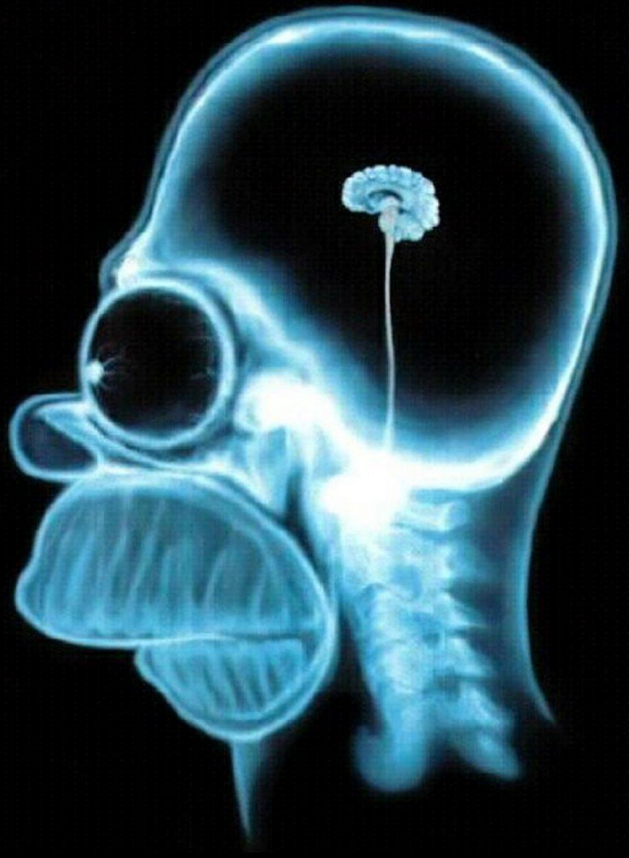
Transparent parallelization of neural network training

Cyprien Noel

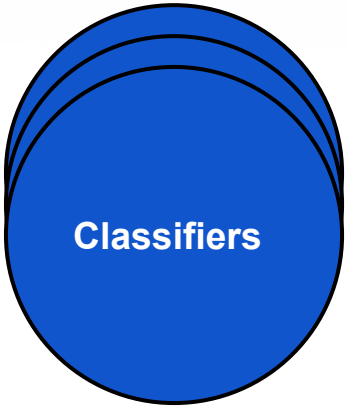
Flickr / Yahoo - GTC 2015



Outline

- **Neural Nets at Flickr**
- Training Fast
  - Parallel
  - Distributed
- Q&A

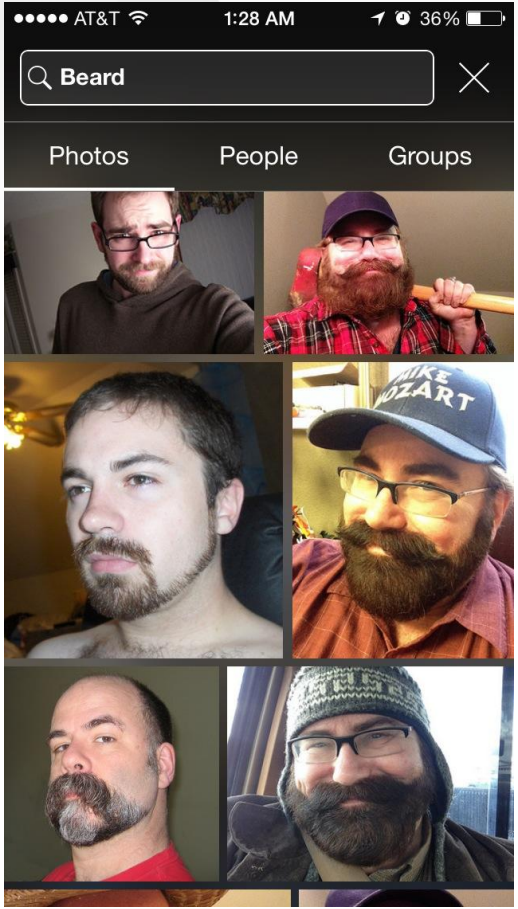
Tagging Photos



Class	Probability
Flowers	0.98
Outdoors	0.95
Cat	0.001
Grass	0.6

Any photo on Flickr is classified using computer vision

flickr YAHOO!



Auto Tags Feeding Search

- Flowers
- Buildings
- Outdoors
- Beach
- Beards
- and more!

flickr YAHOO!

Tagging the Flickr corpus

- Classify millions of new photos per day
- Apply new models to billions of photos
- Train new models using Caffe

flickr YAHOO!

Training new models

- Manual experimentation
- Hyperparameter search
- Limitation is training time

→ **Parallelize Caffe**

flickr YAHOO!

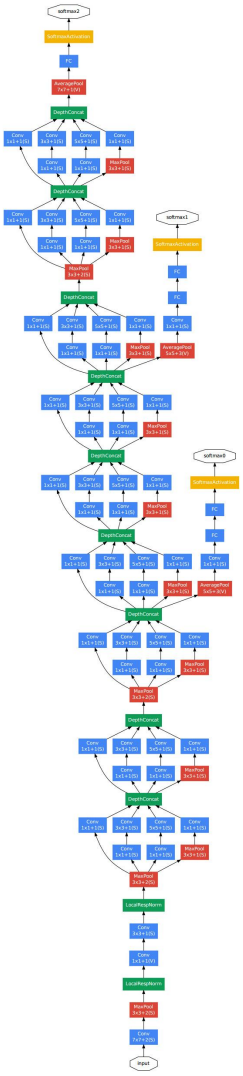
Goals

- “Transparent”
  - Code Isolation
  - Existing Models
  - Globally connected layers
- Existing Infrastructure

flickr YAHOO!

Outline

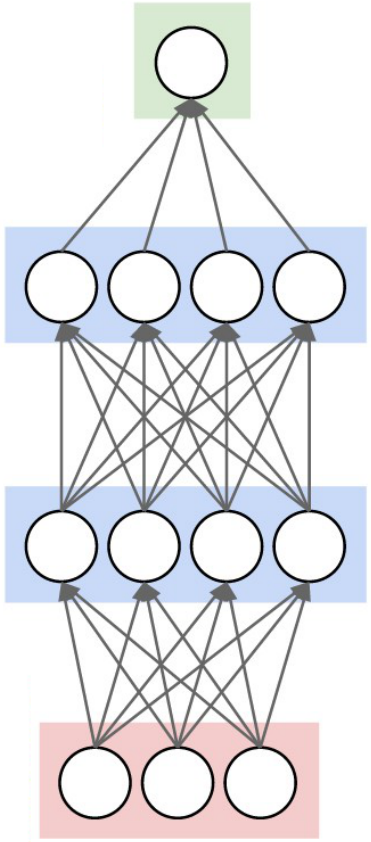
- Neural Nets at Flickr
- **Training Fast**
  - Parallel
  - Distributed
- Q&A



GoogLeNet, 2014

flickr YAHOO!

Ways to Parallelize



- Model
  - Caffe team enabling this now
- Data
  - Synchronous
  - Asynchronous

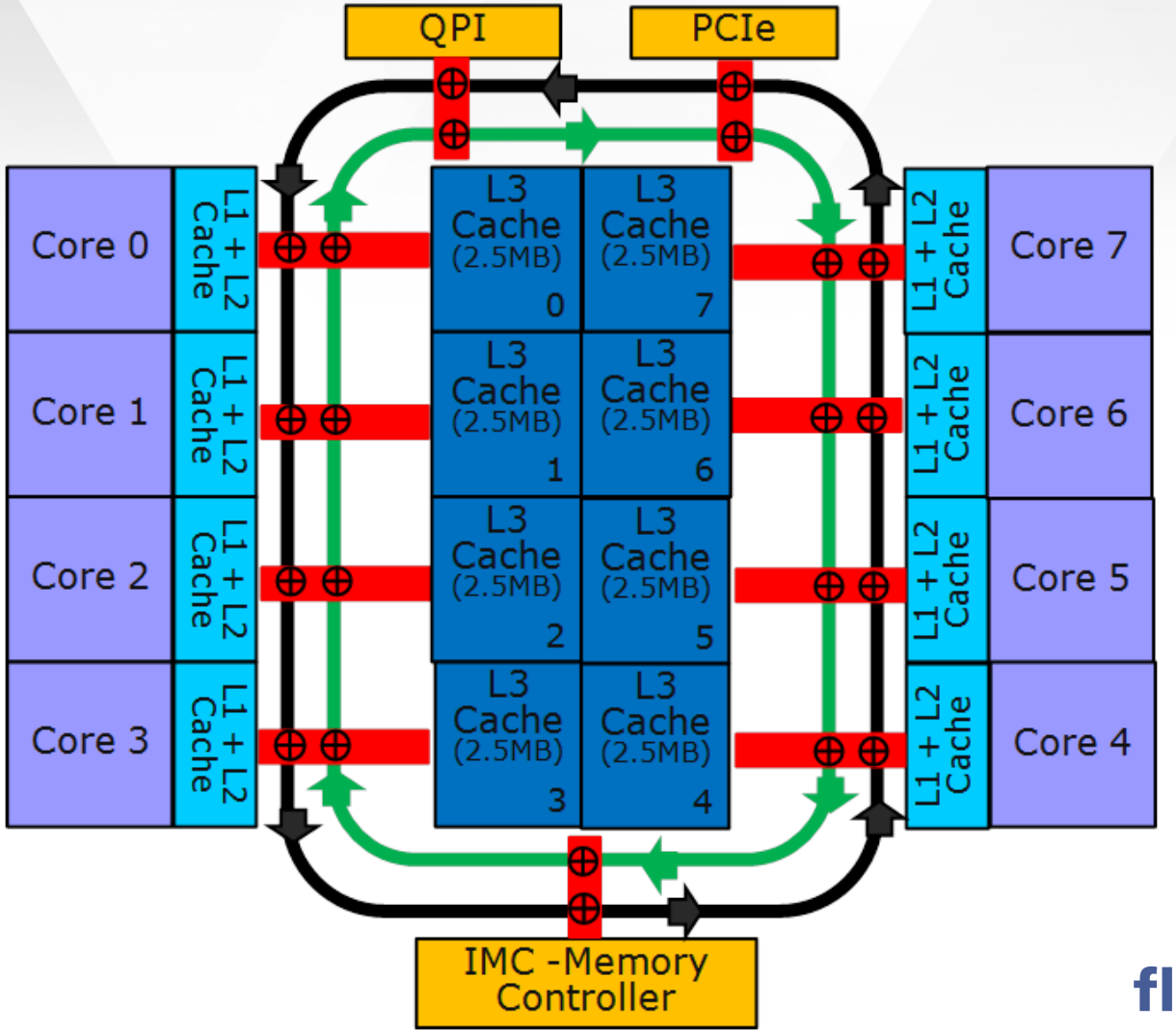
flickr YAHOO!

Outline

- Neural Nets at Flickr
- Training Faster
  - **Parallel**
  - Distributed
- Q&A

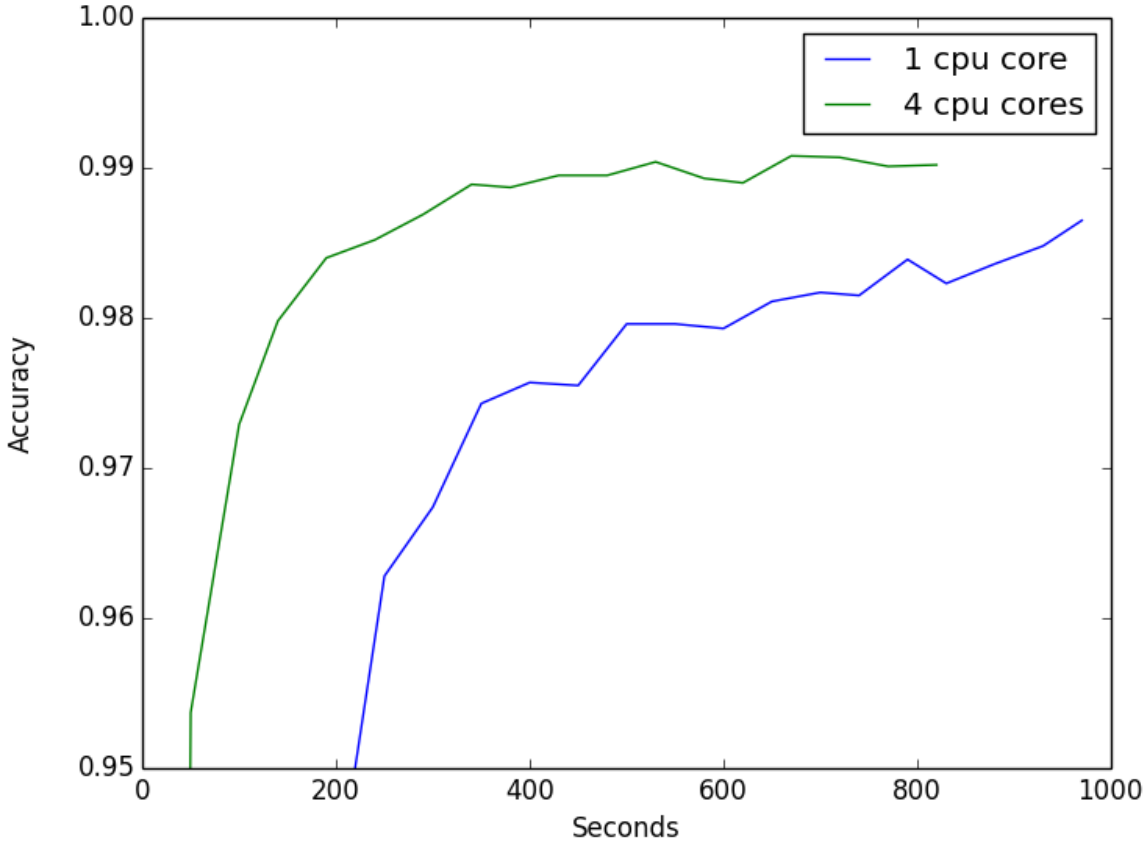
First Approach: CPU

- Hogwild! (2011)
- Cores read and write from shared buffer
- No synchronization
- Data races surprisingly low



flickr YAHOO!

MNIST CPU



flickr YAHOO!

Hogwild

- Plateaus with core counts
- Some Potential
  - On a grid
  - With model parallelism

But we are at GTC

flickr YAHOO!

GPU Cluster

- A lot of time spent preparing experiments
- Code Deployment
  - 

- Data Handling
  - On the fly datasets for “big data”

flickr YAHOO!

Outline

- Neural Nets at Flickr
- Training Fast
  - Parallel
  - **Distributed**
- Q&A

Second Approach: Lots of Boxes

flickr YAHOO!

Second Approach: Lots of Boxes

- Exchange gradients between nodes
- Parameter server setup
- Easy: move data fast

GPU memory - PCI - Ethernet

flickr YAHOO!

Second Approach: Lots of Boxes

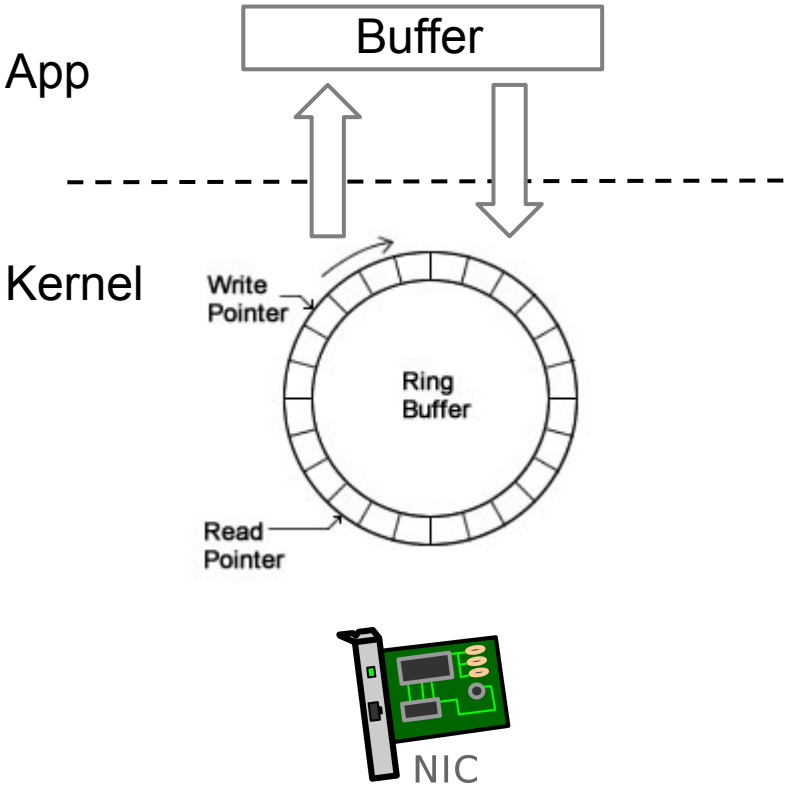
- 230MB * 2 * N per batch
- TCP/UDP chokes
  - Machines unreachable
- No InfiniBand or RoCE

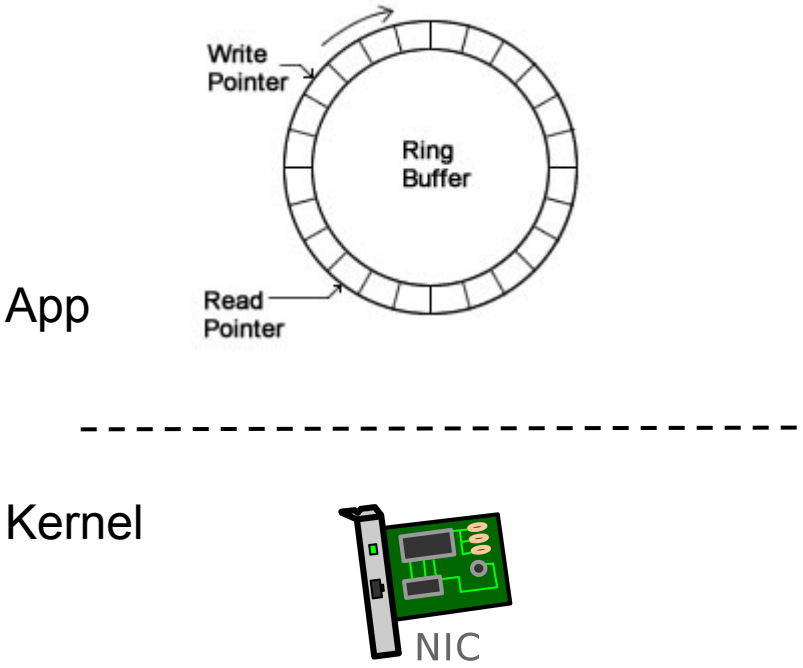
Second Approach: Lots of Boxes

- Modify Caffe: chunk parameters

flickr YAHOO!

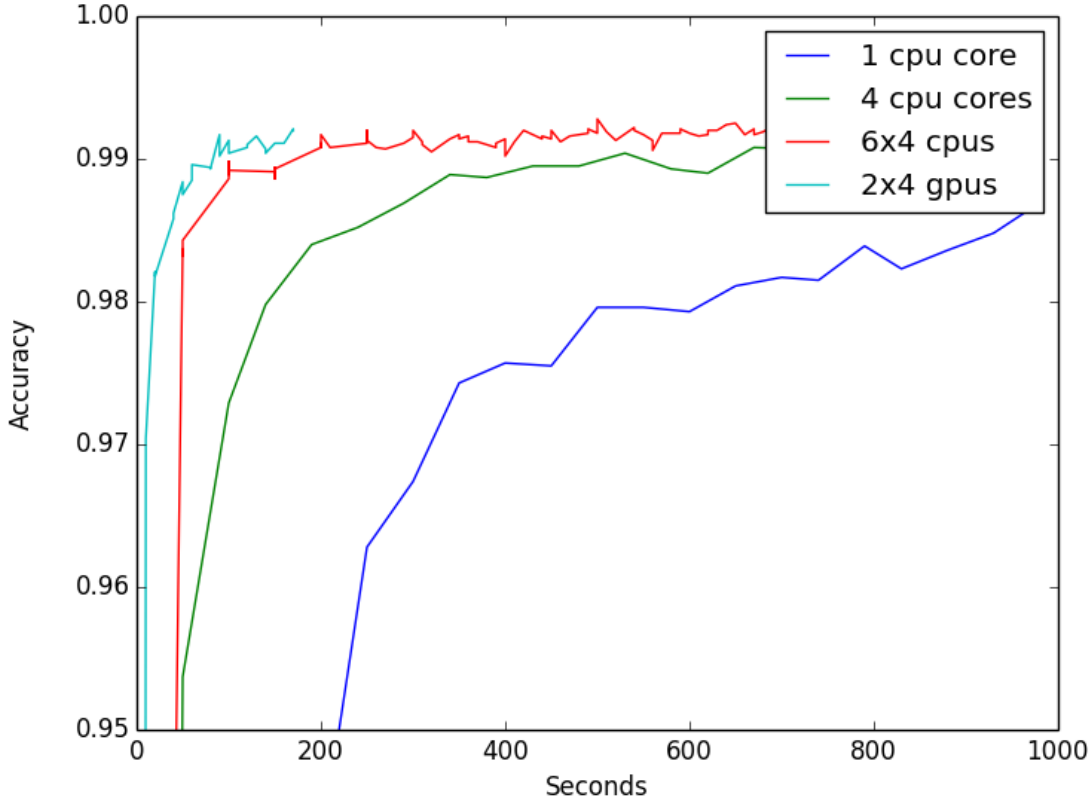
Packet_mmap





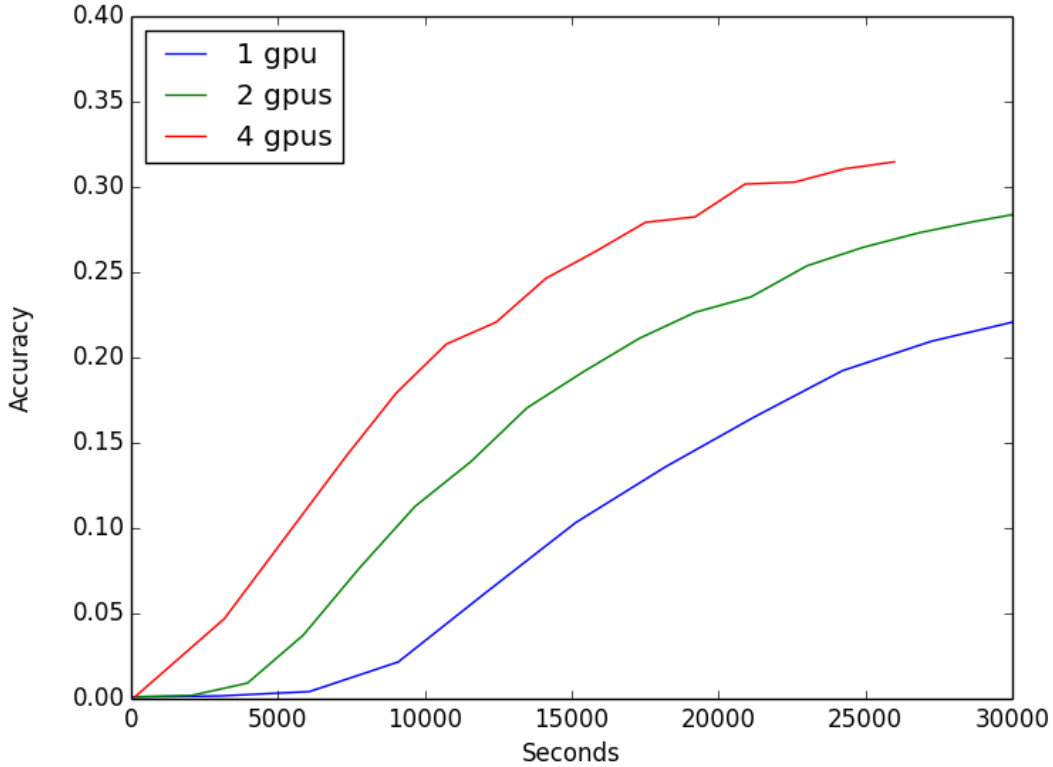
flickr YAHOO!

MNIST



flickr YAHOO!

ImageNet



flickr YAHOO!

NVIDIA

- Large Machines
  - 4 or 8 GPUs
  - Root PCI switches
- InfiniBand

flickr YAHOO!

Third Approach: CUDA P2P

- GPUs on single machine
- Data Feeding
  - Caffe Pipeline
  - Async Streams

State of Things

- Async ~8x but no momentum
- Sync ~2x
- Combining both, and model parallelism
- Working on auto tuning of params (batch, rate)
- Different ratios of compute vs. IO

Takeaway

- Check Caffe, including Flickr's contributions
- CUDA + Docker = Love
- Small SOC servers might be interesting for ML

Thanks!

Flickr vision team
Flickr backend team
Yahoo labs

cypof@yahoo-inc.com