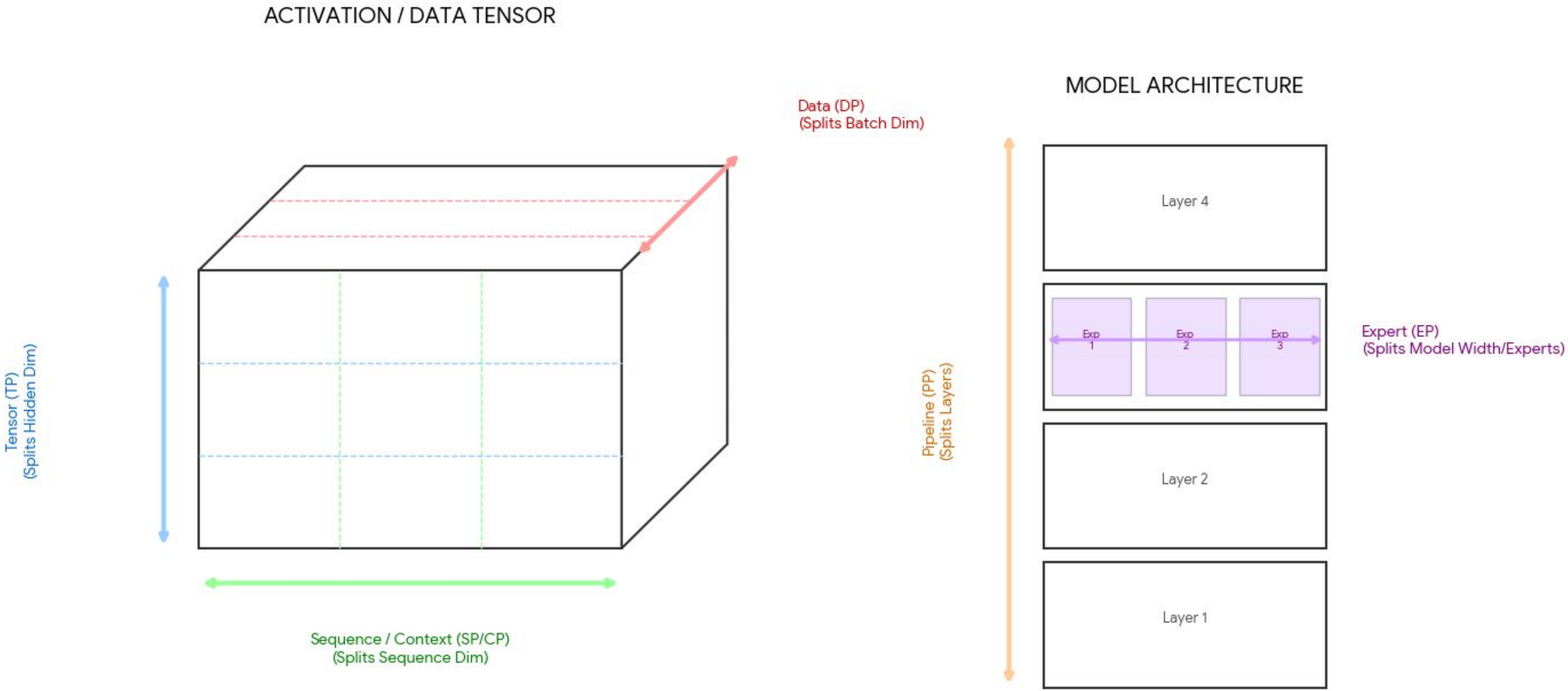
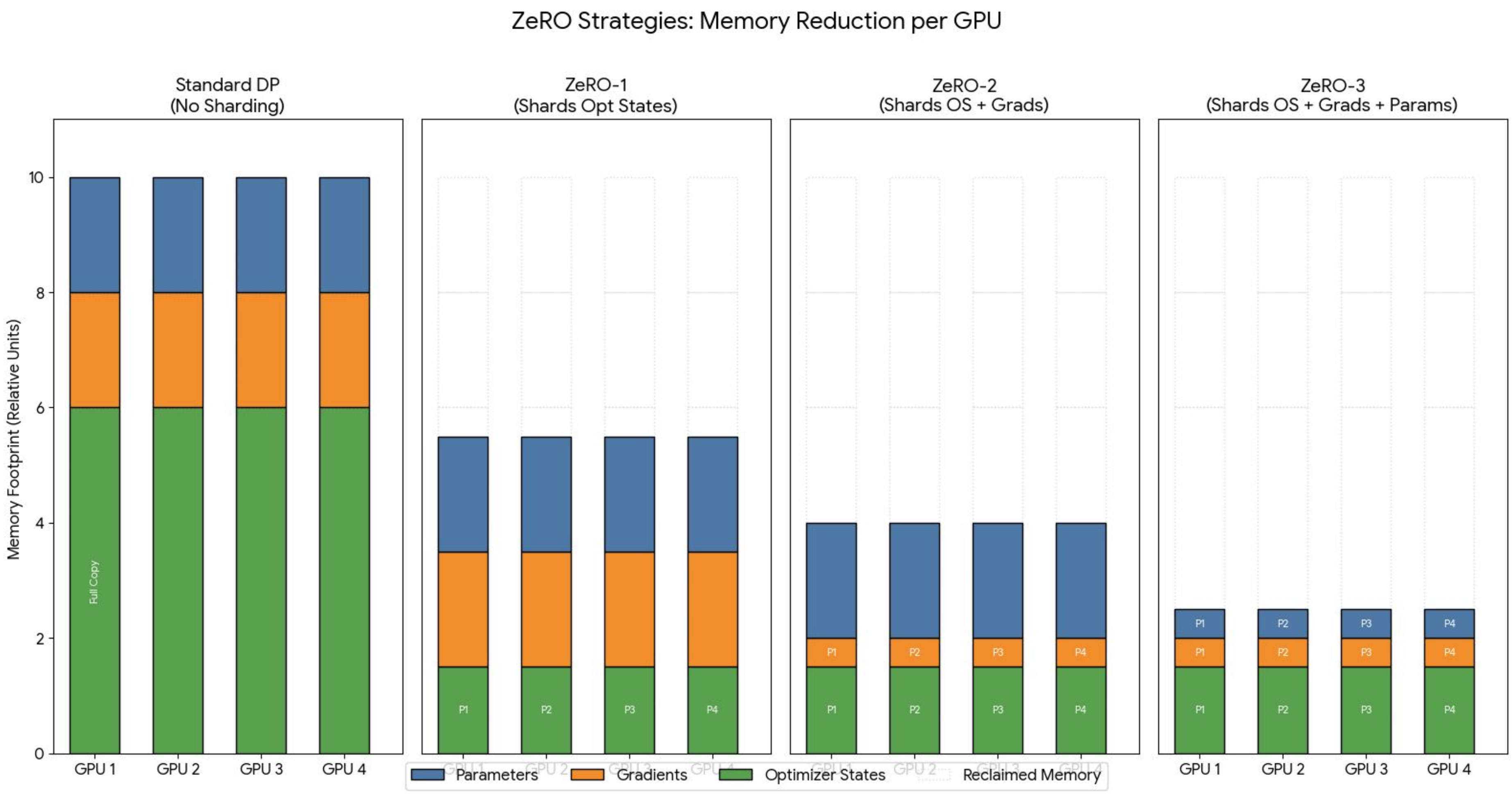
5D Parallelism

made with ❤️ for “Little ML book club”

The 5D Parallelism Landscape

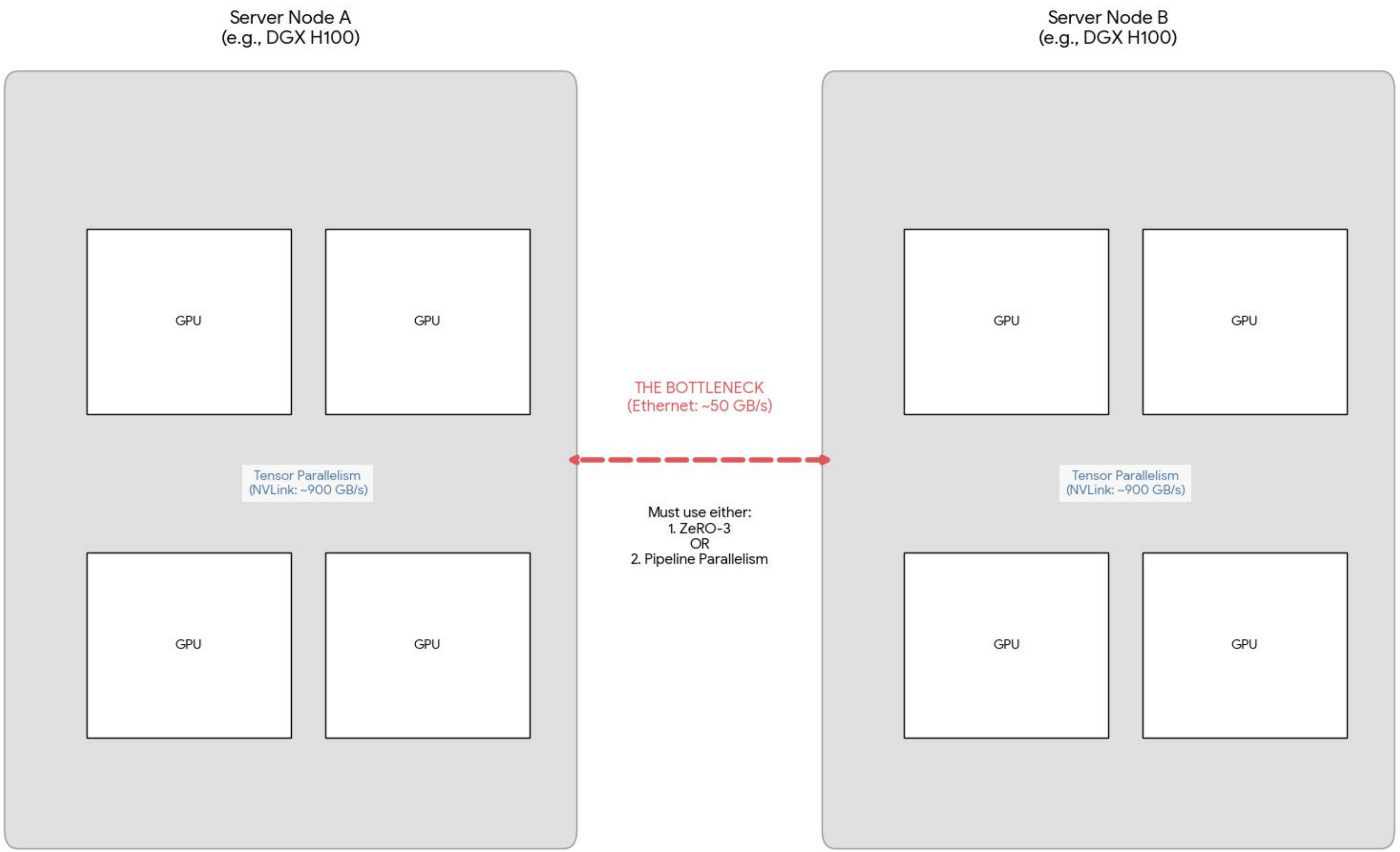


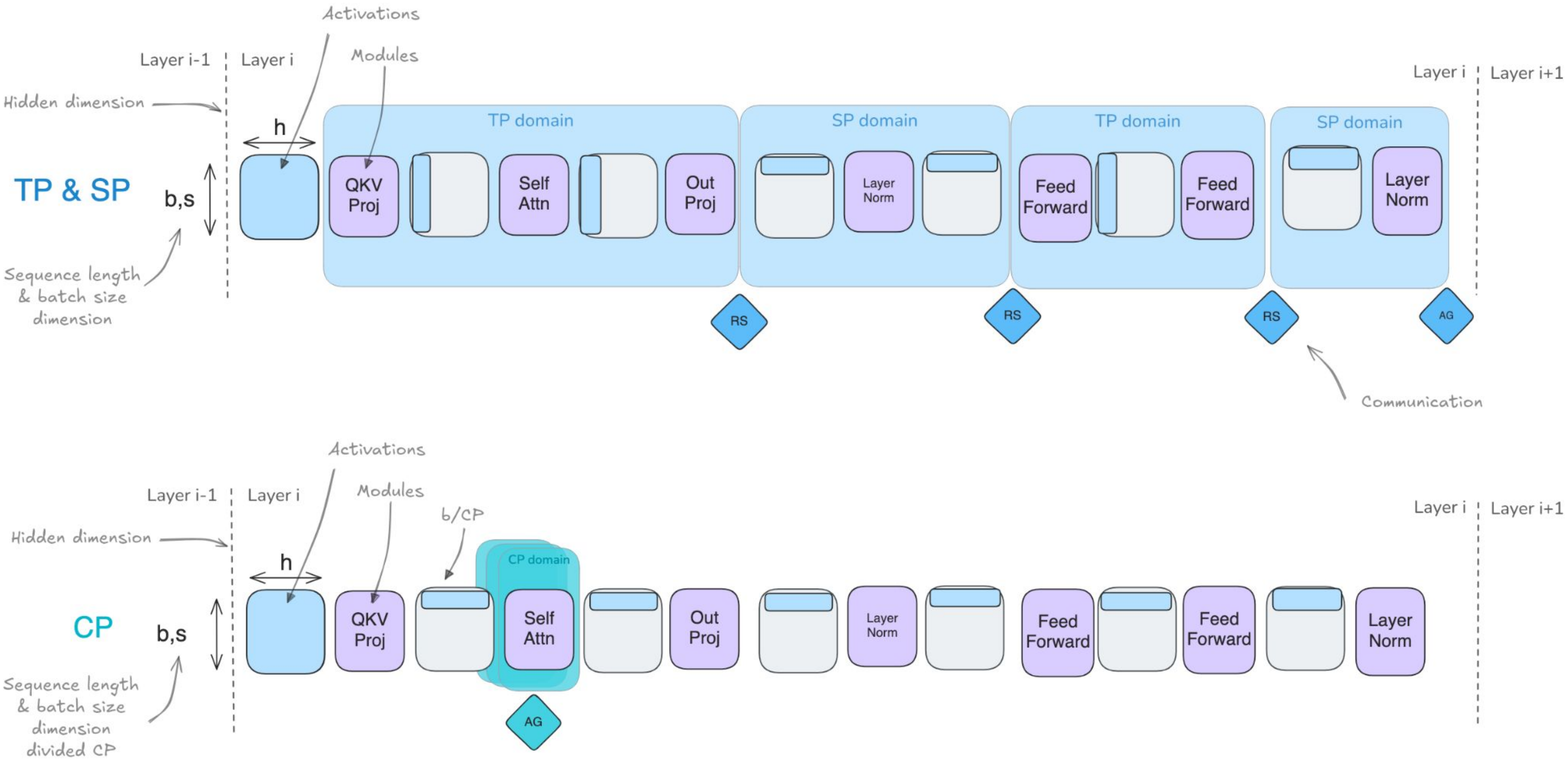
Left: Data Tensor cuts (DP, TP, SP/CP) | Right: Model Architecture cuts (PP, EP)

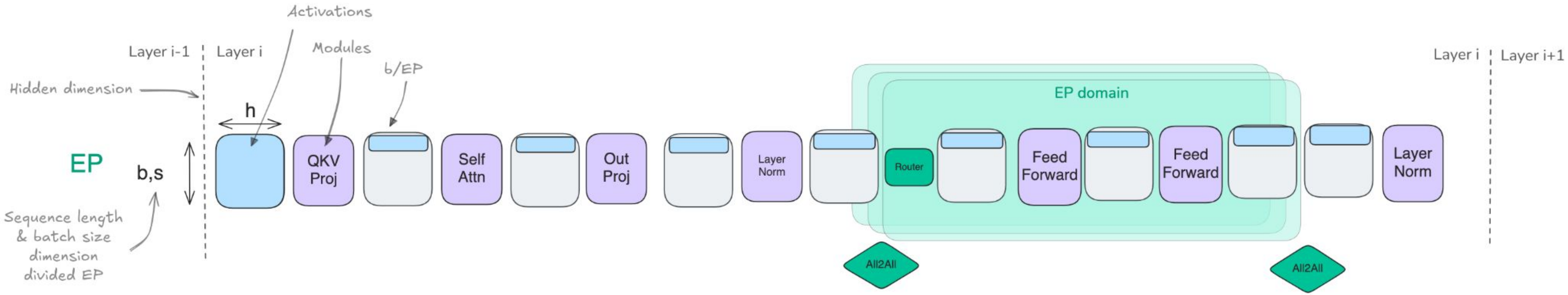

ZeRO Strategies: Memory Reduction per GPU
Standard DP
(No Sharding)
ZeRO-1
(Shards Opt States)
ZeRO-2
(Shards OS + Grads)
ZeRO-3
(Shards OS + Grads + Params)
Memory Footprint (Relative Units)
0
2
4
6
8
10
Full Copy
P1
P2
P3
P4
GPU 1
GPU 2
GPU 3
GPU 4
Parameters
Gradients
Optimizer States
Reclaimed Memory

ZeRO-3 In Action: The Fetch-Compute-Discard Cycle

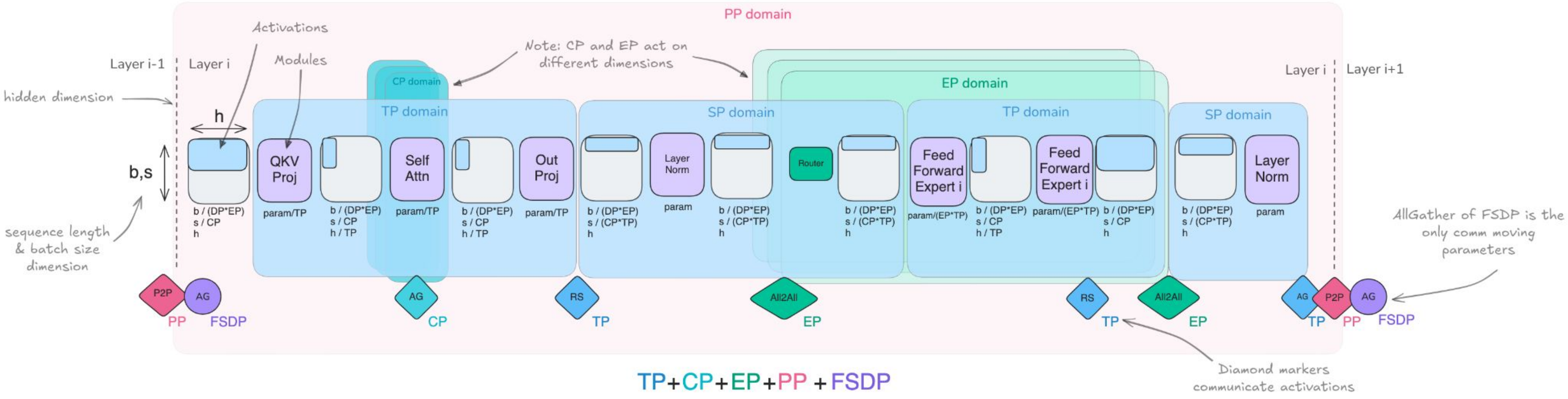
	ZeRO-3	**Pipeline Parallelism**
Each compute unit stores...	only a fraction of a layer	a full layer
Communication is used to transfer...	weights	activations
Orchestration	Model-agnostic	Model-agnostic
Implementation challenges	Complex to handle model partitioning and communications	Complex to handle efficient PP schedules
Scaling considerations	Prefers large mbs and seq_len to hide comms	Prefers large $grad\_acc$ to hide bubble


Server Node A
(e.g., DGX H100)
Server Node B
(e.g., DGX H100)
GPU
GPU
GPU
GPU
Tensor Parallelism
(NVLink: ~900 GB/s)
THE BOTTLENECK
(Ethernet: ~50 GB/s)
Must use either:
1. ZeRO-3
OR
2. Pipeline Parallelism
GPU
GPU
GPU
GPU
Tensor Parallelism
(NVLink: ~900 GB/s)


Activations
Modules
Layer i-1
Layer i
Hidden dimension
h
TP & SP
b,s
Sequence length & batch size dimension
TP domain
QKV Proj
Self Attn
Out Proj
SP domain
Layer Norm
TP domain
Feed Forward
Feed Forward
SP domain
Layer Norm
Layer i
Layer i+1
RS
RS
RS
AG
Communication
Activations
Modules
Layer i-1
Layer i
b/CP
Hidden dimension
h
CP domain
CP
b,s
QKV Proj
Self Attn
Out Proj
Layer Norm
Feed Forward
Feed Forward
Layer Norm
Layer i
Layer i+1
Sequence length & batch size dimension divided CP
AG



Activations
Modules
Layer i-1
Layer i
b/EP
Hidden dimension
h
EP
b,s
Sequence length & batch size dimension divided EP
QKV Proj
Self Attn
Out Proj
Layer Norm
EP domain
Router
Feed Forward
Feed Forward
Layer Norm
Layer i
Layer i+1
All2All
All2All


PP domain
Activations
Modules
Note: CP and EP act on different dimensions
Layer i-1
Layer i
hidden dimension
h
b,s
sequence length & batch size dimension
CP domain
TP domain
SP domain
EP domain
TP domain
SP domain
Layer i
Layer i+1
QKV Proj
Self Attn
Out Proj
Layer Norm
Router
Feed Forward Expert i
Feed Forward Expert i
Layer Norm
b / (DP*EP)
s / CP
h
param/TP
b / (DP*EP)
s / CP
h / TP
param/TP
b / (DP*EP)
s / CP
h / TP
param/TP
b / (DP*EP)
s / (CP*TP)
h
param
b / (DP*EP)
s / (CP*TP)
h
b / (DP*EP)
s / (CP*TP)
h
param/(EP*TP)
b / (DP*EP)
s / CP
h / TP
param/(EP*TP)
b / (DP*EP)
s / CP
h
b / (DP*EP)
s / (CP*TP)
h
param
P2P
AG
PP
FSDP
AG
CP
RS
TP
All2All
EP
RS
TP
All2All
EP
AG
P2P
AG
TP
PP
FSDP
AllGather of FSDP is the only comm moving parameters
Diamond markers communicate activations
TP+CP+EP+PP + FSDP

1. Meta LLaMA Family

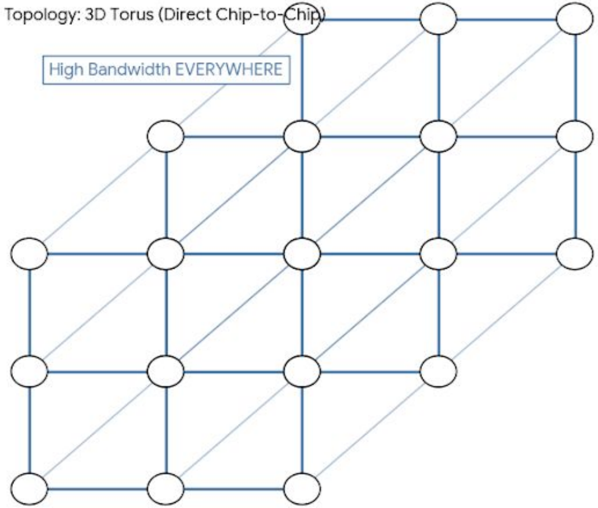
Model	Date	Parameters	Hardware	TP	PP	DP/ FSDP	CP	EP	Key Innovations
LLaMA 1	Feb 2023	7B–65B	2,048 A100 80GB	—	—	DP	—	—	Basic data parallelism; RSC cluster
LLaMA 2	Jul 2023	7B–70B	RSC + prod clusters	—	—	FSDP	—	—	Introduced FSDP; GQA for 70B; 4K context; 1.73M GPU-hours for 70B
LLaMA 3	Apr 2024	8B–70B	16,384 H100	8	16	FSDP (128)	1	—	4D parallelism; 8K context; 126 layers (not 128) for balanced PP
LLaMA 3.1	Jul 2024	8B–405B	16,384 H100	8	16	FSDP (128)	1-16	—	128K context via CP=16; all-gather CP (not ring attention); 38-43% MFU

2. Google PaLM/Gemini Family

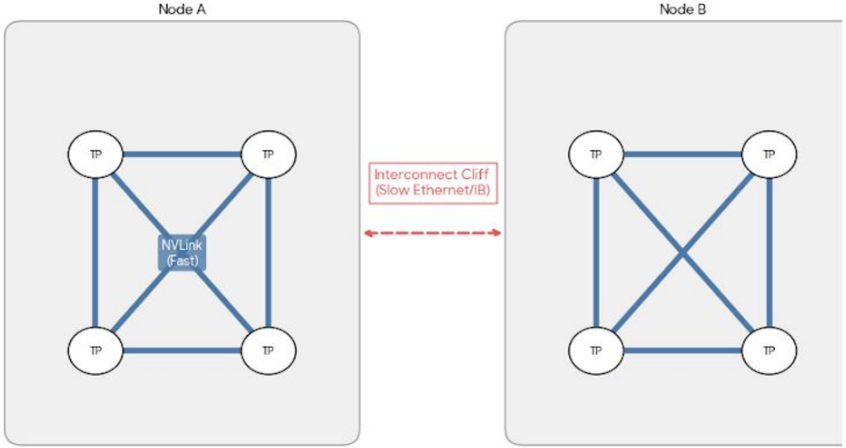
Model	Date	Parameters	Hardware	TP	PP	DP	CP	EP	Key Innovations
PaLM	Apr 2022	540B	6,144 TPU v4	12	**None**	256 (2D FSDP)	—	—	Pipeline-free; 57.8% HW utilization; Pathways system
PaLM 2	May 2023	Undisclosed	TPU v4	✓	—	✓	—	✓ (sparse)	MoE architecture; improved compute-optimal scaling
Gemini 1.0 Ultra	Dec 2023	Undisclosed	Multi-DC TPU v4/ v5e	✓	—	✓	—	✓	**Multi-datacenter training**; 97% goodput; optical circuit switching
Gemini 1.5 Pro	Feb 2024	Undisclosed (MoE)	TPU v5+	✓	—	✓	✓	✓	Sparse MoE; up to 1M context; long-context specialization

The Core Tension: Topology Drives Parallelism Strategy

Google TPU Architecture
"Uniform 3D Torus"



GPU Clusters (Meta/DeepSeek)
"Hierarchical Topology"



Why skip PP?

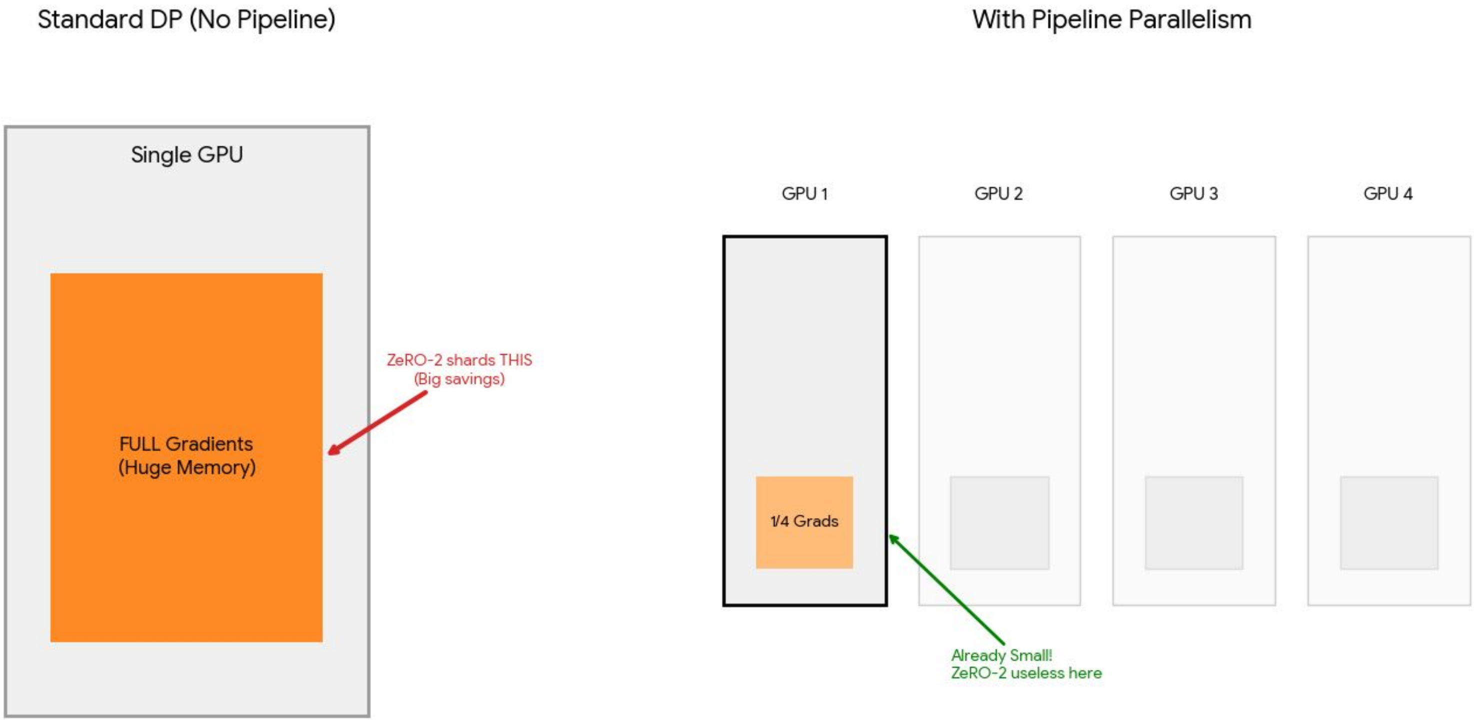
- All-Reduce is CHEAP everywhere
- PP Bubble overhead > Comm savings
- Avoid complexity & HBM stress

Why use PP?

- Cannot do All-Reduce across nodes efficiently
- PP limits traffic to just 'Activations'

3. DeepSeek Family

Model	Date	Total Params	Active Params	Hardware	TP	PP	DP	EP	Key Innovations
DeepSeek 67B	Jan 2024	67B	67B (dense)	H800 cluster	✓	✓	ZeRO	—	Baseline dense model
DeepSeek-V2	May 2024	236B	21B	H800 cluster	—	16 (ZeroBubble)	ZeRO-1	8	MLA attention; DeepSeekMoE; 42.5% cost reduction vs 67B
DeepSeek-V3	Dec 2024	671B	37B	2,048 H800	**None**	16 (DualPipe)	ZeRO-1	64	Aux-loss-free balancing; FP8; **$5.6M total cost**; 180K GPU-hr/T tokens


Standard DP (No Pipeline)
With Pipeline Parallelism
Single GPU
FULL Gradients
(Huge Memory)
ZeRO-2 shards THIS
(Big savings)
GPU 1
GPU 2
GPU 3
GPU 4
1/4 Grads
Already Small!
ZeRO-2 useless here

see you next time