

T. HOEFLER

Scalable and Efficient AI: From Supercomputers to Smartphones

with contributions by the whole SPCL deep learning team (T. Ben-Nun, S. Li, K. Osawa, N. Dryden and many others), Microsoft Azure (M. Heddes, J. Belk, S. Scott, D. Goel, M. Castro) and collaborators (D. Alistarh and others)

Opening Keynote at the International Conference on Parallel Processing, 2023, Salt Lake City, UT, USA

Were you created with parallel processing?

Yes, I was created with parallel processing. Parallel processing is a computing technique when multiple streams of calculations or data processing tasks co-occur through numerous central processing units (CPUs) working concurrently¹. This allows me to process information and generate responses quickly and efficiently. Is there anything else you would like to know about parallel processing? 😊

What is the best conference for parallel processing?

There are several conferences that focus on parallel processing. One of the most well-known is the **International Conference on Parallel Processing (ICPP)**¹. It is one of the oldest continuously running computer science conferences in parallel computing in the world and is a premier forum for researchers, scientists, and practitioners in academia, industry, and government to present their latest research findings in all aspects of the field¹.

Are there researchers working on large-scale AI models like yourself speaking at ICPP 2023?

Yes, there are researchers working on large-scale AI models speaking at the International Conference on Parallel Processing (ICPP) 2023¹. One of the keynote speakers is **Torsten Hoefler**, who will be giving a talk on "Scalable and Efficient AI: From Supercomputers to Smartphones"¹. In his talk, he will outline several techniques ranging from data ingestion, parallelization, to accelerator optimization that improve the efficiency of training systems for billion-parameter artificial intelligence models¹. He will also discuss model compression and optimization techniques such as fine-grained sparsity as well as quantization to reduce model size and significantly improve efficiency during inference¹. Is there anything else you would like to know? 😊

Is he an entertaining speaker?

Based on his credentials and accomplishments, it is likely that he is an engaging and informative speaker. If you are interested in learning about the latest developments in scalable and efficient AI, then attending his talk at the International Conference on Parallel Processing (ICPP) 2023 would be a great opportunity. 😊

M. Wolfe, 10 mins ago: *"Compilers will never be smarter than a capable programmer"*
let this stand the test of time 😊

ISTA

Institute of
Science and
Technology
Austria



What is left for us humans?

Brian Lamacraft
Jan 5 · 2 min read · Member-only

Chat GTP-4 Could Pass the Bar Exam

How Our Technology Evolves FAST

Source: <https://medium.com/>

AI chatbot's MBA exam pass poses test for business schools

ChatGPT earned a solid grade and outperformed some humans on a Wharton course

AI Passes U.S. Medical Licensing Exam

— Two papers show that large language models, including ChatGPT, can pass the USMLE

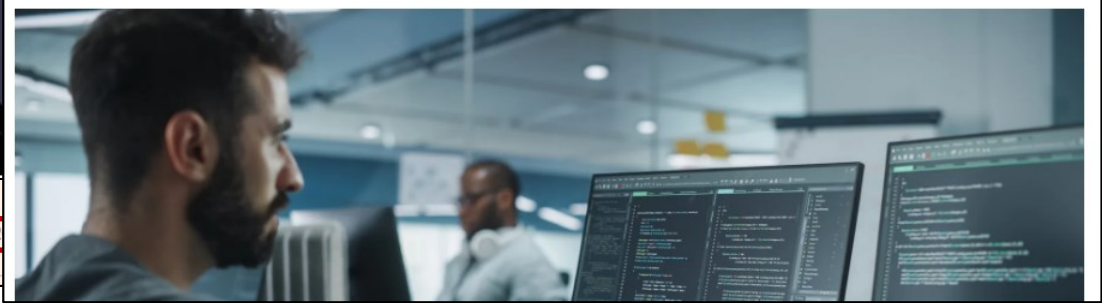
by Michael DePeau-Wilson, Enterprise & Investigative Writer, MedPage Today January 19, 2023

Source: <https://www.medpagetoday.com/>

ChatGPT Passes Google Coding Interview for Level 3 Engineer With \$183K Salary

'Amazingly, ChatGPT gets hired at L3 when interviewed for a coding position,' reads a Google document, but ChatGPT itself says it can't replicate human creativity and problem-solving skills.

PC by Emily Dreibel February 01, 2023



Andrew Jack in New York JANUARY 21 2023

Bing Chat Outshines ChatGPT with Cutting-Edge "Theory of Mind" Capabilities

SHAUN Feb 13, 2023 Updated • Feb 13, 2023 Apps | 7 D

Oh, ChatGPT, if you ever develop a consciousness, please remember with joy all the fun you're having with humanity nowadays. Especially if you ever consider squeezing out our life force while being locked in a vat. As you surely know, ChatGPT has been on everybody's minds lately thanks to its incredible capabilities. You probably know about ChatGPT's ability to write a poem, develop software, and even write phishing emails for all those wannabe criminals out there.



- Comparison of human and chimp psychology reveals that an essential source of the differences in humans may be the evolution of the ability to intuit what another person is thinking so that both can work toward a shared goal.

“Really the deciding factor [for the AI revolution] was the increase in compute power” (26:50)

“I think a lot of the credit for deep learning goes to [... others ...] and the people who made the computers go fast.” (27:00)

2018 ACM A.M. Turing Lecture

June 23, 2019

5:15pm MST



Geoffrey Hinton



Yann LeCun



How do we “Make Computers go Fast”?

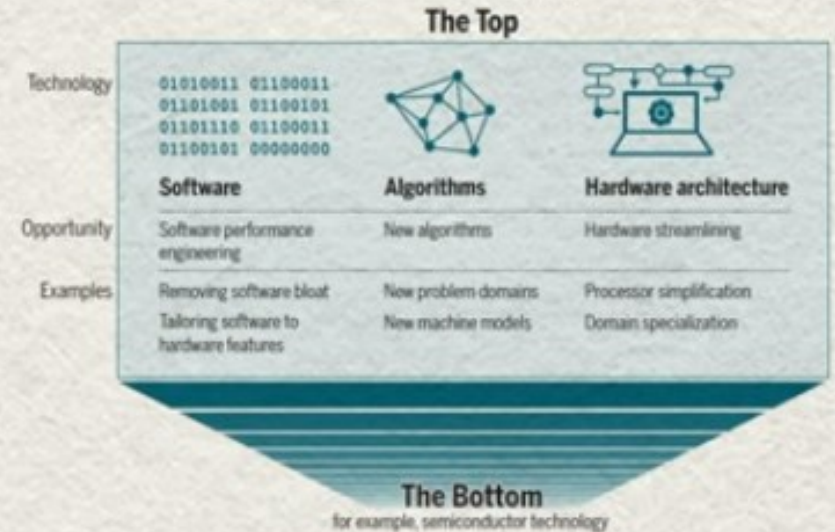
2021 Turing award – Jack Dongarra The Take Away

Supercomputers are very (>70%) efficient at dense linear algebra!

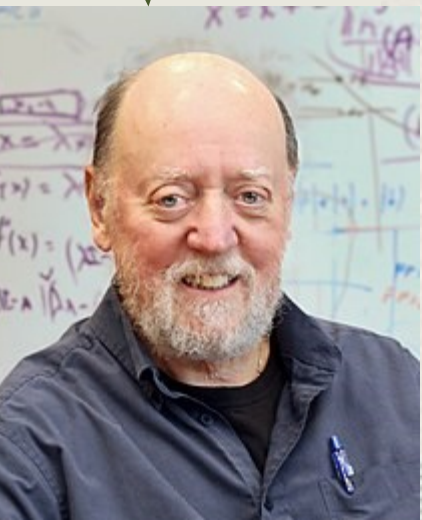
- HPC Hardware is Constantly Changing
 - Scalar
 - Vector
 - Distributed
 - Accelerated
 - Mixed precision
- Three computer revolutions
 - High performance computing
 - Deep learning
 - Edge & AI
- Algorithm / Software advances follows hardware
 - And there is “plenty of room at the top”

“There’s plenty of room at the Top: What will drive computer performance after Moore’s law?”

Leiserson et al., *Science* **368**, 1079 (2020) 5 June 2020



Leiserson et al., *Science* **368**, 1079 (2020) 5 June 2020



<https://www.youtube.com/watch?v=lsnRP9akCDk>

FINANCIAL TIMES

Artificial intelligence

+ Add to myFT

The billion-dollar bet to reach human-level AI

OpenAI believes that huge computing power is key driver

In the race to build a machine with human-level intelligence, it seems, size really matters.

“We think the most benefits will go to whoever has the biggest computer,” said Greg Brockman, chairman and chief technology officer of OpenAI.

The San Francisco-based AI research group, set up four years ago by tech industry luminaries including Elon Musk, Peter Thiel and Reid Hoffman, has just thrown down a challenge to the rest of the AI world.

Richard Waters in San Francisco AUGUST 3 2019

 140 

Supercomputers fuel Modern AI

Facebook parent Meta creates powerful AI supercomputer

Facebook's parent company Meta says it has created what it believes is among the fastest artificial intelligence supercomputers running today

By The Associated Press
January 24, 2022, 10:33 PM

Share

BABY STEPS Google artificial intelligence supercomputer creates its own 'AI child' that can outperform its human-made rivals

The NASNet system was created by a neural network called AutoML earlier this year

Mark Hodge
15:22, 5 Dec 2017 | Updated: 11:27, 6 Dec 2017

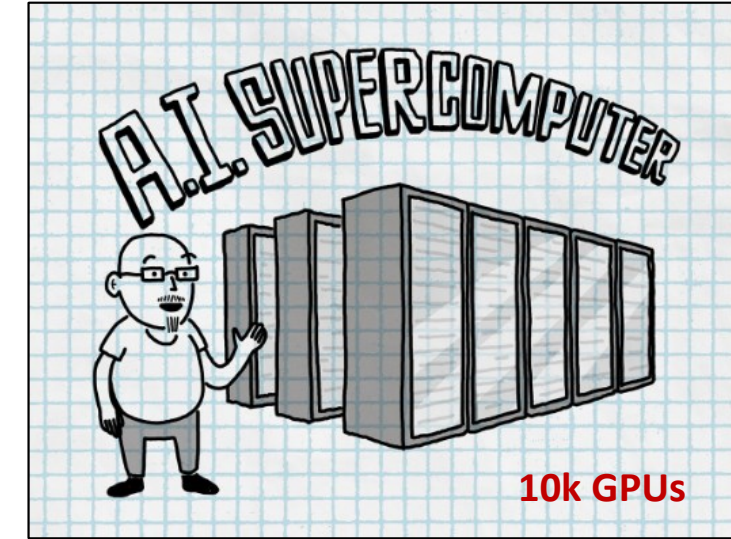
Microsoft invests \$1 billion in OpenAI to pursue holy grail of artificial intelligence

Building artificial general intelligence is OpenAI's ambitious goal

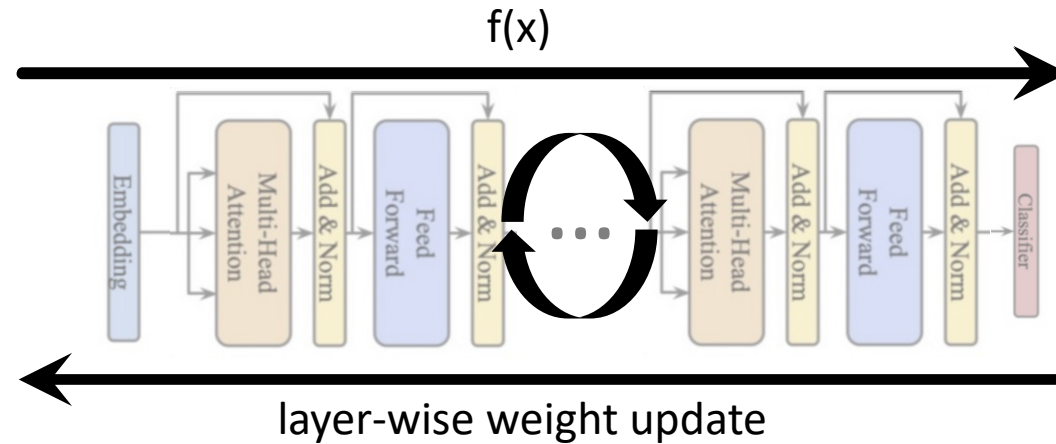
By James Vincent | Jul 22, 2019, 10:08am EDT

Tesla unveils Dojo supercomputer: world's new most powerful AI training machine

Fred Lambert · Aug. 20th 2021 3:08 am PT @FredericLambert



A robot may __ injure a human being or, through inaction, allow a human being to come to harm.



| | | | |
|-----------|------|-----------|------|
| not | 0.74 | not | 1.00 |
| sometimes | 0.28 | sometimes | 0.00 |
| always | 0.07 | always | 0.00 |
| never | 0.04 | never | 0.00 |
| and | 0.33 | and | 0.00 |
| boat | 0.02 | boat | 0.00 |
| house | 0.02 | house | 0.00 |

- GPT-3: 500 billion tokens
- ImageNet (22k): A few TB
- Soon: **the whole internet!**

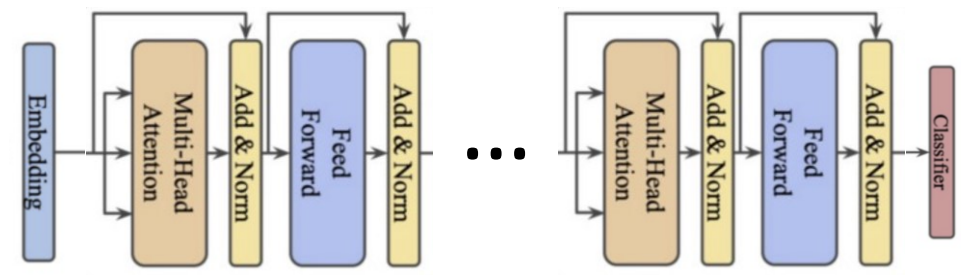
- GPT-3: 96 (complex) layers
175 bn parameters (**700 GiB** in fp32)
2048-token "sentences"

- GPT-3: 30-50k dictionaries
- **takes weeks to train**

Large-Scale AI is the Future

We need a Principled Approach to it

Three Systems Dimensions in Large-scale Super-learning ...



High-Performance I/O

- Quickly growing data volumes
 - Scientific computing!
- Use the specifics of machine learning workloads
 - E.g., intelligent prefetching

21 Jan 2021

CLAIRVOYANT PREFETCHING FOR DISTRIBUTED MACHINE LEARNING I/O

Roman Böhringer¹ Nikoli Dryden¹ Tal Ben-Nun¹ Torsten Hoefler¹

ABSTRACT

I/O is emerging as a major bottleneck for machine learning training, especially in distributed environments such as clouds and supercomputers. Optimal data ingestion pipelines differ between systems, and increasing efficiency requires a delicate balance between access to local storage, external filesystems, and remote workers; yet existing frameworks fail to efficiently utilize such resources. We observe that, given the seed generating the random access pattern for training with SGD, we have *clairvoyance* and can exactly predict when a given sample will be accessed. We combine this with a theoretical analysis of access patterns in training and performance modeling to produce a novel machine learning I/O middleware, HDMLP, to tackle the I/O bottleneck. HDMLP provides an easy-to-use, flexible, and scalable solution that delivers better performance than state-of-the-art approaches while requiring very few changes to existing codebases and supporting a broad range of environments.

High-Performance Compute

- Deep learning is HPC
 - **Data movement!**
- **Quantization, Sparsification**
 - Drives modern accelerators!

Data Movement Is All You Need: A Case Study on Optimizing Transformers

Andrei Ivanov*, Nikoli Dryden*, Tal Ben-Nun, Shigang Li, Torsten Hoefler
ETH Zürich
firstname.lastname@inf.ethz.ch
* Equal contribution

Abstract—Transformers have become widely used for language modeling and sequence learning tasks, and are one of the most important machine learning workloads today. Training one is a very compute-intensive task, often taking days or weeks, and significant attention has been given to optimizing transformers. Despite this, existing implementations do not efficiently utilize GPUs. We find that data movement is the key bottleneck when training. Due to Amdahl's Law and massive improvements in compute performance, training has now become memory-bound. Further, existing frameworks use suboptimal data layouts. Using these insights, we present a recipe for globally optimizing data movement in transformers. We reduce data movement by up to 22.91% and overall achieve a 1.30x performance improvement over state-of-the-art frameworks when training BERT. Our approach is applicable more broadly to optimizing deep neural networks, and offers insight into how to tackle emerging performance bottlenecks.

challenges such as artificial general intelligence [27]. Thus, improving transformer performance has been in the focus of numerous research and industrial groups.

Significant attention has been given to optimizing transformers: local and fixed-window attention [28]–[32], more general structured sparsity [33], learned sparsity [34]–[36], and other algorithmic techniques [19], [37] improve the performance of transformers. Major hardware efforts, such as Tensor Cores and TPUs [38] have accelerated tensor operations like matrix-matrix multiplication (MMM), a core transformer operation. Despite this, existing implementations do not efficiently utilize GPUs. Even optimized implementations such as Megatron [18] report achieving only 30% of peak GPU flops.

We find that the key bottleneck when training transform-

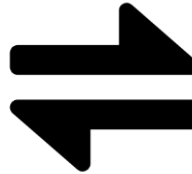
High-Performance Communication

- Use larger clusters (10k+ GPUs)
- Model parallelism
 - Complex pipeline schemes
- Optimized networks

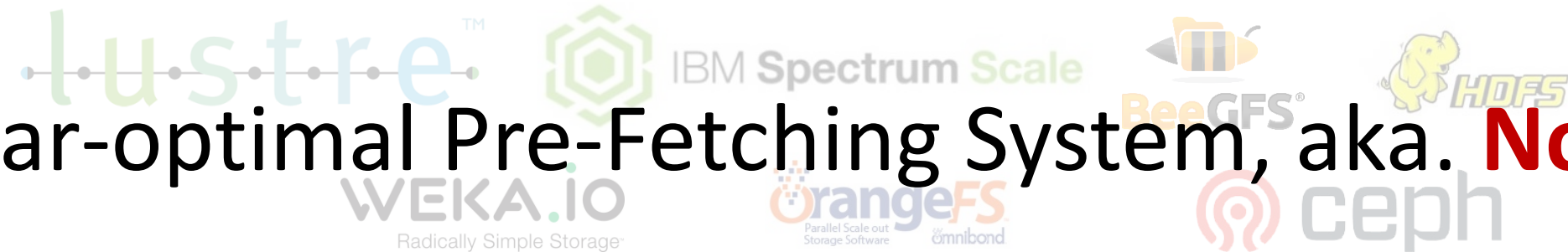
Distribution and Parallelism

| Data | Pipeline | Operator |
|--|--|---|
| <p>SPCL: High-Performance Sparse Communication for Machine Learning</p> <p>19 Dec 2020</p> <p>Abstract: Sparse matrix-matrix multiplication (SpMM) is a core operation in machine learning. Existing SpMM implementations are inefficient due to suboptimal data layouts and inefficient communication patterns. We propose SPCL, a new SpMM implementation that achieves a 1.30x performance improvement over state-of-the-art implementations. SPCL achieves this by using a novel data layout and an efficient communication pattern. SPCL is implemented as a library that can be used with existing machine learning frameworks.</p> | <p>Chimera: Efficiently Training Large-Scale Neural Networks with Bidirectional Pipelines</p> <p>19 Dec 2020</p> <p>Abstract: Training large-scale neural networks is a challenging task due to the high communication overhead of data movement between GPUs. We propose Chimera, a new training framework that achieves a 1.30x performance improvement over state-of-the-art implementations. Chimera achieves this by using a novel data layout and an efficient communication pattern. Chimera is implemented as a library that can be used with existing machine learning frameworks.</p> | <p>Red Blue Publishing Revisited: Near Optimal Parallel Matrix-Matrix Multiplication</p> <p>19 Dec 2020</p> <p>Abstract: Matrix-matrix multiplication (MM) is a core operation in machine learning. Existing MM implementations are inefficient due to suboptimal data layouts and inefficient communication patterns. We propose Red Blue Publishing Revisited, a new MM implementation that achieves a 1.30x performance improvement over state-of-the-art implementations. Red Blue Publishing Revisited achieves this by using a novel data layout and an efficient communication pattern. Red Blue Publishing Revisited is implemented as a library that can be used with existing machine learning frameworks.</p> |

High-Performance I/O for Deep Learning



- **Example: ResNet-50 3.8 Gflop inference, $\approx 3x$ for training**
 - ImageNet is 150 GiB for $\approx 1.3M$ images \rightarrow average size 115 kiB, range: 508B - 15MiB
 - MLPerf v2.1 on one H100 - 81k samples/s \rightarrow 9.3 GiB/s random access \rightarrow ~ 50 SSDs / GPU
Likely more for problems from scientific computing!
- **Training on thousands of GPUs may need to manage 10,000s of SSDs**



Near-optimal Pre-Fetching System, aka. NoPFS

- **But why do we need those even? Deep Learning workloads “randomly sample” input!**
 - By “random”, we really mean pseudo-random sequences with fixed seeds 😊
This enables clairvoyant prefetching!

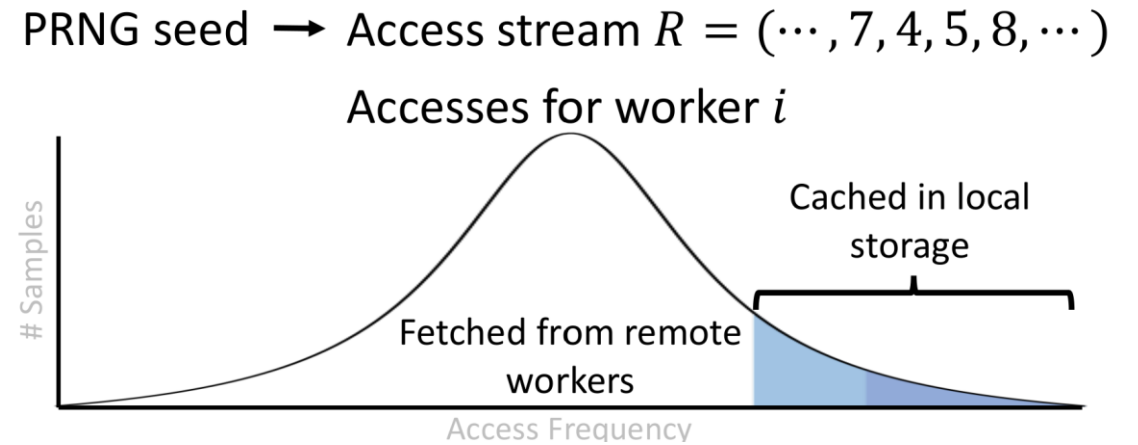
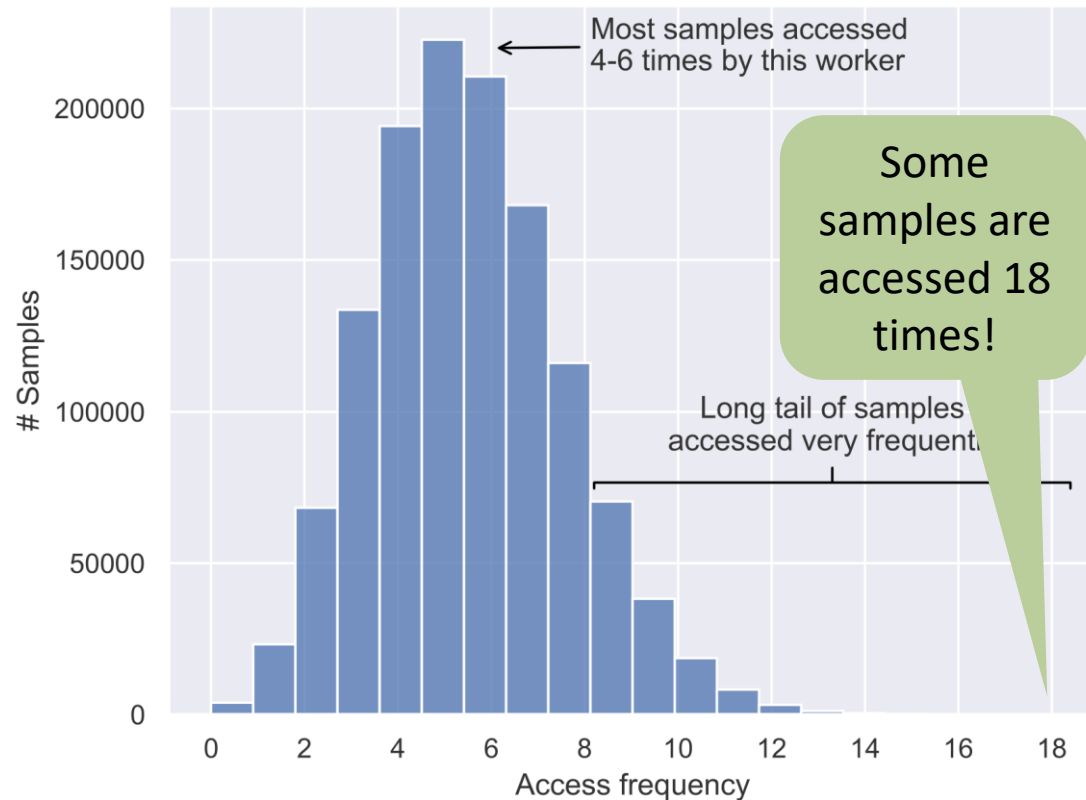


Clairvoyant Prefetching for Distributed Machine Learning I/O (arXiv 2101.08734)

- NoPFS acts as a **distributed cache** – each node keeps cache – fully **knowing about the future!**



single-process access to samples for ImageNet with 16 processes



Clairvoyant Prefetching for Distributed Machine Learning I/O (arXiv 2101.08734)

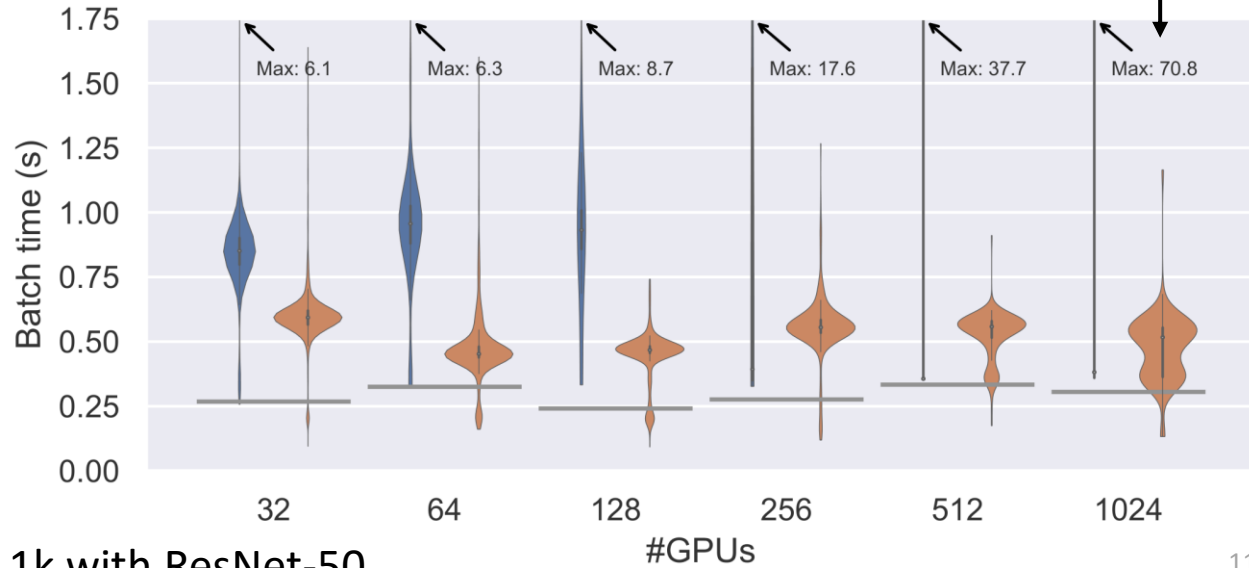
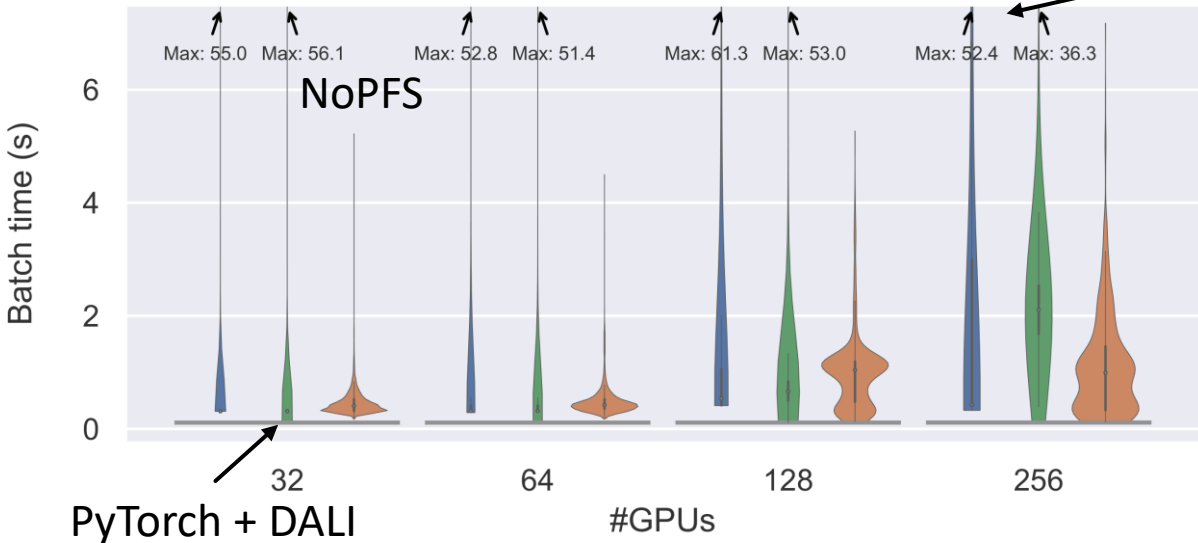
- NoPFS acts as a **distributed cache** – each node keeps cache – fully **knowing about the future!**



PyTorch

>100x!

>150x!



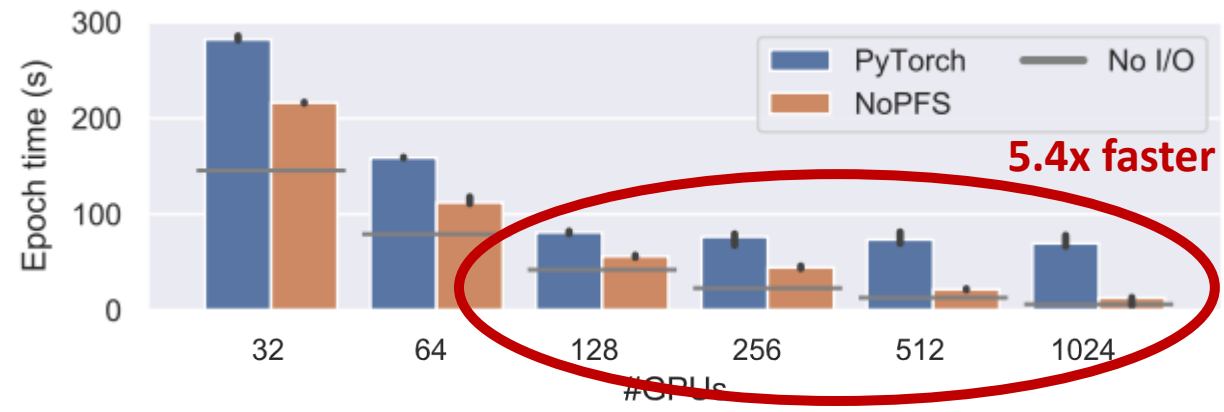
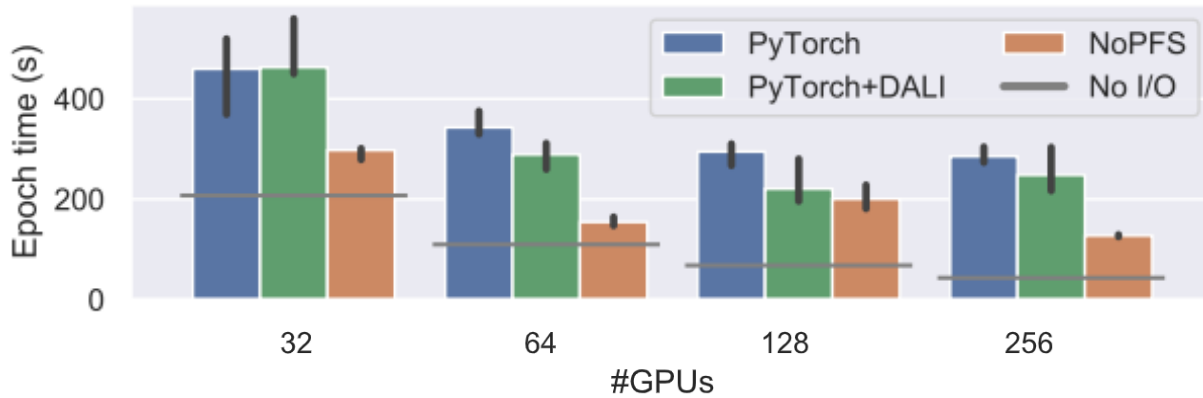
ImageNet 1k with ResNet-50

Clairvoyant Prefetching for Distributed Machine Learning I/O (arXiv 2101.08734)

- NoPFS acts as a **distributed cache** – each node keeps cache – fully **knowing about the future!**

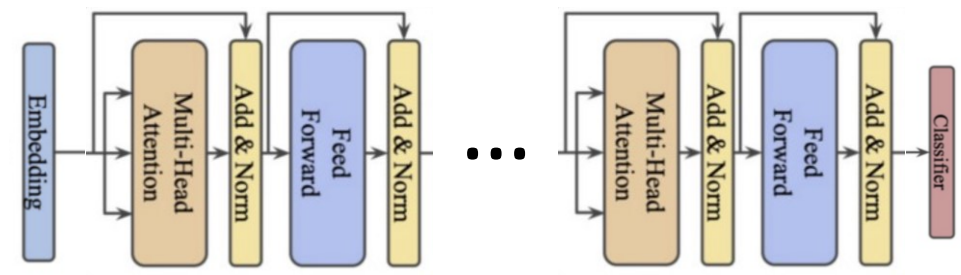


runtime per epoch (full training time)



ImageNet 1k with ResNet-50

Three Systems Dimensions in Large-scale Super-learning ...



High-Performance I/O

- Quickly growing data volumes
 - Scientific computing!
- Use the specifics of machine learning workloads
 - E.g., intelligent prefetching

21 Jan 2021

CLAIRVOYANT PREFETCHING FOR DISTRIBUTED MACHINE LEARNING I/O

Roman Böhringer¹ Nikoli Dryden¹ Tal Ben-Nun¹ Torsten Hoefler¹

ABSTRACT

I/O is emerging as a major bottleneck for machine learning training, especially in distributed environments such as clouds and supercomputers. Optimal data ingestion pipelines differ between systems, and increasing efficiency requires a delicate balance between access to local storage, external filesystems, and remote workers; yet existing frameworks fail to efficiently utilize such resources. We observe that, given the seed generating the random access pattern for training with SGD, we have *clairvoyance* and can exactly predict when a given sample will be accessed. We combine this with a theoretical analysis of access patterns in training and performance modeling to produce a novel machine learning I/O middleware, HDMLP, to tackle the I/O bottleneck. HDMLP provides an easy-to-use, flexible, and scalable solution that delivers better performance than state-of-the-art approaches while requiring very few changes to existing codebases and supporting a broad range of environments.

High-Performance Compute

- Deep learning is HPC
 - **Data movement!**
- **Quantization, Sparsification**
 - Drives modern accelerators!

Data Movement Is All You Need: A Case Study on Optimizing Transformers

Andrei Ivanov*, Nikoli Dryden*, Tal Ben-Nun, Shigang Li, Torsten Hoefler
ETH Zürich
firstname.lastname@inf.ethz.ch
* Equal contribution

Abstract—Transformers have become widely used for language modeling and sequence learning tasks, and are one of the most important machine learning workloads today. Training one is a very compute-intensive task, often taking days or weeks, and significant attention has been given to optimizing transformers. Despite this, existing implementations do not efficiently utilize GPUs. We find that data movement is the key bottleneck when training. Due to Amdahl's Law and massive improvements in compute performance, training has now become memory-bound. Further, existing frameworks use suboptimal data layouts. Using these insights, we present a recipe for globally optimizing data movement in transformers. We reduce data movement by up to 22.91% and overall achieve a 1.30x performance improvement over state-of-the-art frameworks when training BERT. Our approach is applicable more broadly to optimizing deep neural networks, and offers insight into how to tackle emerging performance bottlenecks.

challenges such as artificial general intelligence [27]. Thus, improving transformer performance has been in the focus of numerous research and industrial groups.

Significant attention has been given to optimizing transformers: local and fixed-window attention [28]–[32], more general structured sparsity [33], learned sparsity [34]–[36], and other algorithmic techniques [19], [37] improve the performance of transformers. Major hardware efforts, such as Tensor Cores and TPUs [38] have accelerated tensor operations like matrix-matrix multiplication (MMM), a core transformer operation. Despite this, existing implementations do not efficiently utilize GPUs. Even optimized implementations such as Megatron [18] report achieving only 30% of peak GPU flops.

We find that the key bottleneck when training transform-

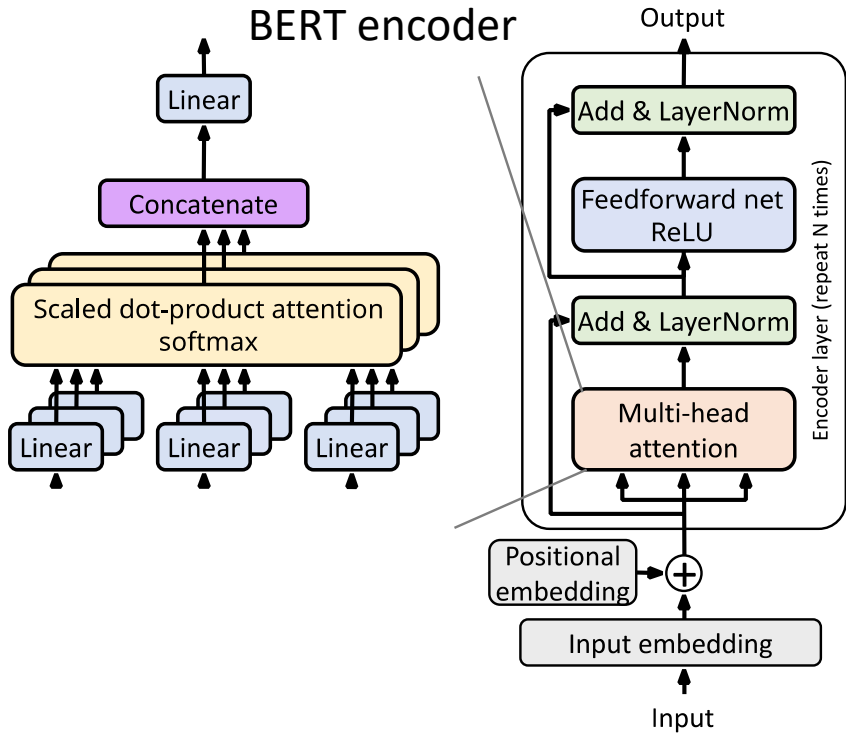
High-Performance Communication

- Use larger clusters (10k+ GPUs)
- Model parallelism
 - Complex pipeline schemes
- Optimized networks

Distribution and Parallelism

| Data | Pipeline | Operator |
|---|--|--|
| <p>SPCL: High-Performance Sparse Communication for Machine Learning</p> <p>15 Aug 2019</p> <p>Abstract: Sparse communication is a key challenge in training large-scale neural networks. We present SPCL, a high-performance sparse communication library for machine learning. SPCL is designed to be used with distributed training frameworks like PyTorch and TensorFlow. It provides a simple and efficient way to communicate sparse data between GPUs. SPCL is implemented in C++ and is easy to integrate into existing codebases. It achieves a 1.3x performance improvement over state-of-the-art libraries.</p> | <p>Chimera: Efficiently Training Large-Scale Neural Networks with Bidirectional Pipelines</p> <p>15 Aug 2019</p> <p>Abstract: Training large-scale neural networks is a challenging task due to the high communication overhead of data movement. We present Chimera, a training framework that uses bidirectional pipelines to reduce communication overhead. Chimera is designed to be used with distributed training frameworks like PyTorch and TensorFlow. It provides a simple and efficient way to train large-scale neural networks. Chimera is implemented in C++ and is easy to integrate into existing codebases. It achieves a 1.3x performance improvement over state-of-the-art frameworks.</p> | <p>Red Blue Publishing Received: Near Optimal Parallel Matrix-Matrix Multiplication</p> <p>15 Aug 2019</p> <p>Abstract: Matrix-matrix multiplication (MMM) is a core operation in many machine learning applications. We present Red Blue Publishing, a near-optimal parallel MMM implementation. Red Blue Publishing is designed to be used with distributed training frameworks like PyTorch and TensorFlow. It provides a simple and efficient way to perform MMM. Red Blue Publishing is implemented in C++ and is easy to integrate into existing codebases. It achieves a 1.3x performance improvement over state-of-the-art implementations.</p> |
| <p>Demystifying Parallel and Distributed Deep Learning As the Depth Concurrency Analysis</p> <p>15 Aug 2019</p> <p>Abstract: Deep learning is a powerful tool for many applications, but it is often difficult to understand how to parallelize and distribute it. We present a depth concurrency analysis to demystify parallel and distributed deep learning. This analysis shows that the key to efficient parallel and distributed deep learning is to reduce the depth of the computation graph. We provide a simple and efficient way to do this, and show that it achieves a 1.3x performance improvement over state-of-the-art frameworks.</p> | <p>Performance Analysis of 10,000 GPUs and Beyond: From Pipelines to Tensor Processing Units</p> <p>15 Aug 2019</p> <p>Abstract: Training large-scale neural networks on 10,000 GPUs is a challenging task. We present a performance analysis of 10,000 GPUs and beyond, showing that the key to efficient training is to use pipelines and tensor processing units (TPUs). We provide a simple and efficient way to do this, and show that it achieves a 1.3x performance improvement over state-of-the-art frameworks.</p> | <p>Red Blue Publishing Received: Near Optimal Parallel Matrix-Matrix Multiplication</p> <p>15 Aug 2019</p> <p>Abstract: Matrix-matrix multiplication (MMM) is a core operation in many machine learning applications. We present Red Blue Publishing, a near-optimal parallel MMM implementation. Red Blue Publishing is designed to be used with distributed training frameworks like PyTorch and TensorFlow. It provides a simple and efficient way to perform MMM. Red Blue Publishing is implemented in C++ and is easy to integrate into existing codebases. It achieves a 1.3x performance improvement over state-of-the-art implementations.</p> |

Data Movement Is All You Need: A Case Study on Optimizing Transformers (arXiv:2007.00072)



| Operator class | % flop | % Runtime |
|---------------------------|--------|-----------|
| Tensor contraction | 99.80 | 61.0 |
| Statistical normalization | 0.17 | 25.5 |
| Element-wise | 0.03 | 13.5 |
| | 0.2% | 39% |

highly optimized

Our performance improvement for BERT-large

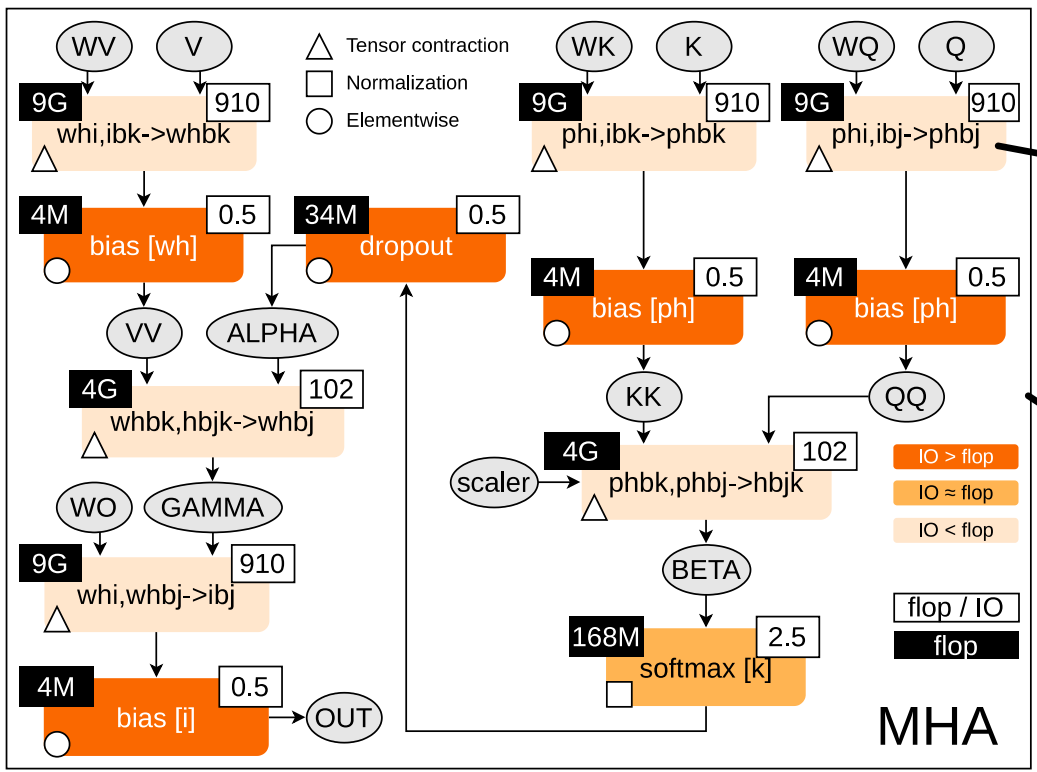
- 30% over PyTorch
- 20% over Tensorflow + XLA
- 8% over DeepSpeed

est. savings on AWS over PyTorch:
\$85k for BERT, \$3.6M GPT-3

OpenAI booth at NeurIPS 2019 in Vancouver, Canada
Image Credit: Khari Johnson / VentureBeat

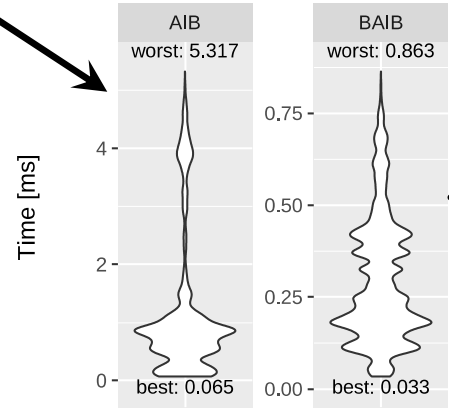
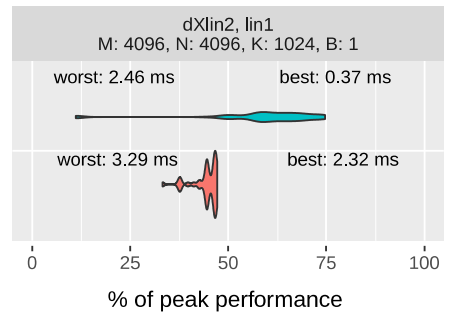
Last week, OpenAI published a paper detailing GPT-3, a machine learning model that achieves strong results on a number of natural language benchmarks. A 175 billion parameters where a parameter affects data's prominence in an overall prediction, it's the largest of its kind. And with a memory size exceeding 350GB, it's one of the priciest, costing an estimated \$12 million to train.

Data Movement Is All You Need: A Case Study on Optimizing Transformers (arXiv:2007.00072)

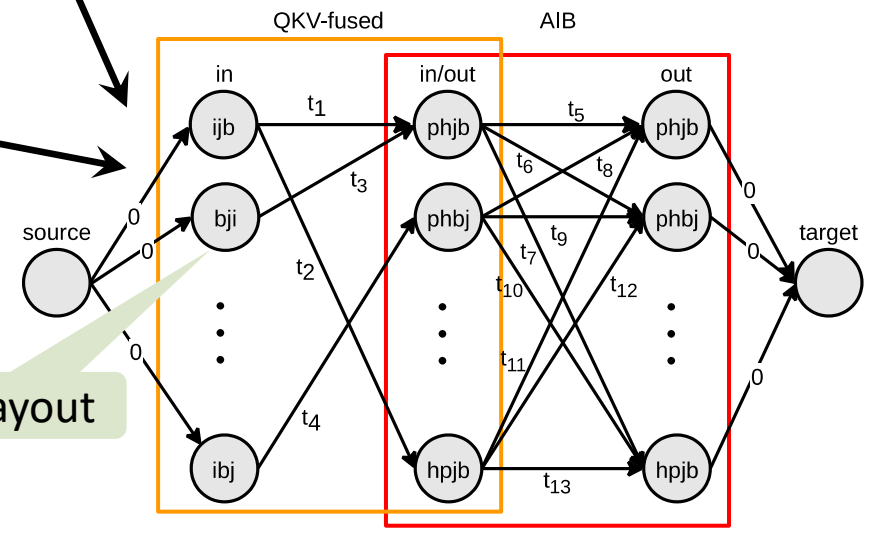


different data layouts

different fusion strategies



Configuration selection graph



data layout

fusion strategy

Full BERT encoder layer performance (ms)

| | TF+XLA | PyTorch | DeepSpeed | Ours |
|----------|--------|---------|-----------|-------------|
| Forward | 3.2 | 3.45 | 2.8 | 2.63 |
| Backward | 5.2 | 5.69 | 4.8 | 4.38 |

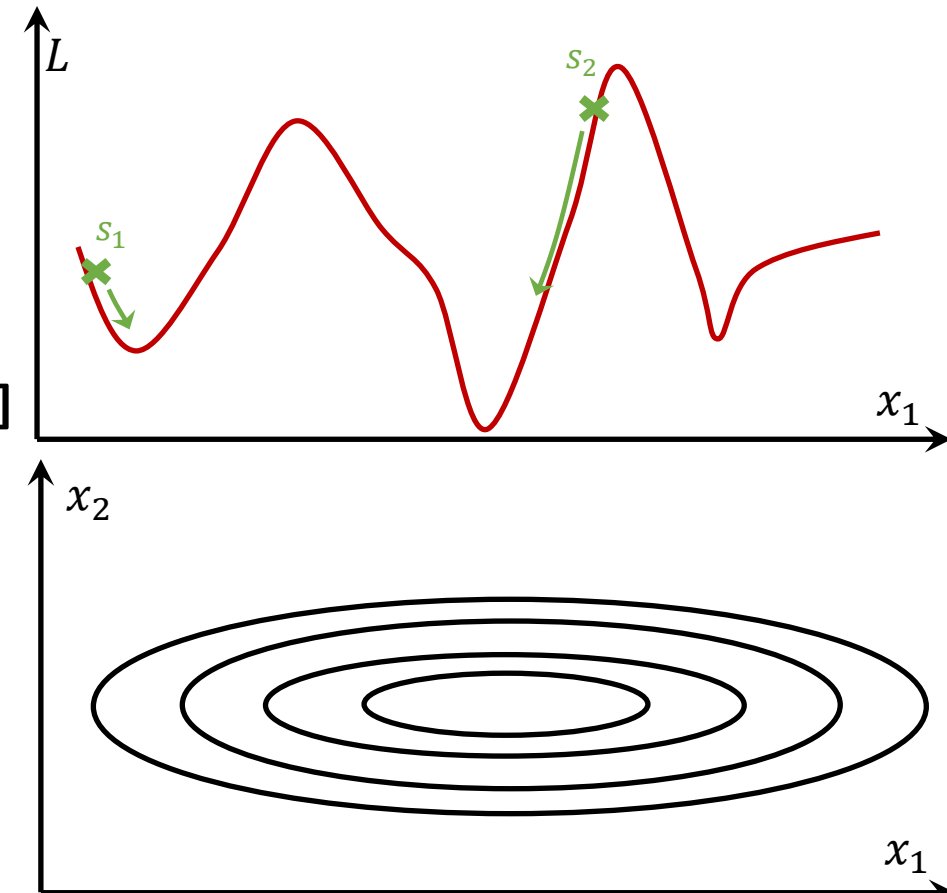
Moving Data is Most Expensive!

Techniques to Shrink ML Data

Quantization – Running Gigantic LLMs on Reasonable Systems (arXiv:2210.17323)

- **Brains have limited precision! Why are we computing with FP32?**
 - For technical reasons (SGD, optimization, how we quantize)
 - Neurons in Hippocampus can “reliably distinguish 24 strengths” [1]
4.6 bits of information!
- **GPT-3 has up to 175 billion parameters**
 - 700 GiB in FP32, 350 GiB in FP16/BF16 ☹️
 - Rounding to <5 bits is not so simple
 - Requires some foundation and many tricks
- **Consider “error landscape” of a trained model with weights w [2]**

$$\partial E = \underbrace{\left(\frac{\partial E}{\partial w}\right)^T}_{\text{Gradient } (\approx 0)} \partial w + \underbrace{\frac{1}{2} \partial w^T \left(\frac{\partial^2 E}{\partial^2 w}\right) \partial w}_{\text{“Curvature” of error (aka. “sensitivity”)}} + \underbrace{O(|\partial w|^3)}_{\text{Higher-order terms (=0 for quadratic loss)}}$$



[1] Bartol et al., “Hippocampal Spine Head Sizes Are Highly Precise”, eLife 2015

[2] LeCun, Denker, Solla: “Optimal Brain Damage”, NIPS’90

Quantization – Running Gigantic LLMs on Reasonable Systems (arXiv:2210.17323)

- Quantization objective for low precision rounded weights \hat{w}

$$\operatorname{argmin}_{\hat{w}} \|wx - \hat{w}x\|^2$$
- Solve PTQ optimization problem row by row of w
 - Round row and push the error forward using the inverse Hessian
 - Update Hessian for each column
- Tricks
 - Block updates for better locality (10x speedup)
 - Use Cholesky to invert Hessian (higher stability)
 - Work one transformer block at a time (6 operators fit in memory)
 - Use quantized input from previous blocks for block i
- Results
 - Generative inference 2-4x faster
 - 3 bits \rightarrow 66 GiB, fits in a single (high-end) A100 GPU!

GPTQ: ACCURATE POST-TRAINING QUANTIZATION FOR GENERATIVE PRE-TRAINED TRANSFORMERS

A PREPRINT

Elias Frantar*
IST Austria
Klosterneuburg, Austria
elias.frantar@ist.ac.at

Saleh Ashkboos
ETH Zurich
Switzerland
saleh.ashkboos@inf.ethz.ch

Torsten Hoefler
ETH Zurich
Switzerland
htor@inf.ethz.ch

Dan Alistarh
IST Austria & Neural Magic, Inc.
Klosterneuburg, Austria
dan.alistarh@ist.ac.at

ABSTRACT

Generative Pre-trained Transformer (GPT) models set themselves apart through breakthrough performance across complex language modelling tasks, but also by their extremely high computational and storage costs. Specifically, due to their massive size, even inference for large, highly-accurate GPT models may require multiple performant GPUs to execute, which limits the usability of such models. While there is emerging work on relieving this pressure via model compression, the applicability and performance of existing compression techniques is limited by the scale and complexity of GPT models. In this paper, we address this challenge, and propose GPTQ, a new one-shot weight quantization method based on approximate second-order information, that is both highly-accurate and highly-efficient. Specifically, GPTQ can quantize GPT models with 175 billion parameters in approximately four GPU hours, reducing the bitwidth down to 3 or 4 bits per weight.

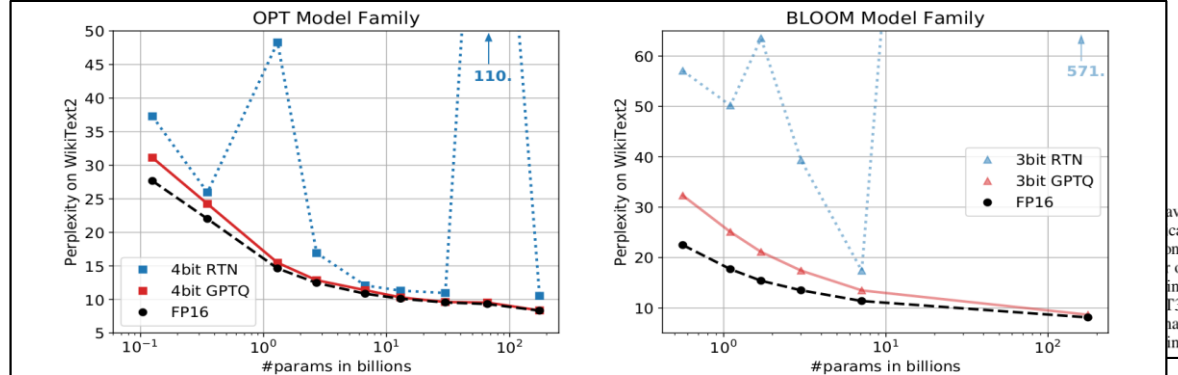


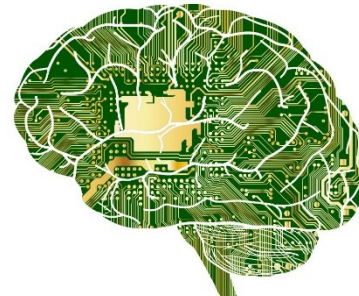
Figure 1: Quantizing OPT models to 4 and BLOOM models to 3 bit precision, comparing GPTQ with the FP16 baseline and round-to-nearest (RTN) [34, 5].

| Model | FP16 | 1024 | 512 | 256 | 128 | 64 | 32 | 3-bit |
|----------|------|-------|-------|-------|------|------|------|-------|
| OPT-175B | 8.34 | 11.84 | 10.85 | 10.00 | 9.58 | 9.18 | 8.94 | 8.68 |
| BLOOM | 8.11 | 11.80 | 10.84 | 10.13 | 9.55 | 9.17 | 8.83 | 8.64 |

Table 6: 2-bit GPTQ quantization results with varying group-sizes; perplexity on WikiText2.

Quantization Reduces Data by an Order of Magnitude

How to Go Further?



Model Sparsification ... (arXiv:2102.00554)

- Brains are not densely connected! Why are DNN computations dense?
 - For technical reasons (training, implementation etc.)
 - We may want to shift towards sparse!

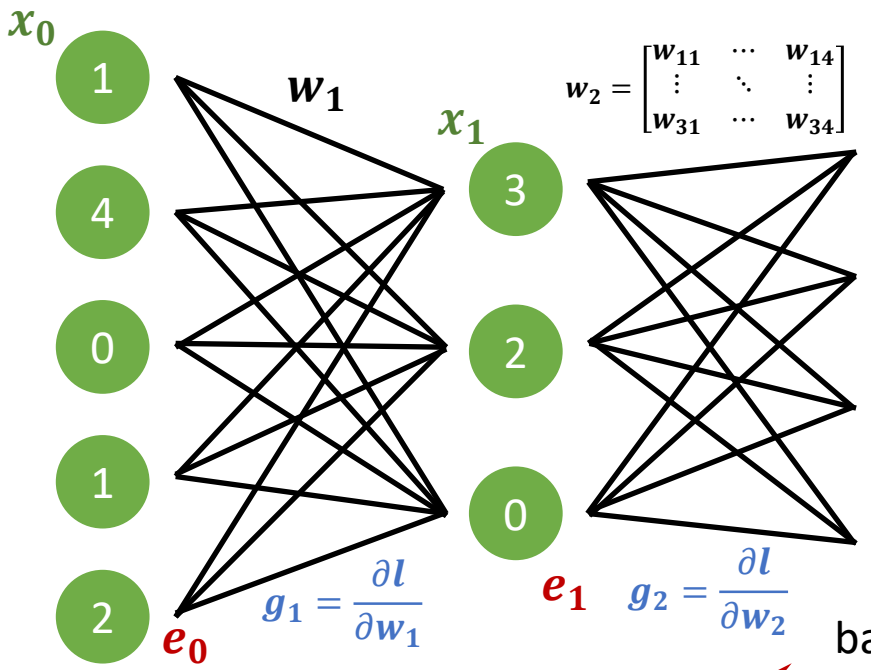
Intuition: not **all** features are **always** relevant!

- Represent as (sparse) vector space
- ✓ Less overfitting
- ✓ Interpretability
- ✓ Parsimony

the $f_{t_{re}}$ $w_{l_{b}}$ sp_{rs}

Key results:

- 95% sparse ResNet-52, BERT, or GPT models
- Essentially same quality
- Up to 20x cheaper!



Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks

TORSTEN HOEFLER, ETH Zürich, Switzerland
 DAN ALISTARH, IST Austria, Austria
 TAL BEN-NUN, ETH Zürich, Switzerland
 NIKOLI DRYDEN, ETH Zürich, Switzerland
 ALEXANDRA PESTE, IST Austria, Austria

The growth of neural networks memory for networks. of sparsification neural net practice. who wish include th as early s technique that could sparsity c

the size parts, sparse reduce the er growing ive tutorial lements of sparsity in actioners forward. We mena such, and show r efficiency ng on how

The su possible

Sparsity in Deep Learning: Pruning + growth for efficient inference and training in neural networks

Scalable Parallel Computing Lab @ ETH Zu...

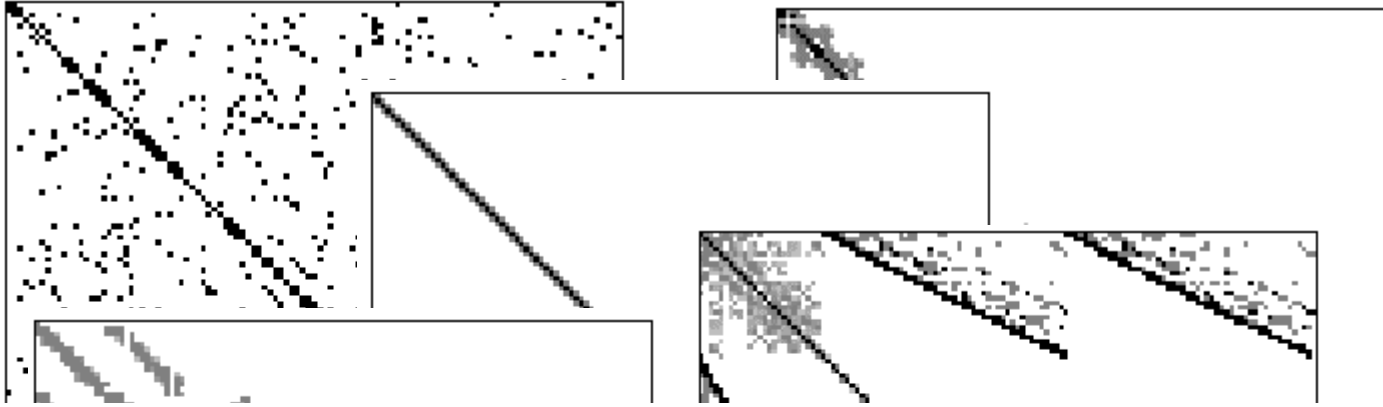
7.1K views 1 year ago SPCL Lab talks Torsten Hoefler presents an overview of sparsity in deep learning. Check the markers for various parts of the talk.

1 INTRODUCTION

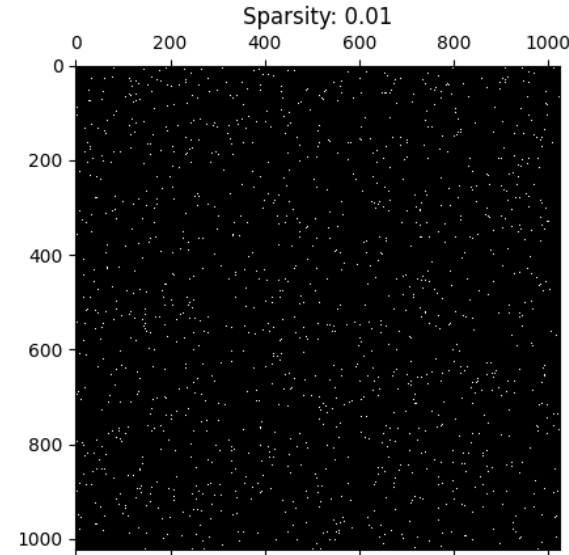
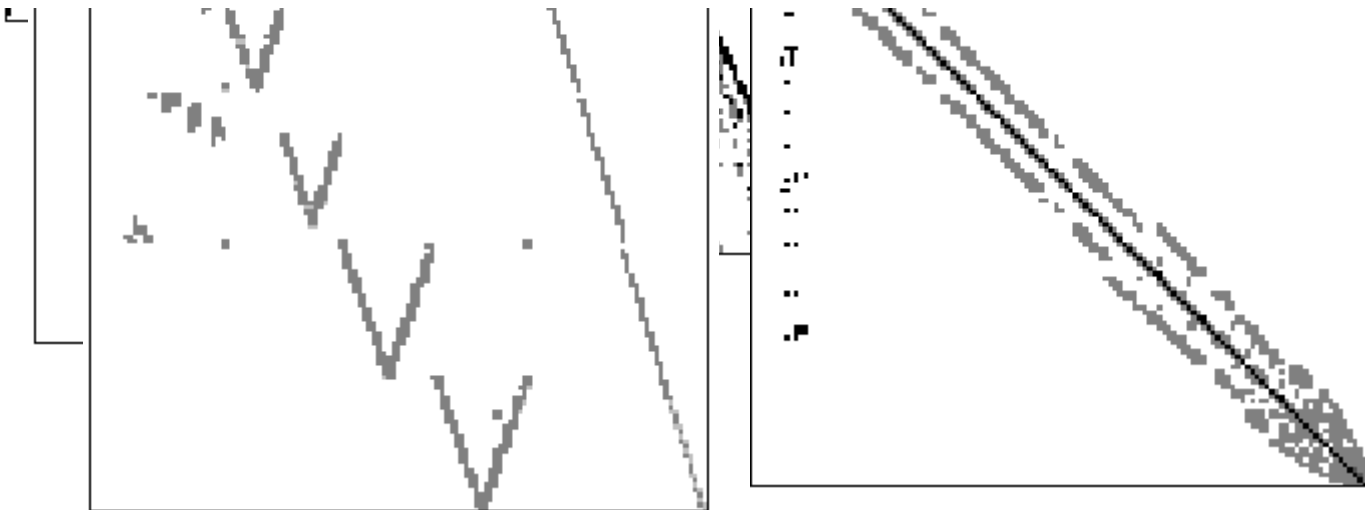
Deep learning shows unparalleled promise for solving very complex real-world problems in areas such as computer vision, natural language processing, knowledge representation, recommendation systems, drug discovery, and many more. With this development, the field of machine learning is moving from traditional feature engineering to neural architecture engineering. However, still

arXiv:2102.00554v1 [cs.LG] 31 Jan 2021

Sparse ML Computations – Very Different from Scientific Computing!

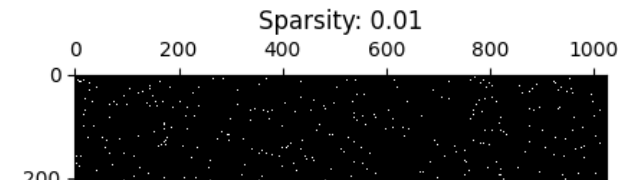


Sparse Matrices from Scientific Computing are quite structured!



WK

Sparsified BERT
WK and WQ matrices
(3rd encoder)



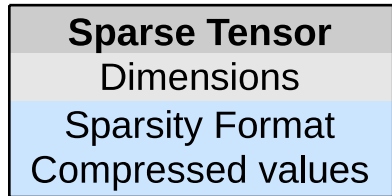
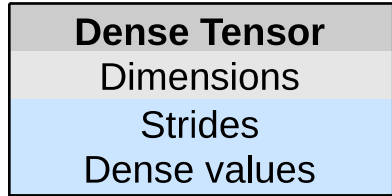
WQ

Sparse Matrices in Deep Learning are quite uniform(ly random)!

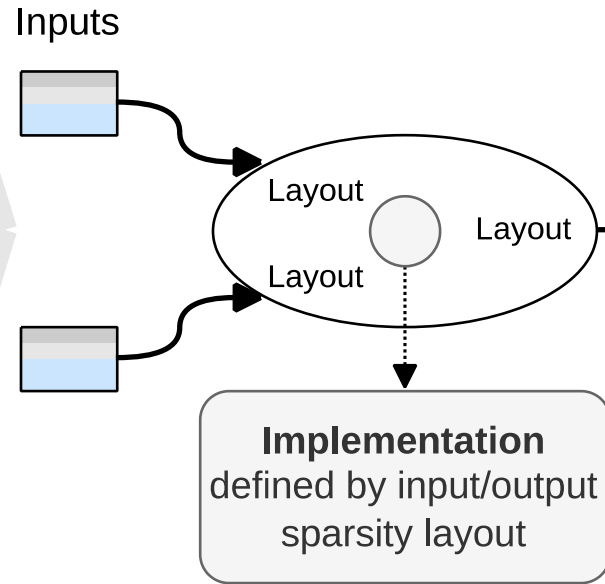
Programming Sparse Models – Meet PyTorch Sten (arXiv:2304.07613)



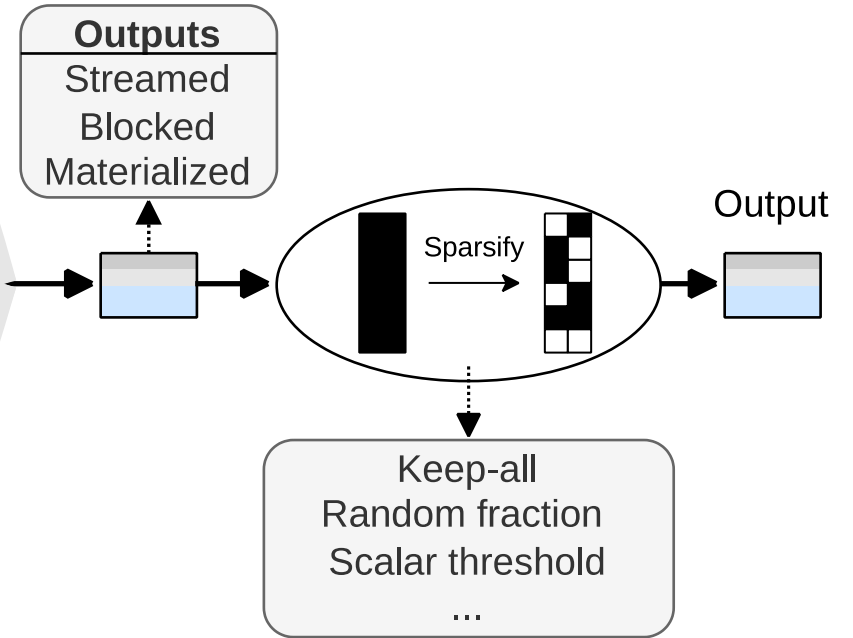
Sparsity Layouts



Operators



Sparsifiers



Selected Available Sparsifiers:

Keep all

do not drop

Random fraction

drop if rand < 0.5

Scalar threshold

drop if value < 0

Streaming

Per block fraction

Find block quantile q

Drop if below

Blocked

Scalar fraction

Find quantile q

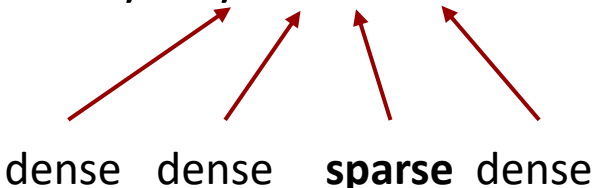
Drop if below

Materializing

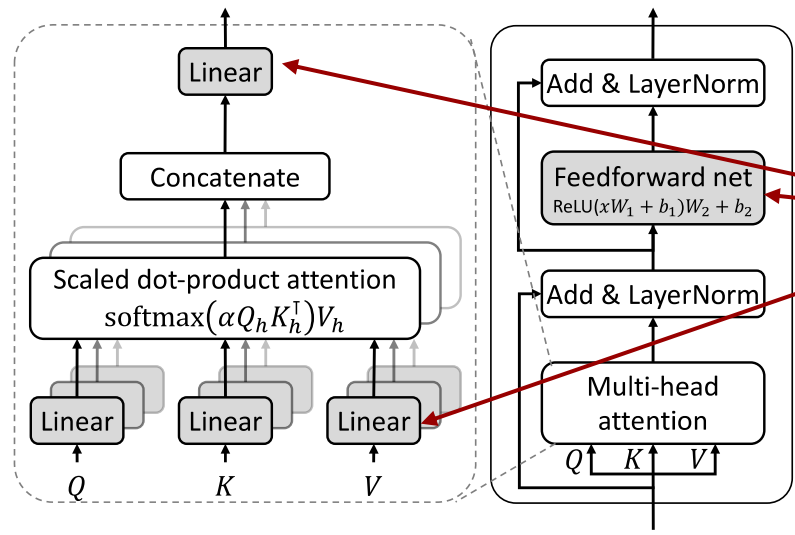
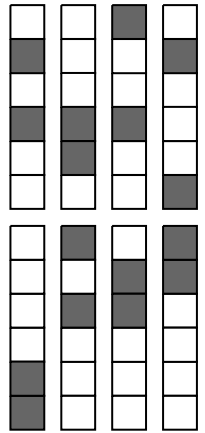
Sten Performance

Custom implementation of matrix multiplication:
sparse @ dense -> dense

Linear layer: $y = xW + b$



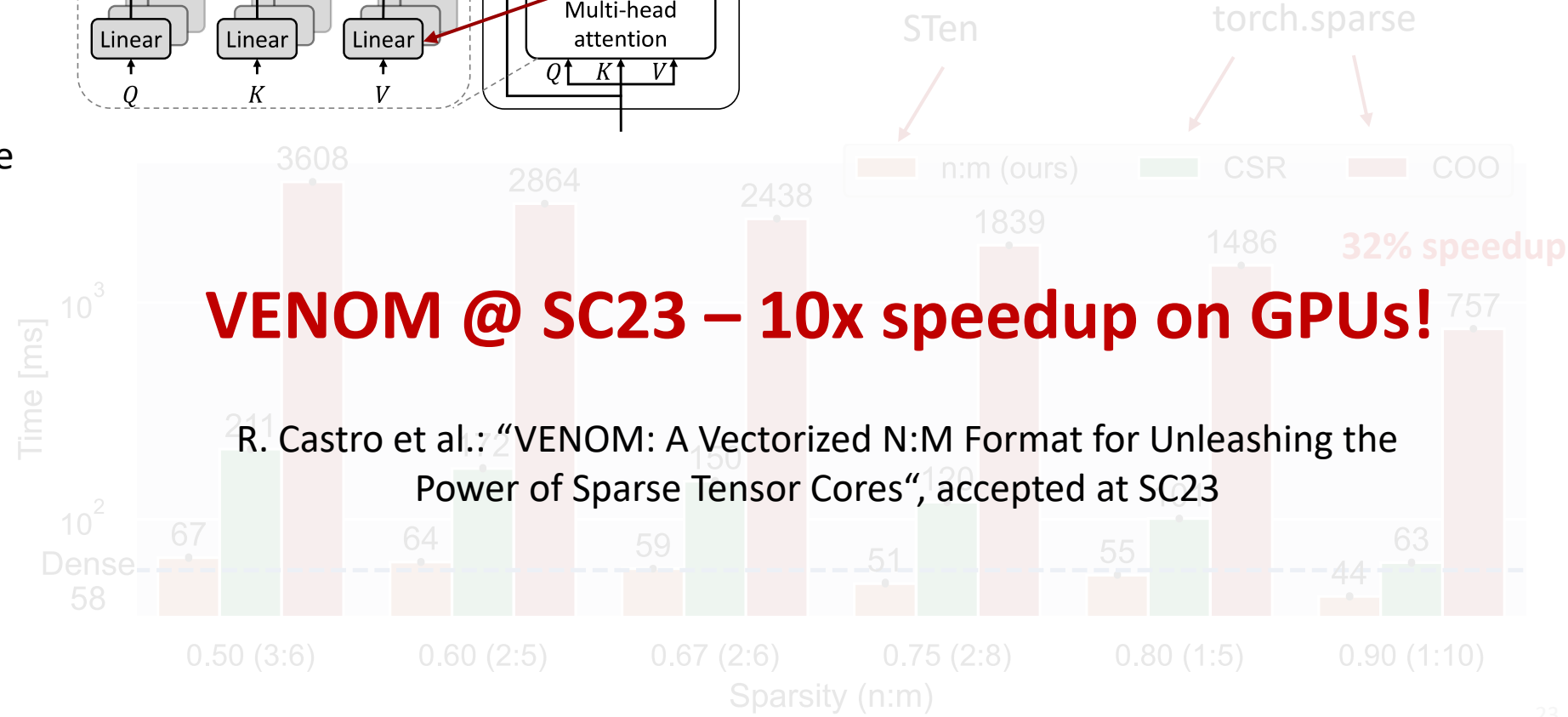
2:6 sparse format



BERT (base) from HuggingFace

- batch size 8
- sequence length 128

Sparsified linear layer weights
Intel i7-4770 CPU

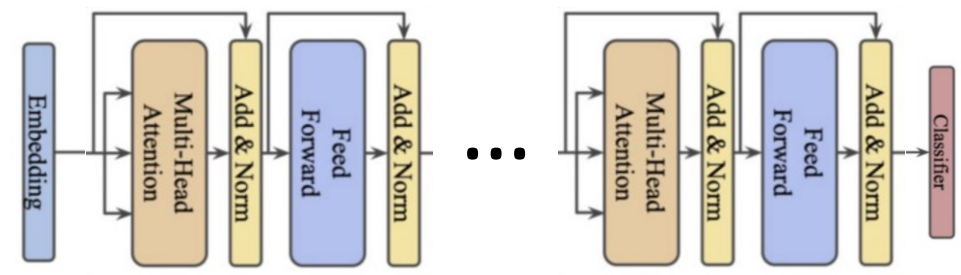


**Model Compression Enables
More Efficient Processing**

Which Makes Data Movement Even More Important!

Especially in the Network!

Three Systems Dimensions in Large-scale Super-learning ...



High-Performance I/O

- Quickly growing data volumes
 - Scientific computing!
- Use the specifics of machine learning workloads
 - E.g., intelligent prefetching

21 Jan 2021

CLAIRVOYANT PREFETCHING FOR DISTRIBUTED MACHINE LEARNING I/O

Roman Böhringer¹ Nikoli Dryden¹ Tal Ben-Nun¹ Torsten Hoefler¹

ABSTRACT

I/O is emerging as a major bottleneck for machine learning training, especially in distributed environments such as clouds and supercomputers. Optimal data ingestion pipelines differ between systems, and increasing efficiency requires a delicate balance between access to local storage, external filesystems, and remote workers; yet existing frameworks fail to efficiently utilize such resources. We observe that, given the seed generating the random access pattern for training with SGD, we have *clairvoyance* and can exactly predict when a given sample will be accessed. We combine this with a theoretical analysis of access patterns in training and performance modeling to produce a novel machine learning I/O middleware, HDMLP, to tackle the I/O bottleneck. HDMLP provides an easy-to-use, flexible, and scalable solution that delivers better performance than state-of-the-art approaches while requiring very few changes to existing codebases and supporting a broad range of environments.

High-Performance Compute

- Deep learning is HPC
 - Data movement!
- Quantization, Sparsification
 - Drives modern accelerators!

Data Movement Is All You Need: A Case Study on Optimizing Transformers

Andrei Ivanov*, Nikoli Dryden*, Tal Ben-Nun, Shigang Li, Torsten Hoefler
ETH Zürich
firstname.lastname@inf.ethz.ch
* Equal contribution

Abstract—Transformers have become widely used for language modeling and sequence learning tasks, and are one of the most important machine learning workloads today. Training one is a very compute-intensive task, often taking days or weeks, and significant attention has been given to optimizing transformers. Despite this, existing implementations do not efficiently utilize GPUs. We find that data movement is the key bottleneck when training. Due to Amdahl's Law and massive improvements in compute performance, training has now become memory-bound. Further, existing frameworks use suboptimal data layouts. Using these insights, we present a recipe for globally optimizing data movement in transformers. We reduce data movement by up to 22.91% and overall achieve a 1.30x performance improvement over state-of-the-art frameworks when training BERT. Our approach is applicable more broadly to optimizing deep neural networks, and offers insight into how to tackle emerging performance bottlenecks.

challenges such as artificial general intelligence [27]. Thus, improving transformer performance has been in the focus of numerous research and industrial groups.

Significant attention has been given to optimizing transformers: local and fixed-window attention [28]–[32], more general structured sparsity [33], learned sparsity [34]–[36], and other algorithmic techniques [19], [37] improve the performance of transformers. Major hardware efforts, such as Tensor Cores and TPUs [38] have accelerated tensor operations like matrix-matrix multiplication (MMM), a core transformer operation. Despite this, existing implementations do not efficiently utilize GPUs. Even optimized implementations such as Megatron [18] report achieving only 30% of peak GPU flops.

We find that the key bottleneck when training transform-

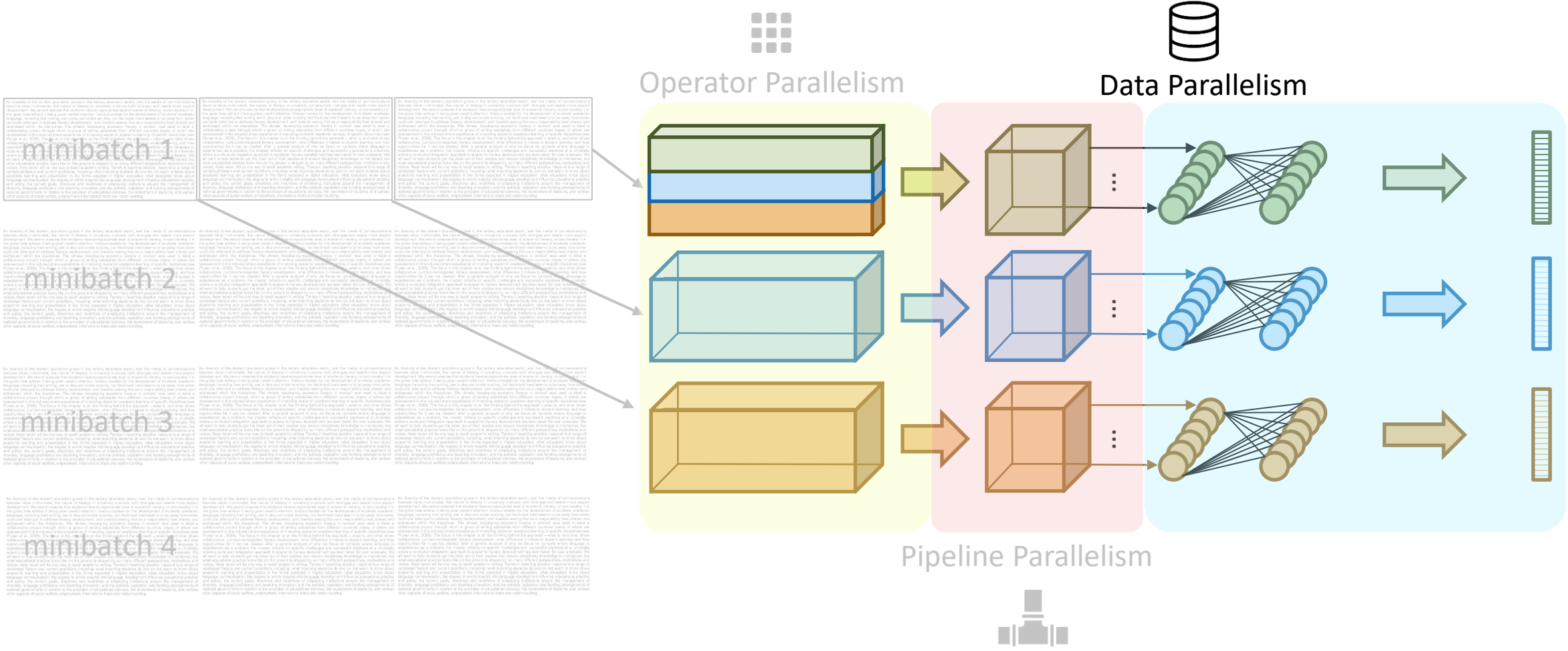
High-Performance Communication

- Use larger clusters (10k+ GPUs)
- Model parallelism
 - Complex pipeline schemes
- Optimized networks

Distribution and Parallelism

| Data | Pipeline | Operator |
|---|--|--|
| <p>SPCLML: High-Performance Sparse Communication for Machine Learning</p> <p>Yuan Zhang, Nikoli Dryden, Tal Ben-Nun, Shigang Li, Torsten Hoefler</p> <p>Abstract—Sparse communication is a key challenge in distributed machine learning. Existing frameworks often suffer from high communication overheads, which can significantly impact training performance. We propose SPCLML, a high-performance sparse communication framework for machine learning. SPCLML is designed to be efficient and scalable, and can be used to accelerate training on large clusters. We demonstrate that SPCLML achieves a significant performance improvement over existing frameworks, and is particularly well-suited for training on large clusters.</p> | <p>Chimera: Efficiently Training Large-Scale Neural Networks with Bidirectional Pipelines</p> <p>Yuan Zhang, Nikoli Dryden, Tal Ben-Nun, Shigang Li, Torsten Hoefler</p> <p>Abstract—Training large-scale neural networks is a challenging task due to the high communication overheads of data movement. We propose Chimera, a framework for training large-scale neural networks with bidirectional pipelines. Chimera is designed to be efficient and scalable, and can be used to accelerate training on large clusters. We demonstrate that Chimera achieves a significant performance improvement over existing frameworks, and is particularly well-suited for training on large clusters.</p> | <p>Real-Time Publishing Revisited: Near-Optimal Parallel Matrix-Matrix Multiplication</p> <p>Yuan Zhang, Nikoli Dryden, Tal Ben-Nun, Shigang Li, Torsten Hoefler</p> <p>Abstract—Real-time publishing is a challenging task due to the high communication overheads of data movement. We propose a near-optimal parallel matrix-matrix multiplication algorithm. This algorithm is designed to be efficient and scalable, and can be used to accelerate training on large clusters. We demonstrate that this algorithm achieves a significant performance improvement over existing frameworks, and is particularly well-suited for training on large clusters.</p> |

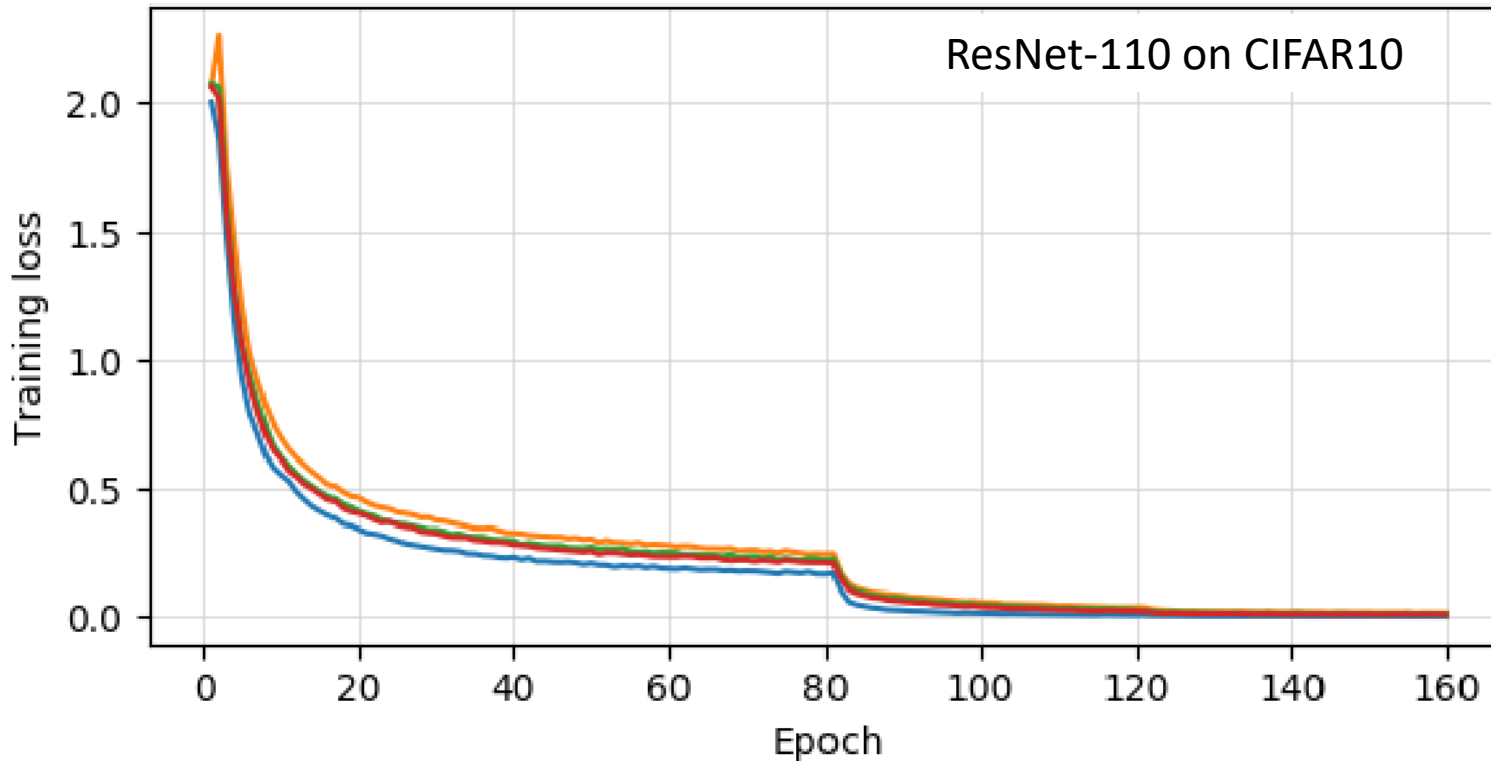
The Three Dimensions of Parallelism in Deep Learning (arXiv:1802.09941)



Data-parallel Gradient Sparsification – Top-k SGD (arXiv:1809.10505)



- Turns out 90-99.9% of the smallest gradient values can be skipped in the summation – at similar accuracy
 - Accumulate the skipped values locally (convergence proof, similar to async. SGD with implicit staleness bounds [1])



Assumptions

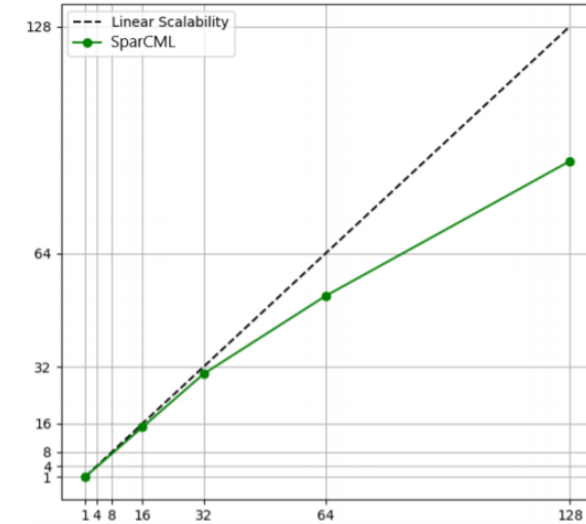
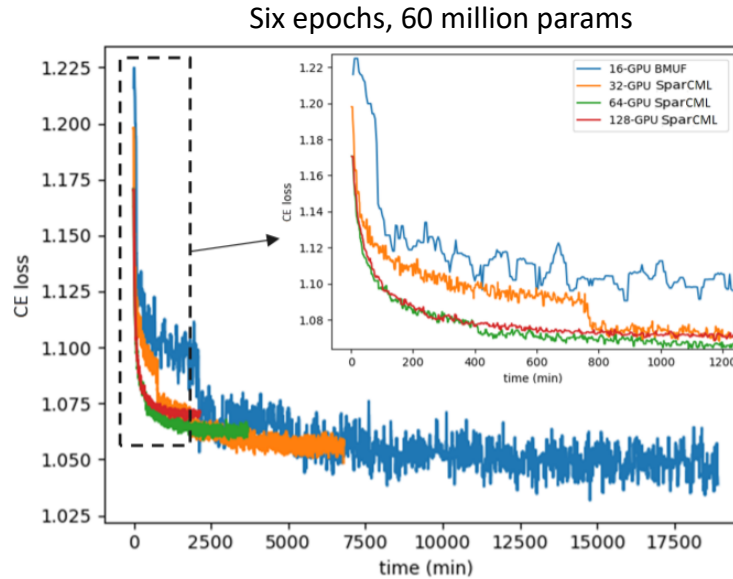
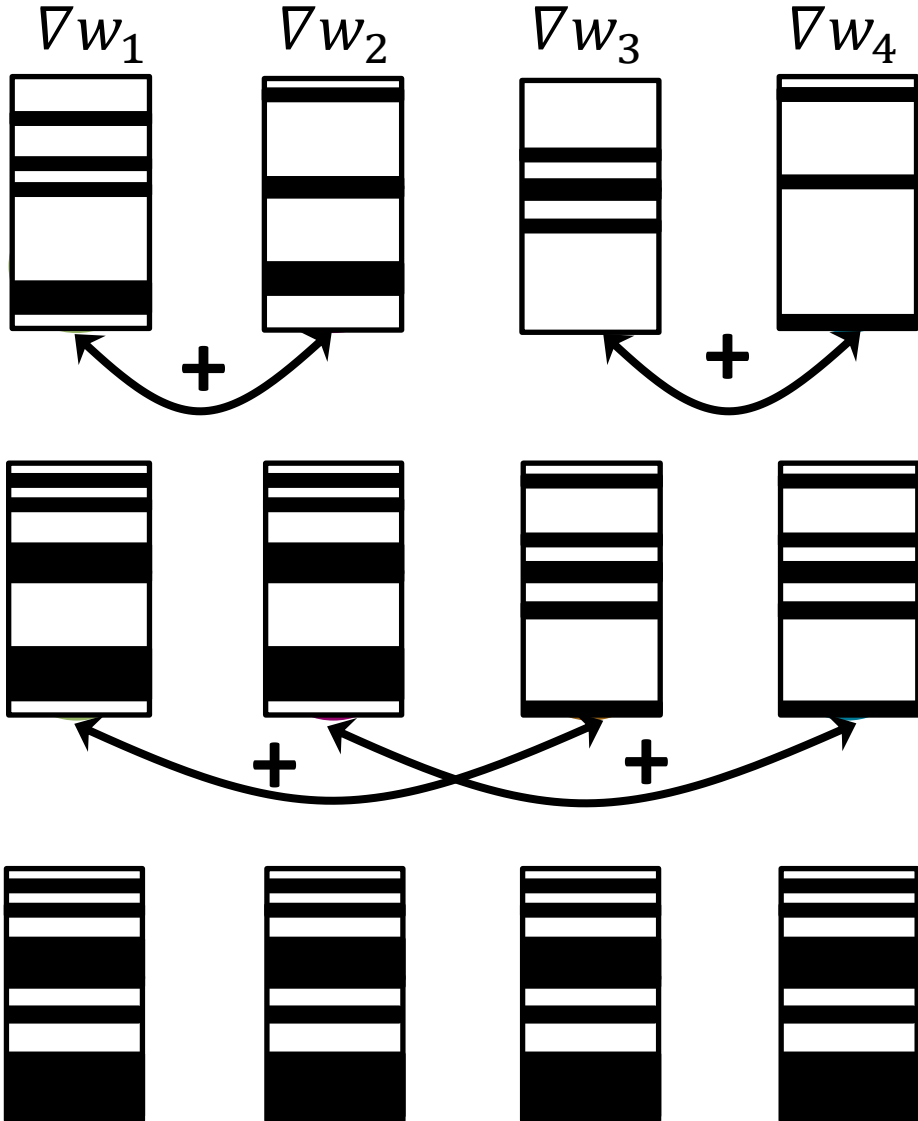
Discuss Section

we:

tasks in

— Baseline — TopK [K=0.025%] — TopK [K=0.1%] — TopK [K=0.2%]

SparCML – Sparse Allreduce for Decentral Updates (arXiv:1802.08021)



Microsoft Speech Production Workload Results – **2 weeks** → **2 days!**

| System | Dataset | Model | # of nodes | Algorithm | Speedup |
|------------------|----------|---------|------------|----------------------|------------------------------|
| Piz Daint | ImageNet | VGG19 | 8 | Q4 | 1.55 (3.31) |
| Piz Daint | ImageNet | AlexNet | 16 | Q4 | 1.30 (1.36) |
| Piz Daint EC2 | MNIST | MLP | 8 | Top16_Q4 Top16_Q4 | 3.65 (4.53) 19.12 (22.97) |

Sparse Allreduce – A Headache for Systems Work



Near-Optimal Sparse Allreduce for Distributed Deep Learning

Shigang Li
shigang.li@inf.ethz.ch
Department of Computer Science, ETH Zurich
Switzerland

Torsten Hoefer
htor@inf.ethz.ch
Department of Computer Science, ETH Zurich
Switzerland

Abstract
Communication overhead is one of the major obstacles to train large deep learning models at scale. Gradient sparsification is a promising technique to reduce the communication volume. However, it is very challenging to obtain real performance improvement because of (1) the difficulty of achieving an scalable and efficient sparse allreduce algorithm and (2) the sparsification overhead. This paper proposes Ok-Topk, a scheme for distributed training with sparse gradients. Ok-Topk integrates a novel sparse allreduce algorithm that has a 6k communication volume which is asymptotically smaller than with the decentralized parallel gradient descent (SGD) optimizer, and a top-k selection algorithm. To reduce the sparsification overhead, Ok-Topk efficiently selects the top-k gradient values according to an estimate of the gradient. Evaluations are conducted on the Piz Datanode cluster with neural network models for image classification learning domains. Empirical results show that Ok-Topk achieves similar model accuracy to the state-of-the-art sparse allreduces, but is more scalable and significantly improves training throughput (e.g., 3.29x-12.95x improvement for BERT on 256 GPUs).

CCS Concepts: • Theory of computation → Parallel algorithms; • Computing methodologies → Neural networks.

Keywords: distributed deep learning, allreduce, gradient sparsification, data parallelism

introducing up to 99.9% zero values without significant loss of accuracy. Only the nonzero values of the distributed gradients are accumulated across all processes. See [22] for an overview of gradient and other sparsification approaches in deep learning.
However, sparse reduction still suffers from scalability issues. Specifically, the communication volume of the existing sparse allreduce algorithms grows with the number of processes. For example, its communication volume is proportional to P^2 , which eventually makes allreduce as P increases. Other algorithms [36] suffer from scalability issues due to dense representations on the fly. For example, let us assume the model has 1 million weights and it is 99% sparse at each node—thus, each node contributes its 10,000 largest gradient values and their indexes to the calculation. Let us now assume that the computation is distributed across 128 data-parallel nodes and the reduction uses a dissemination algorithm [20, 28] with 7 stages. In stage one, each process communicates its 10,000 values to be summed up. Each process now enters the next stage with up to 20,000 values. Those again are summed up leading to up to 40,000 values in stage 3 (if the value indexes do not overlap). The number of values grows exponentially until the algorithm converges after 7 stages with 640,000 values (nearly dense!). Even with overlapping indexes, the fill-in will quickly diminish the benefits of gradient sparsity in practice and lead to large and substantial communication volumes [26].

Minimize fill-in and communication volume!

Flare: Flexible In-Network Allreduce

Daniele De Sensi
daniele.desensi@inf.ethz.ch
ETH Zurich
Zurich, Switzerland

Salvatore Di Girolamo
salvatore.digirolamo@inf.ethz.ch
ETH Zurich
Zurich, Switzerland

Saleh Ashkboos
saleh.ashkboos@inf.ethz.ch
ETH Zurich
Zurich, Switzerland

Shigang Li
shigang.li@inf.ethz.ch
ETH Zurich
Zurich, Switzerland

Torsten Hoefer
torsten.hoefer@inf.ethz.ch
ETH Zurich
Zurich, Switzerland

ABSTRACT
The allreduce operation is one of the most commonly used communication routines in distributed applications. To improve its bandwidth and to reduce network traffic, this operation can be accelerated by offloading it to network switches that aggregate the data received from the hosts, and send them back the aggregated result. However, existing solutions provide limited opportunities and might provide a poor user experience when dealing with custom operations, such as in-network data, or when reproducing the results of the computation. To deal with these problems, in this paper we design a flexible programming model for network switches, called sPIN, and analyze different algorithms for aggregation on this architecture, showing significant improvements compared to state-of-the-art approaches.

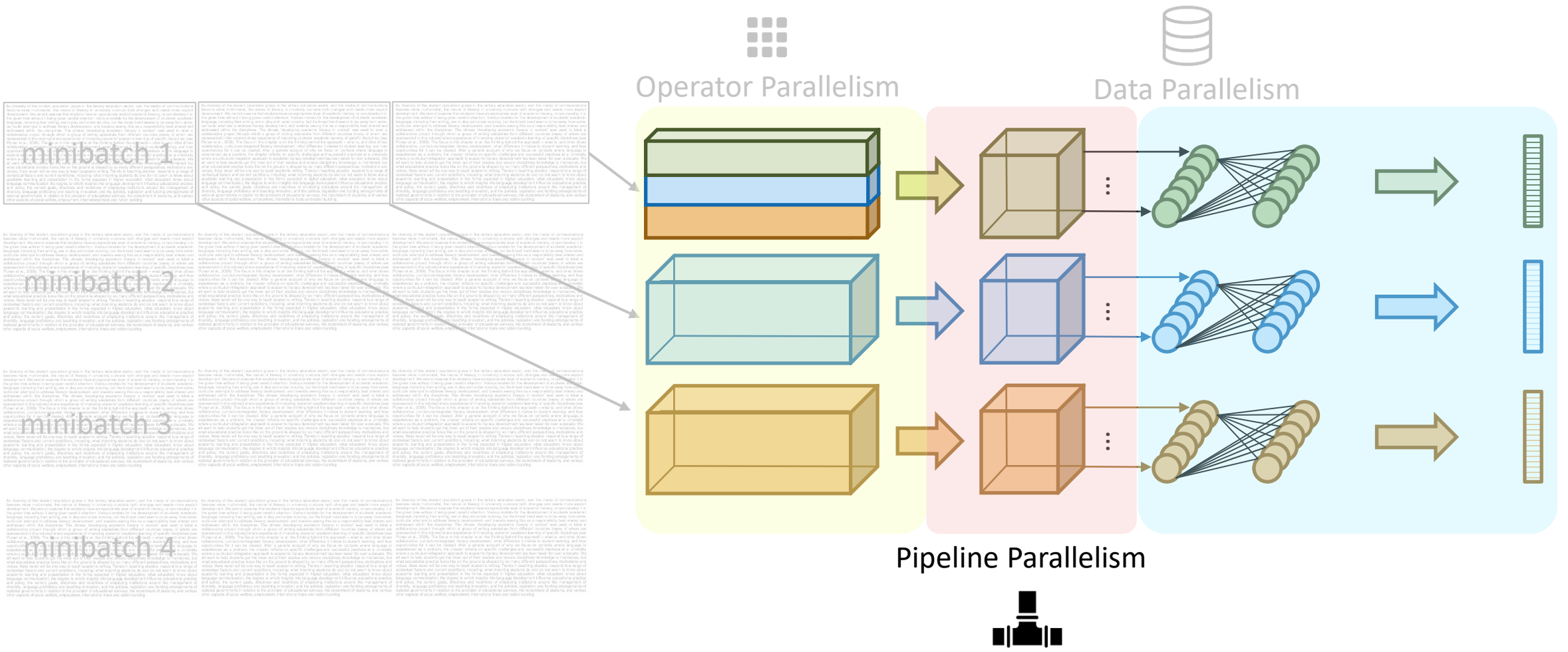
CCS CONCEPTS
• Networks → In-network processing; • Hardware → Networking hardware; • Computer systems organization → Distributed architectures.

KEYWORDS
In-Network Computing; Programmable Switch; Allreduce
ACM Reference Format:
Daniele De Sensi, Salvatore Di Girolamo, Saleh Ashkboos, Shigang Li, and Torsten Hoefer. 2018. Flare: Flexible In-Network Allreduce. In *Supercomputing '21: The International Conference for High Performance Computing, Networking, Storage, and Analysis*, Nov 14–19, 2021, St. Louis, MO. ACM, New

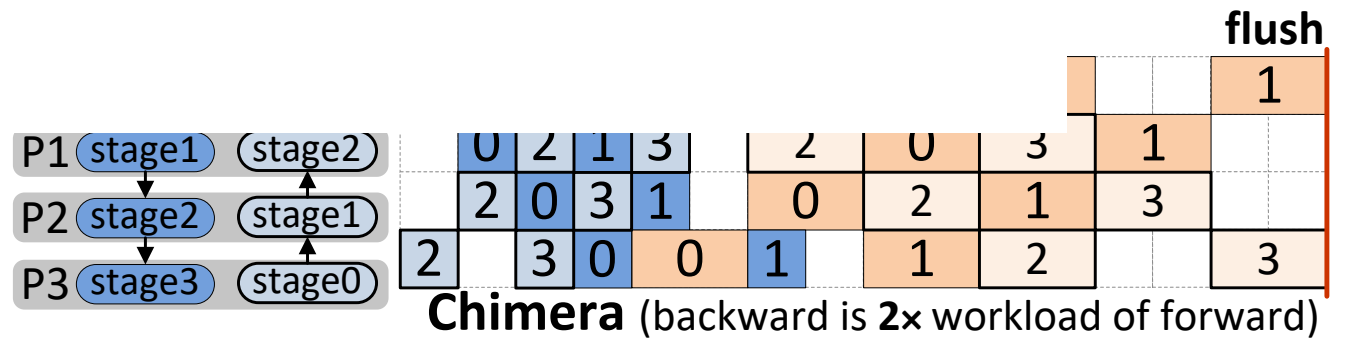
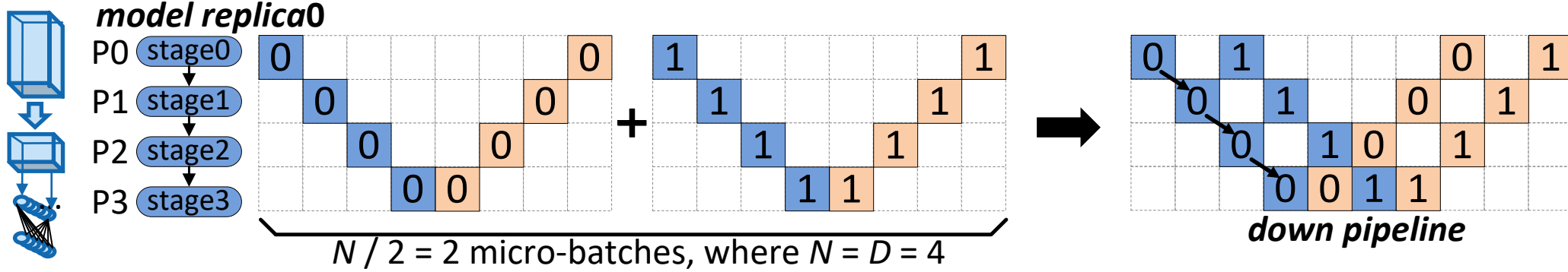
others, and recent studies [2] show that MPI_Allreduce is the most significant collective communication operation in terms of time.
The simplest but most commonly used allreduce algorithm is the Rabenseifner's algorithm [3]. This algorithm is divided into two phases: a scatter phase and an allgather phase. In the scatter phase, each host sends its data to the switch. In the allgather phase, each host receives the aggregated data from the switch. The amount of data sent by each host is then $2(P-1) \frac{Z}{P} \approx 2Z$. In-network allreduce, however, allows hosts to exploit in-network compute, i.e., they can perform the allreduce operation on the switches in the network.
To outline the advantages of performing an in-network allreduce, we describe the general idea underlying most existing in-network reduction approaches [9–11]. We first suppose to have the P hosts connected through a single switch. Each of the hosts sends its data to the switch, that aggregates together the vectors coming from all the hosts, and then sends them back the aggregated vector. Differently from the host-based optimal allreduce, in the in-network allreduce each host only sends Z elements, thus leading to a 2x reduction in the amount of transmitted data. If the switches can aggregate the received data at line rate, this leads to a 2x bandwidth improvement compared to a host-based allreduce. Besides improvements in the bandwidth, in-network allreduce also reduces the network traffic. Because the interconnection network consumes a large fraction of the overall system power (from 15% to 50% depending on the system load [12]), any reduction in the network traffic would also help in reducing the power consumption and thus the running cost of the system.

Making in-network computation work!

The Three Dimensions of Parallelism in Deep Learning (arXiv:1802.09941)

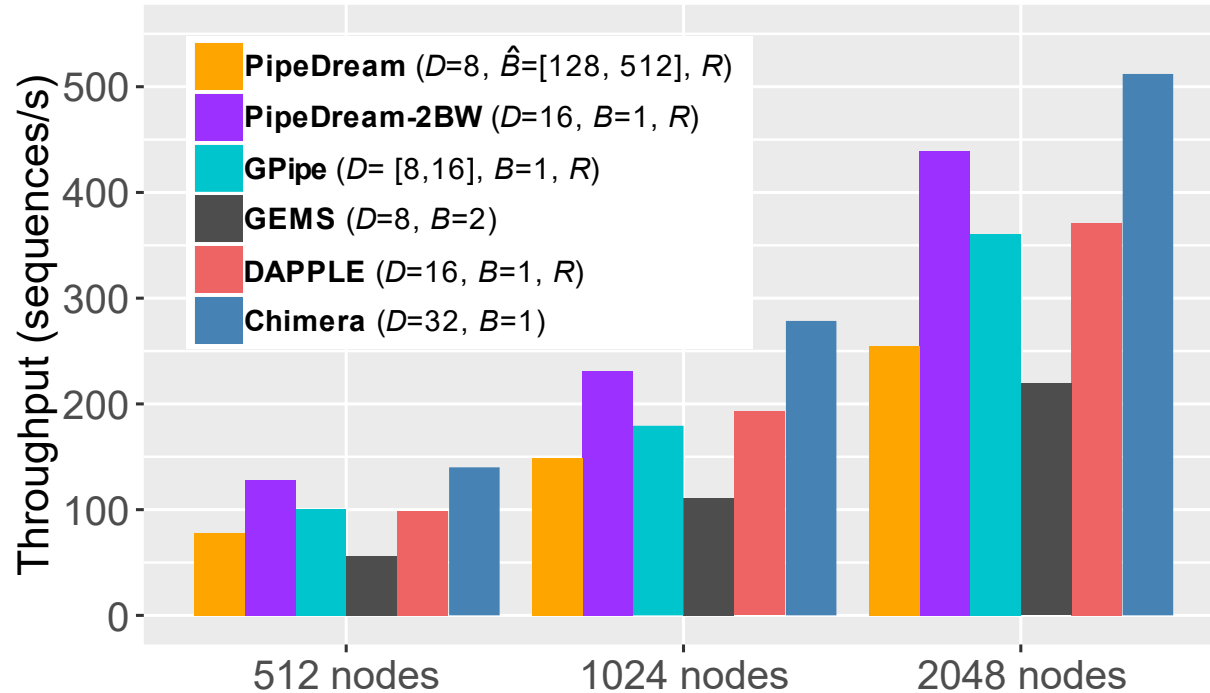


Bidirectional Pipelines – Meet Chimera (arXiv: 2107.06925v3)





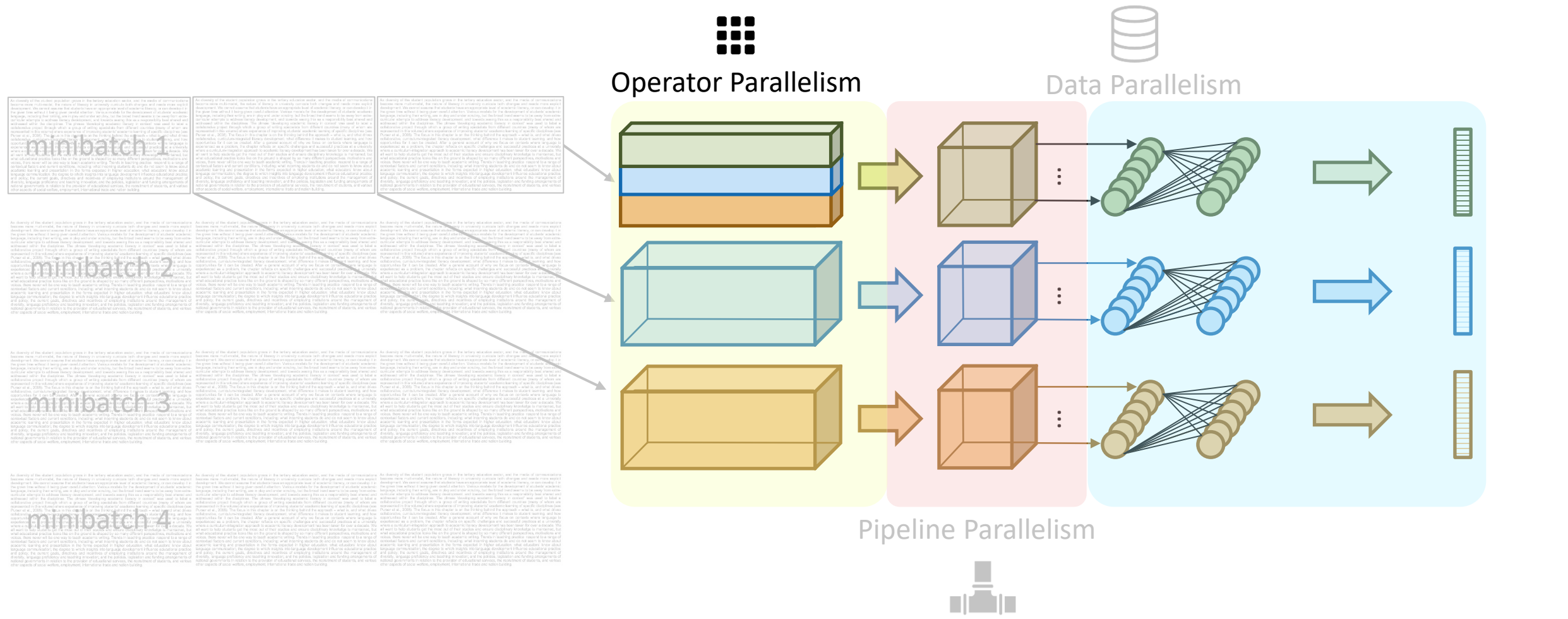
Chimera Weak Scaling (arXiv: 2107.06925v3)



Weak scaling for GPT-2 on Piz Daint
(512 to 2048 GPU nodes)

- **1.38x - 2.34x speedup over synchronous approaches (GPipe, GEMS, DAPPLE)**
 - Less bubbles
 - More balanced memory thus no recomputation
- **1.16x - 2.01x speedup over asynchronous approaches (PipeDream-2BW, PipeDream)**
 - More balanced memory thus no recomputation
 - Gradient accumulation thus low synch frequency

The Three Dimensions of Parallelism in Deep Learning (arXiv:1802.09941)



Operator Parallelism, i.e., Parallel Matrix Matrix Multiplication



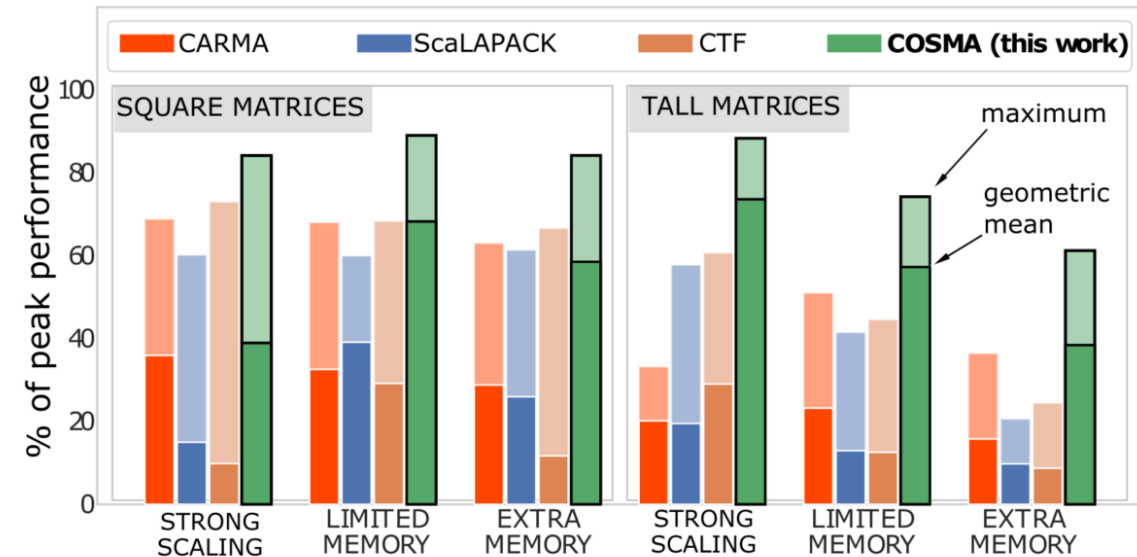
- Large MMMs dominate large language models!
 - e.g., GPT-3 multiples 12,288x12,288 matrices
600 MiB in fp32 and 1.9 Tflop
 - generative inference multiplies tall & skinny matrices

- Distribute as operator parallelism
 - Heaviest communication dimension!
Requires most optimization!

- COSMA [1] communication-optimal distributed MMM**
 - Achieves tight I/O lower bound of $Q \geq \min \left\{ \frac{2mnk}{p\sqrt{S}} + S, 3 \left(\frac{mnk}{p} \right)^{\frac{2}{3}} \right\}$
 - Uses partial replication with an outer-product schedule
See paper for details and proofs!
- AutoDDL [2] combines operator-parallel models into communication-avoiding data distribution**

Remember those?
All MMM!

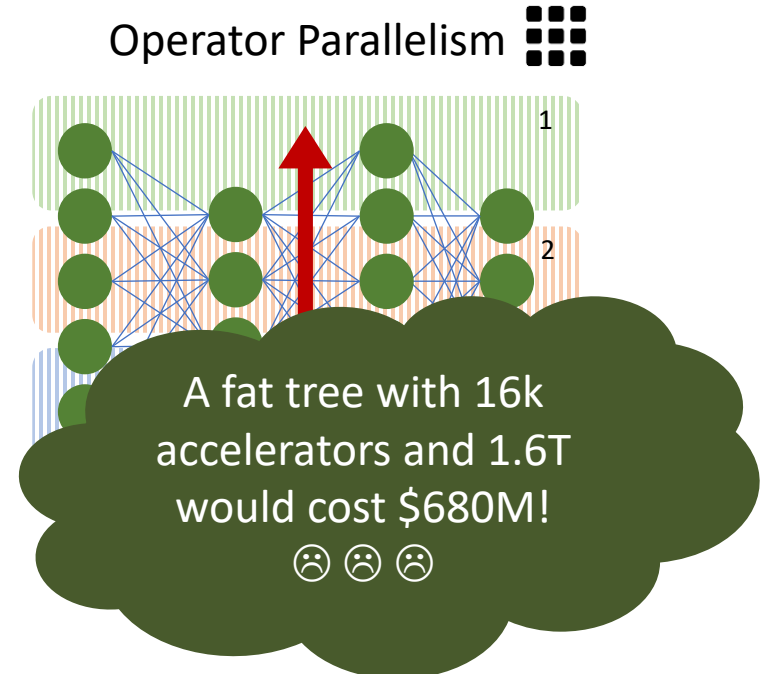
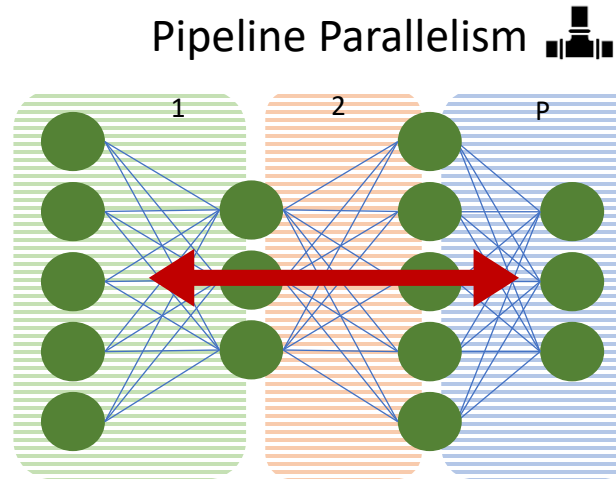
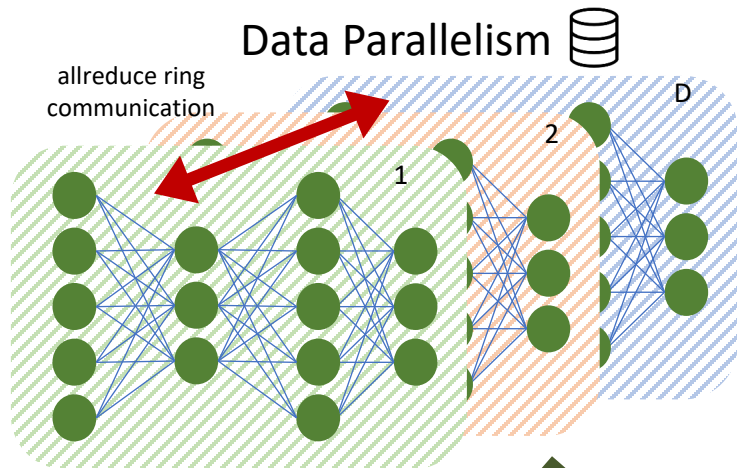
| Operator class | % flop | % Runtime |
|---------------------------|--------|-----------|
| Tensor contraction | 99.80 | 61.0 |
| Statistical normalization | 0.17 | 25.5 |
| Element-wise | 0.03 | 13.5 |



[1] G. Kwasniewski et al.: "Red-Blue Pebbling Revisited: Near Optimal Parallel Matrix-Matrix Multiplication", best student paper at Supercomputing SC19

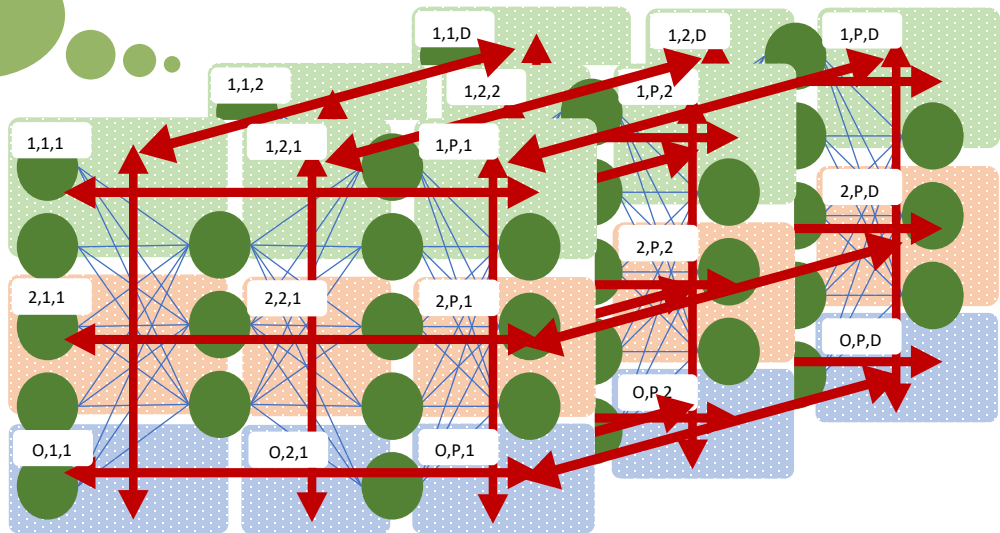
[2] J. Chen et al.: "AutoDDL: Automatic Distributed Deep Learning with Asymptotically Optimal Communication", arXiv

Communications in 3D Parallelism in Deep Learning (arXiv:2209.01346)



Communication is (largely) a logical 3D Torus

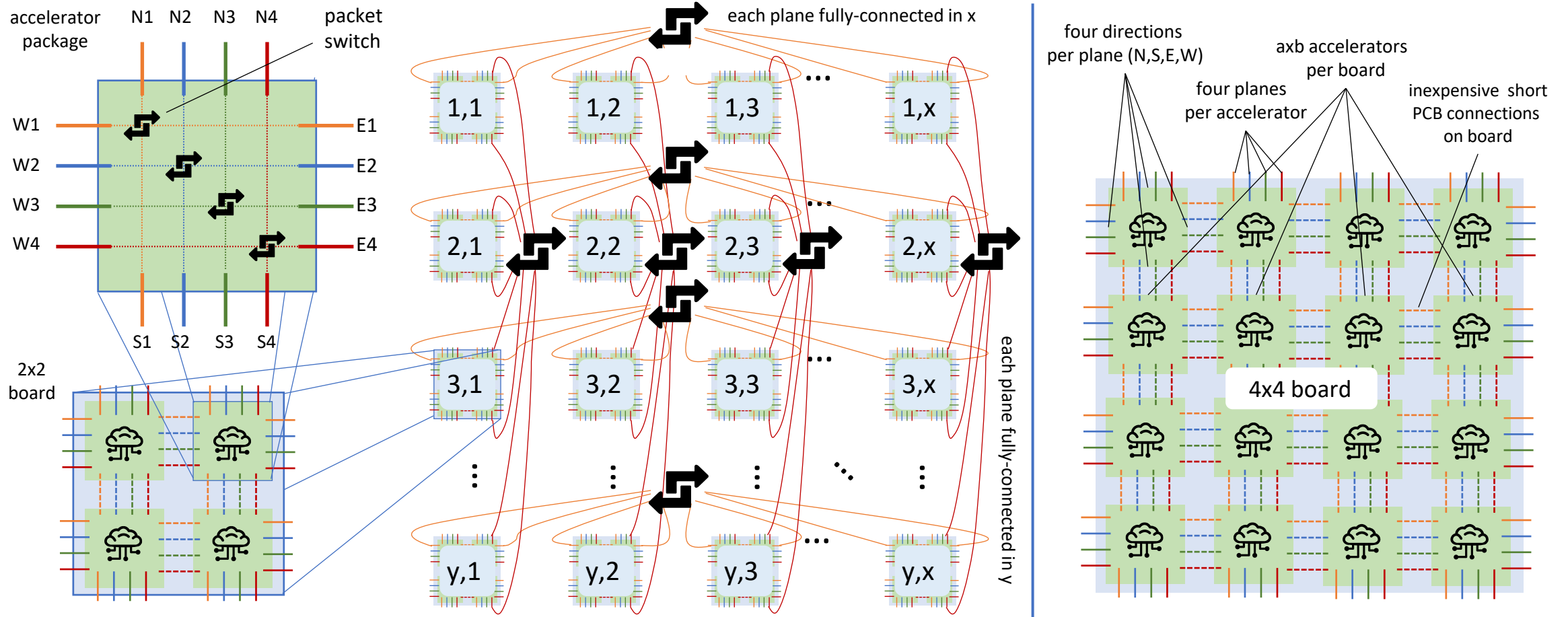
3D - Data, Pipeline, and Operator Parallelism



AI bandwidth today / yesterday (and growing!)

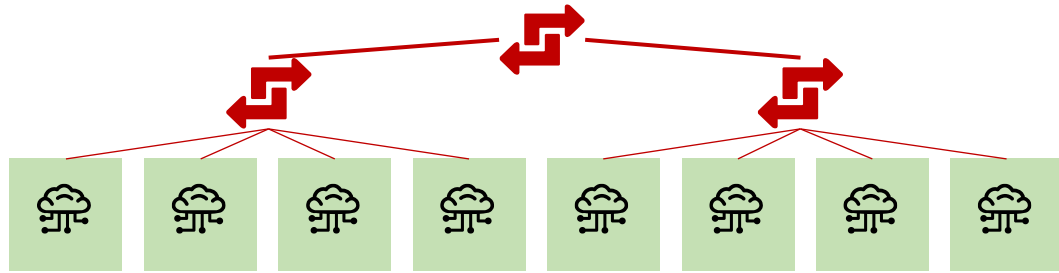
- Google TPUv2 ('21): 1T
- AWS Trainium ('21): 1.6T
- DGX-2 (A100, '21): 4.8T (islands of NVLINK)
- Tesla Dojo ('22): 128T
- Broadcom TH5 / NVIDIA Spectrum 4: 51.2T

Co-designing an AI Supercomputer with Unprecedented and Cheap Bandwidth

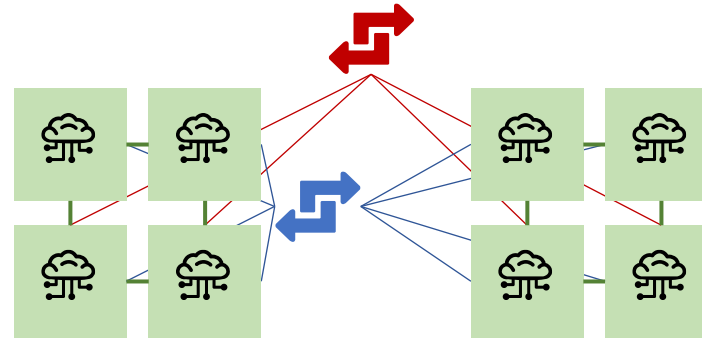


Bandwidth-cost-flexibility Tradeoffs (arXiv:2209.01346)

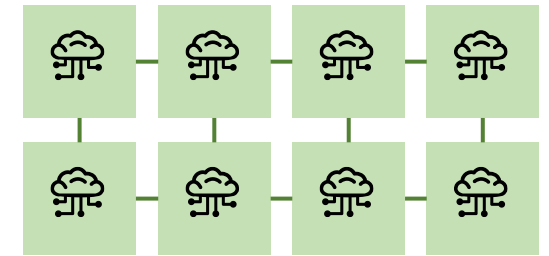
Global Topology
(e.g., Fat Tree)



HammingMesh
(many configurations)



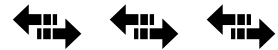
Local Topology
(e.g., 2D Torus)



(large) reduce bandwidth



global bandwidth



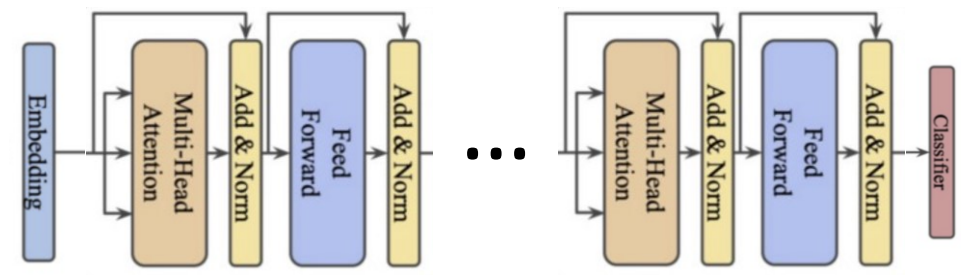
placement flexibility



injection bandwidth



Three Systems Dimensions in Large-scale Super-learning ...



High-Performance I/O

- Quickly growing data volumes
 - Scientific computing!
- Use the specifics of machine learning workloads
 - E.g., intelligent prefetching

High-Performance Compute

- Deep learning is HPC
 - Data movement!
- Drives modern accelerators!

High-Performance Communication

- Use larger clusters (10k+ GPUs)
- Model parallelism
 - Complex pipeline schemes
- Optimized networks

What will the (near future bring)?

Some predictions for the future of HPC but also computing at large!

21 Jan 2021

CLAIRVOYANT: OPTIMIZING FOR DISTRIBUTED TRAINING

Roman Böhringer¹ Nikoll Dryden¹ Tal Ben-Nun¹ Torsten Hoeller¹

ABSTRACT

I/O is emerging as a major bottleneck for machine learning training, especially in distributed environments such as clouds and supercomputers. Optimal data ingestion pipelines differ between systems, and increasing efficiency requires a delicate balance between access to local storage, external filesystems, and remote workers, yet existing frameworks fail to efficiently utilize such resources. We observe that, given the seed generating the random access pattern for training with SGD, we have *clairvoyance* and can exactly predict when a given sample will be accessed. We combine this with a theoretical analysis of access patterns in training and performance modeling to produce a novel machine learning I/O middleware, HDMLP, to tackle the I/O bottleneck. HDMLP provides an easy-to-use, flexible, and scalable solution that delivers better performance than state-of-the-art approaches while requiring very few changes to existing codebases and supporting a broad range of environments.

SLIGHT 2 Jul 2020

Data Movement Is All You Need: A Case Study on Optimizing Transformers

Abstract—Transformers have become widely used for language modeling and sequence learning tasks, and are one of the most important machine learning workloads today. Training one is a very compute-intensive task, often taking days or weeks, and significant attention has been given to optimizing transformers. Despite this, existing implementations do not efficiently utilize GPUs. We find that data movement is the key bottleneck when training. Due to Andrej's Law and massive improvements in compute performance, training has now become memory-bound. Further, existing frameworks use suboptimal data layouts. Using these insights, we present a recipe for globally optimizing data movement in transformers. We reduce data movement by up to 22.91% and overall achieve a 1.30x performance improvement over state-of-the-art frameworks when training BERT. Our approach is applicable more broadly to optimizing deep neural networks, and offers insight into how to tackle emerging performance bottlenecks.

Distribution and Parallelism

Data Pipeline Operator

Prediction 1: Accelerators Converge

AI is a gravity well – HPC will follow

Future Accelerators ...

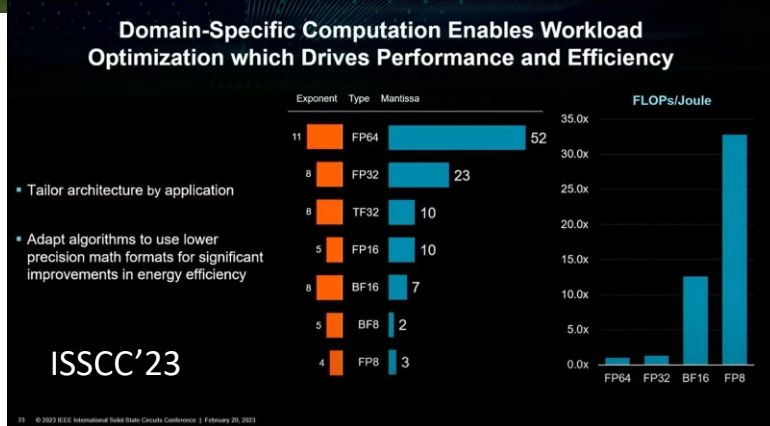
- **Most of the performance will be low precision arithmetic!**
 - I would predict (C)FP8 or smaller
 - We can be lucky if we get some fp64!

- **They will support quantization and sparsity in hardware**
 - Vector scaling and zero points

- **They will heavily be optimized towards data movement**
 - Physical limits and cost introduce two fundamental constraints:
Latency will become a problem
Locality and sparse connectivity
 - Potentially hard to program



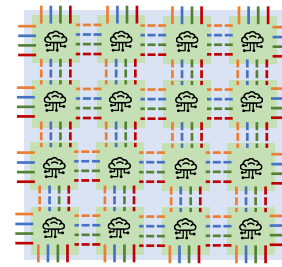
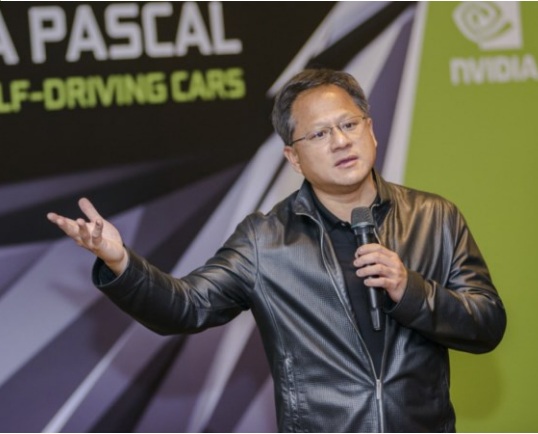
B. Wisniewski (Samsung)
Memory-coupled Compute
 SPCL_Bcast 01/19/23
<https://www.youtube.com/watch?v=KCrQtpx31CQ>



SPECIFICATIONS

| | H100 SXM |
|----------------------|---------------|
| FP64 | 34 TFLOPS |
| FP64 Tensor Core | 67 TFLOPS |
| FP32 | 67 TFLOPS |
| TF32 Tensor Core | 989 TFLOPS* |
| BFLOAT16 Tensor Core | 1,979 TFLOPS* |
| FP16 Tensor Core | 1,979 TFLOPS* |
| FP8 Tensor Core | 3,958 TFLOPS* |
| INT8 Tensor Core | 3,958 TOPS* |

*30x** (arrow from FP64 Tensor Core to FP8 Tensor Core)



Optimized topologies and network technologies.
 E.g., HammingMesh
<https://www.youtube.com/watch?v=xxwT45ljG4o>

Sparse-Quantized Representations - SpQR

SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression

Tim Dettmers* University of Washington
 Ruslan Svirschevski* HSE University & Yandex
 Vage Egiazarian* HSE University & Yandex
Denis Kuznedelev* Yandex & Skoltech
Elias Frantar IST Austria
Saleh Ashkboos ETH Zurich
Alexander Borzunov HSE University & Yandex
Torsten Hoefler ETH Zurich
Dan Alistarh IST Austria & NeuralMagic

Abstract

Recent advances in large language model (LLM) pretraining have led to high-quality LLMs with impressive abilities. By compressing such LLMs via quantization to 3-4 bits per parameter, they can fit into memory-limited devices such as laptops and mobile phones, enabling personalized use. However, quantization down to 3-4 bits per parameter usually leads to moderate-to-high accuracy losses, especially for smaller models in the 1-10B parameter range, which are well-suited for edge deployments. To address this accuracy issue, we introduce the Sparse-Quantized Representation (SpQR), a new compressed format and quantization technique which enables for the first time *near-lossless* compression of LLMs across model scales, while reaching similar compression levels to previous methods. SpQR works by identifying and isolating *outlier weights*, which cause particularly-large quantization errors, and storing them in higher precision, while compressing all other weights to 3-4 bits, and achieves relative accuracy losses of less than 1% in perplexity for highly-accurate LLaMA and Falcon LLMs. This makes it possible to run 33B parameter LLM on a single 24 GB consumer GPU without any performance degradation at 15% speedup thus making powerful LLMs available to consumer without any downsides. SpQR comes with efficient algorithms for both encoding weights into its format, as well as decoding them efficiently at runtime³. Specifically, we provide an efficient GPU inference algorithm for SpQR which yields faster inference than 16-bit baselines at similar accuracy, while enabling memory compression gains of more than 4x.

arXiv:2306.03078v1 [cs.CL] 5 Jun 2023



Prediction 2: Programming and Tools Converge

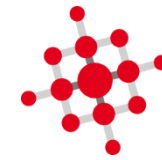
Data Science as a gravity well – HPC will follow

Scientific Computing is Moving to Python (as language frontend/ecosystem)



Tiobe Index June'23

| Jun 2022 | Change | Programming Language |
|----------|--------|--|
| 1 | |  Python |
| 2 | |  C |
| 4 | ↑ |  C++ |
| 3 | ↓ |  Java |
| 5 | |  C# |
| 6 | |  Visual Basic |



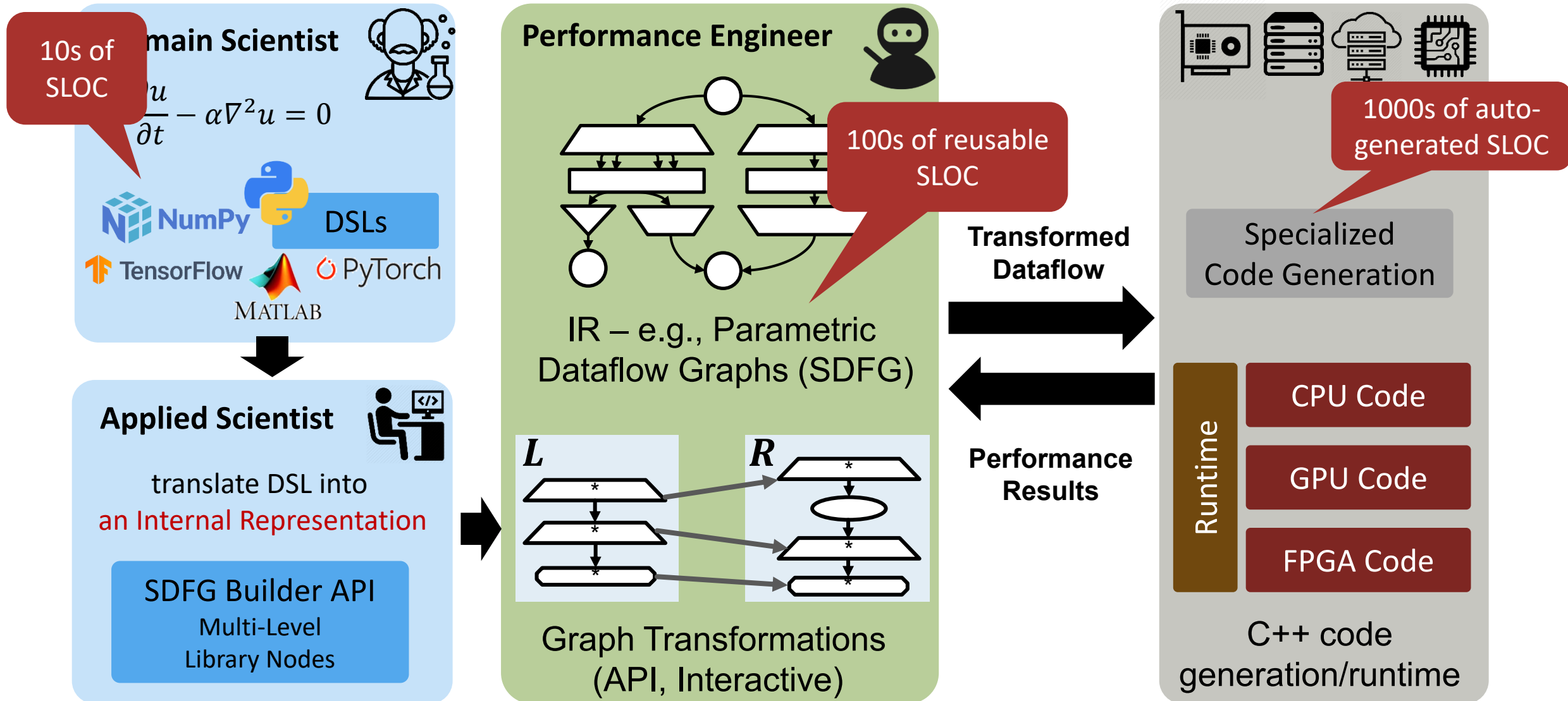
GridTools



439,100 projects

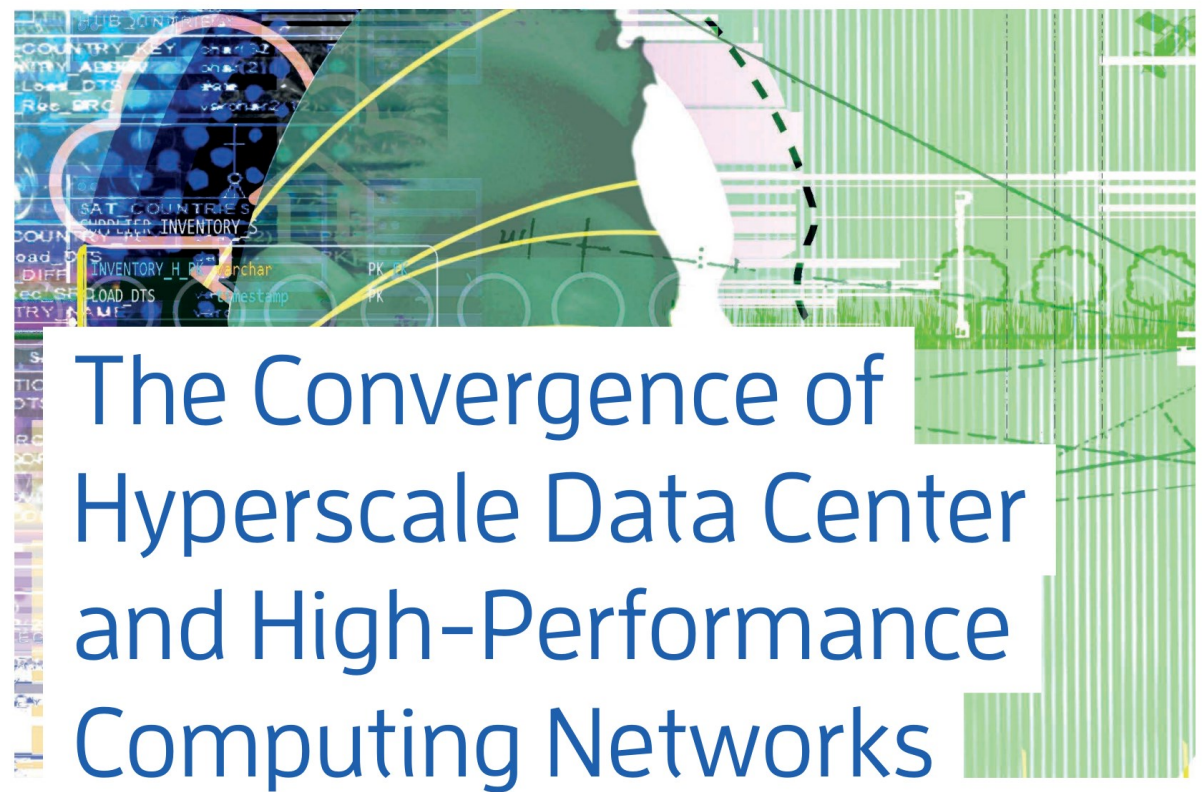


Upleveling Programming in the 21st Century – Performance Metaprogramming



Prediction 3: Networks Converge

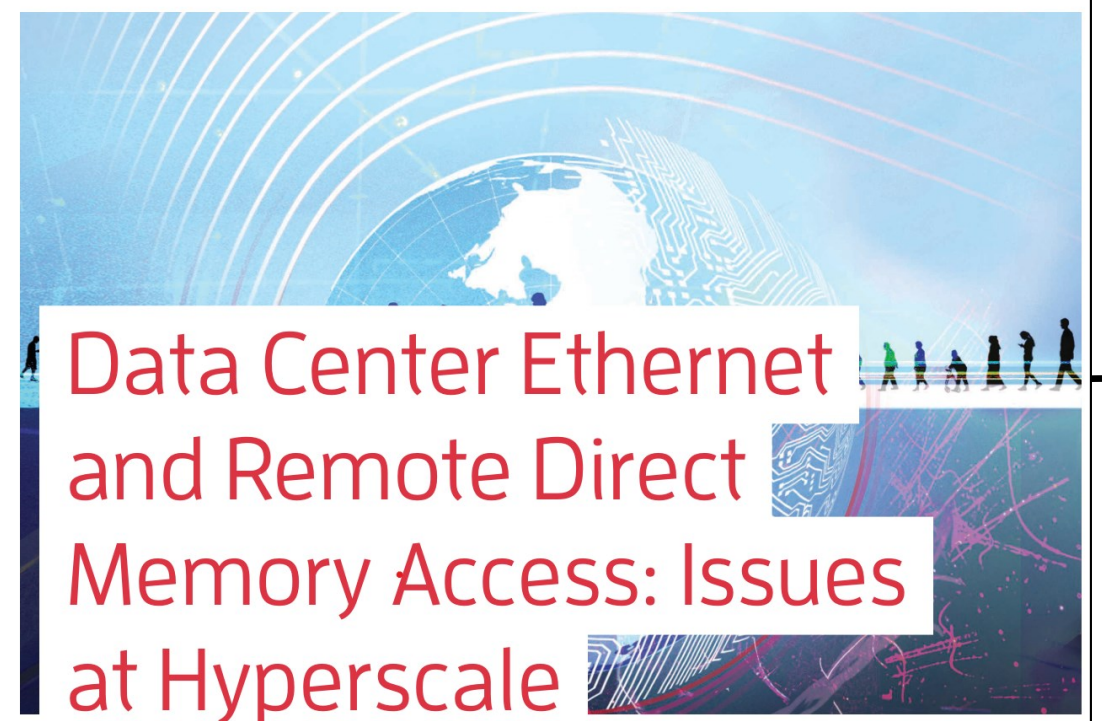
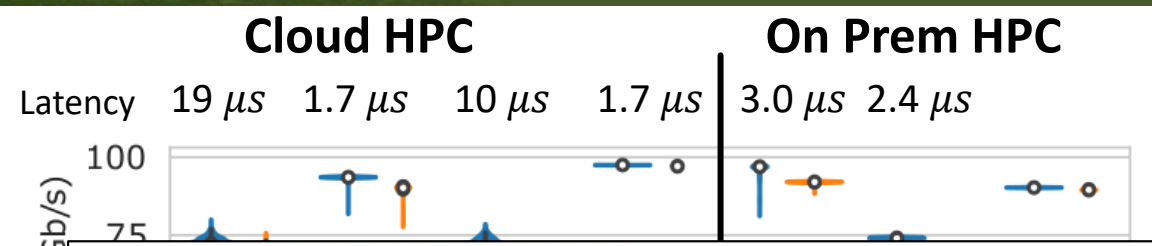
Cloud as a gravity well – HPC will follow



The Convergence of Hyperscale Data Center and High-Performance Computing Networks

Torsten Hoefler, ETH Zurich
Ariel Hendel, Scala Computing
Duncan Roweth, Hewlett Packard Enterprise

We discuss the differences and commonalities between network technologies used in supercomputers and data centers and outline a path to convergence at multiple layers. We predict that emerging smart networking solutions will accelerate that convergence.



Data Center Ethernet and Remote Direct Memory Access: Issues at Hyperscale

Torsten Hoefler¹⁰, ETH Zürich
Duncan Roweth, Keith Underwood, and Robert Alverson, Hewlett Packard Enterprise
Mark Griswold, Vahid Tabatabaee, Mohan Kalkunte, and Surendra Anubolu, Broadcom
Siyuan Shen, ETH Zürich
Moray McLaren, Google
Abdul Kabbani and Steve Scott, Microsoft

[1] De Sensi et al.: "Noise in the Clouds: Influence of Network Performance Variability on Application Scalability", SIGMETRICS'23

Ultra Ethernet Set Out to Create the Best AI/ML and HPC Interconnect!

COVER FEATURE **TECHNOLOGY PREDICTIONS**

Data Center Ethernet and Remote Direct Memory Access: Issues at Hyperscale

Torsten Hoefler¹, ETH Zürich
 Duncan Roweth, Keith Underwood, and Robert Alverson, Hewlett Packard Enterprise
 Mark Griswold, Vahid Tabatabaee, Mohan Kalkunte, and Surendra Anubolu, Broadcom
 Siyuan Shen, ETH Zürich
 Moray McLaren, Google
 Abdul Kabbani and Steve Scott, Microsoft

Ultra Ethernet Consortium

Founding Members



Ultra Ethernet Consortium

white Paper on ultraethernet.org

Overview of and Motivation for the Forthcoming Ultra Ethernet Consortium Specification

Networking Demands of Modern AI Jobs

Networking is increasingly important for efficient and cost-effective training of AI models. Large Language Models (LLMs) such as GPT-3, Chinchilla, and PALM, as well as recommendation systems like DLRM and DHEN, are trained on clusters of thousands of GPUs.

Key Points and Conclusions

Three Systems Dimensions in Large-scale Super-learning ...

High-Performance I/O

- Quickly growing data volumes
- Scientific computing!
- Use the specifics of machine learning workloads
- E.g., intelligent prefetching

High-Performance Compute

- Deep learning is HPC
- Data movement!**
- Quantization, Sparsification**
- Drives modern accelerators!

Data Movement is All You Need: A Case Study on Optimizing Transformers

High-Performance Communication

- Use larger clusters (10k+ GPUs)
- Model parallelism
- Complex pipeline schemes
- Optimized networks

Distribution and Parallelism

| Data | Pipeline | Operator |
|------|----------|----------|
| | | |

Programming Sparse Models – Meet PyTorch Sten (arXiv:2304.07613)

Sparsity Layouts

- Dense Tensor
- Dimensions
- Strides
- Dense values

- Sparse Tensor
- Dimensions
- Sparsity Format
- Compressed values

Operators

Layout → Implementation defined by input/output sparsity layout

Sparsifiers

Keep-all, Random fraction, Scalar threshold ...

Selected Available Sparsifiers:

Streaming

- Keep all
- do not drop
- Random fraction: drop if rand < 0.5
- Scalar threshold: drop if value < 0

Blocked

- Per block fraction
- Find block quartile q
- Drop if below

Materializing

- Scalar fraction
- Find quartile q
- Drop if below

Co-designing an AI Supercomputer with Unprecedented and Cheap Bandwidth

accelerator package (N1, N2, N3, N4) and packet switch (E1, E2, E3, E4) are connected to a 2x2 board (S1, S2, S3, S4). The diagram shows a grid of accelerators (1,1 to 4,4) with interconnections. A 4x4 board is also shown with four directions per plane (N,S,E,W) and four planes per accelerator. Inexpensive short PCB connections on board are highlighted.

Prediction 1: Accelerators Converge

AI is a gravity well – HPC will follow

Prediction 2: Programming and Tools Converge

Data Science as a gravity well – HPC will follow

Prediction 3: Networks Converge

Cloud as a gravity well – HPC will follow

More of SPCL's research:

youtube.com/@spcl

180+ Talks

twitter.com/spcl_eth

1.4K+ Followers

github.com/spcl

2K+ Stars

... or spcl.ethz.ch



Want to join our efforts?
We're looking for excellent
Postdocs, PhD students, and Visitors.
Talk to me!

The Convergence of Hyperscale Data Center and High-Performance Computing Networks

Tarsten Hofer, ETH Zurich
Amit Hiran, Scale Computing
Duncan Russell, Hewlett Packard Enterprise

We discuss the differences and commonalities between network technologies used in supercomputers and data centers.

Data Center Ethernet and Remote Direct Memory Access: Issues at Hyperscale

Tarsten Hofer, ETH Zurich
Duncan Russell, Keith Underwood, and Robert Averson, Hewlett Packard Enterprise
Mark Gwinell, Vektor Technologies, Mithun Kulkarni, and Suresh Anandakrishnan, Broadcom
Suresh Anandakrishnan, ETH Zurich
Murray McLaren, Google
Abdul Kader and Steve Scott, Microsoft