A Distributed Multilinear Algebra Library for Deep Learning

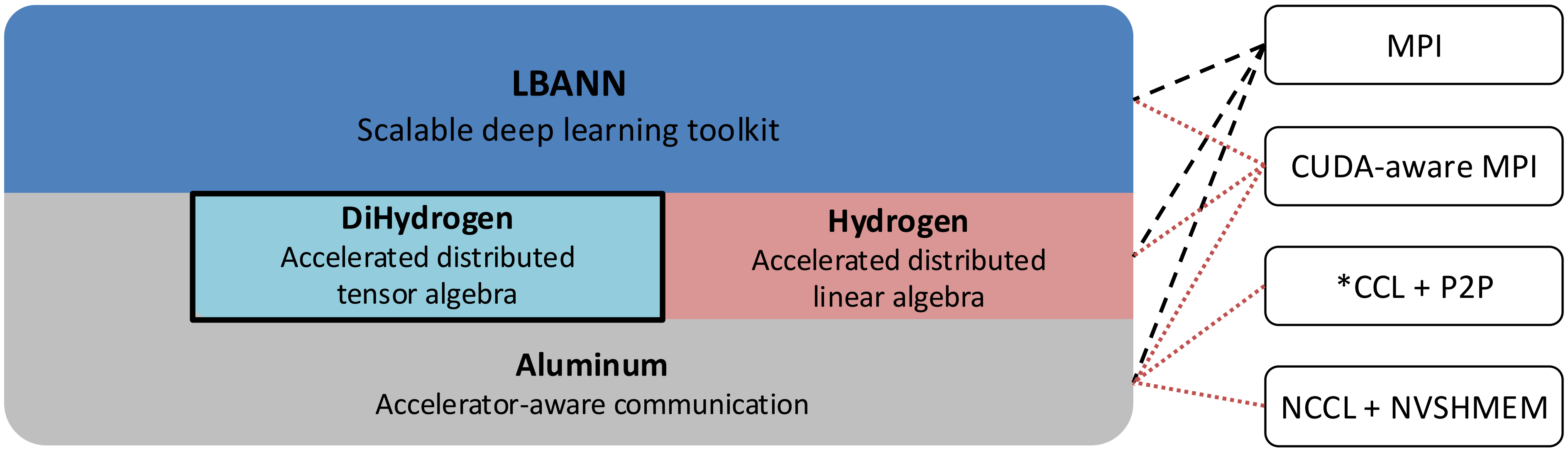
BLIS Retreat 2024

26 September, 2024

Nikoli Dryden
on behalf of the Hydrogen, Aluminum, & LBANN teams
and many collaborators

LLNL-PRES-869692
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

HAL: LLNL's deep learning stack for leadership-class HPC systems



- Open-source libraries
- C++ / MPI / OpenMP
  - CUDA + cuDNN + NCCL + NVSHMEM
  - ROCm + MIOpen + RCCL
  - OneDNN
- PyTorch interface via Torch Dynamo and Torch Inductor
- Support for model exchange with PyTorch

A brief history of HAL

≤2013: Elemental

2015: LBANN

2017: Hydrogen (fork of Elemental)

2018: Aluminum

2019: Distconv

[Many other things omitted…]

2023: DiHydrogen (in development)

… Today

Elemental: A New Framework for Distributed Memory Dense Matrix Computations ACM TOMS 2013

LBANN: Livermore Big Artificial Neural Network HPC Toolkit MLHPC 2015

Aluminum: An Asynchronous, GPU-Aware Communication Library Optimized for Large-Scale Training of Deep Neural Networks on HPC Systems MLHPC 2018

Improving Strong-Scaling of CNN Training by Exploiting Finer-Grained Parallelism IPDPS 2019

Channel and Filter Parallelism for Large-Scale CNN Training Supercomputing 2019

Looking for info on use of Elemental in LBANN #179

Closed **rvdg** opened this issue on Mar 28 · 1 comment

rvdg commented on Mar 28 ···

Greetings,

I am trying to get in touch with whoever forked Elemental for use in LBANN and/or has or had involvement in that effort. Kindly contact me at rvdg@cs.utexas.edu.

Thanks
Robert

Challenges of large-scale scientific machine learning

Massive data sets (number of samples)

- Challenges: Data parallelism provides limited scaling as learning is impacted by large mini-batch sizes
- Solutions: **Tournament learning methods with partitioned data sets**

Large sample sizes

- Challenges: Single sample and neural network activations do not fit on single accelerator
- Solutions: **Distributed convolutions with halo exchanges**

Large models

- Challenges: Model weights do not fit on a single accelerator
- Solutions: **Model- and sub-graph parallelism splits model compute graph over multiple accelerators**

Complex models

- Challenges: Models are highly interconnected and require irregular communication (graph neural networks)
- Solutions: **Communication-efficient dense-scatter algorithms**

Complex algorithms

- Challenges: Second-order optimization methods are expensive to compute and have high memory requirements
- Solutions: **Sub-graph parallelism splits optimizer state over multiple accelerators**

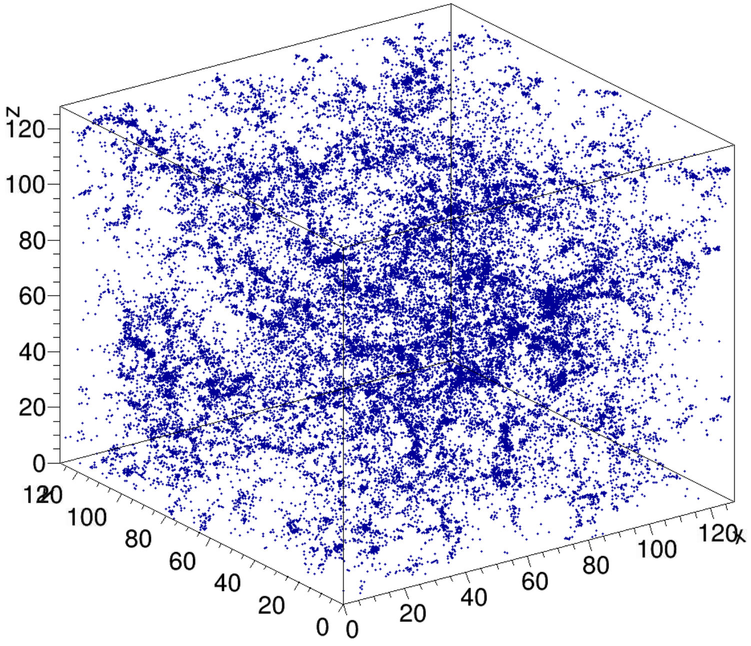
Why another DL framework?

- Existing frameworks did not offer sufficient performance at scale
  - Not easy to conduct surgery to improve them
  - (Becoming less true for LLM workloads: Megatron, DeepSpeed, Torch Titan, etc.)
- Python is a distributed denial-of-service attack on your supercomputer
- Memory and communication inefficiencies
- Support leadership-class systems with unusual hardware
- Enable near-peak performance for critical workloads
- Be a vehicle for DL systems R&D

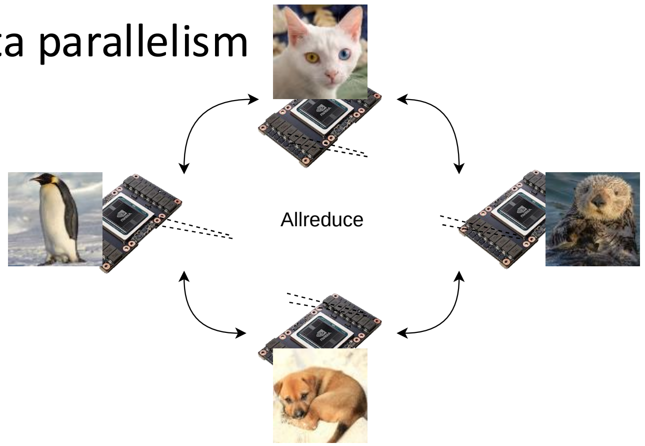
LBANN is a training deployment framework for LLNL's bespoke application needs

Distributed convolutions for very large data samples

- Surrogate models for simulations require very large input data volumes
- CosmoFlow: 512^3, 4 channels, 2 bytes/element
  - MLPerf-HPC uses a smaller version (128^3)
  - 1 GiB per sample
  - Regression model does not fit into most accelerator's memory
  - Tensor strides need 64-bit integers

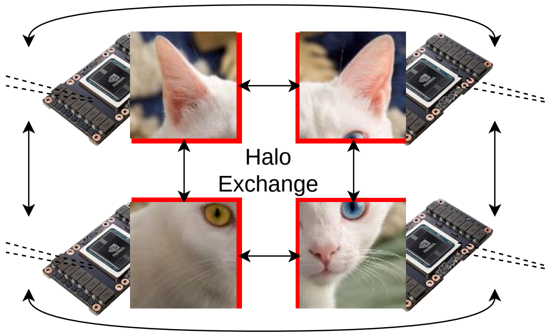


Data parallelism



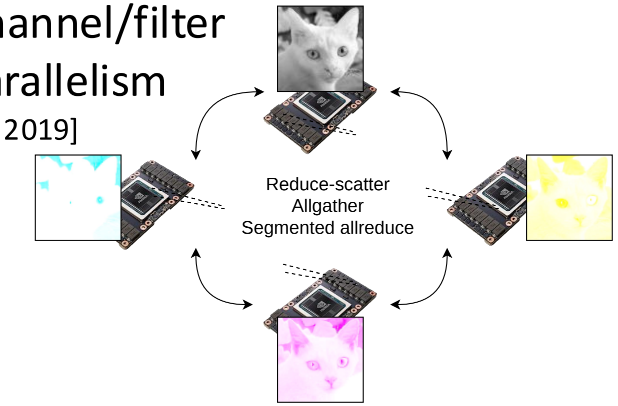
Spatial parallelism

[IPDPS 2019]



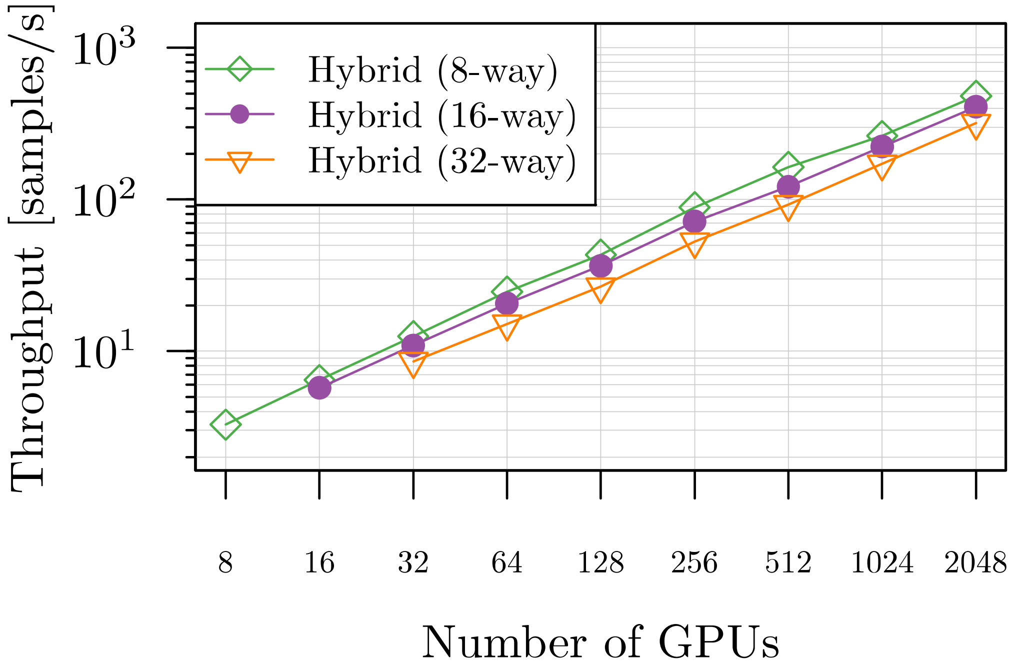
Channel/filter parallelism

[SC 2019]



Spatial parallelism enables new scales for CNNs

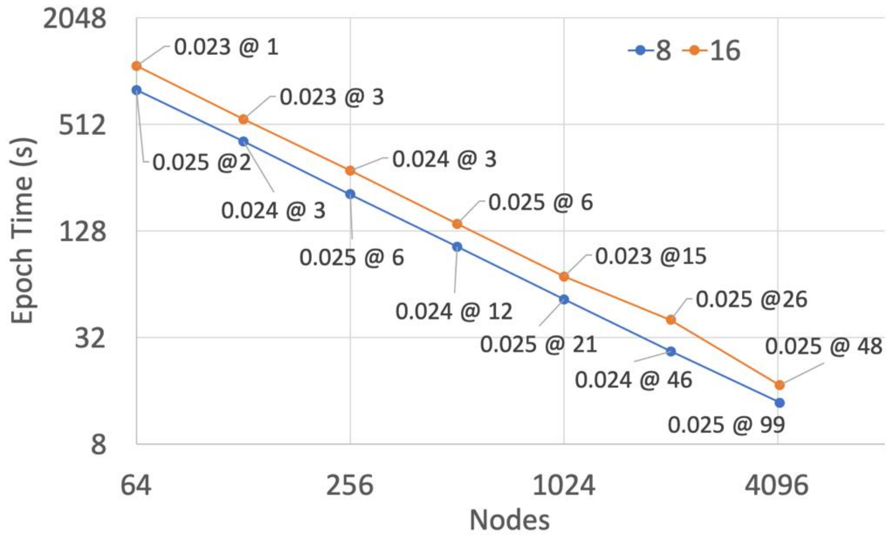
- CosmoFlow network with 512^3 samples
- Lassen (4x V100 / node)
- Network requires ~53 GiB / sample
- Standard data parallelism is not possible

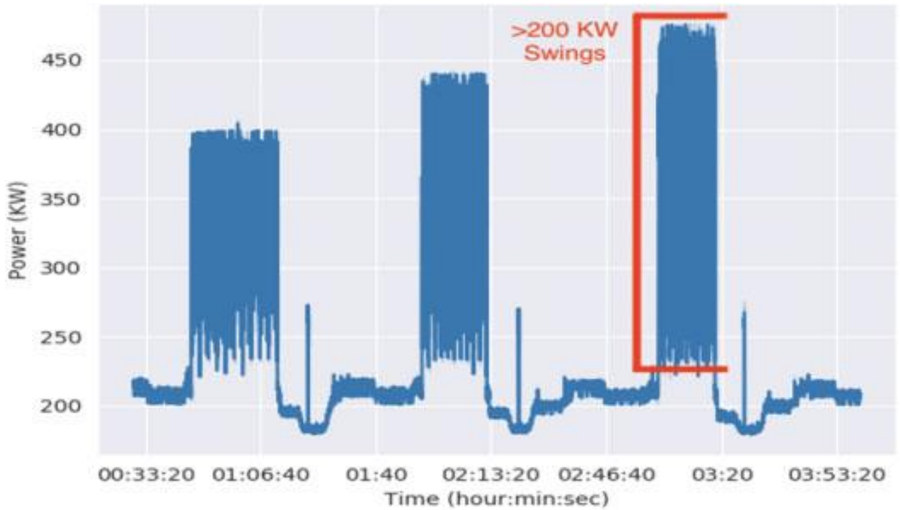


Oyama et al., "The Case for Strong Scaling in Deep Learning: Training Large 3D CNNs with Hybrid Parallelism." IEEE TPDS 2020

Tournament voting algorithms for extreme-scale training

- Many trainers with partitioned datasets
- Periodically exchange models with random peers and run local tournament
- Enables scaling to full Sierra (4160 nodes)
- 2020 Gordon Bell COVID-19 Special Prize finalist





≥2.4 MW power swings for the whole system!

Jacobs et al., “Enabling rapid COVID-19 small molecule drug design through scalable deep learning of generative models”, IJHPCA 2021

What does a deep learning need from a multilinear algebra library?

Not a lot

- (But if you give us more toys, we'll find a way to (ab)use them.)

- (Distributed) (Batched) Matrix-matrix multiply + BLAS1
  — More generally: Einstein summation support
- Convolutions
- A handful of sparse operations for GNNs
- Block distributions of multi-dimensional arrays
- Communication operations
- Low & mixed precision computations (FP16, BF16, FP8, int8, …)
- Really high performance on accelerators

$\text{BERT}_{\text{LARGE}}$

Operator class	% flop	% runtime
Tensor contraction	99.8	61.0
Statistical normalization	0.17	25.5
Element-wise	0.03	13.5
	0.2%	**39%**

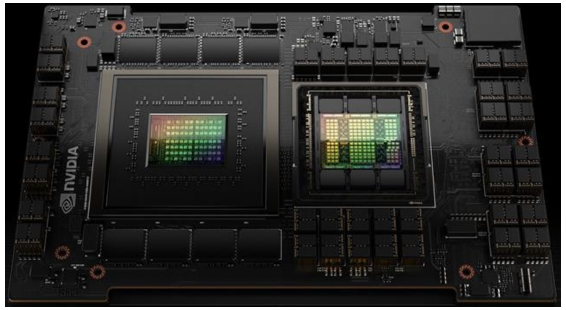
Ivanov et al., "Data Movement Is All You Need: A Cast Study on Optimizing Transformers", MLSys 2021

Limitations of Elemental/Hydrogen: We need tensors

- CNNs and transformers need 3d–5d tensors
  - Batch x Channels x Height x Width x Depth or Batch x Sequence x Embedding
- Block distributions
  - Elemental distributions are less useful
- Multi-dimensional permutations are critical
  - Convolution prefers channels-last
  - Multi-head attention shifts sequence and embedding
- Data partitioning and redistribution needs this semantic information
- **Matrices (order-2 tensors) are not sufficient**
- **More complicated partitioning schemes are needed**

Future needs for large-scale deep learning training (non-exhaustive)

- Enable performance on emerging architectures:
  - El Capitan supercomputer at LLNL
  - MI300A APUs & Grace-Hopper superchips provide unified memory
  - Multi-node NVLink (NVL72) provide large cliques of high-bandwidth connectivity
- Fault tolerance and elasticity for long runs (weeks to months)
- Composition of many parallelism modes while maintaining efficiency
- High performance for our workloads: Being 10% faster matters!



Pictures: LLNL; Nvidia

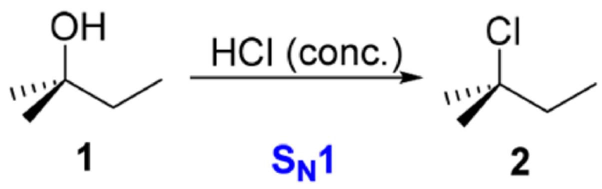
FLASK: Foundation Learning AI for Synthesis Knowledge

Supply chain issues and new threats require rapid discovery and manufacture of materials

FLASK is creating a foundation model for molecular design and synthesis pathway prediction:

1. Predict novel molecules with specified properties
2. Enable lead molecule design — generate candidate molecules with similar structure and properties
3. Predict synthesis pathways for known and novel compounds
4. Enable pathway optimization based on SME inputs

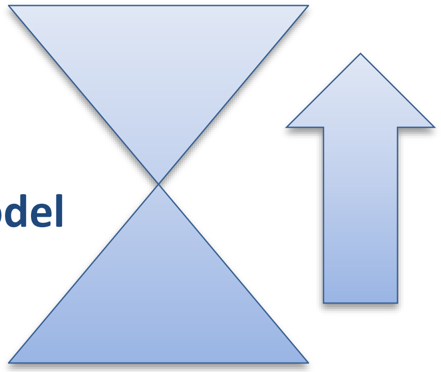
Proposed Synthesis Pathway

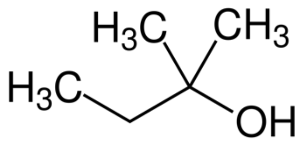


Density:
0.805 g/cc.

Heat of formation:
-380.0 kJ/mol

Generative AI Foundation Model



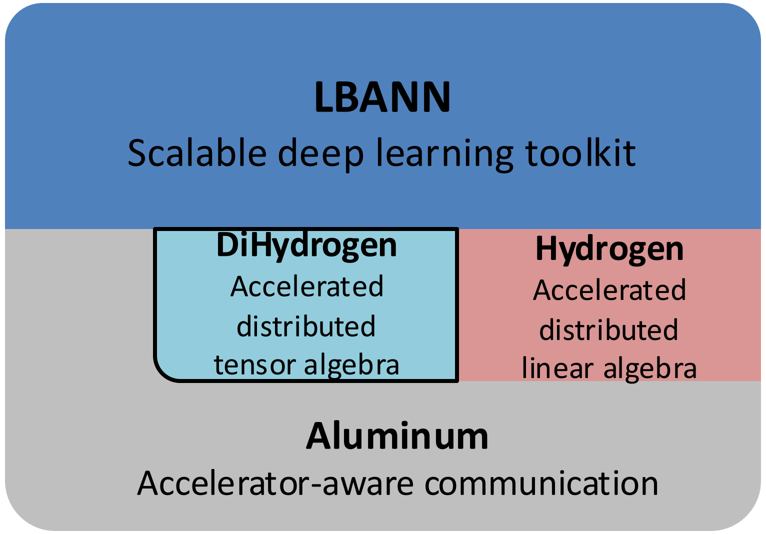


Known Molecule

Tert-Amyl alcohol

DiHydrogen is LBANN's distributed multilinear algebra library for DL

Supports LBANN as a high-performance training deployment framework for our apps



github.com/LLNL/LBANN
github.com/LLNL/Elemental
github.com/LLNL/DiHydrogen
github.com/LLNL/Aluminum