HyPar: Towards Hybrid Parallelism for Deep Learning Accelerator Array

Linghao Song*, Jiachen Mao*, Youwei Zhuo#,
Xuehai Qian#, Hai Li*, Yiran Chen*

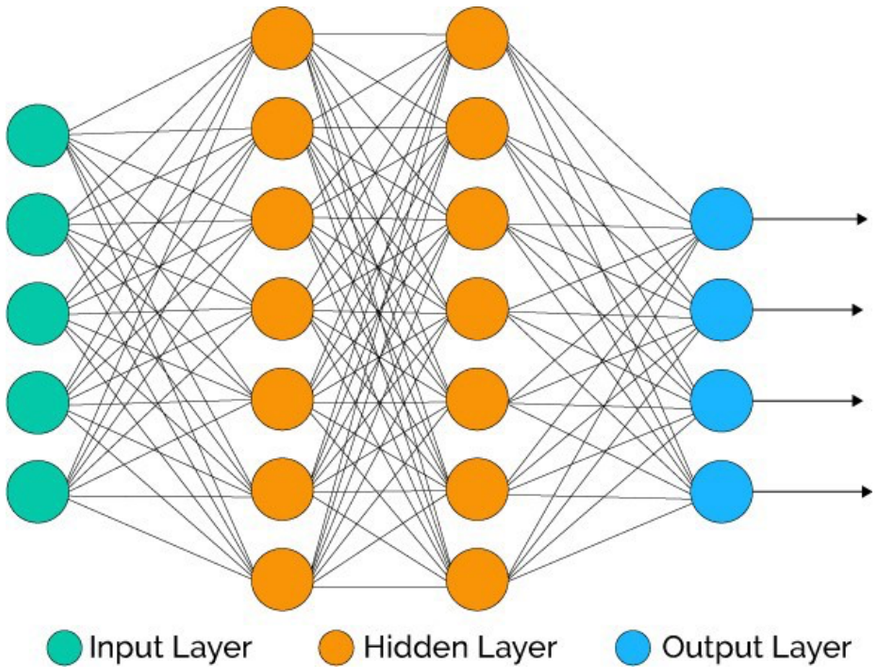
*Duke University

#University of Southern California

CEI
cei.pratt.duke.edu

ALCHEM
alchem.usc.edu

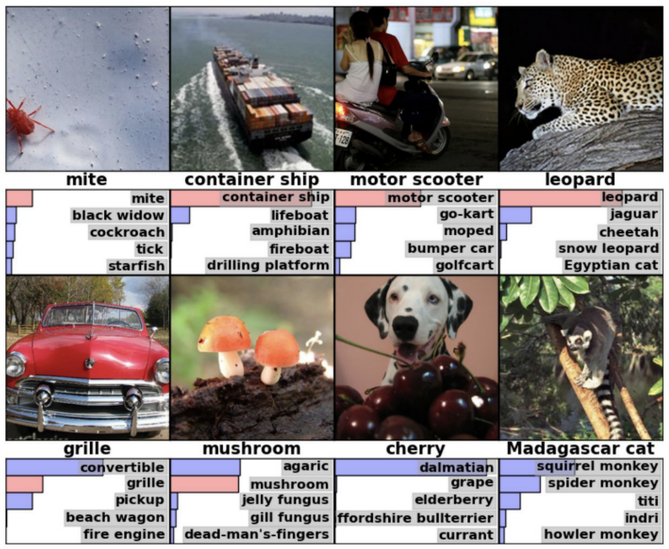
As a computer architect
we know deep learning applications in various domains…

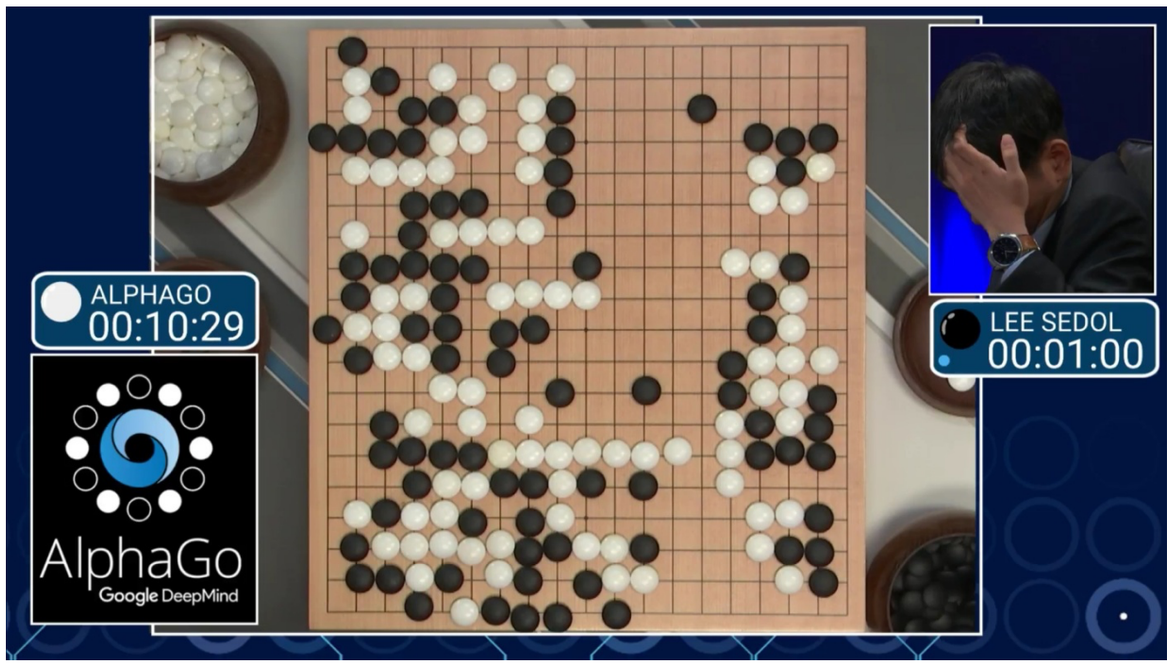


ImageNet Challenge (IJCV'15)

IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.





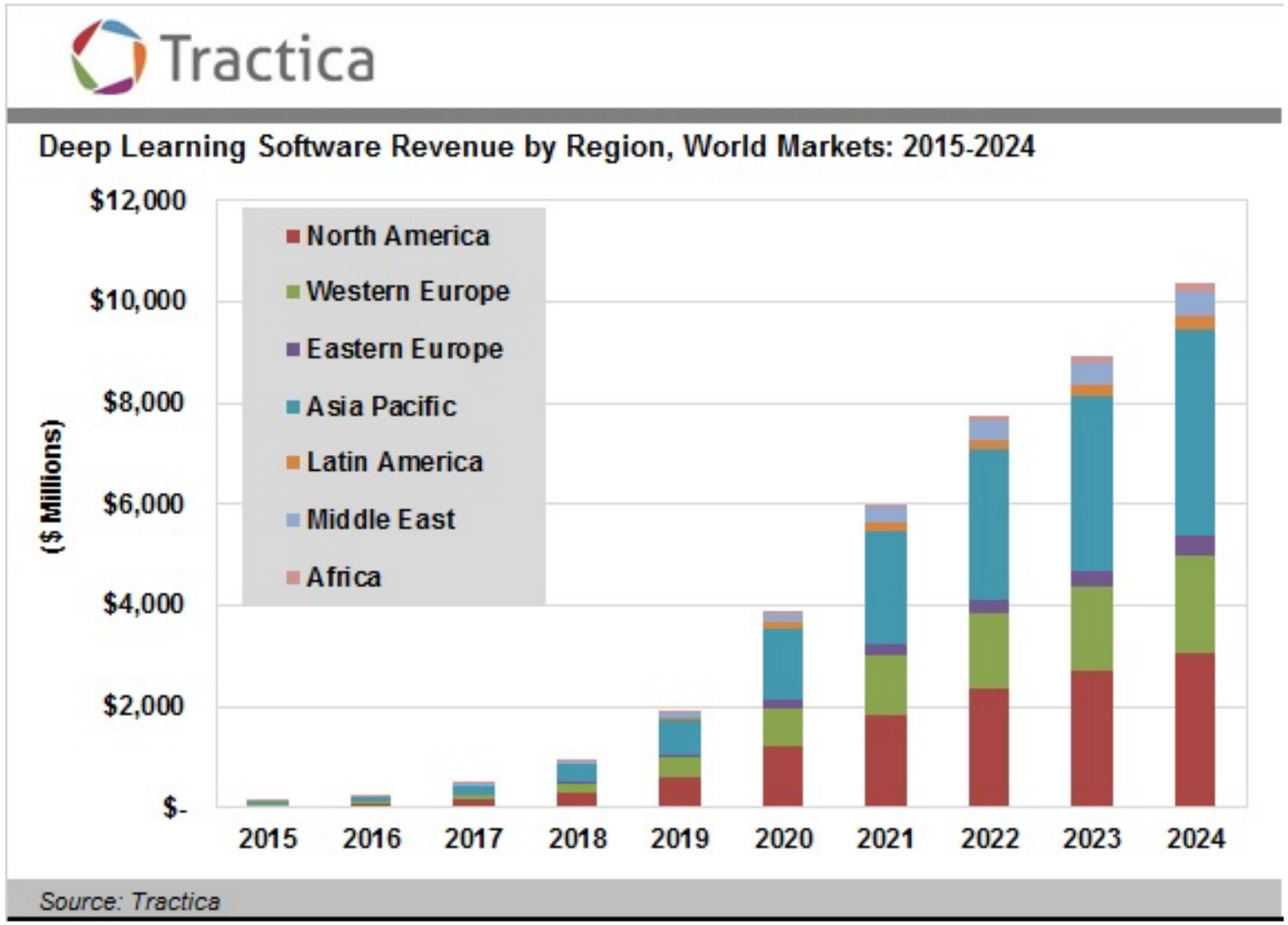
CEI
cei.pratt.duke.edu

(AlphaGo, DeepMind)

(Biggan, ICLR'19)

ALCHEM
alchem.usc.edu

We also know deep learning is rapidly growing..



So we agree it is an application domain for specialized architecture.

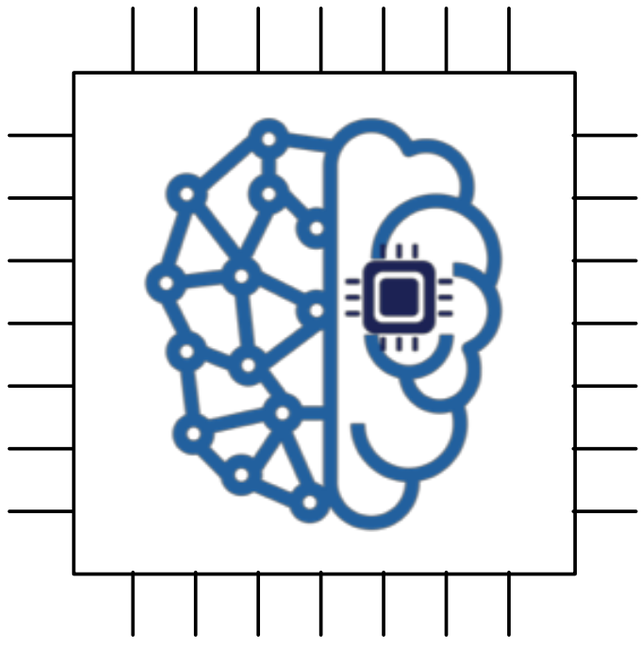
CEI

cei.pratt.duke.edu

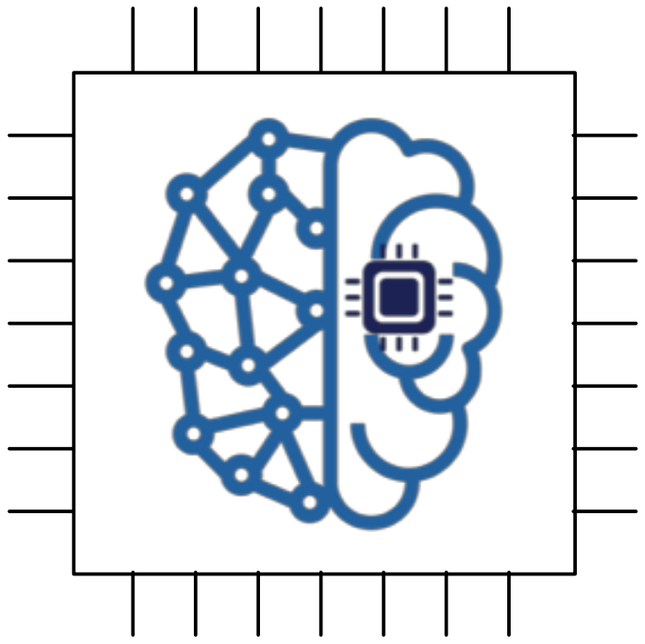
ALCHEM

alchem.usc.edu

Current approach: NN->Chip, Mapping, Dataflow, etc.



Current approach: NN->Chip, Mapping, Dataflow, etc.



Current approach: NN->Chip, Mapping, Dataflow, etc.

But…

- That is not new, even boring…
- Morning Session Warning: Accelerator Wall
- So many DL accelerators…
- Why HyPar? **What's next for DL accelerators?**

Outline

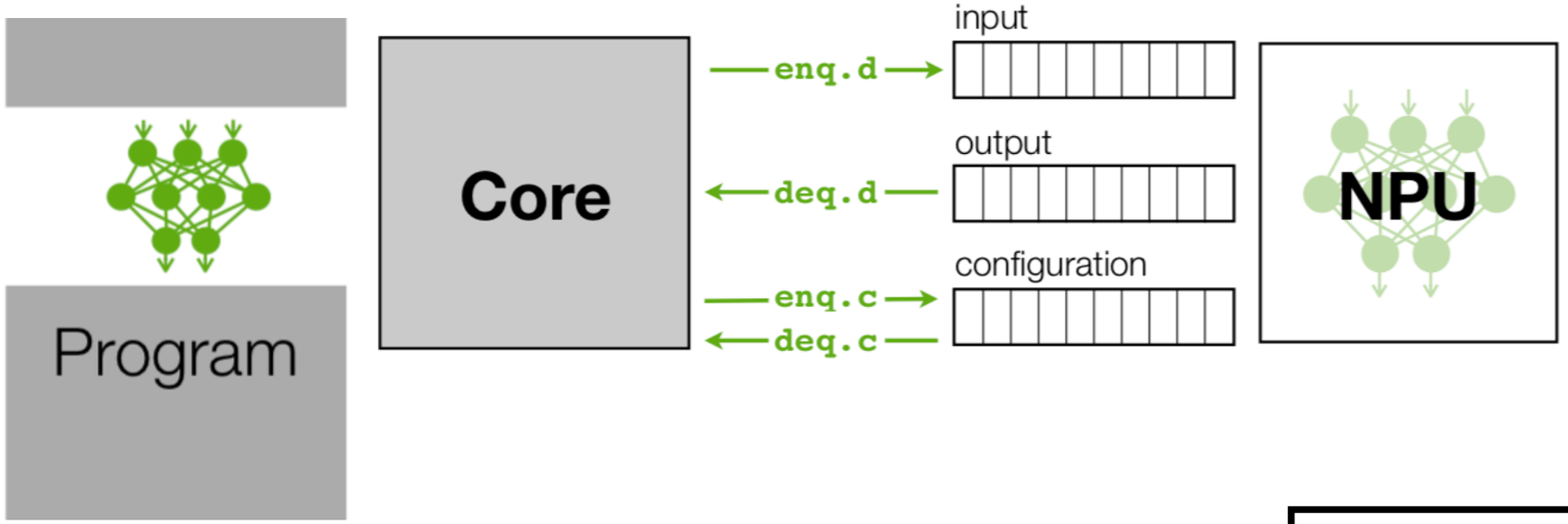
- The development of deep learning accelerators
- Why HyPar
- HyPar: hybrid parallelism for deep learning accelerator array
  - Communication model
  - Tensor partition
  - Evaluation
- Conclusion

Outline

- The development of deep learning accelerators
- Why HyPar
- HyPar: hybrid parallelism for deep learning accelerator array
  - Communication model
  - Tensor partition
  - Evaluation
- Conclusion

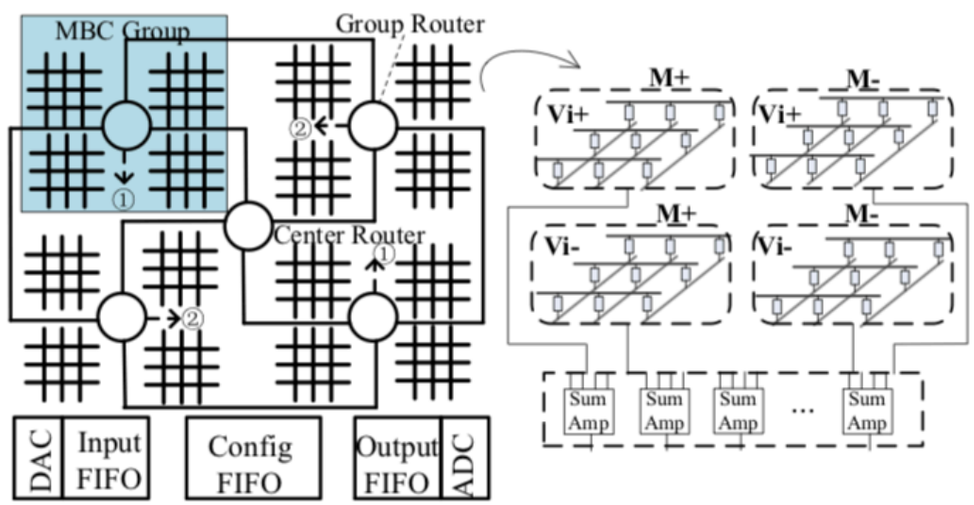
DLA early stage: on-chip design

- NPU: To run **some program segment** on NPU (neural processing unit) instead of CPU (MICRO'12)



- Reno

(DAC'15)



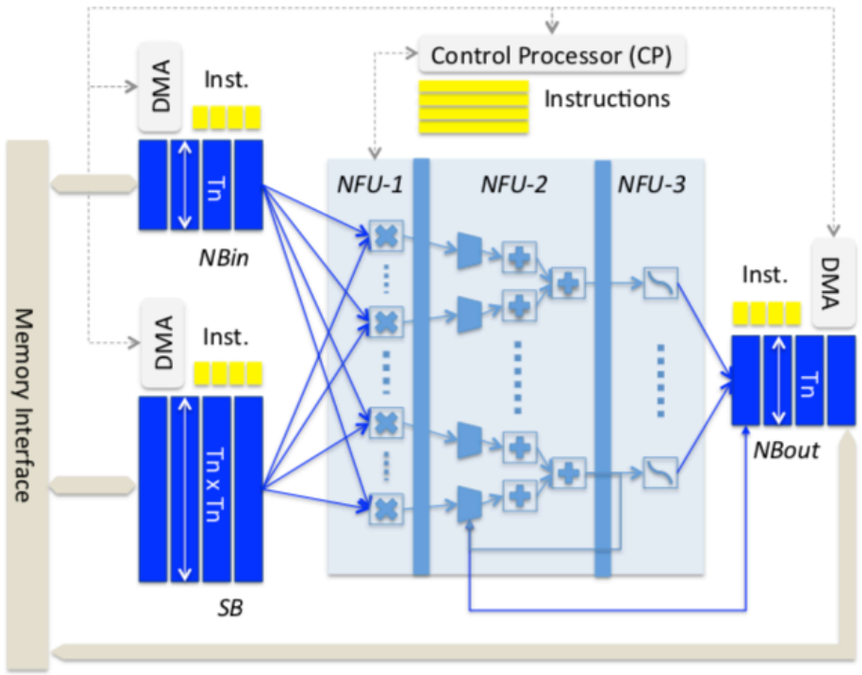
Limited application and performance

CEI
cei.pratt.duke.edu

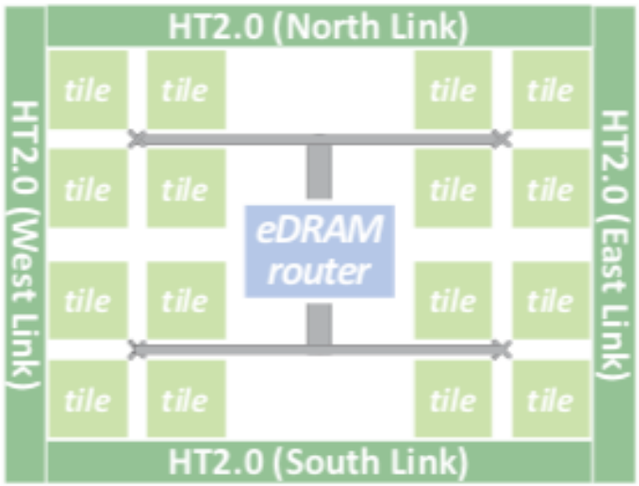
ALCHEM
alchem.usc.edu

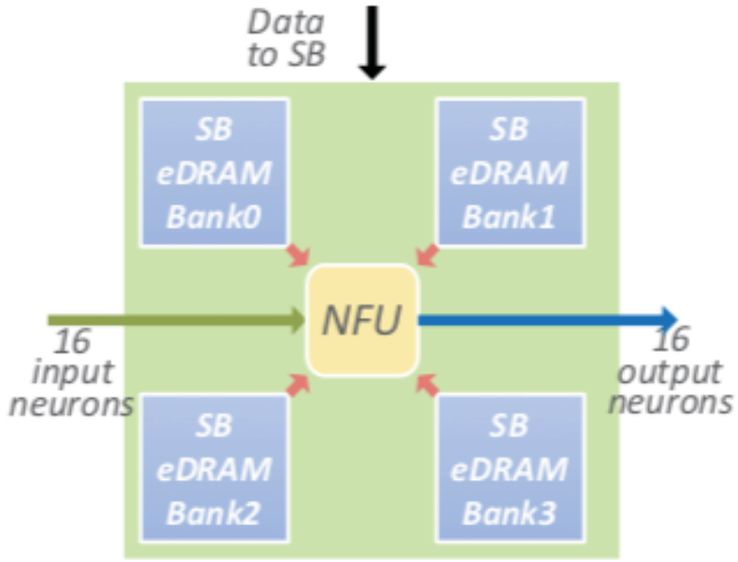
DLA current stage: stand-alone accelerators

- Diannao (ASPLOS'14)



- DaDiannao (MICRO'14)





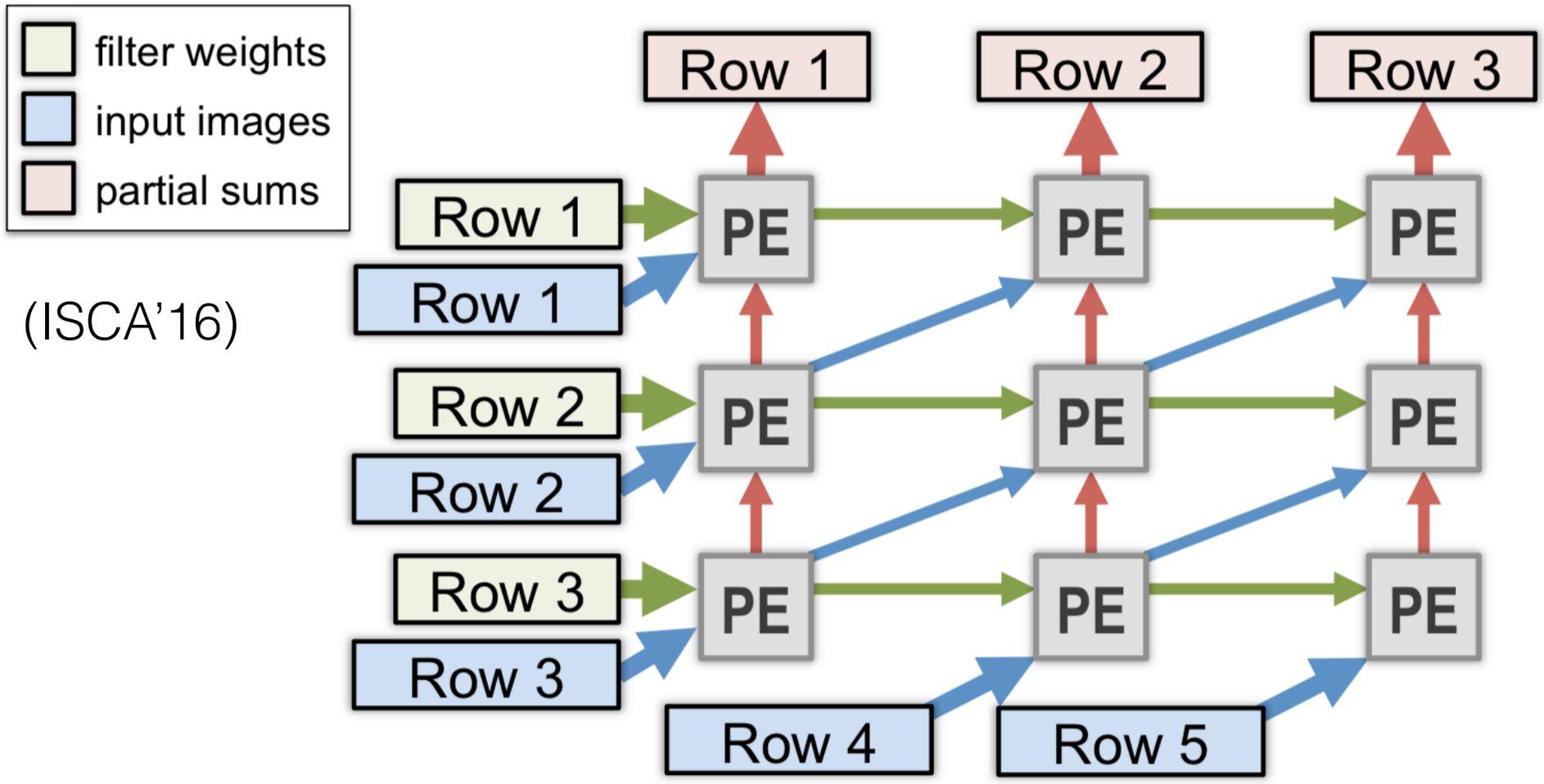
- ShiDiannao (ISCA'15) ,PuDiannao (ASPLOS'15),Cambricon-X (MICRO'16)

Support a whole application (NN) on an accelerator.

CEI
cei.pratt.duke.edu

ALCHEM
alchem.usc.edu

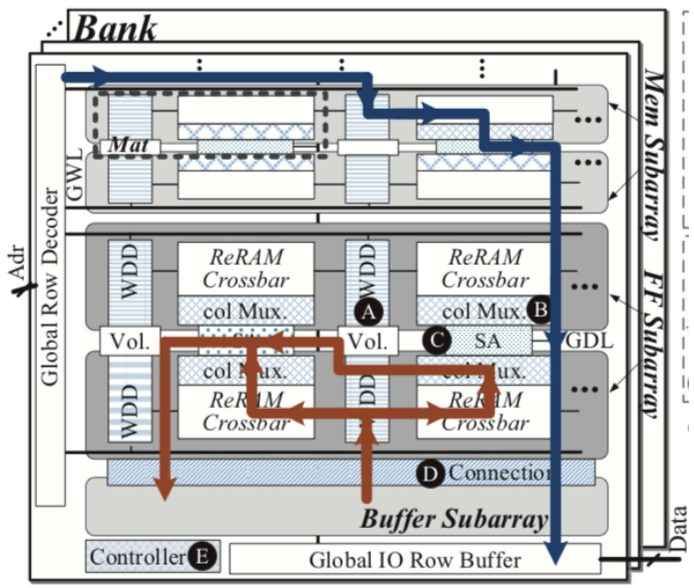
DLA current stage: data flow architecture(Eyeriss)



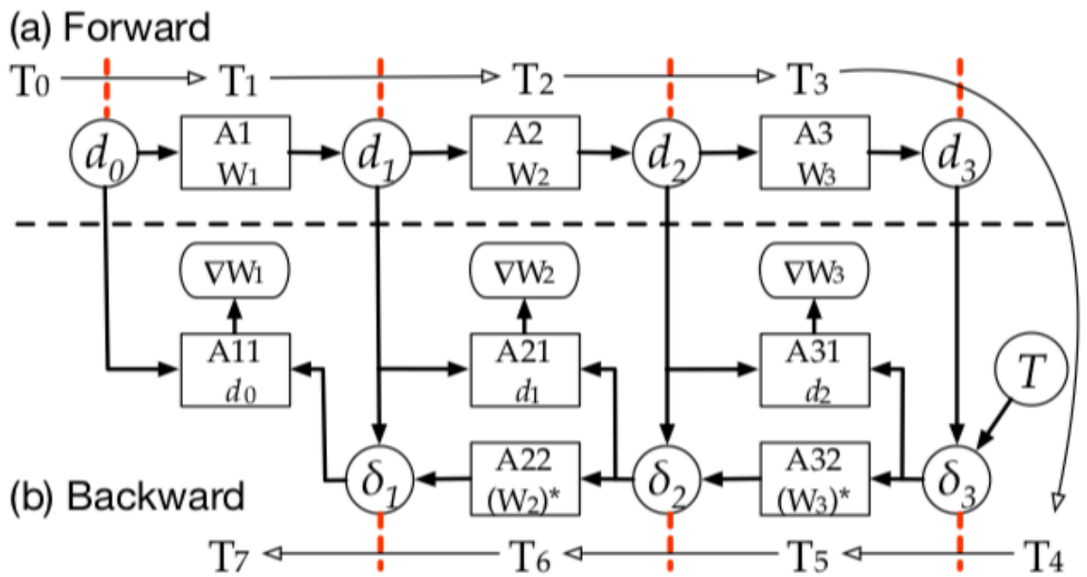
Fine exploration on data/resource mapping, sharing, reusing.

DLA current stage: emerging memory(ReRAM)

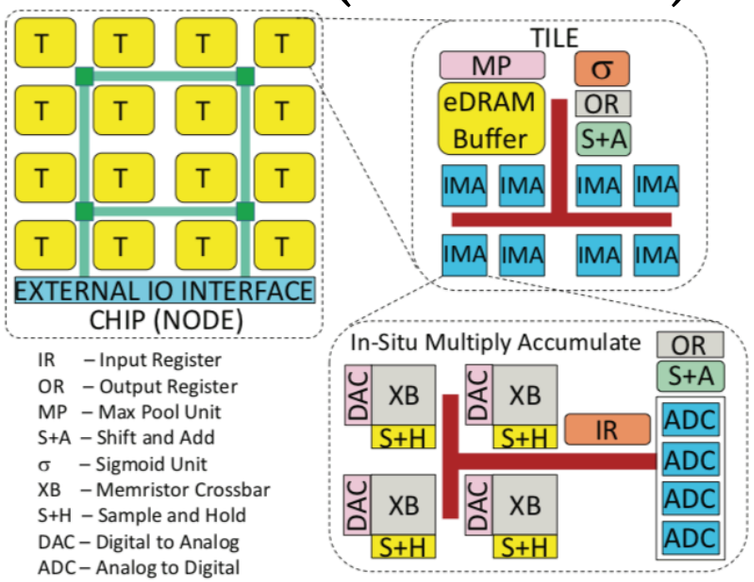
PRIME (ISCA'16)



PipeLayer (HPCA'17)



ISAAC (ISCA'16)



Take the benefit from emerging technologies beyond CMOS.

CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

Where are we in the development of DLAs?

CEI

cei.pratt.duke.edu

ALCHEM

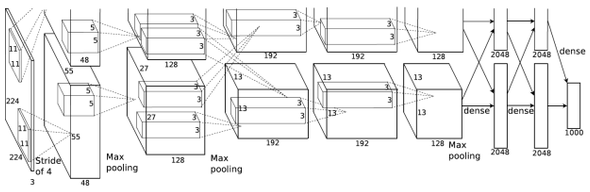
alchem.usc.edu

Where are we in the development of DLAs?

CEI
cei.pratt.duke.edu

ALCHEM
alchem.usc.edu

Where are we in the development of DLAs?



AlexNet

2012

NPU

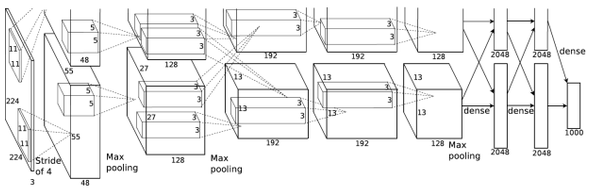
CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

Where are we in the development of DLAs?



AlexNet

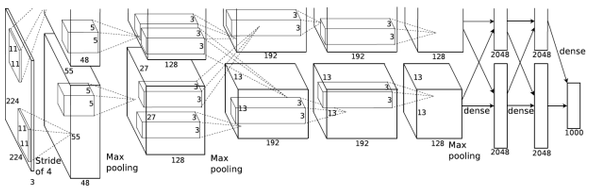
2012

NPU

2014

Diannao, DaDinnao, ShiDiannao,
PuDiannao, Cambricon, Cambricon-X

Where are we in the development of DLAs?



RedEye Cnvlutn

PRIME ISAAC

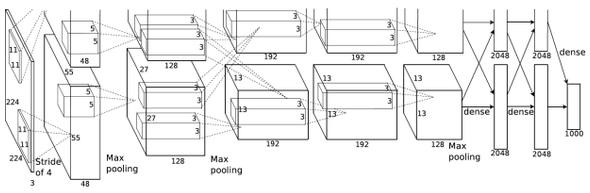
AlexNet Eyeriss EIE

2012 2014 2016

NPU

Diannao, DaDinnao, ShiDiannao,
PuDiannao, Cambricon, Cambricon-X

Where are we in the development of DLAs?



RedEye Cnvlutn CirCNN

PRIME ISAAC Scalpel

AlexNet Eyeriss EIE SCNN

FlexFlow

DNPU Tetris

2012 2014 2016 ESE 2017

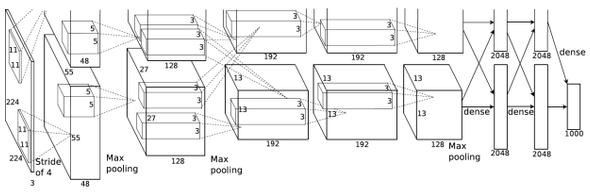
TPU FINN

PipeLayer

NPU Diannao, DaDinnao, ShiDiannao,

PuDiannao, Cambricon, Cambricon-X

Where are we in the development of DLAs?



RedEye Cnvlutn CirCNN DelftNN

PRIME ISAAC Scalpel ReGAN

AlexNet Eyeriss EIE SCNN CONV-RAM

FlexFlow QUEST

DNPU Tetris MAERI

VIBNN

2012 2014 2016 ESE 2017 C-LSTM

TPU FINN DeltaRNN

PipeLayer RANA

NPU Diannao, DaDinnao, ShiDiannao, GANAX

PuDiannao, Cambricon, Cambricon-X UCNN

CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

Why HyPar

As a deep learning accelerator architect, we have well-designed DL accelerators, and we are approaching the accelerator optimization limitation (Accelerator Wall).

Why not use an accelerator array?

How can we parallelize them?

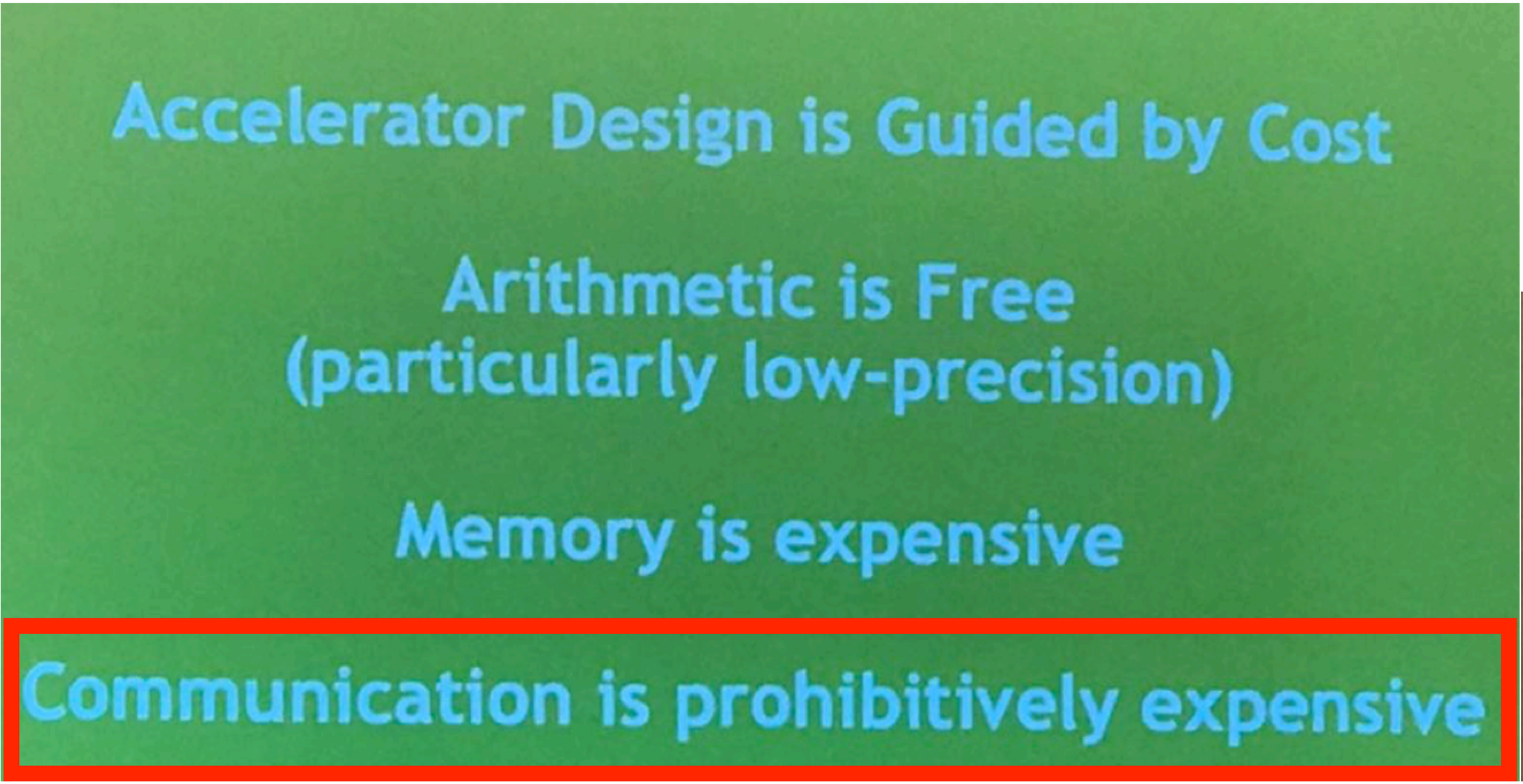
Outline

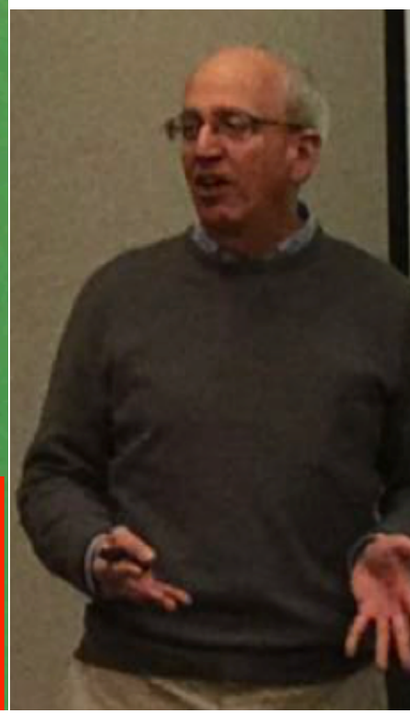
- The development of deep learning accelerators
- Why HyPar
- HyPar: hybrid parallelism for deep learning accelerator array
  - Communication model
  - Tensor partition
  - Evaluation
- Conclusion

Outline

- The development of deep learning accelerators
- Why HyPar
- HyPar: hybrid parallelism for deep learning accelerator array
  - Communication model
  - Tensor partition
  - Evaluation
- Conclusion

HyPar: communication model





Bill Dally@AACBB(Sat.)

=> Let's reduce communication!

CEI
cei.pratt.duke.edu

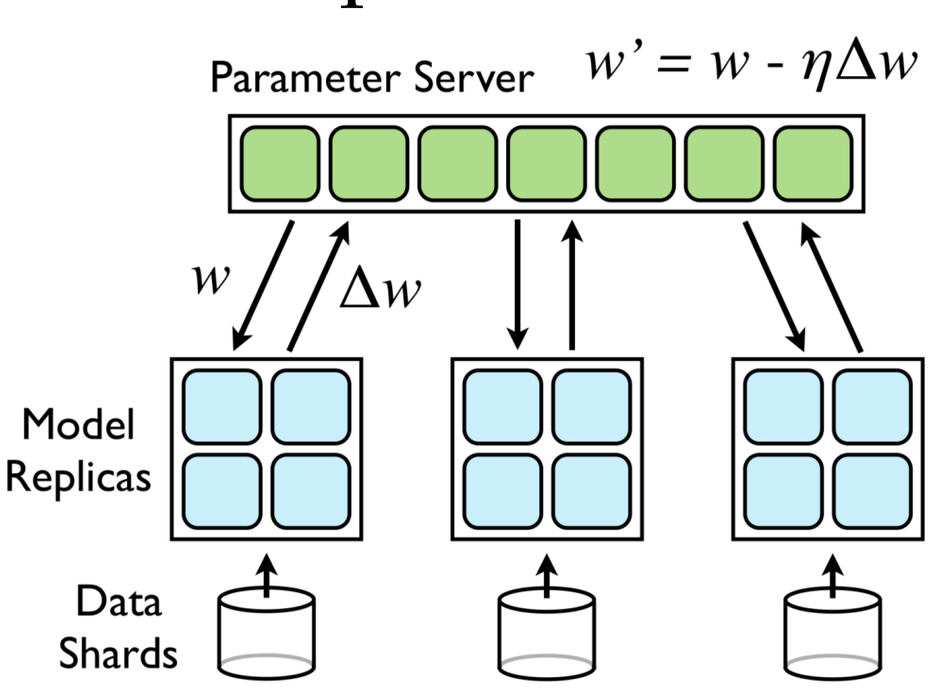
ALCHEM
alchem.usc.edu

HyPar: communication model

Two default parallelism:

- Data parallelism:

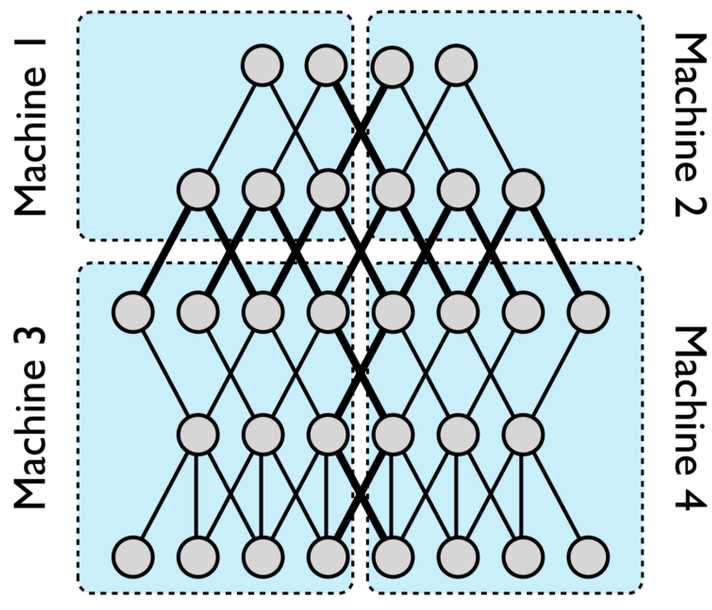
data partitioned



(NIPS'12)

- Model parallelism:

model partitioned



CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

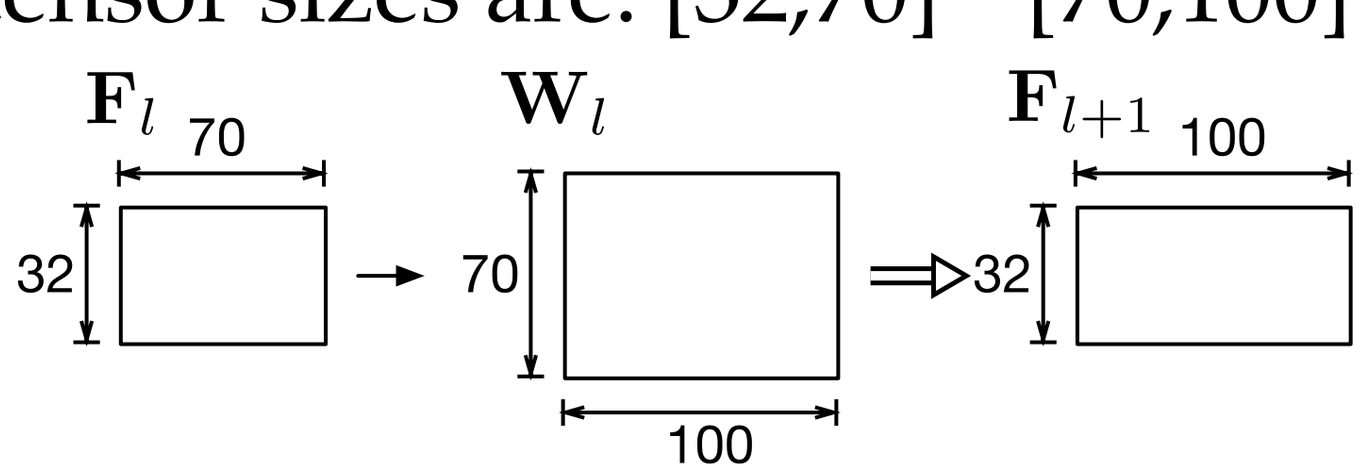
HyPar: communication model

Where is the communication from? What's the amount?

- data parallelism:

Assume we are computing matrix multiplication with two accelerators.

The tensor sizes are: [32,70] * [70,100] = [32,100]

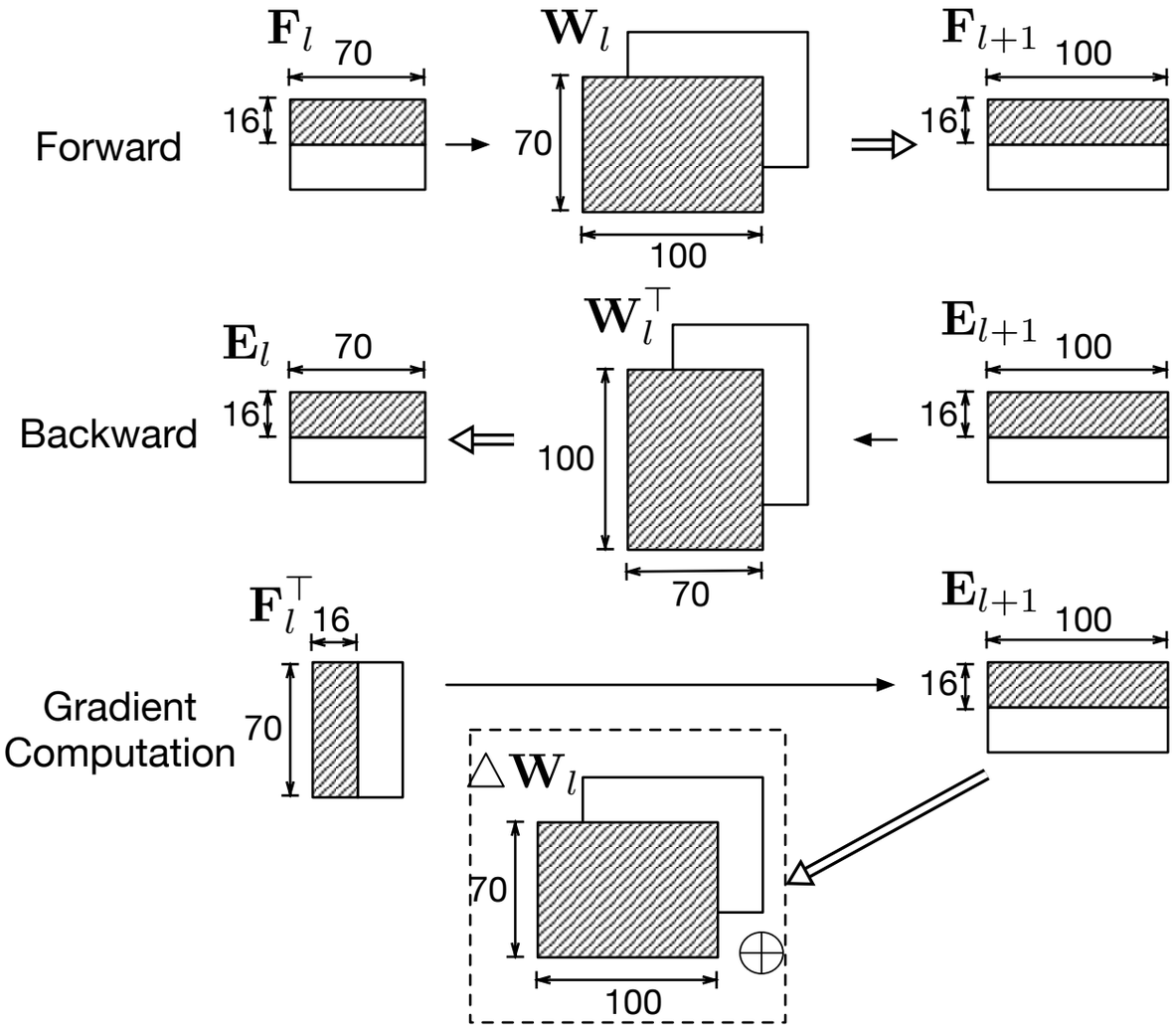


Tensors are partitioned and assigned to two accelerators.

[16,70] * [70,100] = [16,100]

HyPar: communication model

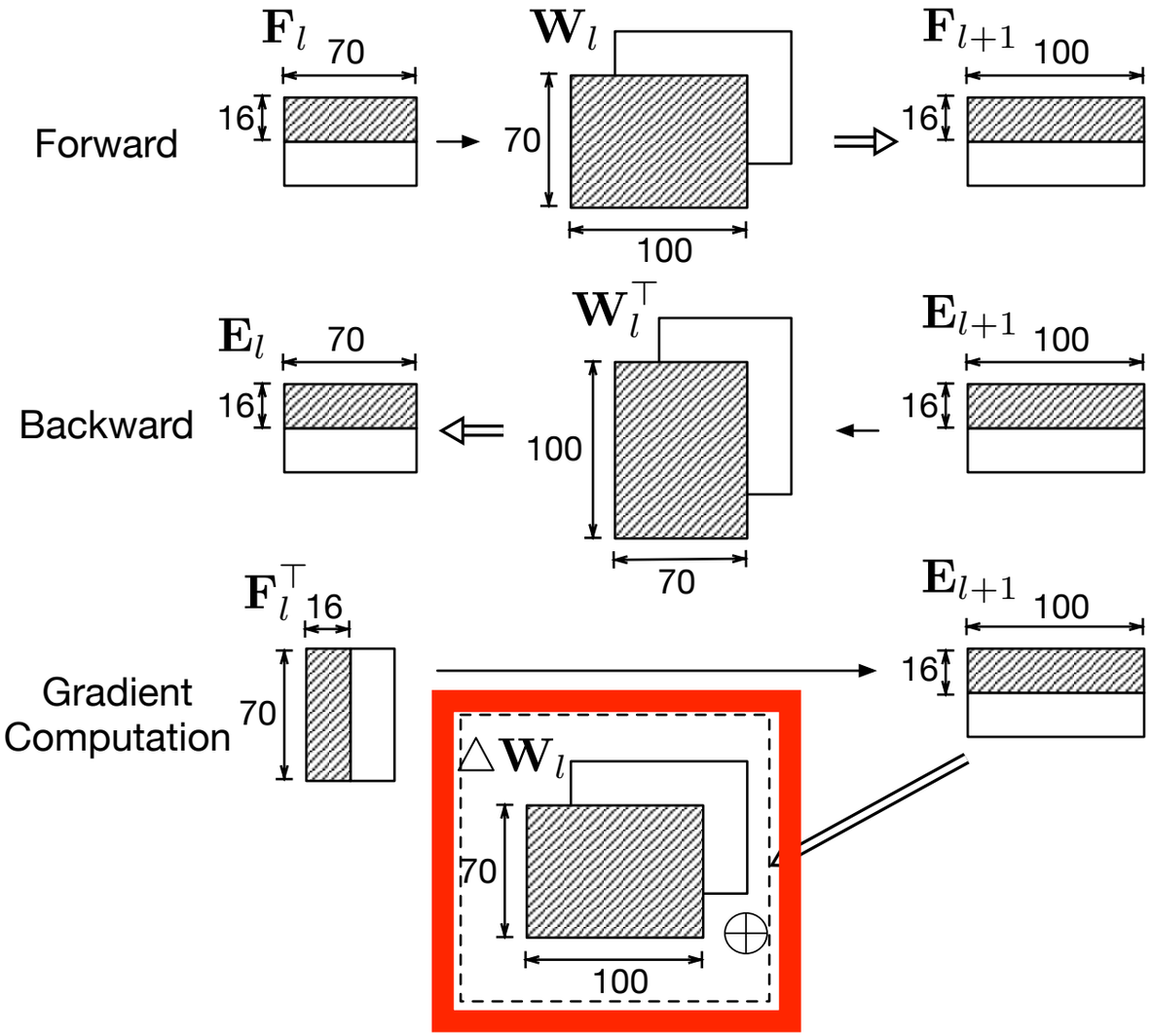
- data parallelism:



$$[16,70] * [70,100] = [16,100]$$
$$[16,100] * [100,70] = [16,70]$$
$$[70,16] * [16,100] = ?[70,100]$$

HyPar: communication model

- data parallelism:



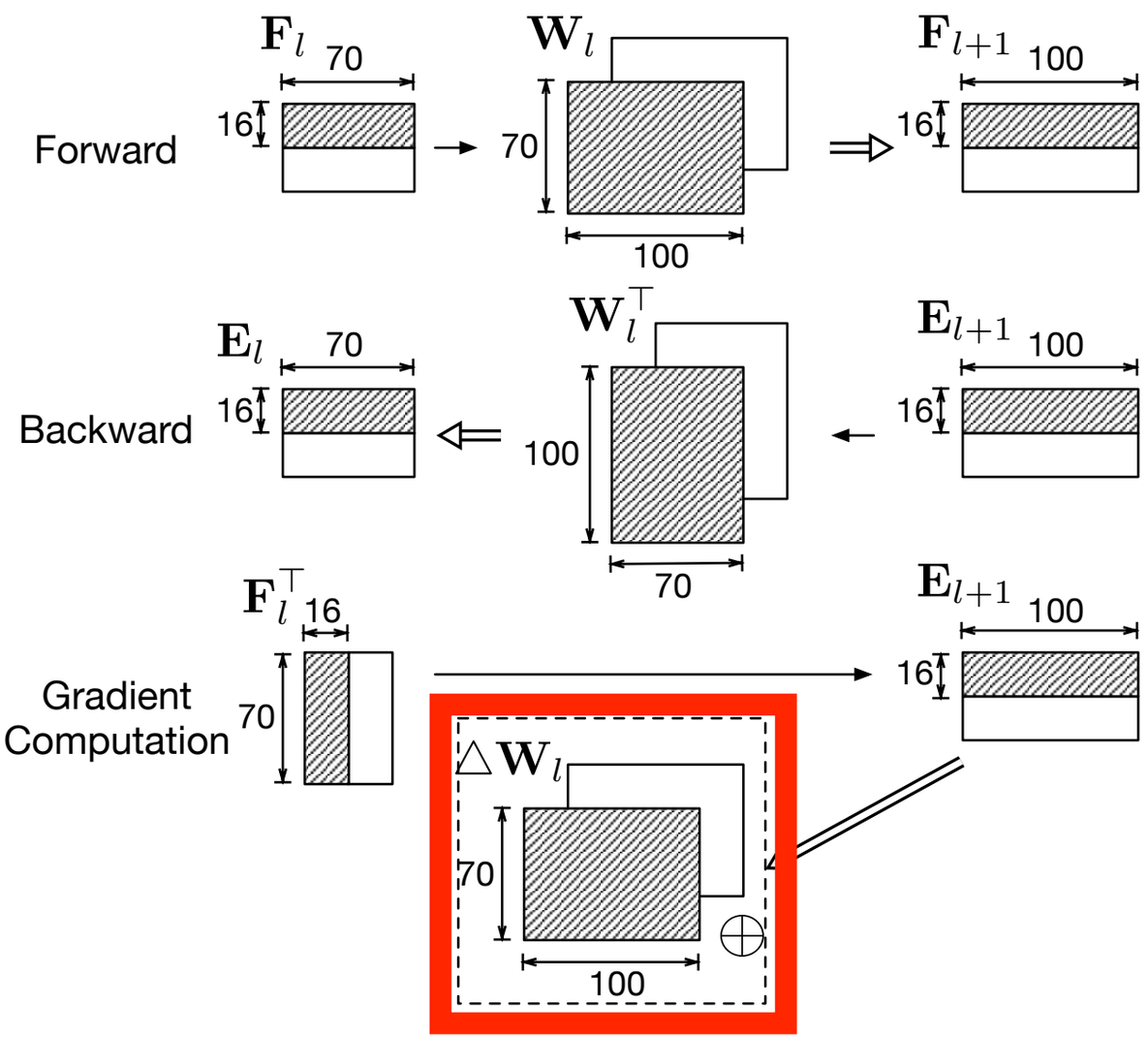
[16,70] * [70,100] = [16,100]

[16,100] * [100,70] = [16,70]

[70,16] * [16,100] = ?[70,100]

HyPar: communication model

- data parallelism:

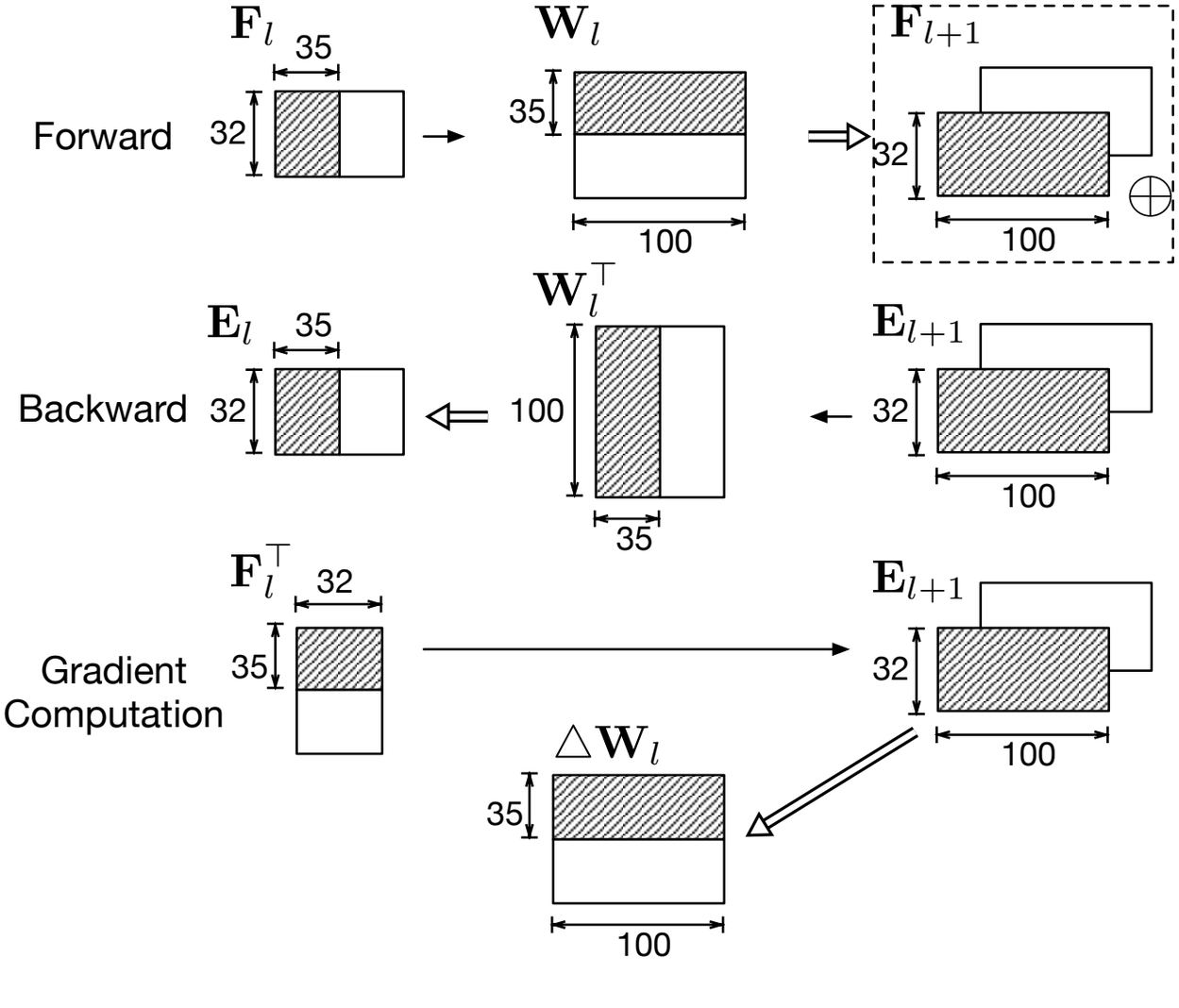


[16,70] * [70,100] = [16,100]

[16,100] * [100,70] = [16,70]

[70,16] * [16,100] = ?[70,100]

- model parallelism:



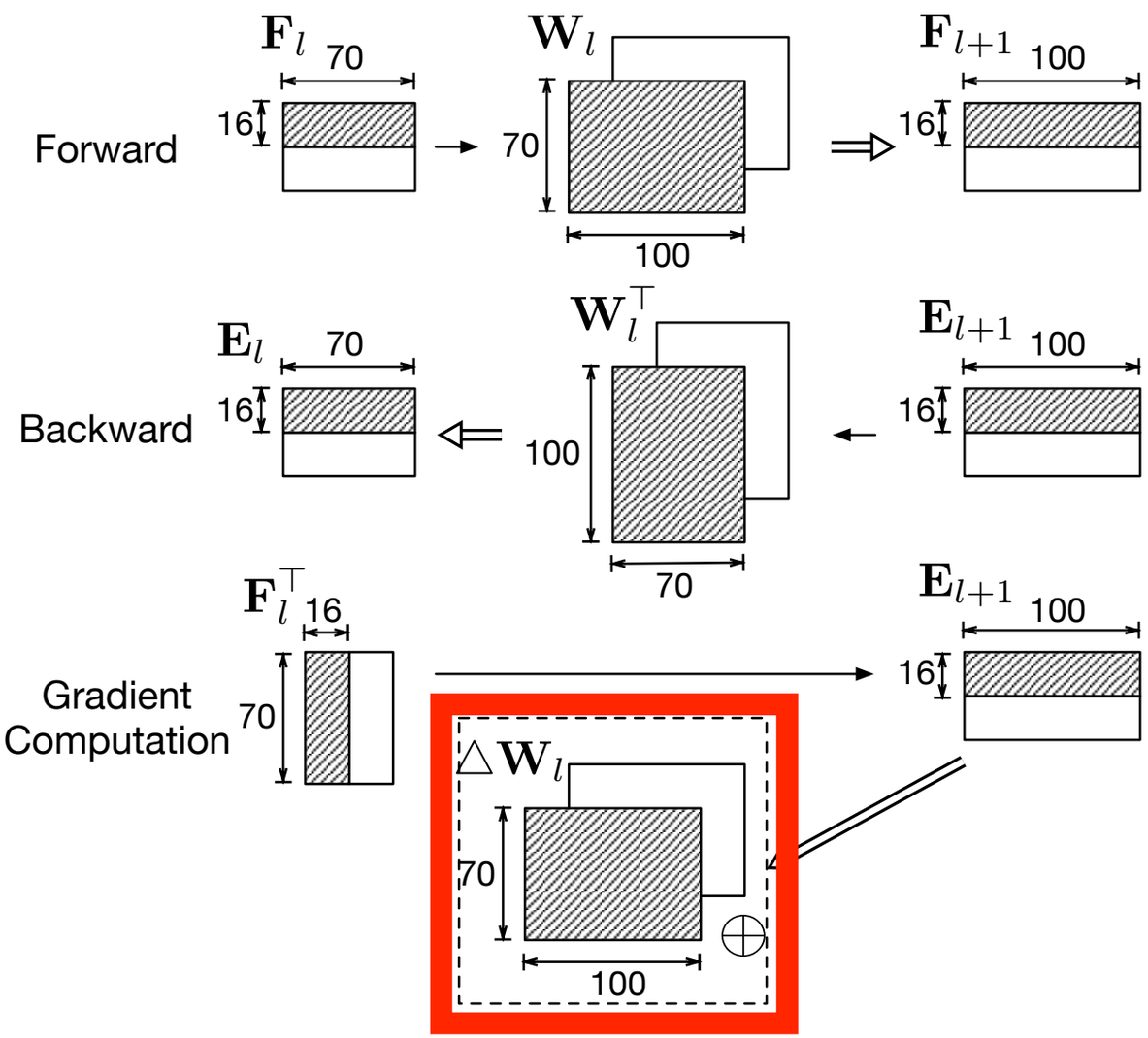
[32,35] * [35,100] = ?[32,100]

[32,100] * [100,35] = [32,35]

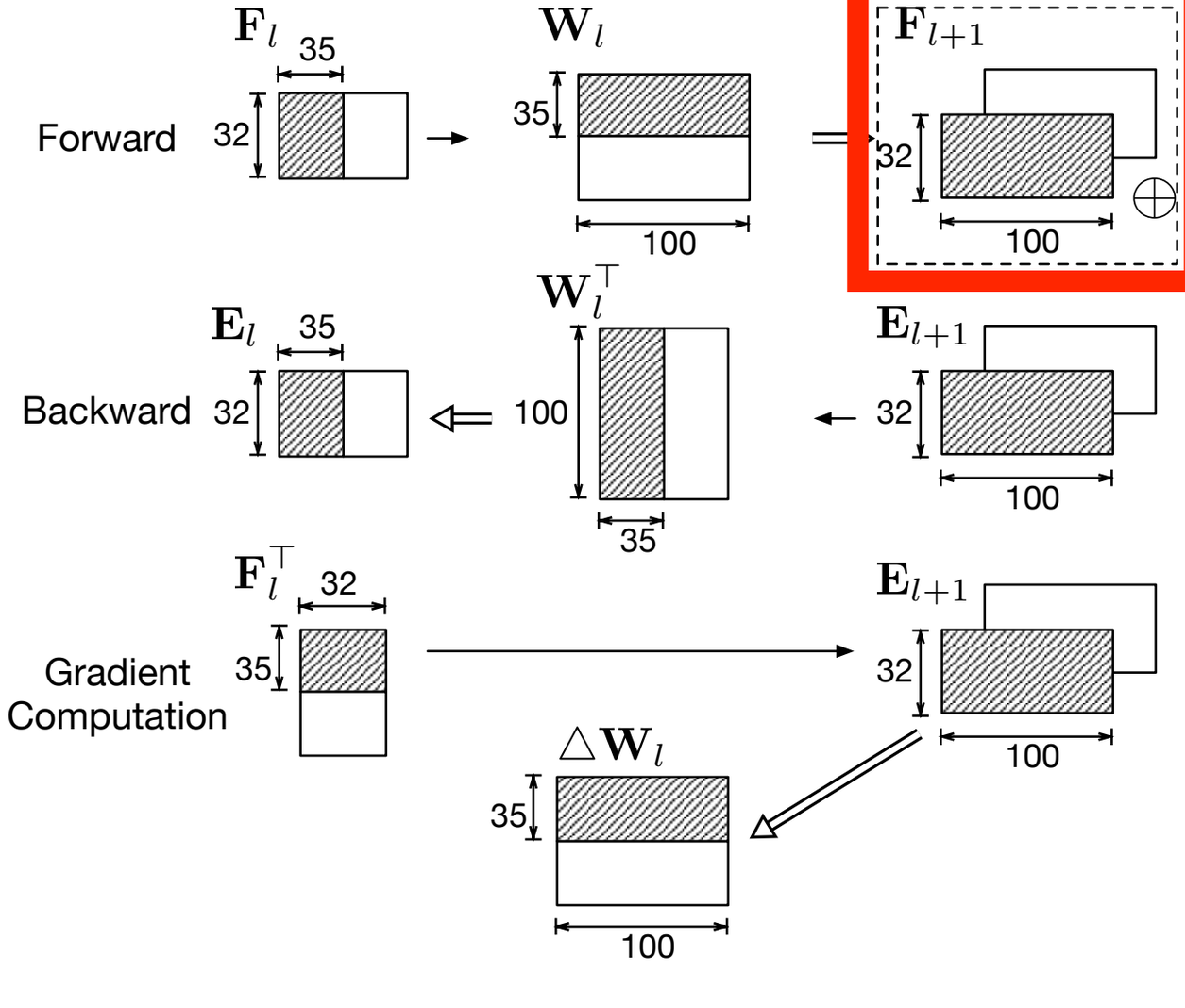
[35,32] * [32,100] = [35,100]

HyPar: communication model

- data parallelism:



- model parallelism:



[16,70] * [70,100] = [16,100]

[16,100] * [100,70] = [16,70]

[70,16] * [16,100] = ?[70,100]

[32,35] * [35,100] = ?[32,100]

[32,100] * [100,35] = [32,35]

[35,32] * [32,100] = [35,100]

HyPar: intra-layer com. model

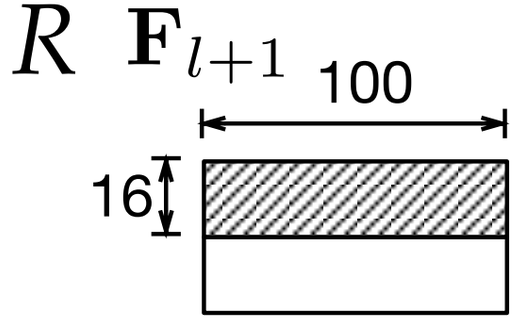
- Where: partial sum accumulation.
- Amount?

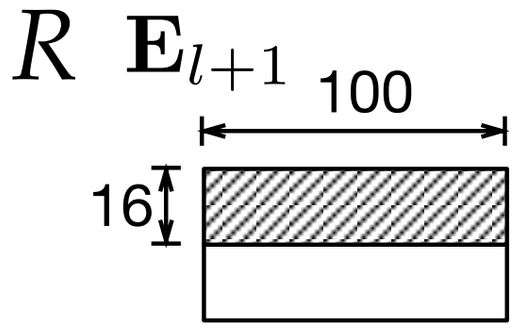
data parallelism	model parallelism
$\mathbb{A}(\triangle \mathbf{W}_l)$	$\mathbb{A}(\mathbf{F}_{l+1})$

HyPar: inter-layer com. model

- Where? Amount?

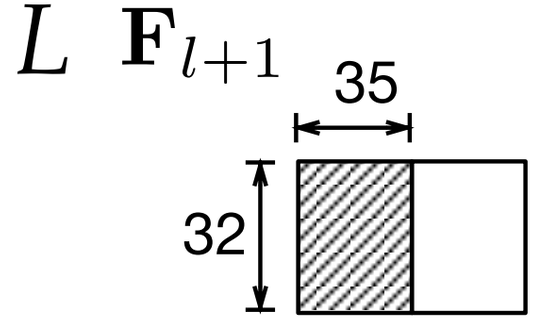
Layer L Output

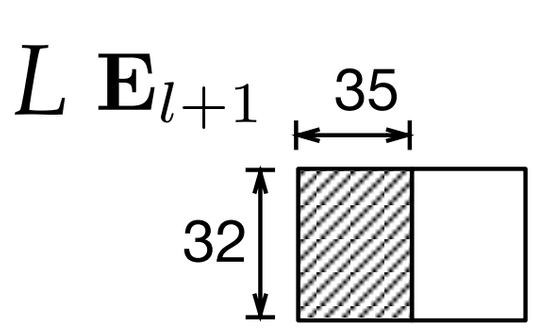




dp: data parallelism

Layer L+1 Input

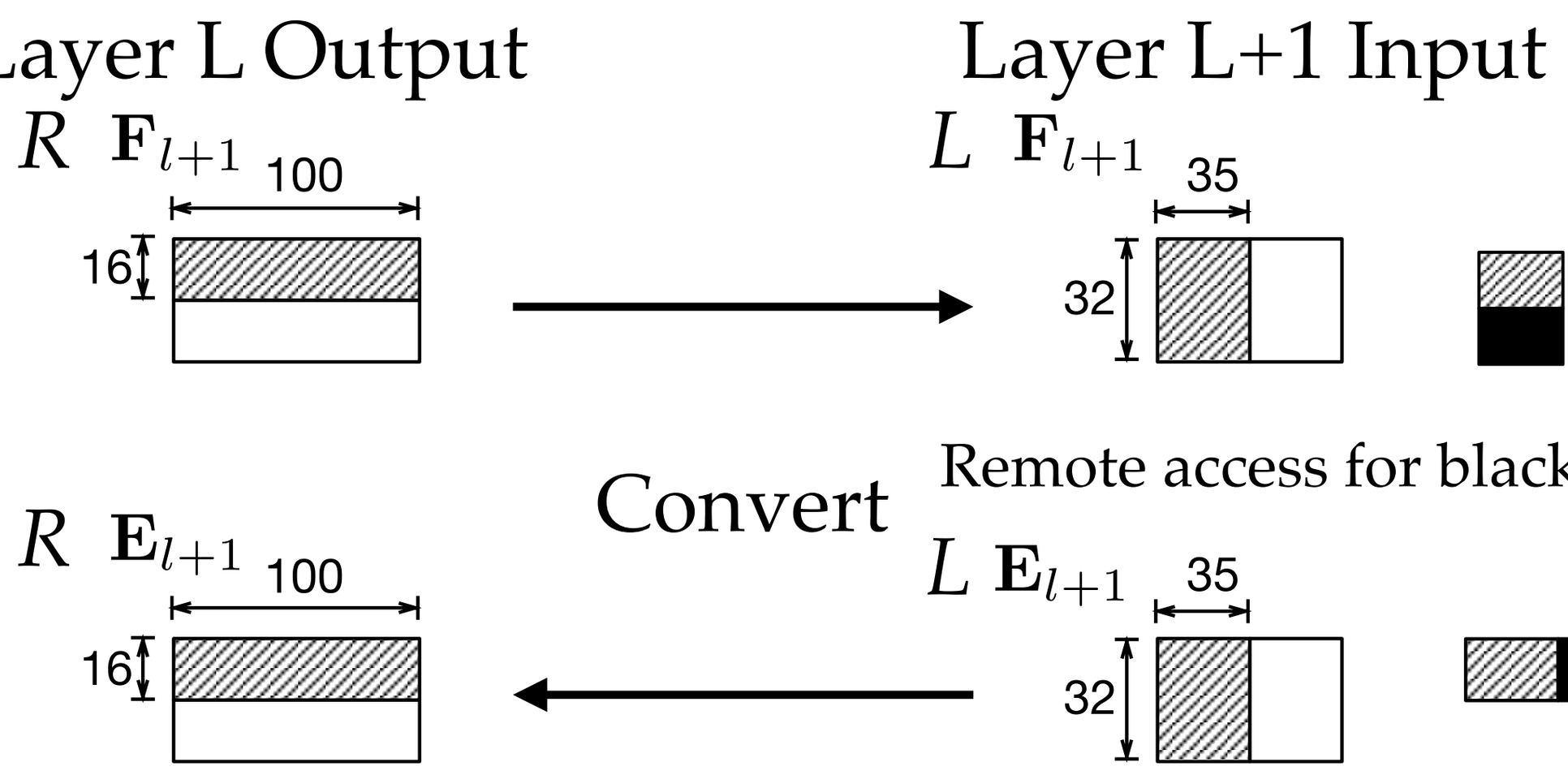




mp: model parallelism

HyPar: inter-layer com. model

- Where? Amount?

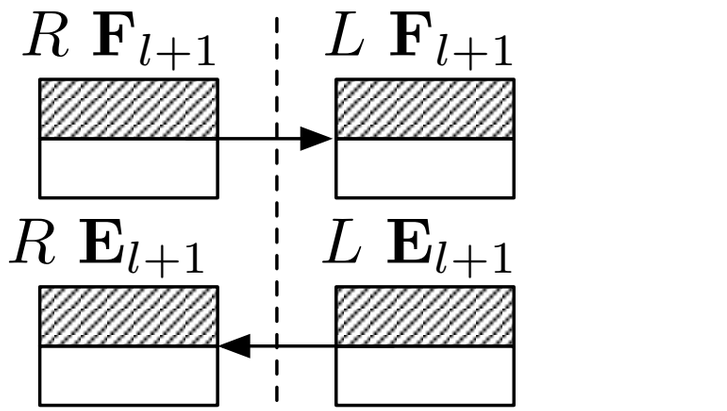


dp: data parallelism mp: model parallelism

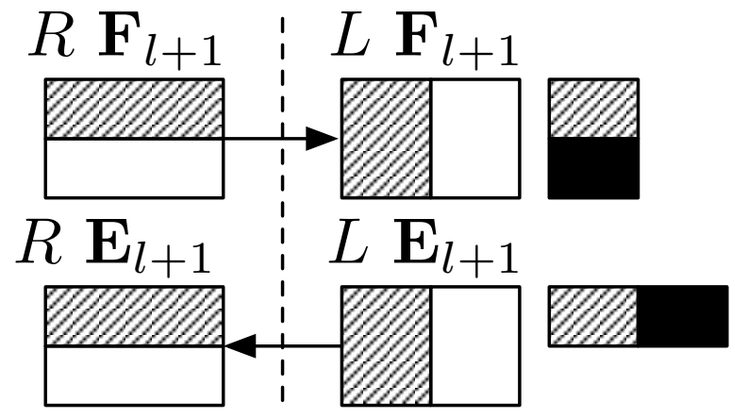
HyPar: inter-layer com. model

- Where: convert tensors between two layers.
- Amount?

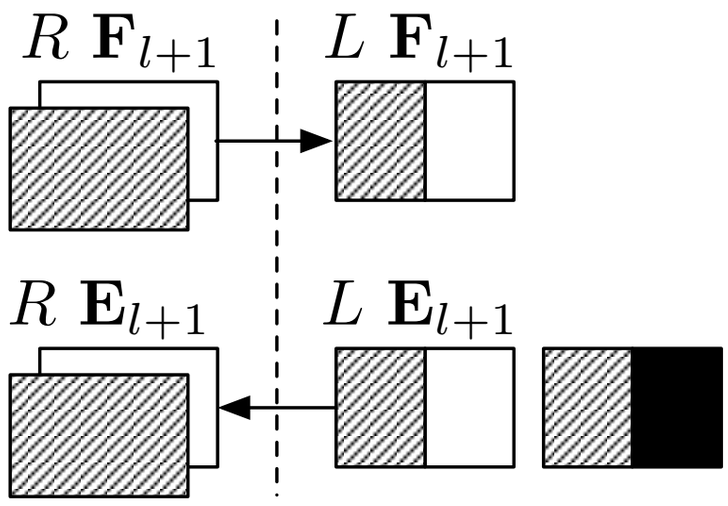
dp-dp	0
dp-mp	$0.25\mathbb{A}(\mathbf{F}_{l+1}) + 0.25\mathbb{A}(\mathbf{E}_{l+1})$
mp-mp	$0.5\mathbb{A}(\mathbf{E}_{l+1})$
mp-dp	$0.5\mathbb{A}(\mathbf{E}_{l+1})$



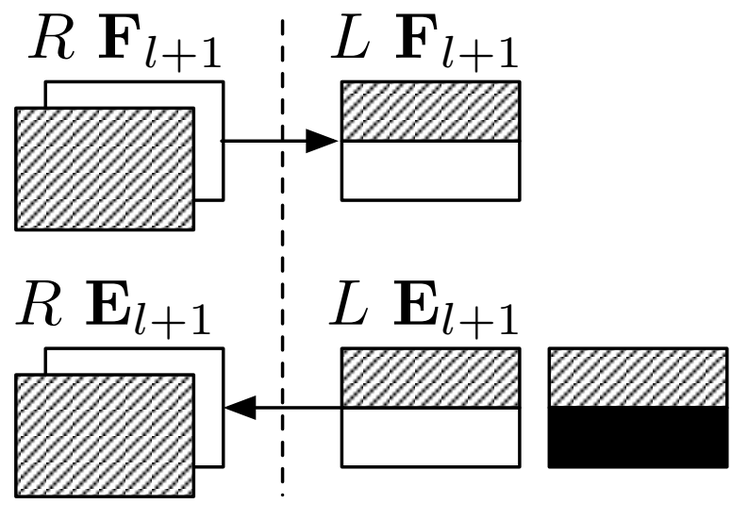
(a) dp-dp



(b) dp-mp



(c) mp-mp



(d) mp-dp

CEI
cei.pratt.duke.edu

ALCHEM
alchem.usc.edu

A slide added after the presentation

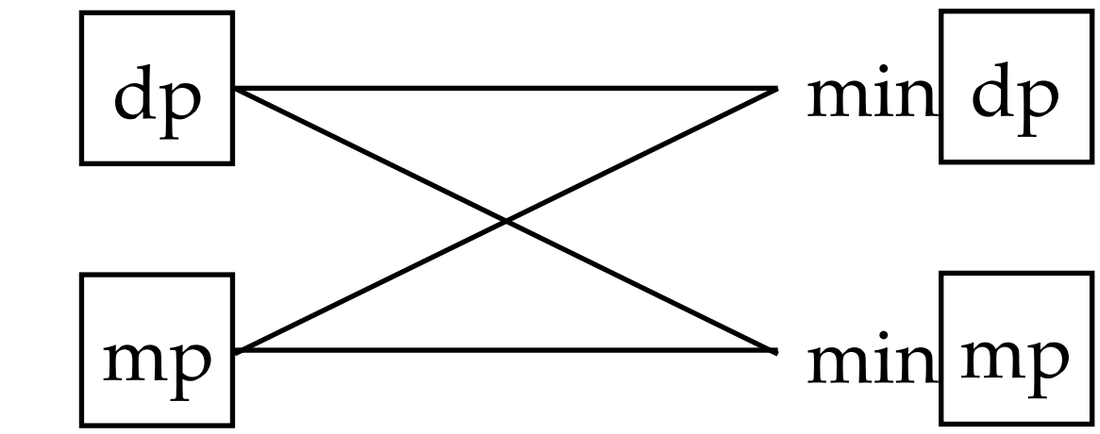
- For inference, we only need to use data parallelism for all layers, because
  - only forward is performed in inference, and no intra-layer communication in dp forward.
  - no inter-layer communication in dp forward

Outline

- The development of deep learning accelerators
- Why HyPar
- HyPar: hybrid parallelism for deep learning accelerator array
  - Communication model
  - Tensor partition
  - Evaluation
- Conclusion

HyPar tensor partition: which parallelism to use?

- To find the partitioning which incurs less communication.
- How to find that? Brute Force?? Dynamic programing!



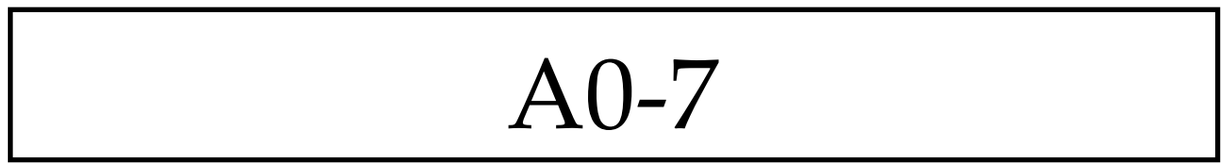
compute intra and inter layer communication using the communication model.

dp: data parallelism

mp: model parallelism

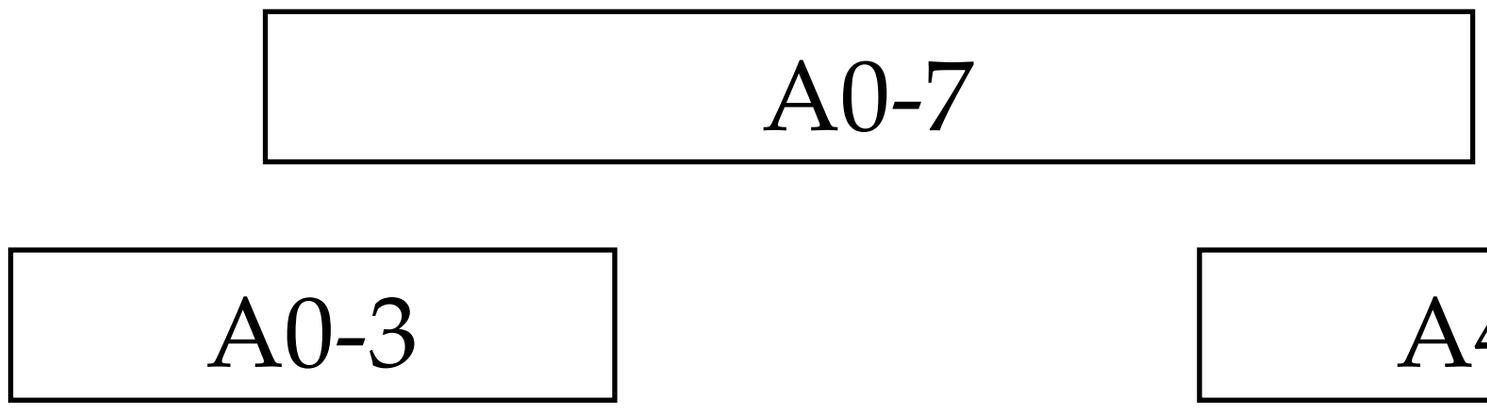
HyPar: hierarchical partition

- We know that for 2 accelerators, how about 4, 8, 16, 32?
  —Hierarchical partition.



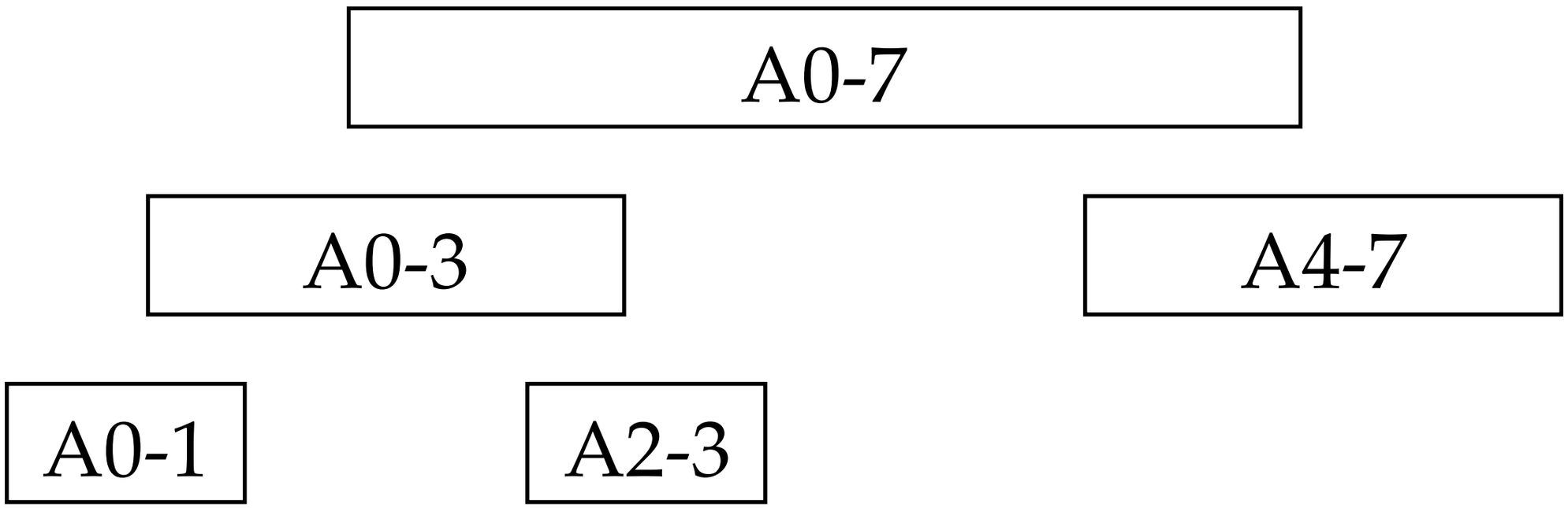
HyPar: hierarchical partition

- We know that for 2 accelerators, how about 4, 8, 16, 32?
  —Hierarchical partition.



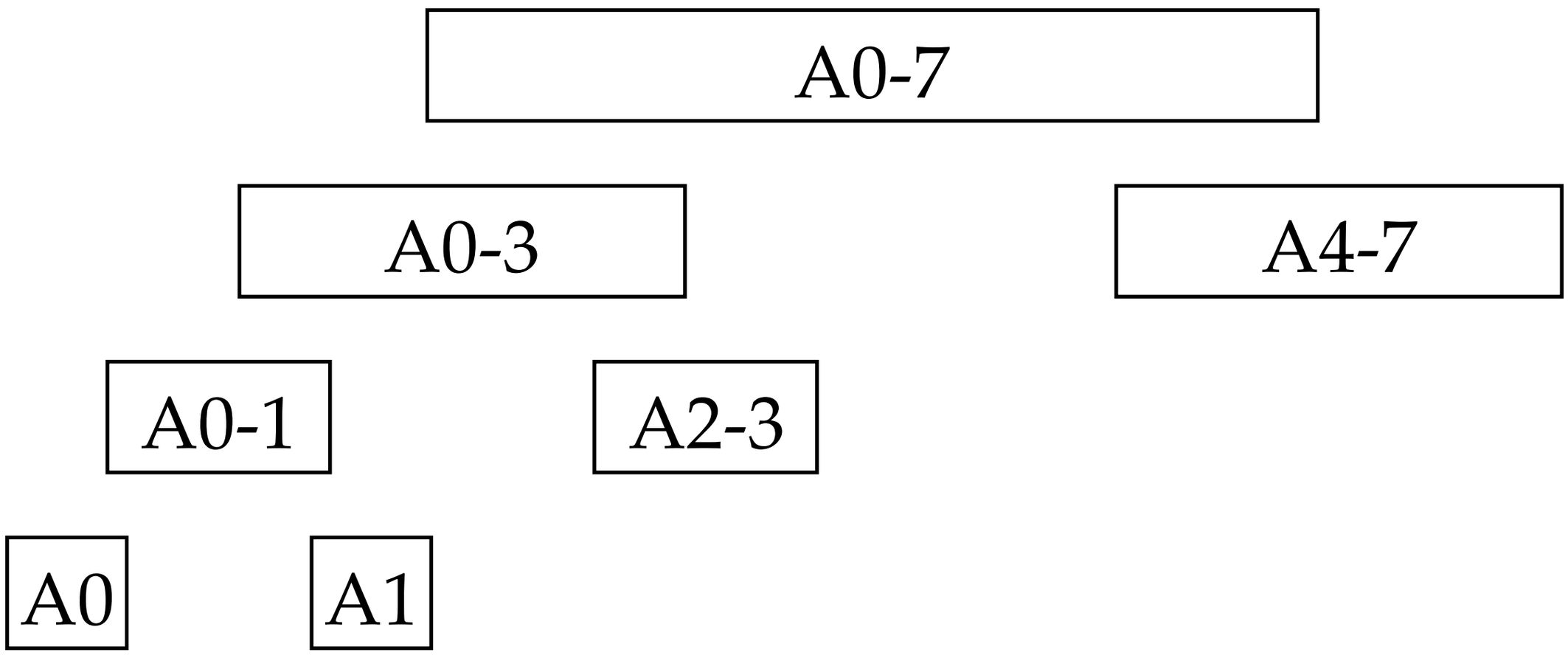
HyPar: hierarchical partition

- We know that for 2 accelerators, how about 4, 8, 16, 32?
  —Hierarchical partition.



HyPar: hierarchical partition

- We know that for 2 accelerators, how about 4, 8, 16, 32?
  —Hierarchical partition.



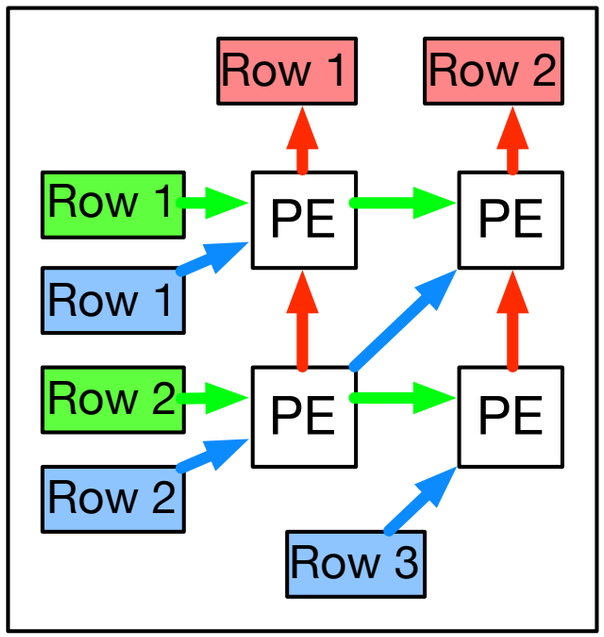
CEI
cei.pratt.duke.edu

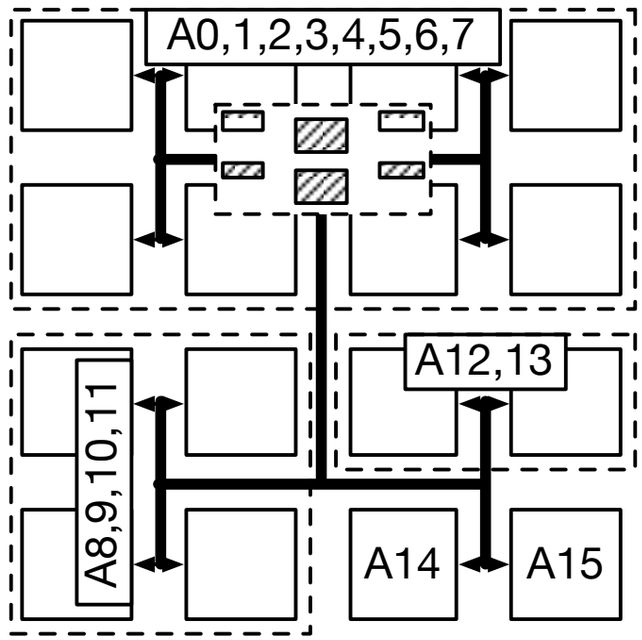
ALCHEM
alchem.usc.edu

Outline

- The development of deep learning accelerators
- Why HyPar
- HyPar: hybrid parallelism for deep learning accelerator array
  - Communication model
  - Tensor partition
  - Evaluation
- Conclusion

HyPar evaluation setup



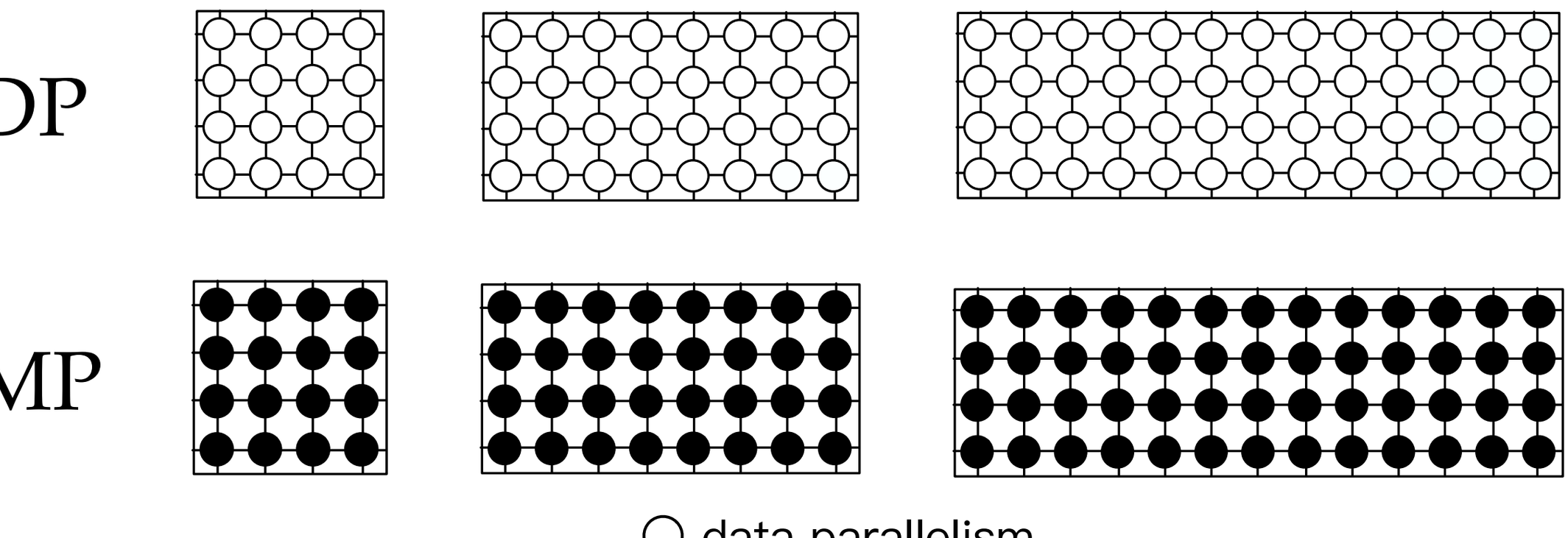


- 16 Eyeriss-like accelerators(4 hierarchies)
  - Total computation density: 1344 (=84*16)GOPS/s
  - Total link bandwidth: 25.6 (=1.6*8)Gb/s
- Ten benchmarks: from Lenet to VGGs.

HyPar parallelism

DP

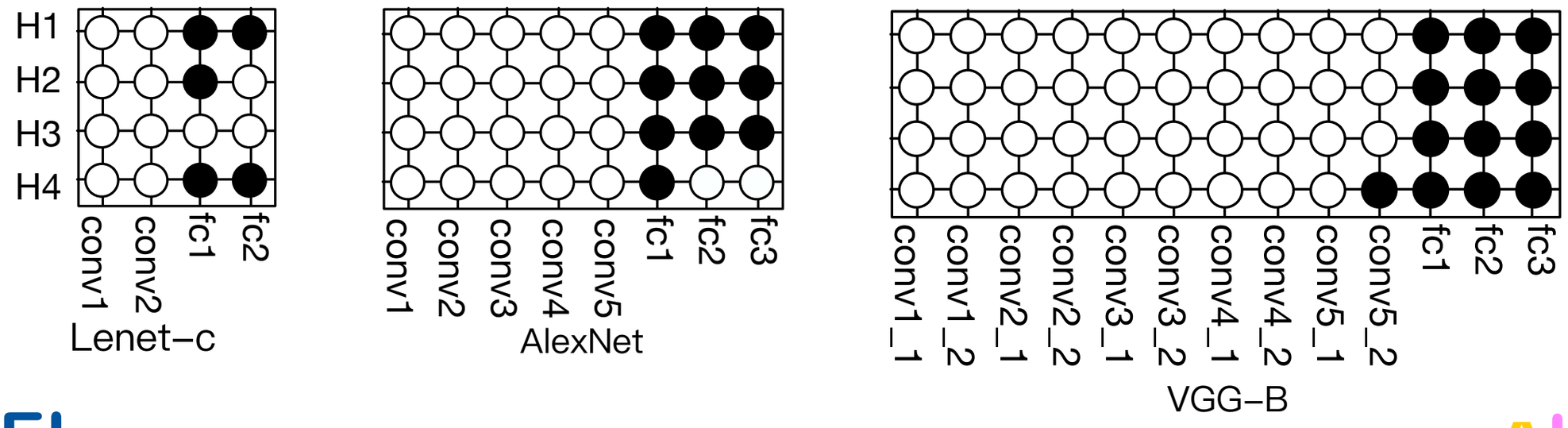
MP



○ data parallelism

● model parallelism

HyPar



CEI

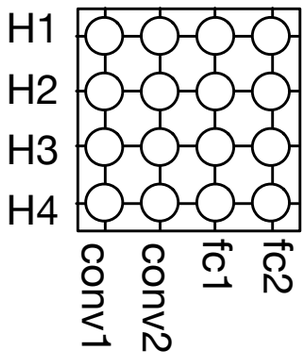
cei.pratt.duke.edu

ALCHEM

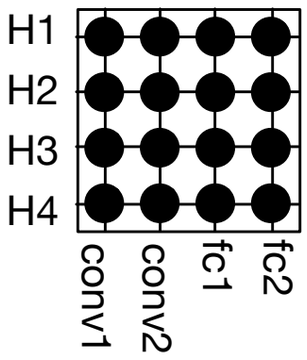
alchem.usc.edu

HyPar parallelism

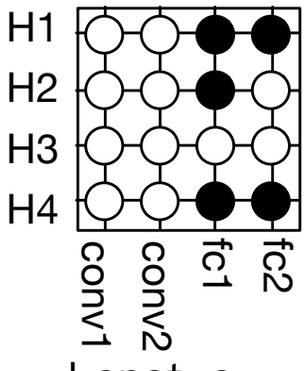
dp



mp



HyPar



Lenet-c

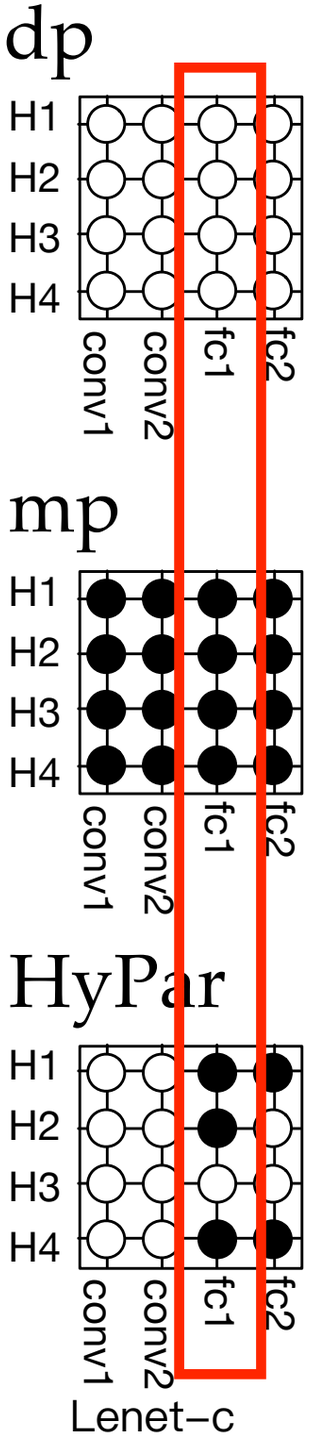
CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

HyPar parallelism



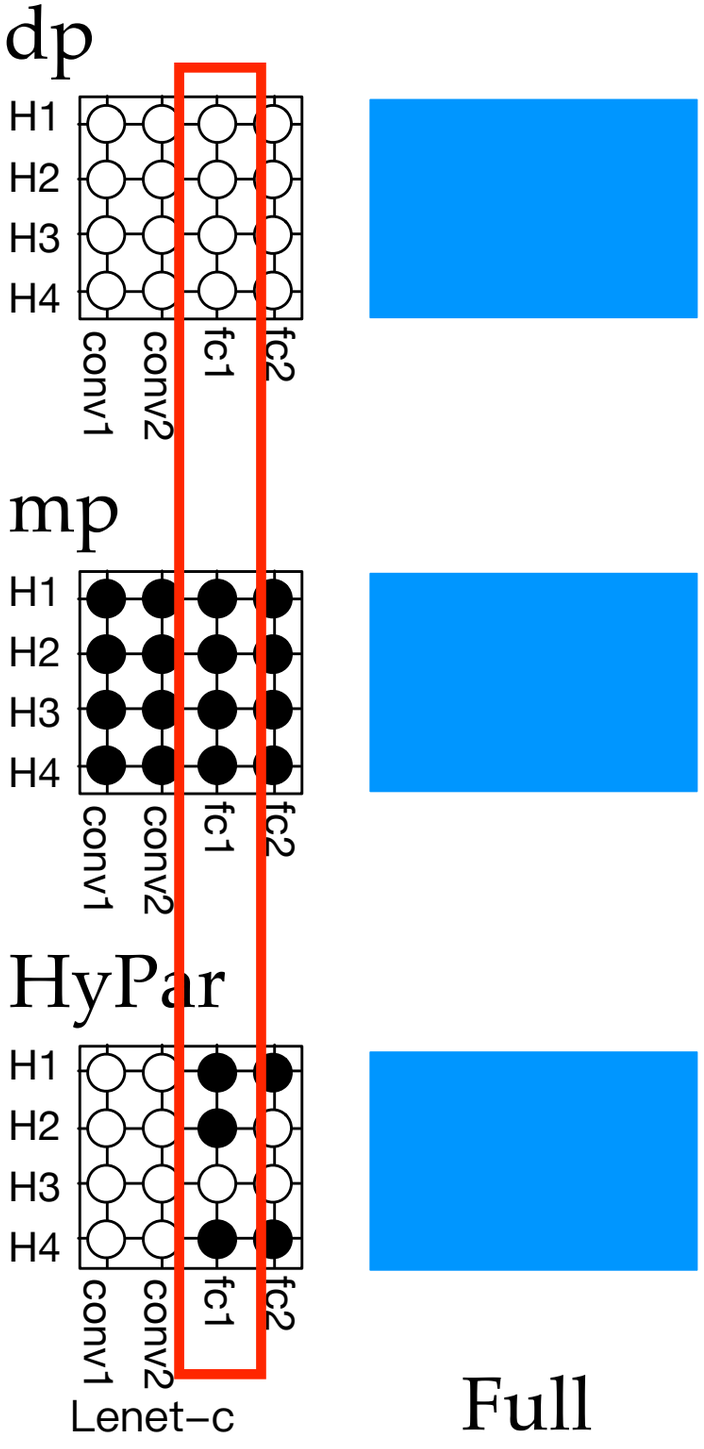
CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

HyPar parallelism



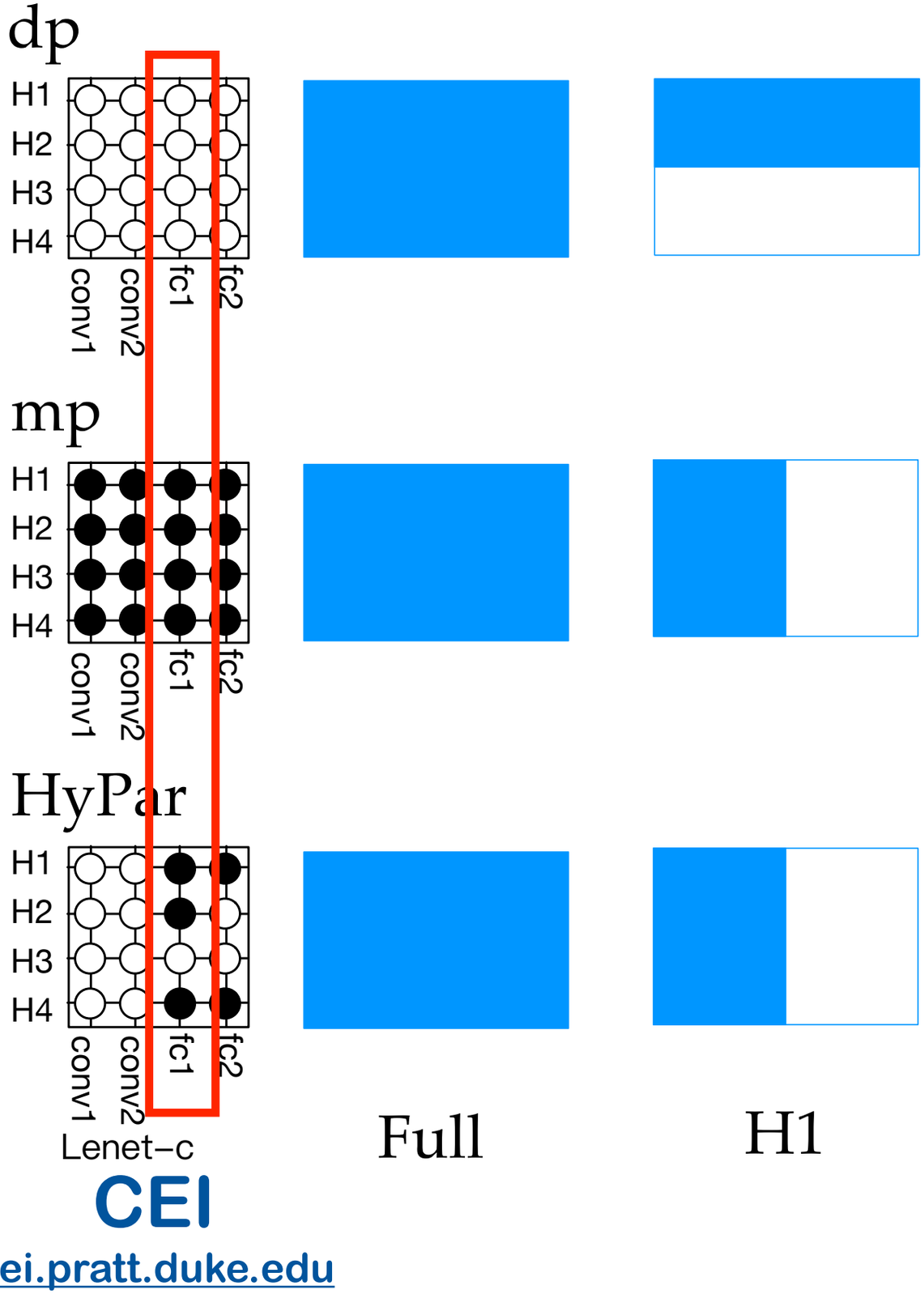
CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

HyPar parallelism



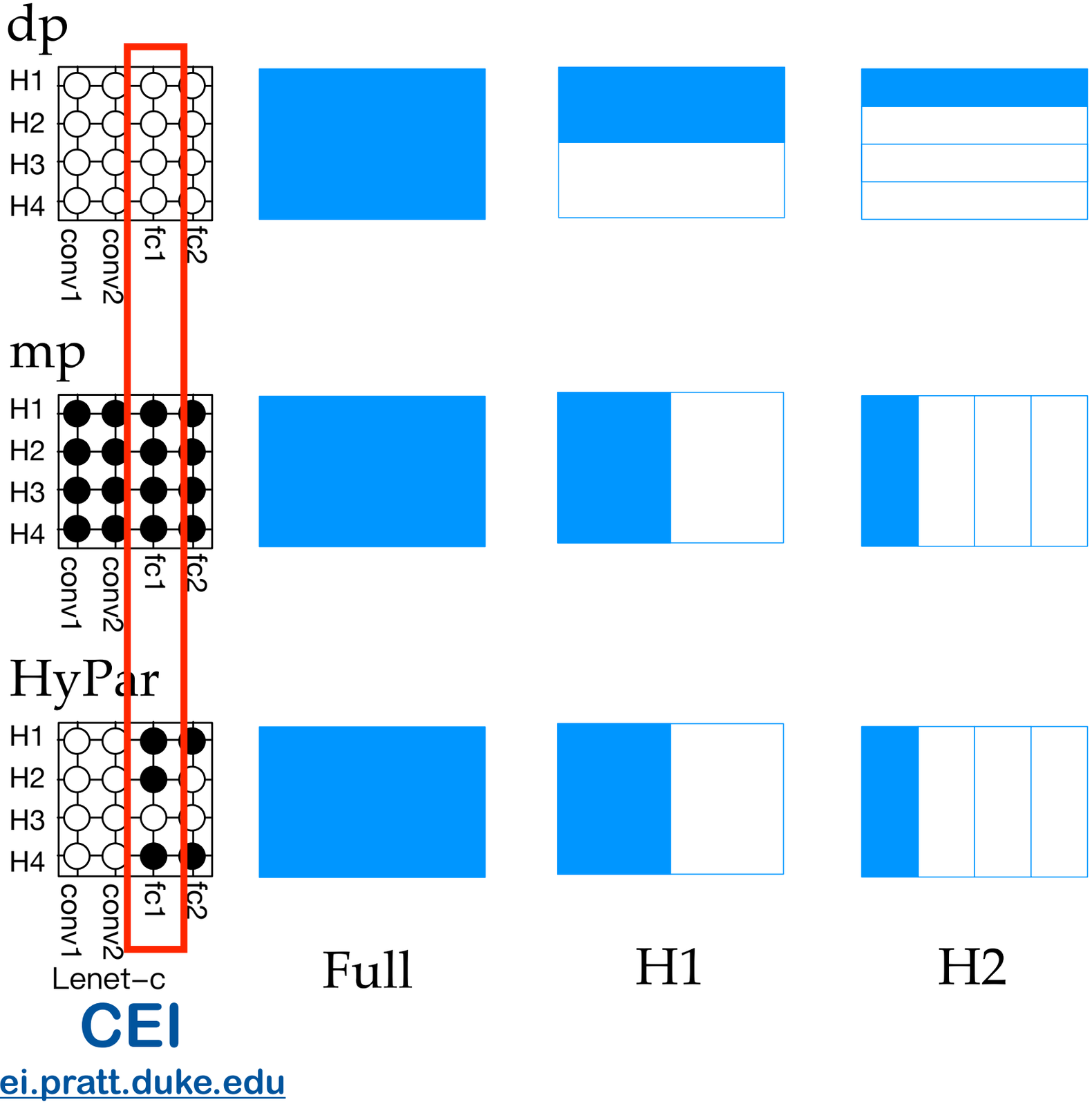
CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

HyPar parallelism



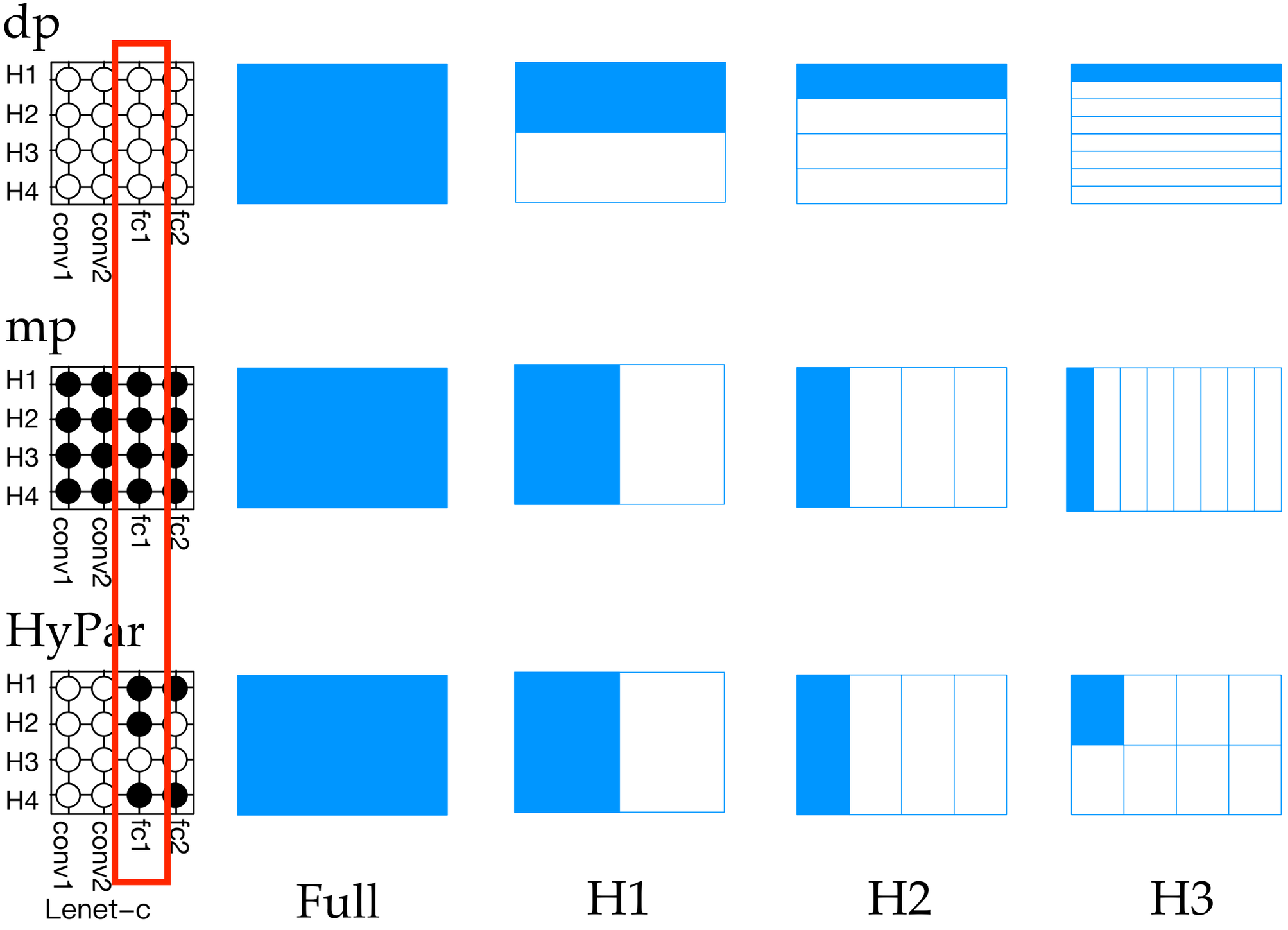
CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

HyPar parallelism



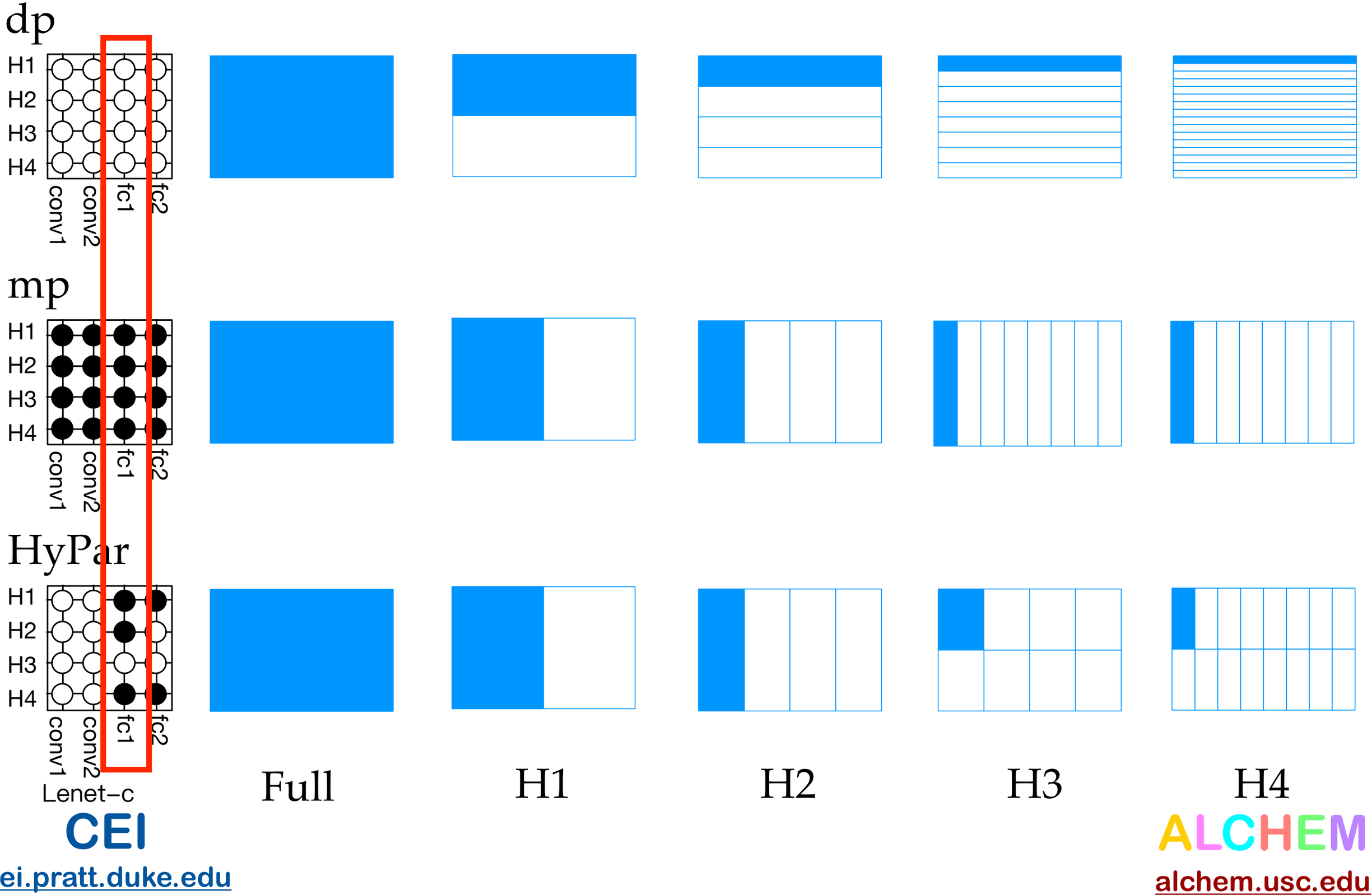
CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

HyPar parallelism



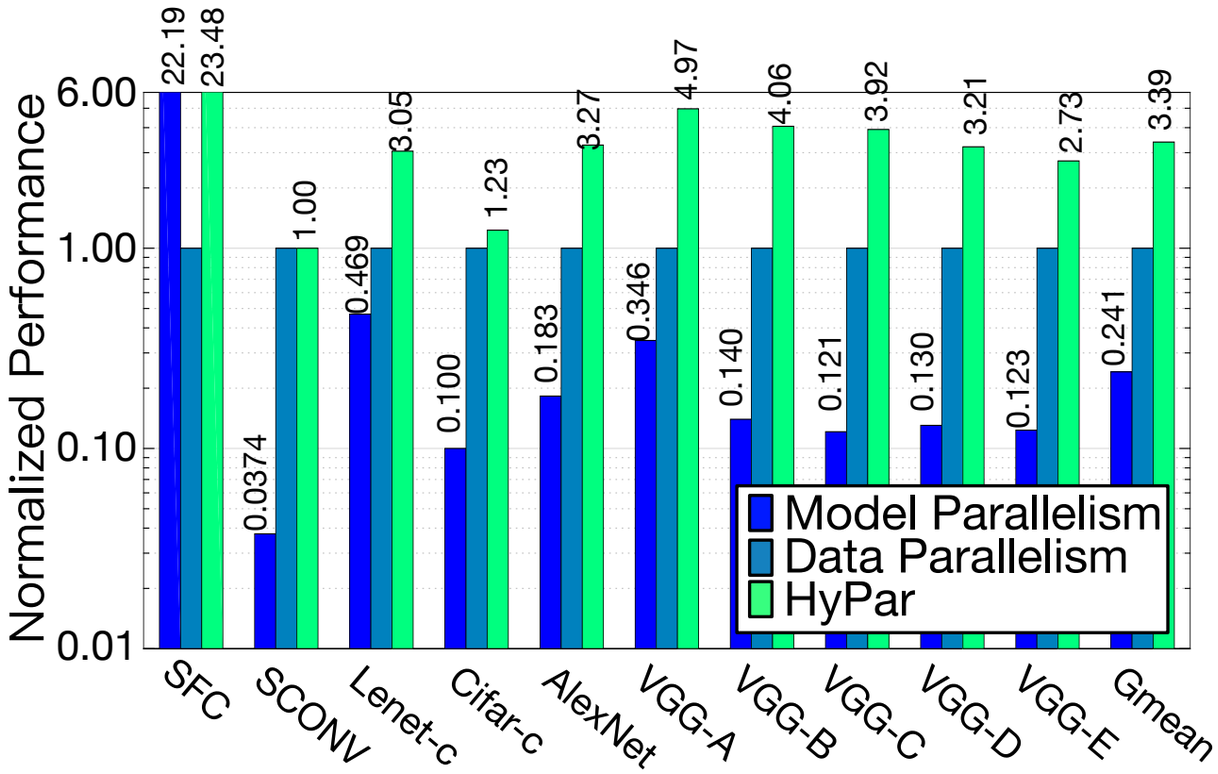
CEI

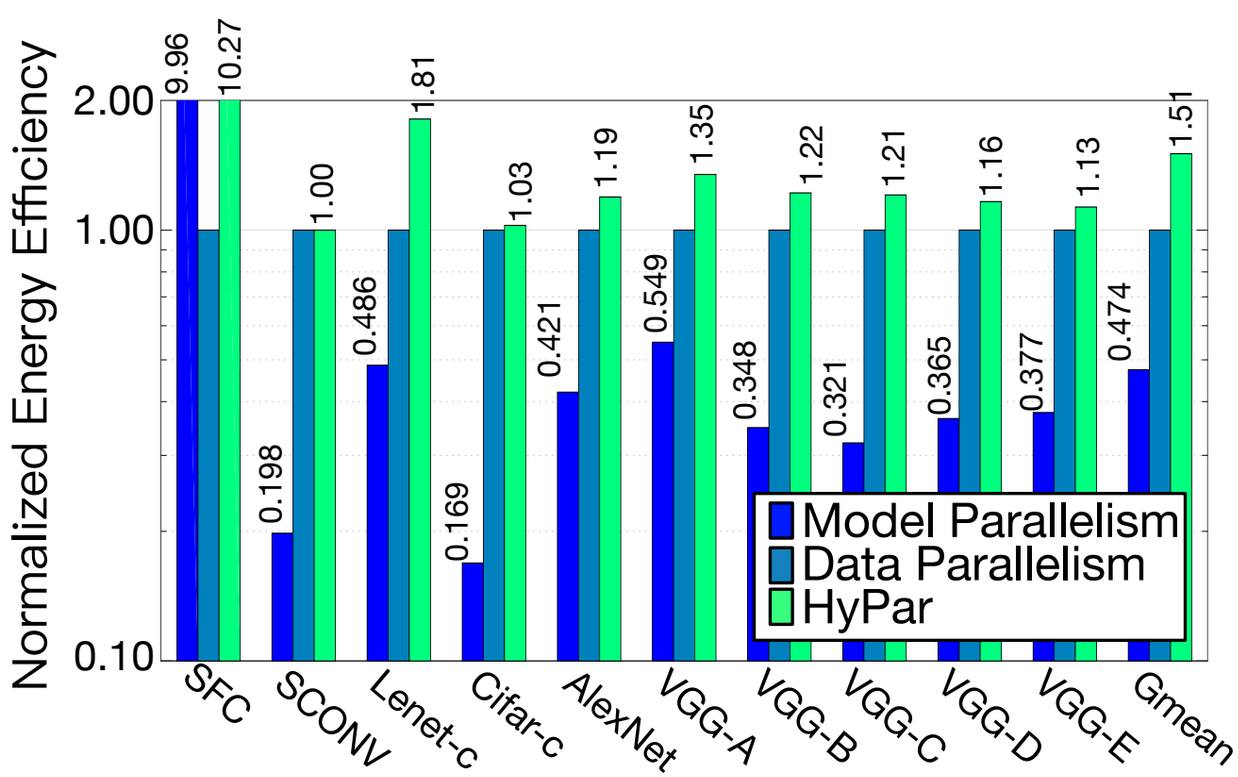
cei.pratt.duke.edu

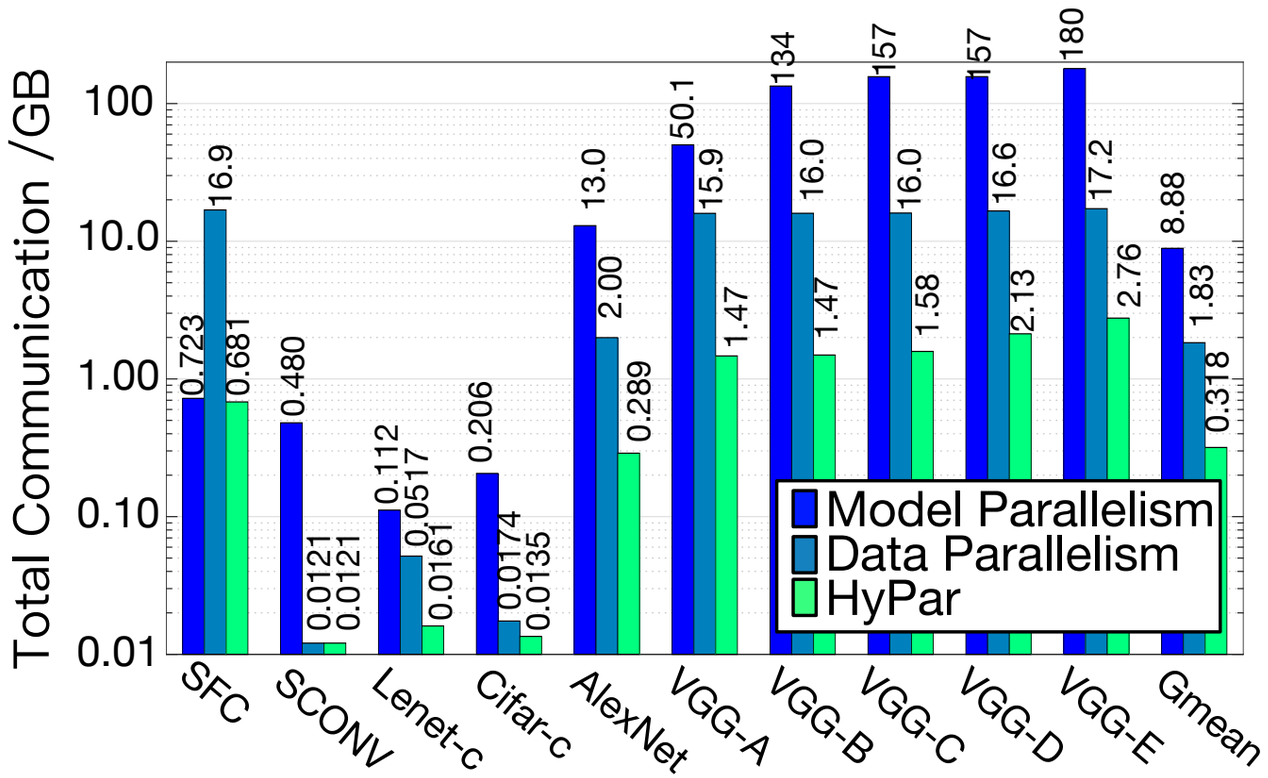
ALCHEM

alchem.usc.edu

HyPar: performance, energy efficiency & communication







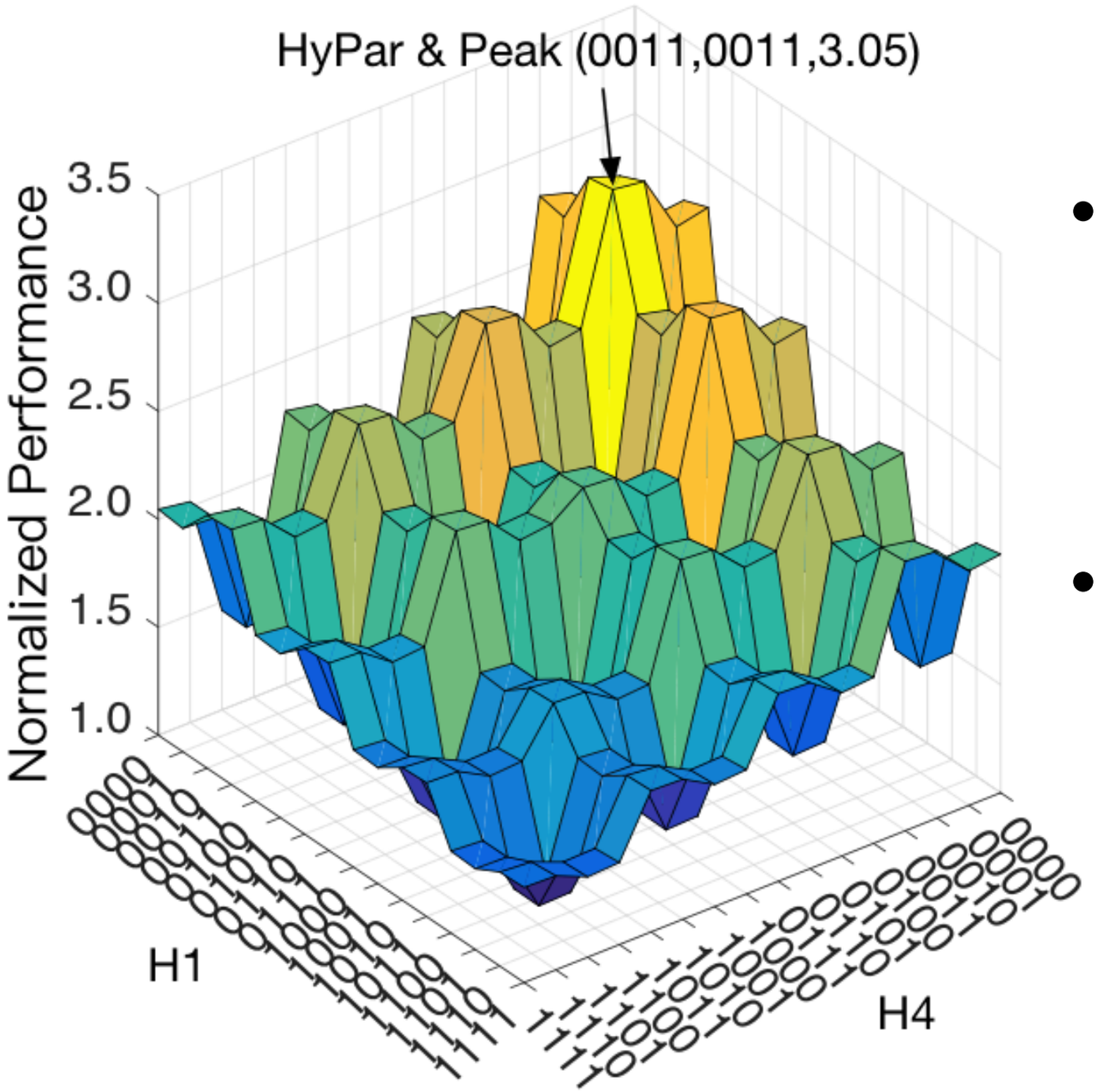
CEI

cei.pratt.duke.edu

ALCHEM

alchem.usc.edu

HyPar: parallelism space exploration for Lenet



- We enumerated all 256 possible combinations for H1 and H4.
- HyPar got the maximum as brute force search.

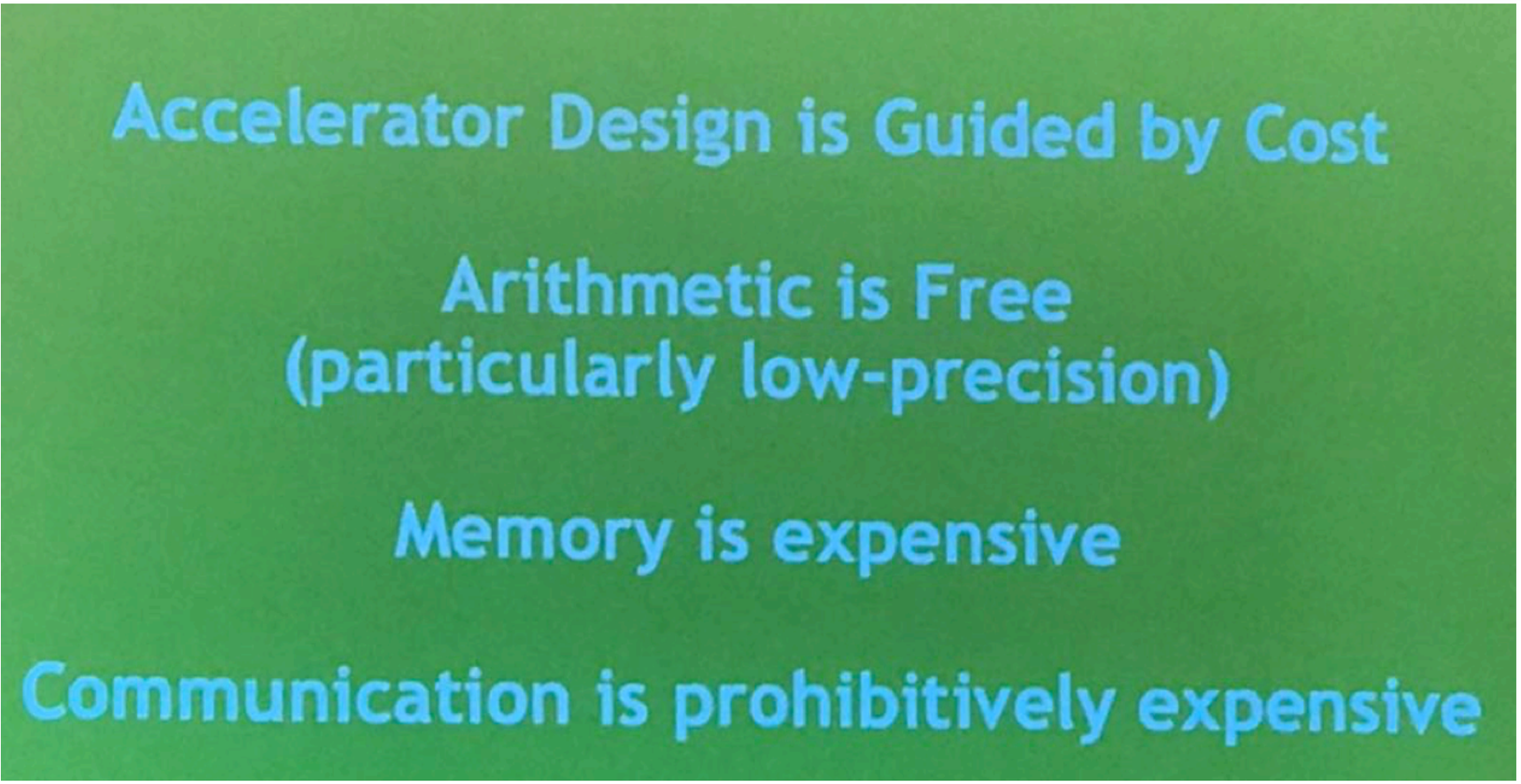
Outline

- The development of deep learning accelerators
- Why HyPar
- HyPar: hybrid parallelism for deep learning accelerator array
  - Communication model
  - Tensor partition
  - Evaluation
- Conclusion

Conclusion

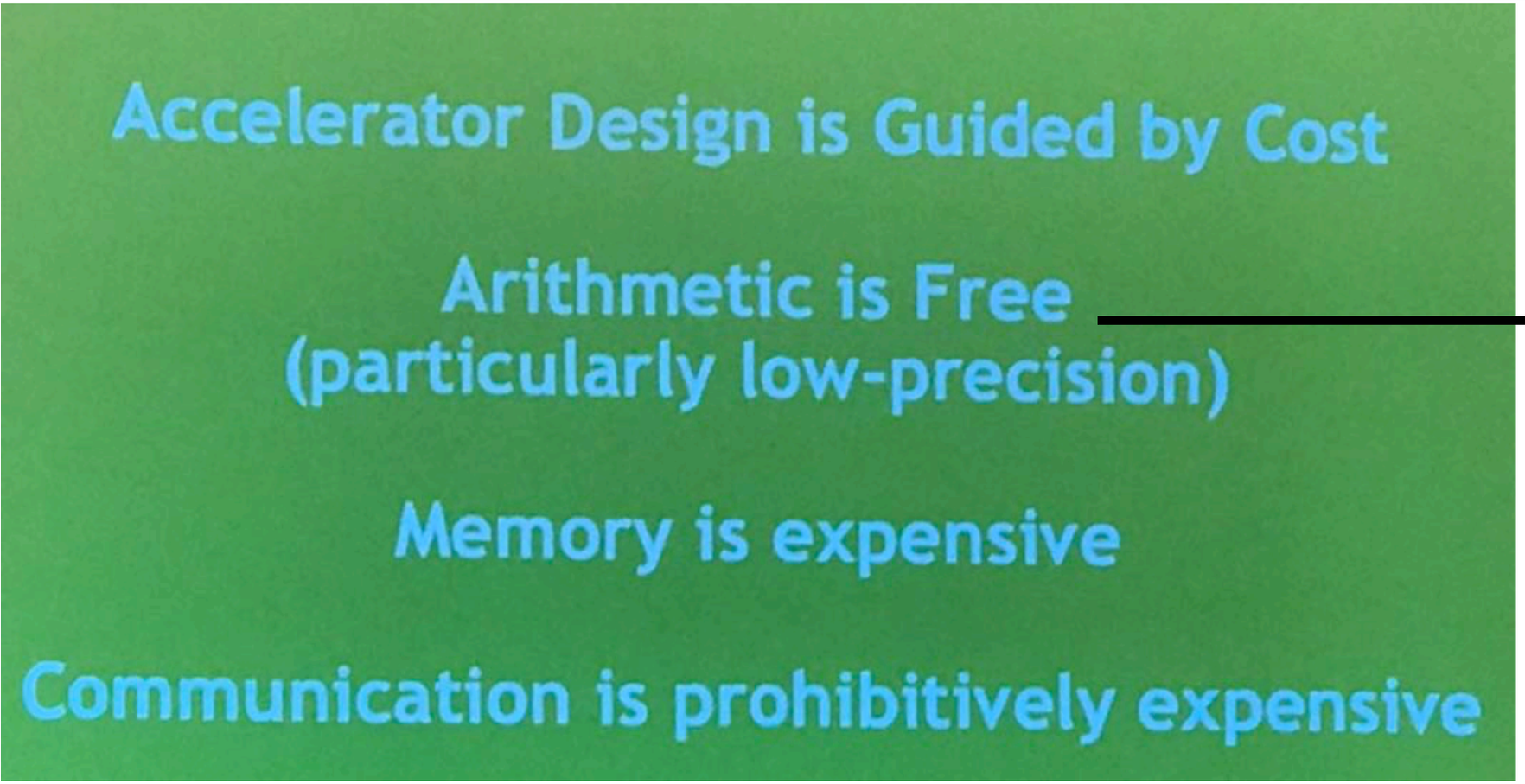
- Optimization within an accelerator is reaching its ceiling.
- Communication cost is high, worth our attention.
- Models to understand communication sources in accelerator array:
  - Intra-layer communication
  - Inter-layer communication
- Dynamic programing and hierarchical partition to minimize the total communication.
- Hybrid parallelism reduces the total communication.

Discussion: What's next for deep learning accelerators?



Bill Dally@AACBB(Sat.)

Discussion: What's next for deep learning accelerators?



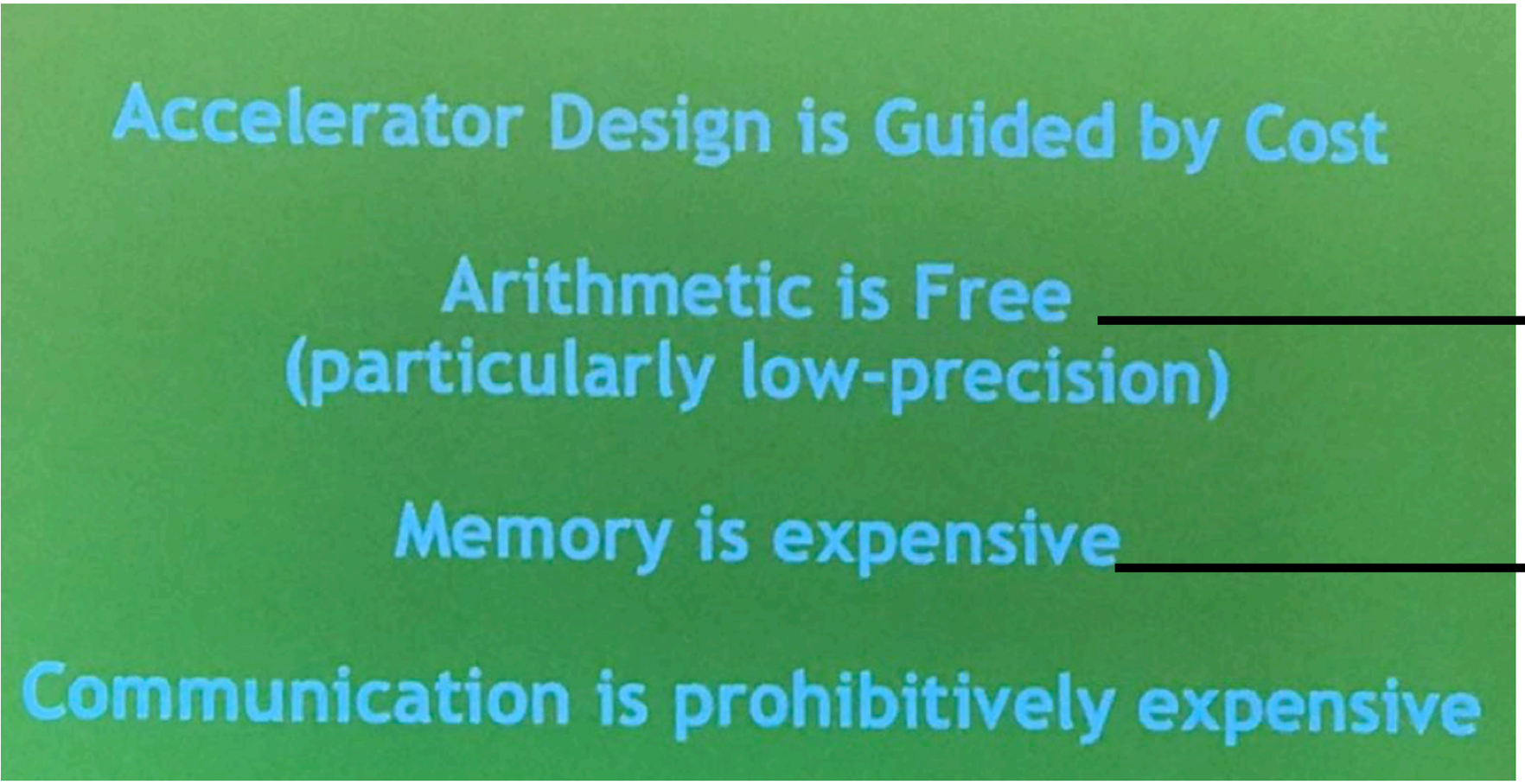
NPU,Diannao,etc

Bill Dally@AACBB(Sat.)

CEI
cei.pratt.duke.edu

ALCHEM
alchem.usc.edu

Discussion: What's next for deep learning accelerators?

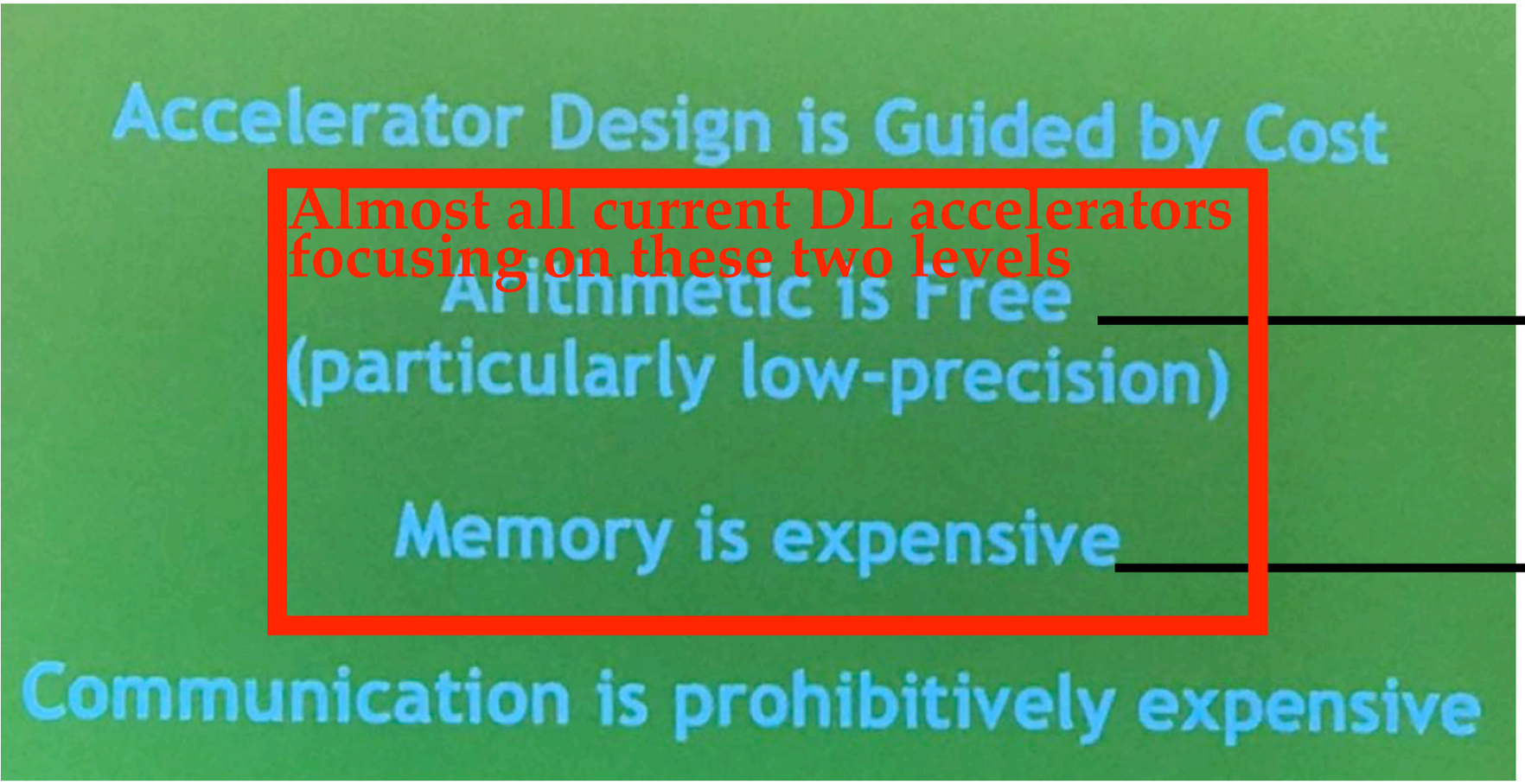


NPU,Diannao,etc

DaDianao, Eyeriss, PipeLayer, etc

Bill Dally@AACBB(Sat.)

Discussion: What's next for deep learning accelerators?

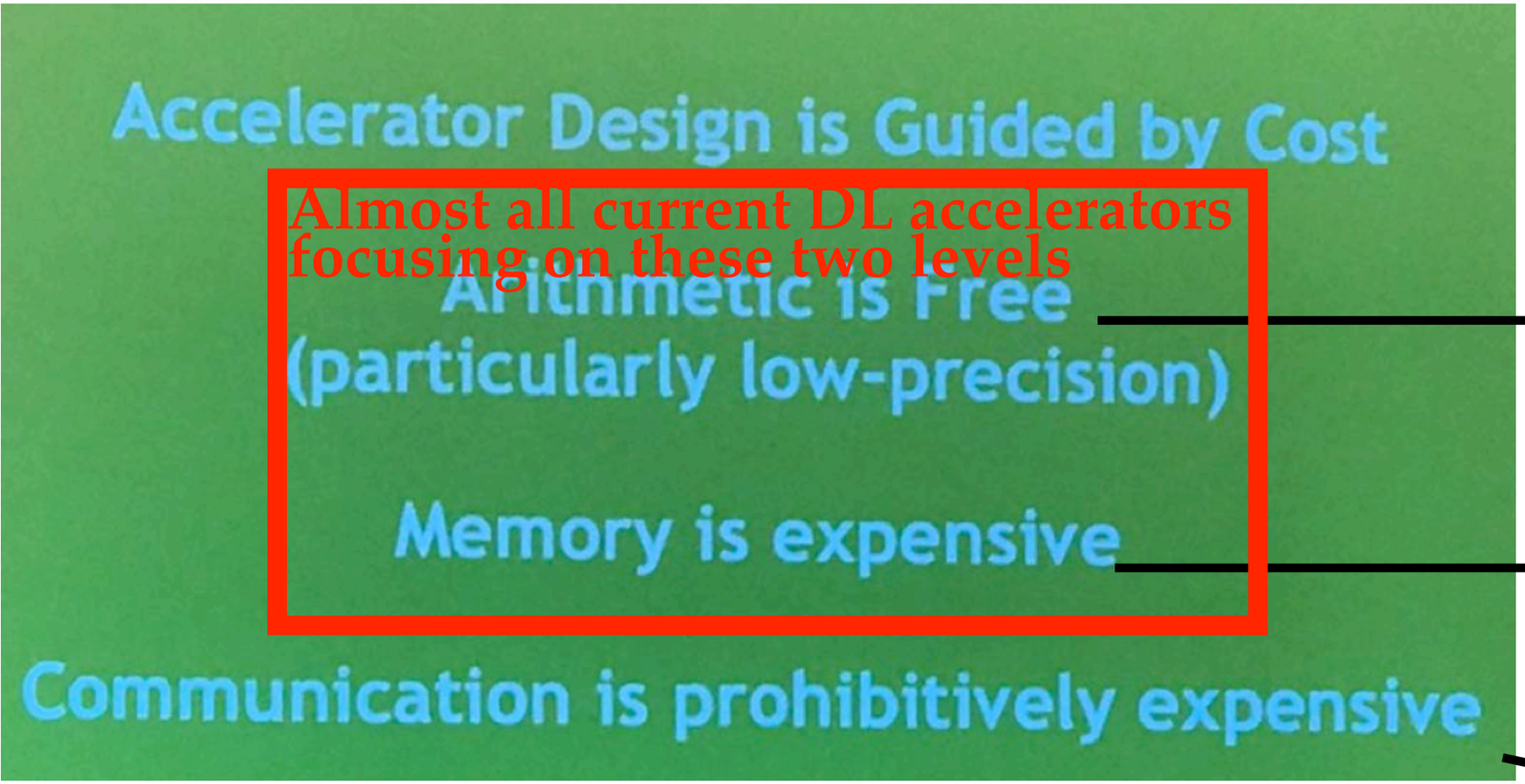


NPU,Diannao,etc

DaDianao, Eyeriss, PipeLayer, etc

Bill Dally@AACBB(Sat.)

Discussion: What's next for deep learning accelerators?



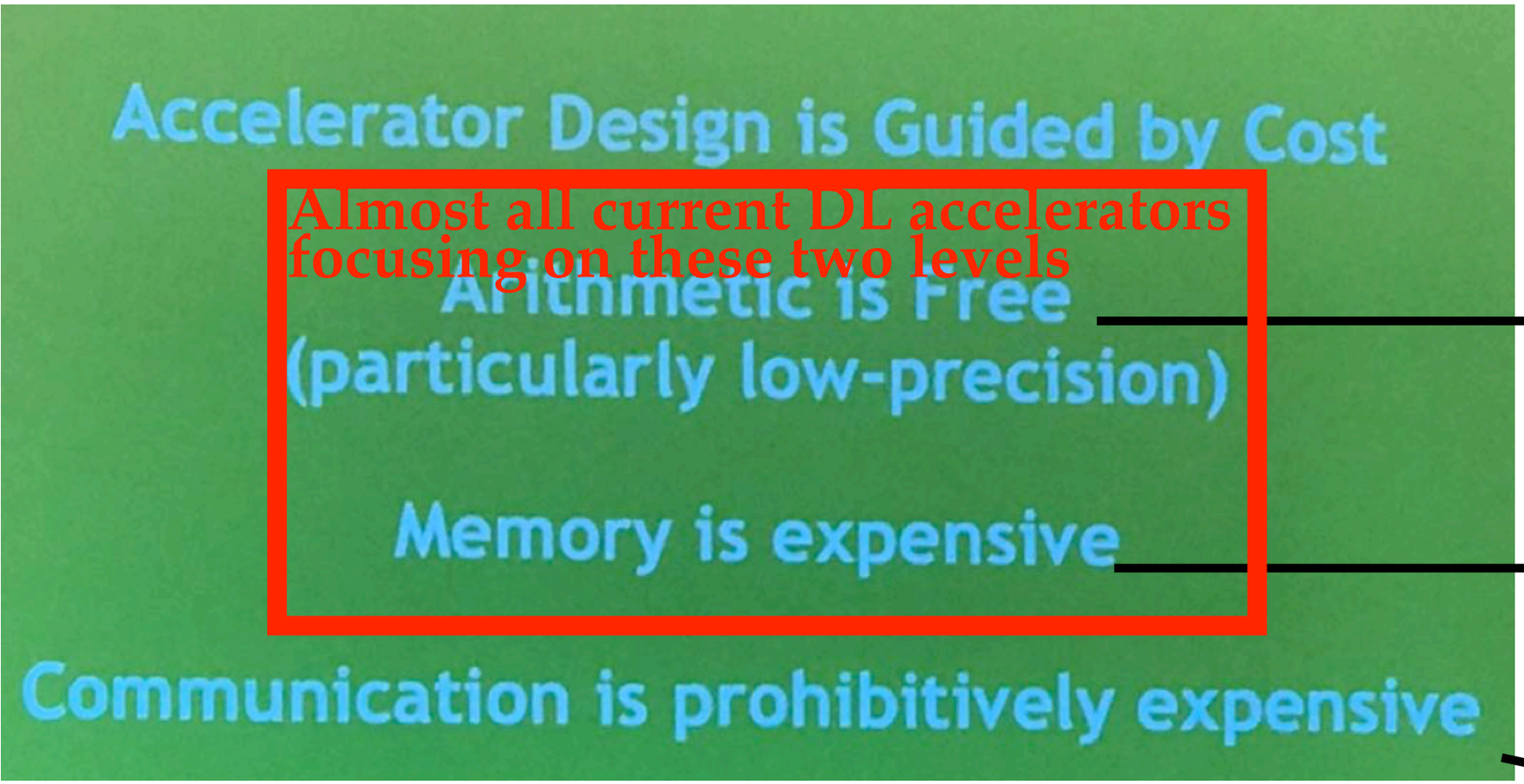
NPU,Diannao,etc

DaDianao, Eyeriss, PipeLayer, etc

HyPar

Bill Dally@AACBB(Sat.)

Discussion: What's next for deep learning accelerators?



NPU,Diannao,etc

DaDianao, Eyeriss, PipeLayer, etc

HyPar

Bill Dally@AACBB(Sat.)

- Accelerator array(multi-accelerators)
- Optimization outside of chip, eg., communication

HyPar: Towards Hybrid Parallelism for Deep Learning Accelerator Array

Linghao Song*, Jiachen Mao*, Youwei Zhuo#,
Xuehai Qian#, Hai Li*, Yiran Chen*

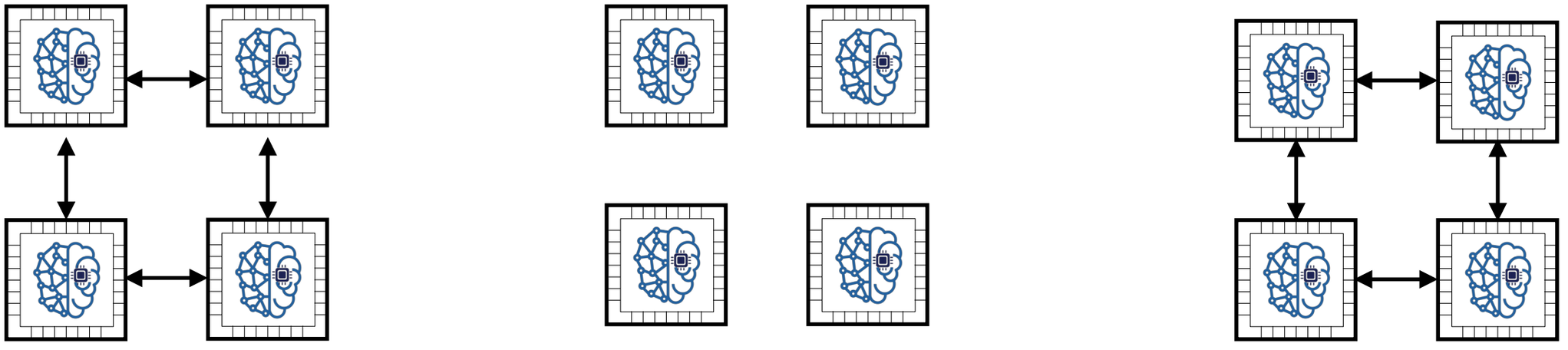
*Duke University

#University of Southern California

CEI
cei.pratt.duke.edu

ALCHEM
alchem.usc.edu

Backup1:Layer wise execution model



Communicate	Compute	Communicate
Layer (i-1)->i	Layer i	Layer i->(i+1)

Communication dominates!

CEI
cei.pratt.duke.edu

ALCHEM
alchem.usc.edu

Backup2: three tensor computation phases in training

Data Forward:

$$\mathbf{F}_{l+1} = f\left(\mathbf{F}_l \otimes \mathbf{W}_l\right)$$

Error Backward:

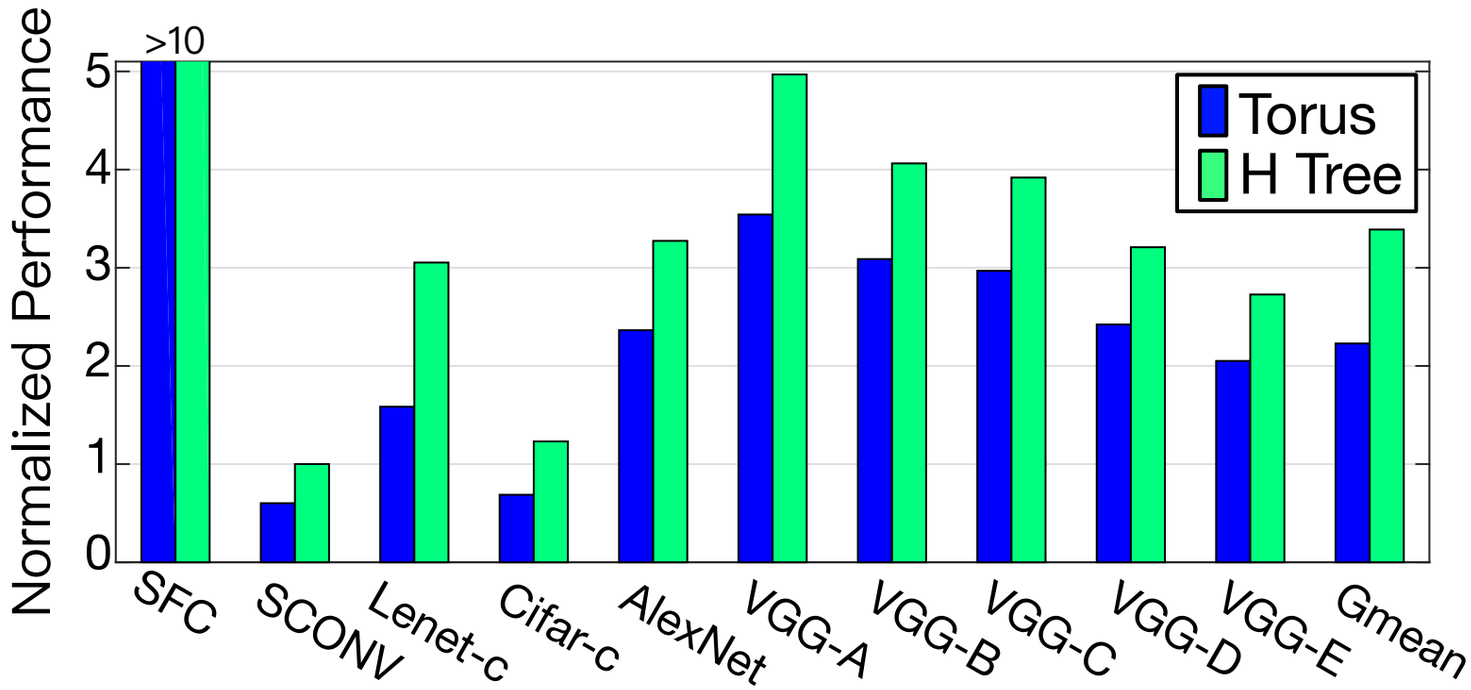
$$\mathbf{E}_l = \left(\mathbf{E}_{l+1} \otimes \mathbf{W}_l^*\right) \odot f'\left(\mathbf{F}_l\right)$$

Gradient Computation:

$$\triangle \mathbf{W}_l = \mathbf{F}_l^* \otimes \mathbf{E}_{l+1}$$

Backup3: topologies

HyPar compared to torus:



Hierarchical partition favors H-Tree, and topology-aware partition is required.