Machine Learning on FPGAs

Jason Cong

Chancellor's Professor, UCLA

Director, Center for Domain-Specific Computing

cong@cs.ucla.edu

http://cadlab.cs.ucla.edu/~cong

Impacts of deep learning for many applications

Unmanned Vehicle

Speech & Audio

Text & Language

WIKIPEDIA
The Free Encyclopedia

REUTERS

AP Associated Press

Genomics

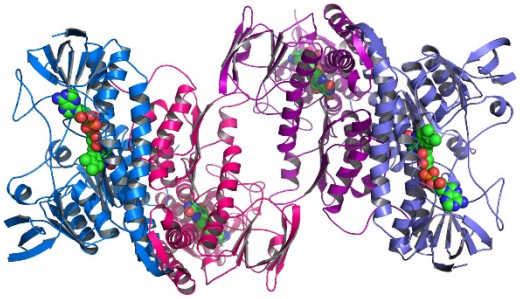


Image & Video

flickr

Google

YouTube

Multi-Media

All images are from internet search

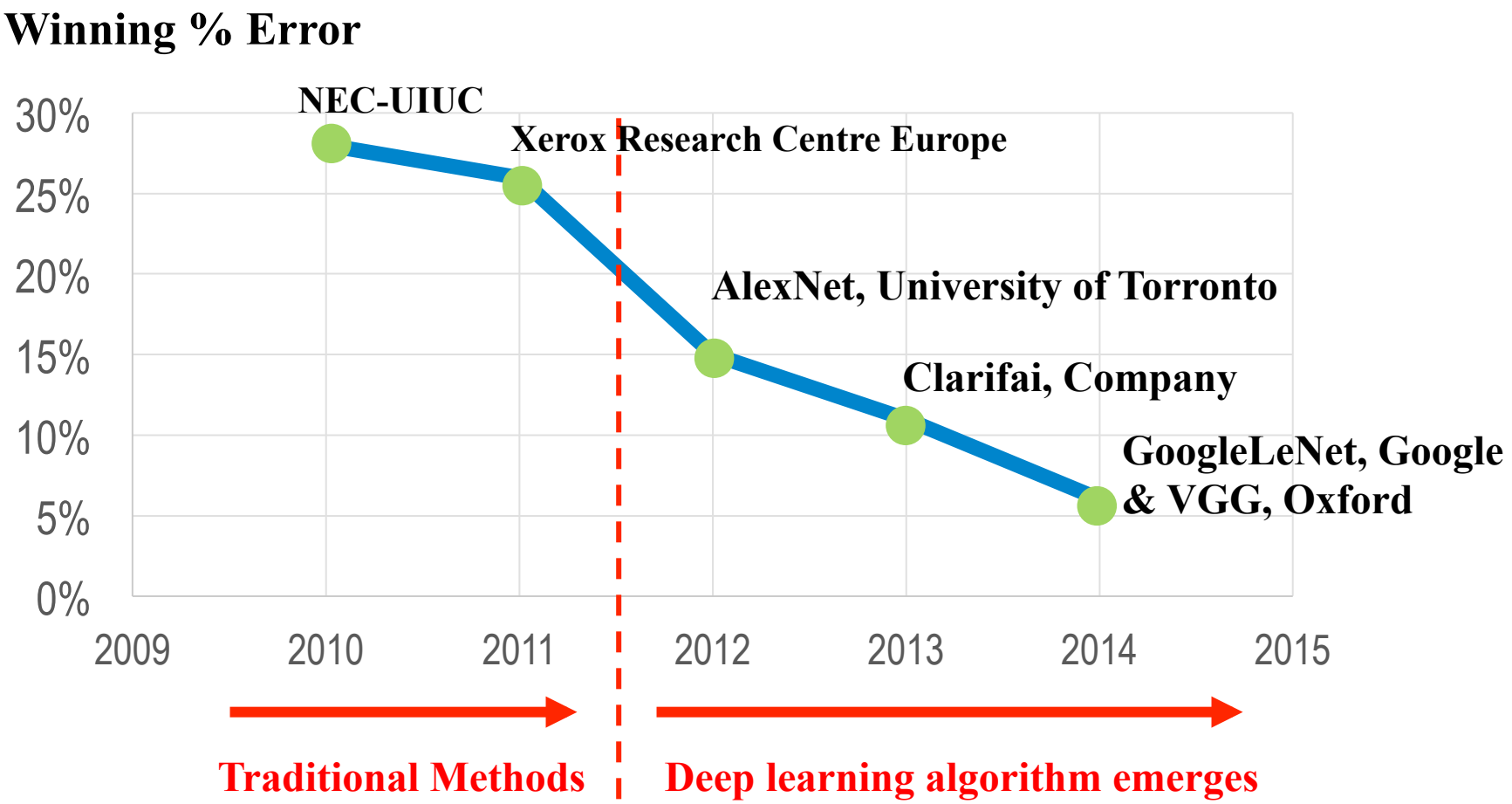
ImageNet Competition

- 1,200,000 Training Images
  - With 50,000 Validation & 100,000 Test Images
- 1000 Category of objects

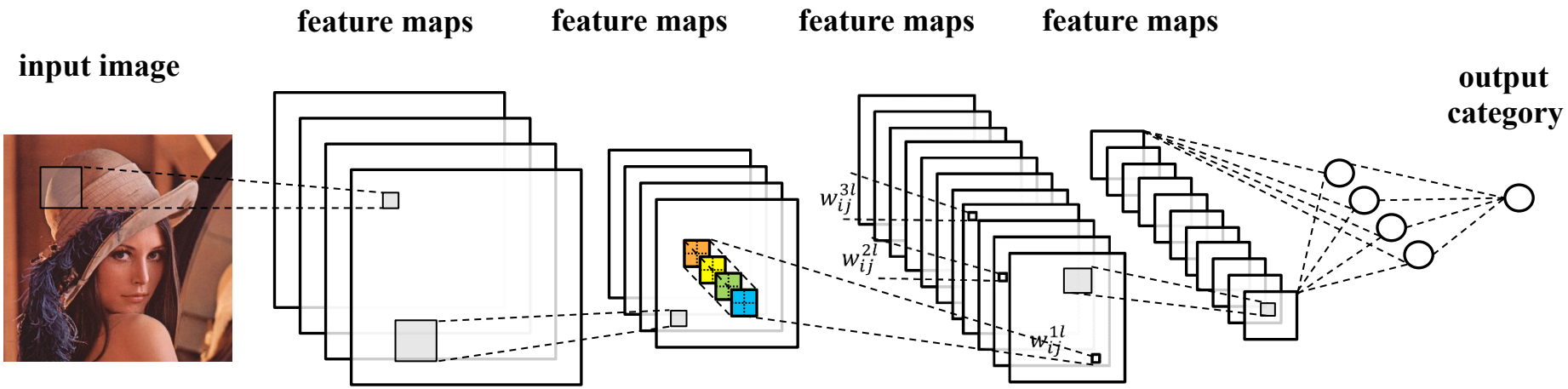
[1]

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. NIPS2012

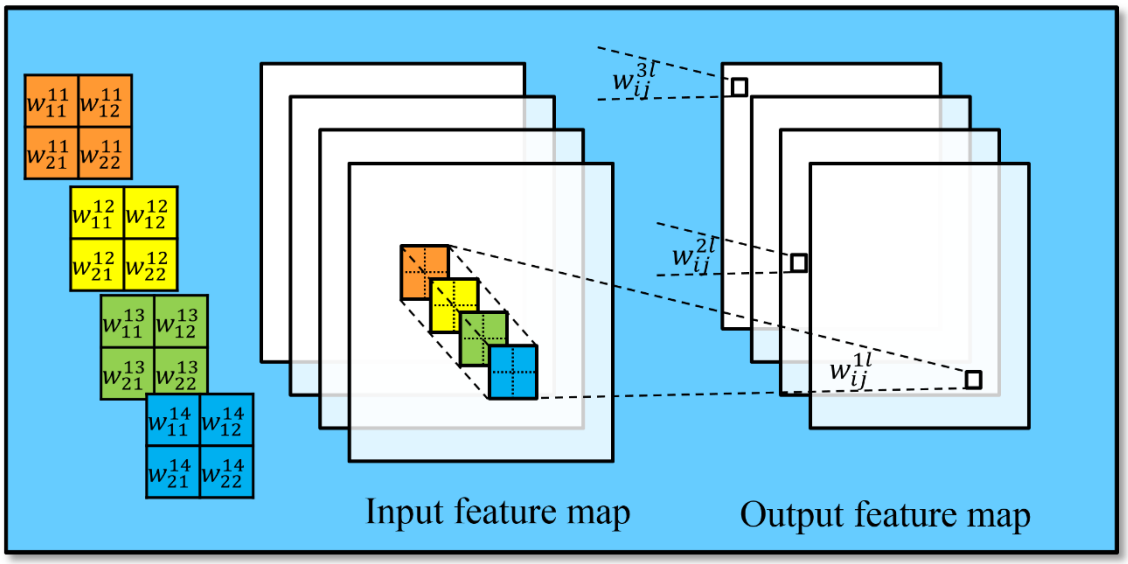
ImageNet Competition Results

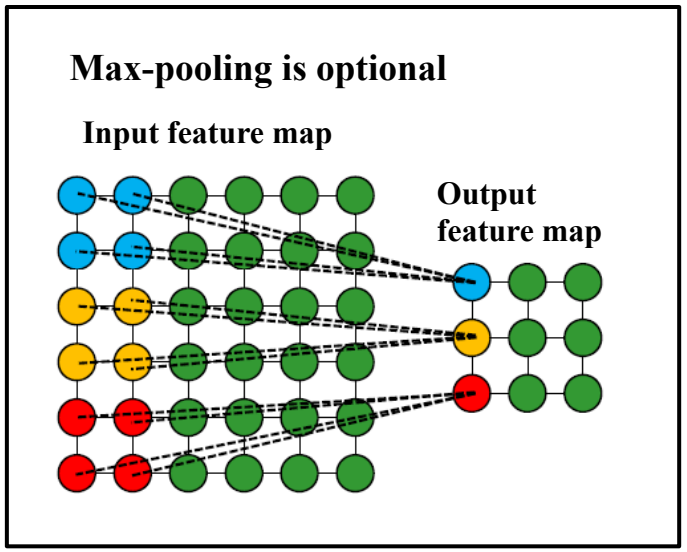


Convolutional Neural Network (CNN)

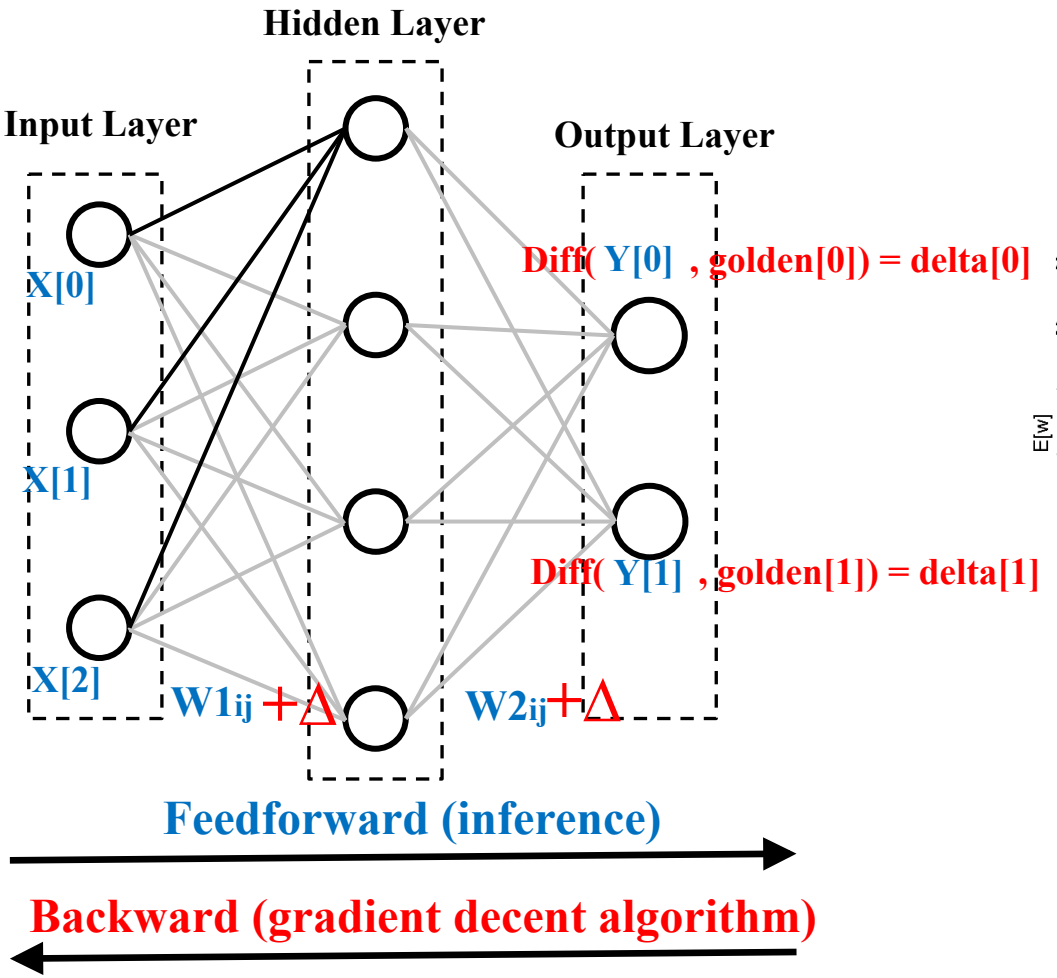


Inference: A feedforward computation

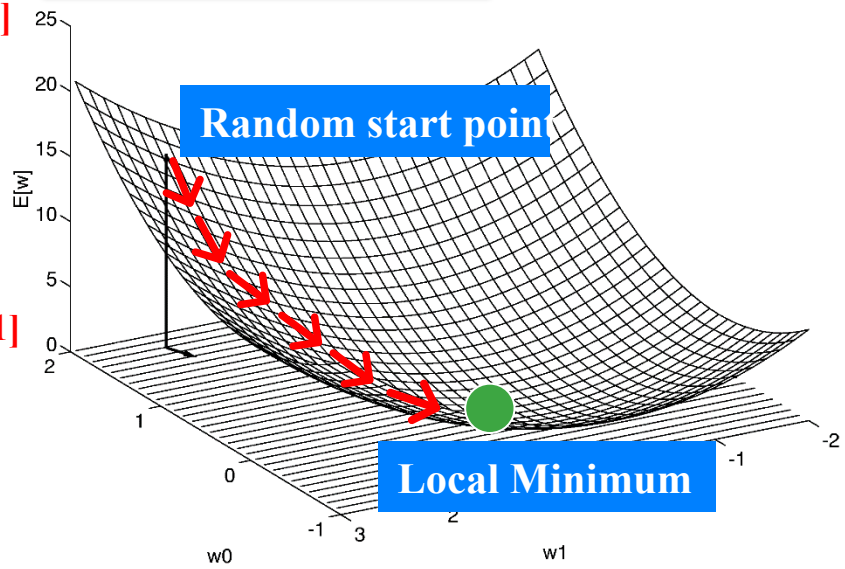




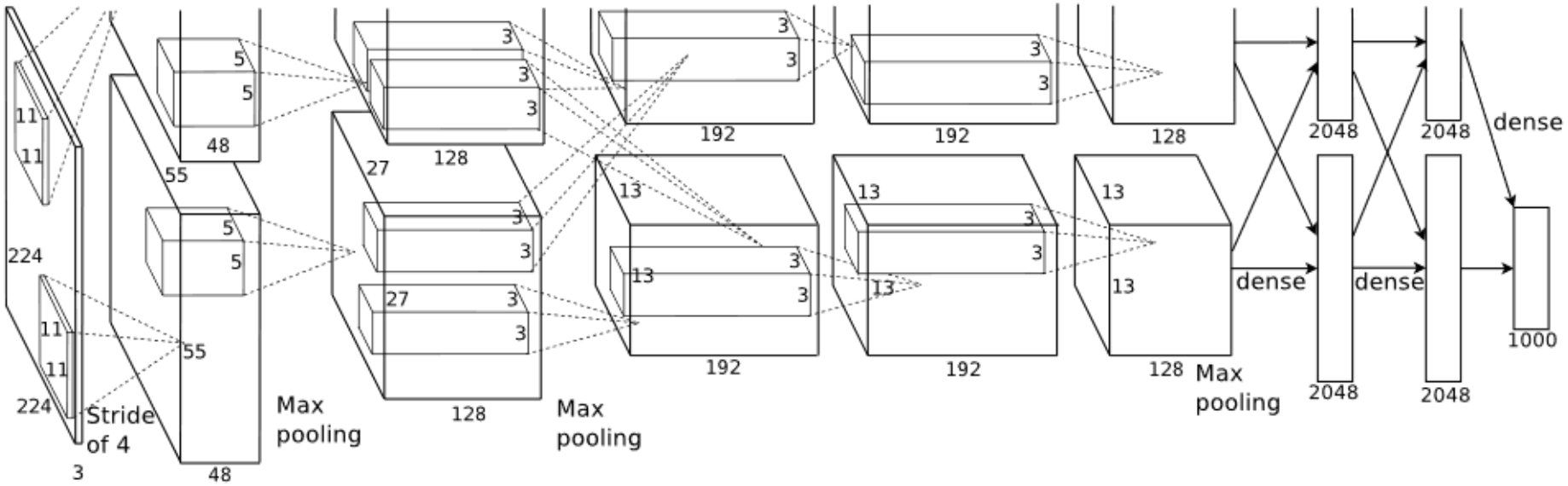
Backward propagation



Optimization target:
Min. Inference error rate



Real-life CNNs



AlexNet [1] : Winner of imagenet 2012 classification task

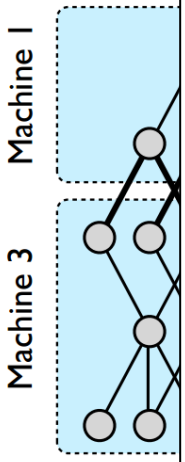
Real-life CNNs	Neurons	layers	Parameter
AlexNet	650, 000	8	60 Million
VGG16	14,000,000	16	140 Million
GoogleNet	8,300,000	22	4 Million

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems*. 2012.

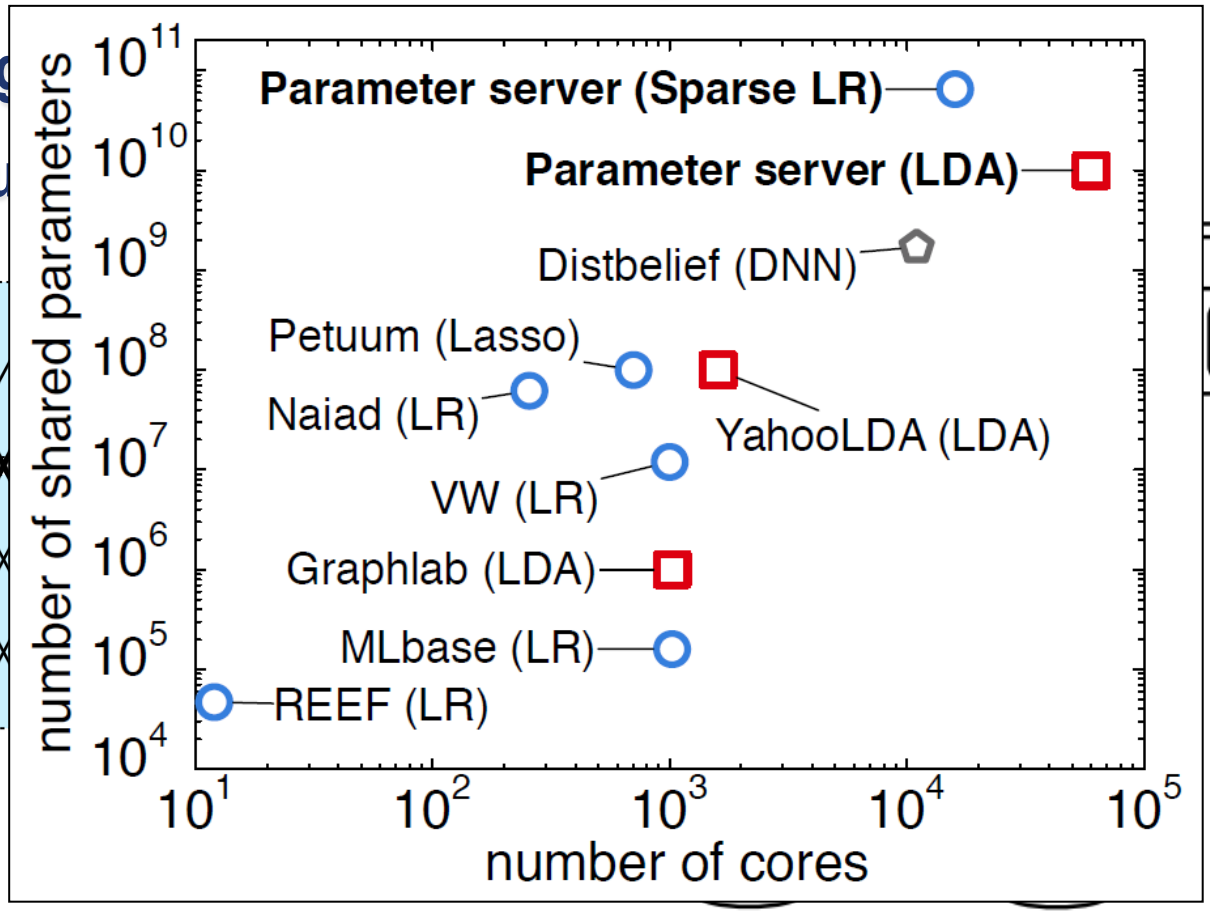
Distributed Deep Learning System

- **Distributed Machine Learning**
  - Goog
  - A clu

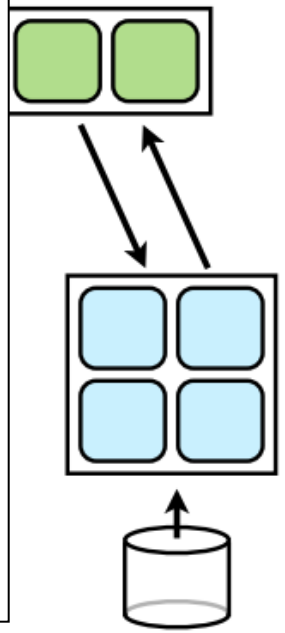
[2]



[3]



$= w - \eta\Delta w$



[2] Dean, Jeffrey, et al. "Large scale distributed deep networks." *Advances in Neural Information Processing Systems*. 2012.
[3] Li, Mu, et al. "Scaling distributed machine learning with the parameter server." *Proc. OSDI*. 2014.

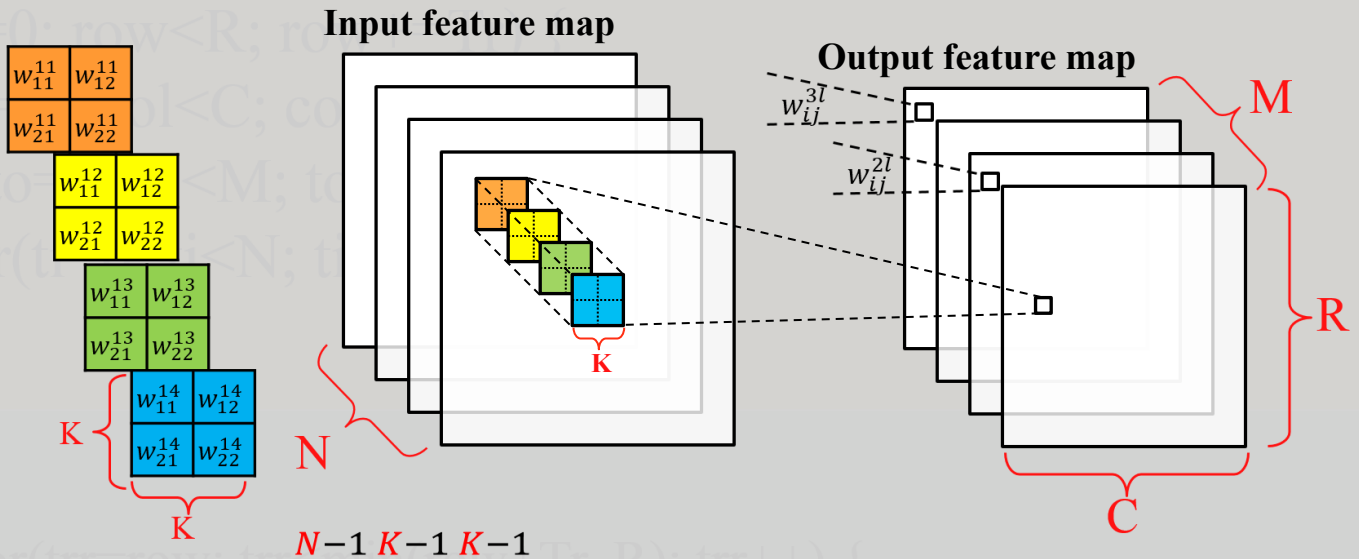
An Example of High-Performance GPU Cluster [NIPS'13]

- **Deep learning with COTS HPC systems**
  - Stanford University
  - A cluster of 12 GPUs
- **High performance**
  - Train 1 billion parameter network in a couple of days
  - Comparable to CPU cluster of 1000 machines
- **Cost Effective**
  - $20,000
  - CPU cluster with comparable performance cost $1 Million

FPGA acceleration of feedforward phase

- In many applications, neural network is trained in back-end CPU or GPU clusters
- FPGA: very suitable for latency-sensitive real-time inference job
  - Unmanned vehicle
  - Speech Recognition
  - Audio Surveillance
  - Multi-media
- Related Work
  - [LeCun'09] [Farabet'10] [Aysegui'13] [Gokhale'15] [Zhang'15], etc.

Inference (or feedforward computation)



$$Y[m][r][c] = \sum_{n=0}^{N-1} \sum_{i=0}^{K-1} \sum_{j=0}^{K-1} W[m][n][i][j] \times X[n][S \times r + i][S \times c + j]$$

```
1  for(row=0; row<R; row++) {
2    for(col=0; col<C; col++) {
3      for(to=0; to<M; to++) {
4        for(ti=0; ti<N; ti++) {
5          for(i=0; i<K; i++) {
6            for(j=0; j<K; j++) {
               output_fm[to][row][col] +=
               weights[to][ti][i][j]*input_fm[ti][S*row+i][S*col+j];
}}}}}}
```

R, C, M, N, K, S are all configuration parameters of the convolutional layer

Feedforward computation on FPGA

```
1  for(row=0; row<R; row+=Tr) {                                   (Tile loop)
2    for(col=0; col<C; col+=Tc) {                                 (Tile loop)
3      for(to=0; to<M; to+=Tm) {                                  (Tile loop)
4        for(ti=0; ti<N; ti+=Tn) {                                (Tile loop)
```

Off-chip Data Transfer: Memory Access Optimization

On-chip Data: Computation Optimization

```
5          for(trr=row; trr<min(row+Tr, R); trr++) {              (Point loop)
6            for(tcc=col; tcc<min(tcc+Tc, C); tcc++) {            (Point loop)
7              for(too=to; too<min(to+Tm, M); too++) {            (Point loop)
8                for(tii=ti; tii<(ti+Tn, N); tii++) {             (Point loop)
9                  for(i=0; i<K; i++) {                           (Point loop)
10                   for(j=0; j<K; j++) {                         (Point loop)
                       output_fm[to][row][col] +=
                       weights[to][ti][i][j]*input_fm[ti][S*row+i][S*col+j];
           }}}}}}
}}}}
```

A large design space

Constraints on CNN configuration

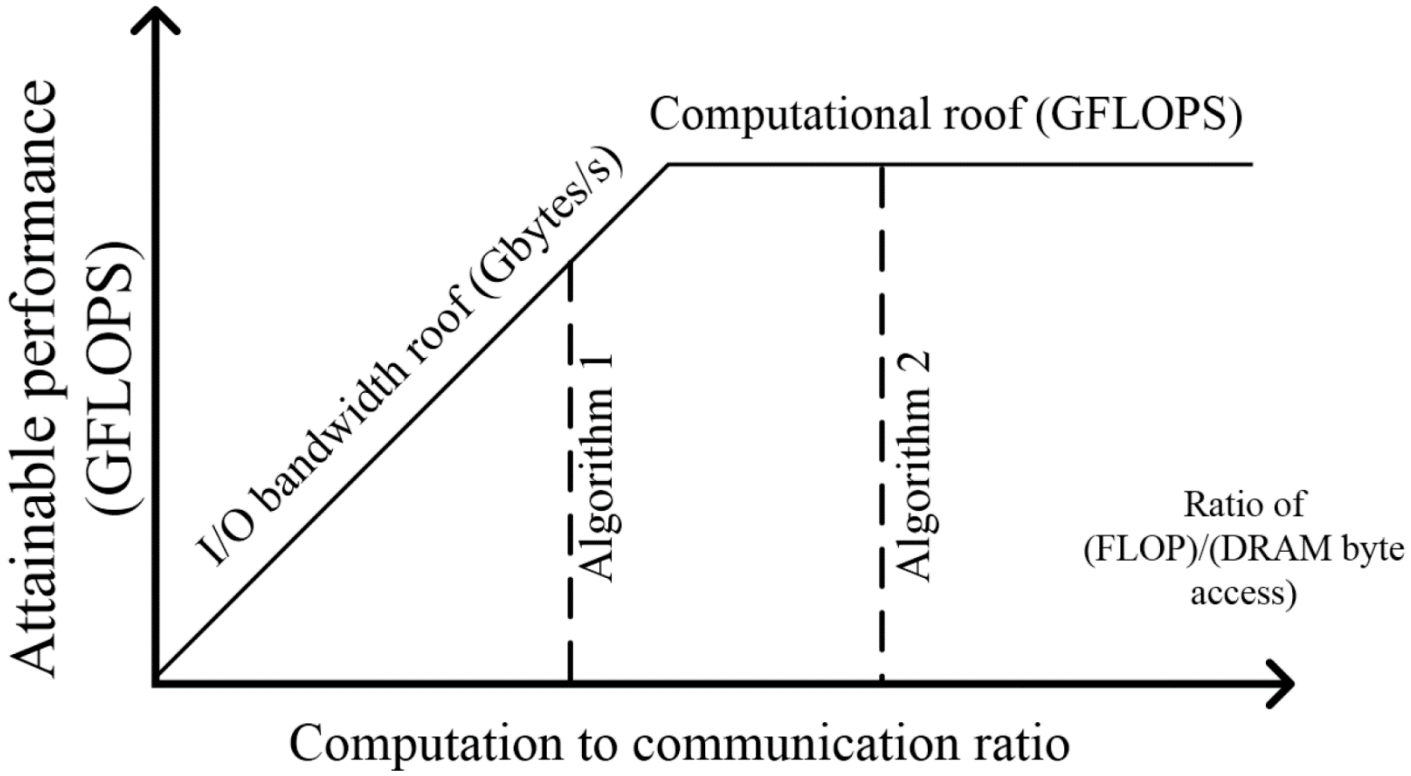
$$\begin{cases} 0 < Tm < M \\ 0 < Tn < N \\ 0 < Tr < R \\ 0 < Tc < C \end{cases}$$

Constraints on FPGA resource

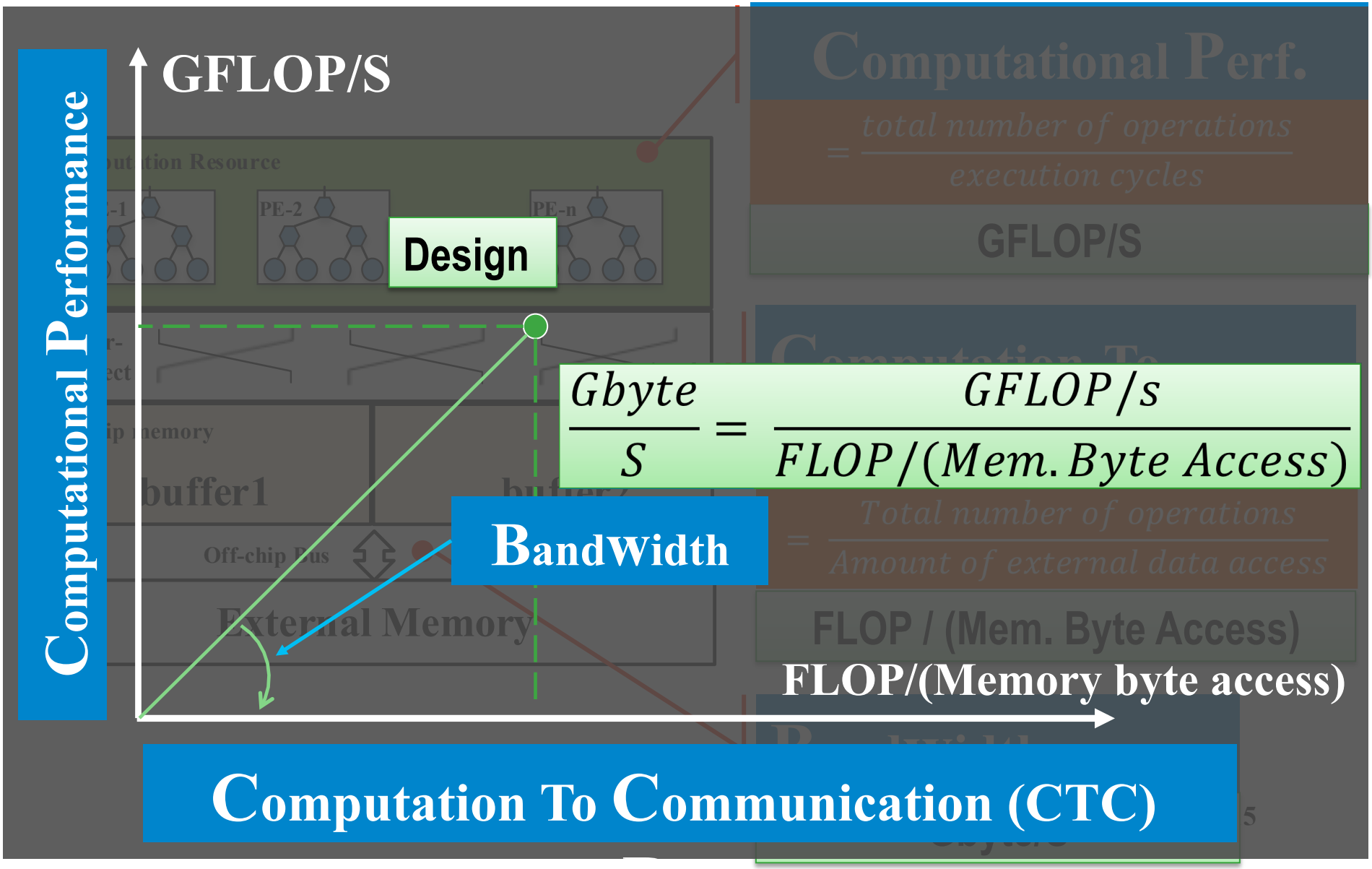
$$\begin{cases} 0 < Tm*Tn < \text{DSP resource} \\ 0 < Tn*Tr*Tc + Tm*Tr*Tc + Tm*Tn*K*K < \text{BRAM resource} \end{cases}$$

	CNN Configuration				FPGA Configuration		Design Space (Legal Solutions)
	R	C	M	N	BRAM	DSP	
Conv3_1(vgg16)	56	56	256	128	6 MB	3600	25, 627, 392
Conv3_2(vgg16)	56	56	256	256	6 MB	3600	28, 788, 480
Conv4_1(vgg16)	28	28	512	256	6 MB	3600	7, 874, 496
Conv5_2(vgg16)	14	14	512	512	6 MB	3600	2, 137, 968
Conv3(AlexNet)	13	13	192	256	6 MB	3600	1, 486, 524
Conv4(AlexNet)	13	13	192	192	6 MB	3600	1, 421, 628

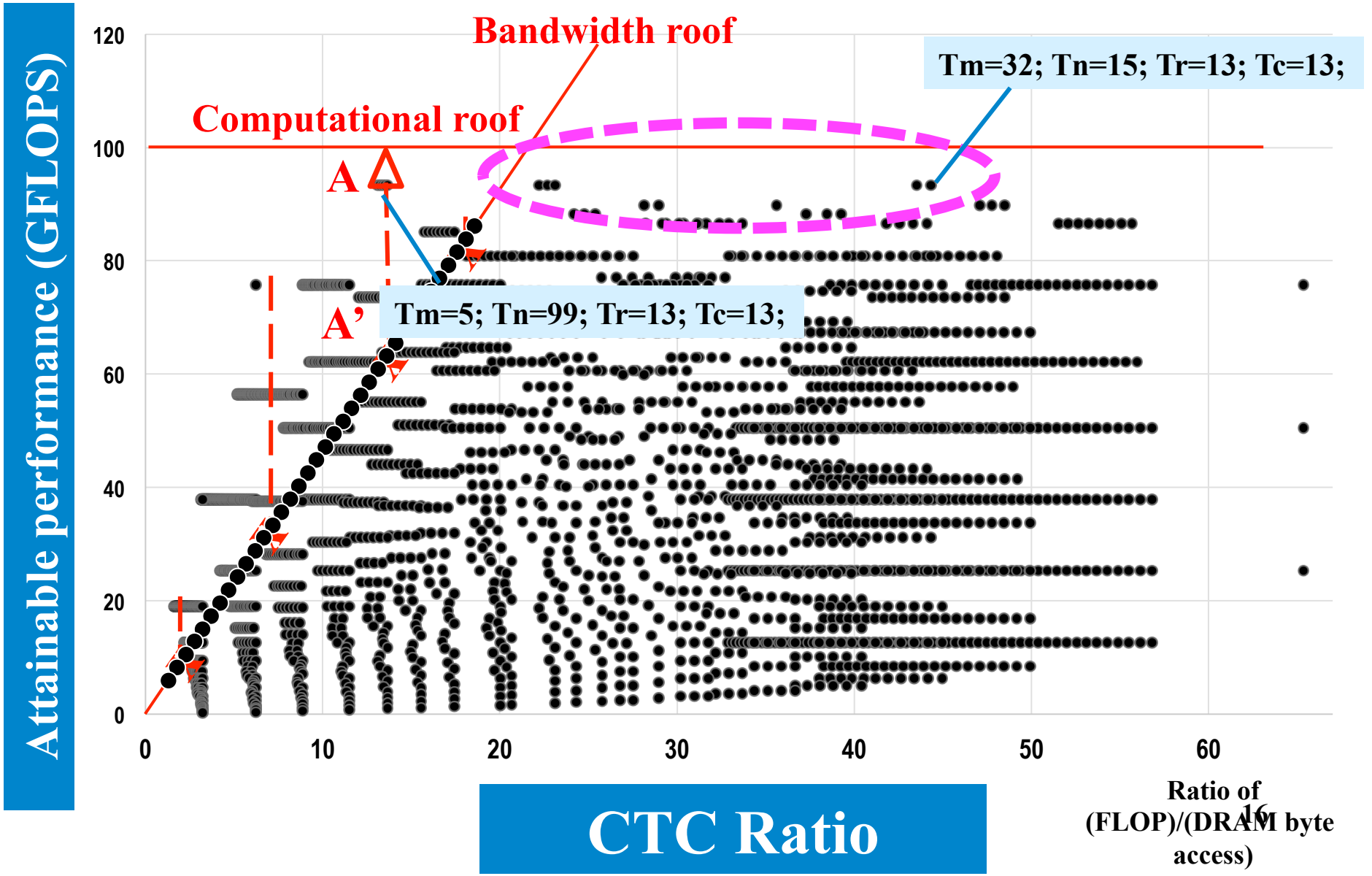
Roofline Model



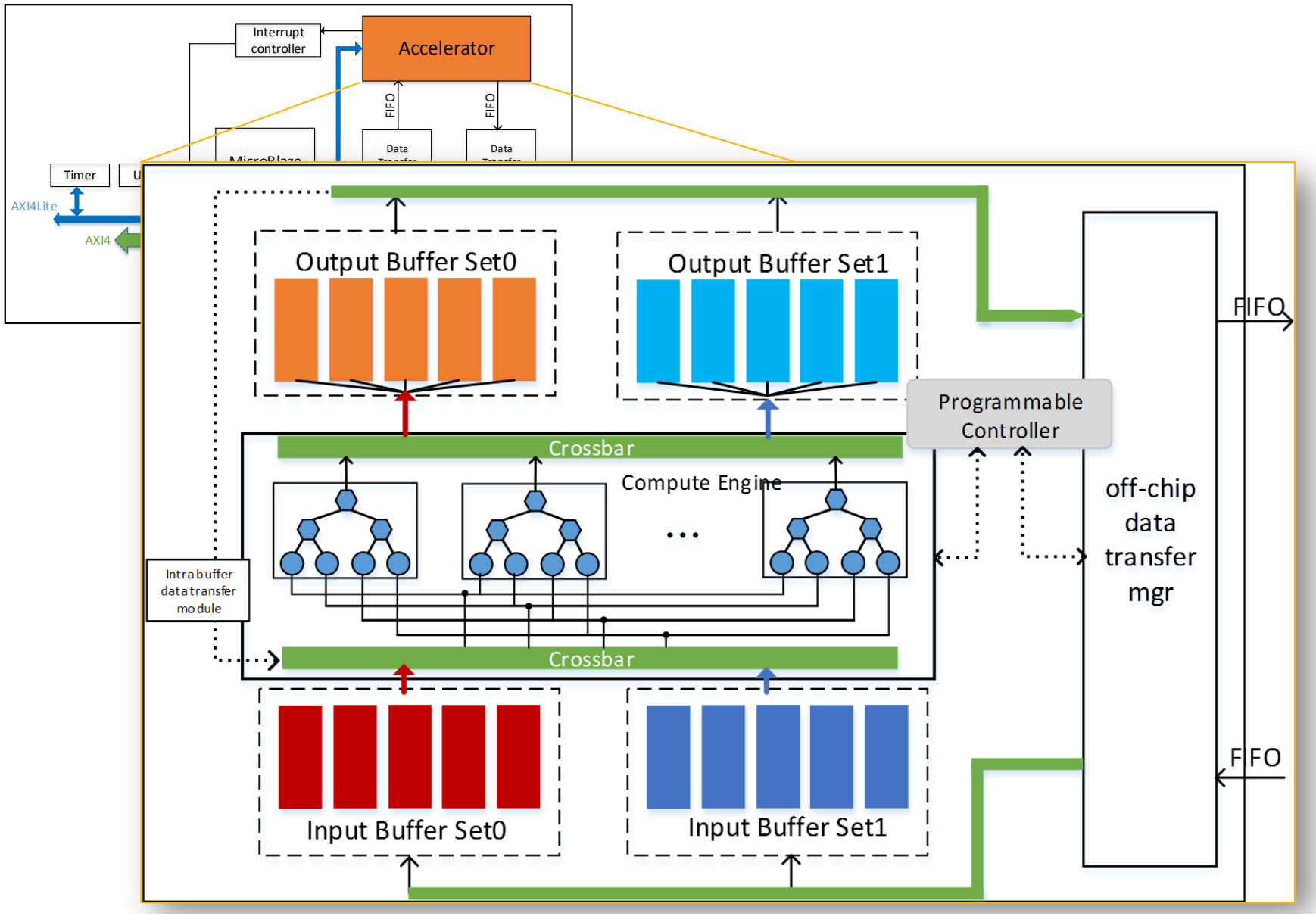
Our Approach: FPGA design in Roofline Model



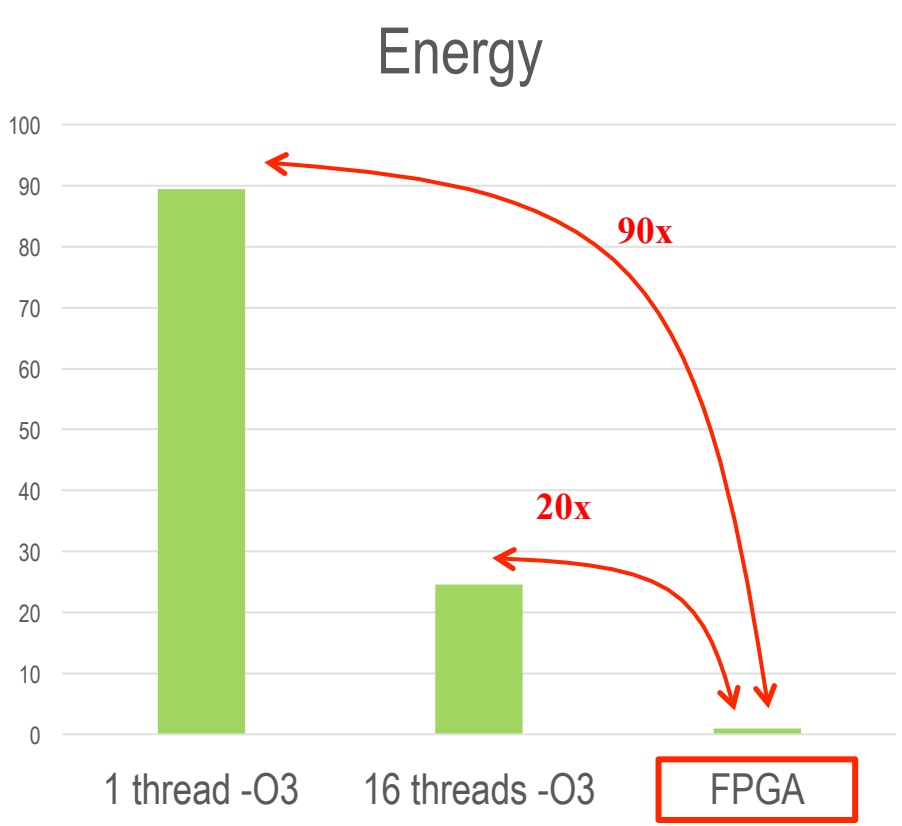
Design Space Exploration

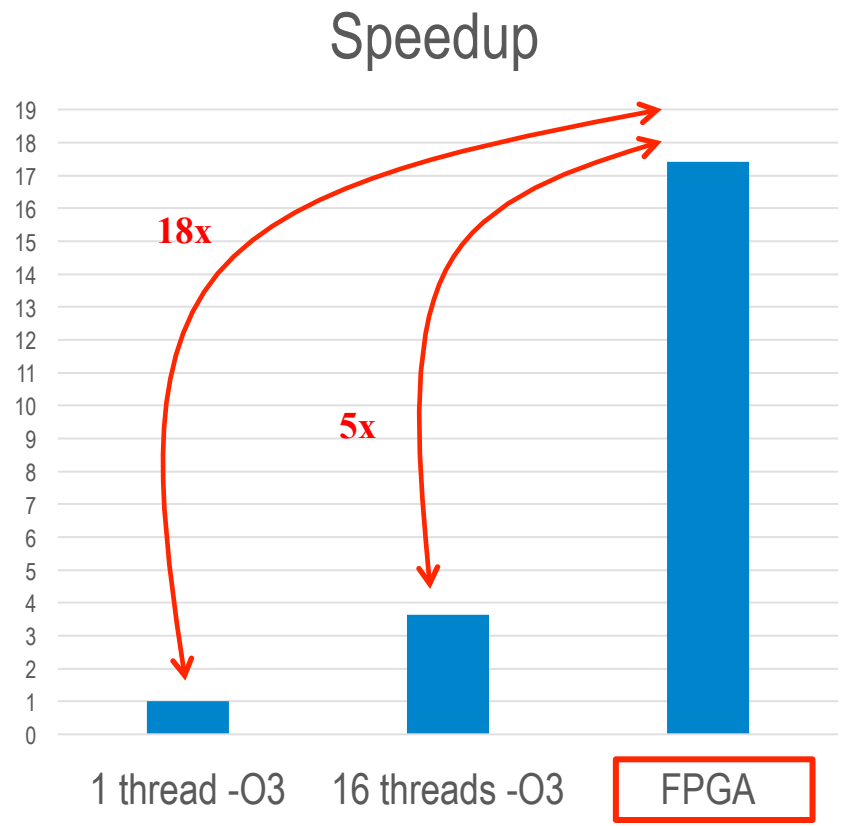


CNN Kernel Accelerator Architecture



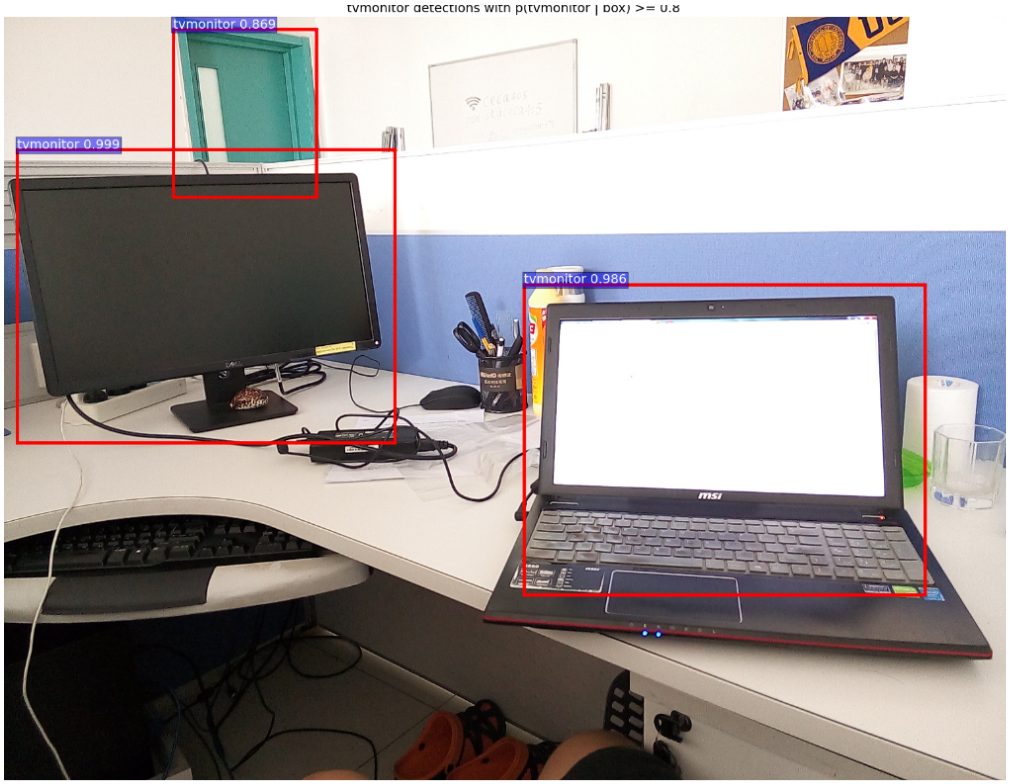
Experimental Results: vs. CPU



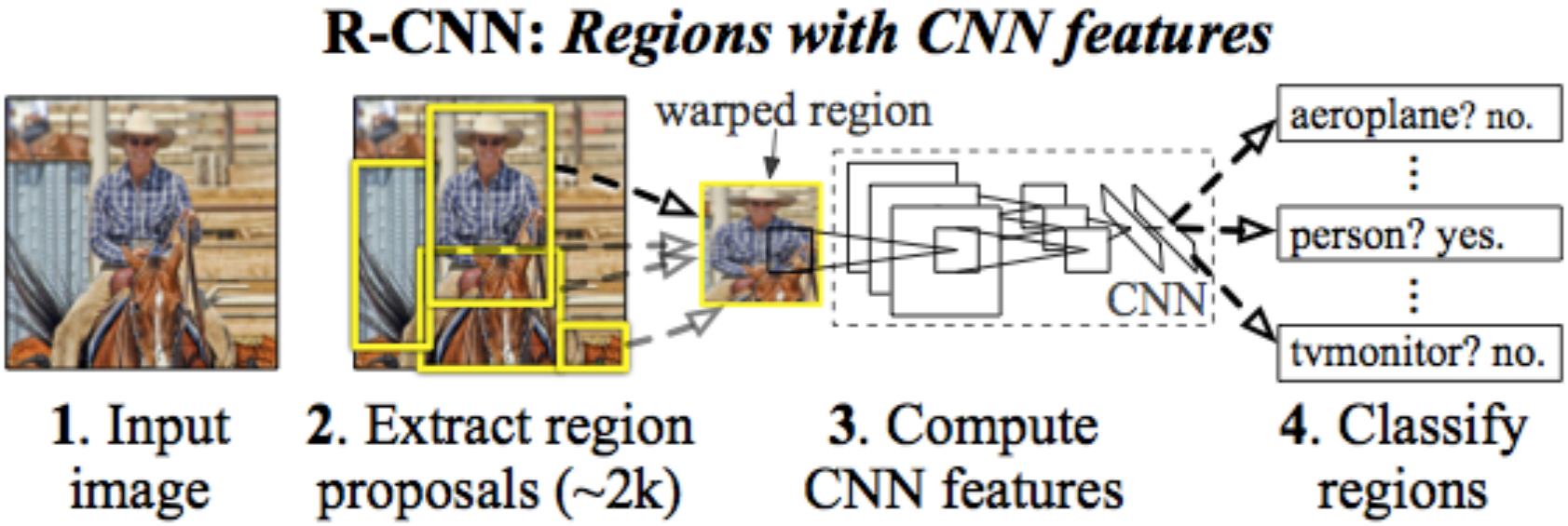


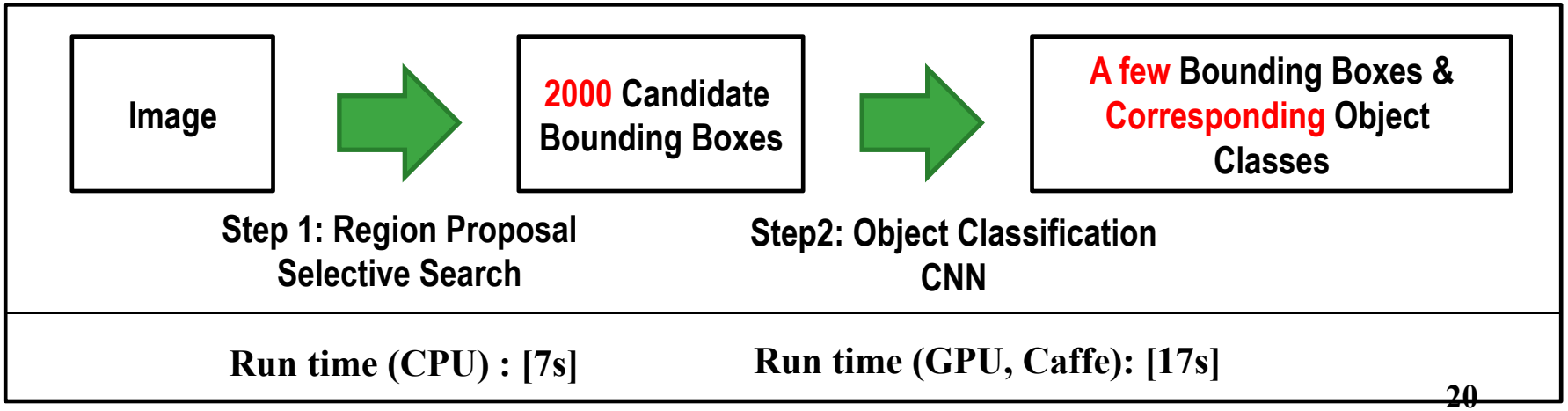
CPU	Xeon E5-2430 (32nm)	16 cores	2.2 GHz	gcc 4.7.2 –O3 OpenMP 3.0
FPGA	Virtex7-485t (28nm)	448 PEs	100MHz	Vivado 2015.2 Vivado HLS 2015.2

Object Detection: CNN-based Application

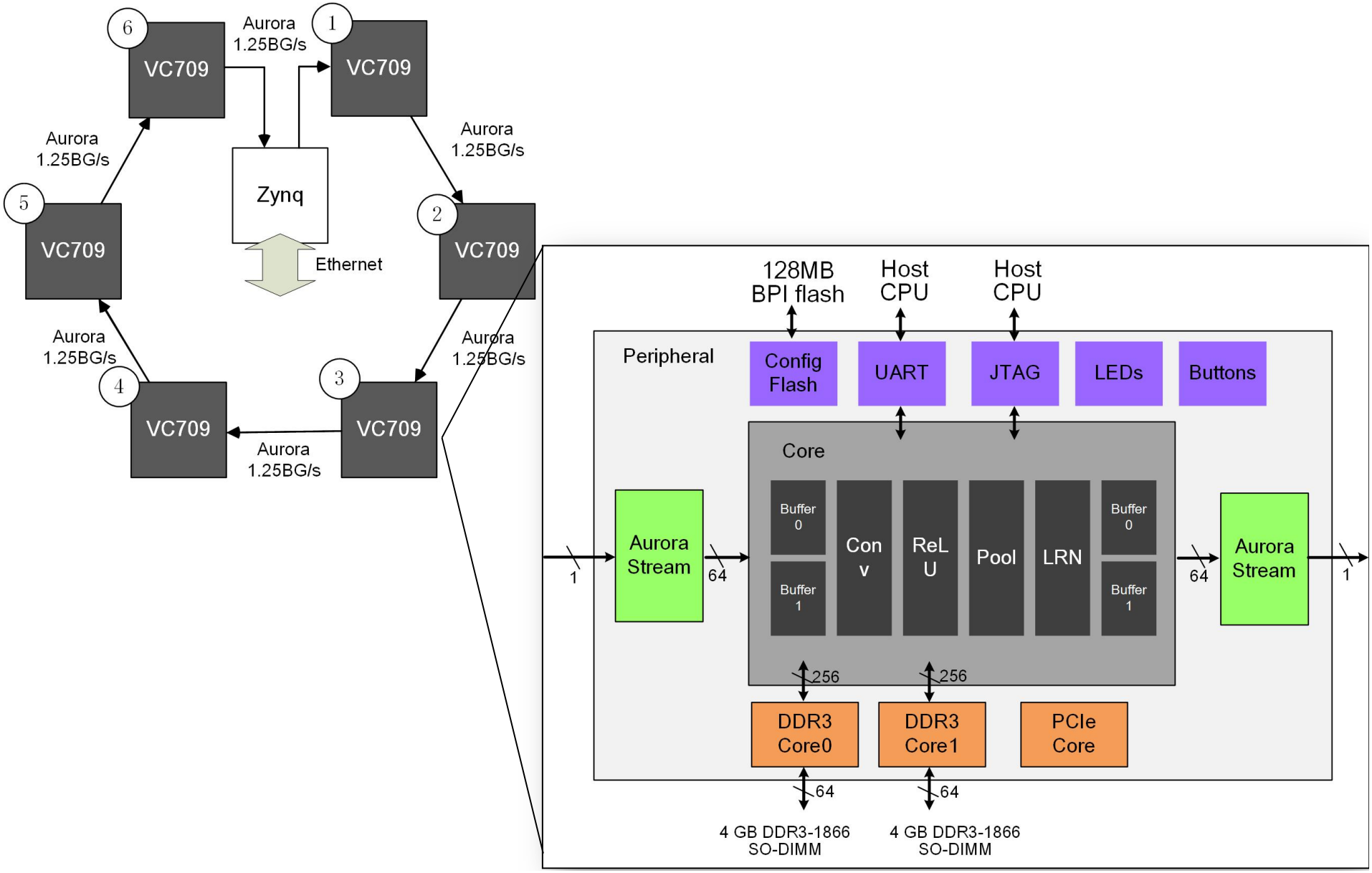


Application of CNN-based acceleration

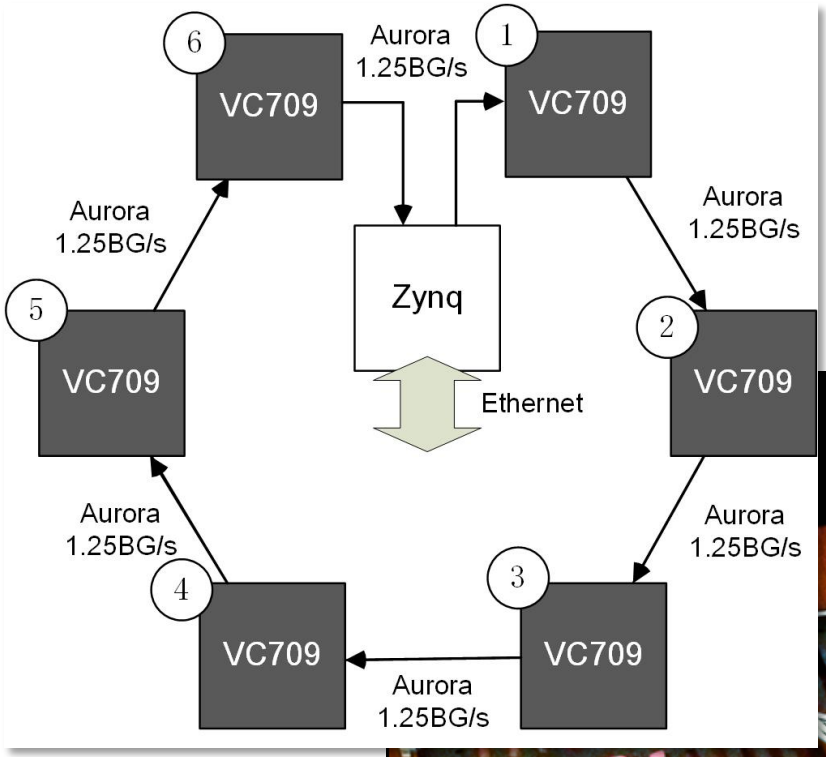


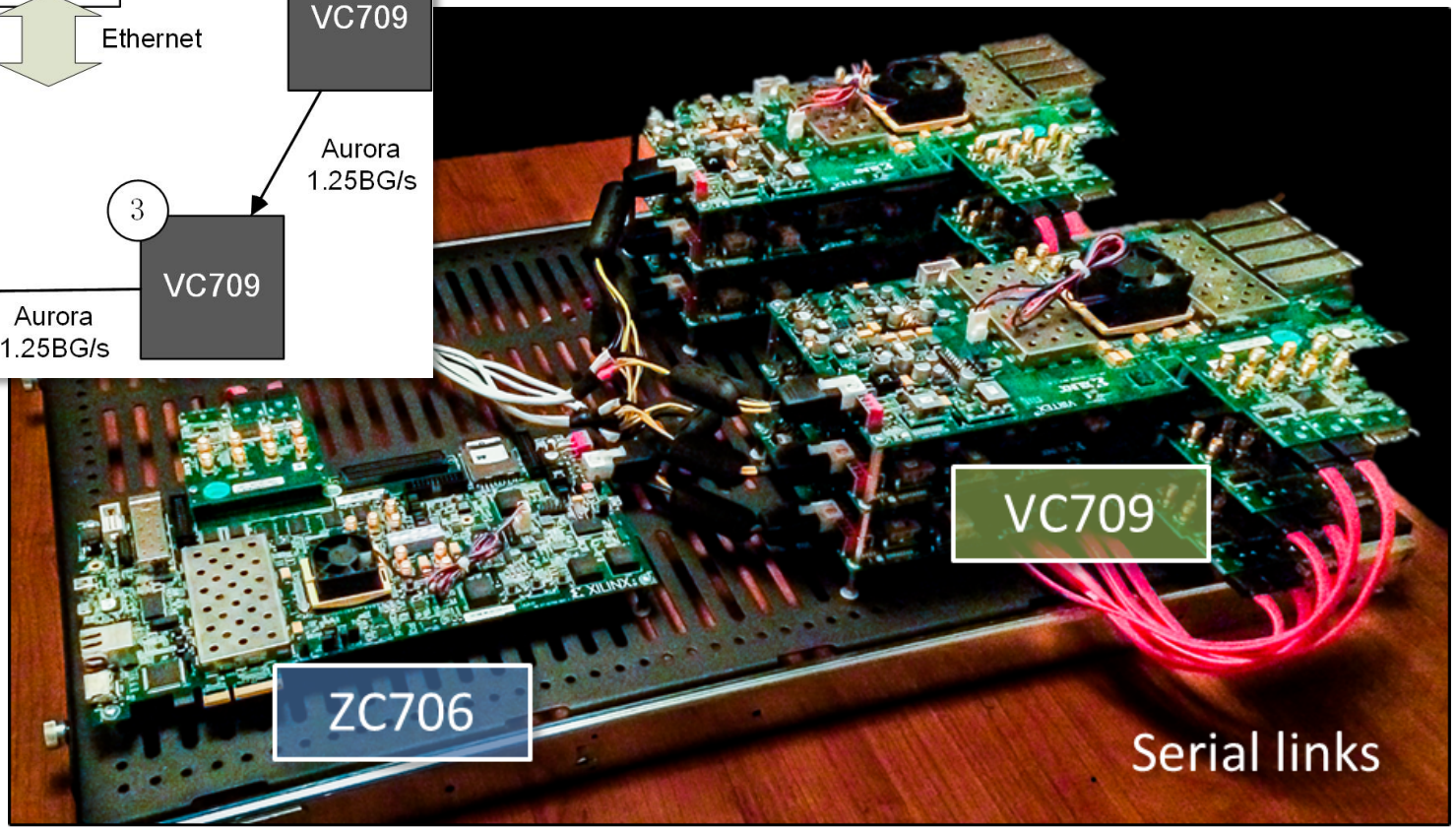


Hardware Architecture

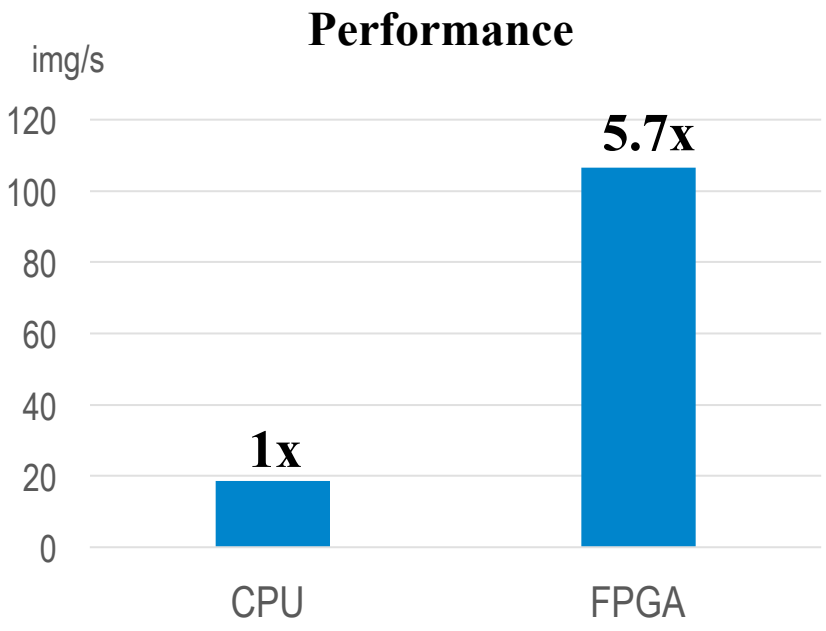


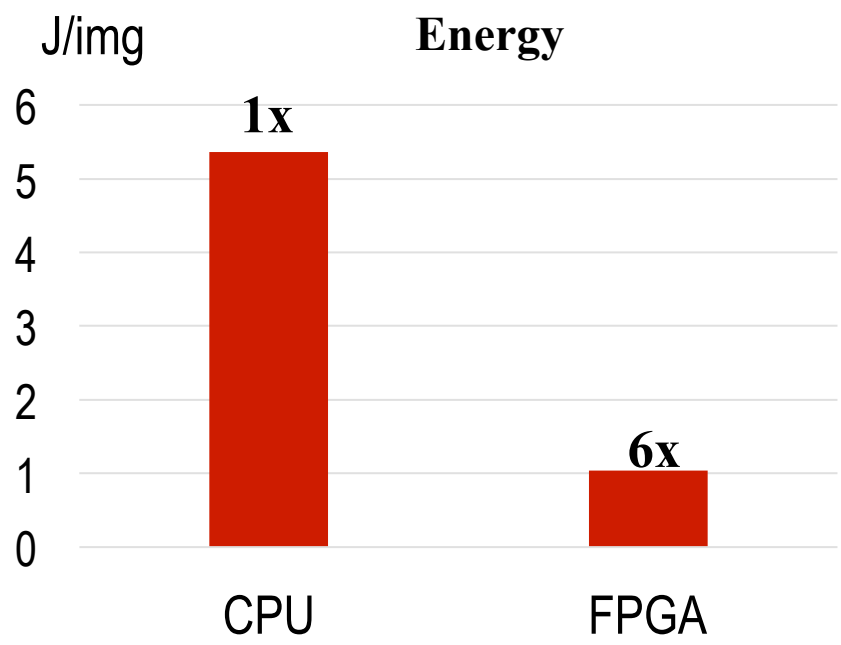
On-board implementation





Performance

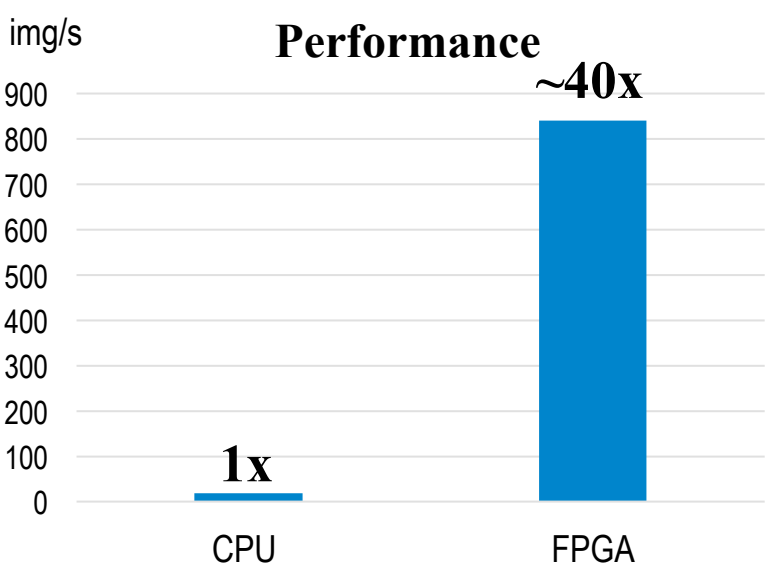


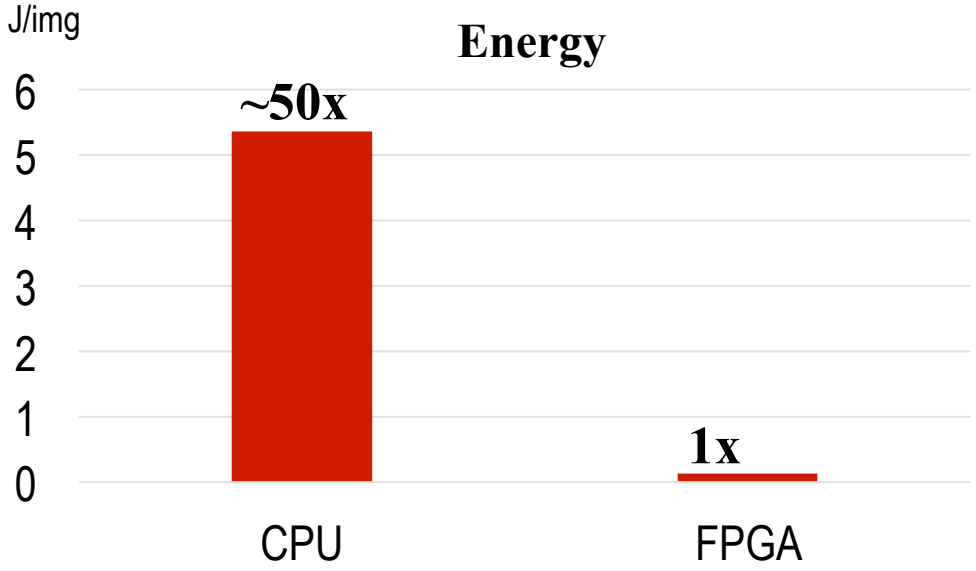


	CPU (Caffe+ATLAS)	FPGA
Device	E7-4807	VC709
Power	100 W	90W
Precision	float	float
Frequency	1.87 GHz	50MHz
Process	32nm	28nm

Projected Performance with tuning

more resource + higher frequency





	CPU (Caffe+ATLAS)	FPGA
Device	E7-4807	VC709
Power	100 W	90W
Precision	float	~~Float~~ fixed-point
Frequency	1.87 GHz	~~50MHz~~ 200MHz
Process	32nm	28nm

Demo

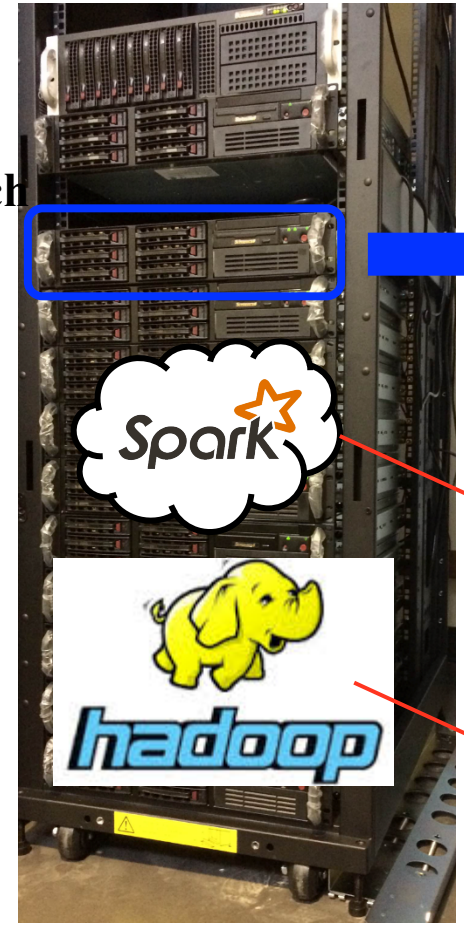
CDSC FPGA-Enabled Cluster

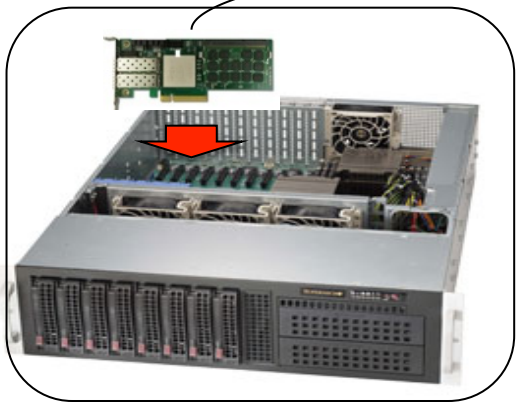
- A 24-node cluster with FPGA-based accelerators
  - Run on top of Spark and Hadoop (HDFS)

1 master / driver
1 10GbE switch

22 workers

1 file server





Alpha Data board:
1. Virtex-7 FPGA
2. 16GB on-board RAM

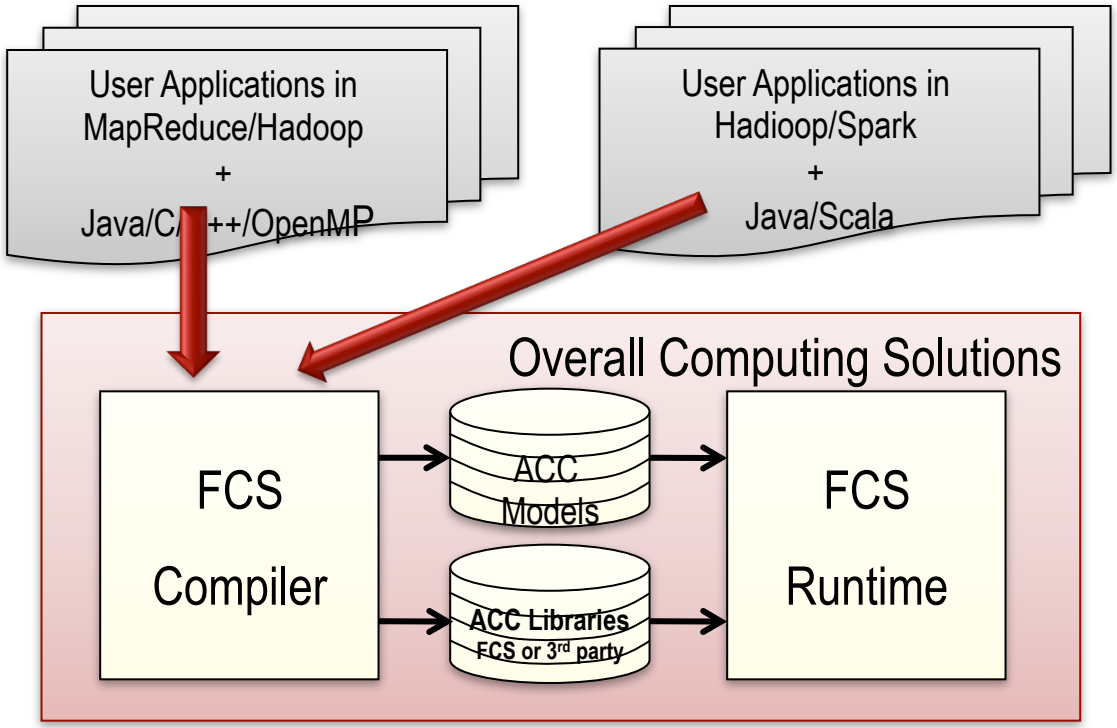
Each node:
1. Two Xeon processors
2. One FPGA PCIe card (Alpha Data)
3. 64 GB RAM
4. 10GBE NIC

Spark:
- Computation framework
- In-memory MapReduce system

HDFS:
- Distributed storage framework

Falcon Computing Solutions, Inc.

http://falcon-computing.com



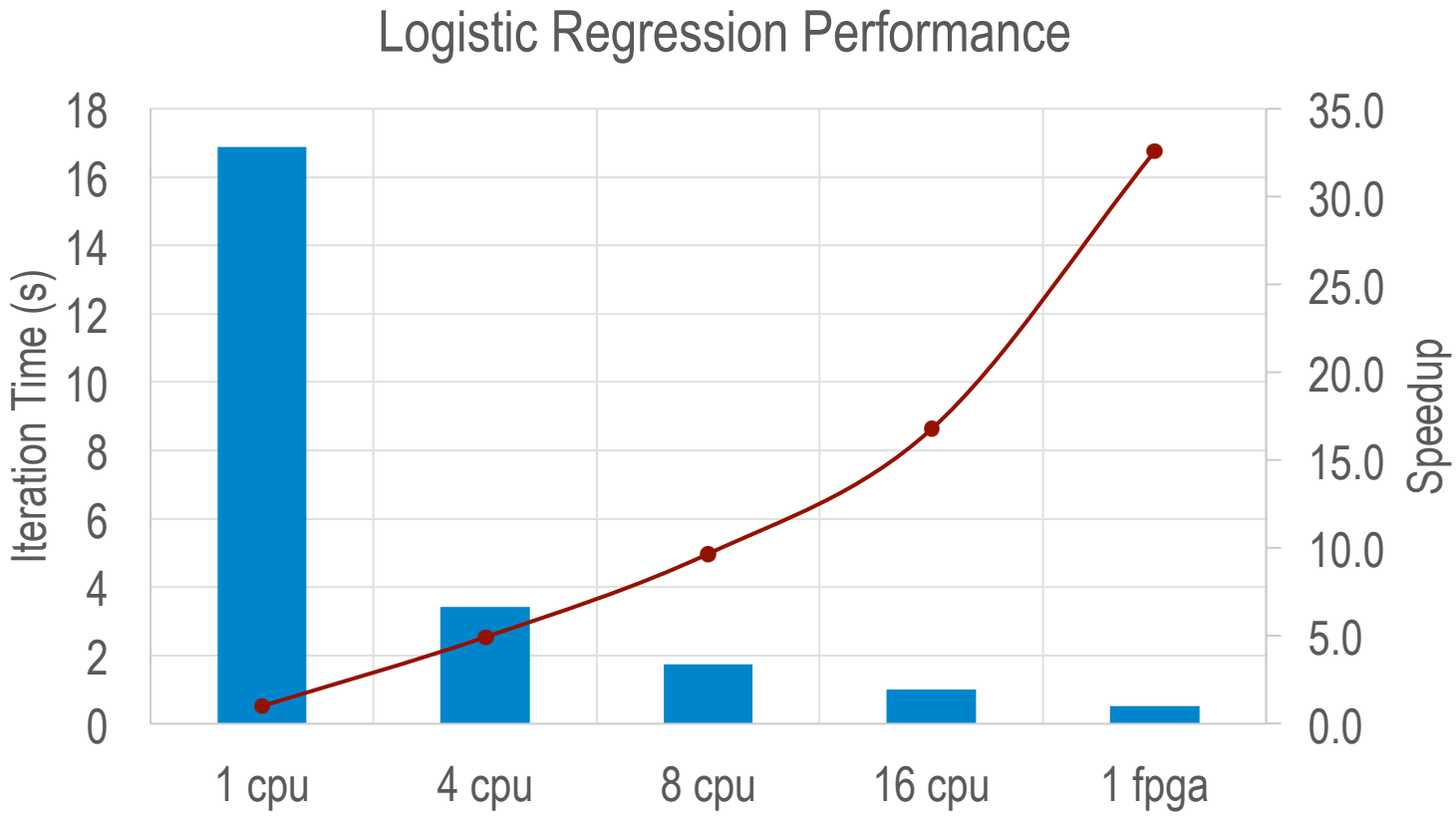
The only solution of FPGA customization and virtualization for Datacenter acceleration!

Experiment Setup

- Mini-cluster:
  - Server: Intel E5-2609 v3, 12-core @1.90Ghz, 96GB DDR4 Memory
  - FPGA: Alphadata KU3 FPGA board (8GB Memory, PCI-E Gen3 x8)
- Configuration
  - 4 Servers, 2 FPGA boards
  - 1Gbps Ethernet
- Software
  - FCS-Runtime
    - **Spark 1.4.0, Hadoop 2.6.0**
- Applications
  - Spark.mllib.classification.LogisticRegressionWithLBFGS
  - Spark.mllib.clustring.KMeans

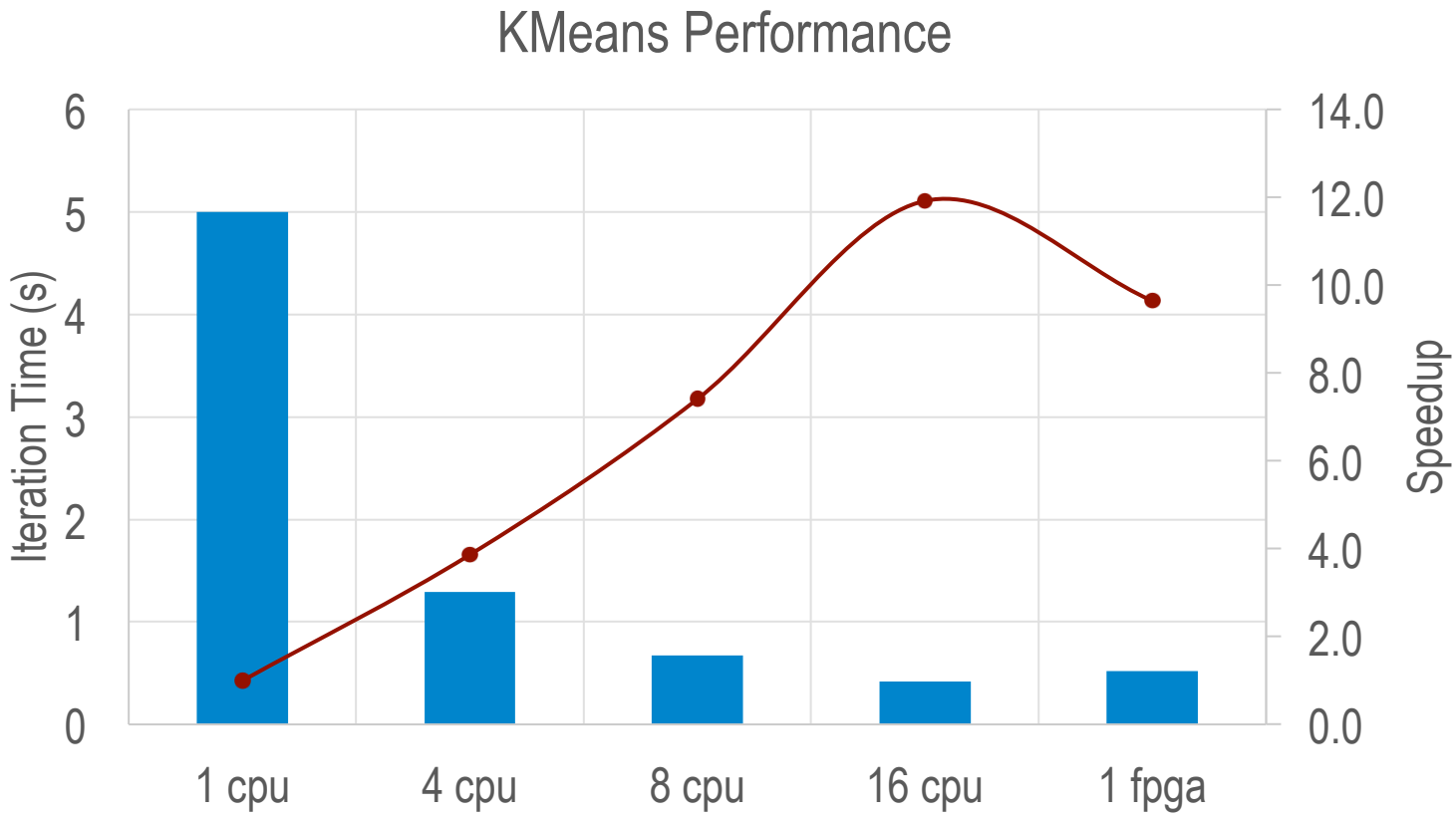
28

Logistic Regression Results



29

KMeans Results



30

Concluding Remarks

- New era of artificial intelligence
- A lot of opportunities for customization and specialization
- Customization at all levels
  - Chip-level
  - Server node level
  - Data center level
- Data center level customization holds great promise
- Software is critical
- Acceleration-aware ML algorithms

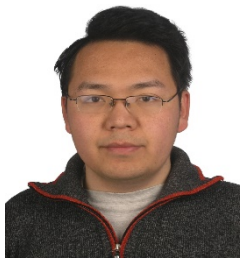
Acknowledgements

- Support from the Center for Domain-Specific Computing (CDSC) under the NSF Expeditions in Computing and InTrans Programs and the C-FAR Center under the STARnet Program
- Collaboration with Falcon Computing, Inc. and Peking University Center for Energy Efficient Computing and Applications (CECA)

Professor Jason Cong (UCLA)

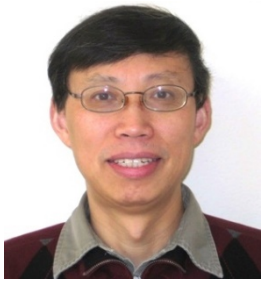
Assistant Prof. Guangyu Sun (PKU)

Assistant Prof. Guojie Luo (PKU)



Dr. Peng Li (UCLA & PKU)

Dr. Peng Zhang (Falcon-computing)

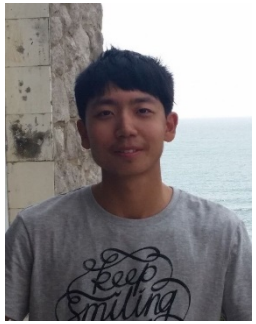


Dr. Peichen Pan (Falcon-computing)

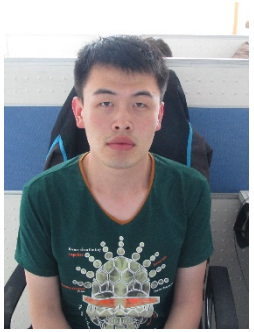
Di Wu (UCLA/Falcon)

Muhuan Huang (UCLA/Flacon)

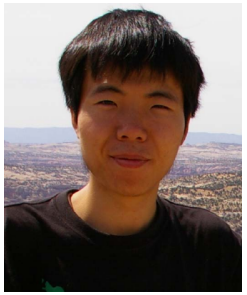
Chen Zhang (PKU /Flacon)



Jiayu Sun (PKU)



Yijin Guan (PKU)



Bingjun Xiao (UCLA)