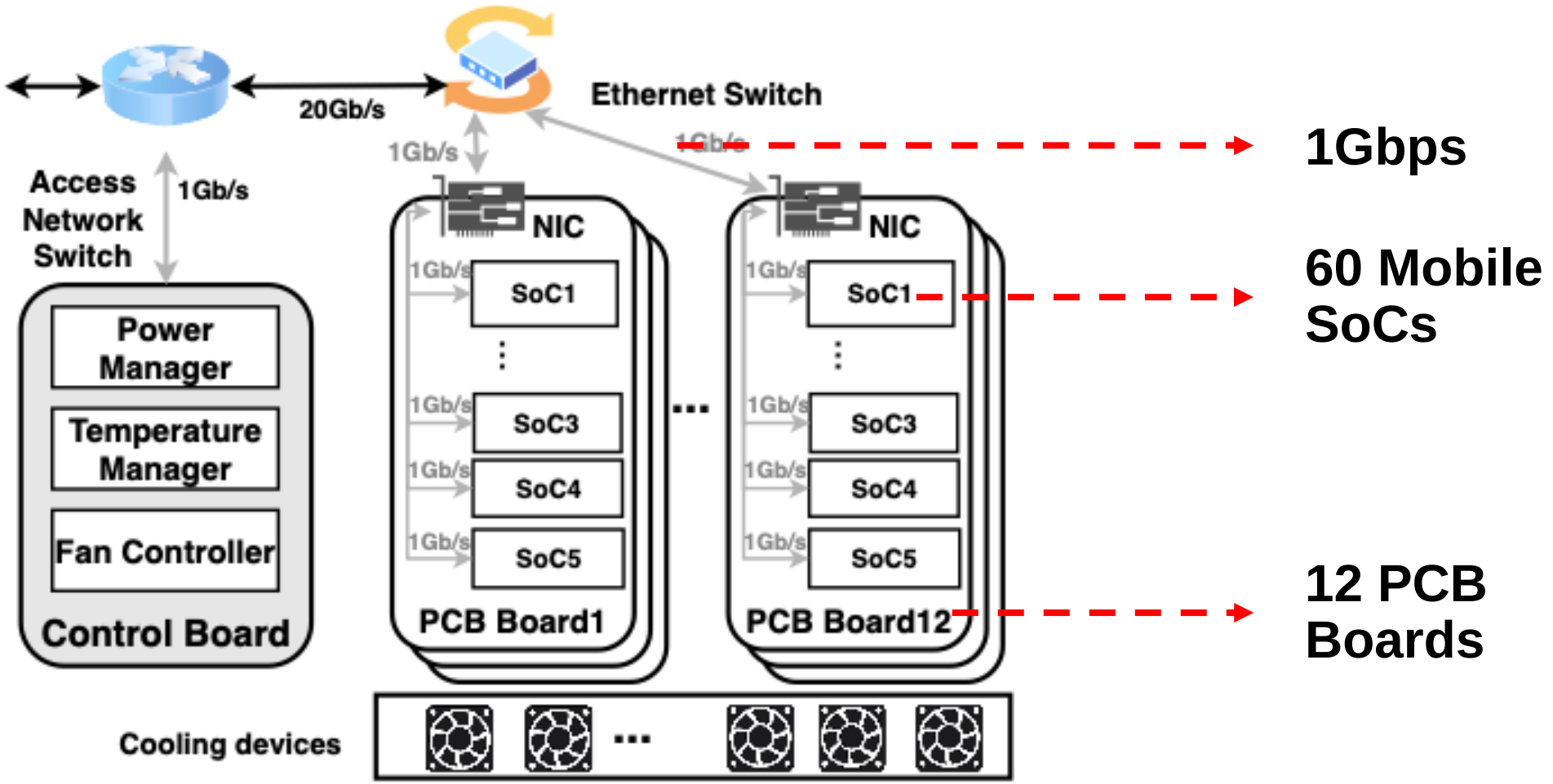
SoCFlow: Efficient and Scalable DNN Training on SoC-Clustered Edge Servers

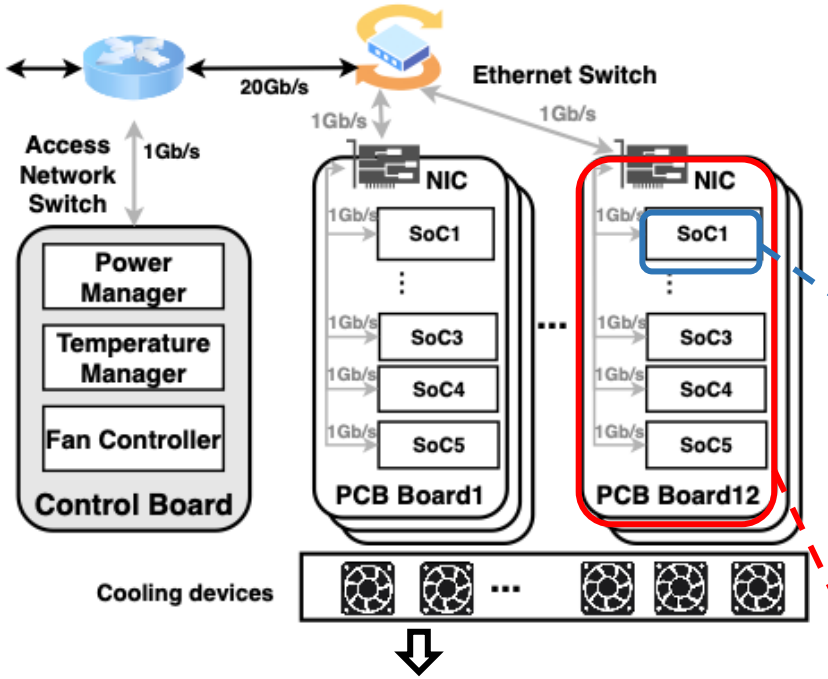
Daliang Xu*, Mengwei Xu*, Chiheng Lou, Li Zhang, Gang Huang, Xin Jin, Xuanzhe Liu

(*co-primary)

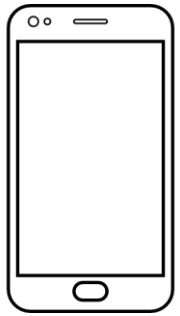
SoC-Cluster


20Gb/s
Ethernet Switch
1Gb/s
Access Network Switch
1Gb/s
Power Manager
Temperature Manager
Fan Controller
Control Board
NIC
SoC1
SoC3
SoC4
SoC5
PCB Board1
NIC
SoC1
SoC3
SoC4
SoC5
PCB Board12
Cooling devices
1Gbps
60 Mobile SoCs
12 PCB Boards

SoC-Cluster



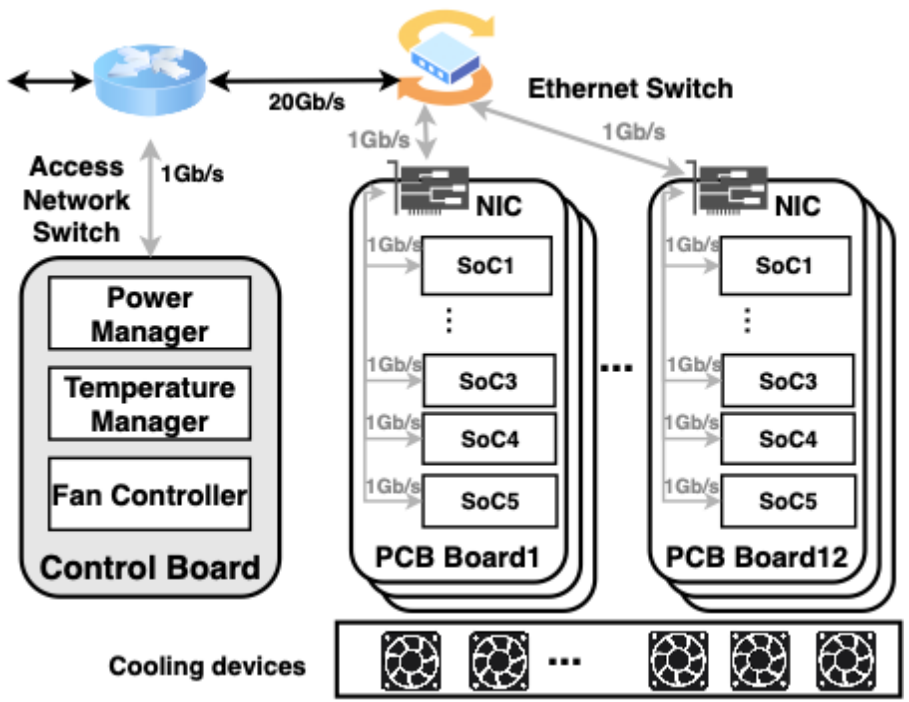
60 Mobile SoCs



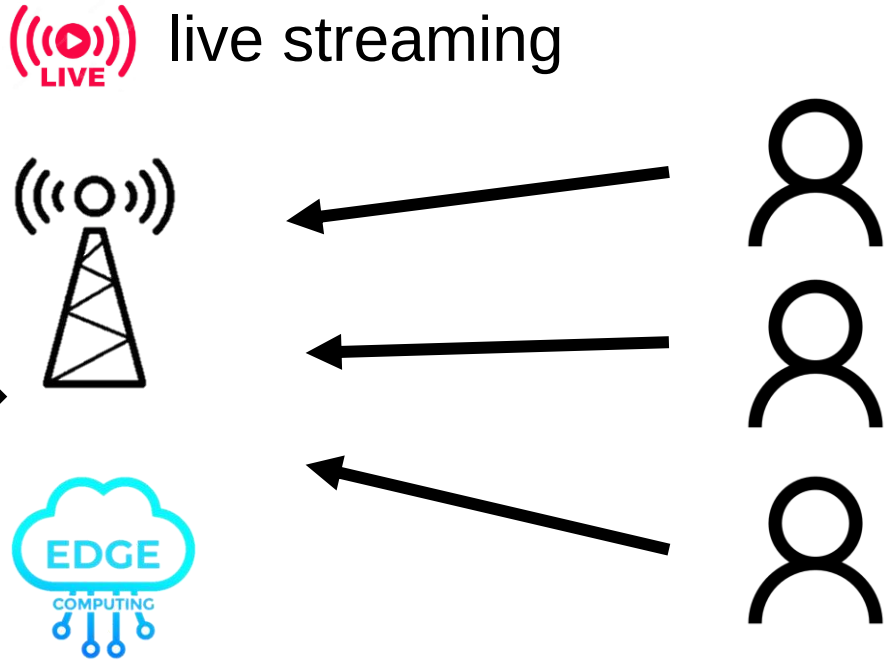
CPU	1 x 2.84 GHz Kryo 585 + 3 x 2.42 GHz Kryo 585 + 4 x 1.8 GHz Kryo 585
GPU	Qualcomm Adreno 650
NPU	Hexagon 698 DSP
Memory	12GB LPDDR5
Disk	UFS 3.1 256GB
OS	Android 10 (Kernel 4.19.81)
Network Interface	1 x 1GE RJ45 Port (1Gbps) + 2 x 10GE SPF+ Port (2 x 10Gbps)
Physical Interface	2 Rack Unit

12 PCBs

SoC-Cluster



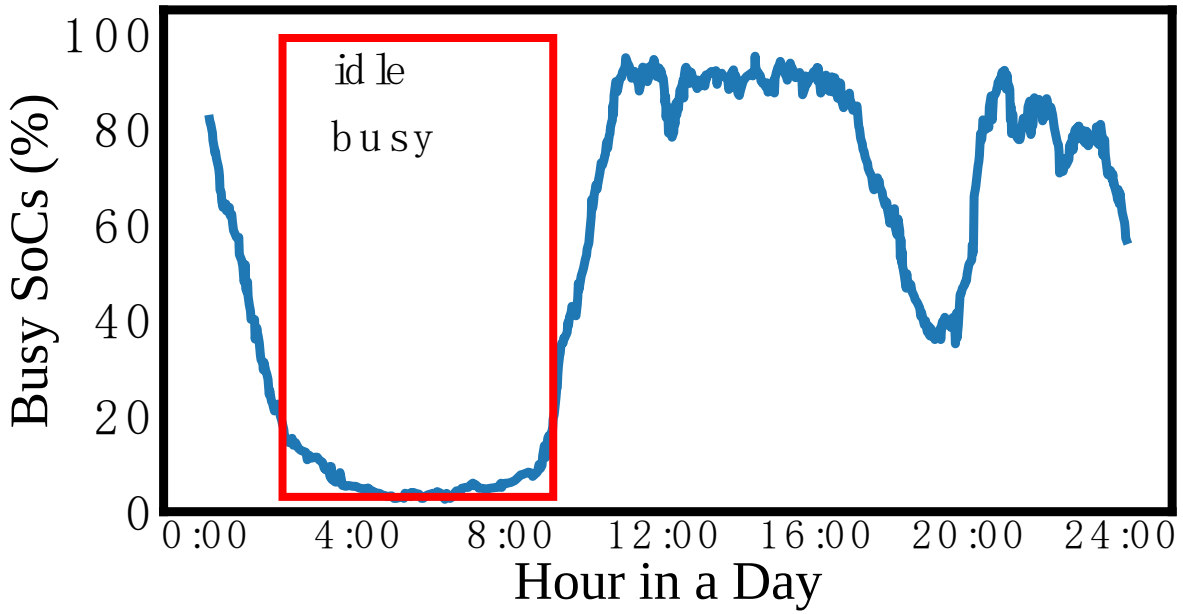
Deploy



live streaming

cloud gaming

SoC-Cluster is under-utilized



The average CPU usage of more than 95% of SoCs is under 20%

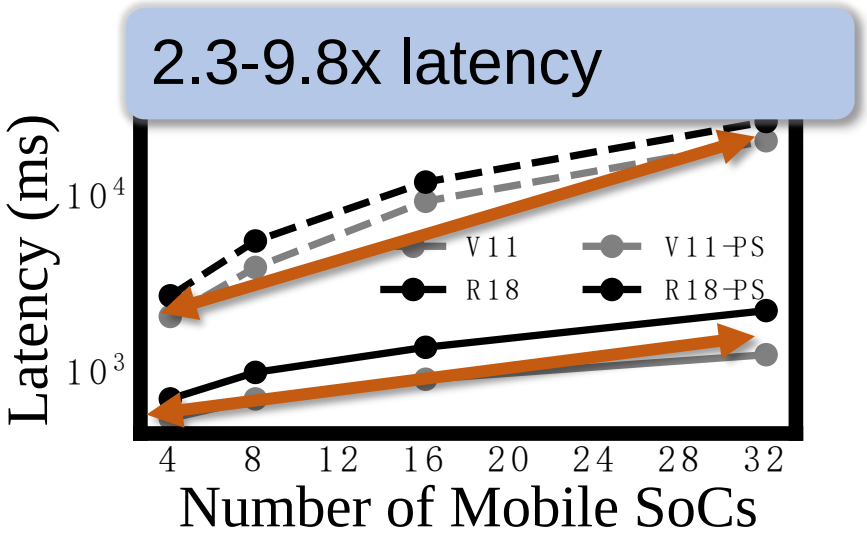
Harvest

Distributed deep learning training?

Challenges

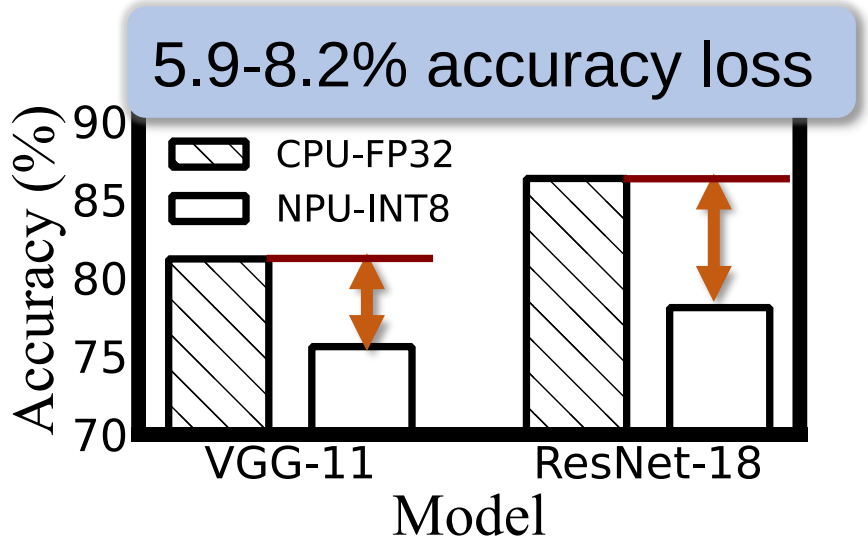
Scarce network bandwidth.

< 1Gbps

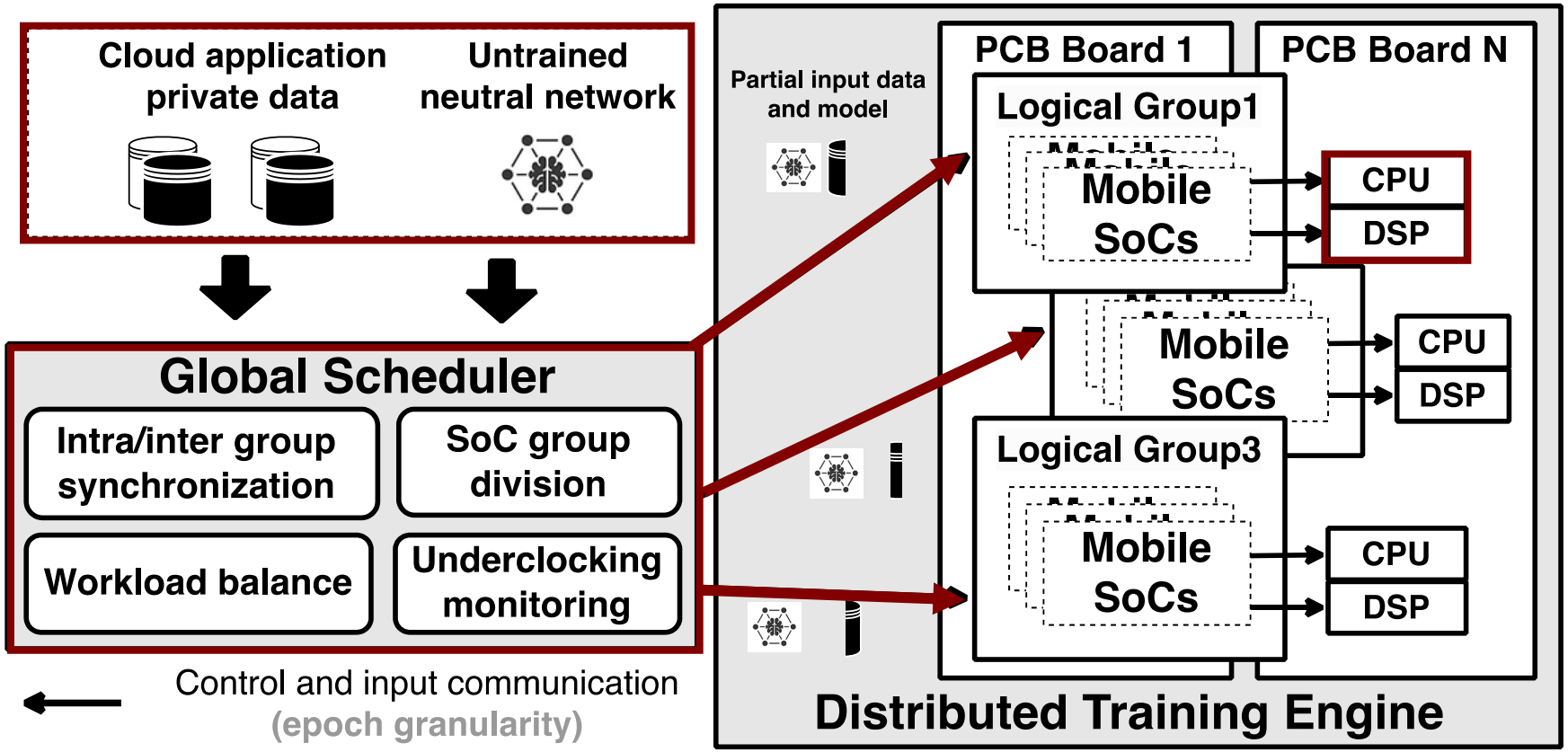


Heterogeneous processors with mixed data formats.

INT8 for NPU
FP32 for CPU

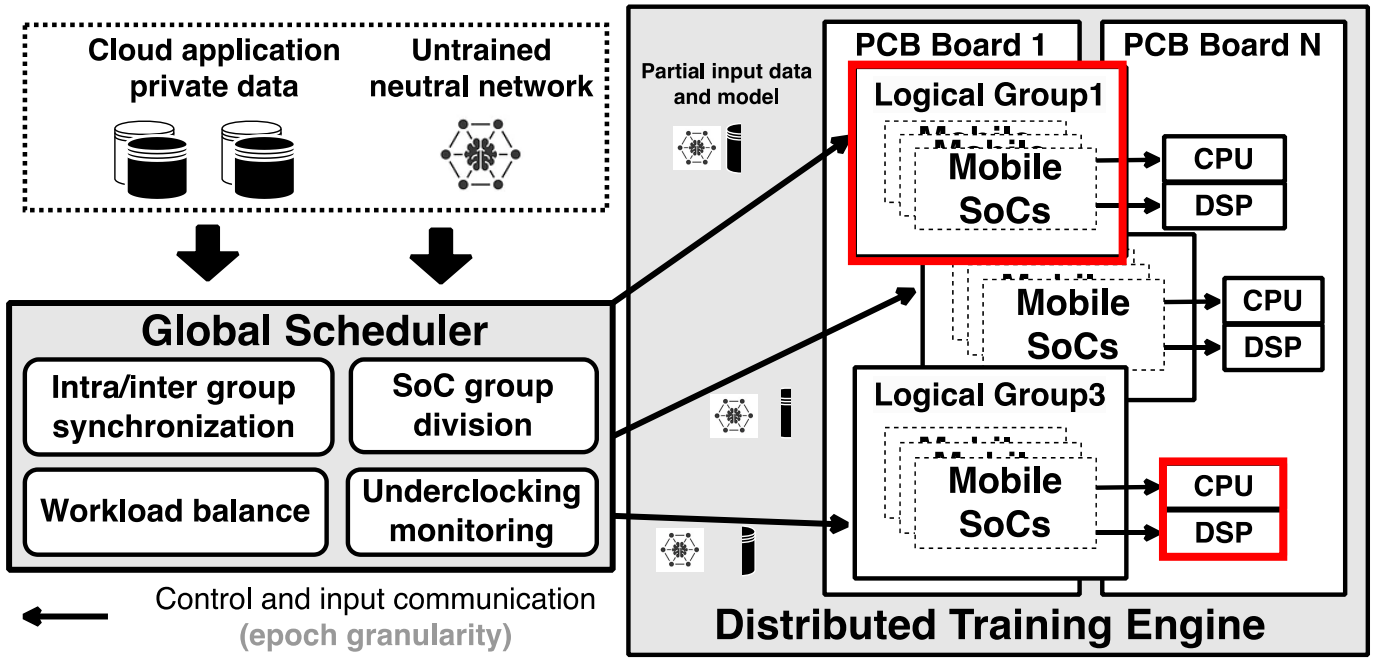


SoCFlow workflow



1. Input the training datasets and the DNN to be trained
2. Determines how SoCs will be orchestrated, such as SoC grouping
3. Dispatch the training data and model to each SoC
4. Perform FP32-based training on CPU, Int8-based training on mobile NPU

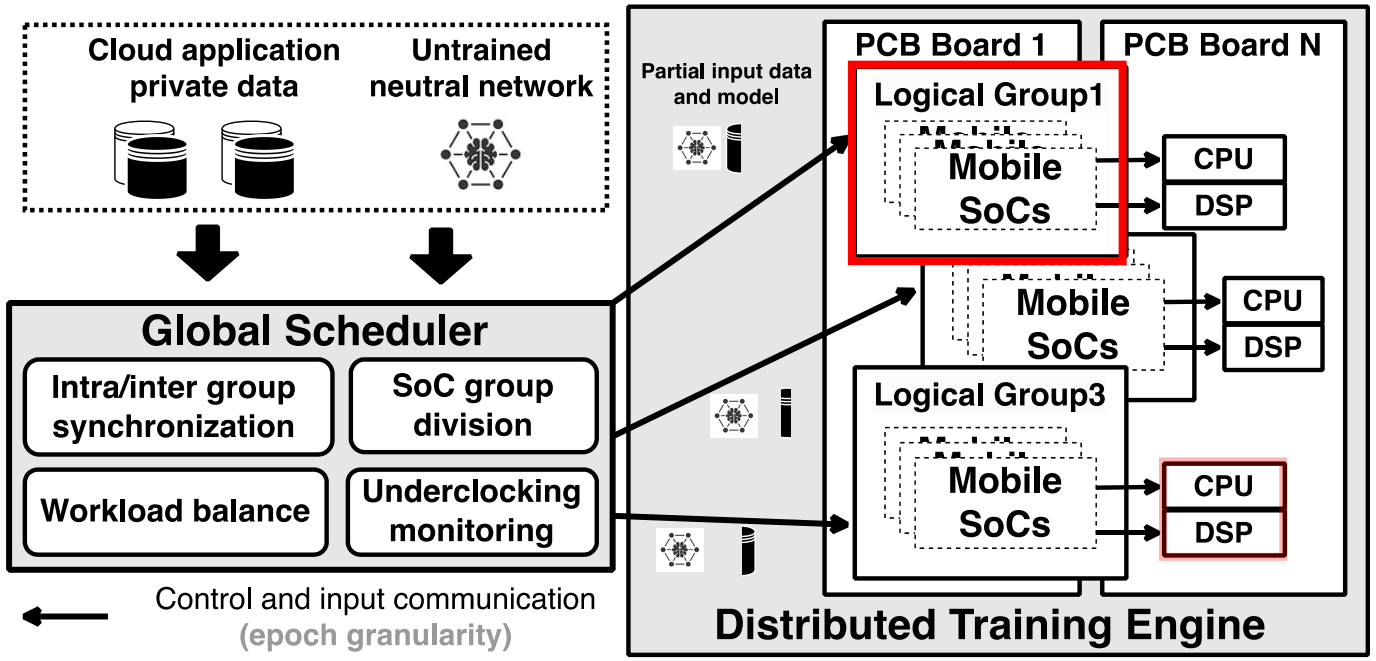
SoCFlow



#1: Group-wise parallelism with delayed aggregation.

#2: Data-parallel Mixed-precision Training.

SoCFlow



#1: Group-wise parallelism with delayed aggregation.

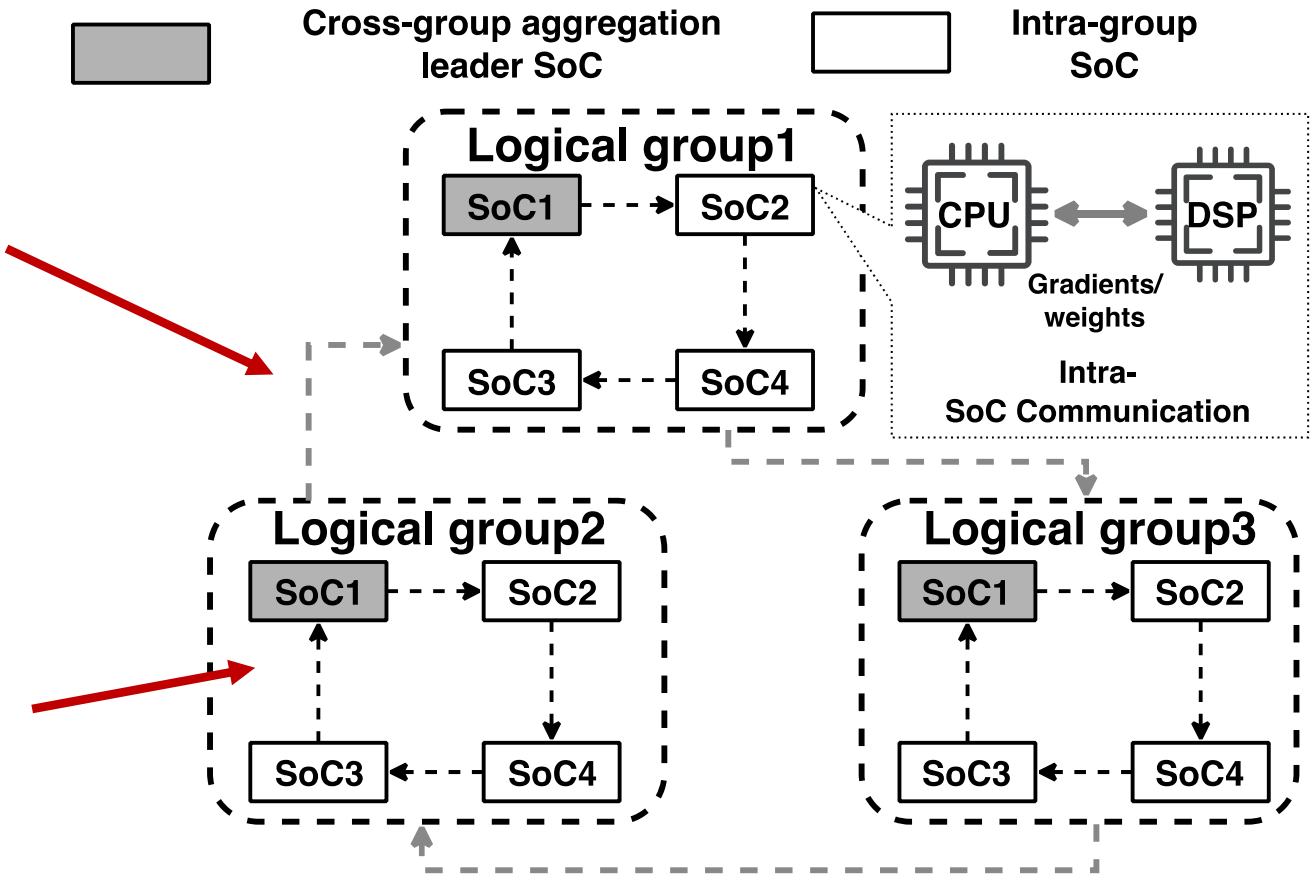
#2: Data-parallel Mixed-precision Training.

Group-wise parallelism with delayed aggregation

Key idea: The network capacity of the SoC-Cluster lies somewhere between the high-speed data center and the wireless network

Epoch-grain: Inter-group weights aggregation

Batch-grain: Intra-group gradients aggregation

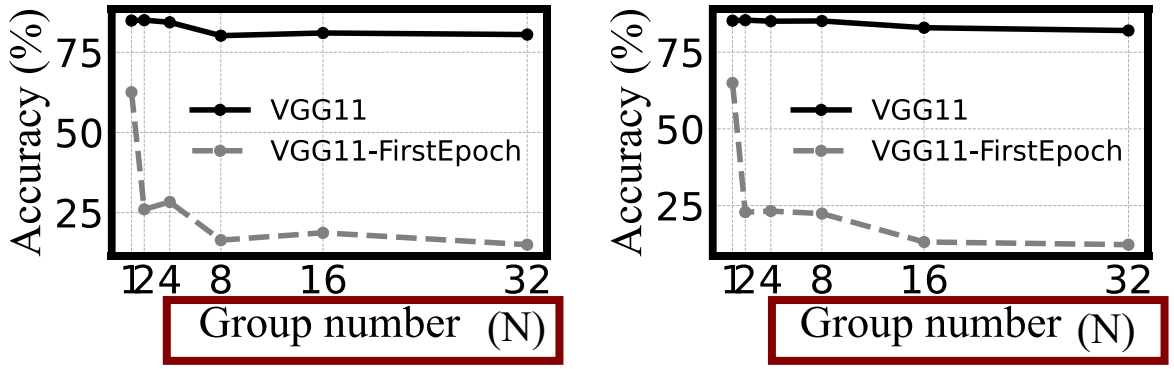


Determine group size (N)

Per-epoch training time is negatively correlated to group number (*N*).

$$T_{epoch} = \frac{NUM_{sample}}{(N * BS_g)} * (T_{train}^{BS_g} * \frac{N}{M} + T_{sync})$$

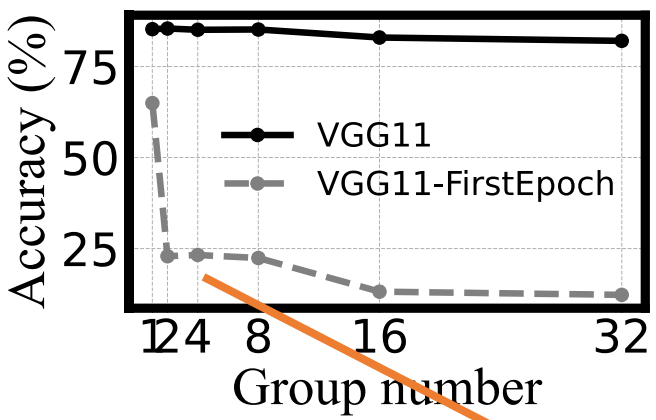
Convergence accuracy exhibits a negative correlation with group number (*N*).

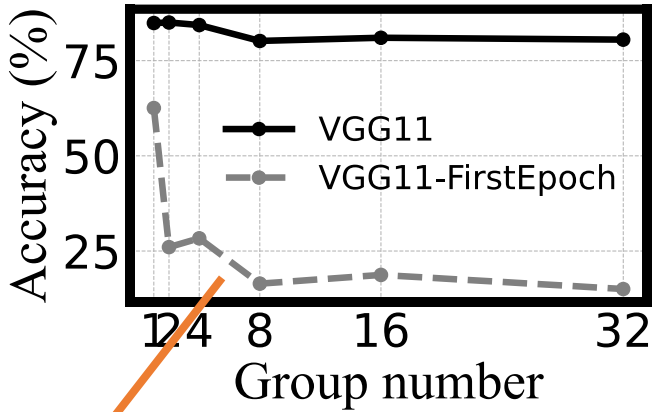


- **Larger N => A higher convergence accuracy**
- **Smaller N => A lower training time**

Determine group size

Observation: The training accuracy observed during the initial epoch closely mirrors the behavior of convergence accuracy.





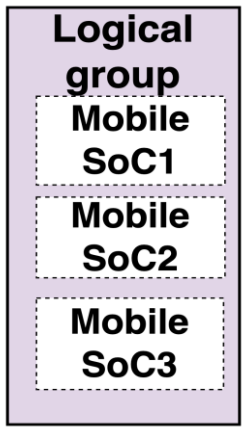
logical group size

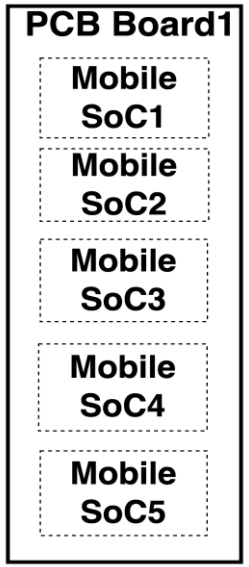
Map logical to physical topologies

Group size != physical PCB SoC size

Logical group size = 3

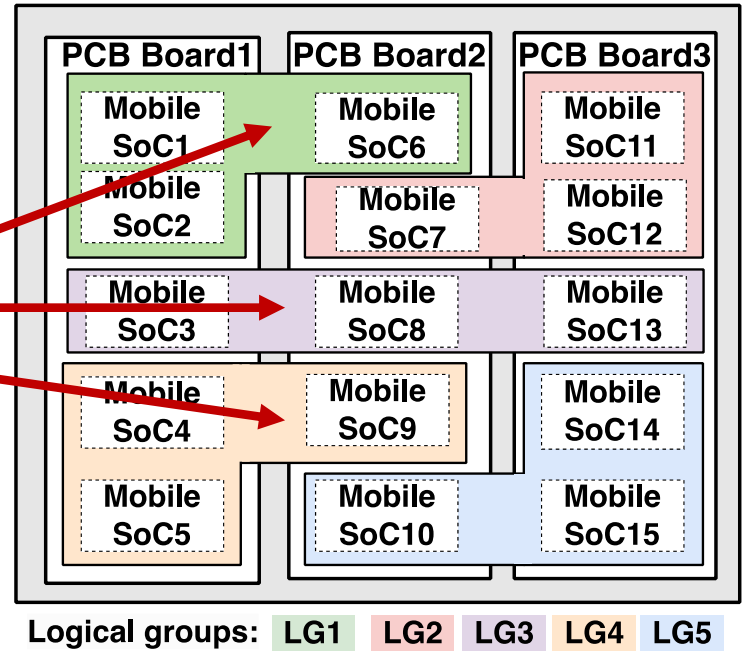
Physical PCB size = 5





problem

Inter-PCB communication contention



Accuracy-speed trade-off

Hardware design

Map logical to physical topologies

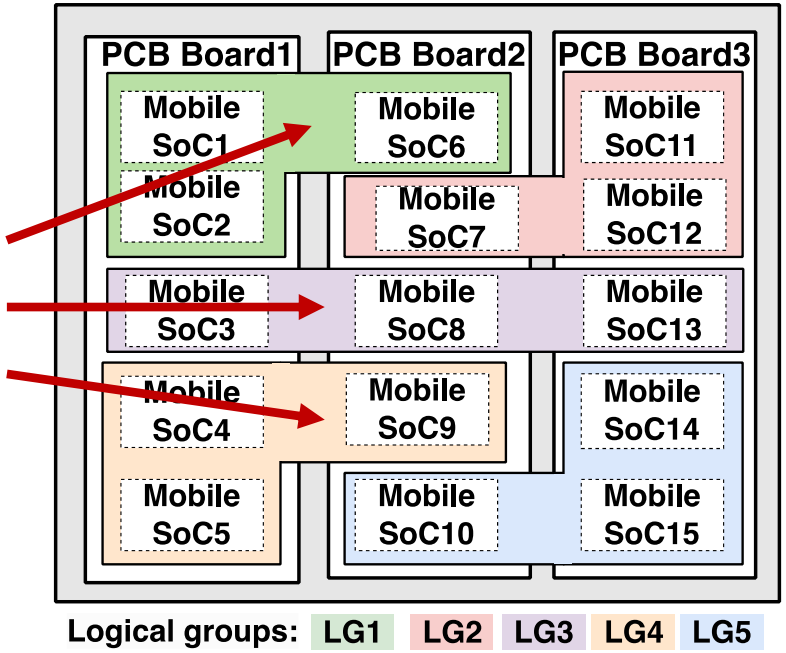
Group size != physical PCB SoC size

- The number of logical groups contending for inter-PCB communication.

$$L_i^{\text{inter}}$$

depict

Inter-PCB communication contention



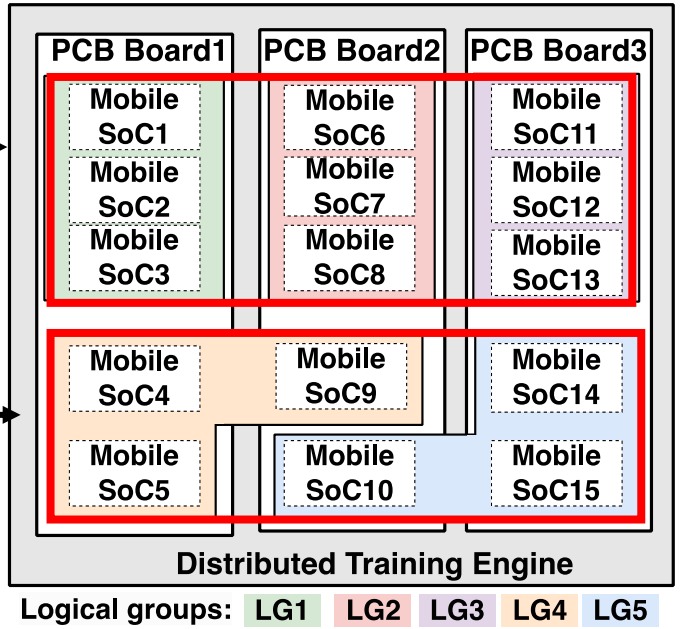
- Mapping goal: find the minimum contention.

$$\min L_i^{\text{inter}}, \forall\ i\ \in PCB$$

Integrity-greedy mapping

Key idea: All SoCs of a logical group mapped to a single PCB board will not contend for inter-PCB network communication.

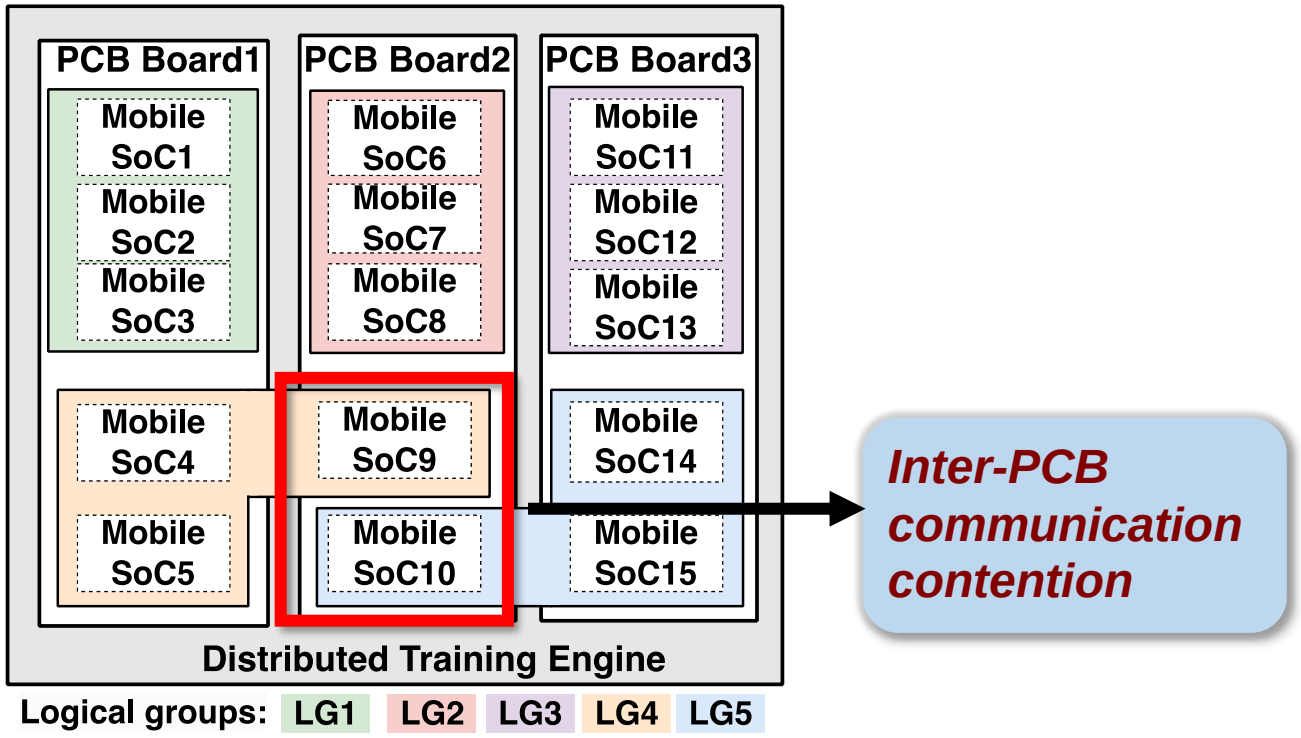
- *Step 1*: map as many logical groups as possible to physical groups without splitting.
- *Step 2*: the rest of the logical nodes are mapped in sequence.



- *Theorem 1*: Integrity-greedy mapping minimizes C.
- *Theorem 2*: Integrity-greedy mapping guarantees that each logical group contends with up to two other logical groups for NIC.

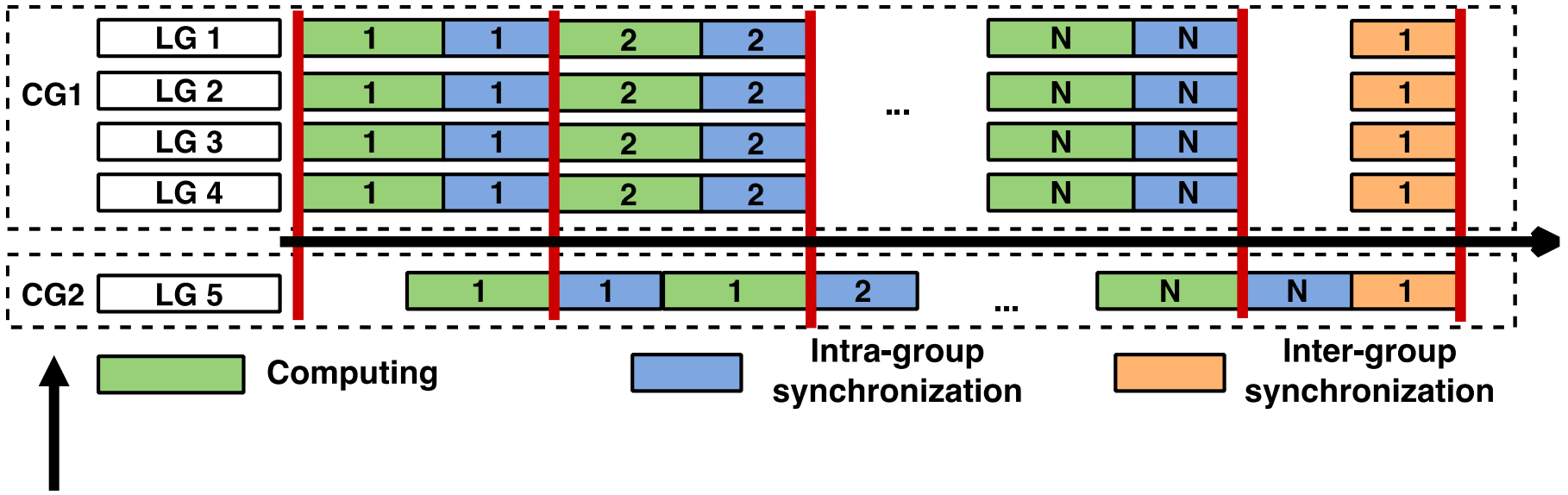
Group-wise communication planning

The maximum value of inter-PCB communication contention is two



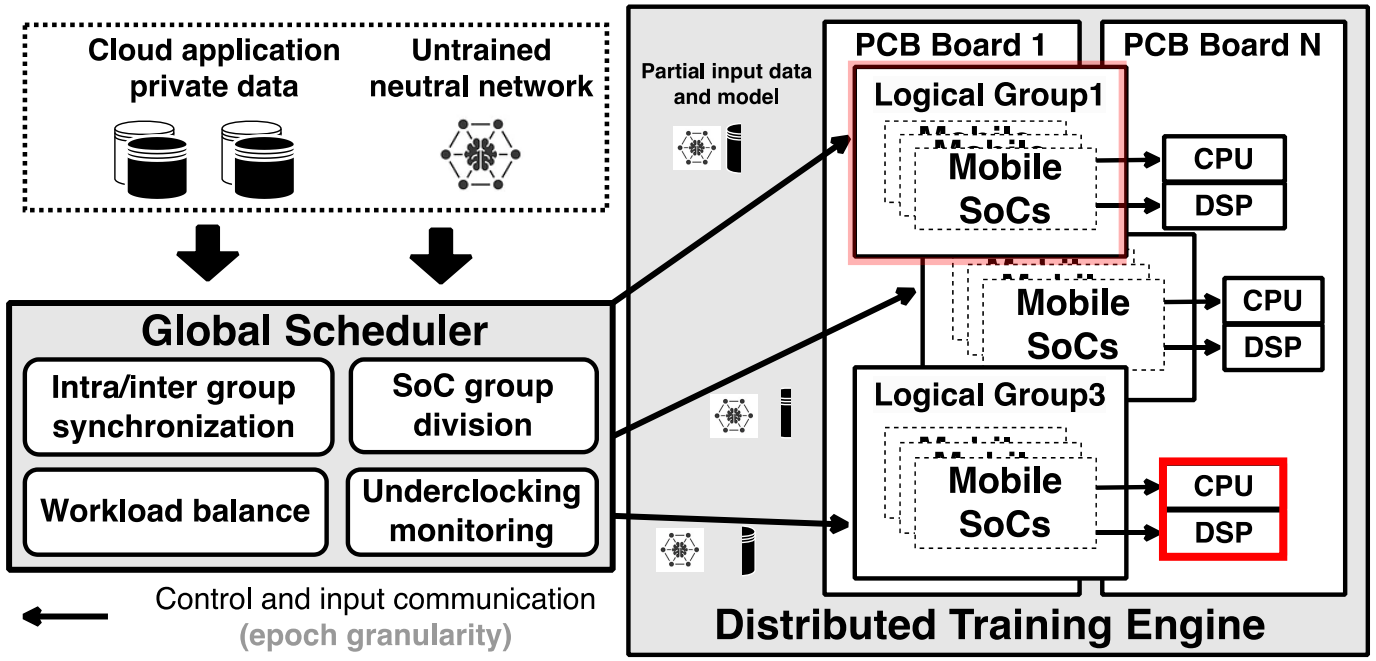
Group-wise communication planning

Key idea: We can divide groups with contention into different communication groups (CGs) to communicate separately in sequence to avoid network contention.



Theorem 2 guarantees 2 communication groups at most.

SoCFlow



#1: Group-wise parallelism with delayed aggregation.

#2: Data-parallel Mixed-precision Training.

Data-parallel Mixed-precision Training

Key idea: Compensate the precision loss of INT8-based training by offloading part of the training to the CPU with FP32 format

- Two metrics
  - α – confidence that indicates the error gap between the INT8 model and the FP32 model.

$$\alpha = Cos(< logits_{FP32}, logits_{INT8} >)$$

- α-> 1 => The INT8 Model is more accurate.
- α-> 1 => More training data should be fed into the INT8 model.

Data-parallel Mixed-precision Training

Key idea: Compensate the precision loss of INT8-based training by offloading part of the training to the CPU with FP32 format

- Two metrics
  - β – compute power ratio that represents the ratio of compute power for heterogeneous processors

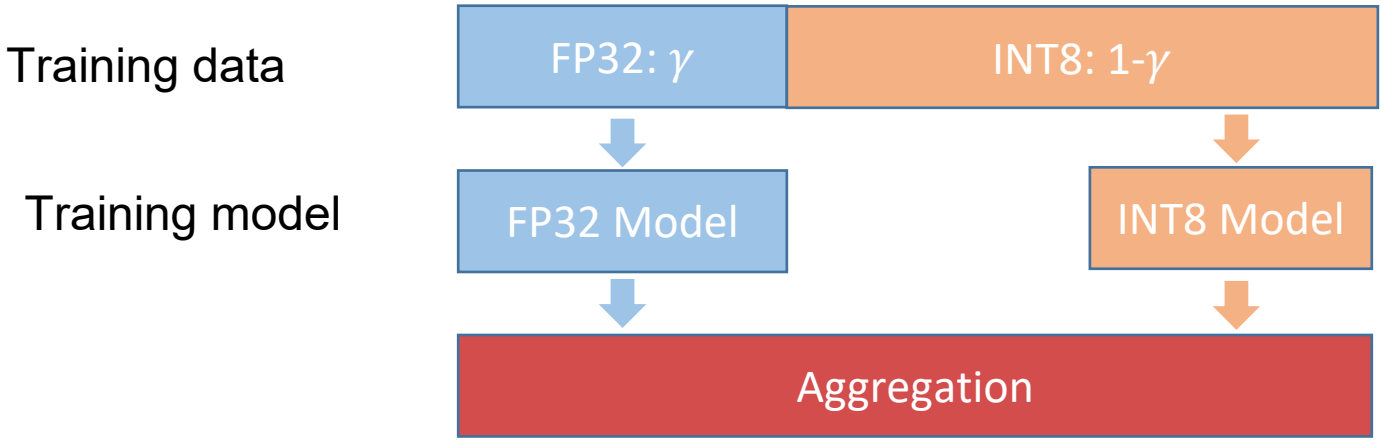
$$\beta = \frac{T_{NPU}}{T_{NPU} + T_{CPU}}$$

- β-> 1 => NPU is more powerful.
- β-> 1 => More training data should be fed into the INT8 model.

Data-parallel Mixed-precision Training

Key idea: Compensate the precision loss of INT8-based training by offloading part of the training to the CPU with FP32 format

- Jointly consider two metrics
  - $\gamma = max\{e^{-\alpha}, 1 - \beta\}$
  - Aggregation rule: $w_{i+1} = \gamma * w_{i+1}^{FP32} + (1-\gamma) * w_{i+1}^{INT8}$



Evaluation: settings

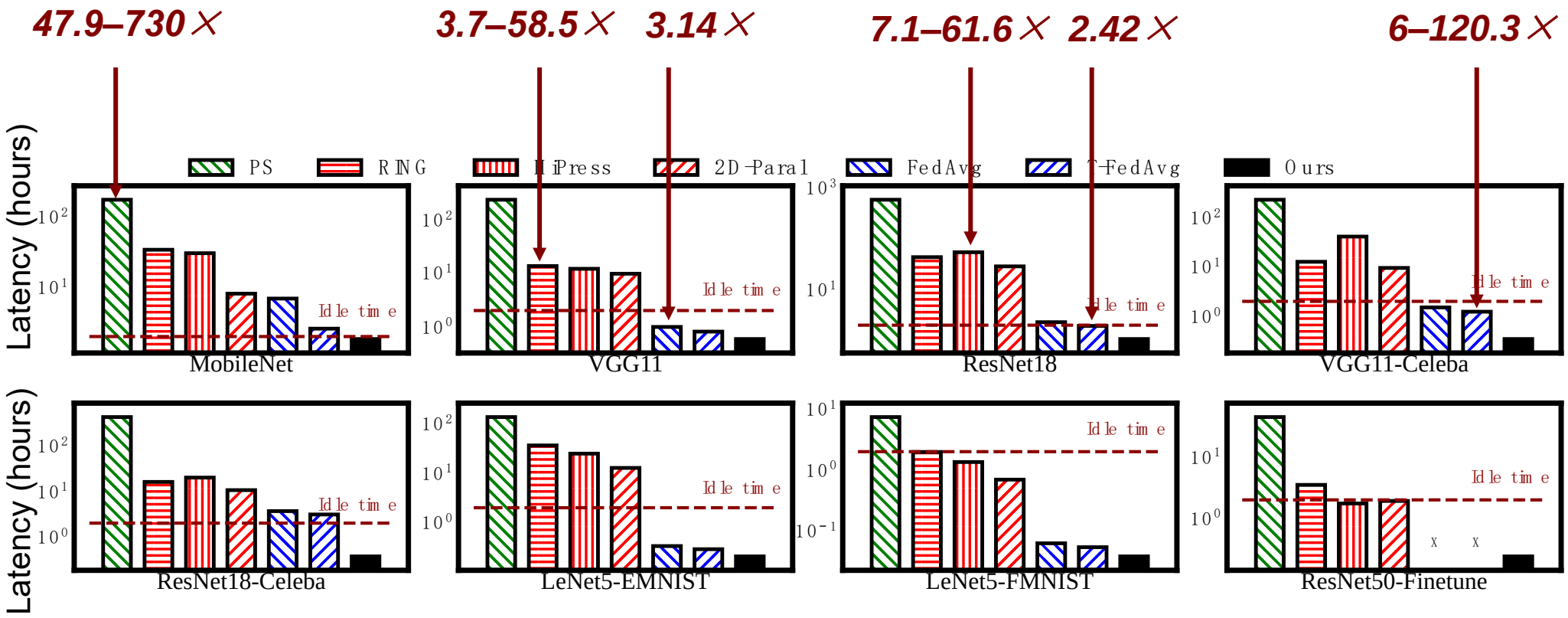
- Models and datasets

Model	Dataset	Learning methods
LeNet	FMNIST and Fashion-MNIST	From scratch
VGG-11	CIFAR-10 and CelebA	
ResNet-18	CIFAR-10 and CelebA	
MobileNet-V1	CIFAR-10	
ResNet-50	CINIC-10	Transfer learning

- SoC-Cluster
  - SnapDragon 865 SoC x 60
- Baselines
  - 4 Distributed machine learning baselines
    - 2 industry baselines : Parameter server and Ring-AllReduce
    - 2 SOTA research baselines: HiPress [SOSP 21] and 2D parallelism [ASPLOS 23]
  - 2 Federated learning baselines: FedAvg and Tree FedAvg

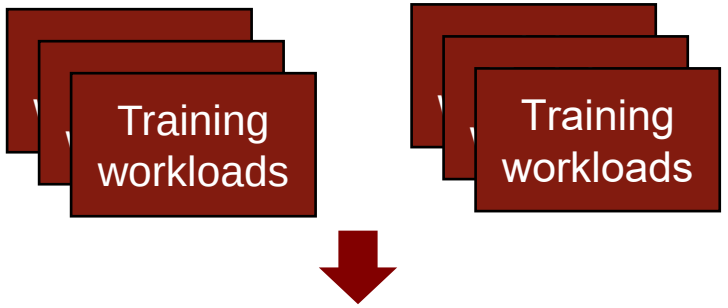
Evaluation: End-to-end Performance

Only SoCFlow can guarantee that all training tasks finish within two hours smaller than the SoC-Cluster idle time.



Evaluation: Comparison with Cloud GPUs

SoCFlow are more suitable for training small-to-medium size models (MobileNet) than Cloud GPUs.

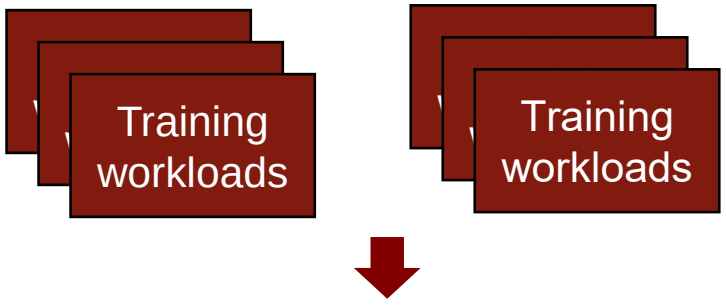


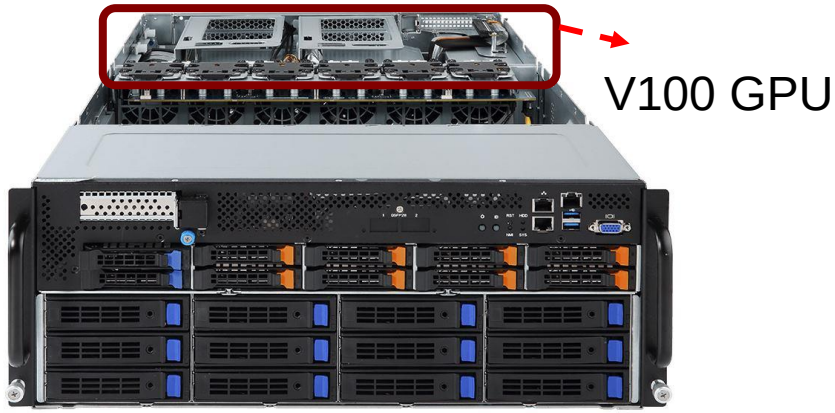
SoC-Cluster

Reduce **10.2×**
Energy consumption

Simpler cooling system

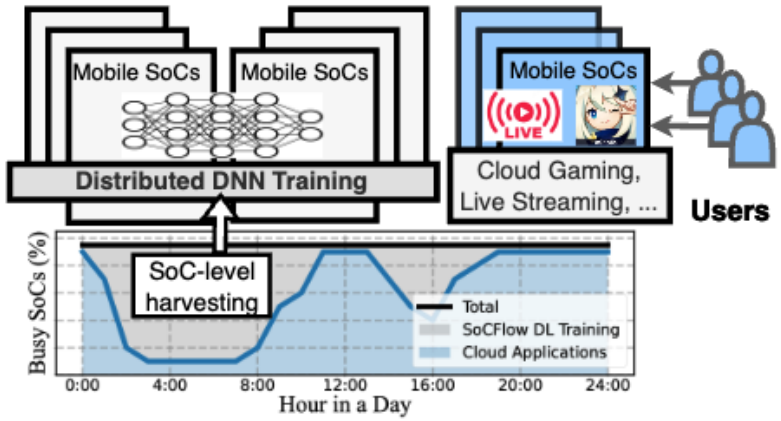
1/2 volume





V100 Server

Take-away



SoCFlow

- Propose first efficient DNN training engine for SoC-Clusters
- Incorporate group-wise parallelism and data-parallel mixed-precision training
- Build prototype and achieve superior performance over existing methods

xudaliang@pku.edu.cn

website https://daliangxu.github.io/

More about our affordable AI systems: http://www.liuxuanzhe.com