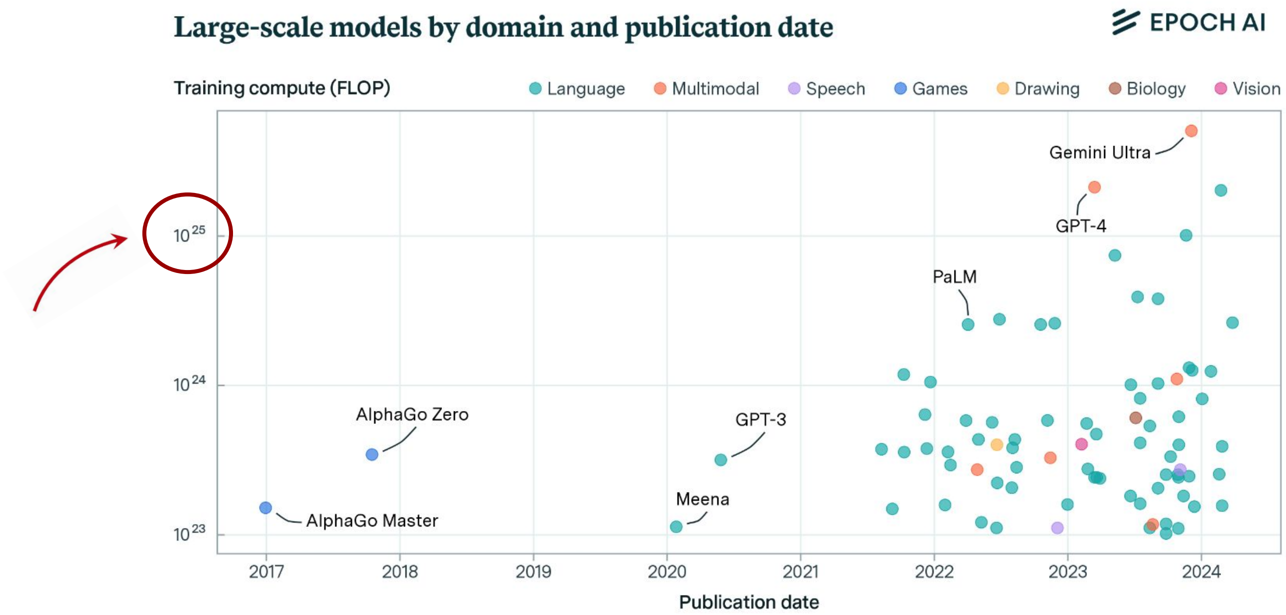
Performance Modeling and System Design Insights for Scientific AI Foundation Models

Shashank Subramanian

NERSC, Berkeley Lab

Comprehensive Performance Modeling and System Design Insights for Foundation Models, PMBS, SC24, Github

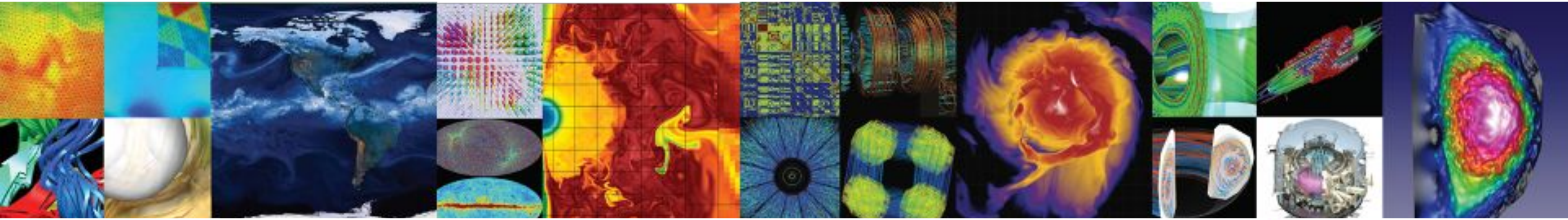
AI Foundation Models are Expensive

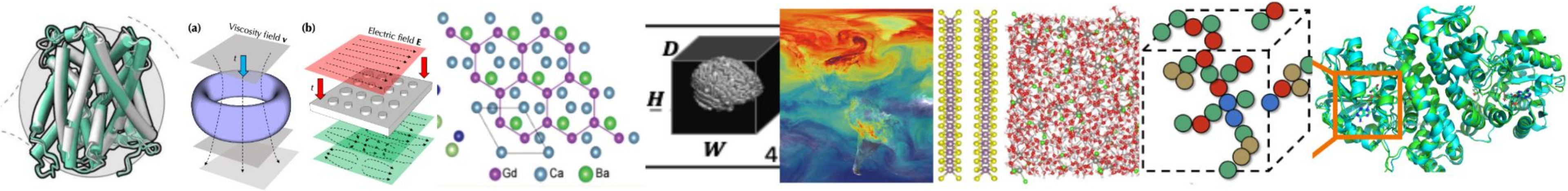


EpochAI

- Transformers are the workhorse: Scaling properties, flexible, SOTA results

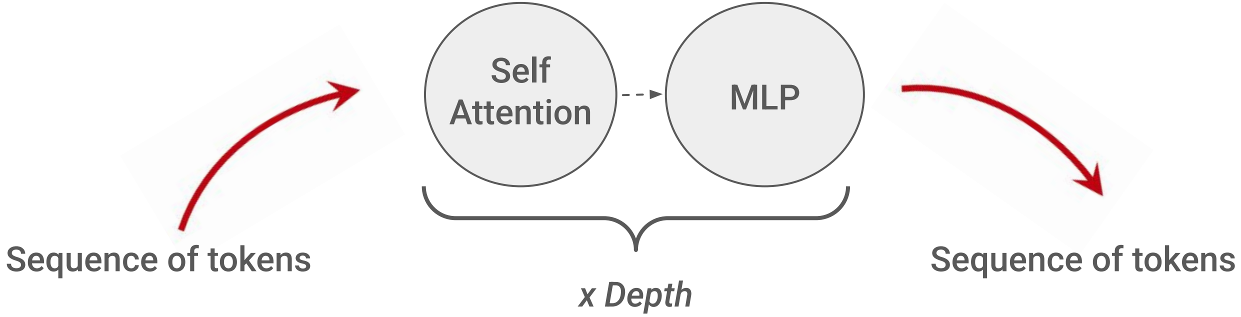
Large-scale AI Models are Growing in Science



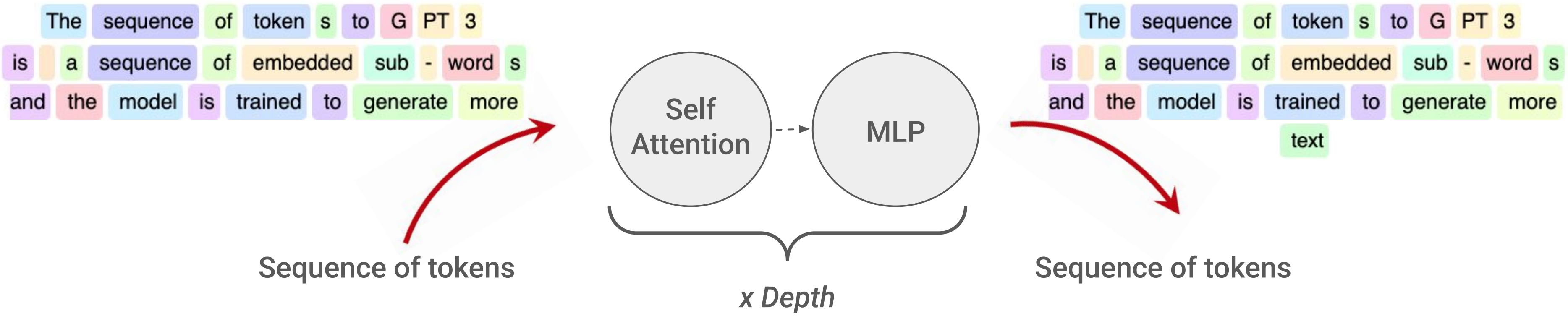


- Range of scientific simulation tasks is enormous
    - Weather/climate, fusion, seismic, fluids, proteins, material sciences, high-energy physics
- Surge of transformer models as possible *foundations* for downstream tasks

Transformers in Science may Operate in Different Regimes

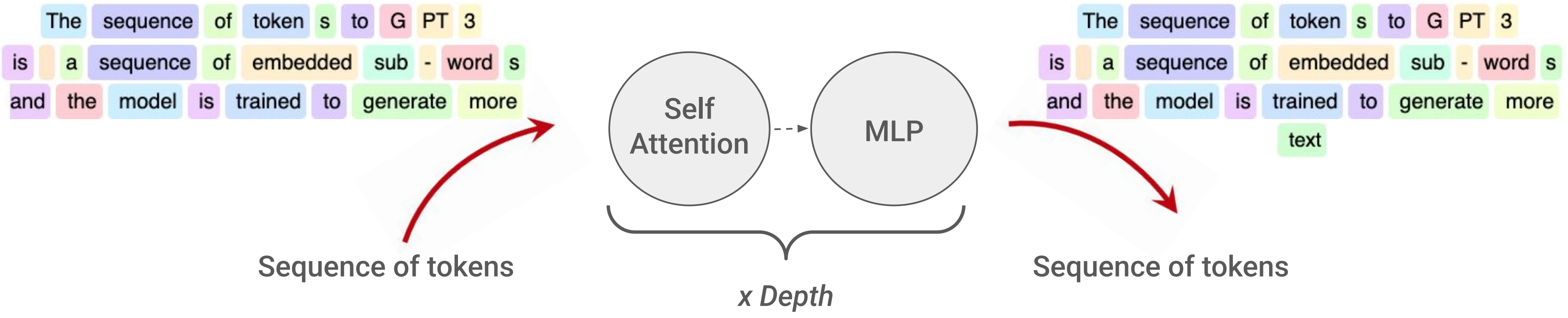


Transformers in Science may Operate in Different Regimes



- A Large Language Model (LLM) example: GPT3

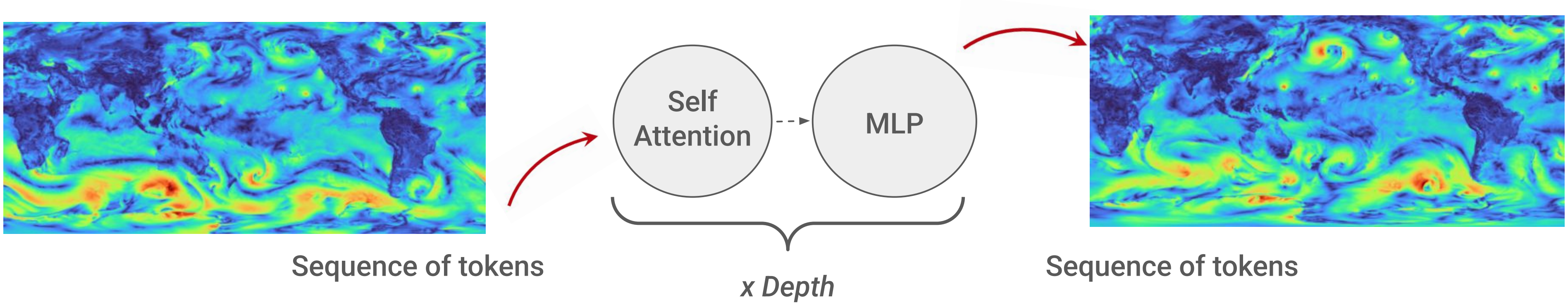
Transformers in Science may Operate in Different Regimes



- A Large Language Model (LLM) example: GPT3
  - #Parameters can be huge ~ **billions to trillions** of parameters
  - Process a sequence of O(1K) tokens (usually 2K, 4K, 8K tokens in pre-training)
  - MLP FLOPs are large (compared to S/A)
  - GPT3-1T on **3072 A100 GPUs** takes **84 days** to train on 450B tokens
  - Understood reasonably well

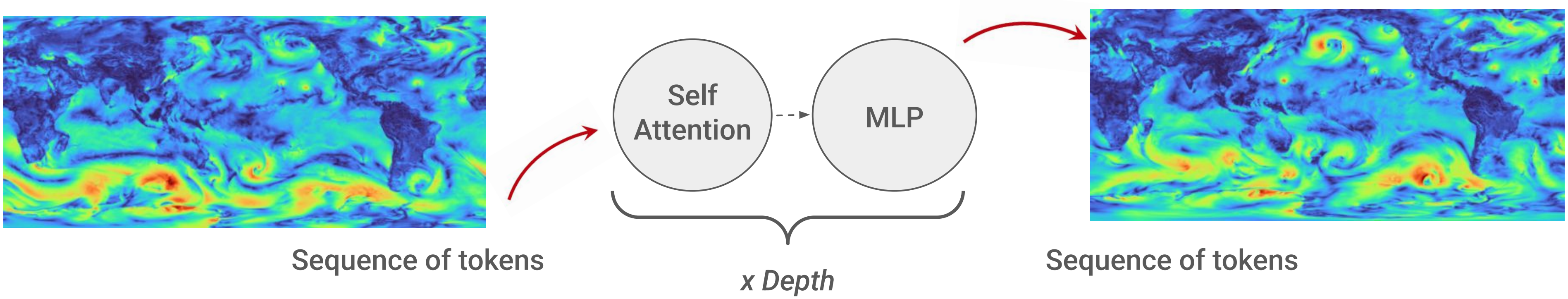
Token Visualizer, Megatron-LM

Transformers in Science may Operate in Different Regimes



- A Scientific Surrogate example: Transformer for global weather forecasting

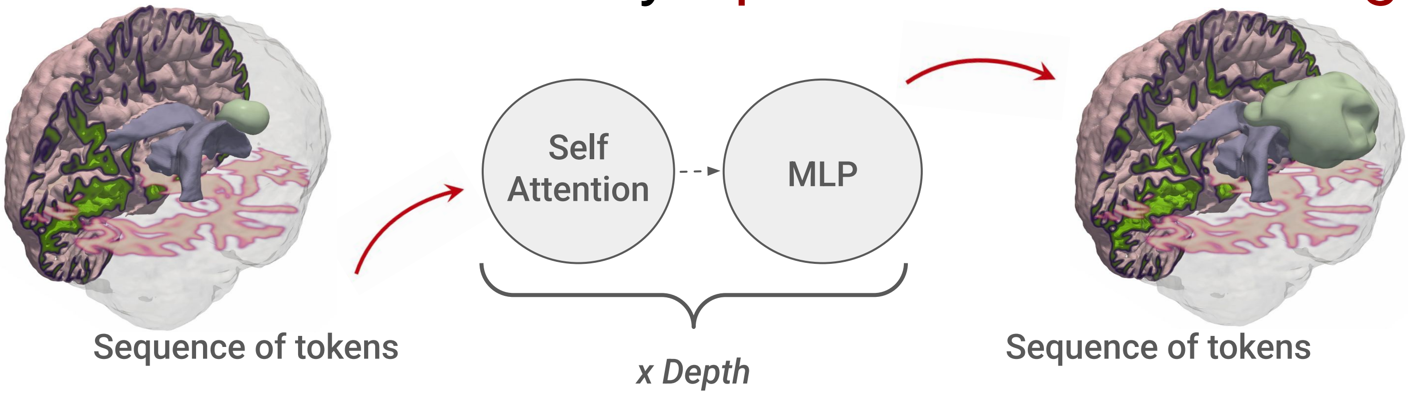
Transformers in Science may Operate in Different Regimes



- A Scientific Surrogate example: Transformer for global weather forecasting
  - #Parameters are moderate ~ million to billion parameters
  - Process a sequence of O(1M) tokens (can be compressed to O(100K) tokens)
  - S/A FLOPs are large (compared to MLP)
  - **A small model could be more expensive than a trillion parameter LLM!**
  - [?] Days on [?] GPUs on [?] tokens. Less understood

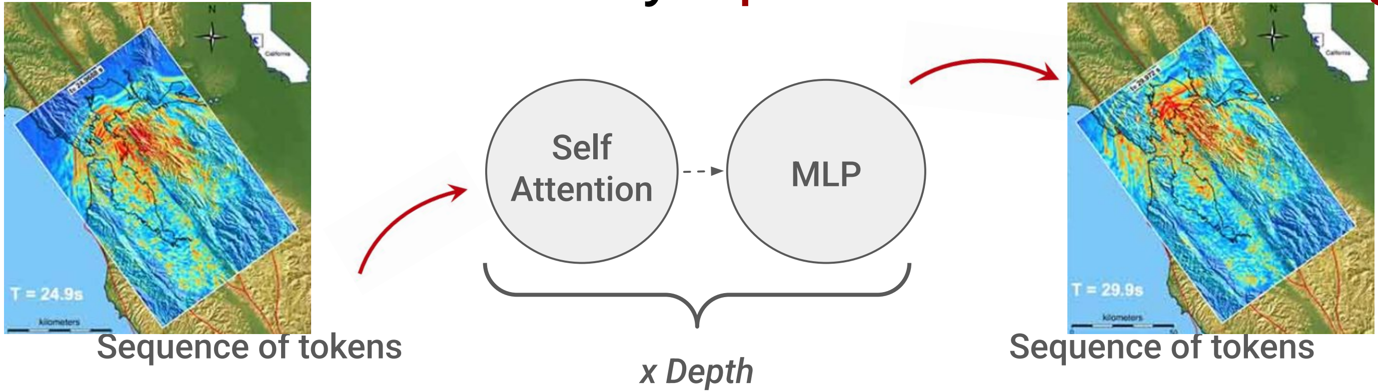
Scientific Foundation Models

Transformers in Science may Operate in Different Regimes



- A Scientific Surrogate example:
  - #Parameters are moderate ~ million to billion parameters
  - Process a sequence of O(1M) tokens
  - S/A FLOPs are large (compared to MLP)
  - **A small model could be more expensive than a trillion parameter LLM!**
  - [?] Days on [?] GPUs on [?] tokens. Less understood

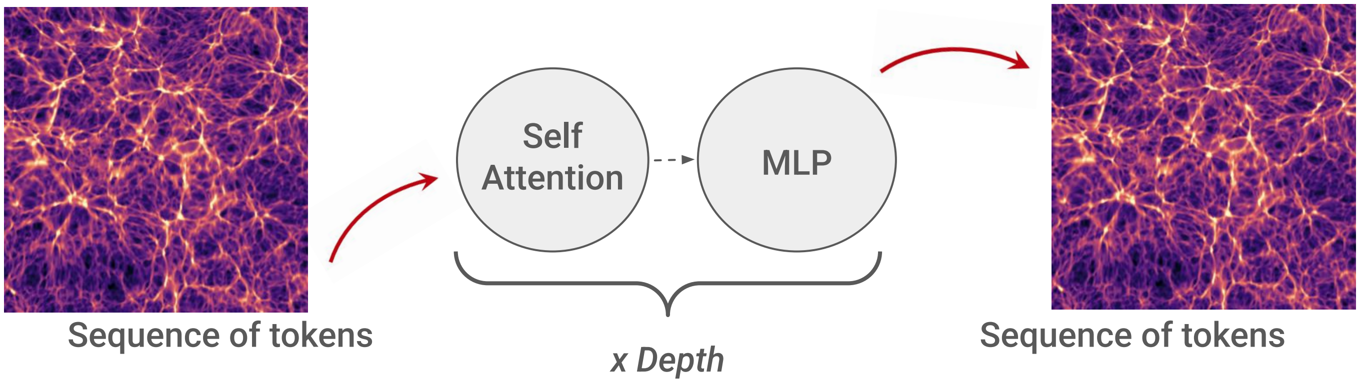
Transformers in Science may Operate in Different Regimes



EQSIM

- A Scientific Surrogate example:
  - #Parameters are moderate ~ million to billion parameters
  - Process a sequence of O(billions) tokens
  - S/A FLOPs are large (compared to MLP)
  - **A small model could be more expensive than a trillion parameter LLM!**
  - [?] Days on [?] GPUs on [?] tokens. Less understood

Transformers in Science may Operate in Different Regimes



[ACCEL2]

- A Scientific Surrogate example:
  - #Parameters are moderate ~ million to billion parameters
  - Process a sequence of O(billions) tokens
  - S/A FLOPs are large (compared to MLP)
  - **A small model could be more expensive than a trillion parameter LLM!**
  - [?] Days on [?] GPUs on [?] tokens. Less understood

Performance Modeling can be Valuable

- Understand **Costs/Bottlenecks** and analyze **Sensitivity of Performance**
  - What bottlenecks w.r.t parallelization strategies?
  - Different Transformer regimes (LLMs vs Science)?
  - Different system hardware (specifically network/NVLINK effects)?
  - Different system scales (10s vs 1000s of accelerators)?

Performance Modeling can be Valuable

- Understand Costs/Bottlenecks and analyze Sensitivity of Performance
  - What bottlenecks w.r.t parallelization strategies?
  - Different Transformer regimes (LLMs vs Science)?
  - Different system hardware (specifically network/NVLINK effects)?
  - Different system scales (10s vs 1000s of accelerators)?

- Value-add for:
  - Users (researchers, engineers)
    - Optimal ways to parallelize AI models? Architecture search with performance in mind?
  - Systems design
    - Which aspects of the HPC system are crucial? Alternate design choices?

AI Performance Modeling is Challenging

Design Space

AI Performance Modeling is Challenging

Parallelism
Data and Tensor (1D, 2D, 3D), Pipeline (Schedules), Optimizer Sharding, CPU Offloading

Design Space

AI Performance Modeling is Challenging

Parallelism
Data and Tensor (1D, 2D, 3D), Pipeline (Schedules), Optimizer Sharding, CPU Offloading

Design Space

AI Architecture
Multiple hyperparameters for Transformer, Different variants (Convs, Spectral), Other Models

AI Performance Modeling is Challenging

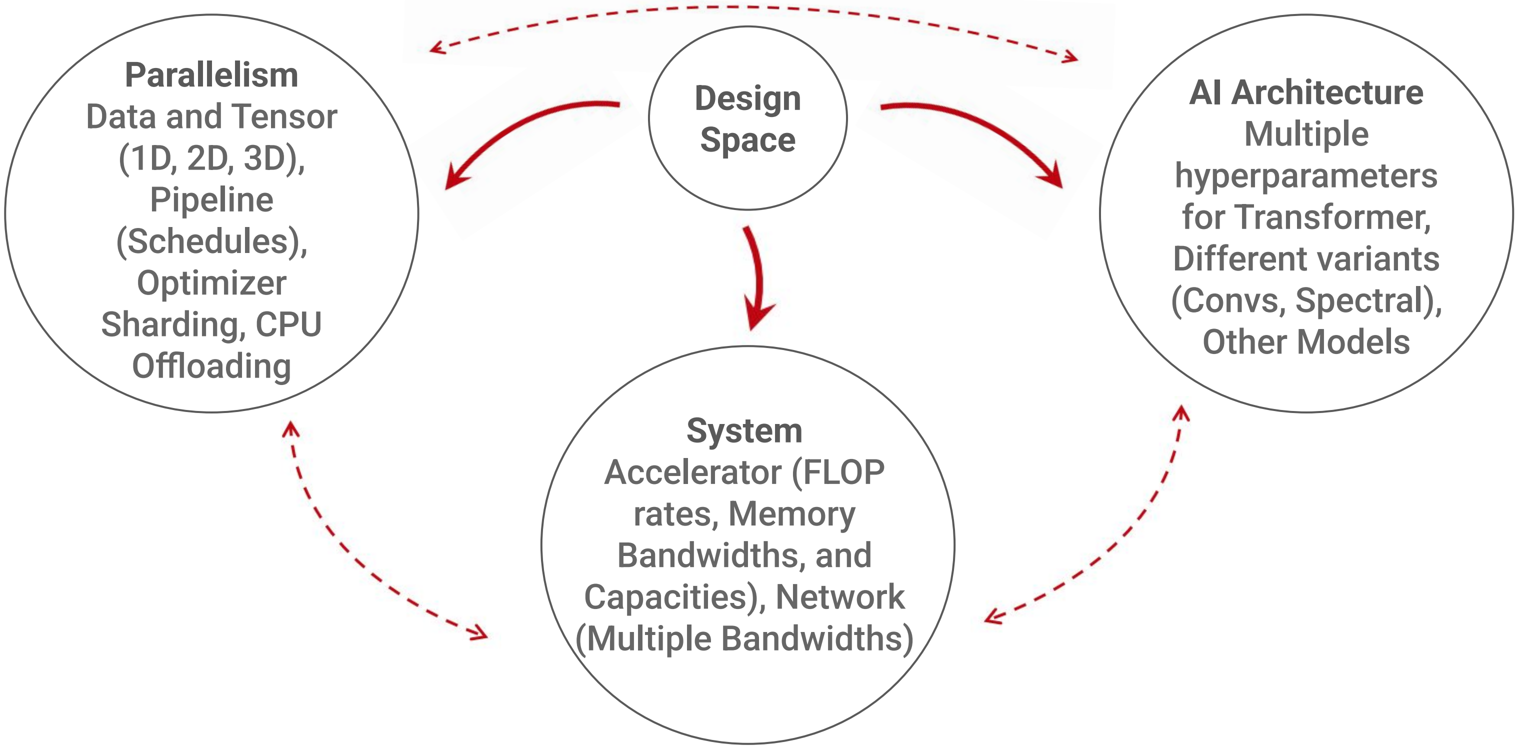
Design Space

Parallelism
Data and Tensor (1D, 2D, 3D), Pipeline (Schedules), Optimizer Sharding, CPU Offloading

System
Accelerator (FLOP rates, Memory Bandwidths, and Capacities), Network (Multiple Bandwidths)

AI Architecture
Multiple hyperparameters for Transformer, Different variants (Convs, Spectral), Other Models

AI Performance Modeling is Challenging



Analytical and Parameterized Models can be Valuable

AMPeD: An Analytical Model for Performance in Distributed Training of Transformers, ISPASS23
Calculon. A Methodology and Tool for High-Level Co-Design of Systems and Large Language Models. SC23
Comprehensive Performance Modeling and System Design Insights for Foundation Models, PMBS, SC24, Github

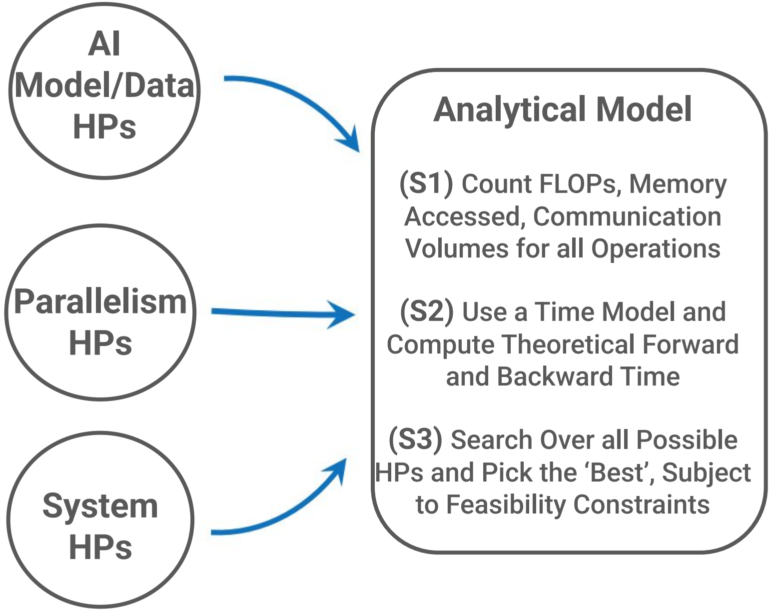
Analytical and Parameterized Models can be Valuable

AI Model/Data HPs

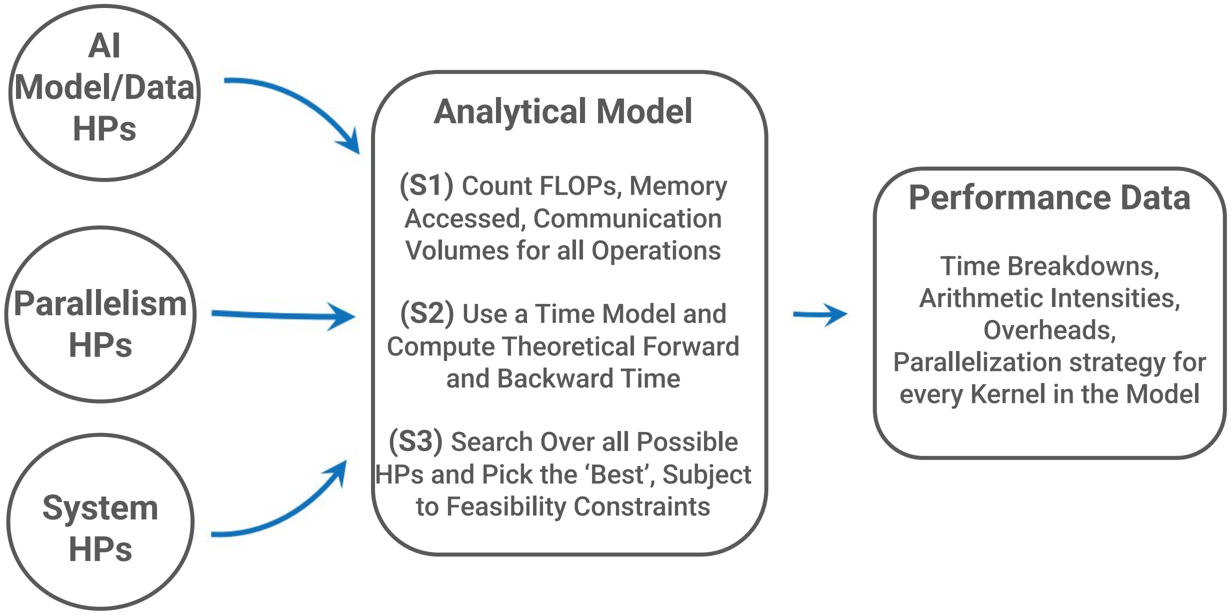
Parallelism HPs

System HPs

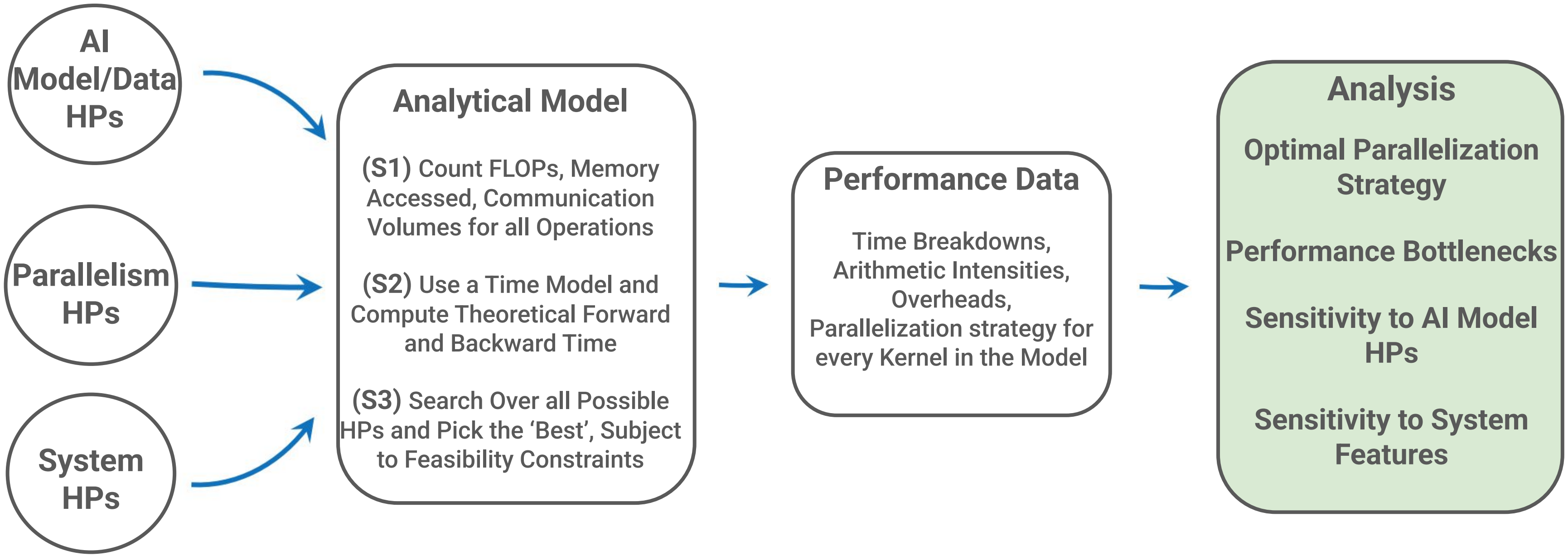
Analytical and Parameterized Models can be Valuable



Analytical and Parameterized Models can be Valuable



Analytical and Parameterized Models can be Valuable



AMPeD: An Analytical Model for Performance in Distributed Training of Transformers, ISPASS23

Calculon. A Methodology and Tool for High-Level Co-Design of Systems and Large Language Models. SC23

Comprehensive Performance Modeling and System Design Insights for Foundation Models, PMBS, SC24, Github

Analyze Varying Needs for Transformers in Science

- Counting FLOPs, communication volume is dependent on the parallelism
- Long sequence lengths may necessitate N-D parallelism

Operation	Partitioned Tensor Shapes	Type	Vol
	1D TP over n_t GPUs		
	SA		
$\tilde{\mathbf{X}} = \mathrm{LN}(\mathbf{X})$	$\tilde{\mathbf{X}} : (b, l, e)$, $\mathbf{X} : (b, \frac{l}{n_t}, e)$,	$\mathcal{AG}$	ble
$\mathbf{Q} = \tilde{\mathbf{X}}\mathbf{W_Q}$	$\mathbf{Q} : (b, \frac{h}{n_t}, l, e_h)$, $\mathbf{W_Q} : (e, \frac{e}{n_t})$,	-	0
$\mathbf{A} = \mathbf{QK}^T$	$\mathbf{A} : (b, \frac{h}{n_t}, l, l)$, $\mathbf{K} : (b, \frac{h}{n_t}, l, e_h)$	-	0
$\mathbf{S} = \mathbf{AV}$	$\mathbf{S} : (b, \frac{h}{n_t}, l, e_h)$, $\mathbf{V} : (b, \frac{h}{n_t}, l, e_h)$	-	0
$\mathbf{Y} = \mathbf{SW_P}$	$\mathbf{Y} : (b, \frac{l}{n_t}, e)$, $\mathbf{W_P} : (\frac{e}{n_t}, e)$	$\mathcal{RS}$	ble
	MLP		
$\tilde{\mathbf{Y}} = \mathrm{LN}(\mathbf{Y})$	$\tilde{\mathbf{Y}} : (b, l, e)$, $\mathbf{Y} : (b, \frac{l}{n_t}, e)$,	$\mathcal{AG}$	ble
$\mathbf{Z} = \tilde{\mathbf{Y}}\mathbf{W_1}$	$\mathbf{Z} : (b, l, f/n_t)$, $\mathbf{W_1} : (e, \frac{f}{n_t})$	-	0
$\mathbf{X} = \mathbf{ZW_2}$	$\mathbf{X} : (b, \frac{l}{n_t}, e)$, $\mathbf{W_2} : (\frac{f}{n_t}, e)$	$\mathcal{RS}$	ble

Analyze Varying Needs for Transformers in Science

- Counting FLOPs, communication volume is dependent on the parallelism
- Long sequence lengths may necessitate N-D parallelism

Operation	Partitioned Tensor Shapes	Type	Vol
	1D TP over n_t GPUs		
	SA		
$\tilde{\mathbf{X}} = \mathrm{LN}(\mathbf{X})$	$\tilde{\mathbf{X}} : (b, l, e)$, $\mathbf{X} : (b, \frac{l}{n_t}, e)$,	$\mathcal{AG}$	ble
$\mathbf{Q} = \tilde{\mathbf{X}}\mathbf{W_Q}$	$\mathbf{Q} : (b, \frac{h}{n_t}, l, e_h)$, $\mathbf{W_Q} : (e, \frac{e}{n_t})$,	-	0
$\mathbf{A} = \mathbf{Q}\mathbf{K}^T$	$\mathbf{A} : (b, \frac{h}{n_t}, l, l)$, $\mathbf{K} : (b, \frac{h}{n_t}, l, e_h)$	-	0
$\mathbf{S} = \mathbf{A}\mathbf{V}$	$\mathbf{S} : (b, \frac{h}{n_t}, l, e_h)$, $\mathbf{V} : (b, \frac{h}{n_t}, l, e_h)$	-	0
$\mathbf{Y} = \mathbf{S}\mathbf{W_p}$	$\mathbf{Y} : (b, \frac{l}{n_t}, e)$, $\mathbf{W_p} : (\frac{e}{n_t}, e)$	$\mathcal{RS}$	ble
	MLP		
$\tilde{\mathbf{Y}} = \mathrm{LN}(\mathbf{Y})$	$\tilde{\mathbf{Y}} : (b, l, e)$, $\mathbf{Y} : (b, \frac{l}{n_t}, e)$,	$\mathcal{AG}$	ble
$\mathbf{Z} = \tilde{\mathbf{Y}}\mathbf{W_1}$	$\mathbf{Z} : (b, l, f/n_t)$, $\mathbf{W_1} : (e, \frac{f}{n_t})$	-	0
$\mathbf{X} = \mathbf{Z}\mathbf{W_2}$	$\mathbf{X} : (b, \frac{l}{n_t}, e)$, $\mathbf{W_2} : (\frac{f}{n_t}, e)$	$\mathcal{RS}$	ble

Analyze Varying Needs for Transformers in Science

- Counting FLOPs, communication volume is dependent on the parallelism
- Long sequence lengths may necessitate N-D (4D) parallelism

Operation	Partitioned Tensor Shapes	Type	Vol
	2D TP over $n_1 \times n_2$ grid of GPUs		
	SA		
$\tilde{\mathbf{X}} = \mathrm{LN}(\mathbf{X})$	$\tilde{\mathbf{X}} : (b, \frac{l}{n_2}, e)$, $\mathbf{X} : (b, \frac{l}{n_1 n_2}, e)$,	$\mathcal{AG}$	$b\frac{l}{n_2}e$
$\mathbf{Q} = \tilde{\mathbf{X}}\mathbf{W_Q}$	$\mathbf{Q} : (b, \frac{h}{n_1}, \frac{l}{n_2}, e_h)$, $\mathbf{W_Q} : (e, \frac{e}{n_1})$,	-	0
$\mathbf{A} = \mathbf{Q}\mathbf{K}^T$	$\mathbf{A} : (b, \frac{h}{n_1}, \frac{l}{n_2}, l)$, $\mathbf{K} : (b, \frac{h}{n_1}, l, e_h)$	$\mathcal{AG}$	$bl\frac{e}{n_1}$
$\mathbf{S} = \mathbf{AV}$	$\mathbf{S} : (b, \frac{h}{n_1}, \frac{l}{n_2}, e_h)$, $\mathbf{V} : (b, \frac{h}{n_1}, l, e_h)$	$\mathcal{AG}$	$bl\frac{e}{n_1}$
$\mathbf{Y} = \mathbf{SW_P}$	$\mathbf{Y} : (b, \frac{l}{n_1 n_2}, e)$, $\mathbf{W_P} : (\frac{e}{n_1}, e)$	$\mathcal{RS}$	$b\frac{l}{n_2}e$
	MLP		
$\tilde{\mathbf{Y}} = \mathrm{LN}(\mathbf{Y})$	$\tilde{\mathbf{Y}} : (b, \frac{l}{n_2}, e)$, $\mathbf{Y} : (b, \frac{l}{n_1 n_2}, e)$,	$\mathcal{AG}$	$b\frac{l}{n_2}e$
$\mathbf{Z} = \tilde{\mathbf{Y}}\mathbf{W_1}$	$\mathbf{Z} : (b, \frac{l}{n_2}, \frac{f}{n_1})$, $\mathbf{W_1} : (e, \frac{f}{n_1})$	-	0
$\mathbf{X} = \mathbf{ZW_2}$	$\mathbf{X} : (b, \frac{l}{n_1 n_2}, e)$, $\mathbf{W_2} : (\frac{f}{n_1}, e)$	$\mathcal{RS}$	$b\frac{l}{n_2}e$

Analyze Varying Needs for Transformers in Science

- Counting FLOPs, communication volume is dependent on the parallelism
- Long sequence lengths may necessitate N-D (4D) parallelism

Operation	Partitioned Tensor Shapes	Type	Vol
	2D TP over $n_1 \times n_2$ grid of GPUs		
	SA		
$\tilde{\mathbf{X}} = \mathrm{LN}(\mathbf{X})$	$\tilde{\mathbf{X}} : (b, \frac{l}{n_2}, e)$, $\mathbf{X} : (b, \frac{l}{n_1 n_2}, e)$,	$\mathcal{AG}$	$b\frac{l}{n_2}e$
$\mathbf{Q} = \tilde{\mathbf{X}}\mathbf{W_Q}$	$\mathbf{Q} : (b, \frac{h}{n_1}, \frac{l}{n_2}, e_h)$, $\mathbf{W_Q} : (e, \frac{e}{n_1})$,	-	0
$\mathbf{A} = \mathbf{Q}\mathbf{K}^T$	$\mathbf{A} : (b, \frac{h}{n_1}, \frac{l}{n_2}, l)$, $\mathbf{K} : (b, \frac{h}{n_1}, l, e_h)$	$\mathcal{AG}$	$bl\frac{e}{n_1}$
$\mathbf{S} = \mathbf{A}\mathbf{V}$	$\mathbf{S} : (b, \frac{h}{n_1}, \frac{l}{n_2}, e_h)$, $\mathbf{V} : (b, \frac{h}{n_1}, l, e_h)$	$\mathcal{AG}$	$bl\frac{e}{n_1}$
$\mathbf{Y} = \mathbf{S}\mathbf{W_P}$	$\mathbf{Y} : (b, \frac{l}{n_1 n_2}, e)$, $\mathbf{W_P} : (\frac{e}{n_1}, e)$	$\mathcal{RS}$	$b\frac{l}{n_2}e$
	MLP		
$\tilde{\mathbf{Y}} = \mathrm{LN}(\mathbf{Y})$	$\tilde{\mathbf{Y}} : (b, \frac{l}{n_2}, e)$, $\mathbf{Y} : (b, \frac{l}{n_1 n_2}, e)$,	$\mathcal{AG}$	$b\frac{l}{n_2}e$
$\mathbf{Z} = \tilde{\mathbf{Y}}\mathbf{W_1}$	$\mathbf{Z} : (b, \frac{l}{n_2}, \frac{f}{n_1})$, $\mathbf{W_1} : (e, \frac{f}{n_1})$	-	0
$\mathbf{X} = \mathbf{Z}\mathbf{W_2}$	$\mathbf{X} : (b, \frac{l}{n_1 n_2}, e)$, $\mathbf{W_2} : (\frac{f}{n_1}, e)$	$\mathcal{RS}$	$b\frac{l}{n_2}e$

Analyze Varying Needs for Transformers in Science

- Counting FLOPs, communication volume is dependent on the parallelism
- Long sequence lengths may necessitate N-D (4D) parallelism

Operation	Partitioned Tensor Shapes	Type	Vol
	2D TP over $n_1 \times n_2$ grid of GPUs		
	SA		
$\tilde{\mathbf{X}} = \text{LN}(\mathbf{X})$	$\tilde{\mathbf{X}} : (b, \frac{l}{n_2}, e)$, $\mathbf{X} : (b, \frac{l}{n_1 n_2}, e)$,	$\mathcal{AG}$	$b\frac{l}{n_2}e$
$\mathbf{Q} = \tilde{\mathbf{X}}\mathbf{W_Q}$	$\mathbf{Q} : (b, \frac{h}{n_1}, \frac{l}{n_2}, e_h)$, $\mathbf{W_Q} : (e, \frac{e}{n_1})$,	-	0
$\mathbf{A} = \mathbf{Q}\mathbf{K}^T$	$\mathbf{A} : (b, \frac{h}{n_1}, \frac{l}{n_2}, l)$, $\mathbf{K} : (b, \frac{h}{n_1}, l, e_h)$	$\mathcal{AG}$	$bl\frac{e}{n_1}$
$\mathbf{S} = \mathbf{A}\mathbf{V}$	$\mathbf{S} : (b, \frac{h}{n_1}, \frac{l}{n_2}, e_h)$, $\mathbf{V} : (b, \frac{h}{n_1}, l, e_h)$	$\mathcal{AG}$	$bl\frac{e}{n_1}$
$\mathbf{Y} = \mathbf{S}\mathbf{W_P}$	$\mathbf{Y} : (b, \frac{l}{n_1 n_2}, e)$, $\mathbf{W_P} : (\frac{e}{n_1}, e)$	$\mathcal{RS}$	$b\frac{l}{n_2}e$
	MLP		
$\tilde{\mathbf{Y}} = \text{LN}(\mathbf{Y})$	$\tilde{\mathbf{Y}} : (b, \frac{l}{n_2}, e)$, $\mathbf{Y} : (b, \frac{l}{n_1 n_2}, e)$,	$\mathcal{AG}$	$b\frac{l}{n_2}e$
$\mathbf{Z} = \tilde{\mathbf{Y}}\mathbf{W_1}$	$\mathbf{Z} : (b, \frac{l}{n_2}, \frac{f}{n_1})$, $\mathbf{W_1} : (e, \frac{f}{n_1})$	-	0
$\mathbf{X} = \mathbf{Z}\mathbf{W_2}$	$\mathbf{X} : (b, \frac{l}{n_1 n_2}, e)$, $\mathbf{W_2} : (\frac{f}{n_1}, e)$	$\mathcal{RS}$	$b\frac{l}{n_2}e$

Analyze Varying Needs for Transformers in Science

- Long sequence lengths may necessitate 4D parallelism
- Different choices for Matrix Multiplies: SUMMA also possible

$$\mathbf{C} = \sum_{\kappa}^{n_b - 1} \mathbf{A}^{\kappa} \mathbf{B}^{\kappa}$$

Algorithm 1 $\mathbf{C} = \mathbf{AB}$ using SUMMA

Input: $\mathbf{A}_{ij}$, $\mathbf{B}_{ij}$
Output: $\mathbf{C}_{ij}$
$\mathbf{C} = 0$
for $\kappa = 0 \to n_b - 1$ **do**
    for $i = 0, ..., n_1 - 1$ Broadcast $\mathbf{A}_i^{\kappa}$ to i^{th} process row
    for $j = 0, ..., n_2 - 1$ Broadcast $\mathbf{B}_j^{\kappa}$ to j^{th} process col
    $\mathbf{C}_{ij} = \mathbf{C}_{ij} + \mathbf{A}_i^{\kappa} \mathbf{B}_j^{\kappa}$
end for
return $\mathbf{C}_{ij}$

SUMMA: Scalable Universal Matrix Multiplication Algorithm, Link

Analyze Varying Needs for Transformers in Science

- Long sequence lengths may necessitate 4D parallelism
- Different choices for Matrix Multiplies: SUMMA also possible

Operation	Partitioned Tensor Shapes	Type	Vol
2D TP with SUMMA over $n_1 \times n_2$ grid of GPUs			
SA			
$\tilde{\mathbf{X}} = \mathrm{LN}(\mathbf{X})$	$\tilde{\mathbf{X}} : (b, \frac{l}{n_2}, \frac{e}{n_1}), \mathbf{X} : (b, \frac{l}{n_2}, \frac{e}{n_1}),$	$\mathcal{AR}$	$b\frac{l}{n_2}e$
$\mathbf{Q} = \tilde{\mathbf{X}}\mathbf{W_Q}$	$\mathbf{Q} : (b, \frac{h}{n_1}, \frac{l}{n_2}, e_h), \mathbf{W_Q} : (\frac{e}{n_2}, \frac{e}{n_1}),$	$\mathcal{B}$	V_1
$\mathbf{A} = \mathbf{Q}\mathbf{K}^T$	$\mathbf{A} : (b, \frac{h}{n_1}, \frac{l}{n_2}, l), \mathbf{K} : (b, \frac{h}{n_1}, l, e_h)$	$\mathcal{AG}$	$bl\frac{e}{n_1}$
$\mathbf{S} = \mathbf{A}\mathbf{V}$	$\mathbf{S} : (b, \frac{h}{n_1}, \frac{l}{n_2}, e_h), \mathbf{V} : (b, \frac{h}{n_1}, l, e_h)$	$\mathcal{AG}$	$bl\frac{e}{n_1}$
$\mathbf{Y} = \mathbf{S}\mathbf{W_p}$	$\mathbf{Y} : (b, \frac{l}{n_1 n_2}, e), \mathbf{W_p} : (\frac{e}{n_1}, e)$	$\mathcal{RS}$	$b\frac{l}{n_2}e$
MLP			
$\tilde{\mathbf{Y}} = \mathrm{LN}(\mathbf{Y})$	$\tilde{\mathbf{Y}} : (b, \frac{l}{n_2}, \frac{e}{n_1}), \mathbf{Y} : (b, \frac{l}{n_2}, \frac{e}{n_1}),$	$\mathcal{AR}$	$b\frac{l}{n_2}e$
$\mathbf{Z} = \tilde{\mathbf{Y}}\mathbf{W_1}$	$\mathbf{Z} : (b, \frac{l}{n_2}, \frac{f}{n_1}), \mathbf{W_1} : (\frac{e}{n_2}, \frac{f}{n_1})$	$\mathcal{B}$	V_2
$\mathbf{X} = \mathbf{Z}\mathbf{W_2}$	$\mathbf{X} : (b, \frac{l}{n_2}, \frac{e}{n_1}), \mathbf{W_2} : (\frac{f}{n_2}, \frac{e}{n_1})$	$\mathcal{B}$	V_3

$$\boxed{V_1 = ble/n_2 + e^2/n_1}$$

NERSC | U.S. DEPARTMENT OF ENERGY | Office of Science | BERKELEY LAB

SUMMA: Scalable Universal Matrix Multiplication Algorithm, Link
Comprehensive Performance Modeling and System Design Insights for Foundation Models, PMBS, SC24, Github

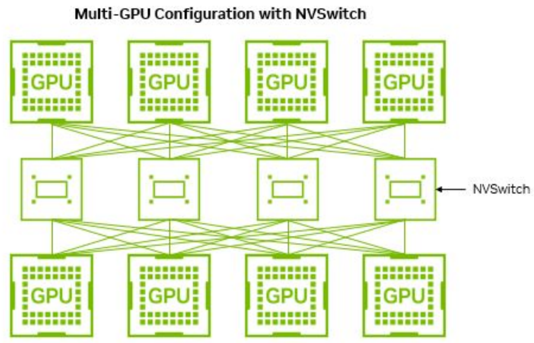
Two Transformer Variants on Different Systems

- Large GPT3 (1T, 2K) on a few trillion tokens
- Large ViT (80B, 250K) on decades of weather data

Two Transformer Variants on Different Systems

- Large GPT3 (1T, 2K) on a few trillion tokens
- Large ViT (80B, 250K) on decades of weather data

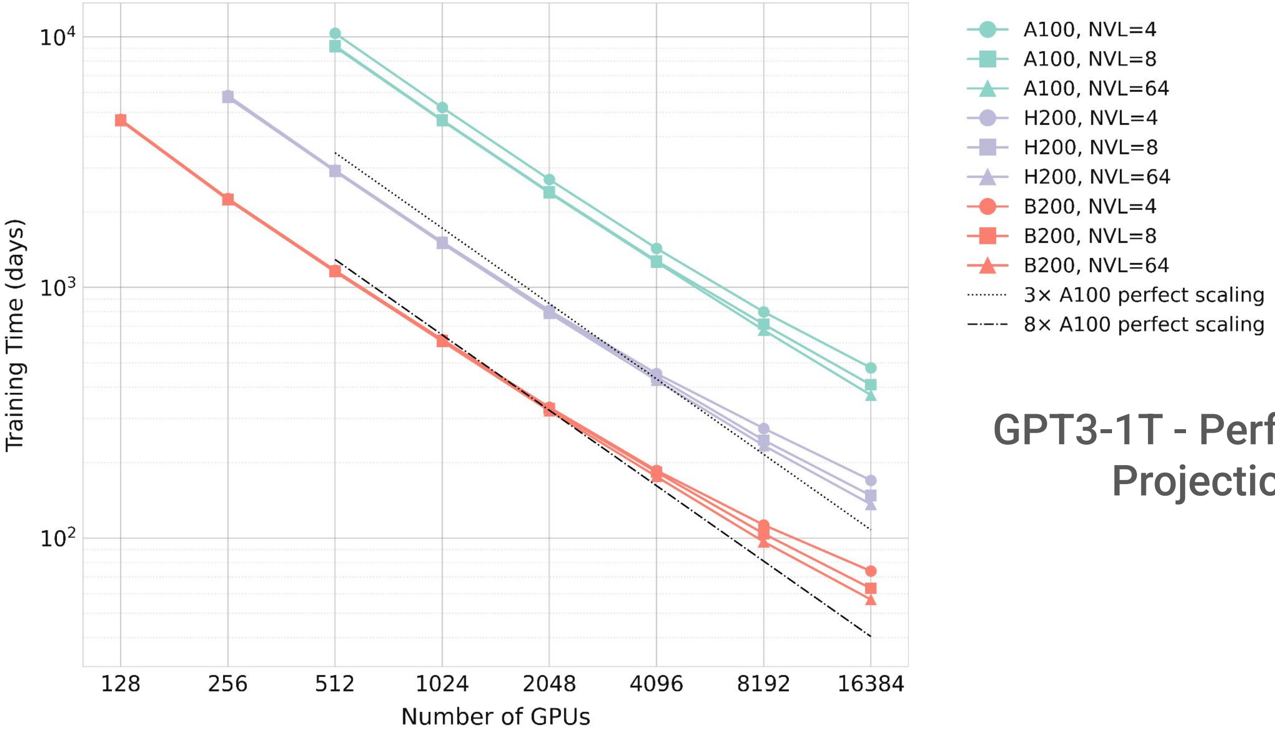
- Three NVIDIA GPU generations: A100, H200, B200
- Three NVLINK/NVL through NVSWITCH domain sizes: 4, 8, 64



NVSWITCH

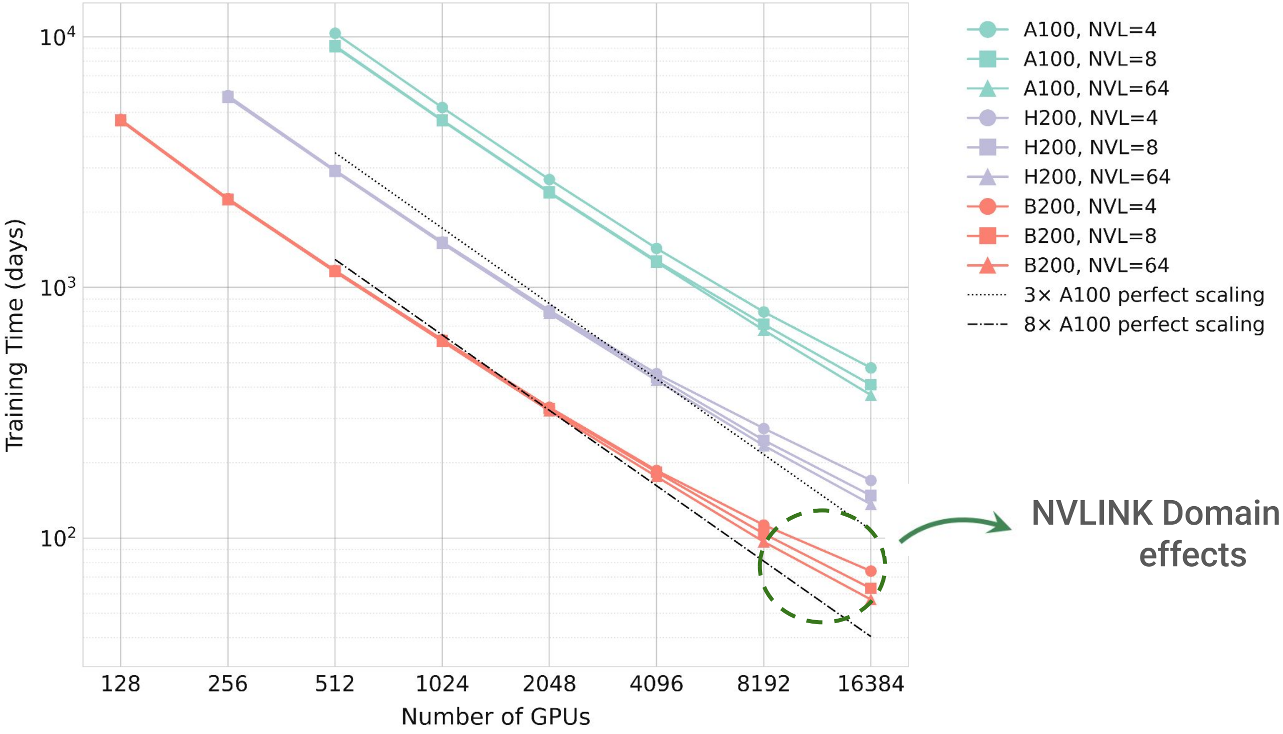
Provides a High-level View of Scaling Behavior

Lower is better



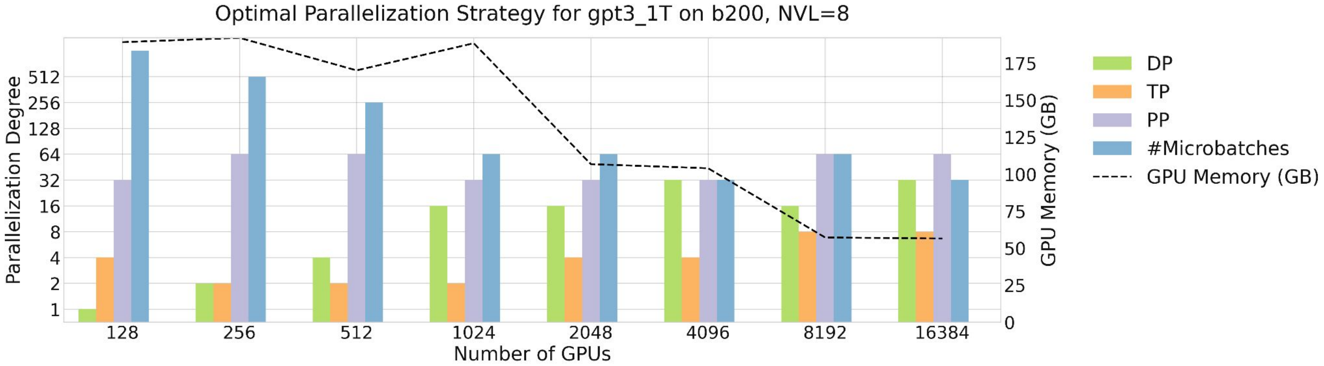
GPT3-1T - Performance Projections

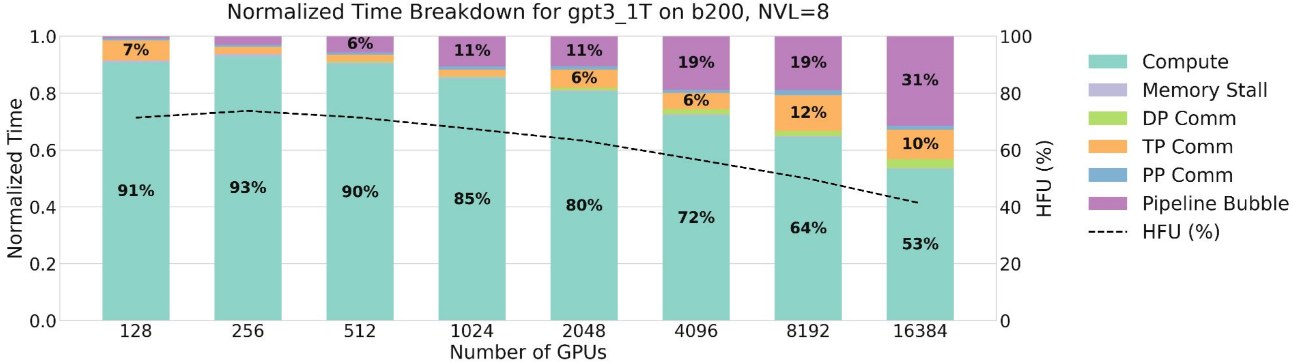
Provides a High-level View of Scaling Behavior



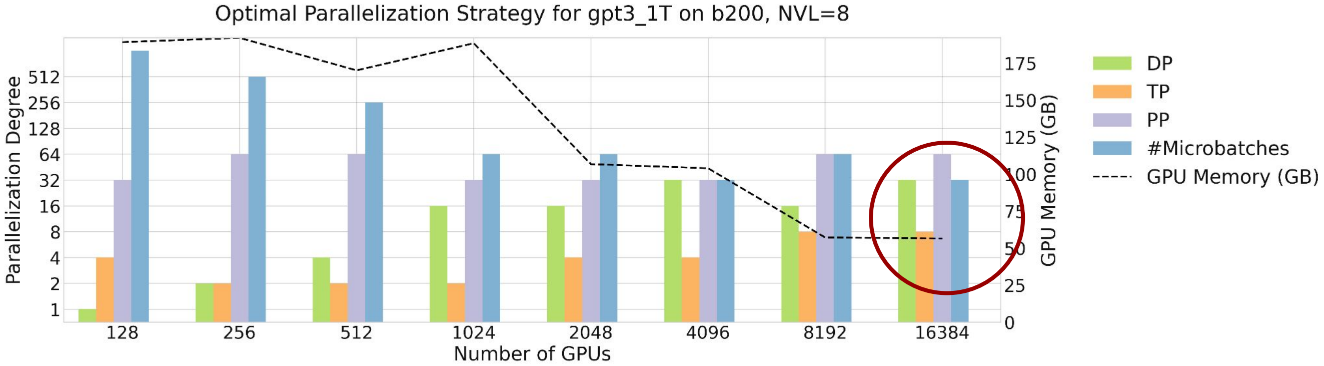
NVLINK Domain Size effects

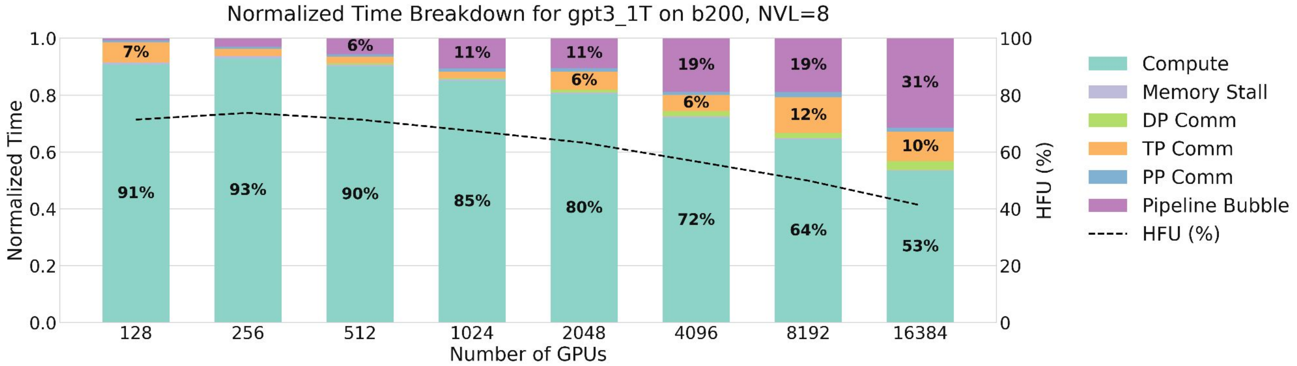
Exposes Bottlenecks and Optimal Parallelism



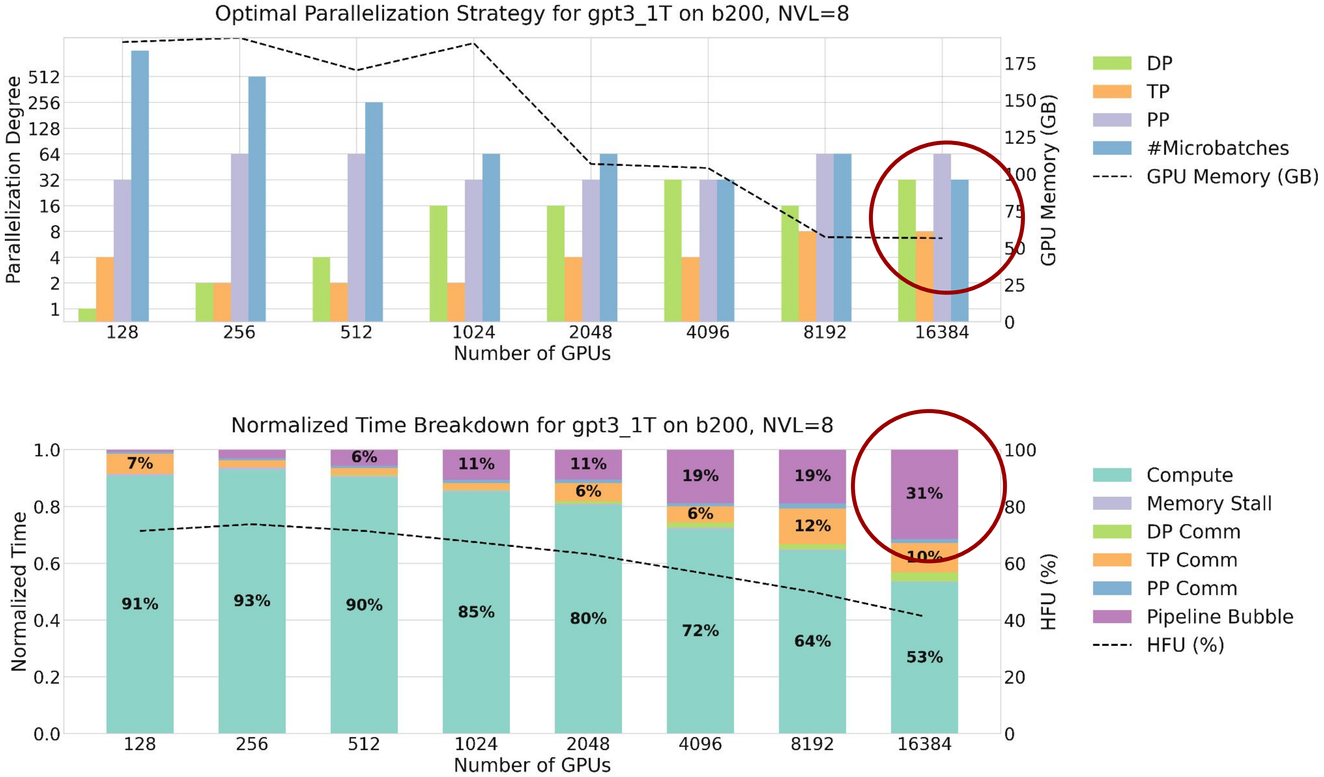


Exposes Bottlenecks and Optimal Parallelism



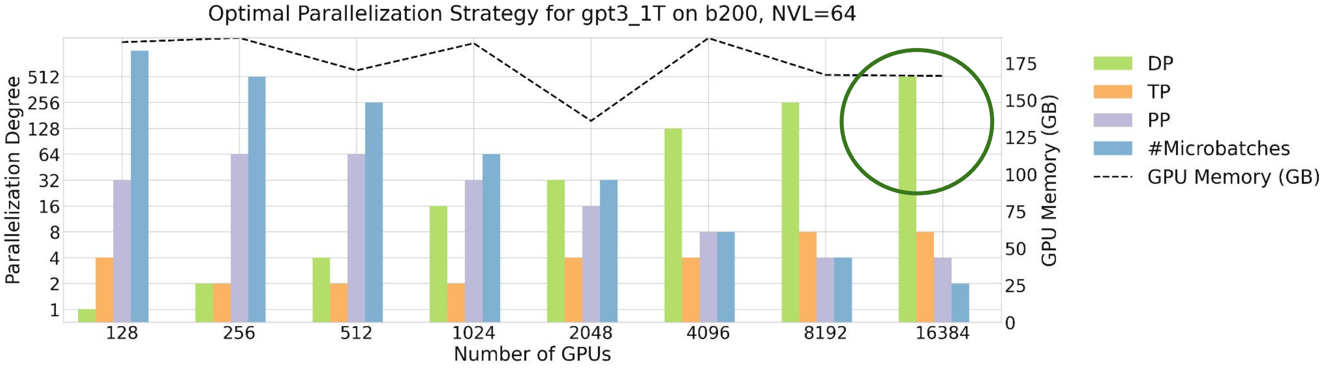


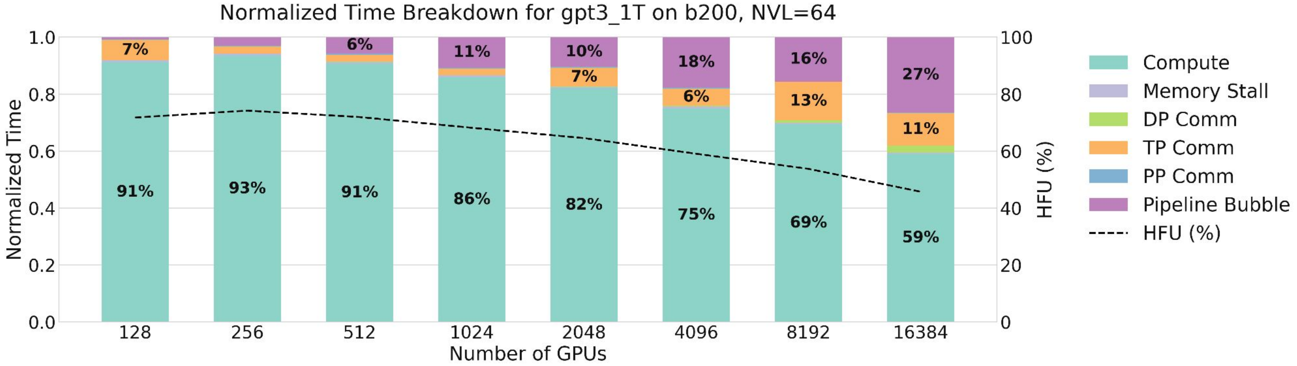
Exposes Bottlenecks and Optimal Parallelism



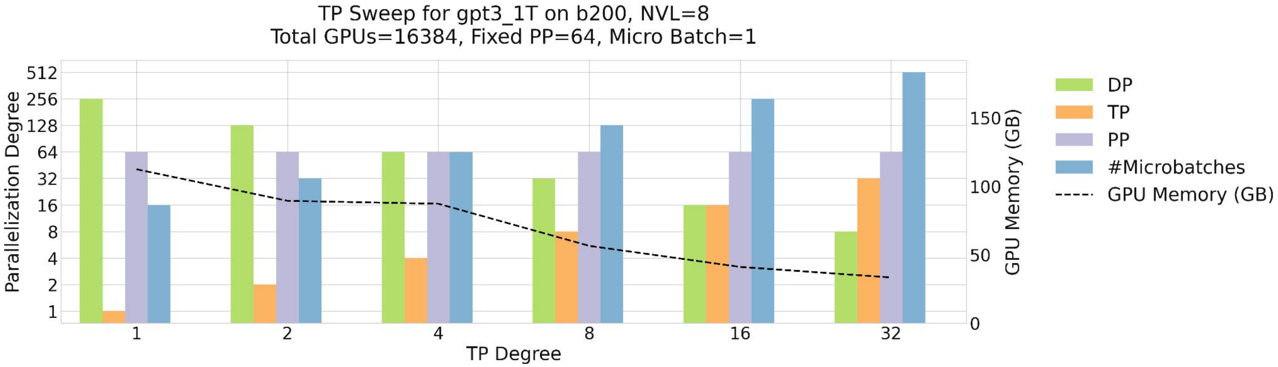
NERSC U.S. DEPARTMENT OF ENERGY Office of Science BERKELEY LAB

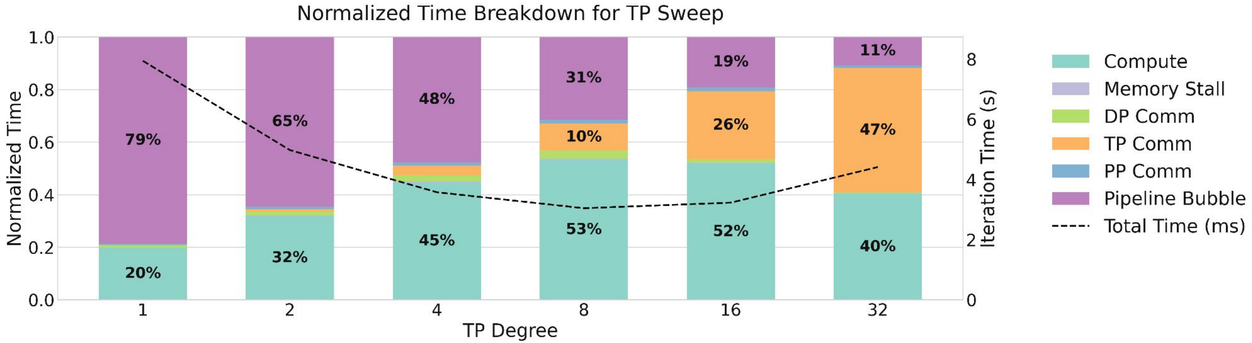
Larger NVLINKs Favor High Data Parallelism





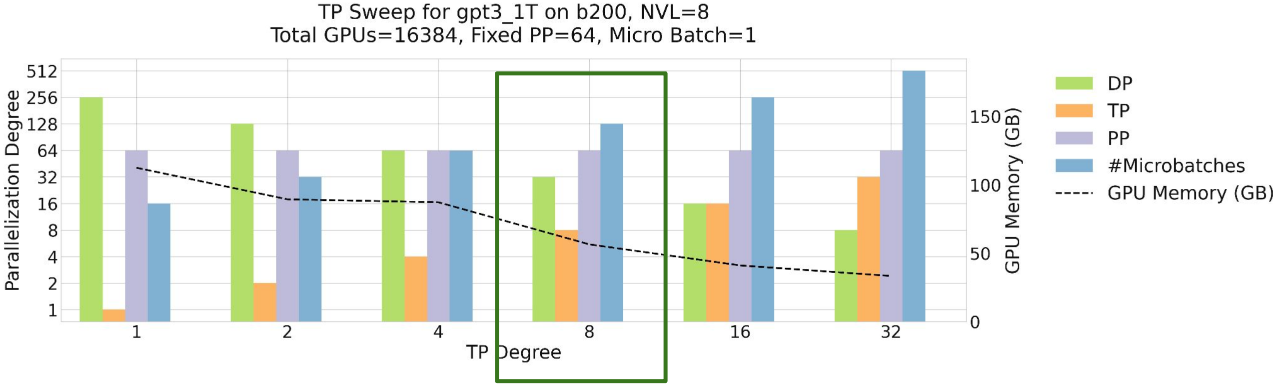
Probe the Model to Get Deeper Insights

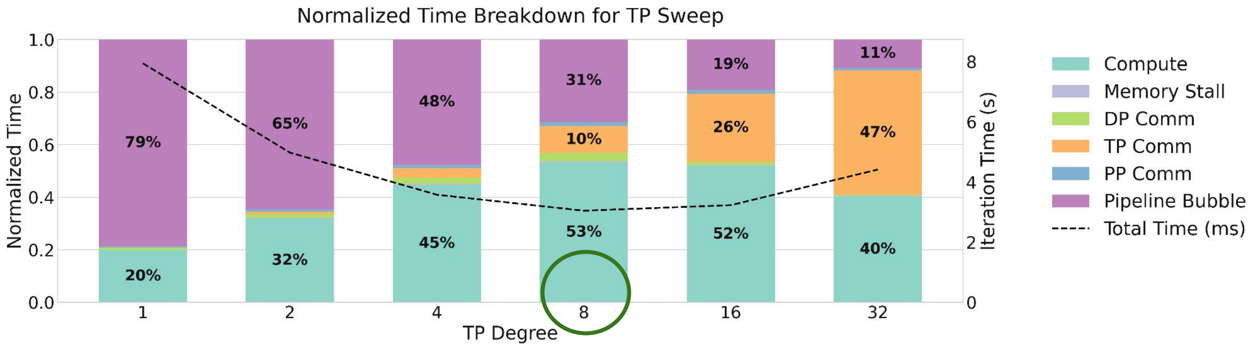




Fix #GPUs and look around the optimal configuration

Probe the Model to Get Deeper Insights

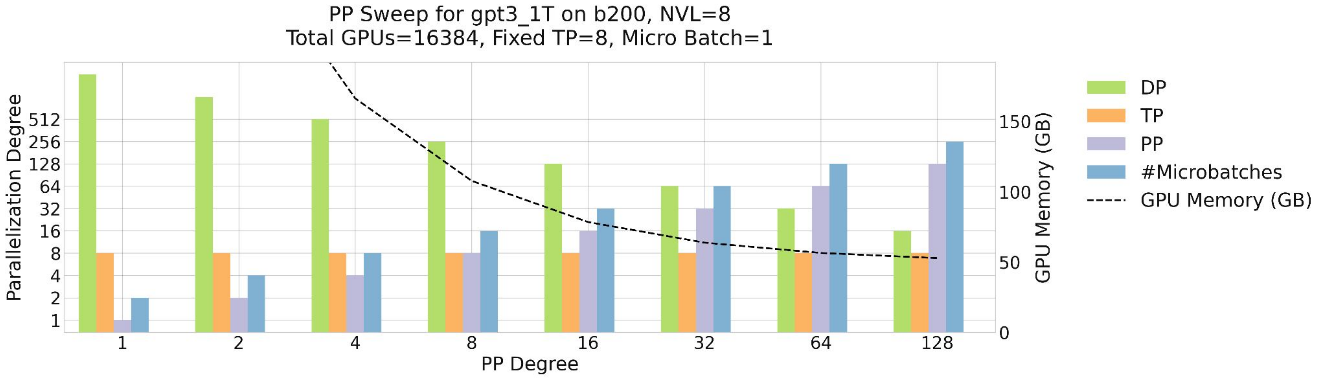


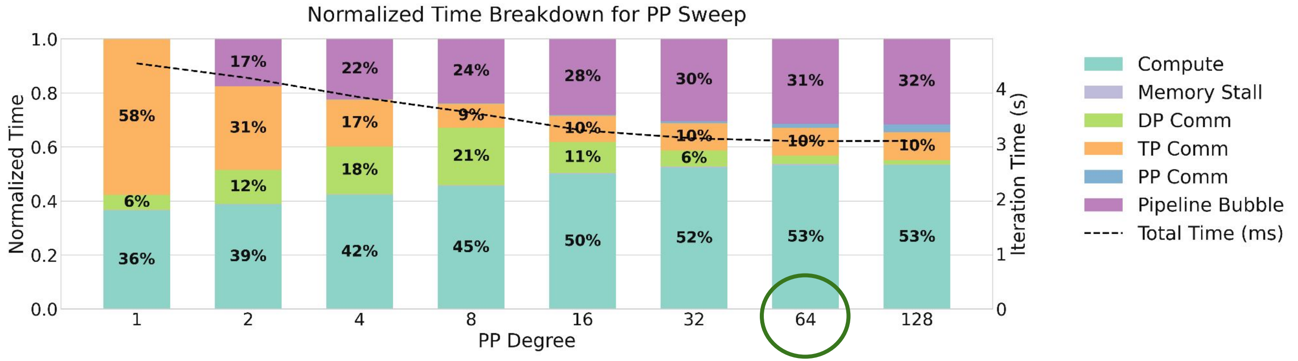


NERSC U.S. DEPARTMENT OF ENERGY Office of Science BERKELEY LAB

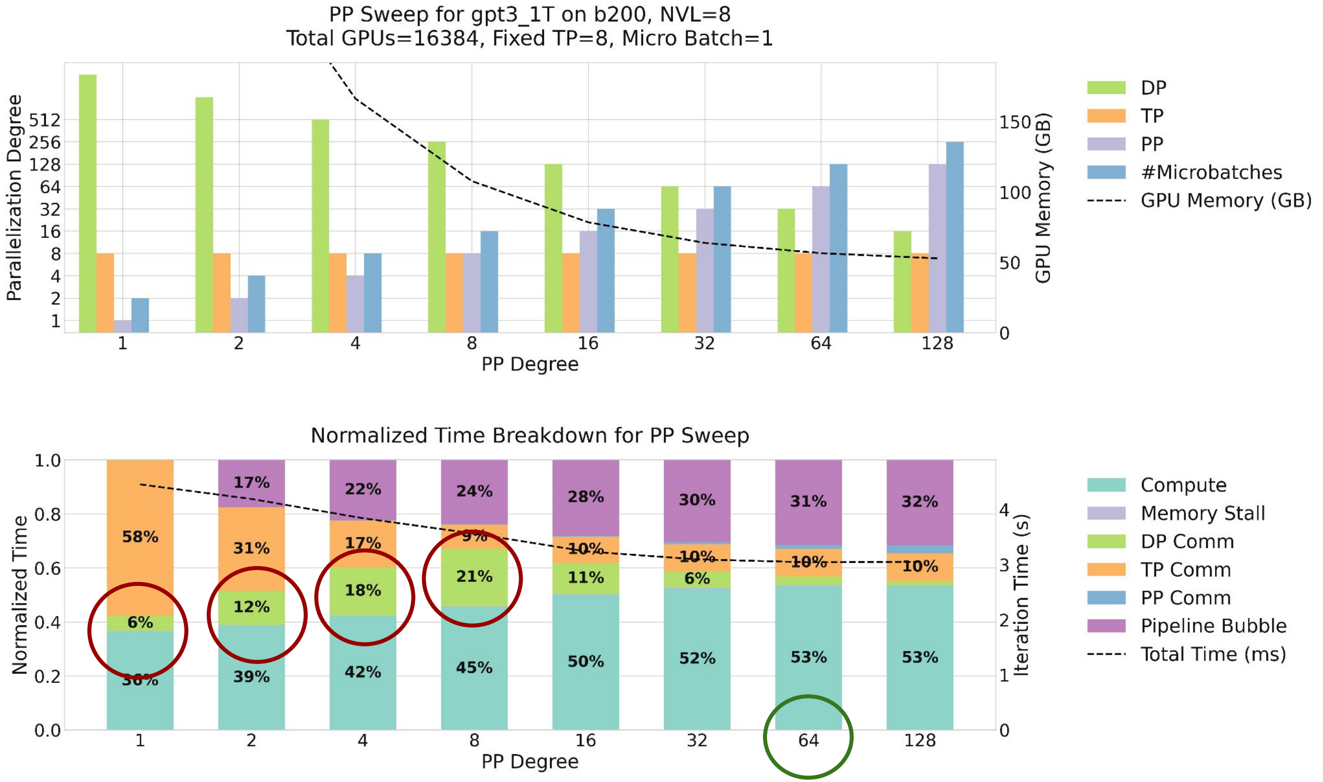
Fix #GPUs and look around the optimal configuration

Placement of GPUs Matters

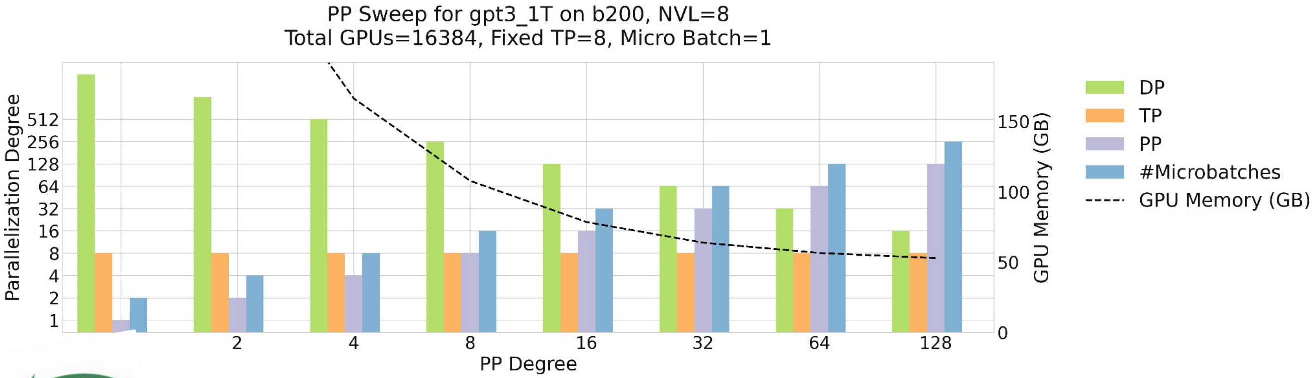




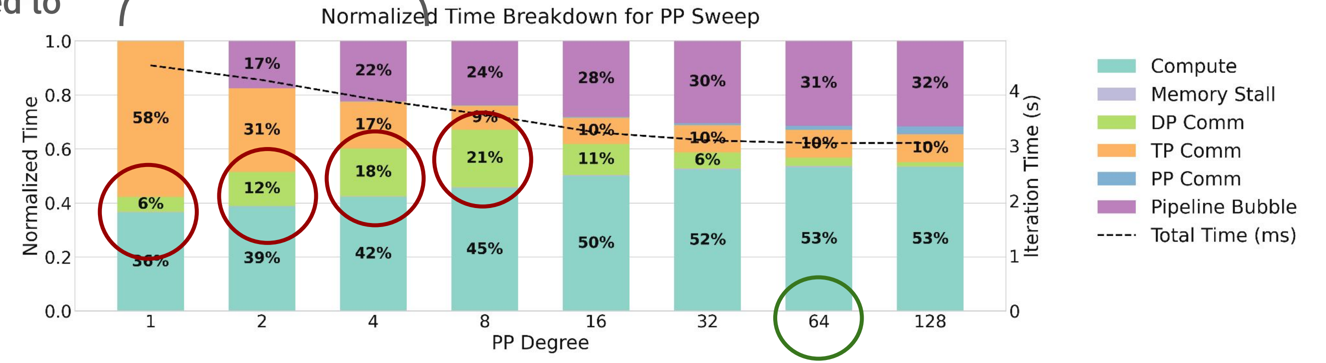
Placement of GPUs Matters



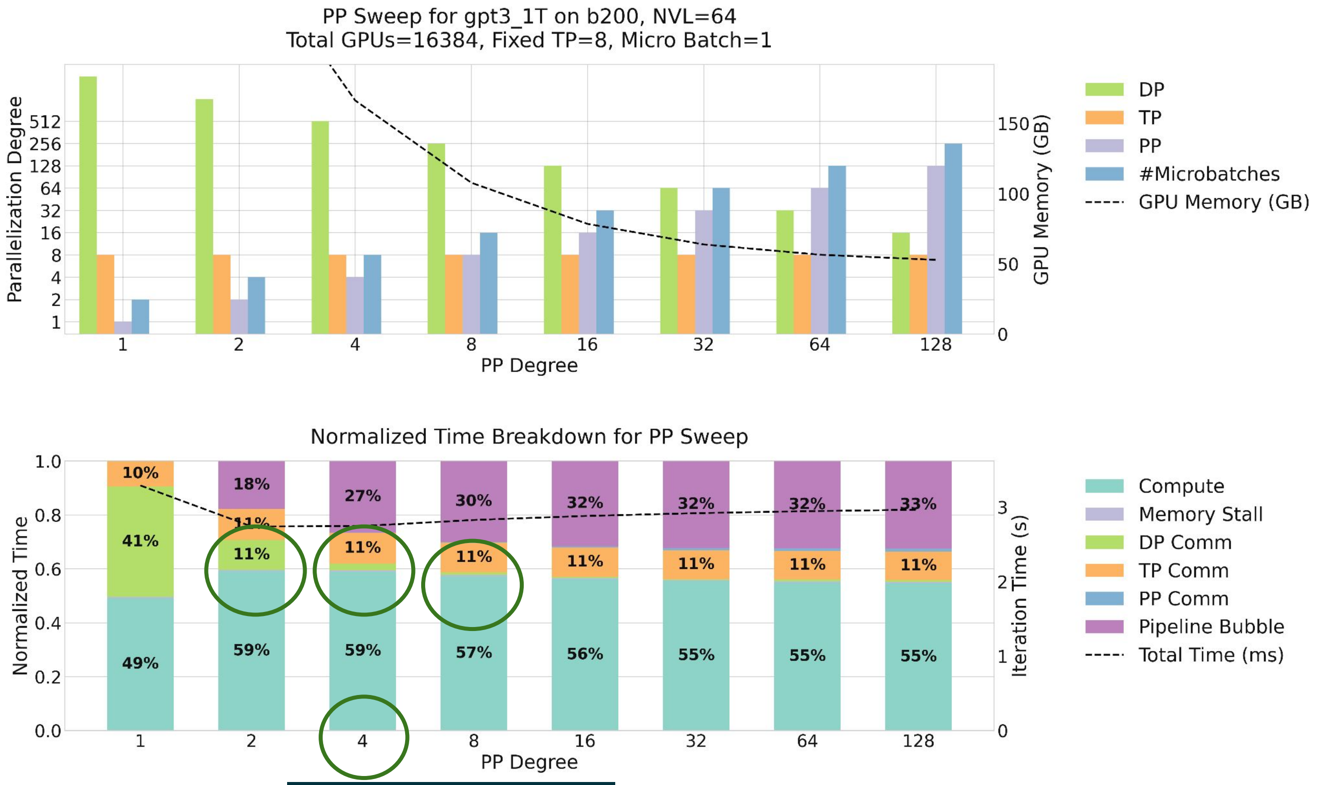
Placement of GPUs Matters



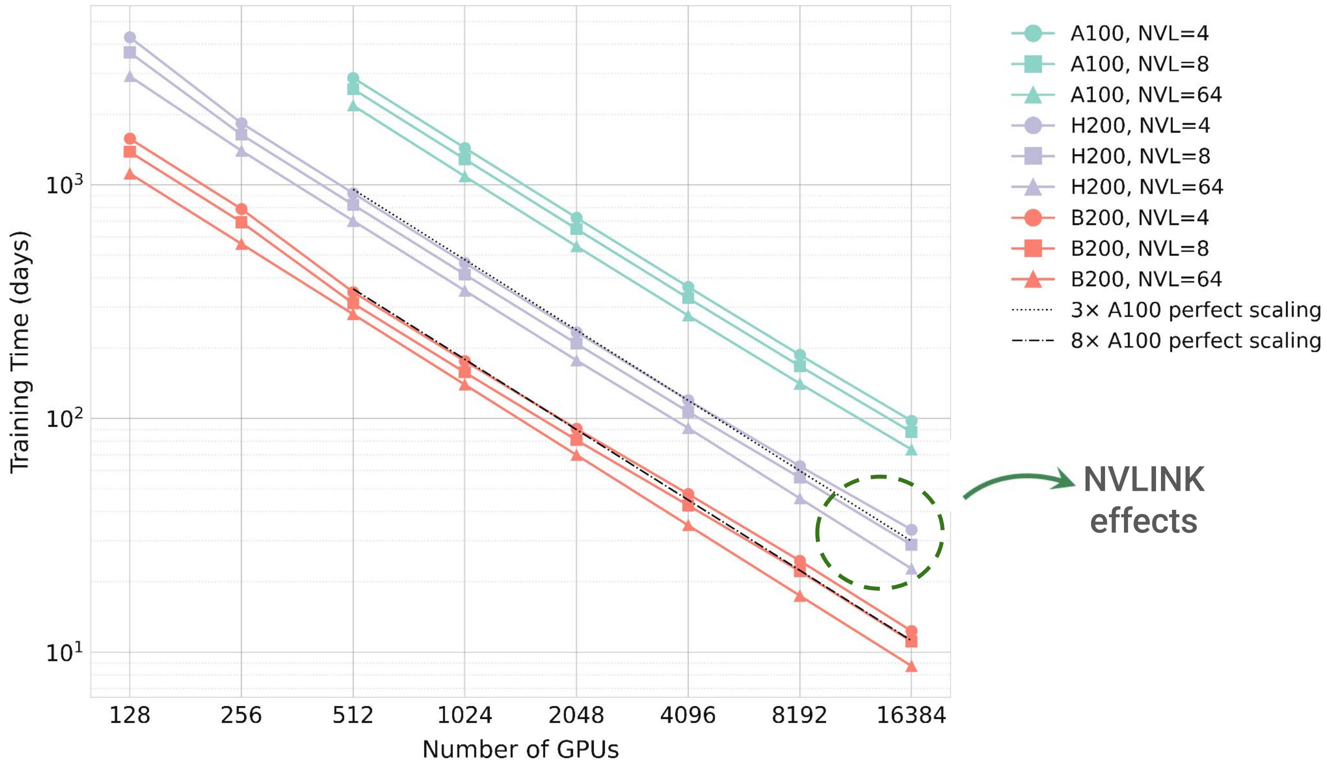
DP GPUs allocated to NVLINK



Placement of GPUs Favor Data Parallelism for Large NVL

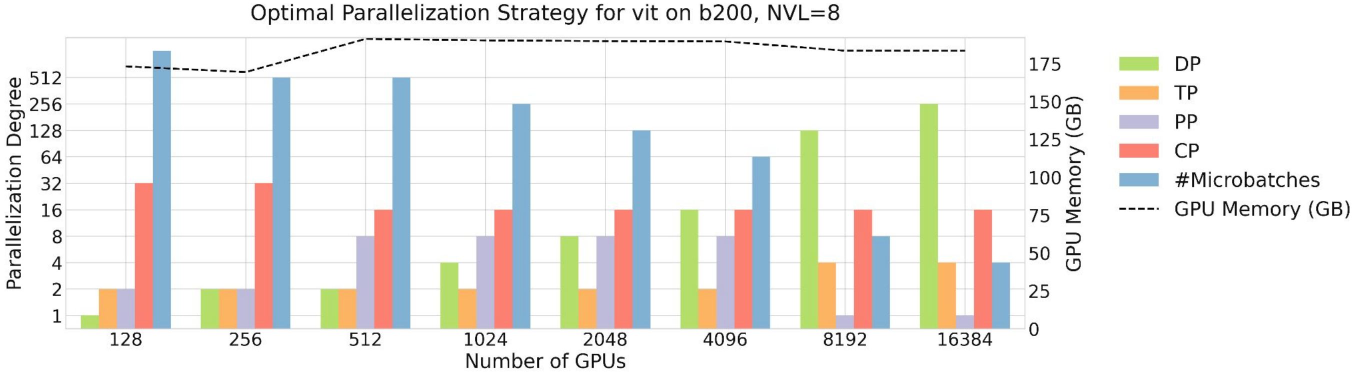


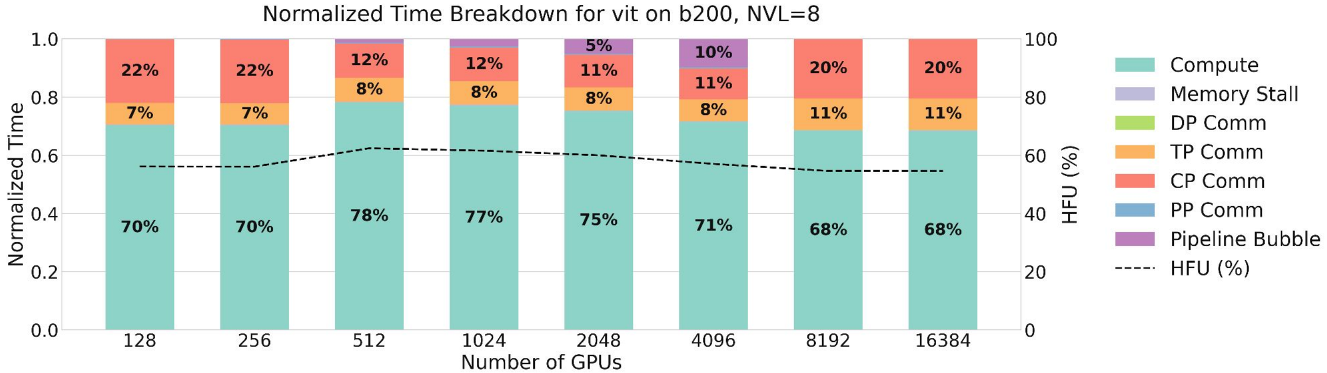
Transformer in Science is More Sensitive to the Network



NERSC U.S. DEPARTMENT OF ENERGY Office of Science BERKELEY LAB

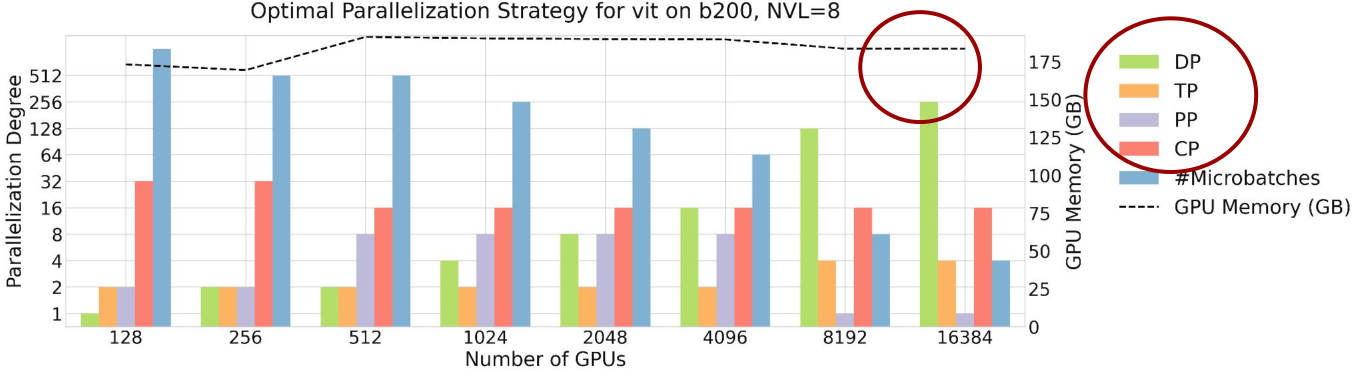
Long Sequences Need 4D Parallelism

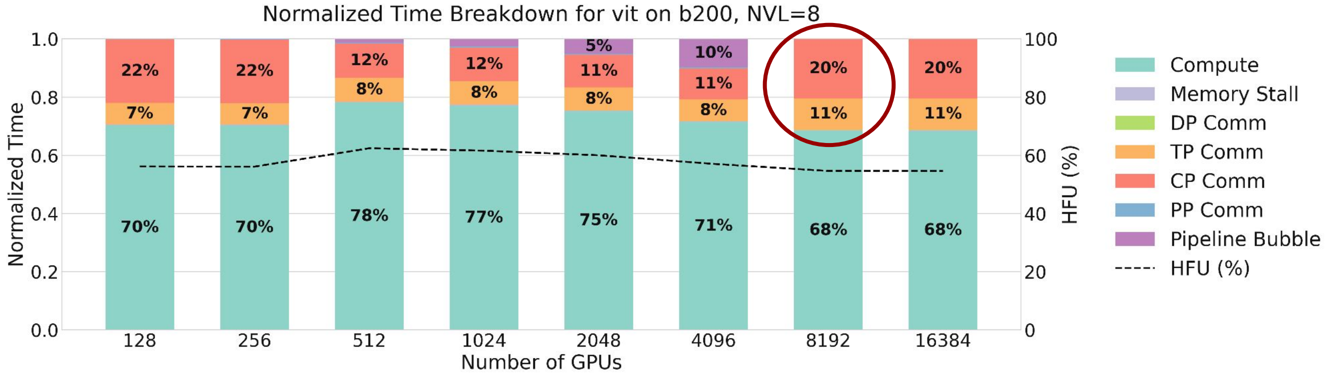




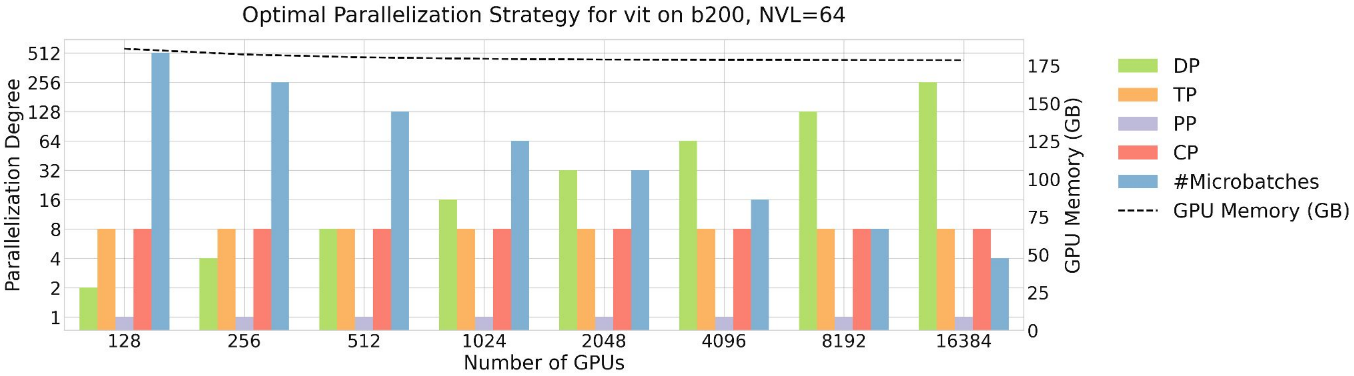
NERSC U.S. DEPARTMENT OF ENERGY Office of Science BERKELEY LAB

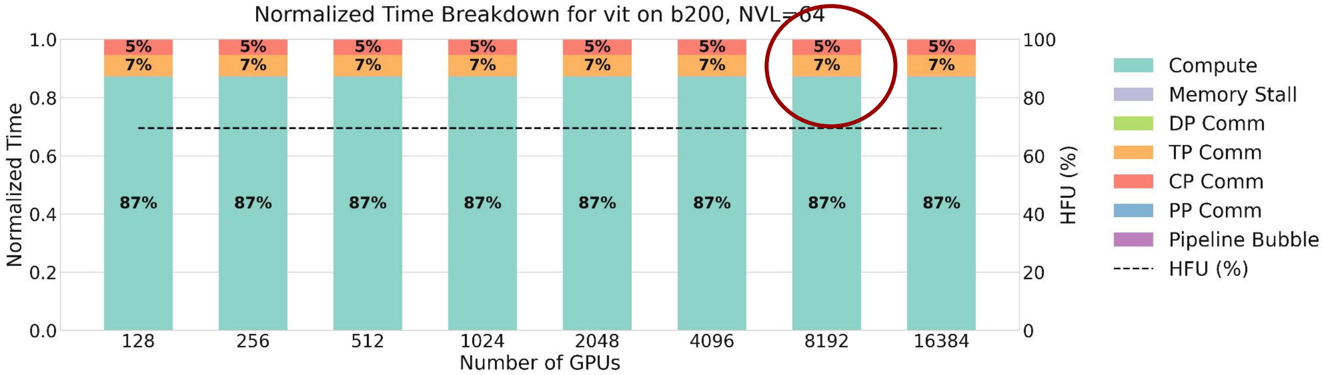
Long Sequences Need 4D Parallelism





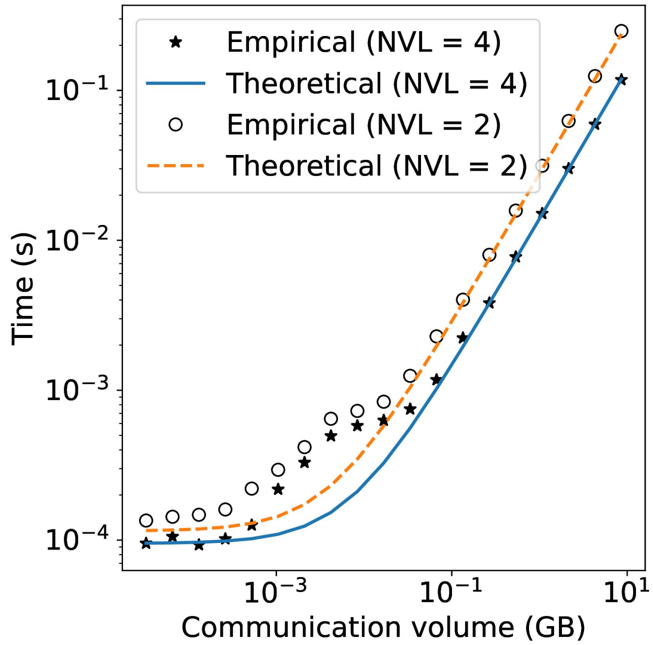
Larger NVLINK Drops Communication Costs





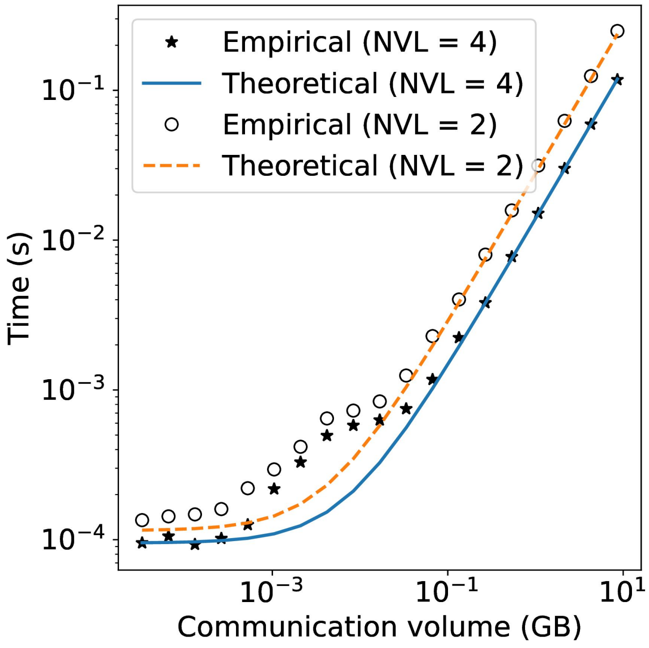
Validation with Megatron-LM

- Validated time models on the Perlmutter supercomputer
    - 4-way NVLINK domain



Validation with Megatron-LM

- Validated time models on the Perlmutter supercomputer
  - 4-way NVLINK domain
- Validated throughput numbers on 512 GPUs
  - GPT3 (175B) and ViT (32K)
- ~10% errors in iteration time
  - Controlled GPU placement with Megatron flags
  - Overlap flags, *FlashAttention*, other optimizations in sync with model
  - Validated sub-optimal configurations as well
- SUMMA validation challenging
  - ColossalAI in future work

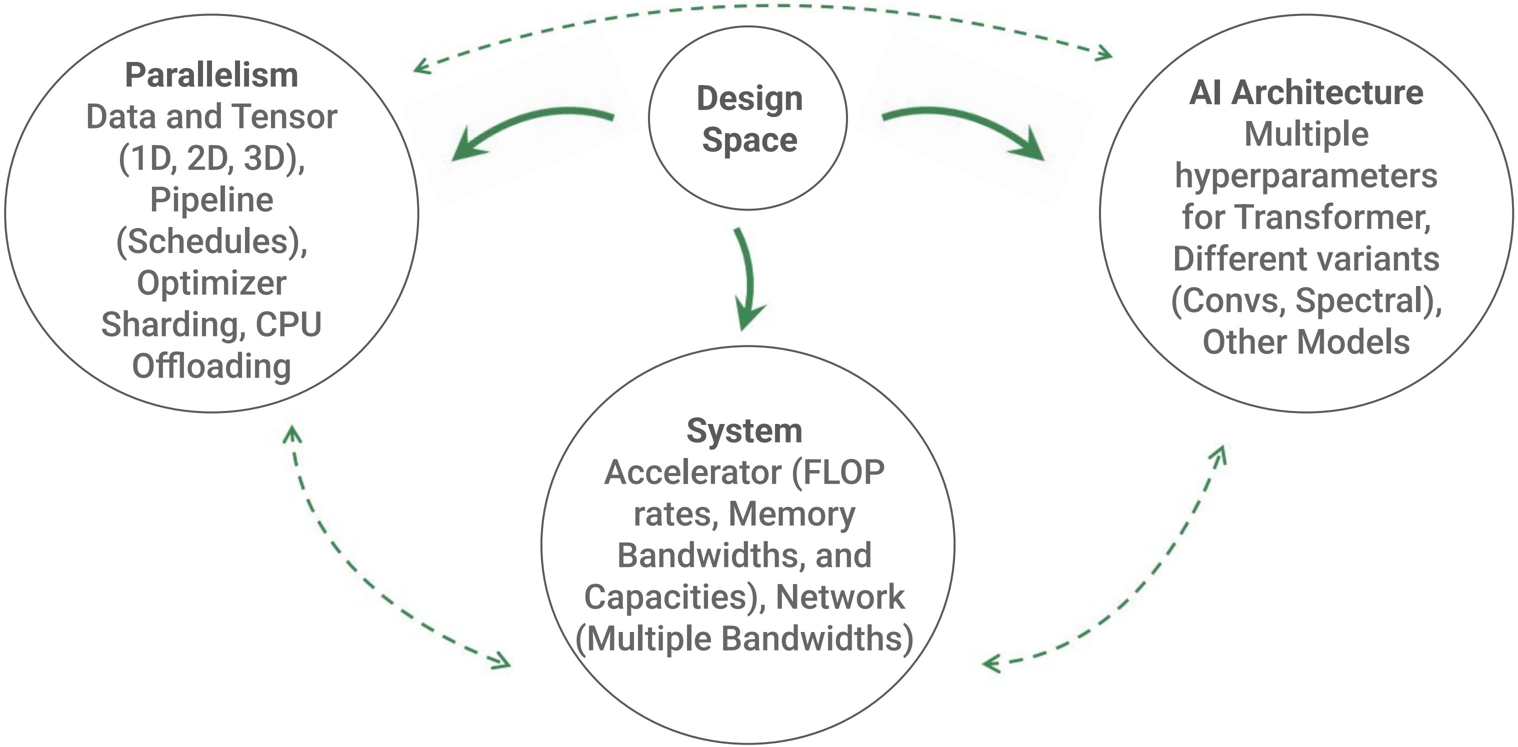


Megatron-LM & Megatron-Core: GPU optimized techniques for training transformer models at-scale, NVIDIA, Github

Some Key Takeaways

- Placement of GPUs on high-bandwidth domain affects the optimal parallelism
  - Software codebases to be flexible to this
  - NVLINK domains help expose "easier" parallelisms from the software POV
- LLMs benefit from large NVLINKs at pre-training scales
  - Fine-tuning scales can leverage other parallelization strategies to be less sensitive
  - HBM capacity is underutilized for the largest scales
- Science Transformers benefit uniformly from NVLINK due to memory pressure
  - Demand 4D parallelism (data + pipeline + 2D tensor + optimizer sharding)
  - Capacity is more critical (High capacity, low bandwidth alternatives?)
- 4D/ND (SUMMA/context) parallelism can give you good speedups

Thank You!



Comprehensive Performance Modeling and System Design Insights for Foundation Models, PMBS, SC24, Github