The exploration of complex Large-scale data based scenario automatic speech recognition

Complex scenario ASR in ZOOM

Haoyu (Charlie) Tang

April 24, 2022

Zoom AI/ML Engineering

Content

Introduction to automatic speech recognition

What is automatic speech recognition

Automatic Speech Recognition (ASR): generate texts from given audio wav, *argmax* ($P(\mathbf{Y}|\mathbf{X})$)

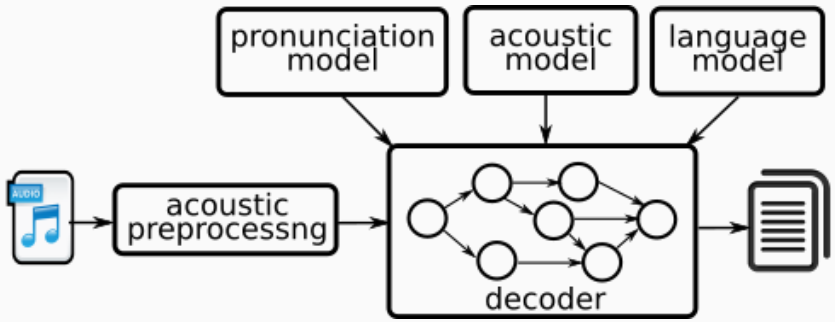


Figure 1: ASR[1]

Conventional Method: Acoustic model, Language model and Pronunciation dict/model.

End-To-End Method: Main Model, and language model (optional).

[1] *https://www.authot.com/en/2016/09/09/automatic-speech-recognition-system/*.

Brief History of ASR

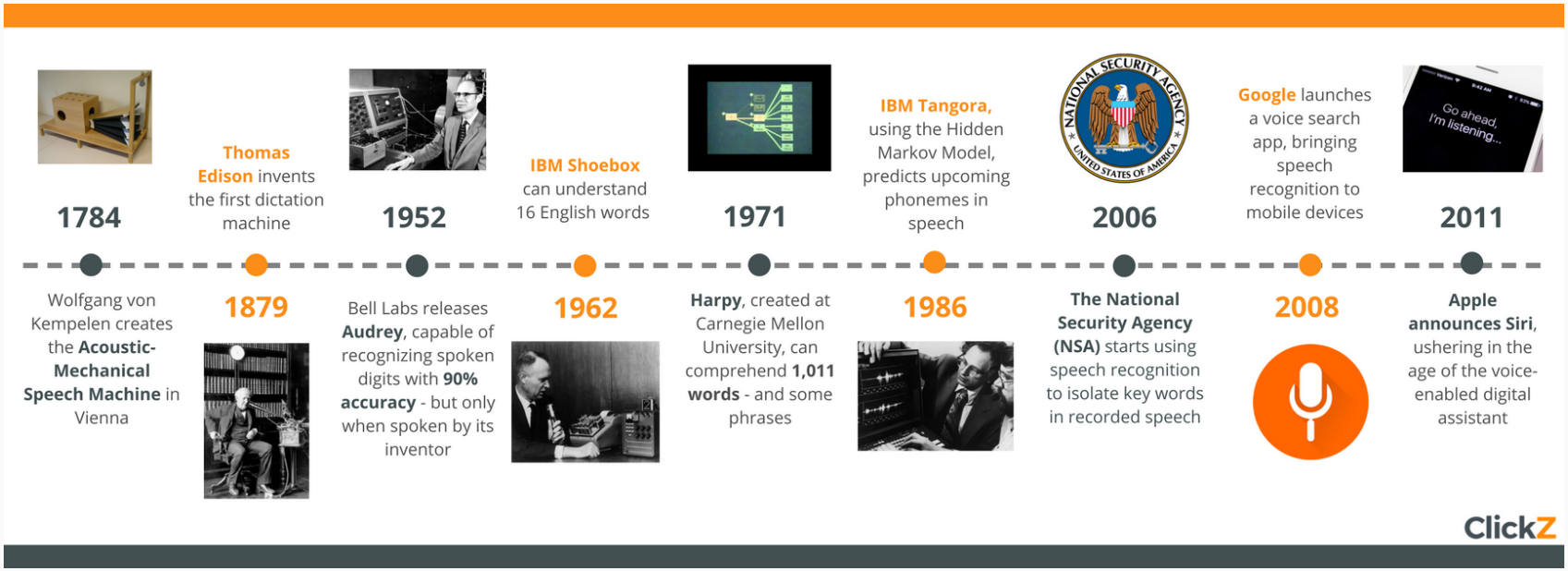


Figure 2: ASR history[2]

[2] *https://sonix.ai/history-of-speech-recognition*.

Brief History of ASR

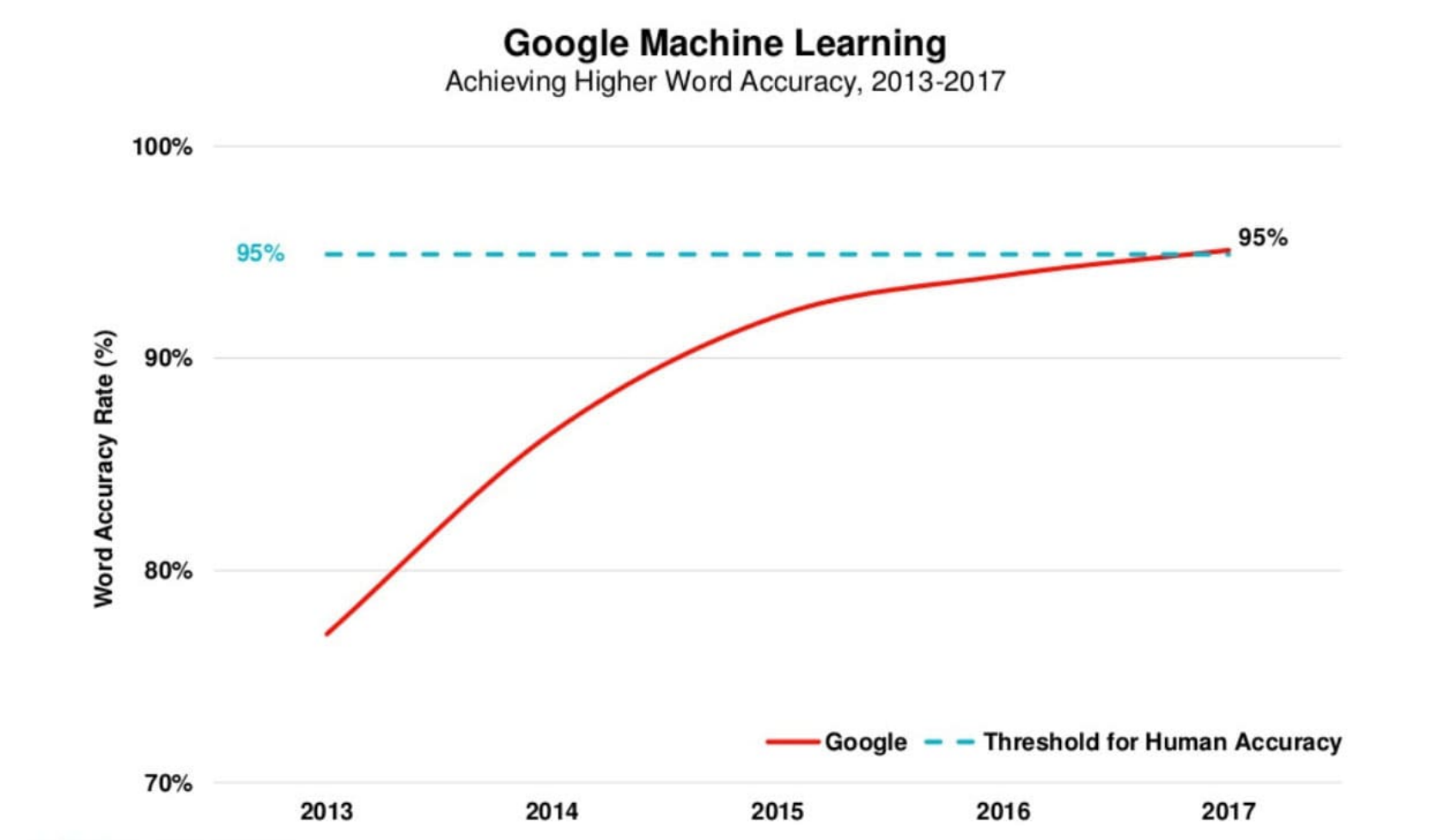


Figure 3: Recent decade ASR history[3]

[3] *https://sonix.ai/history-of-speech-recognition*.

ASR: current problems

Current problem in ASR for live, meeting and online chat scenario:

1. Spontaneous but most ASR open data is read sound
2. Open-set recognition + Large vocabulary
3. Noise especially for background music
4. Accent independent
5. Code-switch
6. Free switch between far-field and near-field since moving speak

End-to-End automatic speech recognition

A standard end-to-end(E2E) ASR architecture

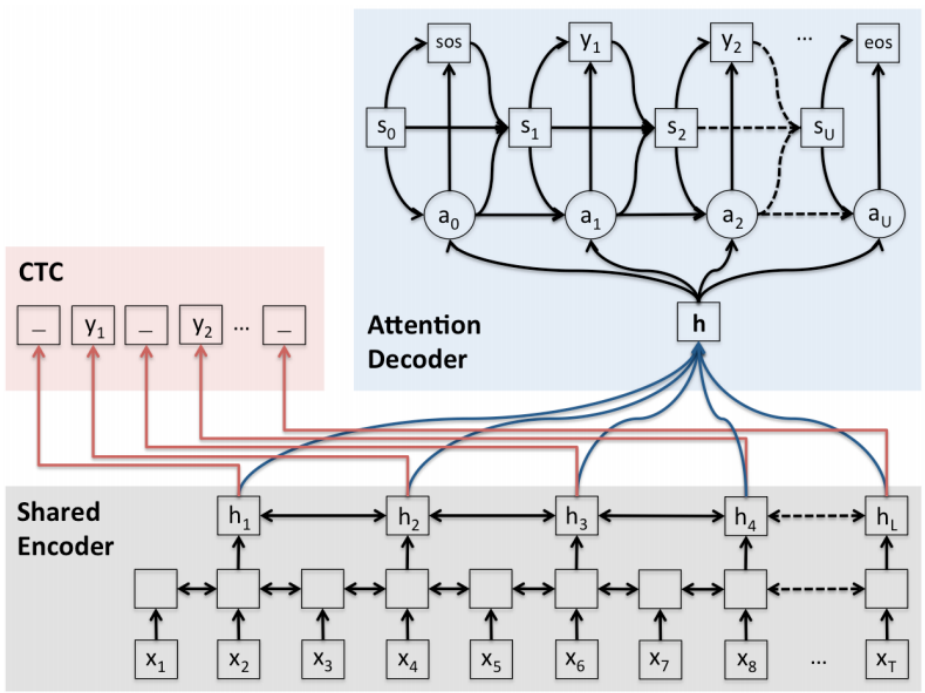


Figure 4: A standard end-to-end(E2E) ASR architecture[4]

This figure shows two standard E2E modeling method: CTC and encoder-attention-decoder. And These could be combined together as CTC-ATT architecture.

[4]Watanabe et al., "Hybrid CTC/attention architecture for end-to-end speech recognition".

ATT-CTC Training and Inference[5]

Loss combine :

$$\mathcal{L}_{\mathrm{MTL}} = \lambda \mathcal{L}_{\mathrm{CTC}} + (1-\lambda)\mathcal{L}_{\mathrm{Attention}} \tag{1}$$

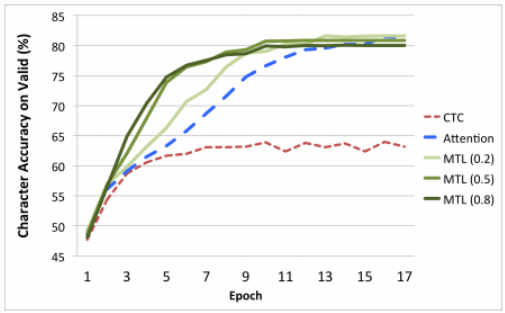


Figure 5: Loss combine

Joint decoding/rescoring:

$$C = argmax\{\lambda \log p_{ctc}(h|X) + (1-\lambda)p_{att}(h|X)\} \tag{2}$$

[5]Watanabe et al., "Hybrid CTC/attention architecture for end-to-end speech recognition".

Model innovation

Model improvement: from ATT to Transformer

Not only replace (B)LSTM(p)[6] in total encoder and docer as transformer[7].

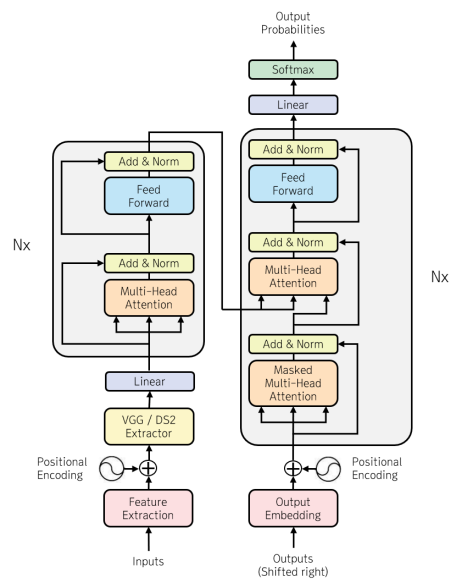


Figure 6: Transformer (Xfmr)

[6] Chan et al., "Listen, attend and spell".

[7] Dong, Xu, and Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition".

Reorder in ASR and Neural Machine Translation

this C-MHA also could be reviewed as a constrain. The next layer's each element could only be affected by neighbours of previous layer.

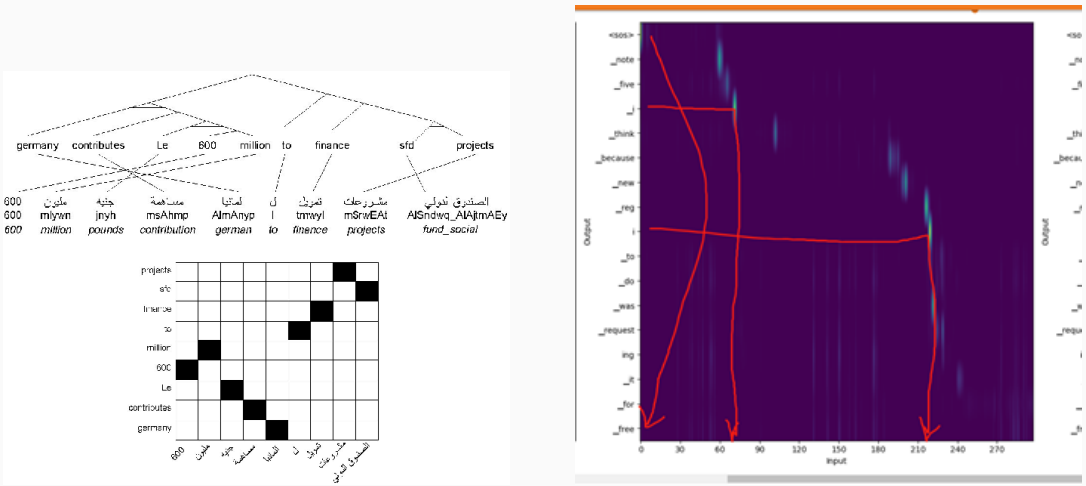


Figure 7: Reorder in NMT[8] and ASR

(a) : A reorder example in NMT

(b) : A heat map of source-target attention in S-Xfmr decoder

[8] Wu, Carpuat, and Shen, "Inversion transduction grammar coverage of arabic-english word alignment for tree-structured statistical machine translation".

Model improvement: chunk

Also since there is no re-order in ASR like neural machine translation (NMT), a Chunked multi-head attention (C-MHA) could be implemented in Speech-Transformer (S-Xfmr) instead of MHA, the core innovation of original Transformer.

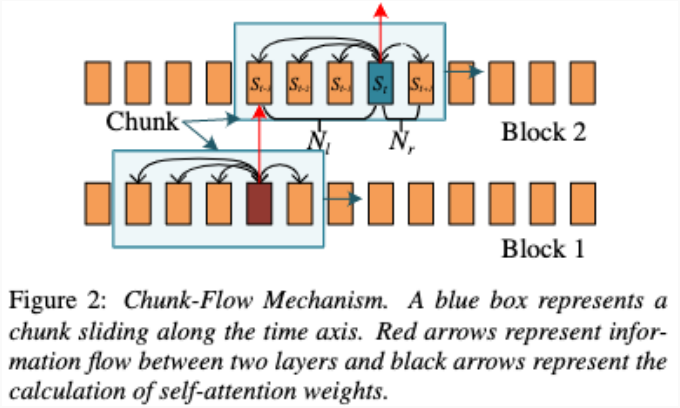


Figure 2: *Chunk-Flow Mechanism. A blue box represents a chunk sliding along the time axis. Red arrows represent information flow between two layers and black arrows represent the calculation of self-attention weights.*

(a)

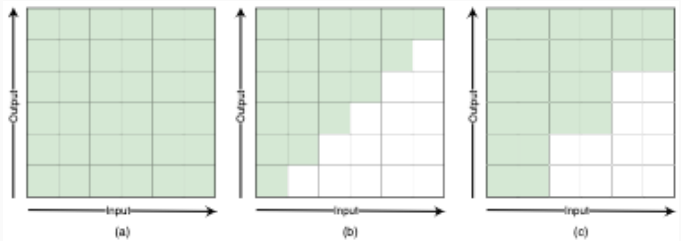


Figure 2: *Full attention, Left attention, Chunk Attention*

(b)

Figure 8: chunk flow attention (CL-MHA)[11] (a) and chunk attention[12] (C-MHA) (b)

[9] Tian et al., "Self-attention transducers for end-to-end speech recognition".
[10] Zhang et al., "Unified streaming and non-streaming two-pass end-to-end model for speech recognition".
[11] Tian et al., "Self-attention transducers for end-to-end speech recognition".

Training pipeline innovation

Weak distillation: from conventional model to E2E

As mentioned before, especially in one set ASR, like meeting transcription. One of problem is open set recognition. You never know what kind of topic will be in the meeting and what words will be spoken.

And mentioned before in Fig 1, conventional method contains pronunciation dict, which is hand-craft work, of-course, containing extra prior information and do help conventional method in NER.

Previous work also have shown that E2E method degrades on NER. Of course best way is adding labelled named entity data, but labelled data itself is expensive.

So one of other choice is distilling extra information from conventional method and unlabelled data to E2E-ASR model.

Weak distillation: seq2seq distillation

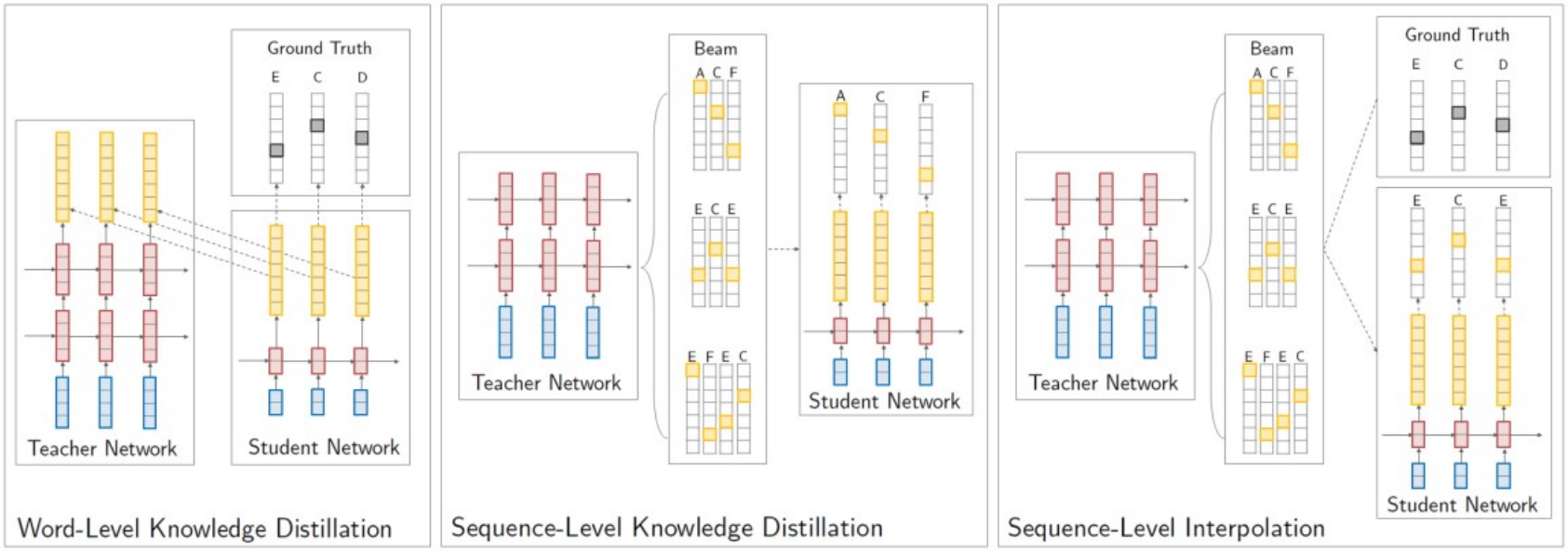


Figure 1: Overview of the different knowledge distillation approaches. In word-level knowledge distillation (left) cross-entropy is minimized between the student/teacher distributions (yellow) for each word in the actual target sequence (ECD), as well as between the student distribution and the degenerate data distribution, which has all of its probabilitiy mass on one word (black). In sequence-level knowledge distillation (center) the student network is trained on the output from beam search of the teacher network that had the highest score (ACF). In sequence-level interpolation (right) the student is trained on the output from beam search of the teacher network that had the highest *sim* with the target sequence (ECE)

Figure 9: seq2seq distillation[13]

[13] Kim and Rush, "Sequence-level knowledge distillation".

Software architecture for Iterative pseudo-labeling (IPL)

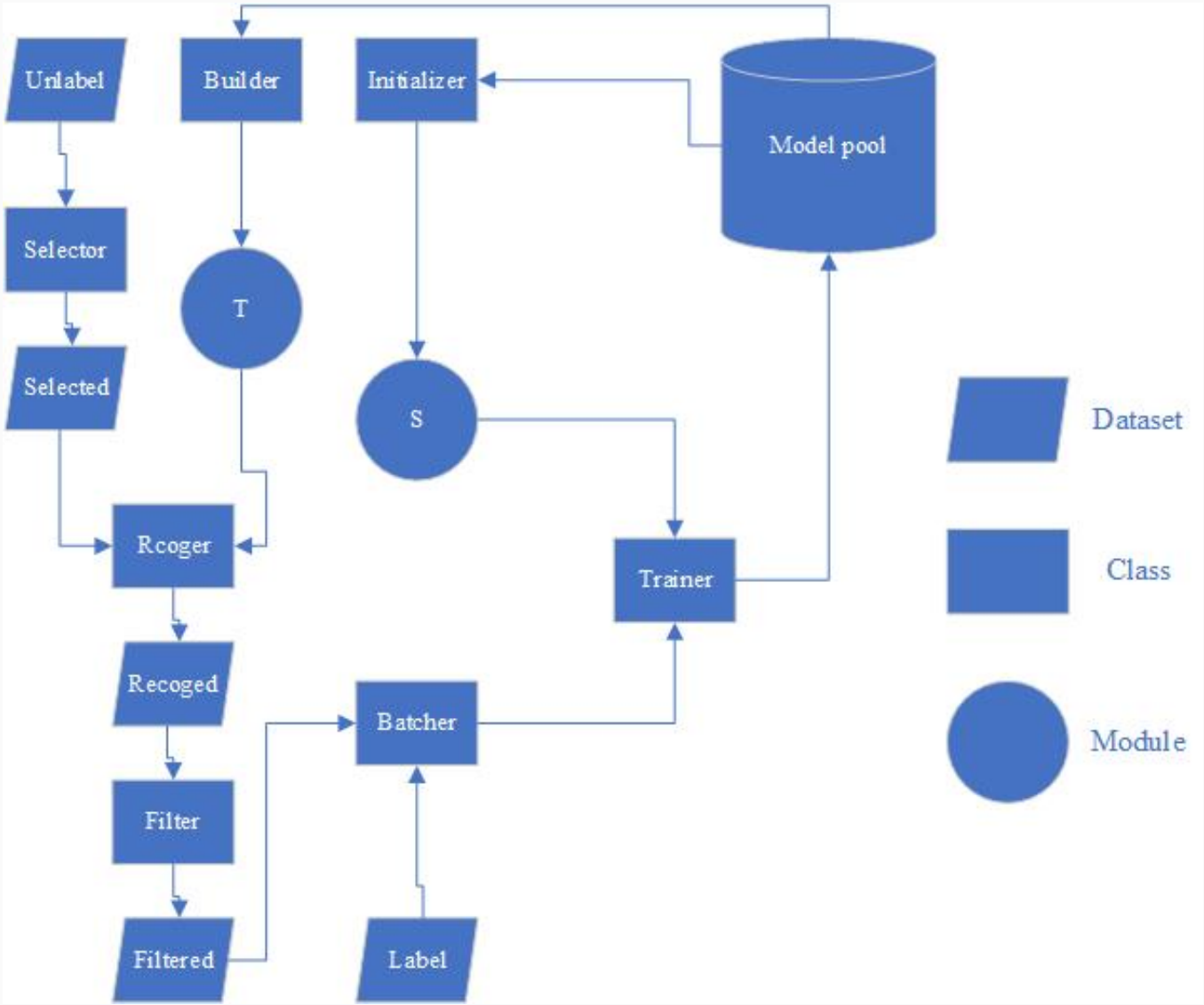


Figure 10: A common Software architecture for IPL

Architecture

```
selector = init_selector()
builder = init_builder()
predictor = init_predictor()
filter = init_predictor()
initializer = init_initializer()
for cycle in range(cycles):
  selected_data = selecter(cycle, unlabel_data)
  teacher_model = builder(model_pool)
  predicted_data = predictor(selected_data, teacher_model)
  filtered_data = filter(predicted_data)
  batcher = init_bather(filtered_data, label_data)
  student_model = initializer(model_pool)
  for epoch in epochs:
    trainer(student_model, batcher)
```

Plug-in description

Plug-in	Description
Selector	selection data from unlabelled dataset according index
Recoger	Recognition selected data with teacher model.
Filter	Filter selected data according score and prediction
Batcher	organizing unlabelled and label data to training
Trainer	Train student model with batcher
Builder	Build a teacher model
Initializer	Initializing the student model state dict

Table 1: Plug-in description

Architecture rule: Low-coupling, high cohesion and simple rule for reuse.

Large scale data model training acceleration in ZOOM

GTC22 zoom model Training[14]

Figure 11: Scale and Accelerate the Distributed Model Training in Kubernetes Cluster

Mainly from our talk in GTC22, focus on DDP communication acceleration today.

[14] *Scale and Accelerate the Distributed Model Training in Kubernetes Cluster.* https://www.nvidia.com/en-us/on-demand/session/gtcspring22-s42498/.

Technical Components[15]

- **DL Training**: Filter, Gradient Compression, PowerSGD, Pytorch Lighting, Determined-AI, Pytorch DDP
- **Machine Learning Framework**: Pytorch, Tensorflow, Horovod, MPI NCCL, APEX, Cuda, Tensor Core
- **Kubernetes**: SRIOV, Virtual Function, peer memory, Operator, PytorchJob, TFJob, MPIJob, Calico
- **OS**: RDMA, RoCE, GPUDirect, OFED, Firmware, Kernel module
- **Server**: GPU, Smart NIC(Mellanox, EFA), PCIe, NVLink, NVSwitch, NvME over Fabric
- **Networking**: PFC, ETS, LLDP, DCBX

[15] *Scale and Accelerate the Distributed Model Training in Kubernetes Cluster.* https://www.nvidia.com/en-us/on-demand/session/gtcspring22-s42498/.

Parallelism

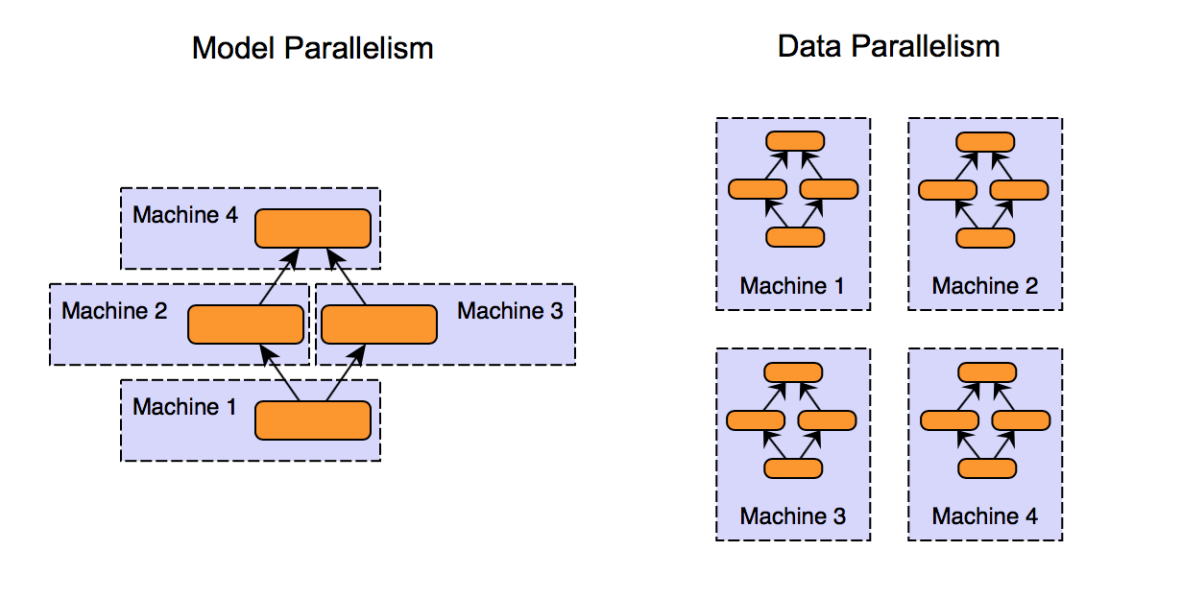


Figure 12: The comparison of model and data parallel

model parallelism: All workers train on same data. Parts of the model are distributed across GPUs.

data parallelism: All workers train on different data. All workers have the same copy of the model.

ASR model training situation analysis

Model usually LSTM/Transformer based, the number of parameter is around: 20M to 200M

Data from 10 hours labelled audio to 100, 000 hours.

The selection of Parallelism from model and data parallel: data parallel. For large scale data model training: Distributed Data Parallelism (DDP), run on multiple servers.

DDP

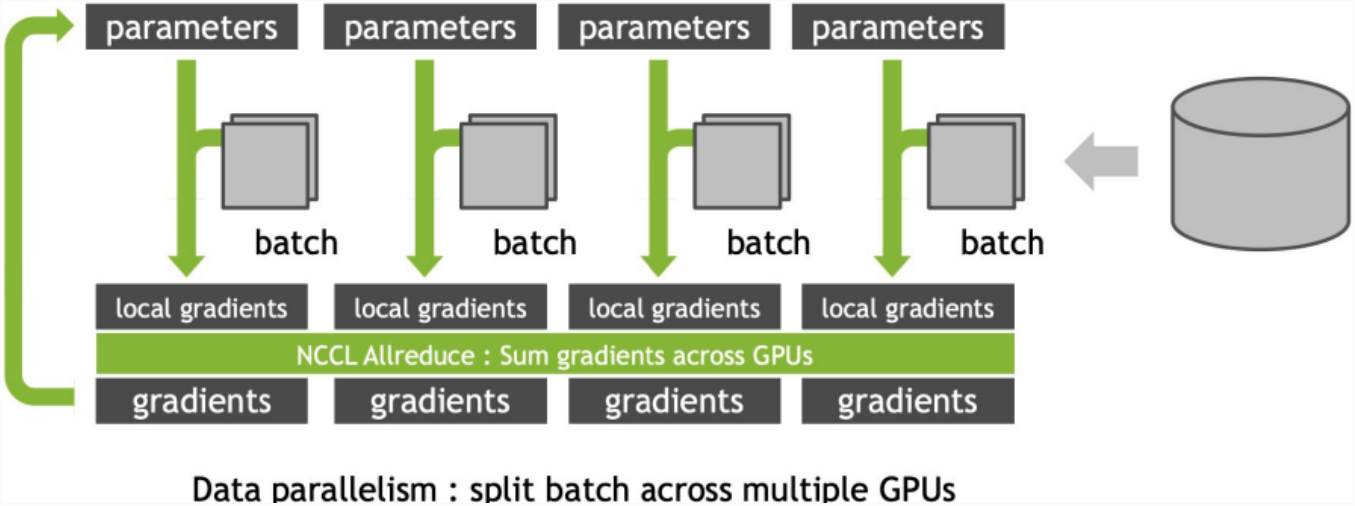


Figure 13: DDP

GPU communication:

- **Within the server**: are all gpu same
- **Between the servers**: network

DDP acceleration

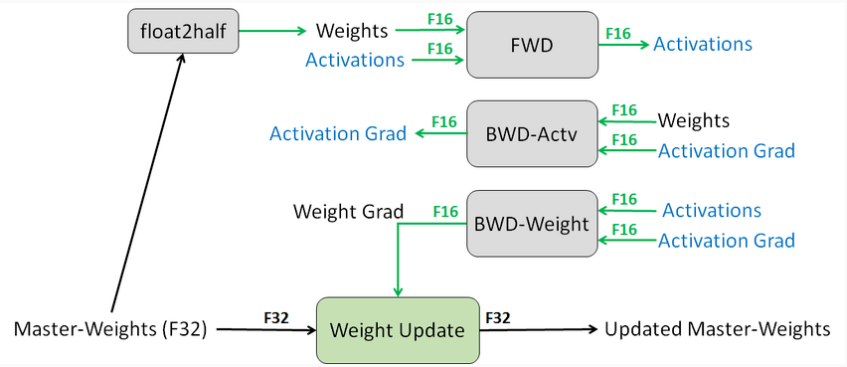


Figure 14: mix-precision example

- **Consume less memory**: Larger neural networks models or mini batches. Halves the size of activation and gradient tensors
- **Consume less memory bandwidth**: Speeding up data transfer operations. Halves memory traffic compared to FP32.
- **Accelerates math-bound operations**: Tensor Cores are 8x faster than FP32
- **Automatic**: Identifying the steps that require full precision and using 32-bit floating point for only those steps. Black/white list control

Smart NIC + RDMA + RoCE

- **Smart NIC**: A SmartNIC, or smart network interface card (NIC), is a programmable accelerator that makes data center networking, security and storage efficient and flexible. (Hardware level)
- **RDMA**: remote direct memory access from the memory of one server into that of another without involving either one's operating system, high-throughput, low-latency networking (OS level)
- **RoCE**: RDMA over Converged Ethernet (OS level)
- **OFED**: OpenFabrics Enterprise Distribution is open-source software for RDMA providing NIC firmware, drivers, kernel modules, and libraries (OS level)

RDMA/ROCE

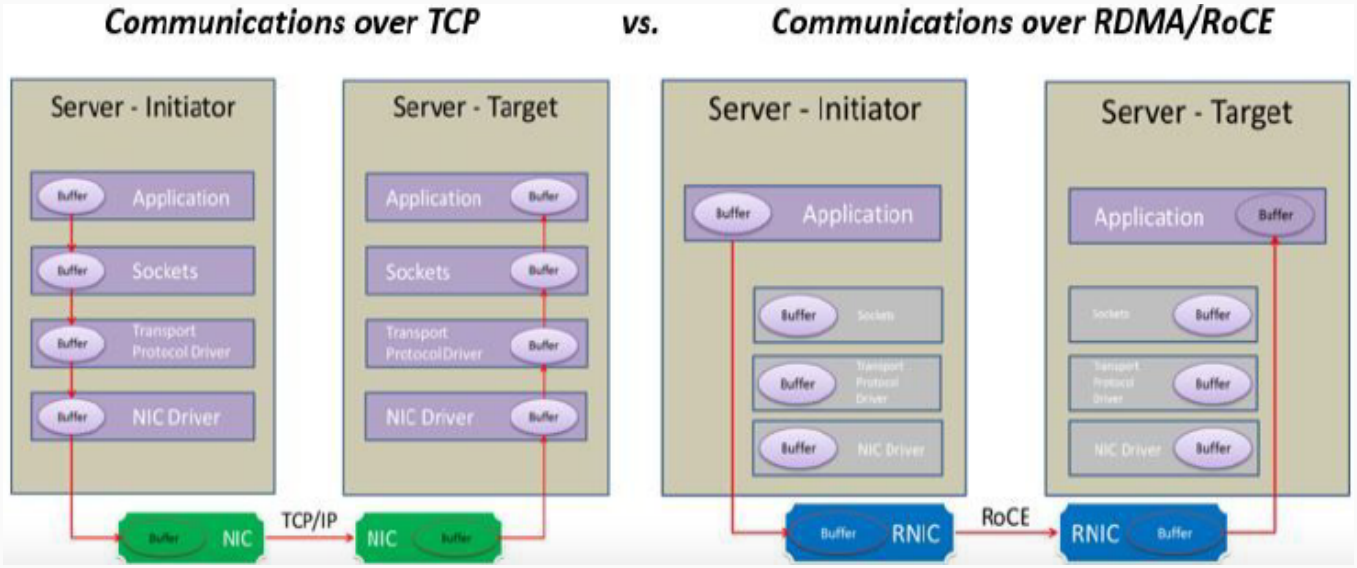
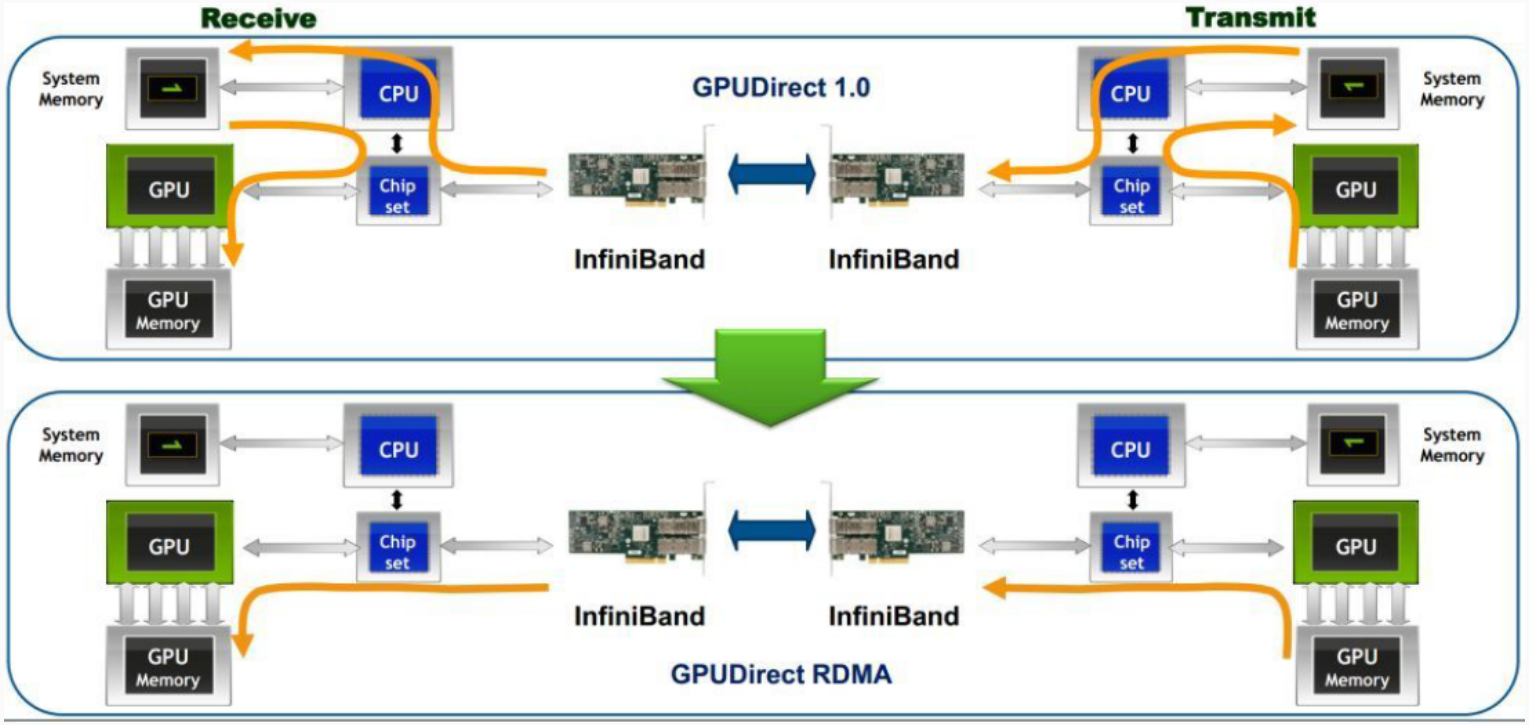


Figure 15: Avantage of RDMA/RoCE

- **Zero-copy** no need to copy user application buffers to a dedicated NIC buffers.
- **Direct user space** hardware interface which bypasses the kernel and TCP/IP in data path

The Infiniband transport over UDP encapsulation. Available for all Ethernet speeds 10 –100G

GPUDirect RDMA



CPU utilization –CPU not involved in the DMA operations

Figure 16: GPUDirect RDMA

Implement: Nvidia Network Operator

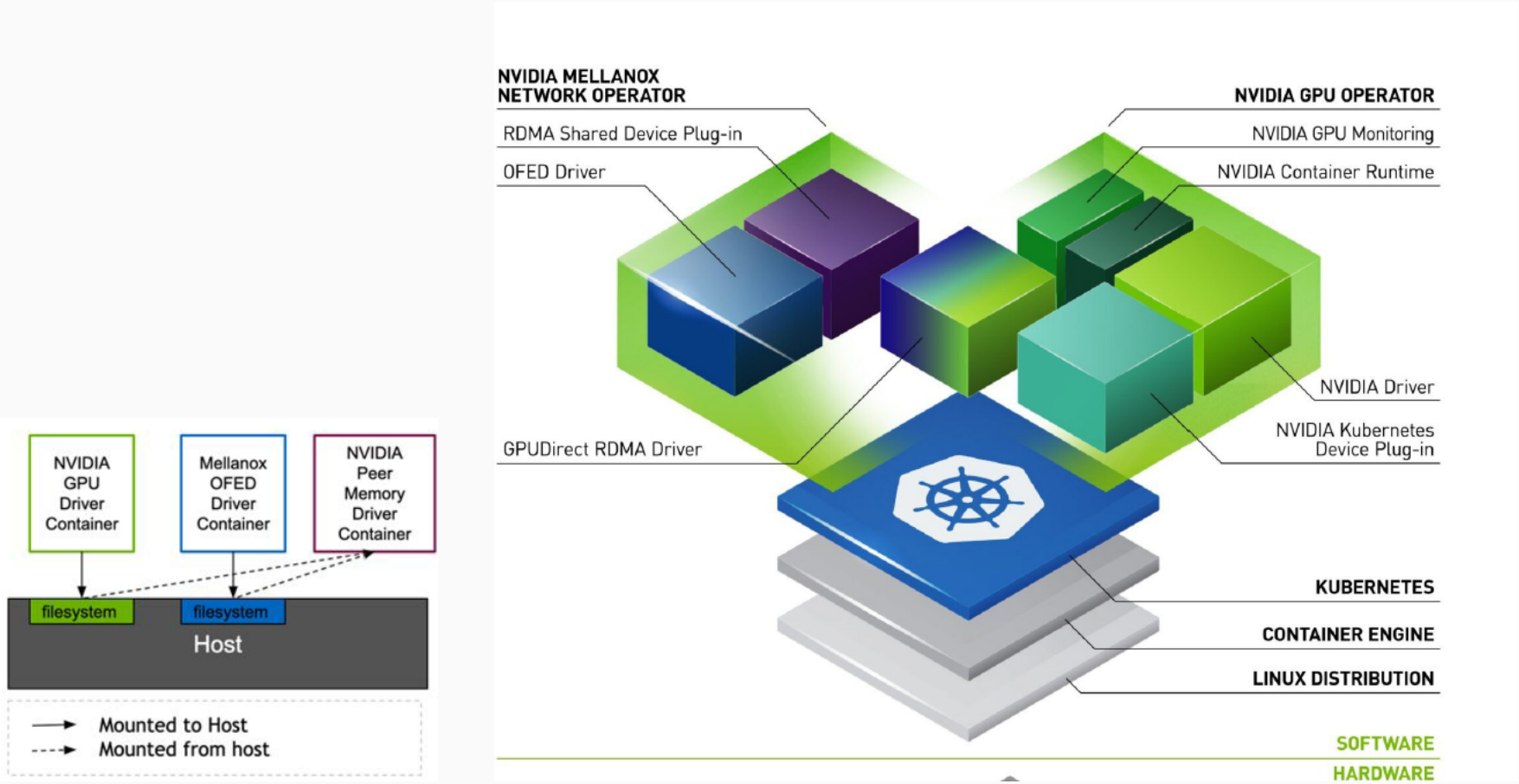


Figure 17: Nvidia Network Operator

RDMA/RoCE test result

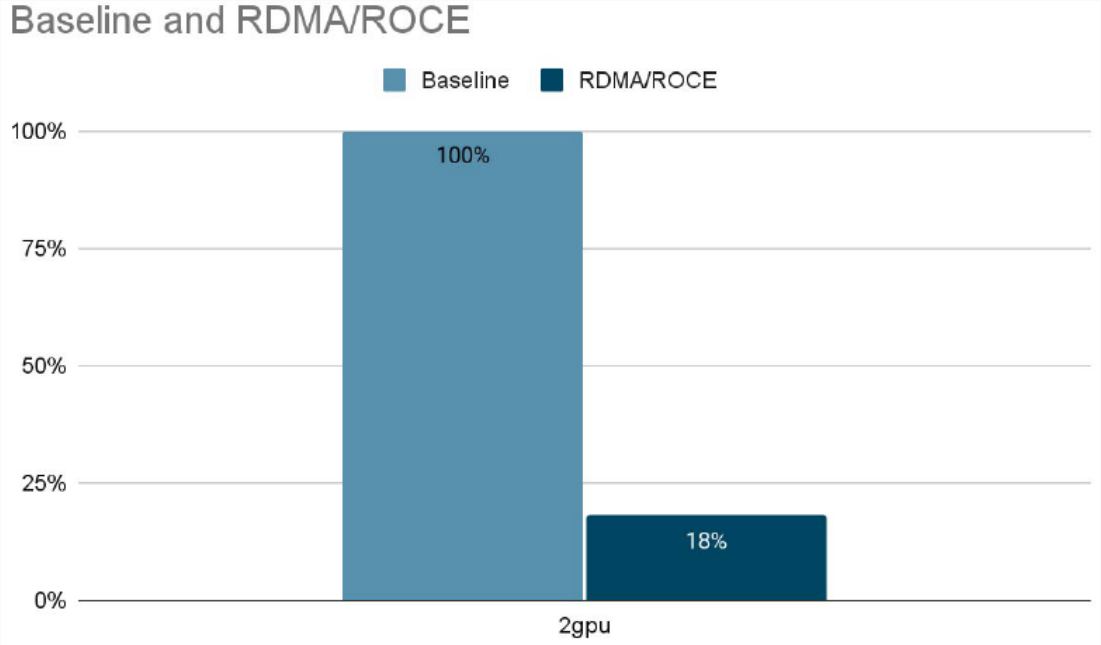


Figure 18: RDMA/RoCE test result

Hardware update to enable faster multi-node communication. With RDMA/ROCE enabled, the multi-node 2gpu training is around 5x faster

Gradient compression DDP

- **grad_fp16**: sending gradient in 16 bits instead of 32 bits thus achieve 2x compression
- **grad_powersgd**: using smaller matrix to estimate the original gradient, it can achieve up to 1000x compression but with some cost of accuracy and additional computation time.

Gradient compression test result

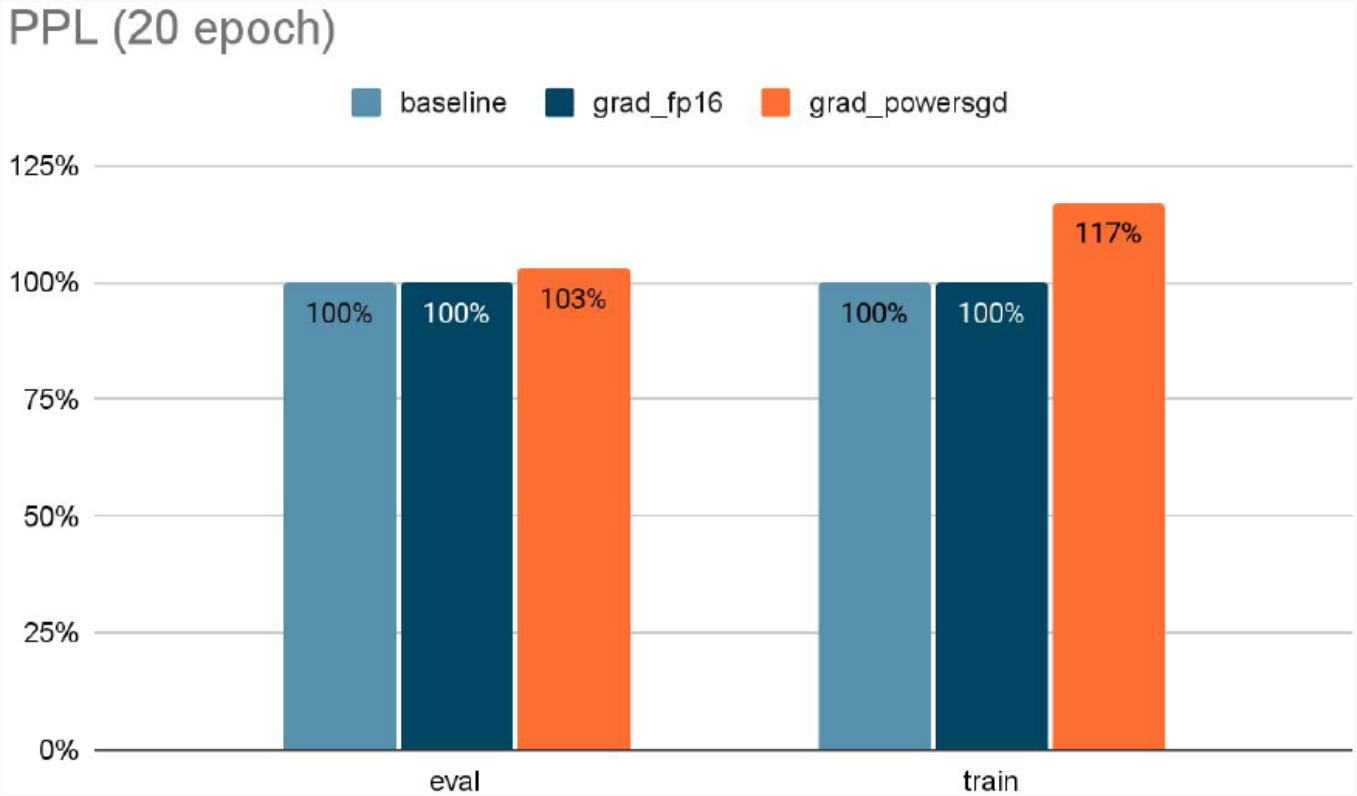


Figure 19: Gradient compression accuracy comparison in 12 gpu setup

From the chart above, **grad_fp16** leads to almost no accuracy loss while **grad_powersgd** still can lead to some accuracy loss. We also noticed a early model convergence at epoch 13 for **grad_powersgd**.

Summary and next step

Summary and next step

Summary:

Model early version of semi-supervised ASR software architecture.

Training DDP GPUdirect acceleration, mixed-precision, GPUDirect RDMA. gp_FP16, gp_powerSGD

Next step:

Model Tuning and upgrade semi-supervised in ASR, beam search acceleration.

Training Communication Efficiency Improvement (1-bit Adam optimizer), Model Distributed Training (ZeRO optimizer), Data IO Improvement (On-the-fly Caching, Load to Memory)

Presenter

Haoyu (Charlie) Tang
haoyu.tang@zoom.us

Working in Zoom AI/ML Engineer department

Graduated from Norwegian University of Science and Technology (NTNU)

Responsible for ASR training & deployment

Github **@qmpzzpm**
https://github.com/qmpzzpmq

zhihu/知乎 **@ 戚少商**
https://www.zhihu.com/people/qi-shao-shang-96

Figure 20: Presenter Haoyu (Charlie) Tang

Any Questions?

Also, hiring:

Video, Audio, Speech, NLP, **前端**, ServiceDevops, **测试**, Java/C++ **全职/实习生**

Base: **杭州**, **苏州**, **合肥**

To haoyu.tang@zoom.us

References

- https://www.authot.com/en/2016/09/09/automatic-speech-recognition-system/.
- https://sonix.ai/history-of-speech-recognition.
- Chan, William et al. “Listen, attend and spell”. In: *arXiv preprint arXiv:1508.01211* (2015).
- Dong, Linhao, Shuang Xu, and Bo Xu. “Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2018, pp. 5884–5888.
- Kim, Yoon and Alexander M Rush. “Sequence-level knowledge distillation”. In: *arXiv preprint arXiv:1606.07947* (2016).
- *Scale and Accelerate the Distributed Model Training in Kubernetes Cluster*. https://www.nvidia.com/en-us/on-demand/session/gtcspring22-s42498/.
- Tian, Zhengkun et al. “Self-attention transducers for end-to-end speech recognition”. In: *arXiv preprint arXiv:1909.13037* (2019).


Watanabe, Shinji et al. “Hybrid CTC/attention architecture for end-to-end speech recognition”. In: *IEEE Journal of Selected Topics in Signal Processing* 11.8 (2017), pp. 1240–1253.


Wu, Dekai, Marine Carpuat, and Yihai Shen. “Inversion transduction grammar coverage of arabic-english word alignment for tree-structured statistical machine translation”. In: *2006 IEEE Spoken Language Technology Workshop*. IEEE. 2006, pp. 234–237.


Zhang, Binbin et al. “Unified streaming and non-streaming two-pass end-to-end model for speech recognition”. In: *arXiv preprint arXiv:2012.05481* (2020).