GPU Architecture
(not covered on exam)

Jon Beaumont

Announcements

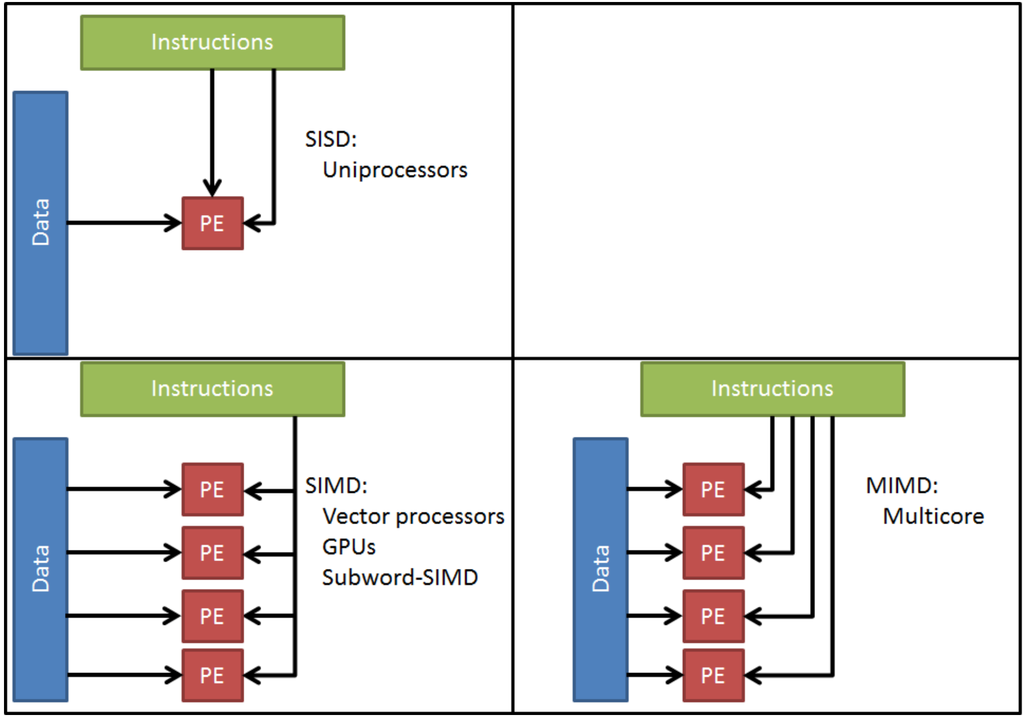
- Lab
  - Assignment due Wednesday
  - Canvas quiz by Thursday
  - Meet Fr/M
- Project 3
  - Checkpoint due Thursday – 5%
  - Full project due next Thursday
- HW 3
  - Out later today
  - ~2 weeks to finish

Data-Level Parallelism (DLP)

- Multiple instances of instructions that operate on different data
- Usually found across iterations of a “for” loop

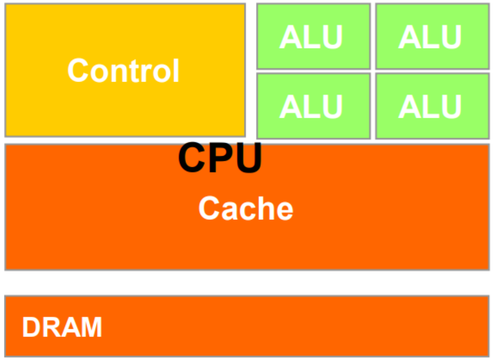
```
for (I = 0; I < 100; I++)
   Z[I] = A*X[I] + Y[I];
```

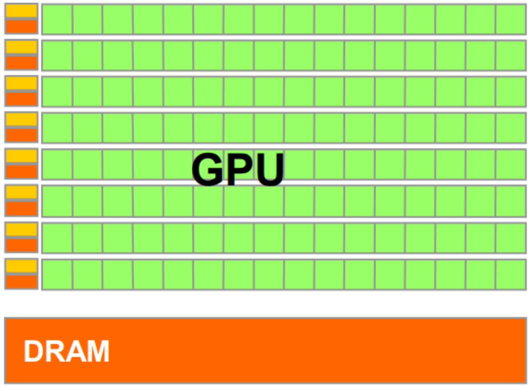
SIMD Methodology



SIMD Methodology

- Fetch / decode / schedule an instruction once
- Execute it several times
- Allows for much more of the die space to be dedicated to ALUs

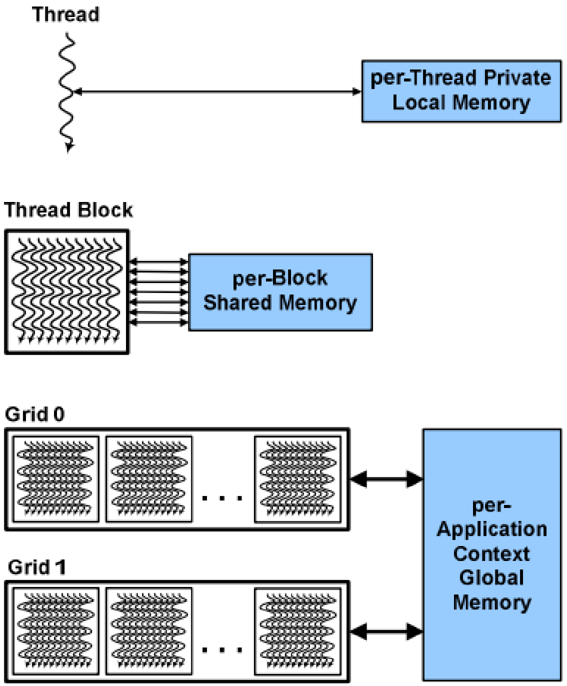




CUDA Programming Model

```
// CPU algorithm: sums two vectors sequentially

for(int a=0; a<size; a++)

        array1[a]+=array2[a];

// GPU algorithm: sums two vectors in parallel
__global__ void AddInts(int *a, int *b, int size) {
        int id = blockIdx.x * blockDim.x + threadIdx.x;
        if(id < size) {
                a[id] += b[id];
        }
}                    # thread blocks    threads / block
AddInts <<< ceil(size / 256),       256        >>>(a, b, size);
```

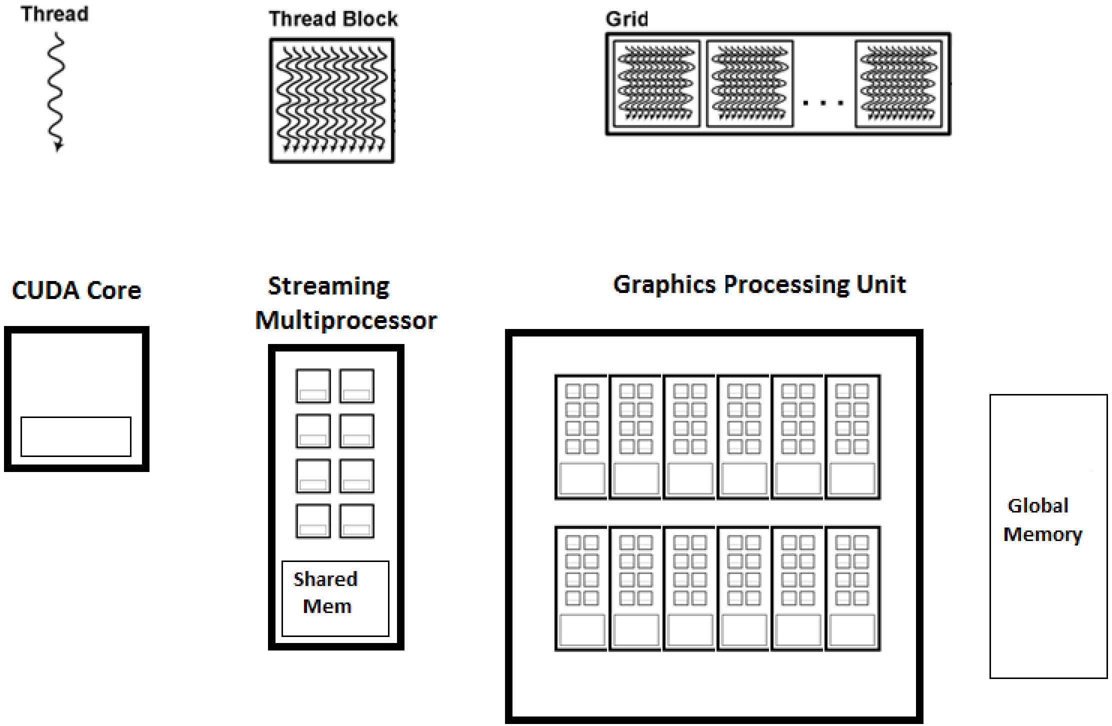
CUDA Programming Model



SIMT Model

- Assumption: there are millions of elements in a vector to apply an operation to
    - Treat each element as a "thread"
- Fetch an instruction to operate on a ~10-100 elements at a time
    - SIMD
- Start fetching next instruction right away
    - Pipelining
- On the next cycle, fetch from a different bundle of 32 threads
    - That way, cache misses won't stall the program
    - Multi-threading
- Include 100s of these SIMD cores to execute different thread blocks
    - Multi-processing
- Result is "Single-Instruction-Multiple-Thread (SIMT)" computing

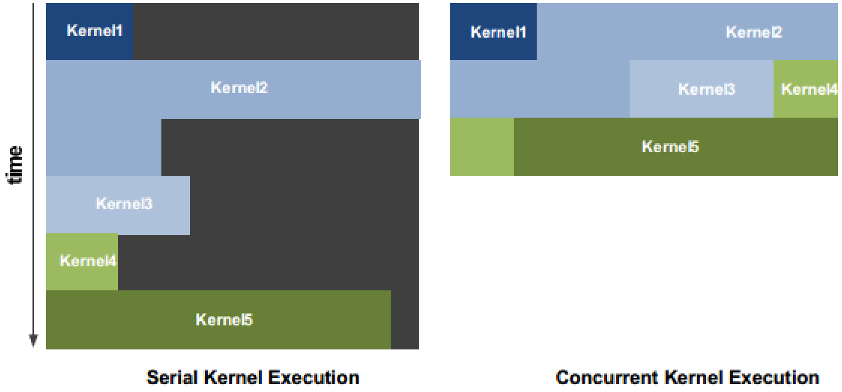
CUDA Hardware Hierarchy



CUDA Architecture

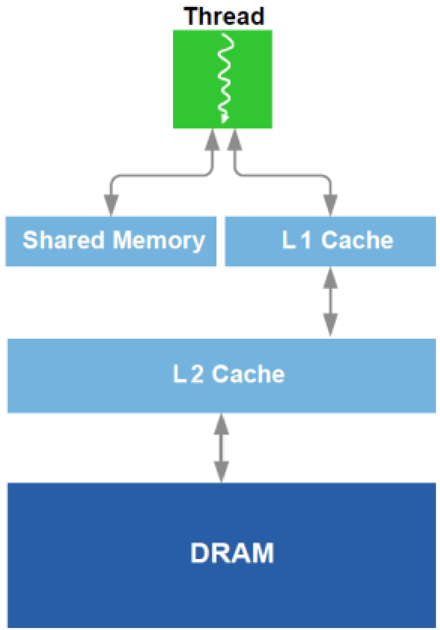
GigaThread Scheduler

- Concurrent Kernel Execution



- 10x Faster Context Switching between applications

True Cache Hierarchy



Shared Memory / L1 Cache

- Shared memory enables threads within same thread block to share data
  - Software controlled scratch pad
  - Typical in fairly deterministic GPU applications
- L1 beneficial for non-deterministic memory accesses

General-Purpose GPUs

- GPUs were originally designed to accelerate graphics workloads
- Millions of vertexes on a screen
  - Most are independent, can be calculated as separate thread
- But tons of workloads match this level of parallelism
- GPUs are designed to be more "general purpose"

Irregular Parallelism

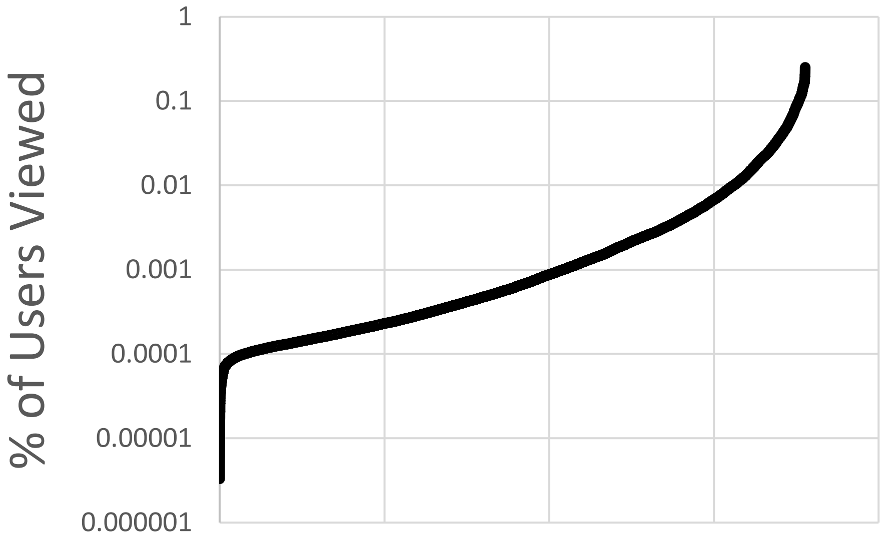
- Some “compute” operator performed over large elements of data
  - Data-level parallelism
- Which elements depends on specific structure
  - “Irregular” or “amorphous”
- Can often be represented as a graph
  - Breadth first search, n-body simulation, SGD

Irregular Data

- Real world data sets becoming increasingly "sparse"
  - Fewer relative connections between data
- E.g. number of Facebook users grew 19.8% from 2017 to 2018
  - Average number of "friends" per user grew 11.2%
  - Average density of "friend" graph decreases by 7.2%

Irregular Data

- E.g. number of Netflix users who've viewed content

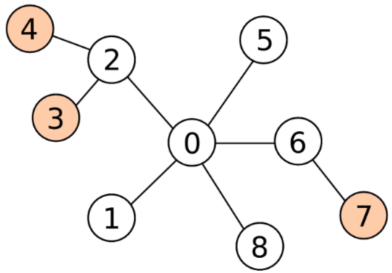


Irregular Algorithm Implementations

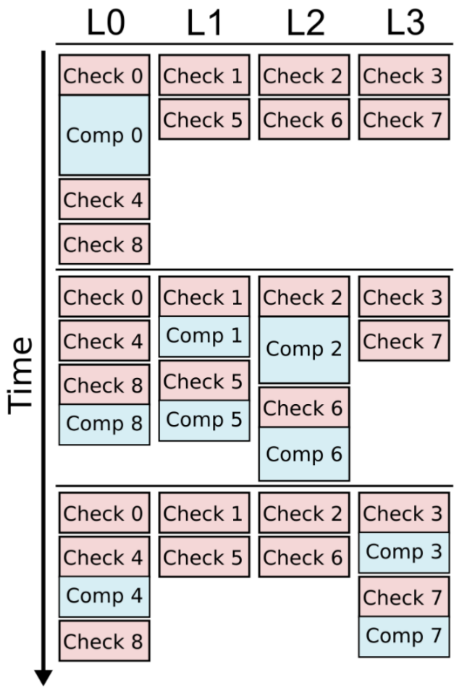
Two common approaches:

1. Topology-driven:
    - Every node is processed on each iteration until completion
2. Data-driven:
    - Active nodes are placed in a worklist, only visited when useful work to be done

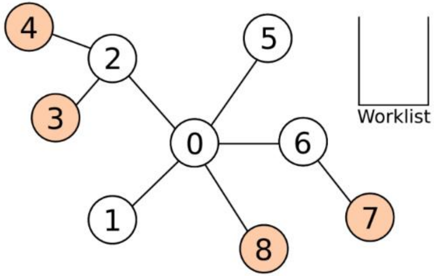
Topology-Driven



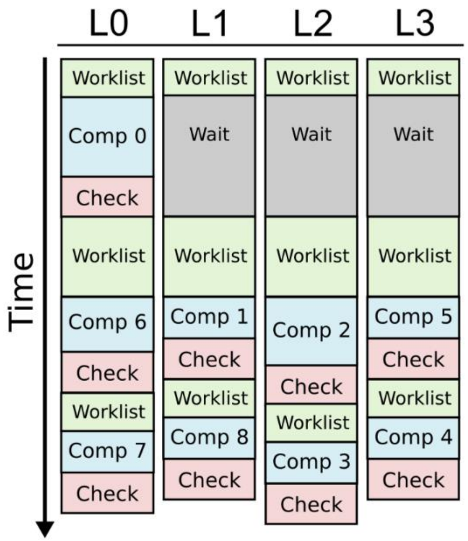

```
__global__ void topo(Node *nodes, bool *done) {
  Node node = nodes[threadIdx];
  if( node.active() ) {
    node.process();
    *done = false;
  }
}

int main() {
  //...
  while (!finished) {
    done = true;
    topo<<<N>>>(nodes, &done);
  }
}
```



Data-Driven



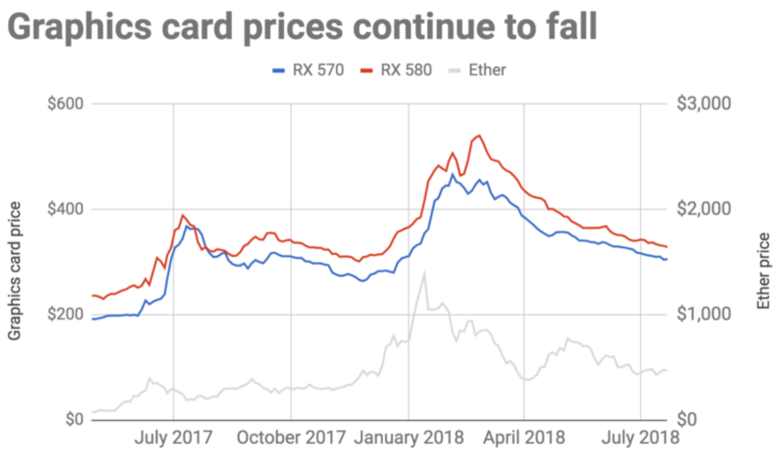

```
__global__ void data(Node *nodes, WL *wl)
  while(idx = wl->pop()) {
    Node node = nodes[idx];
    node.process();
    for(i=0; i<node.num_neighbors; i++) {
      wl->push(node.neighbor(i));
    }
  }
}

int main() {
  //...
  init<<<N>>>(nodes, wl);
  data<<<M>>>(nodes, wl);
}
```



Why Have GPUs Been in the News?

- GPUs surged in popularity during the crypto-boom
- Cryptomining
  - A transaction is verified by someone performing "proof-of-work"
  - Basically, reversing a hash function
  - Whoever reverse-hashes first gets rewarded with currency
- GPUs can attempt multiple reverse-hashes simultaneously
- Power efficiency is more important than raw throughput
  - Easy to spend more money on electricity than what you earn

GPU Costs



Credit: Sam Huitfeld, Viperlair

Why Have GPUs Been in the News?

- AI Boom
  - Spurred by bots like ChatGPT
- Large language models, deep learning, other AI/ML techniques rely on processing massive amounts of data
  - Language of linear algebra
  - Massive vectors and matrices
- GPUs can provide huge benefits over CPUs

Will We See Another GPU Shortage?

PCWorld NEWS BEST PICKS REVIEWS

HOT TOPICS Black Friday Deals Best laptops Best VPN

When you purchase through links in our articles, we may earn a small commiss

Home / PCs & Components / News

NEWS

The AI boom could create a new crypto-style GPU shortage

Reports of huge GPU purchases from AI companies big and small are causing fears

REUTERS World Business Markets Sustainability Legal Breakingviews Technology

Technology

Nvidia bets $25 bln that AI boom is far from over

By Stephen Nellis and Max A. Cherney

August 24, 2023 2:14 PM EDT · Updated 3 months ago

tom'sHARDWARE US Edition

Reviews Best Picks Raspberry Pi CPUs GPUs Coupons New

TRENDING Best Black Friday Tech Deals Core i9-14900K Lowest CPU Pri

When you purchase through links on our site, we may earn an affiliate commission. Here's how it works.

Tech Industry > Artificial Intelligence

Evidence Shows AI-Driven Companies Are Buying up Gaming GPUs

News By Mark Tyson published August 01, 2023

Hopefully we can avoid a repeat of the cryptomining-GPU situation.

ars TECHNICA BIZ & IT TECH SCIENCE POLICY CARS GAMING

THE KING OF MATRIX MULTIPLICATION —

For Nvidia, it's AI or bust as it reports a record-breaking quarter

Everybody wants GPUs for AI, and that's making Nvidia very happy (and rich).

BENJ EDWARDS - 8/24/2023, 3:03 PM

CNBC MARKETS BUSINESS INVESTING TECH POLITICS CNBC TV INVESTING CLUB PRO

TECH

ChatGPT and generative AI are booming, but the costs can be extraordinary

PUBLISHED MON, MAR 13 2023·8:58 AM EDT | UPDATED MON, APR 17 2023·2:09 AM EDT

Jonathan Vanian Kif Leswing

Wanna Learn More?

- EECS 570 – Parallel computer architecture
  - Learn more about how GPUs are designed (one topic in the class)
- EECS 471 – Applied Parallel Programming with GPUs
  - How to efficiently program these things