

Neural networks and deep learning



Machine learning overview

Kun Suo

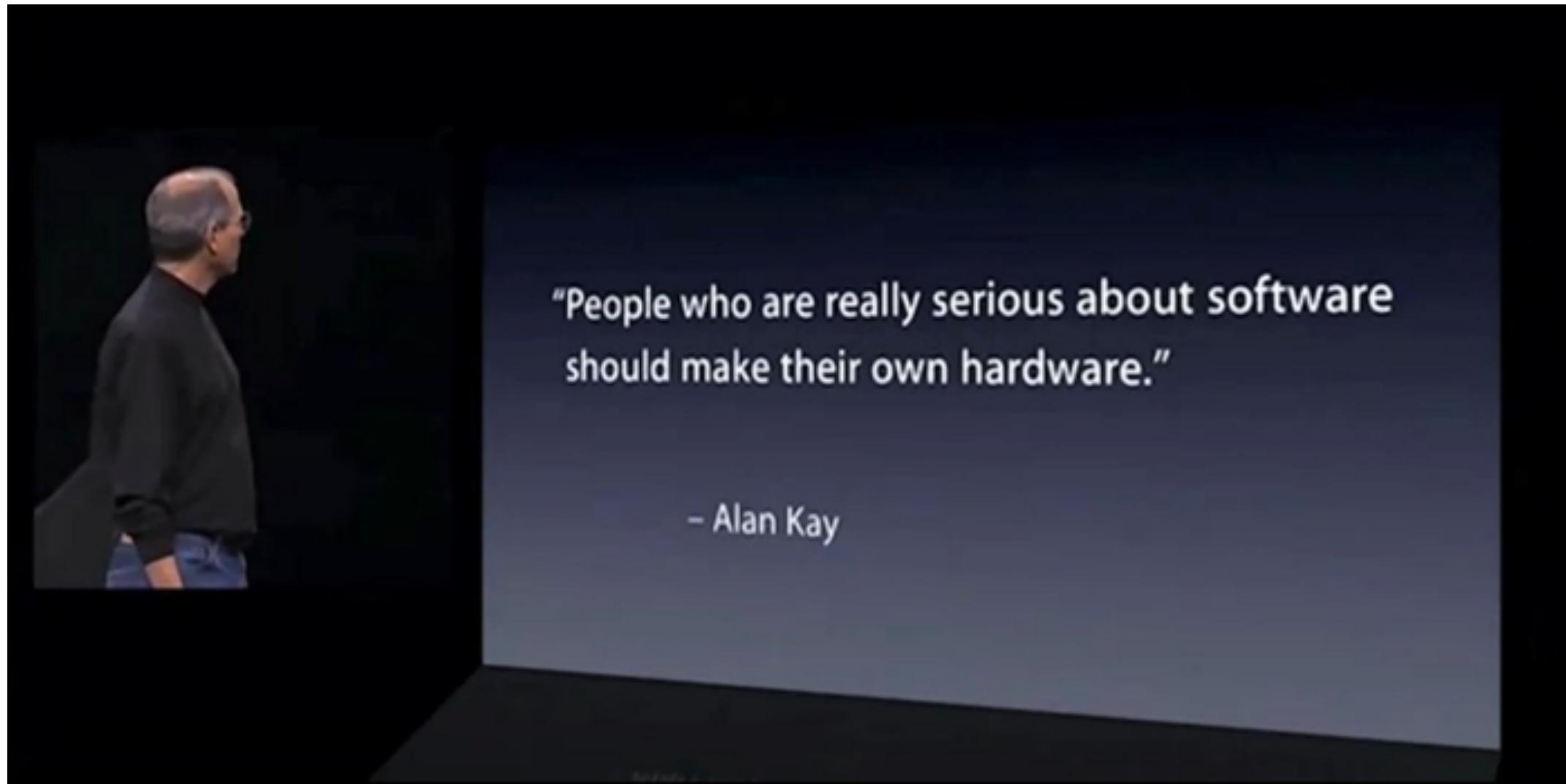
Computer Science, Kennesaw State University

<https://kevinsuo.github.io/>



Hardware

Computer Hardware Review



Moore's law

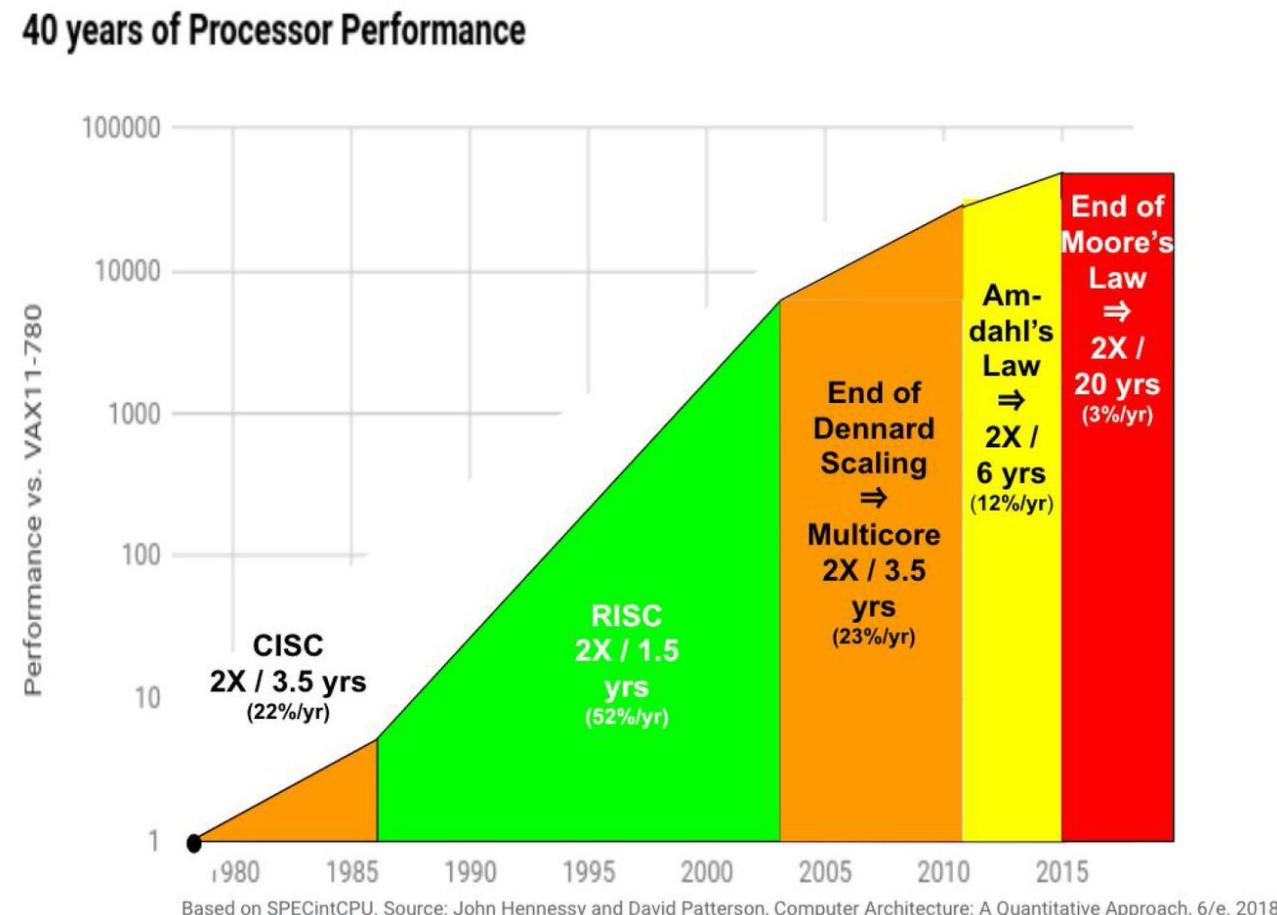
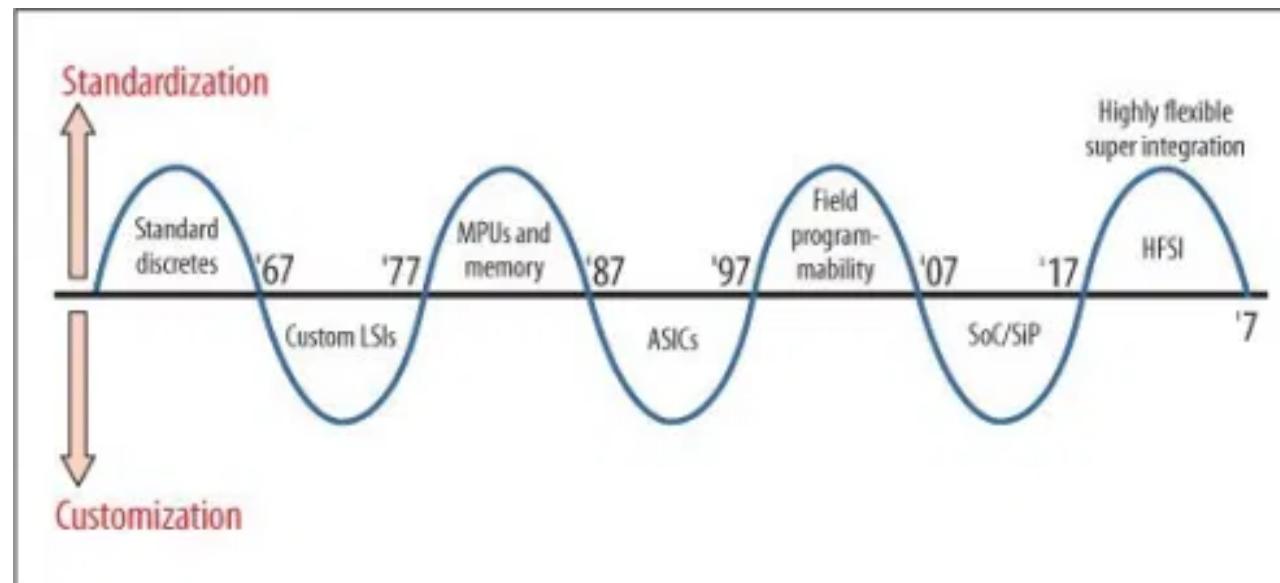


Figure 2: Computing Performance in the Moore's Law and the Post-Moore's Law Periods

- ▶ Performance efficiency v.s. Development efficiency, a pair of goals that cannot be met at the same time
- ▶ Special hardware → performance priority
- ▶ General structure → development efficiency priority

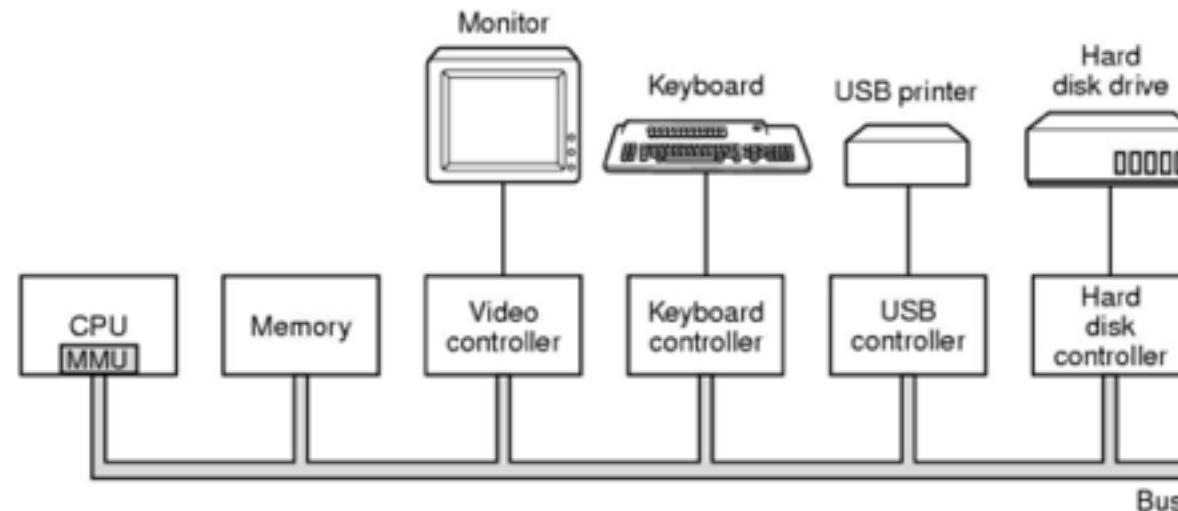


Artificial intelligence stack



Computer Hardware Review

► Basic components of a simple personal computer

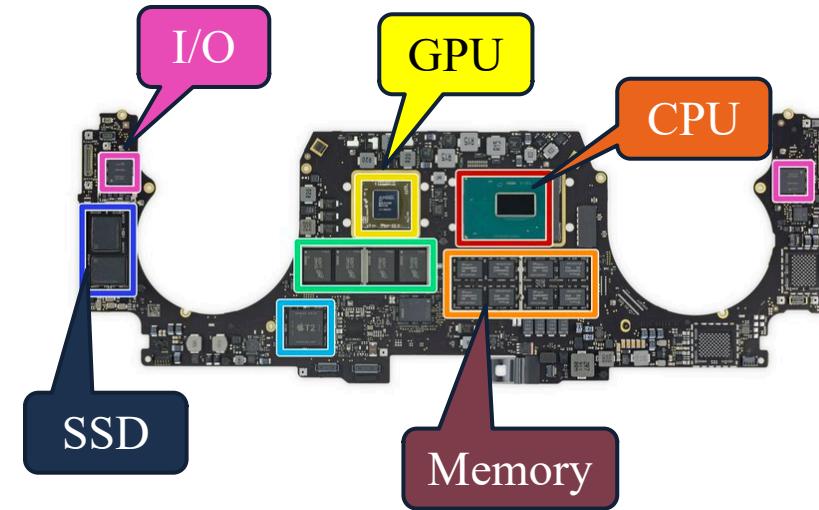


- CPU: data processing
- Memory: volatile data storage
- Disk: persistent data storage
- NIC: inter-machine communication
- Bus: intra-machine communication

Computer Hardware Review

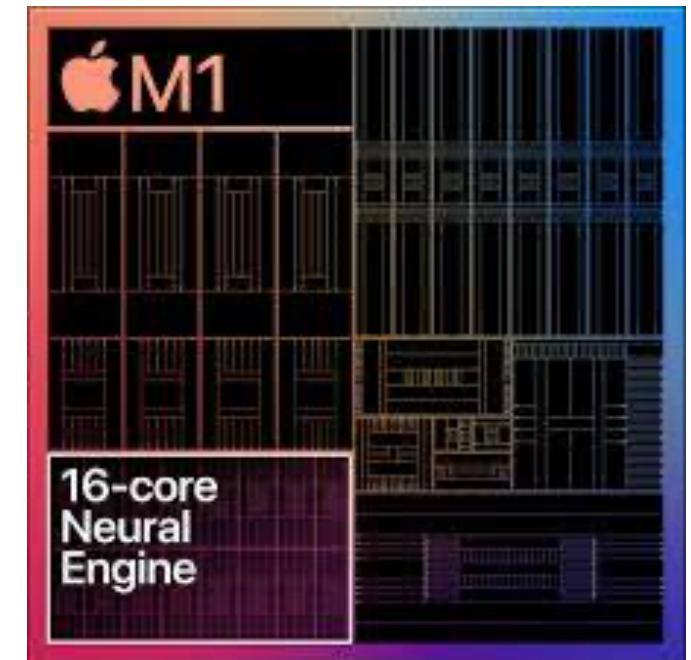
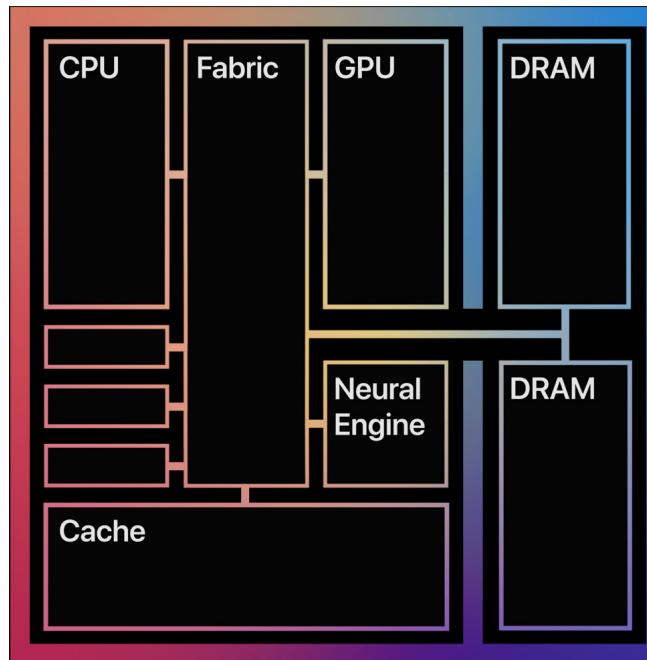
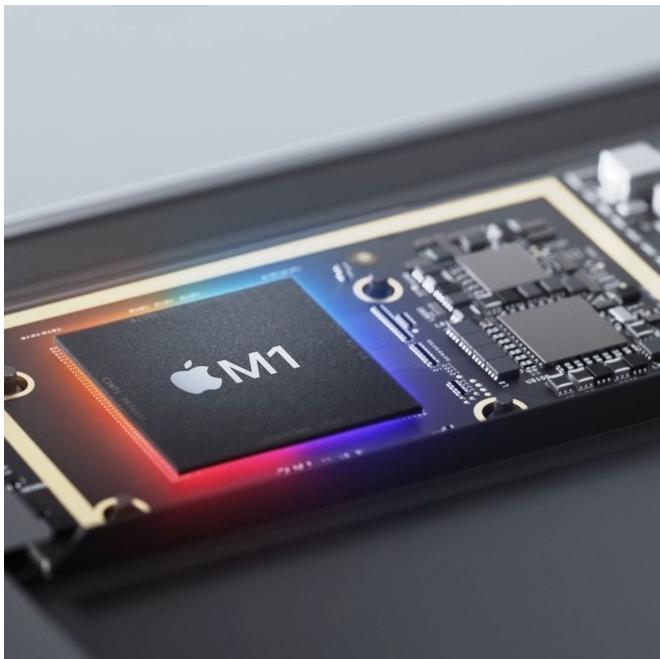


Computer Hardware Review



Computer Hardware Review

► Apple M1 chip



Central Processing Unit (CPU)

▶ Components

- Arithmetic Logic Unit (ALU) -> Compute and data
- Control Unit (CU) -> control device and system

▶ Clock rate

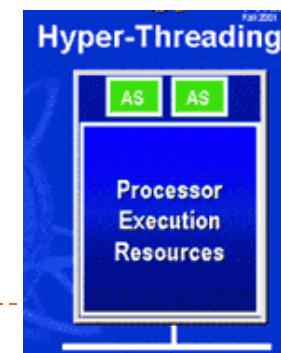
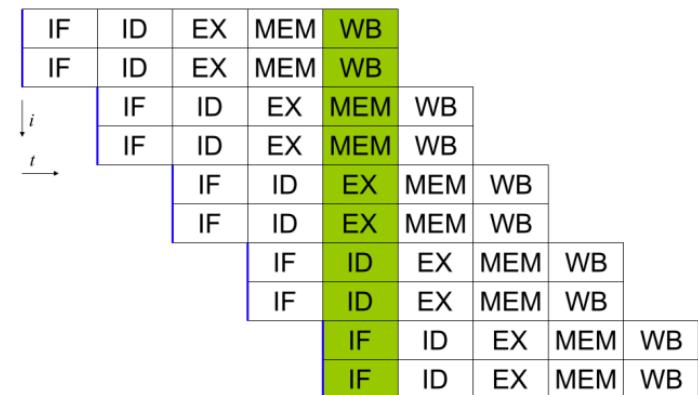
- The speed at which a CPU is running

▶ Data storage

- General-purpose registers: EAX, EBX ...
- Special-purpose registers: PC (program counter), SP (stack), IR (instruction register) ...

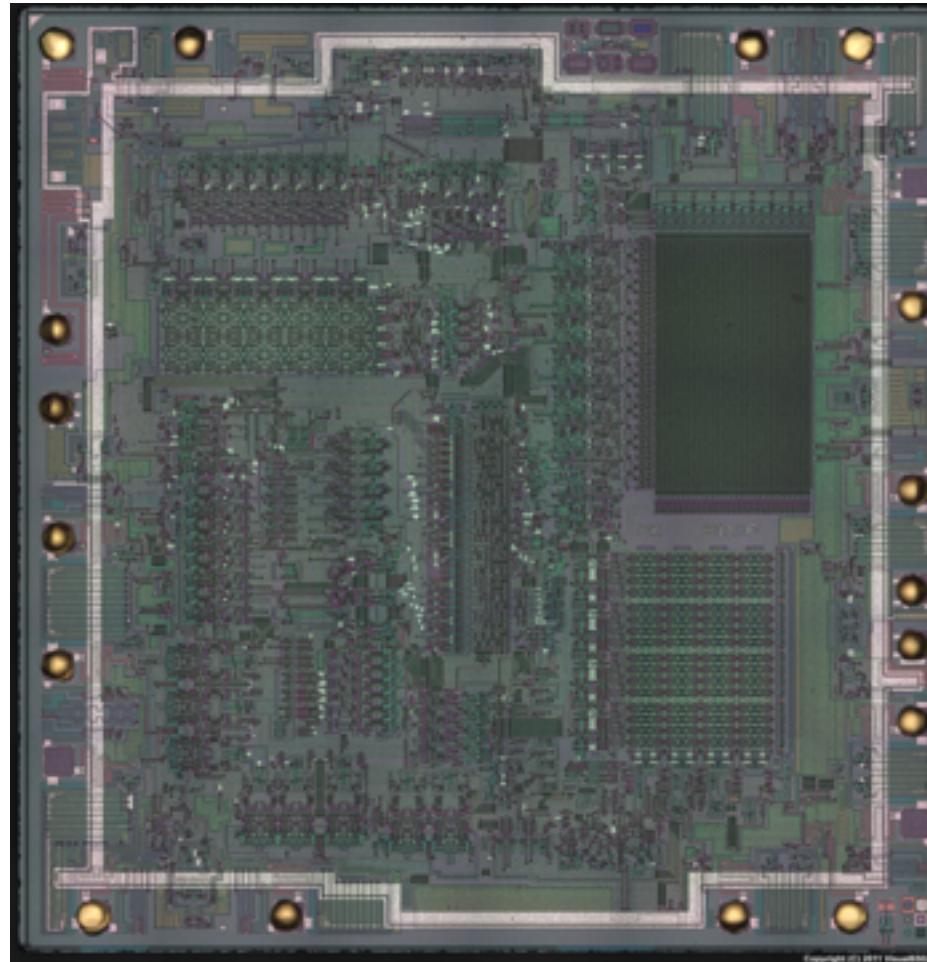
▶ Parallelism

- Instruction-level parallelism
- Thread-level parallelism
 - Hyper-threading: duplicate units that store architectural states
 - Replicated: registers. Partitioned: ROB, load buffer...
Shared: reservation station, caches



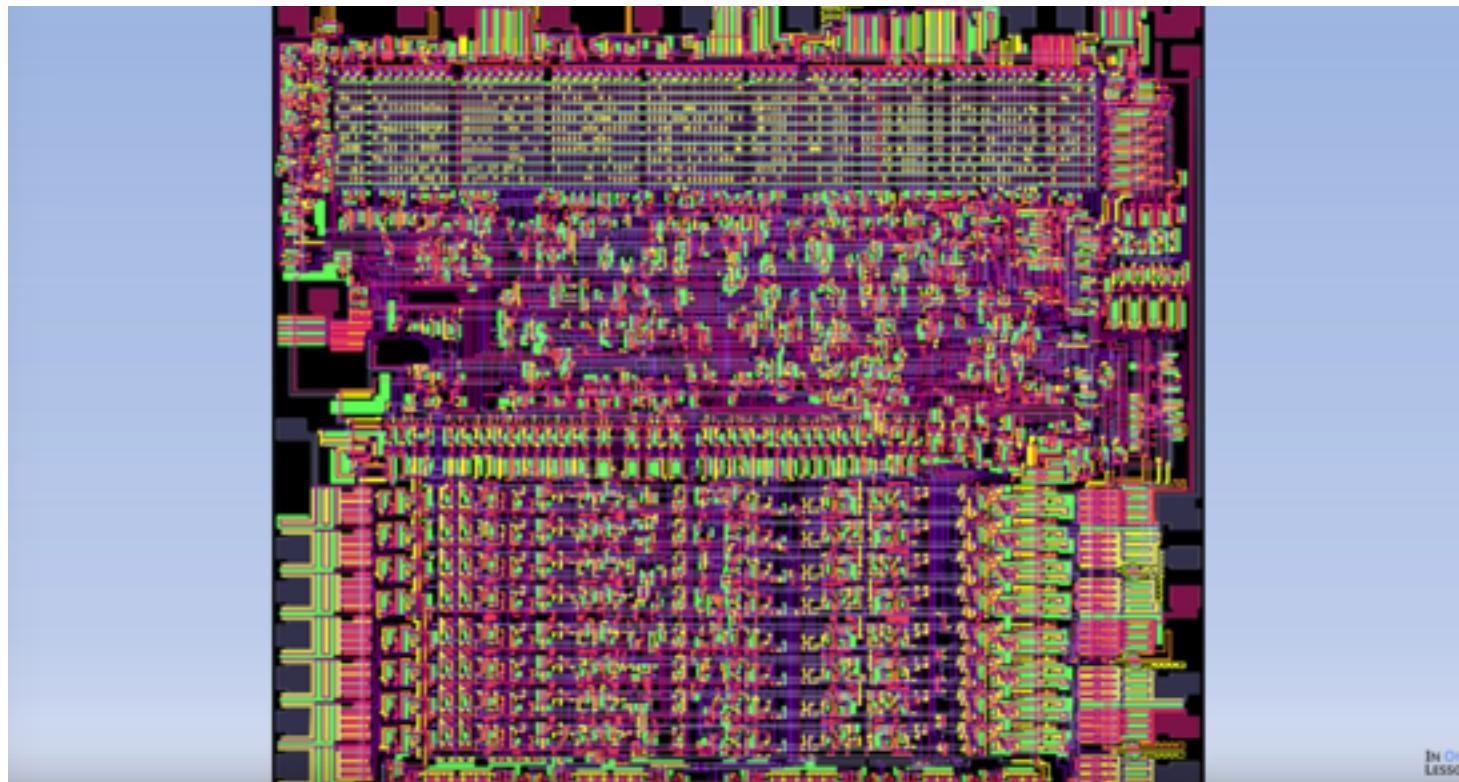
What's inside of CPU?

- ▶ <http://www.visual6502.org/>

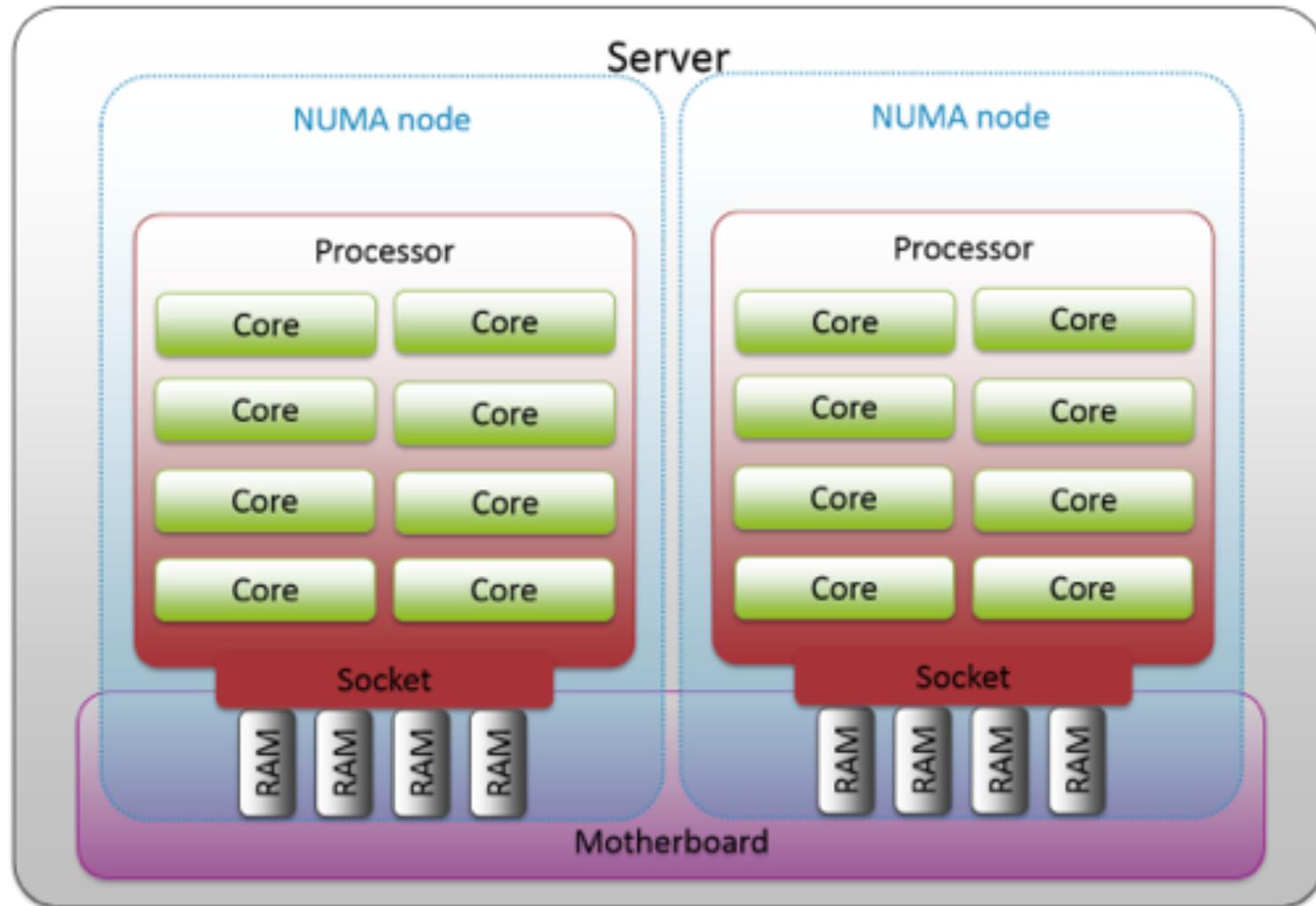


How CPU works?

- ▶ https://youtu.be/cNN_tTXABUA?t=494



NUMA node vs Socket vs Core relationship



CPU information

► lscpu

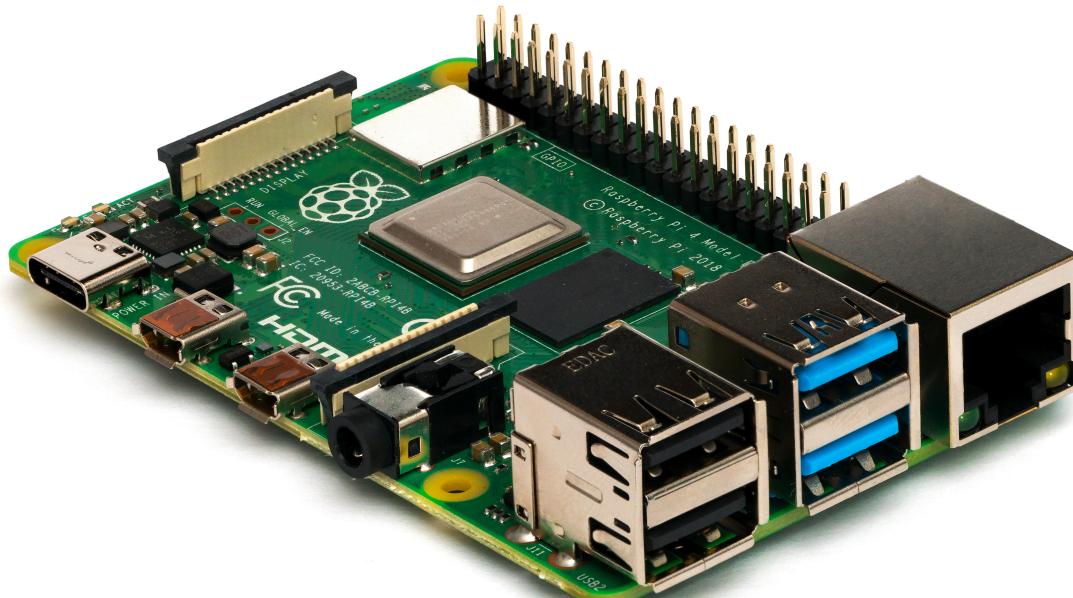


```
administrator@ubuntuvm-1604 ~> lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                2
On-line CPU(s) list:  0,1
Thread(s) per core:   1
Core(s) per socket:   1
Socket(s):             2
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 79
Model name:            Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz
Stepping:               1
CPU MHz:               2199.998
BogoMIPS:              4399.99
Hypervisor vendor:    VMware
Virtualization type:  full
L1d cache:             32K
L1i cache:             32K
L2 cache:               256K
L3 cache:               51200K
NUMA node0 CPU(s):    0,1
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep m
all nx pdpe1gb rdtscp lm constant_tsc arch_perfmon nopl xtopology ts
id sse4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave av
lt invpcid_single pti ssbd ibrs ibpb stibp fsgsbase tsc_adjust bmi1 i
flush_l1d arch_capabilities
```

CPU information

► lscpu

```
pi@raspberrypi:~ $ lscpu
Architecture:          armv7l
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:  0-3
Thread(s) per core:   1
Core(s) per socket:   4
Socket(s):             1
Vendor ID:             ARM
Model:                 3
Model name:            Cortex-A72
Stepping:               r0p3
CPU max MHz:           1500.0000
CPU min MHz:           600.0000
BogoMIPS:              108.00
Flags:                 half thumb fastmult vfp edsp neon vfpv3 tls vfpv4 idiva idivt vfpd32 lpae evtstrm crc32
```



CPU information

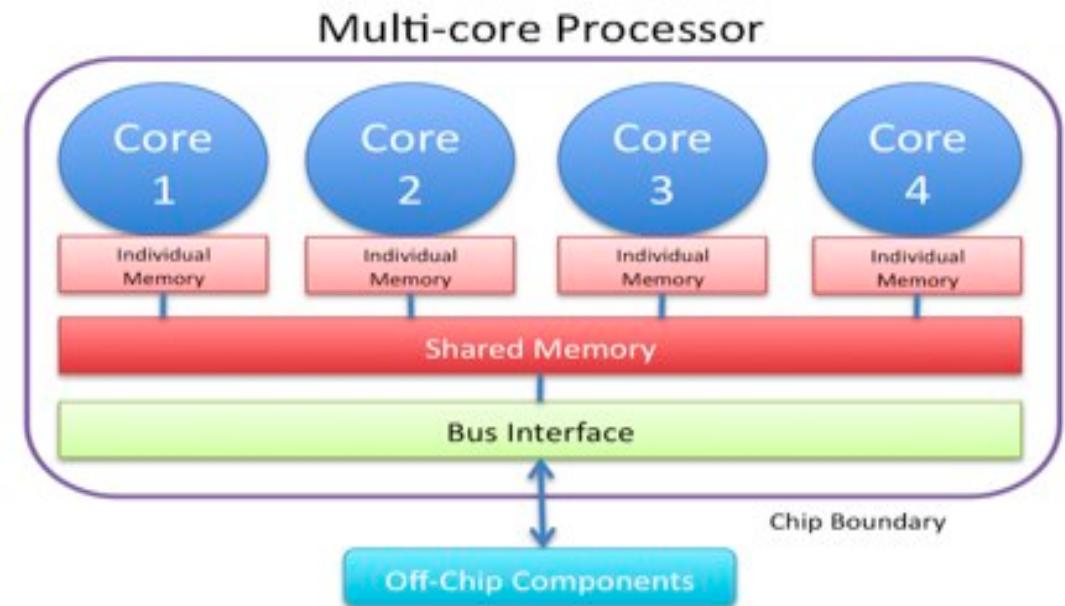
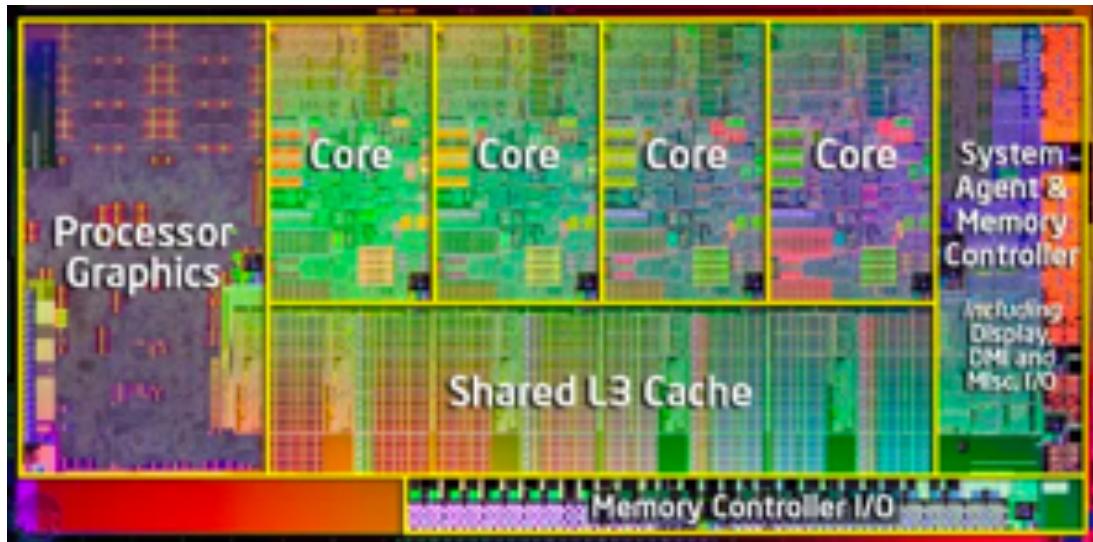
► \$ cat /proc/cpuinfo

```
administrator@ubuntuvm-1604 ~> cat /proc/cpuinfo
processor       : 0
vendor_id      : GenuineIntel
cpu family     : 6
model          : 79
model name    : Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz
stepping        : 1
microcode      : 0xb000036
cpu MHz        : 2199.998
cache size     : 51200 KB
physical id    : 0
siblings        : 1
core id         : 0
cpu cores      : 1
apicid          : 0
initial apicid : 0
fpu             : yes
fpu_exception   : yes
cpuid level    : 20
wp              : yes
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep
x pdpe1gb rdtscp lm constant_tsc arch_perfmon nopl xtopology t
e4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave a
vpcid_single pti ssbd ibrs ibpb stibp fsgsbase tsc_adjust bmi1
h_lid arch_capabilities
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_stor
bogomips       : 4399.99
clflush size   : 64
cache_alignment : 64
address sizes  : 42 bits physical, 48 bits virtual
power management:
```

processor : 0	processor : 1
vendor_id : GenuineIntel	vendor_id : GenuineIntel
cpu family : 6	cpu family : 6
model : 79	model : 79
model name : Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz	model name : Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz
stepping : 1	stepping : 1
microcode : 0xb000036	microcode : 0xb000036
cpu MHz : 2199.998	cpu MHz : 2199.998
cache size : 51200 KB	cache size : 51200 KB
physical id : 0	physical id : 2
siblings : 1	siblings : 1
core id : 0	core id : 0
cpu cores : 1	cpu cores : 1
apicid : 0	apicid : 2
initial apicid : 0	initial apicid : 2
fpu : yes	fpu : yes
fpu_exception : yes	fpu_exception : yes
cpuid level : 20	cpuid level : 20
wp : yes	wp : yes
flags : fpu vme de pse tsc msr pae mce cx8 apic sep x pdpe1gb rdtscp lm constant_tsc arch_perfmon nopl xtopology t e4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave a vpcid_single pti ssbd ibrs ibpb stibp fsgsbase tsc_adjust bmi1 h_lid arch_capabilities	flags : fpu vme de pse tsc msr pae mce cx8 apic sep x pdpe1gb rdtscp lm constant_tsc arch_perfmon nopl xtopology t e4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave a vpcid_single pti ssbd ibrs ibpb stibp fsgsbase tsc_adjust bmi1 h_lid arch_capabilities
bugs : cpu_meltdown spectre_v1 spectre_v2 spec_stor	bugs : cpu_meltdown spectre_v1 spectre_v2 spec_stor
bogomips : 4399.99	bogomips : 4399.99
clflush size : 64	clflush size : 64
cache_alignment : 64	cache_alignment: 64
address sizes : 42 bits physical, 48 bits virtual	address sizes : 42 bits physical, 48 bits virtual
power management:	power management:

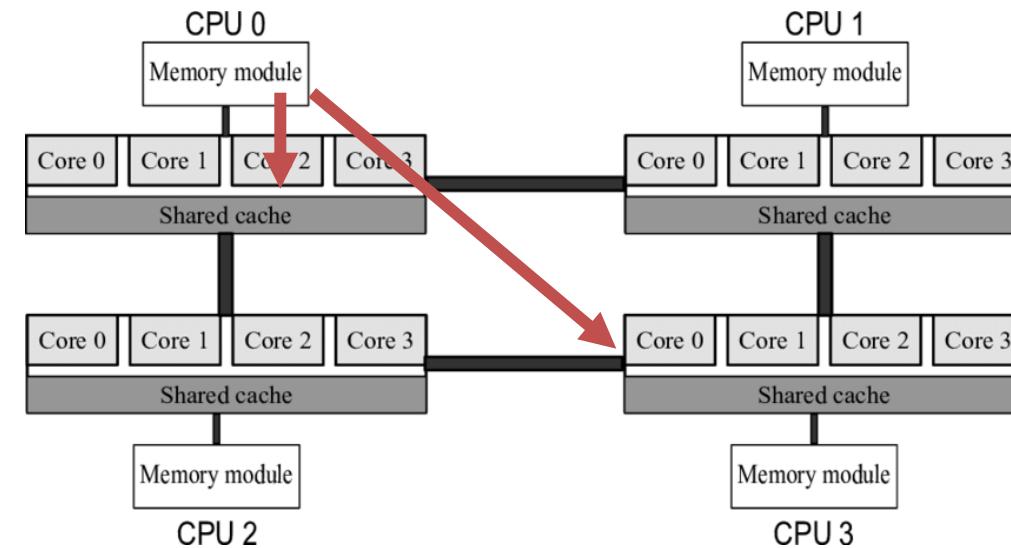
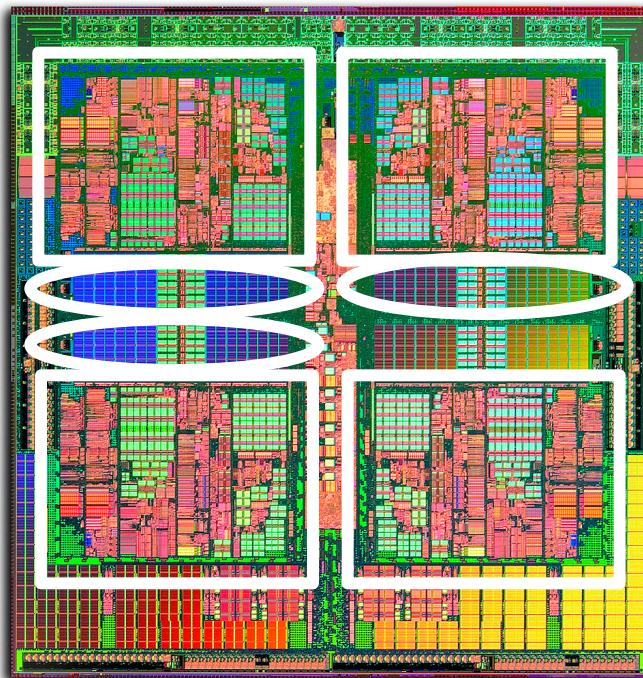
Multi-Core Processors (SMP)

- ▶ Multiple CPUs on a single chip



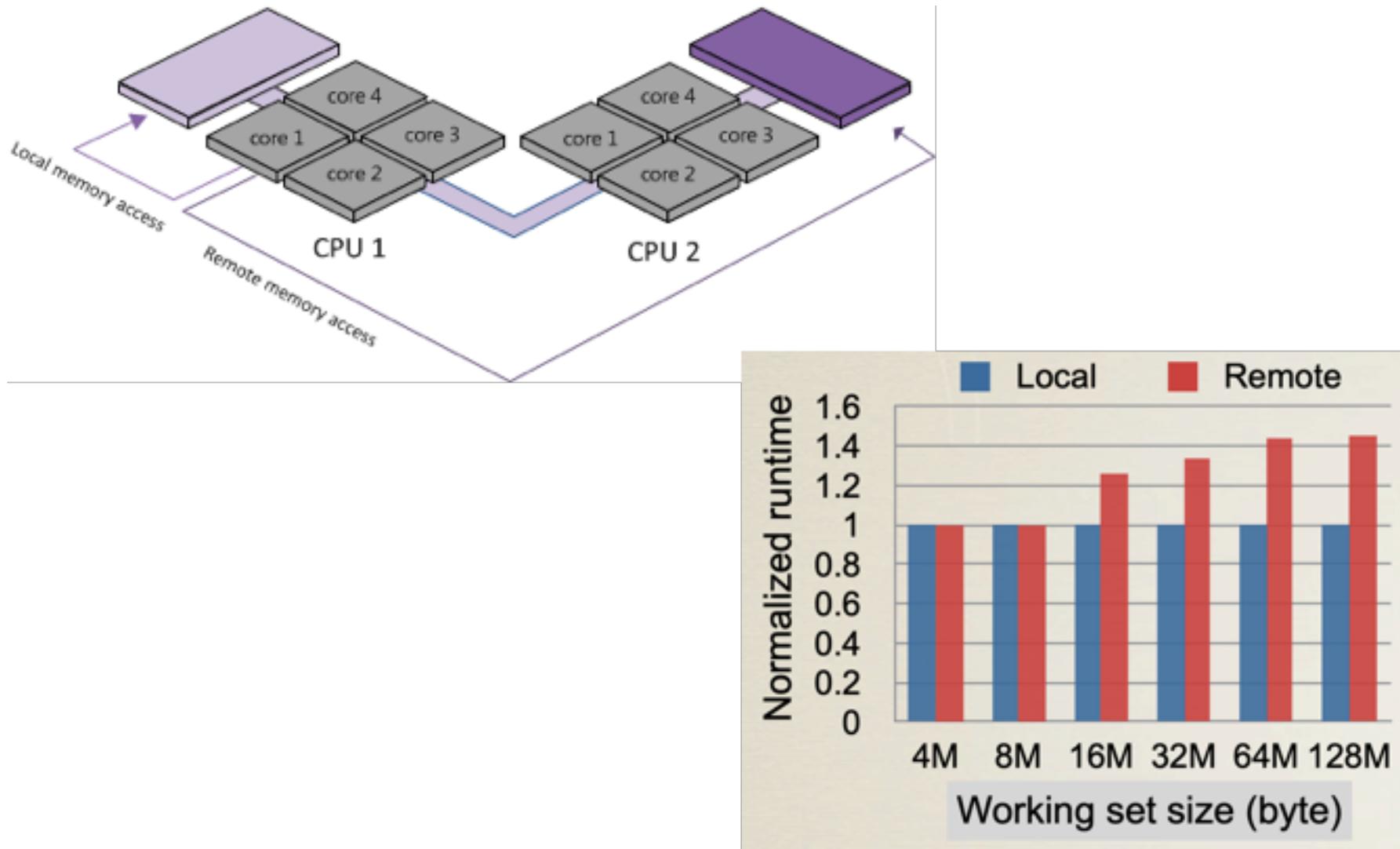
Multi-Core Processors (NUMA)

- ▶ Multiple CPUs on a single chip



Non-uniform memory access (**NUMA**)

Multi-Core Processors (NUMA)



Check CPU topology

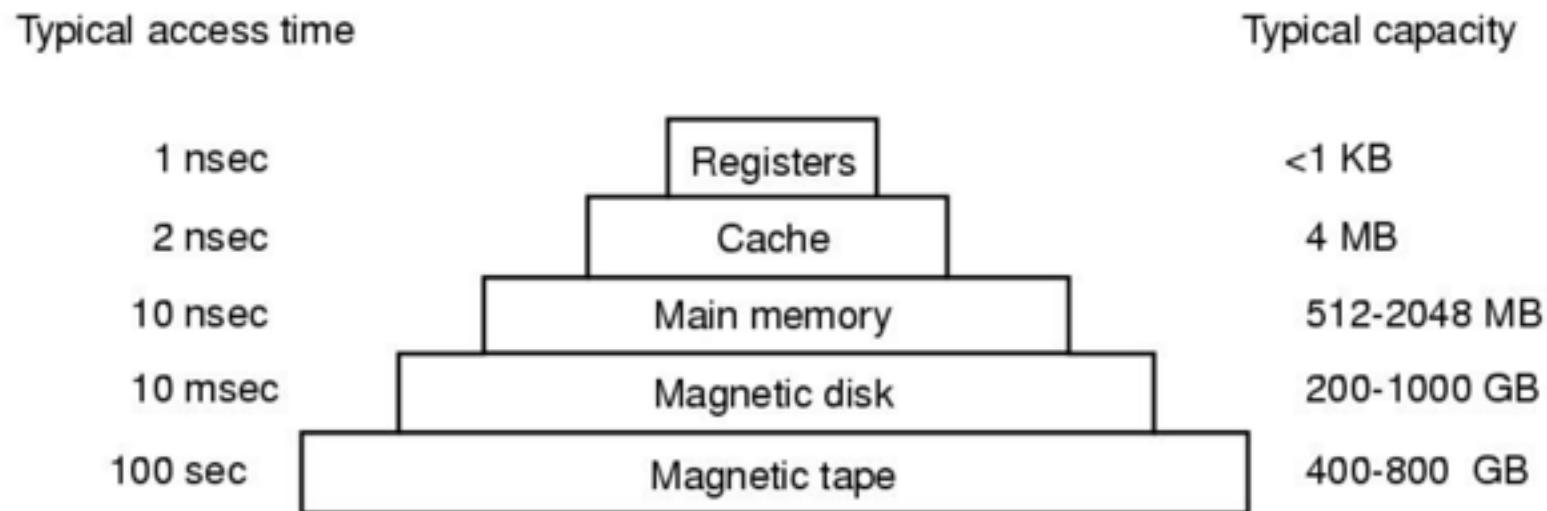
► \$ likwid-topology -g

► NUMA example:
<https://github.com/RZERHEPC/likwid/wiki/TutorialNUMA>

```
ksuo@ksuo-VirtualBox ~/likwid-5.0.0> likwid-topology -g
-----
CPU name: Intel(R) Core(TM) i9-9880H CPU @ 2.30GHz
CPU type: Intel CoffeeLake processor
CPU stepping: 13
*****
Hardware Thread Topology
*****
Sockets: 1
Cores per socket: 4
Threads per core: 1
-----
HWThread Thread Core Socket Available
0 0 0 0 *
1 0 1 0 *
2 0 2 0 *
3 0 3 0 *
-----
Socket 0: ( 0 1 2 3 )
-----
*****
Graphical Topology
*****
Socket 0:
+-----+
| +---+ +---+ +---+ +---+ |
| | 0 | | 1 | | 2 | | 3 | |
| +---+ +---+ +---+ +---+ |
| +---+ +---+ +---+ +---+ |
| | 32 kB | | 32 kB | | 32 kB | | 32 kB | |
| +---+ +---+ +---+ +---+ |
| +---+ +---+ +---+ +---+ |
| | 256 kB | | 256 kB | | 256 kB | | 256 kB | |
| +---+ +---+ +---+ +---+ |
| +---+ +---+ +---+ +---+ |
| | 16 MB | | 16 MB | | 16 MB | | 16 MB | |
| +---+ +---+ +---+ +---+ |
+-----+
```

Memory

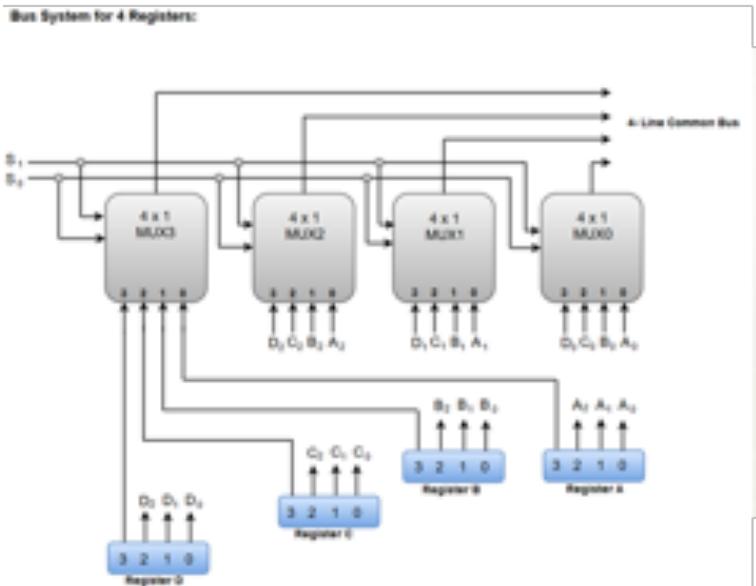
► A typical memory hierarchy



Minimize the access time vs. Cost

Memory

► A typical memory hierarchy



Memory information

► free

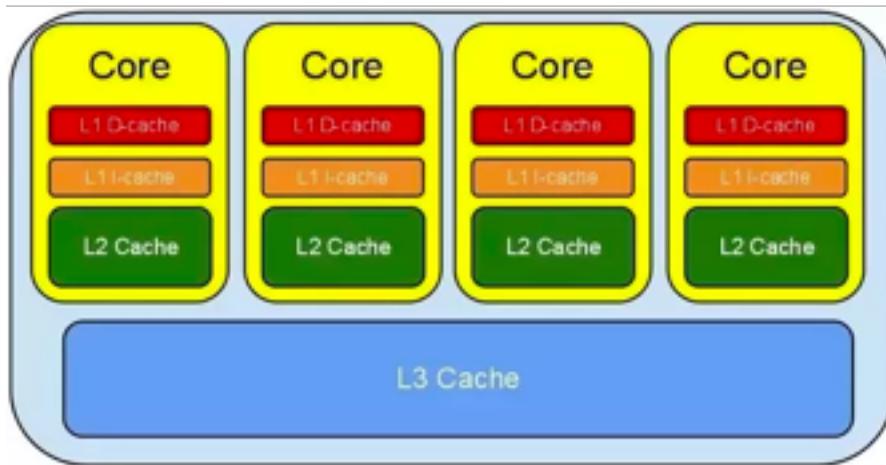
```
administrator@ubuntuvm-1604 ~> free
              total        used        free      shared  buff/cache   available
Mem:       8168756     1348160     3031888          97424    3788708     6383036
Swap:      998396          0     998396
```

► The free command displays:

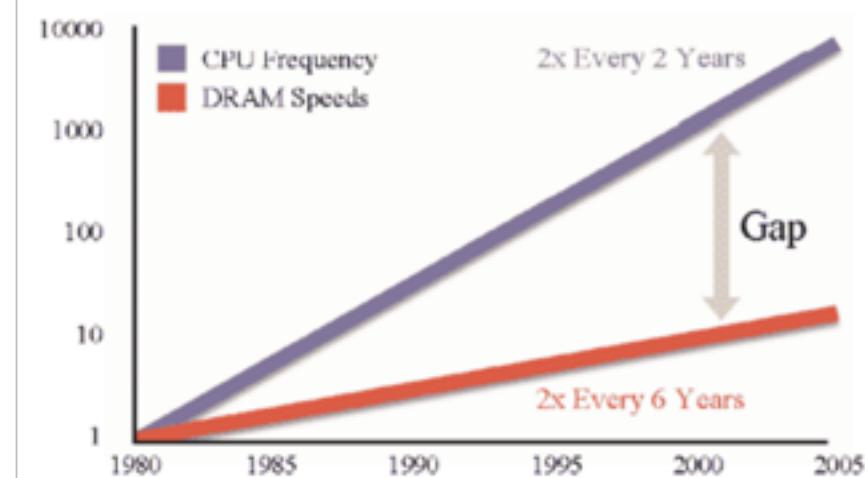
- Total amount of free and used physical memory
- Total amount of swap memory in the system
- Buffers and caches used by the kernel

Cache

► Why Cache is important?



- A larger size than registers
- A much faster speed than memory
- Tradeoff between performance and cost



MacBook Pro	
Hardware Overview:	
Model Name:	MacBook Pro
Model Identifier:	MacBookPro15,1
Processor Name:	Intel Core i9
Processor Speed:	2.3 GHz
Number of Processors:	1
Total Number of Cores:	8
L2 Cache (per Core):	256 KB
L3 Cache:	16 MB
Hyper-Threading Technology:	Enabled
Memory:	16 GB
Boot ROM Version:	220.270.99.0.0 (iBridge: 16.16.6568.0.0,0)
Serial Number (system):	C02YR4JHLVCJ
Hardware UUID:	DCC2D30A-9630-57B5-89A2-5F2BB85254DC1

Check Cache info

- ▶ \$ likwid-topology –g
- ▶ \$ lscpu | grep cache

```
*****
Cache Topology
*****
Level:          1
Size:           32 kB
Cache groups:  ( 0 ) ( 1 ) ( 2 ) ( 3 )

Level:          2
Size:           256 kB
Cache groups:  ( 0 ) ( 1 ) ( 2 ) ( 3 )

Level:          3
Size:           16 MB
Cache groups:  ( 0 ) ( 1 ) ( 2 ) ( 3 )

*****
NUMA Topology
*****
NUMA domains:  1

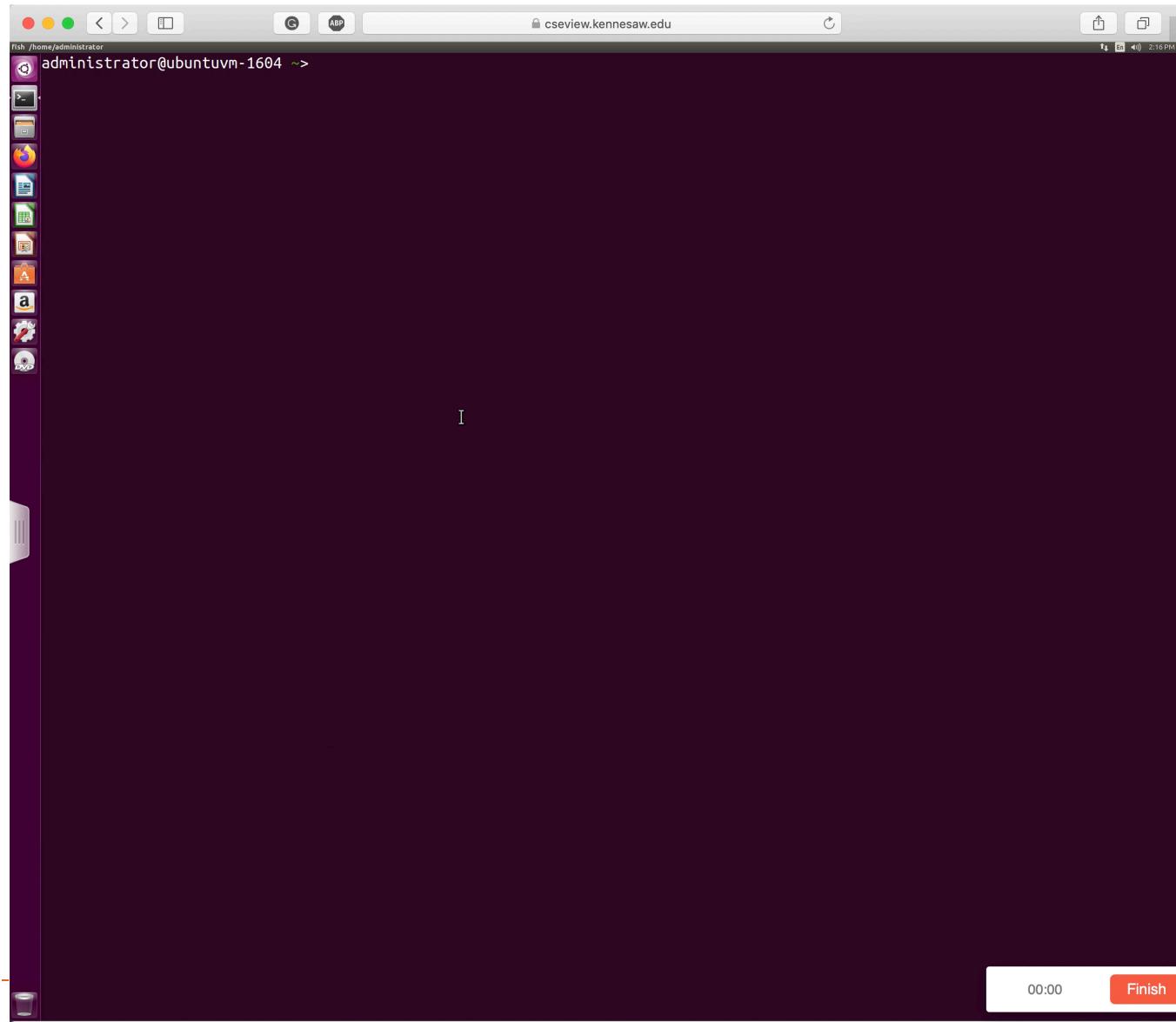
Domain:        0
Processors:    ( 0 1 2 3 )
Distances:     10
Free memory:   1967.79 MB
Total memory:  3942.19 MB
```

```
ksuo@ksuo-VirtualBox ~/likwid-5.0.0> lscpu | grep cache
L1d cache:      32K
L1i cache:      32K
L2 cache:       256K
L3 cache:      16384K
```

Memory information

- ▶ **Istopo:** show the CPU cache and logical CPU layout
- ▶ **Intall:** \$ sudo apt-get install hwloc
- ▶ **Run:** \$ Istopo

Memory information



Test: what is the speed of your memory?

► STREAM Benchmark:

- it can measure the performance of the memory system, including bandwidth and latency.
- <https://www.cs.virginia.edu/stream/>

Function	Rate (MB/s)	Avg time	Min time	Max time
Copy:	16302.4084	0.0026	0.0020	0.0031
Scale:	13411.0440	0.0029	0.0024	0.0040
Add:	15529.6662	0.0041	0.0031	0.0050
Triad:	12889.0264	0.0042	0.0037	0.0047

Test: what is the speed of your memory?

Copy

```
void tuned_STREAM_Copy()
{
    int j;
    for (j=0; j<N; j++)
        c[j] = a[j];
}
```

Read from one memory cell

Write to another memory cell

Test: what is the speed of your memory?

Scale

```
void tuned_STREAM_Scale(double scalar)
{
    int j;
    for (j=0; j<N; j++)
        b[j] = scalar*c[j];
}
```

Read from one memory cell

Make a multiply operation

Write to another memory cell

Test: what is the speed of your memory?

Sum

```
void tuned_STREAM_Add()
{
    int j;
    for (j=0; j<N; j++)
        c[j] = a[j]+b[j];
}
```

Read from two memory cell

Make an add operation

Write to another memory cell

Test: what is the speed of your memory?

Triad

```
void tuned_STREAM_Triad(double scalar)
{
    int j;
    for (j=0; j<N; j++)
        a[j] = b[j]+scalar*c[j];
}
```

Read from two memory cell

Make an add operation and a multiply operation

Write to another memory cell

Test: what is the speed of your memory?

► 1, Download STREAM Benchmark

- \$ wget <https://www.nersc.gov/assets/Trinity--NERSC-8-RFP/Benchmarks/Jan9/stream.tar>

► 2, Compile & run

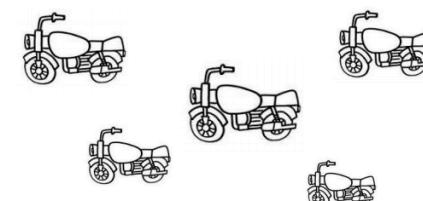
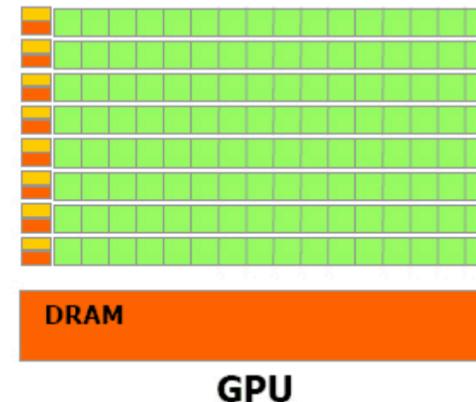
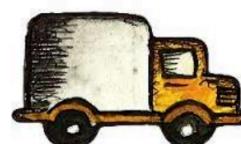
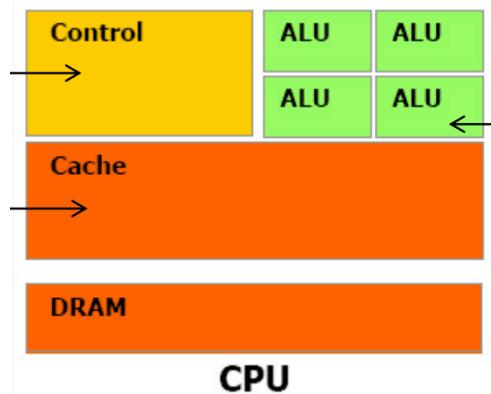
- \$ tar xvf stream.tar
\$ gcc stream.c -o stream
\$./stream

GPU

► Graphical Processing Unit

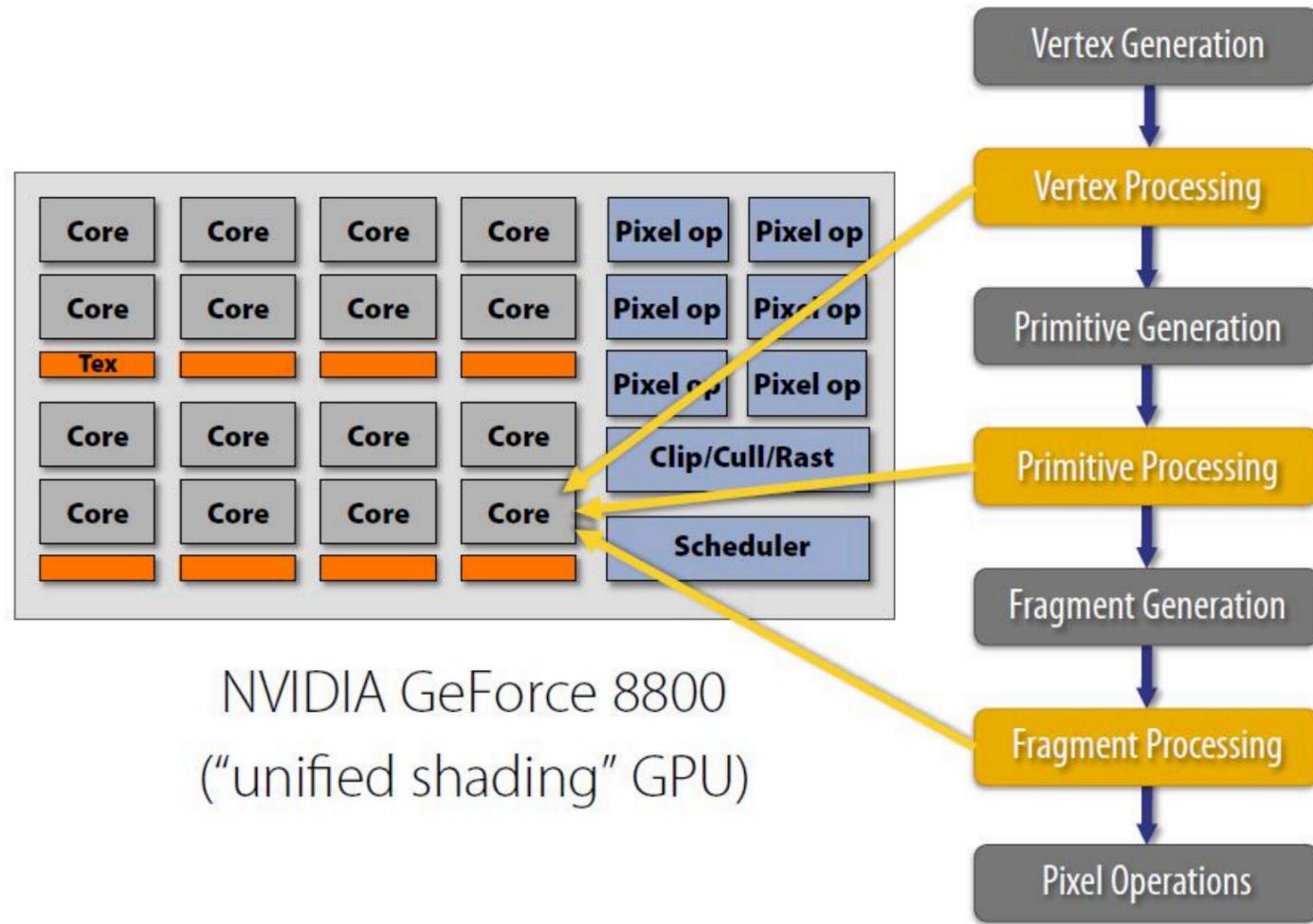
- The task is to composite and display images with millions of pixels on the screen
- Good at large numbers, simple, and inherently difficult to do complex calculations (logic, process control)

► GPU has many cores



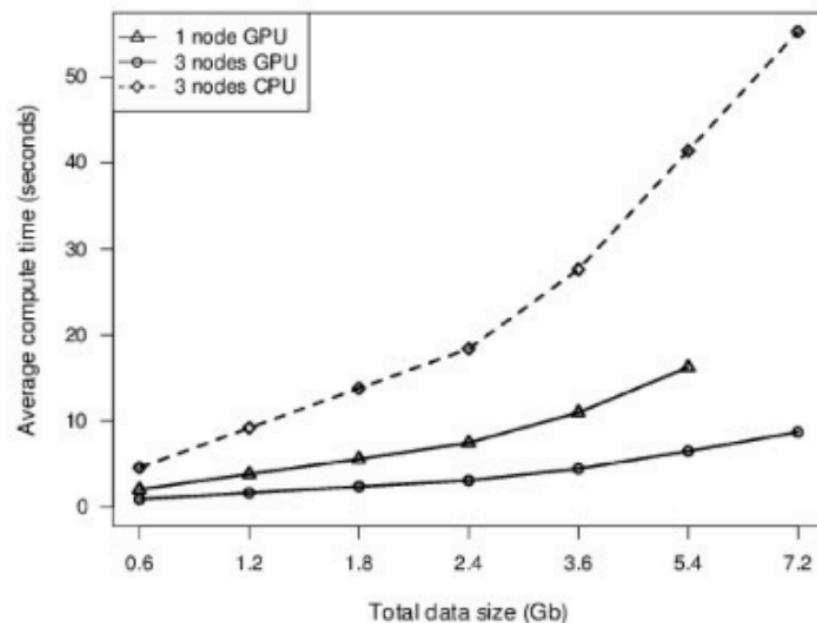
GPU

- ▶ Each pixel can be given to A CORE processing

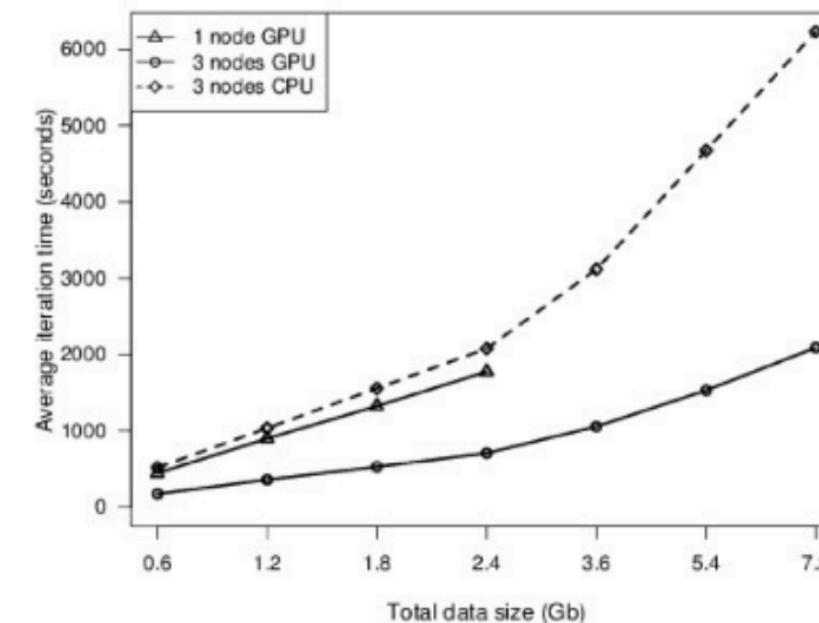


SPEEDUPS

Naive Bayes classifier average model compute time



k-Means average iteration time



Up to 12x speedup for a same task when nodes are equipped with GPU

GPU



Check GPU on your machine

► \$ lspci | grep -i vga

```
administrator@ubuntu1804vm ~> lspci | grep -i vga
00:0f.0 VGA compatible controller: VMware SVGA II Adapter
```

If you have Nvidia GPU, use

► \$ lspci | grep -i nvidia

or

► \$ nvidia-smi

NVIDIA-SMI 367.57			Driver Version: 367.57		
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util Compute M.
0	Tesla K80	Off	0000:05:00.0	Off	0
N/A	35C	P0	66W / 149W	8221MiB / 11439MiB	0% Default
1	Tesla K80	Off	0000:06:00.0	Off	0
N/A	52C	P0	81W / 149W	5822MiB / 11439MiB	0% Default
2	Tesla K80	Off	0000:09:00.0	Off	0
N/A	33C	P0	65W / 149W	5822MiB / 11439MiB	0% Default
3	Tesla K80	Off	0000:0A:00.0	Off	0
N/A	35C	P8	31W / 149W	2MiB / 11439MiB	0% Default
Processes:			GPU Memory Usage		
GPU	PID	Type	Process name		

TPU: Tensor Processing Unit

- ▶ **Proposal:** Design a custom ASIC (Application-specific integrated circuit) for the inference phase of NN (training still happens using GPUs)
- ▶ **Principles:**
 - improve cost-performance by 10X compared to GPUs
 - simple design for response time guarantees(single-thread, no prefetching, no OOO etc.)
- ▶ **Characteristics:**
 - More like a co-processor to reduce time-to-market delays
 - Host sends instructions to TPU
- ▶ **Connected through PCIe I/O bus**



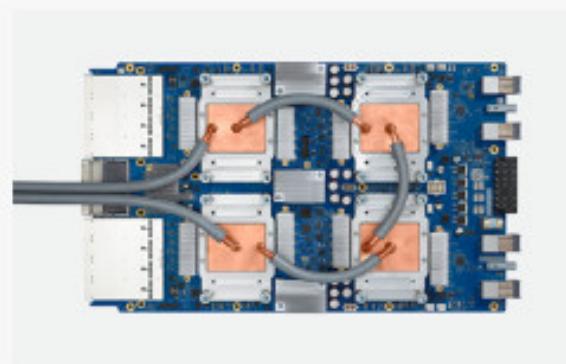
TPU: Tensor Processing Unit



Cloud TPU v2

180 teraflops

64 GB High Bandwidth Memory
(HBM)



Cloud TPU v3 Alpha

420 teraflops

128 GB HBM



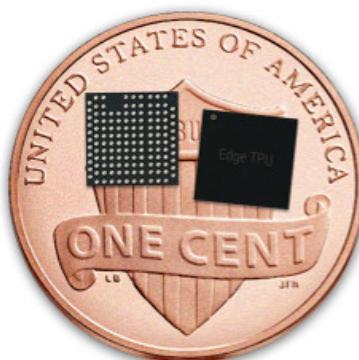
Cloud TPU v2 Pod Alpha

11.5 petaflops

4 TB HBM

2-D toroidal mesh network

Edge TPU



TPU: Tensor Processing Unit

Edge TPU



An interesting video introducing hardware

Computer Parts!

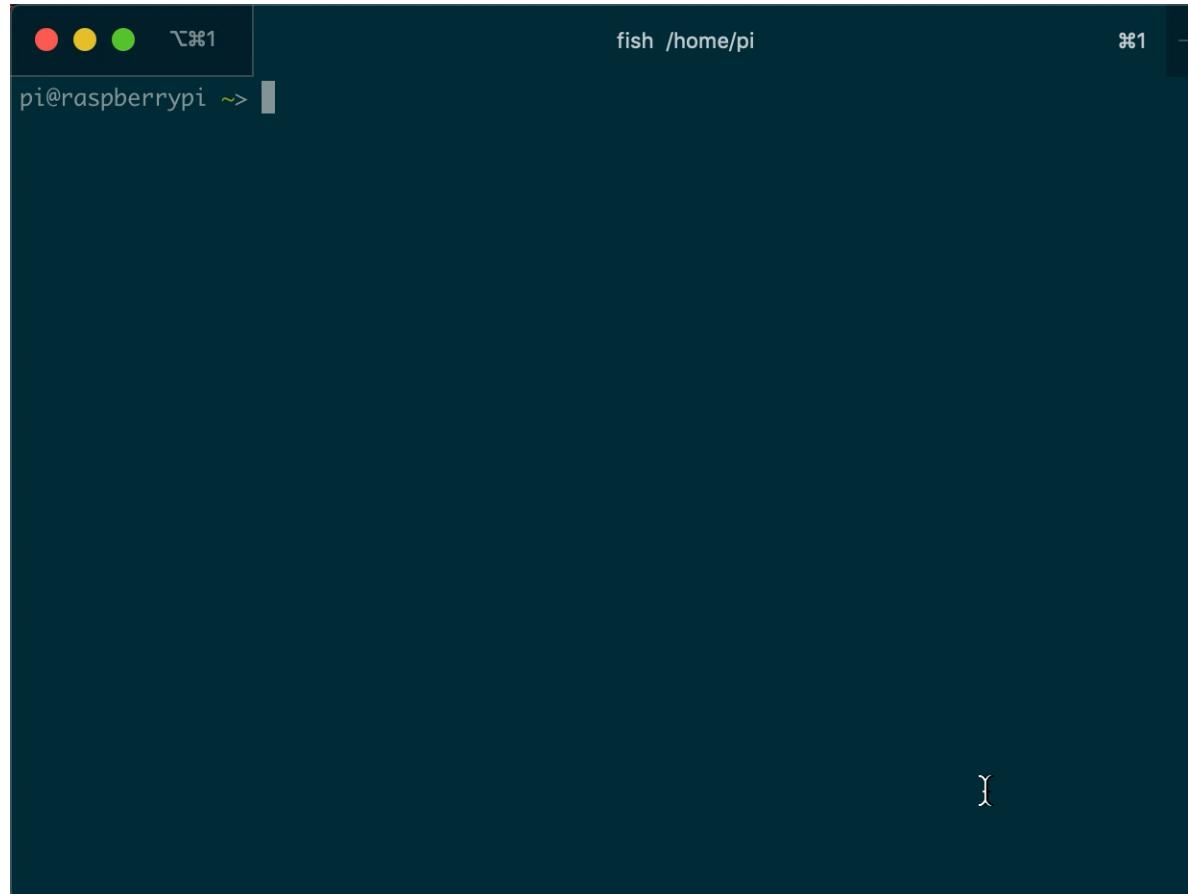


Test: Your VM hardware specs?

- ▶ CPU? → How many cores? Frequency? Topology?
- ▶ Cache? → Size?
- ▶ Memory? → Size? Speed?
- ▶ GPU? → Specs? Speed?
- ▶ TPU? → Specs? Speed?

How to get my machine spec?

- ▶ Inxi: <https://www.tecmint.com/inxi-command-to-find-linux-system-information/>



```
pi@raspberrypi ~> inxi -r
```

How to get my machine spec? One command for All!

