

计算机系统

简介

仅谈我个人理解的部分。

四种处理器架构

CPU - Scalar

GPU - Vector

AI - Matrix

FPGA - Spatial

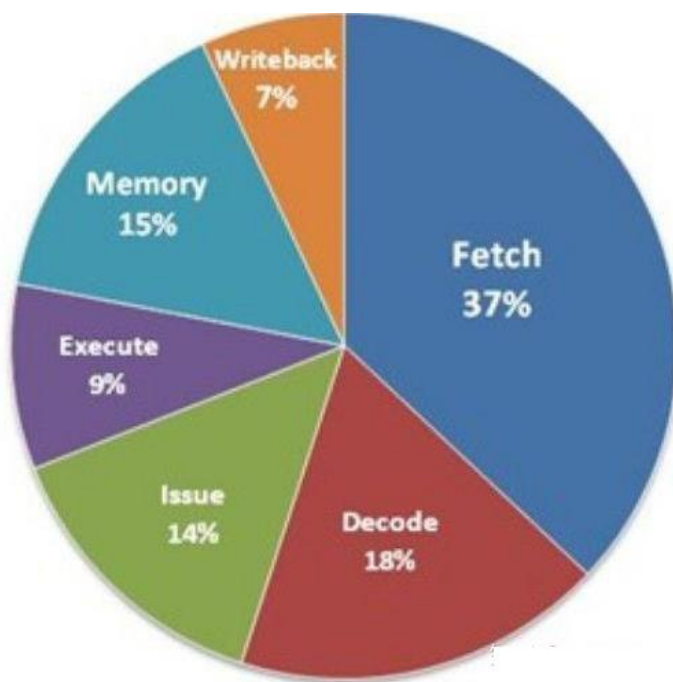
CPU

对于 CPU 来说，单核的微架构和性能已经很难再有大的提升了，只能往 multicores/manycores 发展。如果不远的将来是 1000 CPUs per chip（可能是大小核的），那么如何让软件利用硬件的这种并行度？Berkeley 提出的思路是软硬件一体化考虑。我感觉这个思路有点类似于现在针对神经网络算法重新设计 NPU。Berkeley 总结了 13 大重要软件模型：

1. Dense Linear Algebra
2. Sparse Linear Algebra
3. Spectral Methods
4. N-Body Methods
5. Structured Grids
6. Unstructured Grids
7. MapReduce
8. Combinational Logic
9. Graph Traversal
10. Dynamic Programming
11. Back-track/Branch & Bound
12. Graphical Model Inference
13. Finite State Machine

将这些算法并行化编程后，对 manycores 的互联架构又会带来挑战。在 32cores 上工作很好未必能在 1000cores 上同样工作很好，正如指令级并行到了 4 以后就有挑战一样。

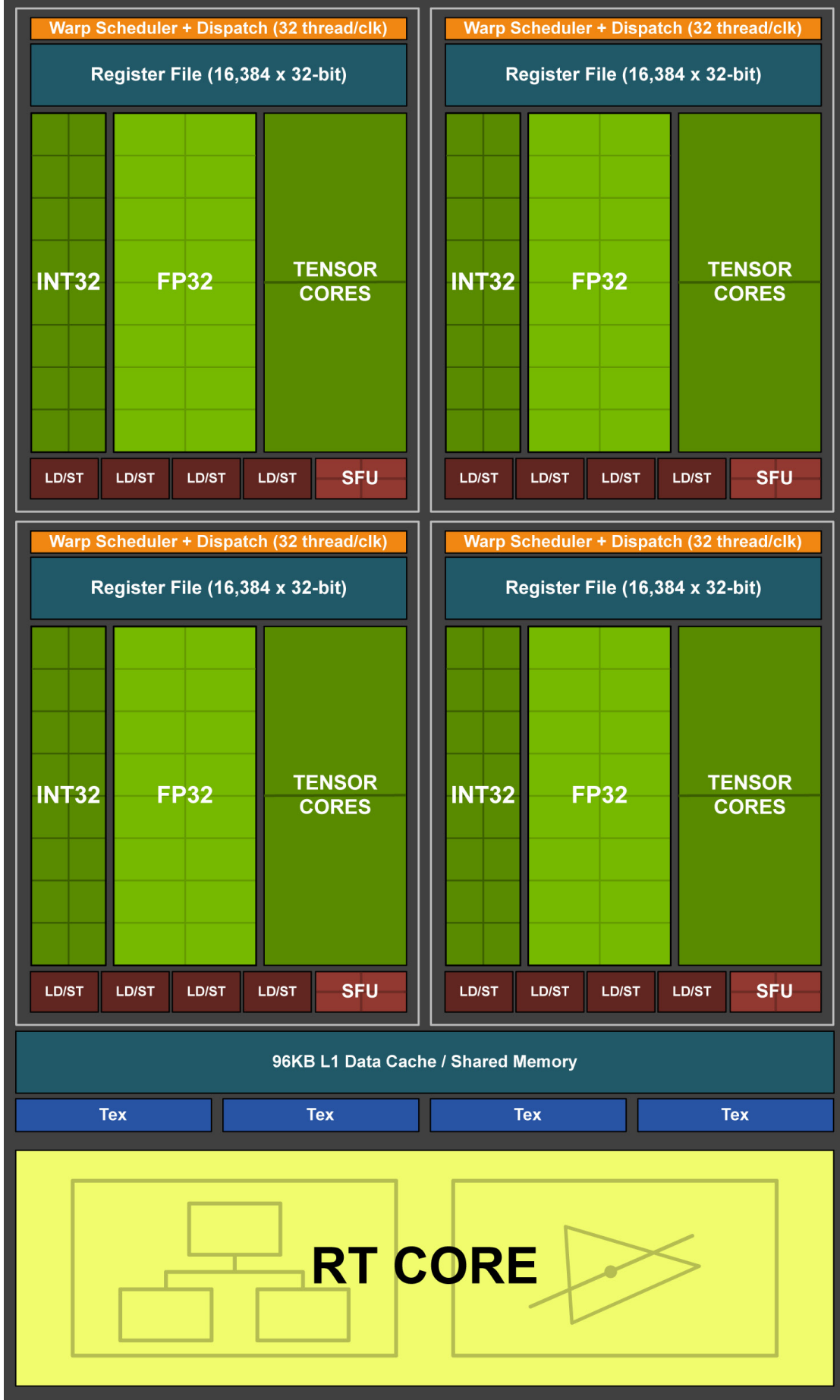
CPU 能耗分布。用 CPU 处理大数据量的简单计算是不高效的。



GPU

Turing 架构的 TU102 GPU， 单个 SM:

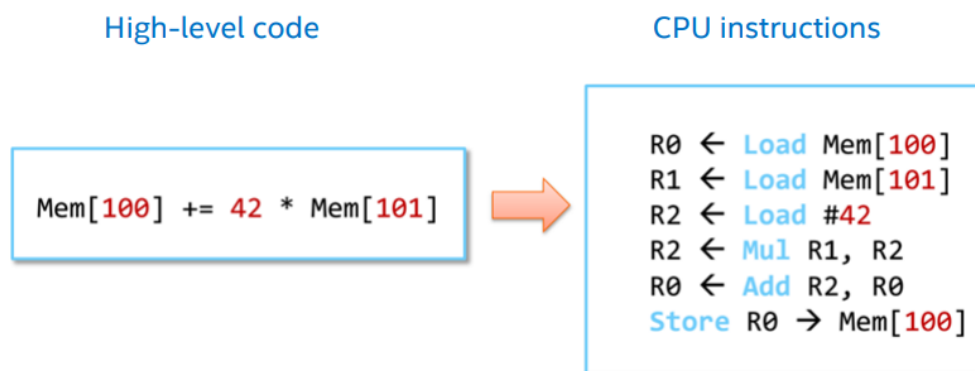
SM



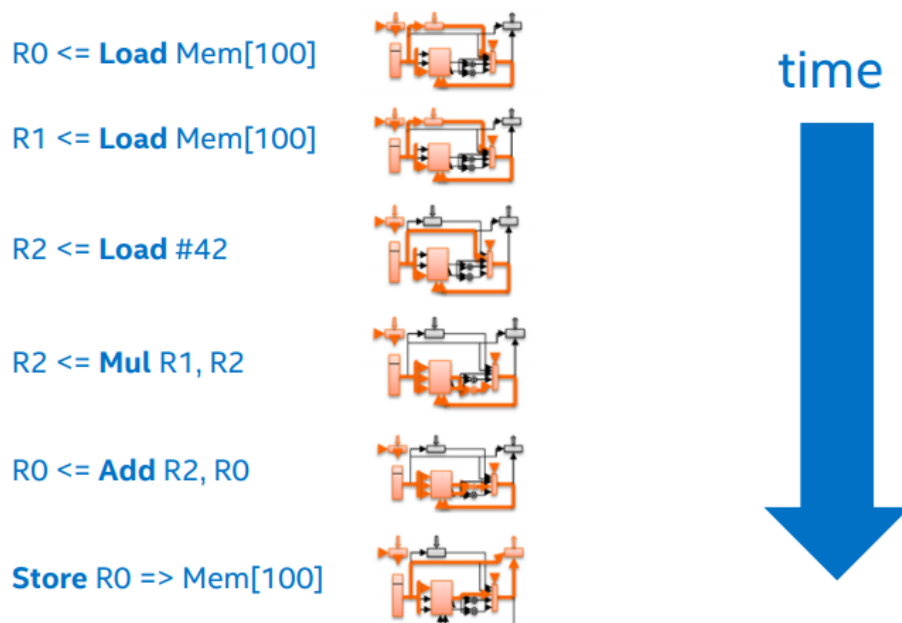
在获取数据之后，在 SM 中以 32 个线程为一组的线程束(Warp)来调度，来开始处理顶点数据。Warp 是典型的单指令多线程（SIMT，SIMD 单指令多数据的升级）的实现，也就是 32 个线程同时执行的指令是一模一样的，只是线程数据不一样，这样的好处就是一个 Warp 只需要一个套逻辑对指令进行解码和执行就可以了。

FPGA

通过对逻辑的整理和固化，干掉了指令。



CPU Activity, step by step

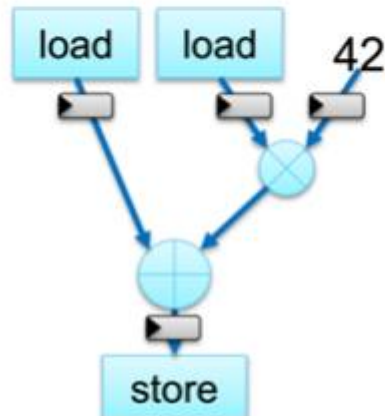


On FPGA:

High-level code

```
Mem[100] += 42 * Mem[101]
```

Custom data-path






AI 芯片

AI 的许多数据处理涉及矩阵乘法和加法。AI 算法，在图像识别等领域，常用的是 CNN；语音识别、自然语言处理等领域，主要是 RNN，这是两类有区别的算法；但是，他们本质上，都是矩阵或 vector 的乘法、加法，然后配合一些除法、指数等算法。

| | 训练 | 推理 |
|----|---|--|
| 云端 | GPU: NVIDIA, AMD FPGA: Intel, Xilinx ASIC: Google | GPU: NVIDIA FPGA: Intel, Xilinx, 亚马逊, 微软, 百度, 阿里, 腾讯 ASIC: Google, 寒武纪, 比特大陆, Wave Computing, Groq |
| 终端 | / | ASIC: 寒武纪, 地平线, 华为海思, 高通, ARM FPGA: 深鉴科技 (Xilinx) GPU: NVIDIA, ARM |

GPU 是有很多个 threads，每个 thread 处理一小部分数据，比如 pixel。AI 处理多是 matrix 运算。

Execution Model Comparison

| | MIMD/SPMD | SIMD/Vector | SIMT |
|-----------------------------|---|---|---|
| |  |  |  |
| Example Architecture | Multicore CPUs | x86 SSE/AVX | GPUs |
| Pros | More general: supports TLP | Can mix sequential & parallel code | Easier to program Gather/Scatter operations |
| Cons | Inefficient for data parallelism | Gather/Scatter can be awkward | Divergence kills performance |

内核

Linux Kernel

讲内核的书已经有很多了。

设备虚拟化

KVM+QEMU

现在虚拟化主要是两个趋势。一个是往下走，用硬件来实现更高性能，省去昂贵 CPU 开销带来的虚拟化税。一个是往上走，用统一的软件接口来管理 bare metal 和 VM。

OS 虚拟化

Container

web 沙盒技术

语言级别虚拟化

JVM

Framework

Android

ROS

Cloud

Applications

Media codec

CNN training

Face recognition

MYSQL

NGINX

robot-MPC

robot-SLAM