AI芯片思考

ZOMI

Course chenzomi12.github.io

GitHub github.com/chenzomi12/DeepLearningSystem

Talk Overview

1. **AI 计算体系**
   - 深度学习计算模式
   - 计算体系与矩阵运算
2. **AI 芯片基础**
   - 通用处理器 CPU
   - 通用图形处理器 GPU
   - AI专用处理器 NPU/TPU
3. **GPU详解**
   - 英伟达GPU架构发展
   - Tensor Core和NVLink
4. **国外 AI 芯片**
   - 特斯拉 DOJO 系列
   - 谷歌 TPU 系列
5. **国内 AI 芯片**
   - 壁仞科技芯片架构
   - 寒武纪科技芯片架构
6. **AI芯片的思考**
   - SIMD&SIMT与编程体系
   - AI芯片的架构思路与思考

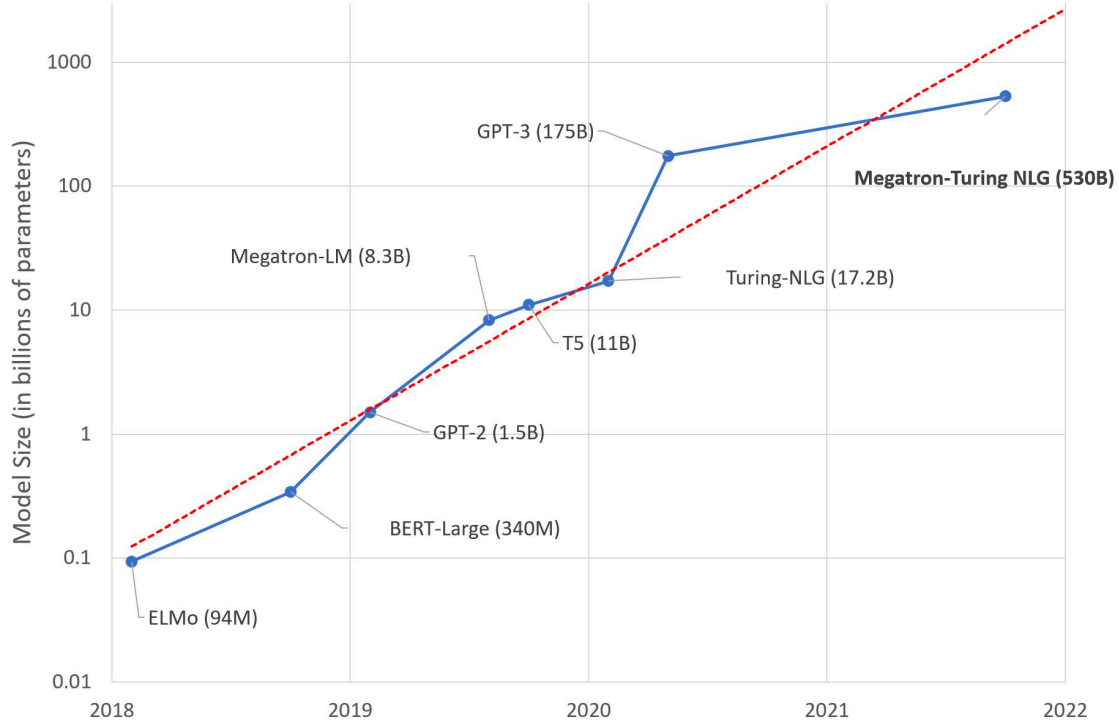
Talk Overview

1. AI 芯片思考

- thought 1：模型 Memory & FLOPs 增长
- thought 2：模型快速演变
- thought 3：生产部署需要多租户技术 Multi-tenancy
- thought 4：较大的 SRM 和存取速度极快的 DRAM
- thought 5：重要的是内存，而非 FLOPs
- thought 6： DSA 既要专业也要灵活
- thought 7：半导体技术发展速度不不同，要做好选型
- thought 8：编译器优化和 AI 应用兼容

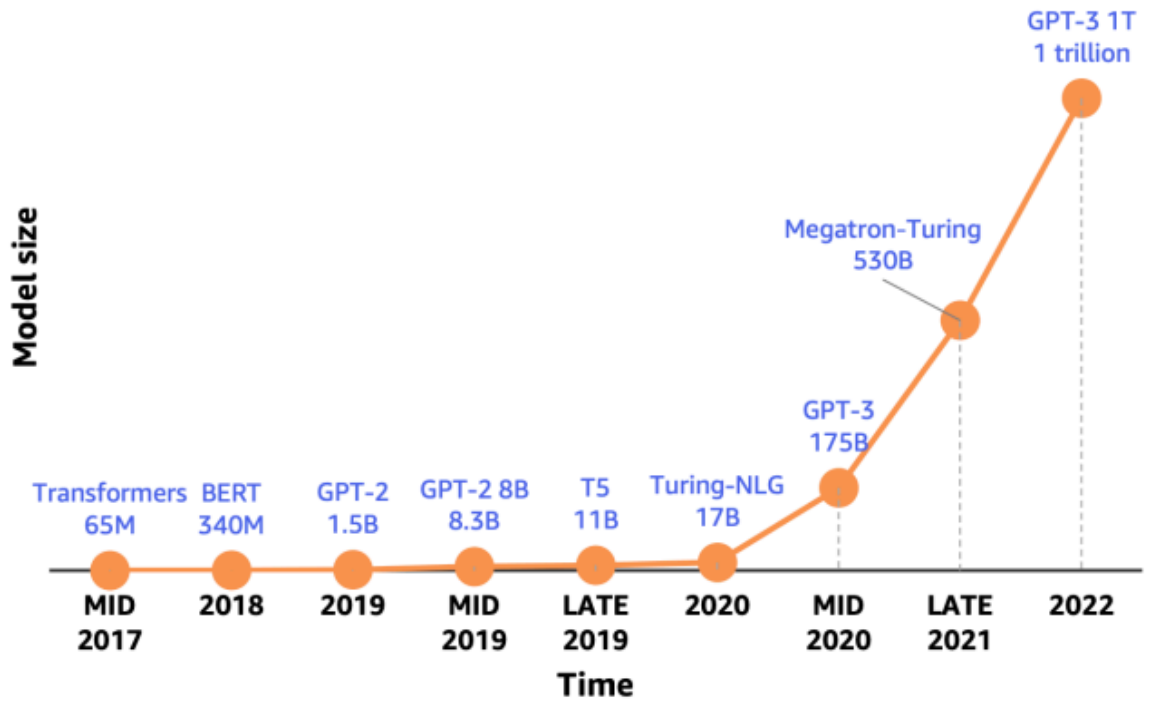
thought 1：模型 Memory & FLOPs 增长

1. 推理模型所需的内存空间和算力平均每年增长50%，模型所需内存和算力增长大约10~20倍
2. 芯片设计1 years，部署1 years，实际使用并优化~3years，共5年

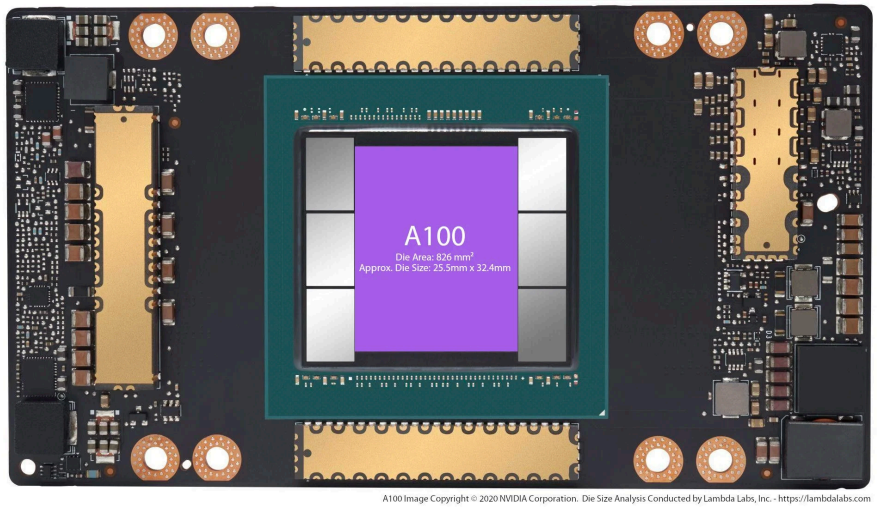


thought 1：模型 Memory & FLOPs 增长

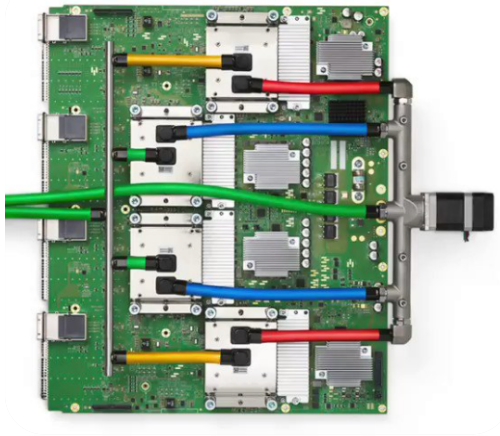
1. 训练模型的增长速度比推理模型更快;
2. 2016-2023年，SOTA训练模型的算力需求年均增长10X；
3. GPT-2 模型的参数量从15亿增长到 GPT-3 1750亿，提高了100X。



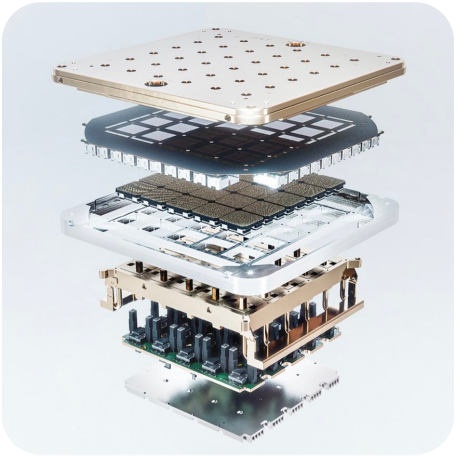
thought 1：模型 Memory & FLOPs 增长



- A100 HBM 40G/80G
- H100 HBM 64GB/94GB/ 188GB



- TPUv4 32G



- DOJO 16G

- Ascend 64G

- MLU 370 16G

thought 2：模型快速演变

- 深度学习是一个日新月异的领域
- 2015 年 MLP 仍为主流
- 2018 年 CNN、RNN和BERT百花齐放
- 2020年 Transformer 大模型独占鳌头
- DSA 需要足够通用支持新的模型

DNN Name	*2020*	*2016*
MLP0	25%	61%
MLP1		
CNN0	18%	5%
CNN1		
LSTM0	0%	29%
LSTM1		
RNN0	29%	0%
RNN1		
BERT0	28%	0%
BERT1		
TOTAL	100%	95%

thought 3：生产部署需要多租户技术 Multi-tenancy

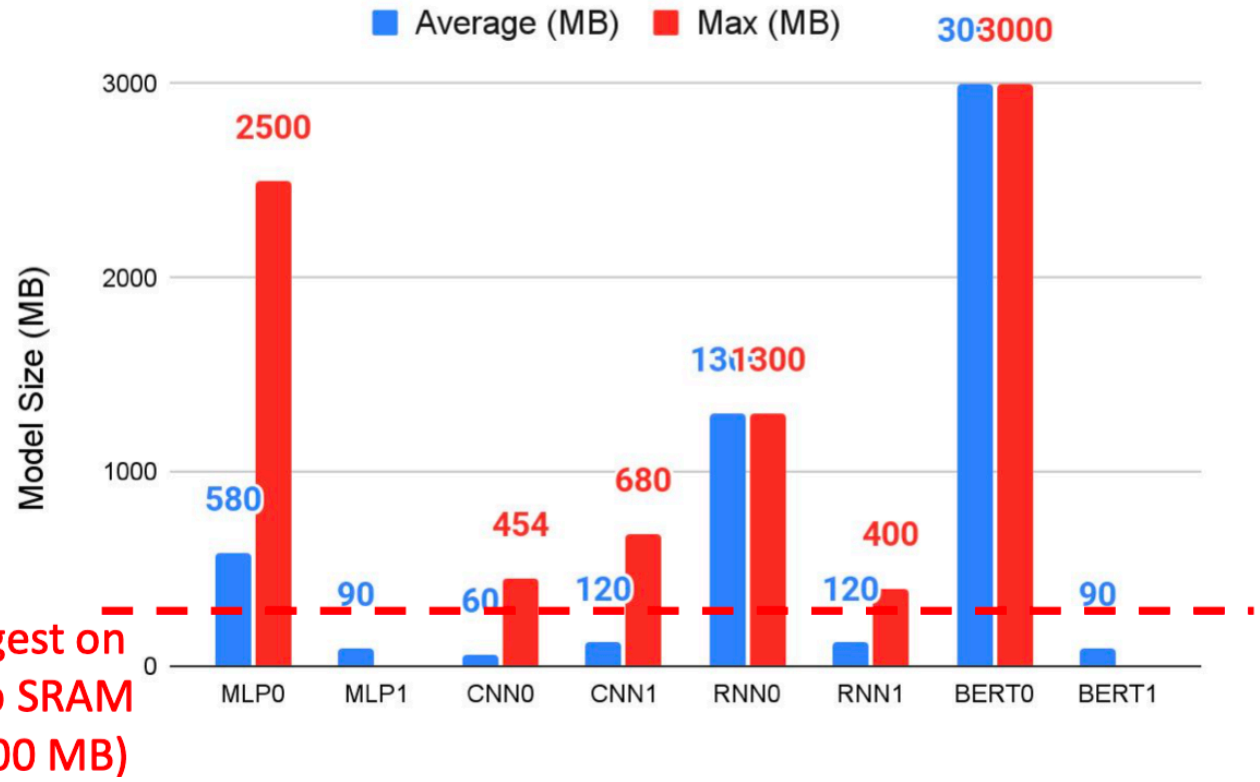
- 大部分AI 相关论文假设同一时间 NPU 只需运行一个模型。实际应用需要切换不同模型：
  - 机器翻译涉及语言对，用不同的模型；
  - 用到一个主模型和配套多个实验进行模型；
  - 对吞吐量和延迟有不同要求，不同模型使用不用 batch size。

算力切分、显存虚拟化、内存寻址、虚拟内存页等技术~

thought 4：较大的 SRAM 和存取速度极快的 DRAM

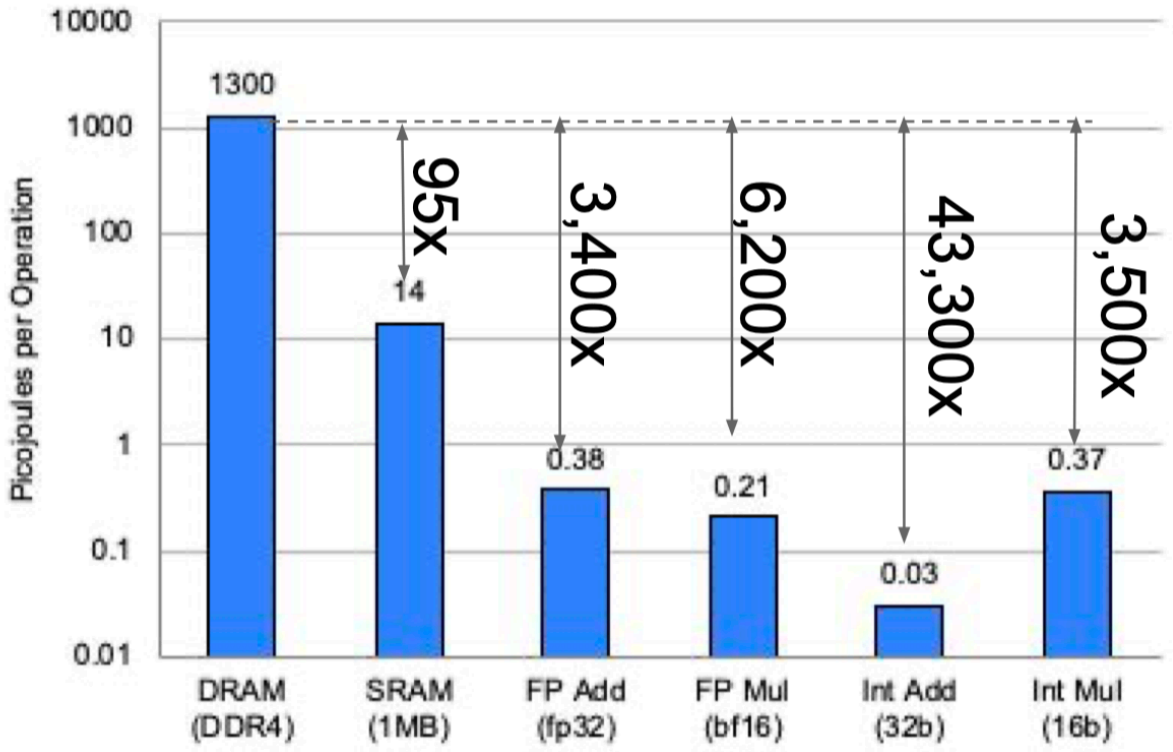
- 较大的 SRAM 和存取速度极快的DRAM：
  - 红色虚线单芯片最大 SRAM，不少模型需要的内存远大于此。
  - 部分芯片设计思路（Data-flow）是期望利用 SRAM 解决所有任务。

	Multi-tenancy?	*Avg # Programs (StdDev), Range*
MLP0	Yes	27 (±17), 1-93
MLP1	Yes	5 (±0.3), 1-5
CNN0	No	1
CNN1	Yes	6 (10), 1-34
RNN0	Yes	13 (±3), 1-29
RNN1	No	1
BERT0	Yes	9 (±2), 1-14
BERT1	Yes	5 (±0.3), 1-5

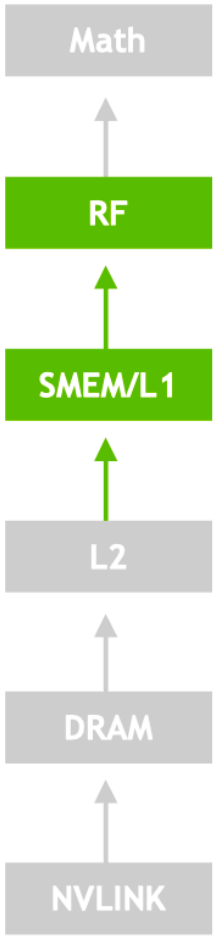


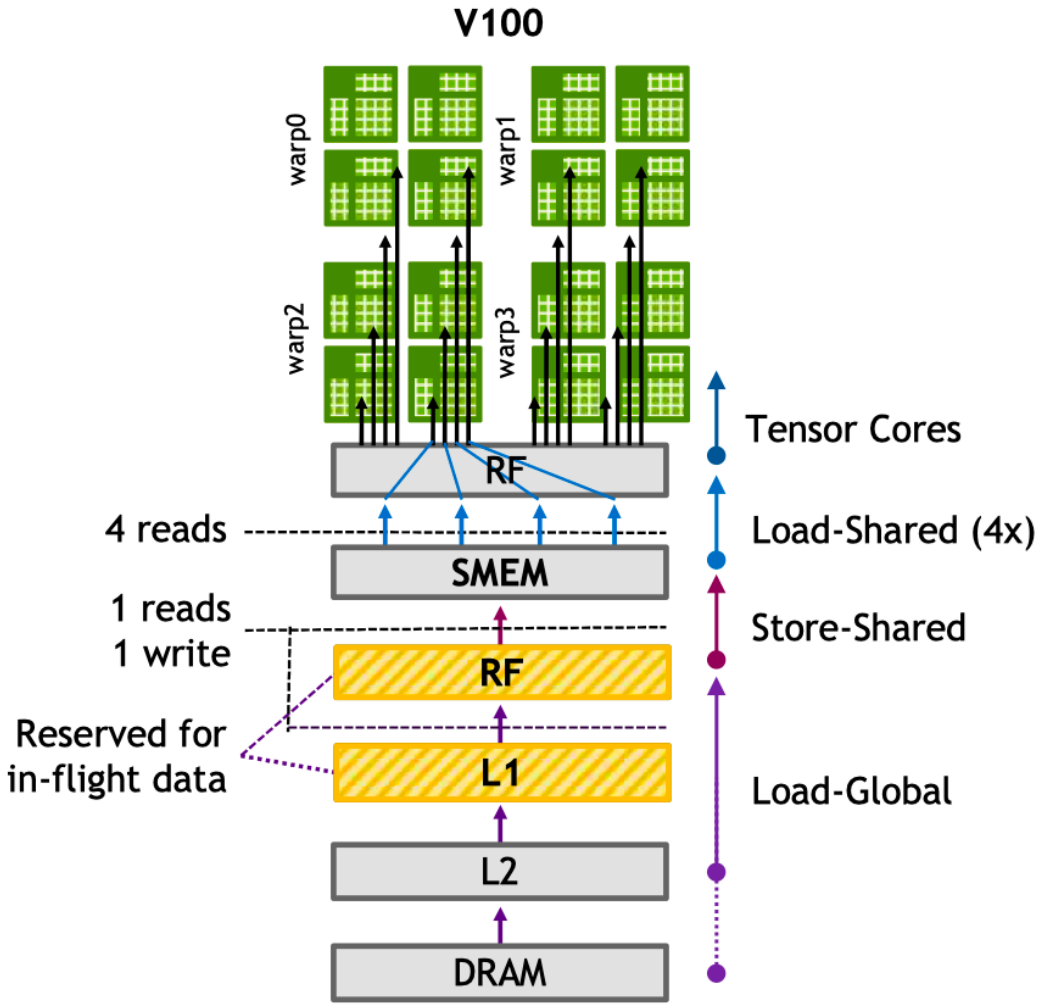
thought 5：重要的是内存，而非 FLOPs

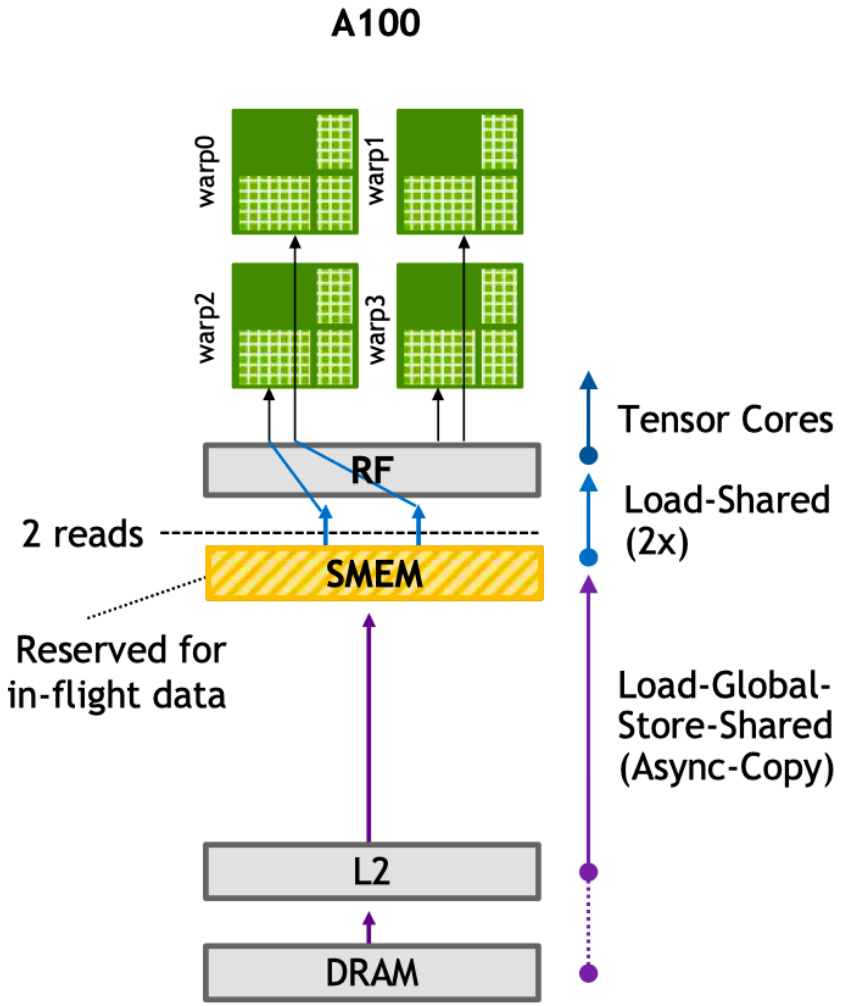
1. 现代微处理器最大的瓶颈是能耗，而不是芯片集成度：
   - 访问 DRAM 能耗是访问片上 SRAM 100X；
   - 算术运算能耗的 5000 ~ 10,000X；
2. AI 芯片通过增加浮点运算单元（FPU）来分摊内存访问开销。
3. AI 系统开发者常通过减少浮点运算数 FLOPs 来优化模型，减少内存访问数是更有效的办法。



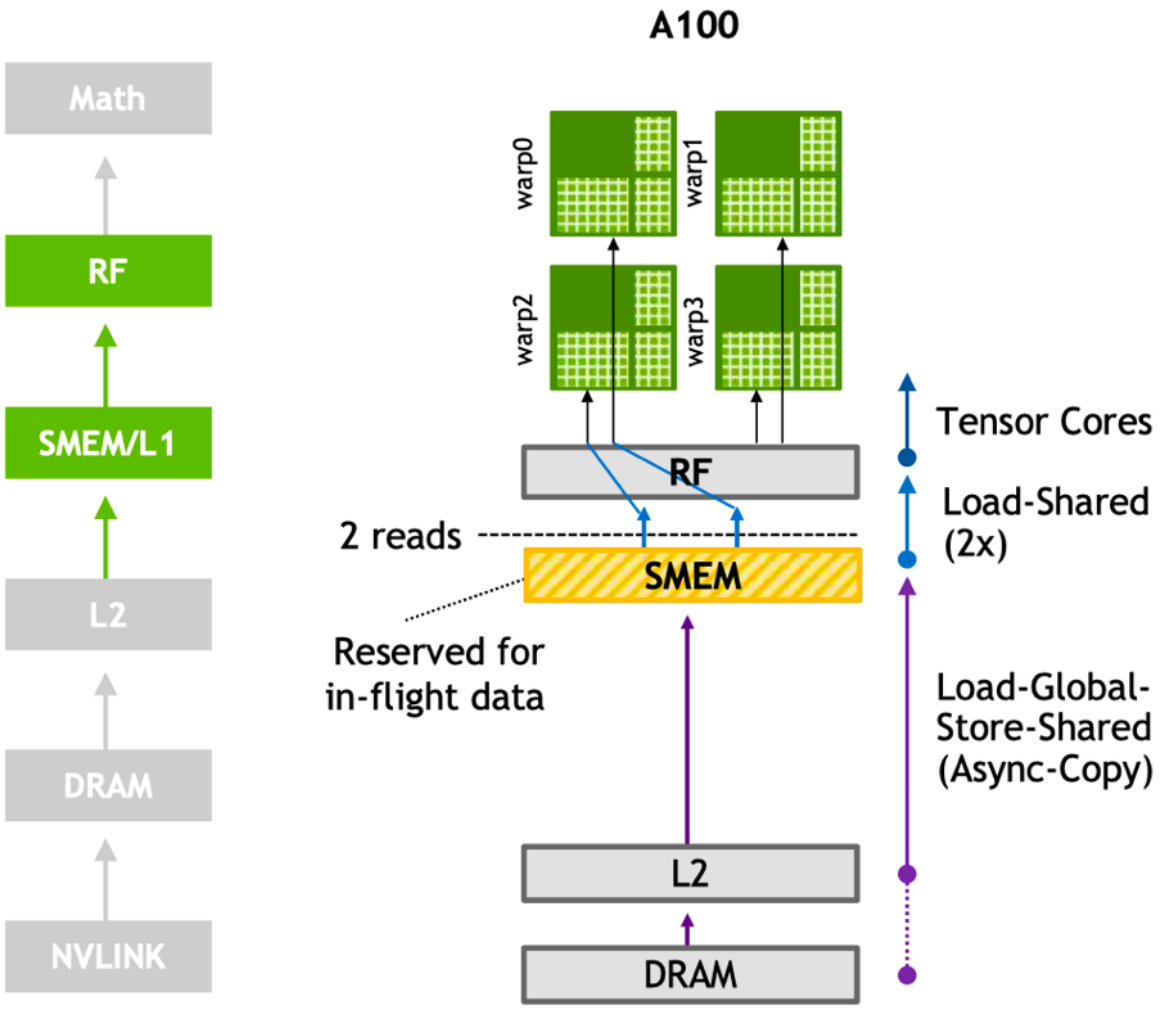
第三代 Tensor Core (Ampere)







第三代 Tensor Core (Ampere)

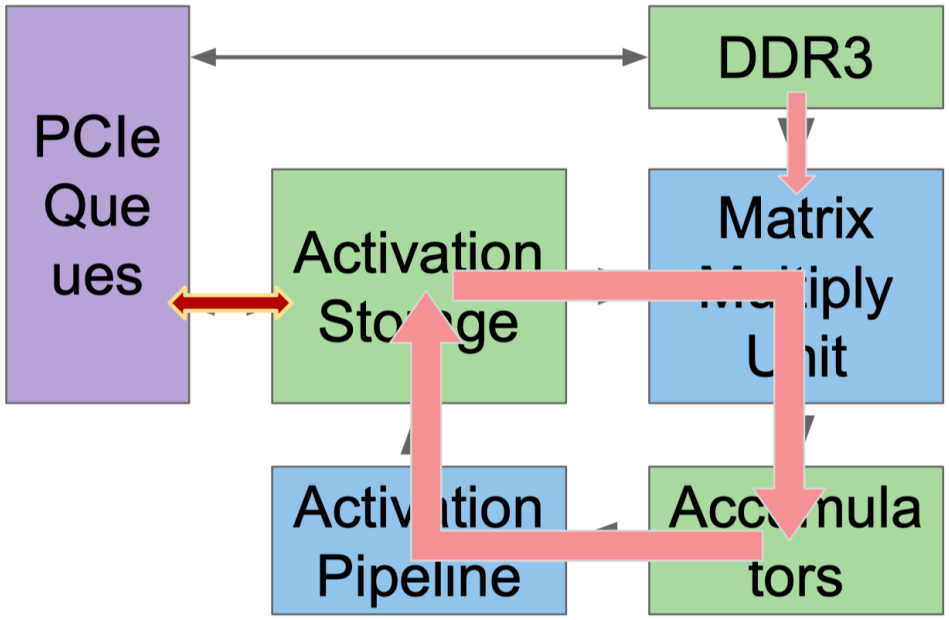


- Ampere架构的 Tensor Core 一个 warp 中 32 个线程间共享数据，而 Volta 架构 Tensor Core 只有8个线程；
- 可以更好地在线程间减少矩阵的数据搬运。

Data-Flow 的硬件架构方式

- TPU v1 有 65,000 个乘法单元（256x256）
- 700 MHz 时钟频率
- 峰值算力：92T Operations/s
  - 65,000×2×700M ≈ 90 TeraOPS
- 4 MB on-chip Accumulator memory
- 24 MB on-chip Activation Storage
- 2133MHz DDR3 DRAM channels for weights(8GB)

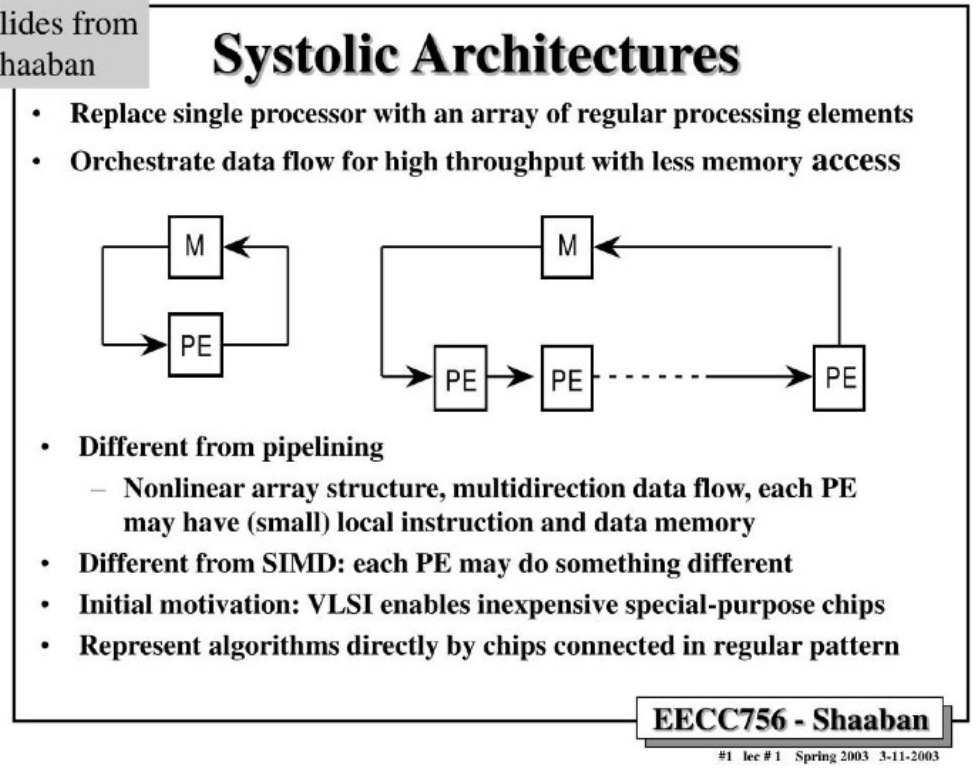
TPUv1: High-level Chip Architecture



脉动阵列 systolic array

1. TPU 使用脉动阵列，以固定时间间隔使数据从不同方向流入阵列中的处理单元，最后将数据累积，以完成大型矩阵乘法运算。
2. 70年代芯片只有单金属层，不能很好地实现互连，Kung和Leiserson提出“脉动阵列“以减少布线，简化连接。
3. 现代芯片多达10个金属层，最大难点是能耗，脉动阵列能效高，使用脉动阵列可以使芯片容纳更多乘法单元，从而分摊内存访问开销。

Slides from Shaaban



thought 6： DSA 既要专业也要灵活

- **DSA难点**：DSA 难点在于既要进行针对性的优化，同时还须保持一定的灵活性。
- **训练难点**：训练之所以比推理更加复杂，是因为训练的计算量更大，包含反向传播、转置和求导等运算。训练时需要将大量运算结果储存起来用于反向传播的计算，因此也需要更大的内存空间。
- **AI芯片需求**：计算的数据格式动态范围支持广泛（如BF16、FP16、HF32），用于 AI计算。指令、流水、可编程性也更高，需要灵活的编译器和上层软硬件配套。

thought 7：半导体技术发展速度不不同，要做好选型

- 计算逻辑的进步速度很快，芯片布线的发展速度则较慢；
- 而 SRAM 和 HBM 比 DDR4 和 GDDR6 速度更快，能效更高。

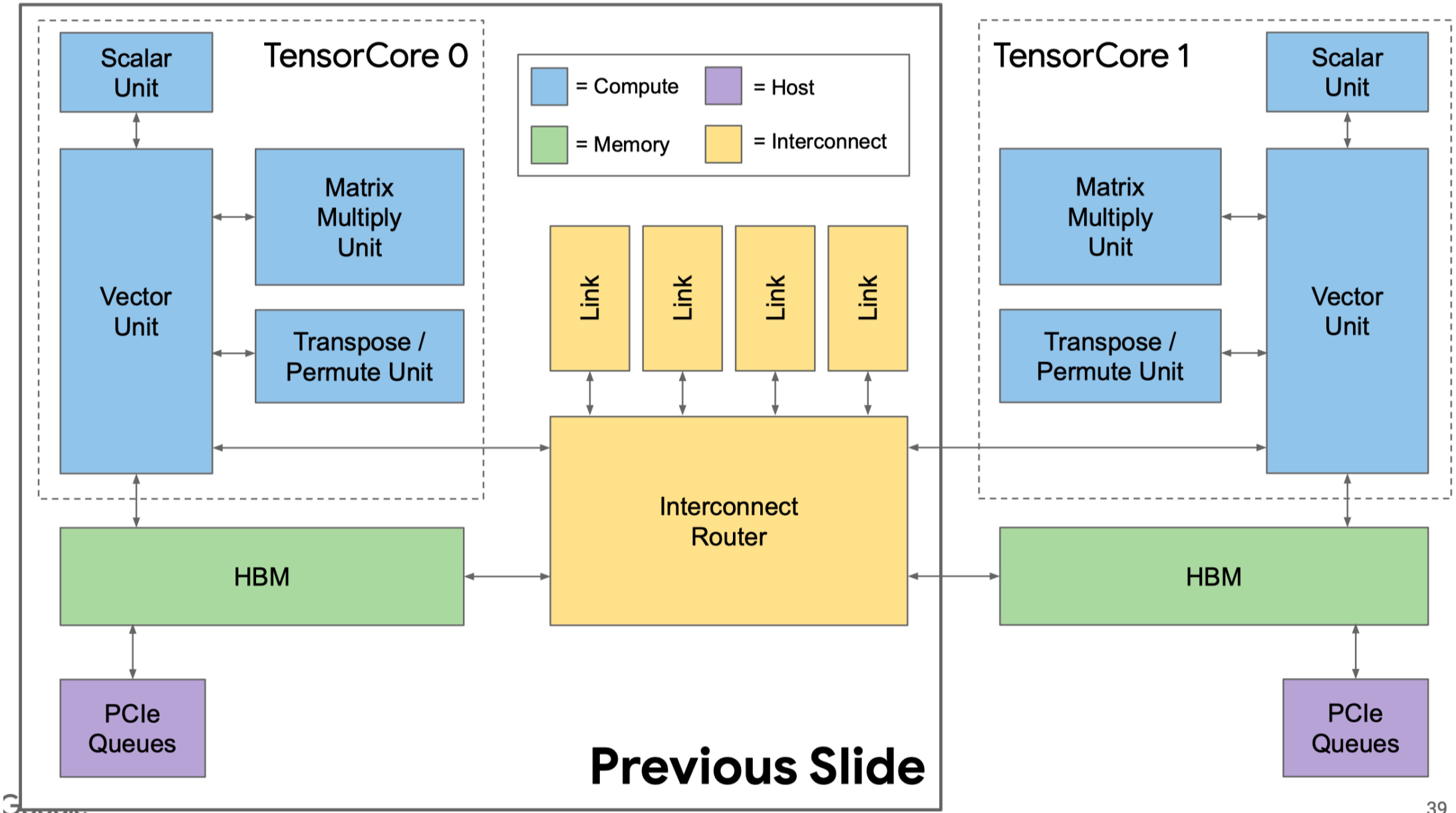
Operation		*Picojoules per Operation*		
		45 nm	*7 nm*	*45 / 7*
+	Int 8	0.03	0.007	4.3
	Int 32	0.1	0.03	3.3
	BFloat 16	--	0.11	--
	IEEE FP 16	0.4	0.16	2.5
	IEEE FP 32	0.9	0.38	2.4
×	Int 8	0.2	0.07	2.9
	Int 32	3.1	1.48	2.1
	BFloat 16	--	0.21	--
	IEEE FP 16	1.1	0.34	3.2
	IEEE FP 32	3.7	1.31	2.8
SRAM	8 KB SRAM	10	7.5	1.3
	32 KB SRAM	20	8.5	2.4
	1 MB SRAM	100[1]	14[1]	7.1
GeoMean		--	--	2.6[1]
DRAM		Circa 45 nm	Circa 7 nm	
	DDR3/4	1300	1300[2]	1.0
	HBM2	--	250-450[2]	--
	GDDR6	--	350-480[2]	--

Horowitz 1MB SRAM value is based on a single bank SRAM. Most engineers would use multiple banks, which is reason for 7.1x reduction in 1MB SRAM vs 2.4 for 32 KB SRAM.

1300 pJ for DDR3/4 DRAM is only the I/O [Sto12]. HBM2 and GDDR6 also list only the I/O energy [Mic17, O'C17, Smi20].

Horowitz M. "Computing's energy problem (and what we can do about it)". IEEE International Solid-State Circuits Conference Digest of Technical Papers, 2014.

Jouppi et al., Ten Lessons From Three Generations Shaped Google's TPUv4i, ISCA, 2021

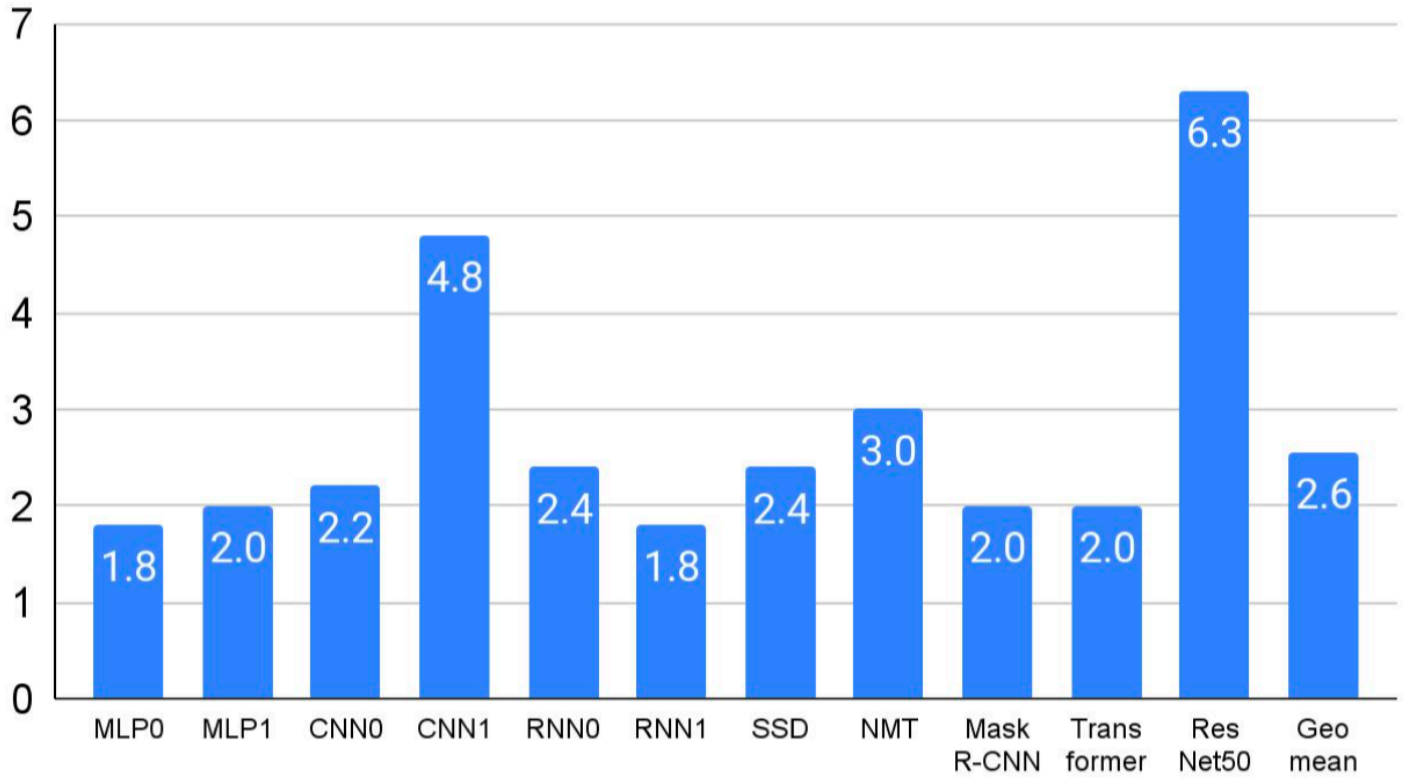

TensorCore 0
Scalar Unit
Vector Unit
Matrix Multiply Unit
Transpose / Permute Unit
= Compute
= Host
= Memory
= Interconnect
Link
Link
Link
Link
Interconnect Router
HBM
PCIe Queues
TensorCore 1
Scalar Unit
Matrix Multiply Unit
Transpose / Permute Unit
Vector Unit
HBM
PCIe Queues
Previous Slide

thought 8：编译器优化和 AI 应用兼容

1. DSA 的编译器需要对AI 模型进行分析和优化，具体可分为与机器无关高级操作和与相关低级操作，从而提供，提供不同维度的优化 API 和 PASS。
2. 编译器维度目前比较多，有类似于 CUDA 提供编程体系，有类似于 TVM/XLA 提供编译优化：
   - 处理如 4096 个芯片的多核并行；
   - 向量、矩阵、张量等功能单元的数据级并行；
   - 322~400 位 VLIW 指令集的指令级并行；
   - 取决于软硬件能否进行缓存，编译器需要管理内存传输；
   - 编译器能够兼容不同功能单元和内存中的数据布局（如 Trans data）；

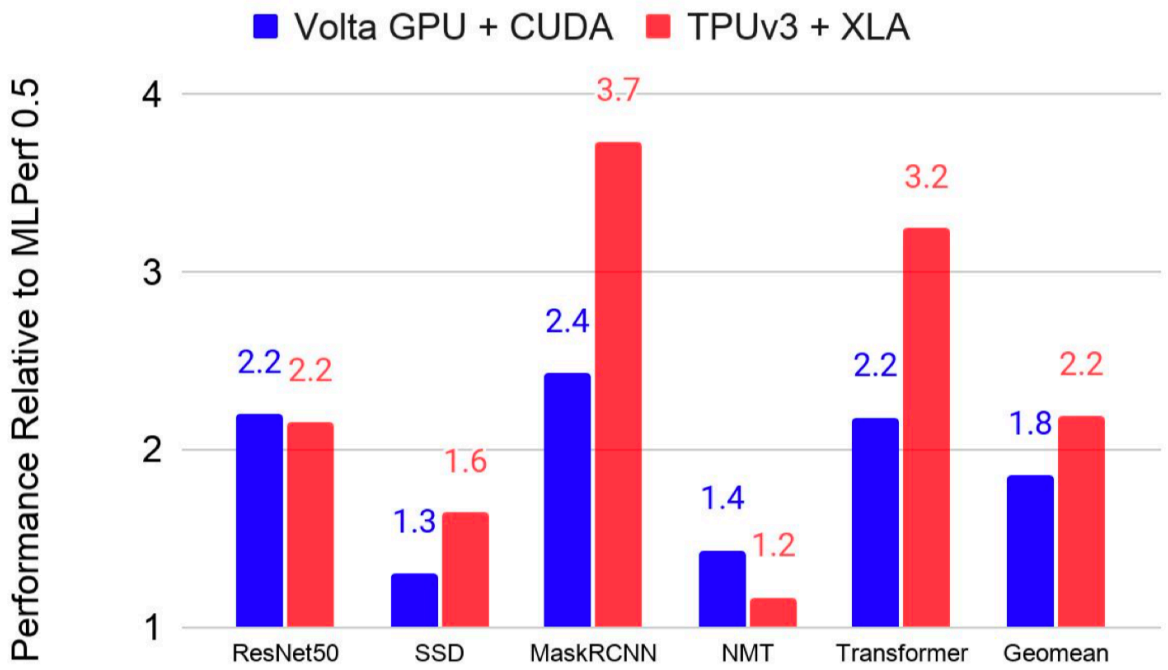
thought 8：编译器优化和 AI 应用兼容

- 与 CPU 和 NV GPU 相比，DSA 的软件栈还不够成熟。编译器优化最终能够提速多少？



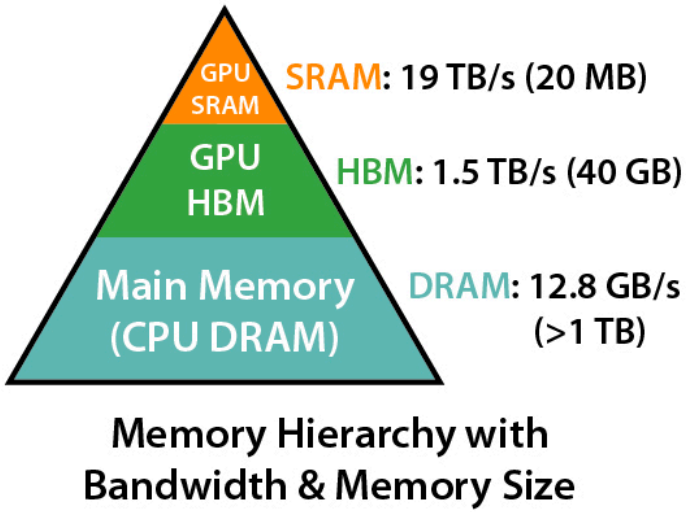
thought 8：编译器优化和 AI 应用兼容

- 蓝色表示使用GPU，红色表示使用TPU，通过编译器优化后模型的性能提升到 2X~。对 C++ 编译器，能在一年内把性能提升 5%-10% 已经很厉害。

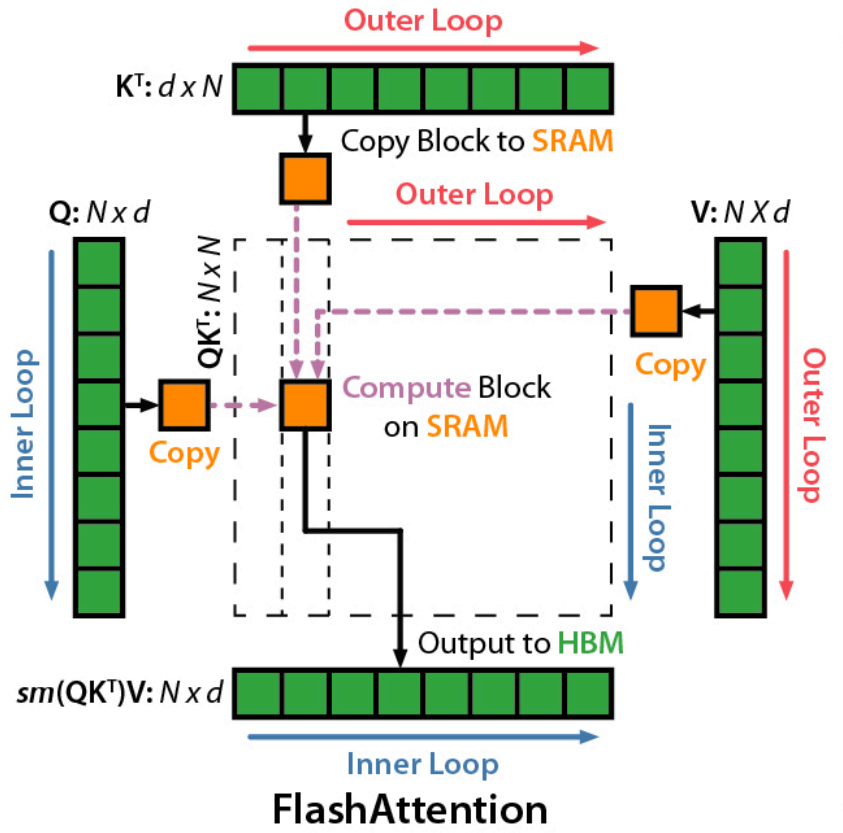


thought 8：编译器优化和 AI 应用兼容

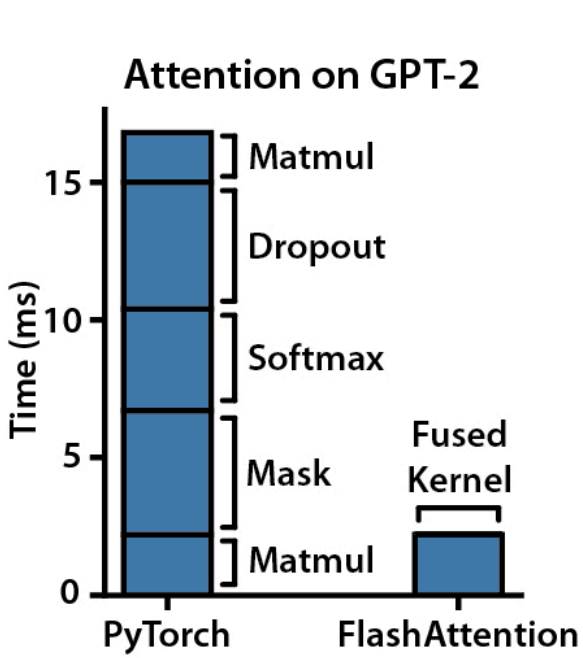
- 一种重新排序注意力计算的算法，Flash Attention 并没有减少计算量 FLOPs，而是从 IO 感知出发,减少 HBM 访问次数，从而减少了计算时间。



Memory Hierarchy with Bandwidth & Memory Size



FlashAttention



Give your idea

- 你还有什么建议吗？

Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.

ZOMI

Course chenzomi12.github.io

GitHub github.com/chenzomi12/DeepLearningSystem