INTRODUCTION TO GPU COMPUTING

Holger Fröning, Heidelberg University, holger.froening@ziti.uni-heidelberg.de

Tutorial: **Instrumentation and Modeling of Performance and Power Consumption for Massively Parallel Processors**

HiPEAC Conference 2021, Budapest, Hungary, January 19, 2021

GPU BACKGROUND

- Primary use in gaming
  - Each console has a (powerful) GPU
  - Meantime photorealistic
- Graphics: big, multi-dimensional floating-point operations in parallel
  - Programmable
- Since ~2007 used for general-purpose computing
  - CUDA

NVIDIA

https://80.lv/articles/ue4arch-interview-about-realiscit-environments-in-ue4/

MOTIVATING STATEMENT

“GPUs are simply crippled processors”

– known (sane) scientist, 2013

PERFORMANCE SCALING

$$Perf(\frac{ops}{s}) = \underbrace{\frac{Instructions}{cycle}}_{\propto PipelineCount \cdot PipelineDepth} \cdot \underbrace{frequency}_{\text{scales with feature size}}$$

CLASSICAL DENNARD SCALING

$$Perf(\frac{ops}{s}) = \underbrace{Power(W)}_{\text{fixed}} \cdot Efficiency(\frac{ops}{Joule})$$

POST DENNARD SCALING

REGIME I

operator cost

Specialization —> heterogeneity and asymmetry

GPU COMPUTING

PROGRAMMING COMPLEXITY

NUMA EFFECTS & LATENCY HIDING

+

REGIME II

data movement cost

3 operands x 64bit/operand

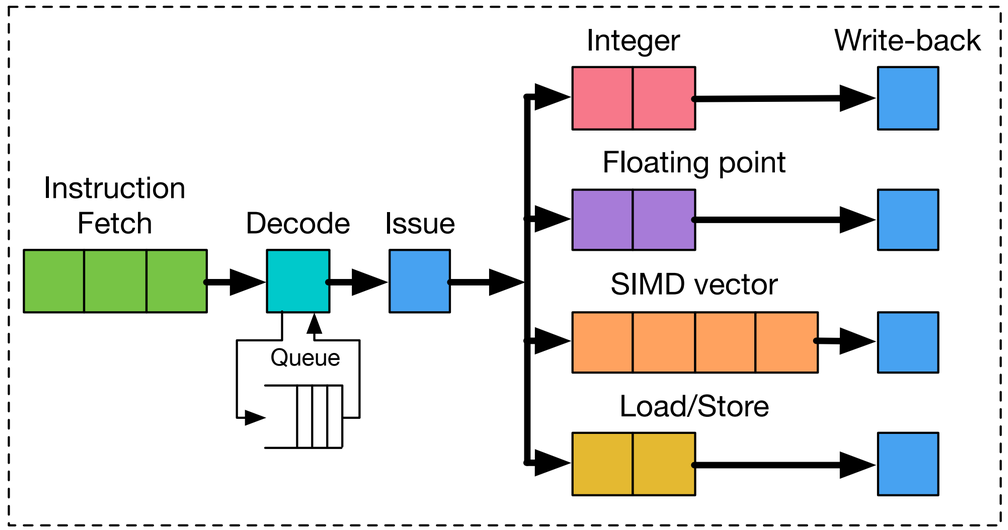
$$Energy = \#bits \cdot dist[mm] \cdot energy_{per\ bit, per\ mm}[\frac{J}{mm}]$$

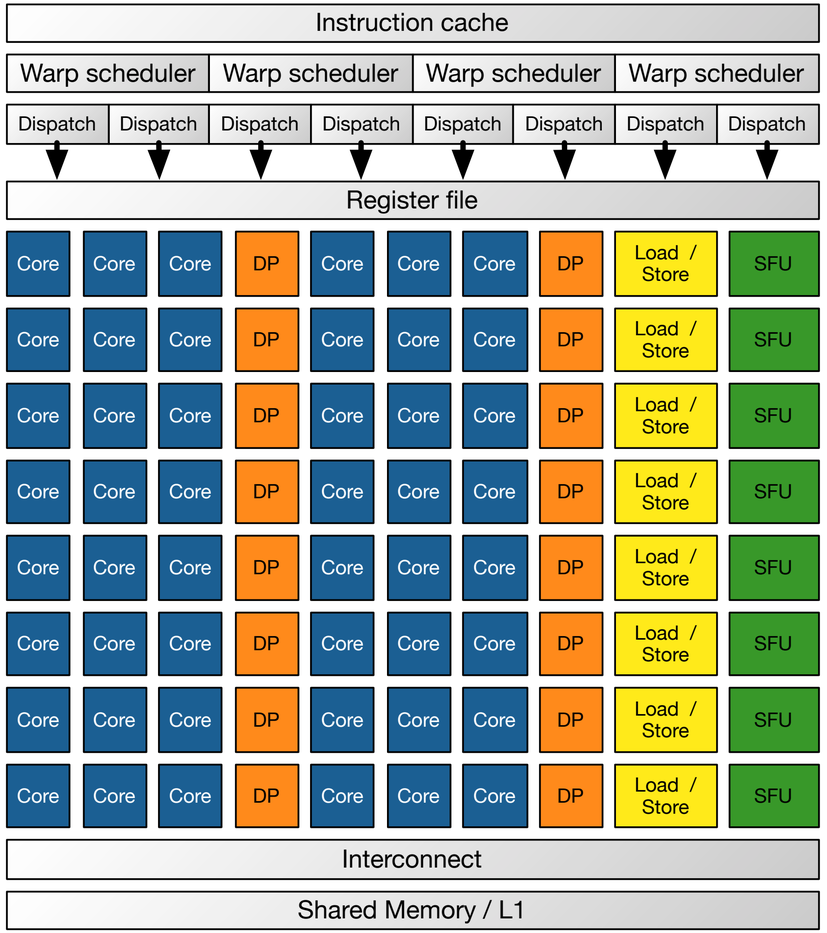
Partly by Bill Dally, Sudha Yalamanchili (UCAA Workshop, 2012)

POST-DENNARD: TRANSITION TO MASSIVELY PARALLEL MICROARCHITECTURES

$$P = afcV^2 + VI_{leakage} \propto f^3$$

$$\propto f$$





Frequency reduction
In-order pipelines

Replication

Massively parallel
Energy efficient

REMINDER: BULK-SYNCHRONOUS PARALLEL

In 1990, Valiant already described GPU computing pretty well

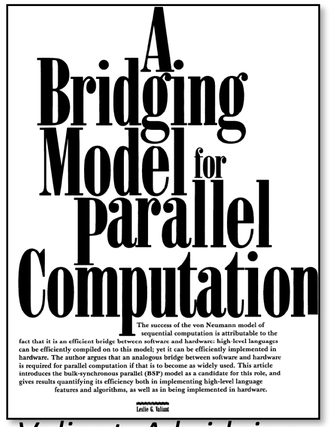
Superstep

- Compute, communicate, synchronize

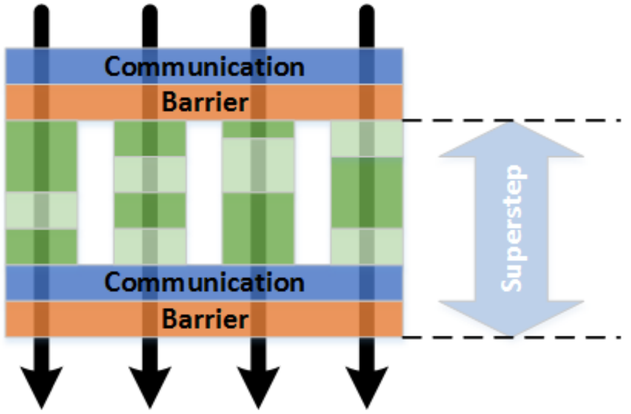
Parallel slackness: # of virtual processors v, physical processors p

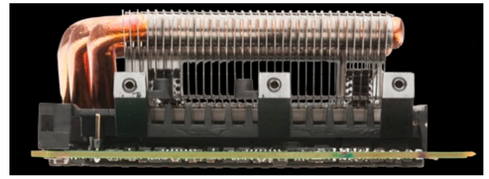
- v = 1: not viable
- v = p: unpromising wrt optimality
- v >> p: leverage slack to schedule and pipeline computation and communication efficiently

Extremely scalable, bad for unstructured parallelism



Leslie G. Valiant, A bridging model for parallel computation, Communications of the ACM, Volume 33 Issue 8, Aug. 1990





OUR VIEW OF A GPU

Software view: a programmable many-core scalar architecture

- Huge amount of scalar threads to exploit parallel slackness, operates in lock-step
- SIMT: single instruction, multiple threads

IT'S A (ALMOST) PERFECT INCARNATION OF THE BSP MODEL

Hardware view: a programmable multi-core vector architecture

- SIMD: single instruction, multiple data
- Illusion of scalar threads: hardware packs them into compound units

IT'S A VECTOR ARCHITECTURE THAT HIDES ITS VECTOR UNITS

GPU HARDWARE VIEW

Scalar threads are an illusion

- GPU HW bundles threads into warps
- Warps run in lockstep on vector-like hardware (SIMD)

SIMT: single instruction, multiple threads

- All threads of one warp share one instruction stream
- Hardware executes SIMD operations
- Static grouping: G200, Fermi, Kepler, Maxwell, Pascal
- Dynamic grouping: Volta, Ampere

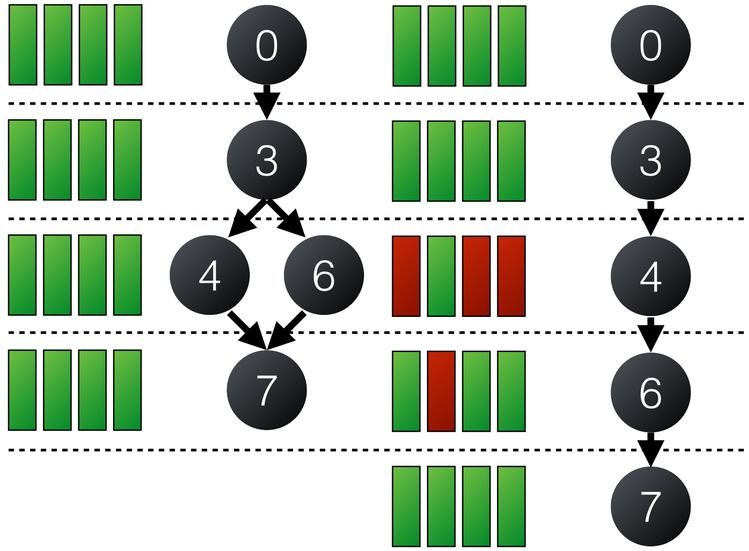
Branches are handled using masks

- Branch divergence problem

(Post-)Volta: independent thread scheduling

- https://developer.nvidia.com/blog/using-cuda-warp-level-primitives/
- Escape: warp-level primitives in their sync-variant, thread groups, compile for Pascal (`-arch=compute_60 -code=sm_70`)

```
0: __global__ kernel1 (...)
1: {
2:   id = threadIdx.x;
3:   if ( id == 1 )
4:     result = foo1();
5:   else
6:     result = foo2();
7:   return result;
}
```



	Xeon E5-2699v4 (Broadwell, 2016)	Tesla K20 (GK110) (Kepler, 2012)	NVIDIA P100 (GP100) (Pascal, 2016)	NVIDIA V100 (GV100) (Volta, 2017)
		~3-4	~1.5	
Core count	22 cores 2 FP-ALUs/core	13 SMs 64/SM(DP), 192/SM(SP)	56 SMs 32/SM(DP), 64/SM(SP)	84 SMs 32/SM(DP), 64/SM(SP)
Frequency	2.2-3.6GHz	0.7GHz	1.328-1.480GHz	1.455GHz
Effective vector width	256bit (SP/DP) AVX 2.0	1024bit (SP), 2048bit (DP) static grouping		1024bit (SP), 2048bit (DP) dynamic grouping
Peak Perf.	633.6 GF/s (DP)	1,165 GF/s (DP), SP x3	5.3 TF/s (DP), SP x2	7.5 TF/s (DP), SP x2
Use mode	latency-oriented		throughput-oriented	
Latency	minimization		toleration	
Programming	10s of threads		10,000s+ of threads	
Memory bandwidth	76.8 GB/s 128bit DDR4-2400	250 GB/s 384-bit GDDR-5	720 GB/s 4096-bit HBM2	
Memory	1.54TB	5 GB	16G	32G
Die size	456 mm^2	550mm^2	610mm^2	815mm^2
Transistor	7.2 billion	7.1 billion	15.3 billion	21.1 billion
Technology	14nm	28nm	16nm FinFET	12 nm FFN
Power	145W	250W	300W	300W
Power efficiency	4.37 GF/Watt (DP) 8.74 GF/Watt (SP)	4.66 GF/Watt (DP) 14 GF/Watt (SP)	17.66 GF/Watt (DP) 35 GF/Watt (SP)	25 GF/Watt (DP) 50 GF/Watt (SP)

SUMMARY

GPUs as one of the most promising compute options today

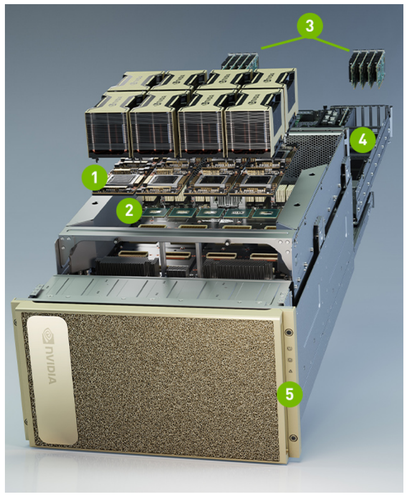
- Fast & energy-efficient

Diversity of GPUs (note our focus on NVIDIA GPUs)

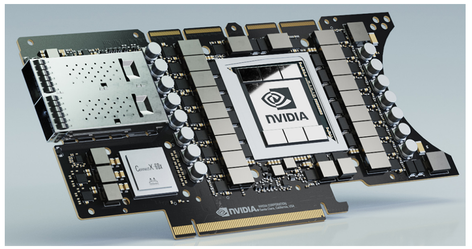
- Different classes (server, workstation, mobile)
- Different specializations (gaming, ML training, ML inference, HPC)
- Different generations (performance scaling?)

One of GPU Mangrove's main objectives is to shed light on this diversity

NVIDIA RTX2080
250W

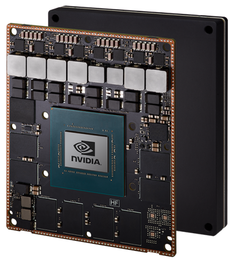


NVIDIA DGX A100
6.5kW



NVIDIA EGX

NVIDIA Jetson Nano
(5-10W)



NVIDIA Jetson AGX Xavier
10-30W