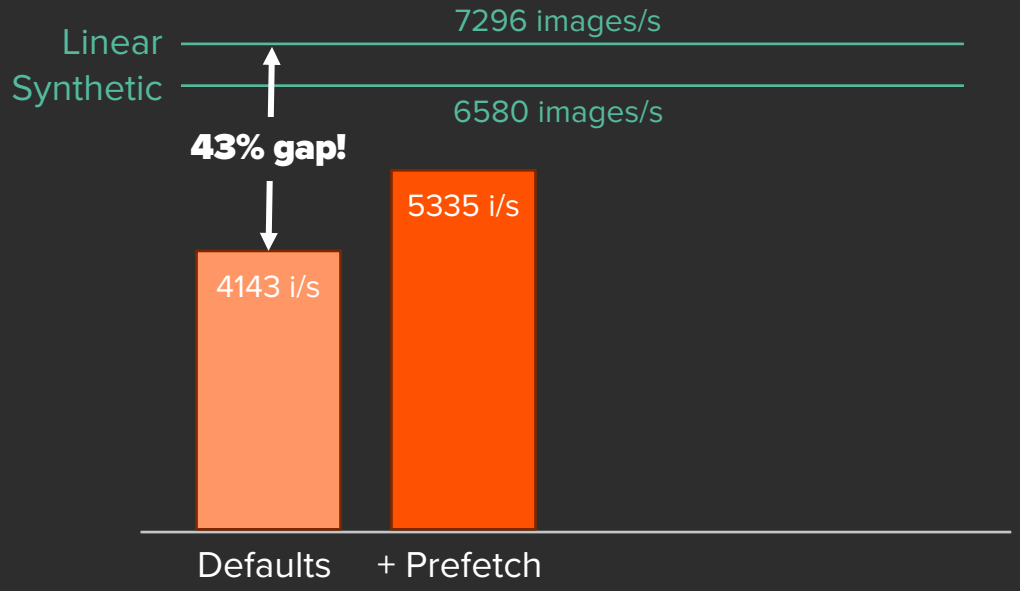
SCALING TO 32 GPUs (4x DGX-1s)

Images per second when training Inception3 (batch size = 64/GPU)



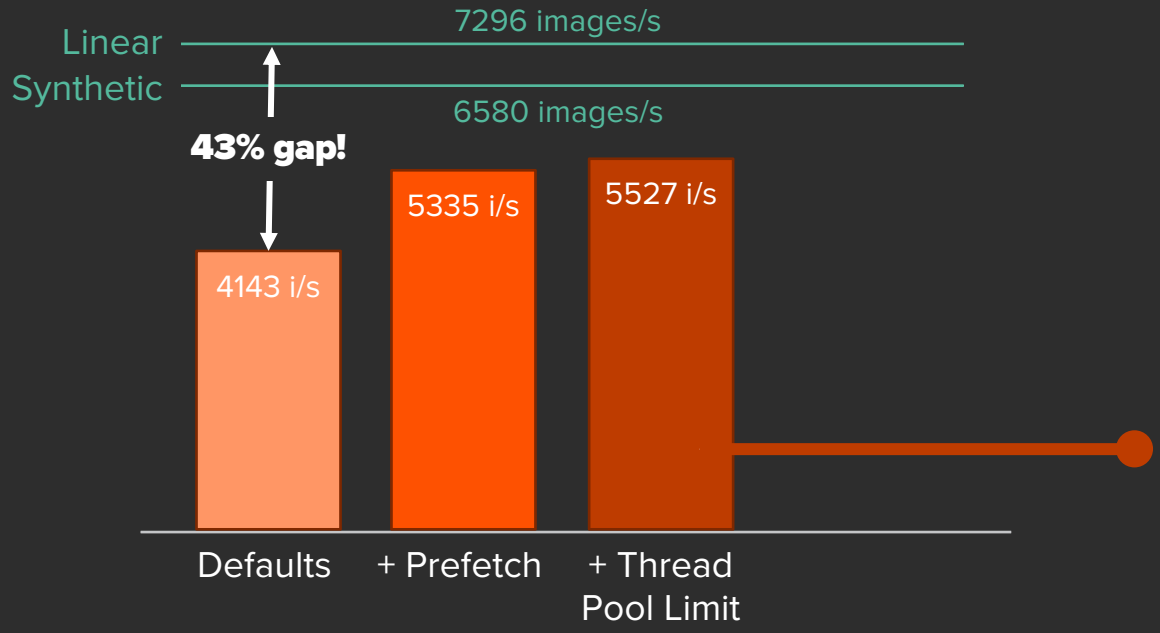
SCALING TO 32 GPUs (4x DGX-1s)

Images per second when training Inception3 (batch size = 64/GPU)



SCALING TO 32 GPUs (4x DGX-1s)

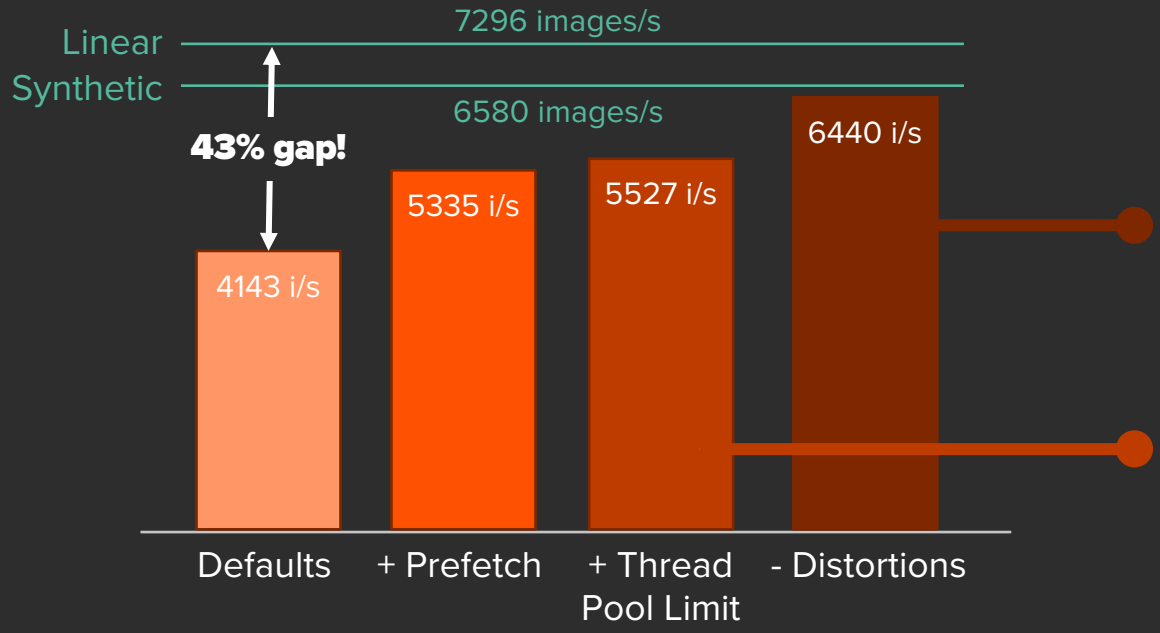
Images per second when training Inception3 (batch size = 64/GPU)



Thread pool limits: Avoid over-subscribing CPU with too many threads.
`(inter_op_parallelism_threads)`

SCALING TO 32 GPUs (4x DGX-1s)

Images per second when training Inception3 (batch size = 64/GPU)

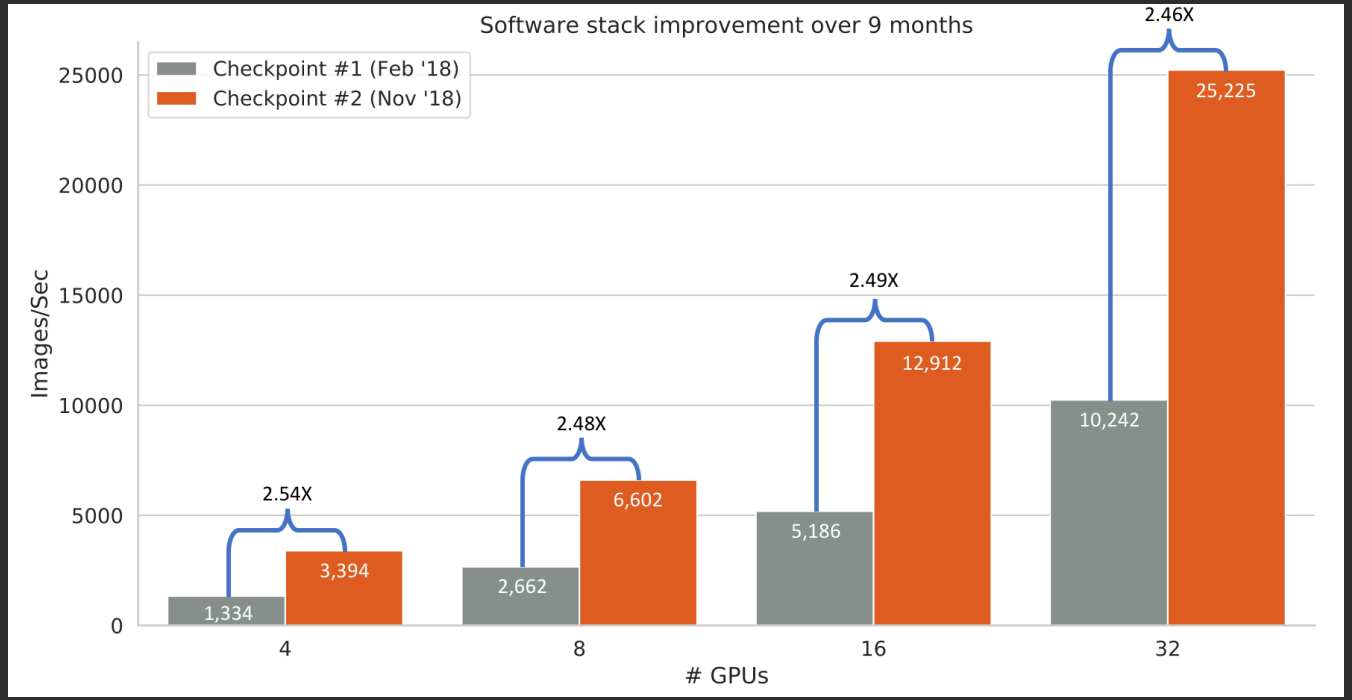


No Distortions: Skip preprocessing step from input pipeline. This is an unrealistic configuration, but it shows the bottleneck.

Thread pool limits: Avoid over-subscribing CPU with too many threads. `(inter_op_parallelism_threads)`

EXCELLENT SCALABILITY, BUT STILL MORE WORK TO BE DONE

2.5X Performance Improvement



PURESTORAGE®