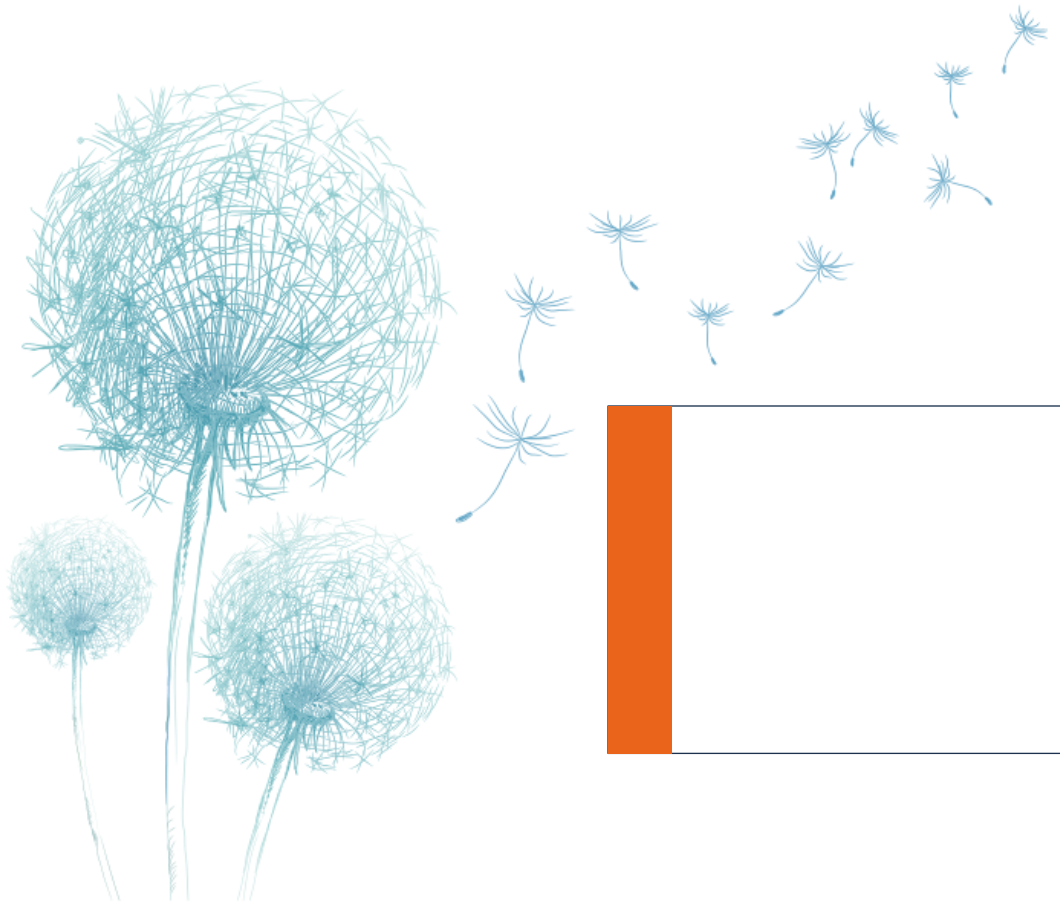
Neural networks and deep learning

Machine learning overview

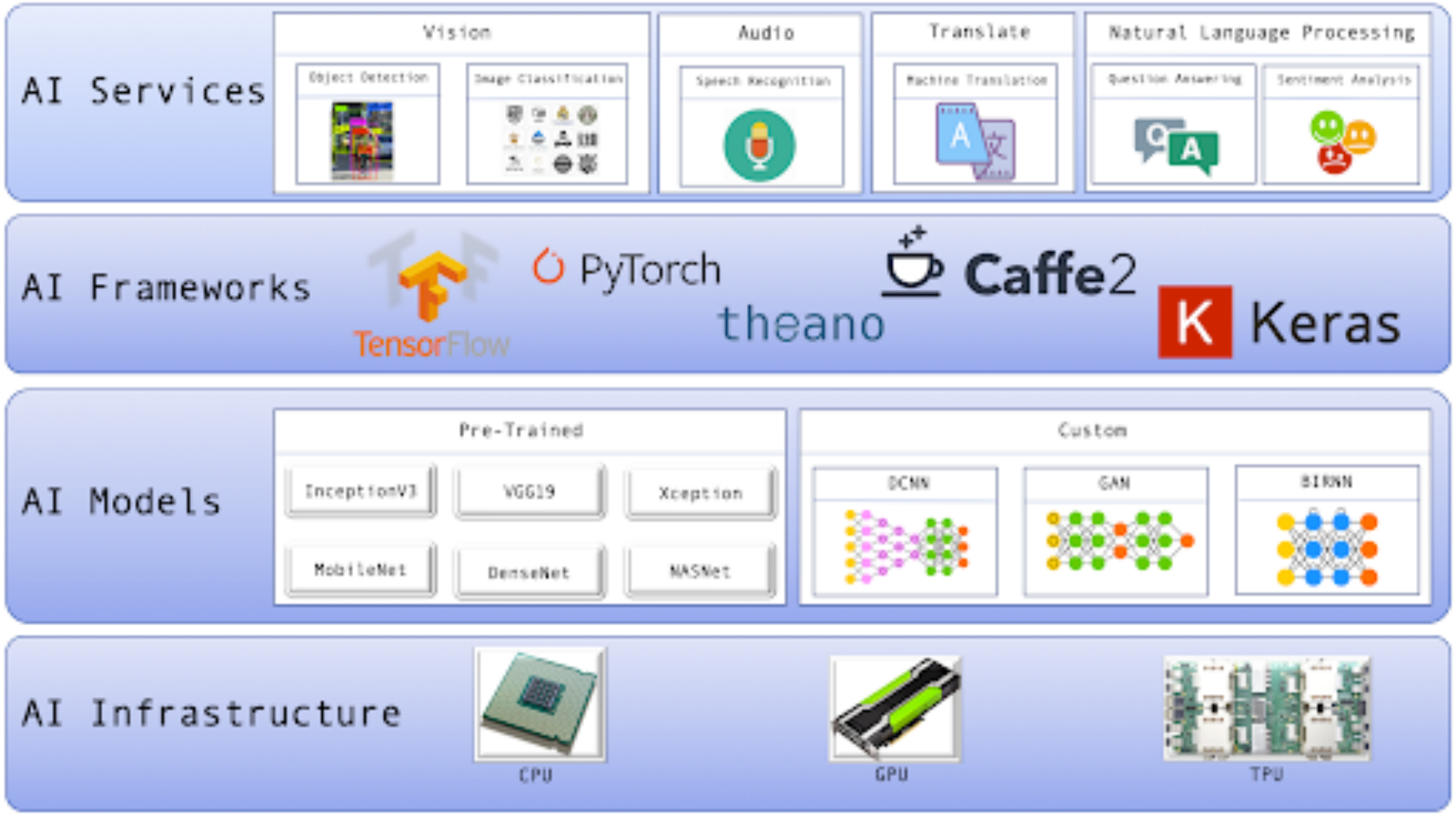
Kun Suo
Computer Science, Kennesaw State University
https://kevinsuo.github.io/



Hardware

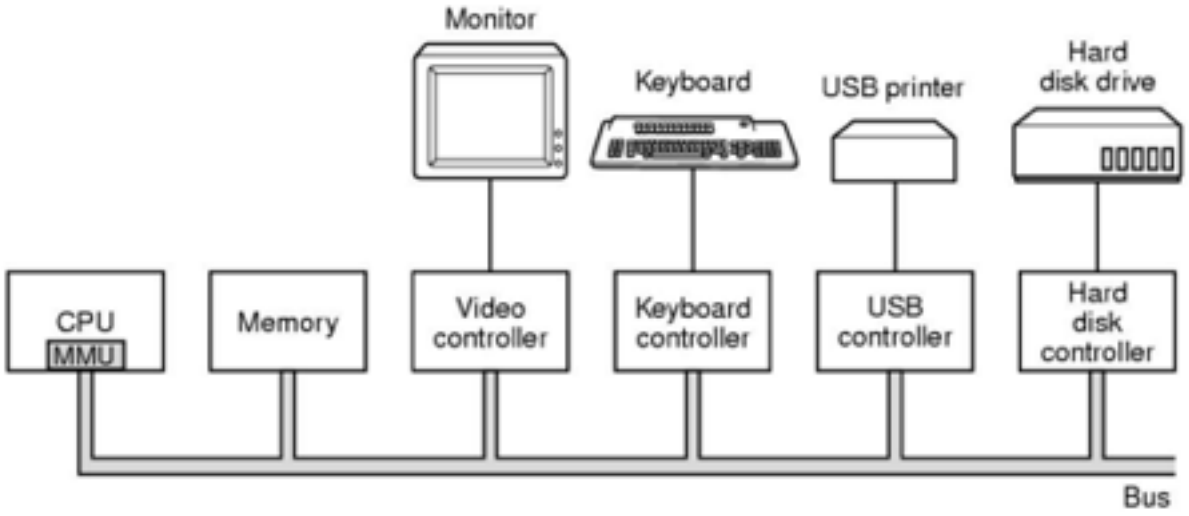
Computer Hardware Review

Artificial intelligence stack



Computer Hardware Review

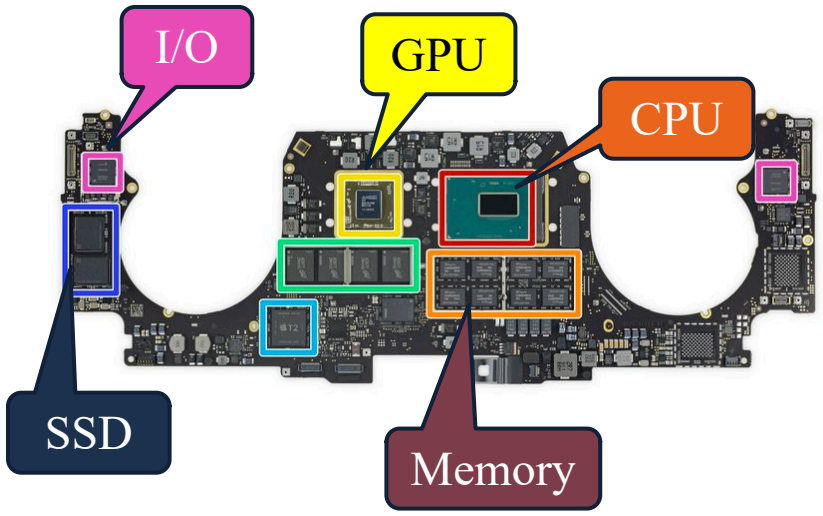
Basic components of a simple personal computer



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

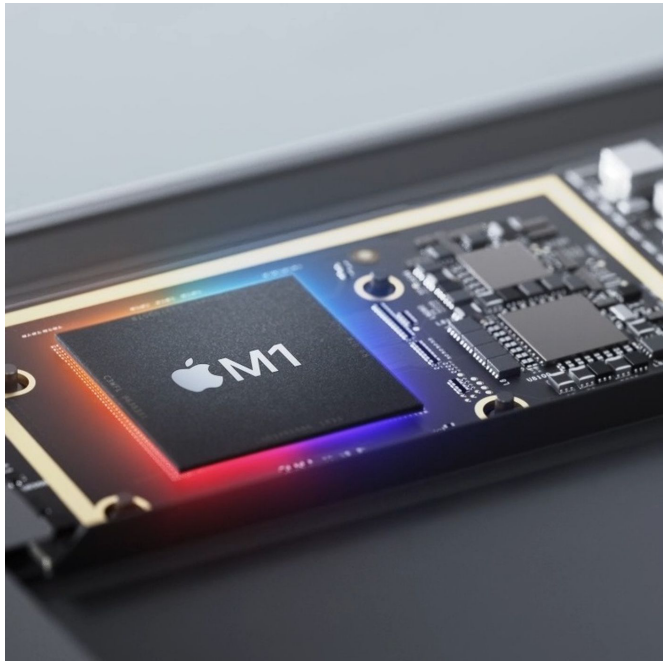
Computer Hardware Review

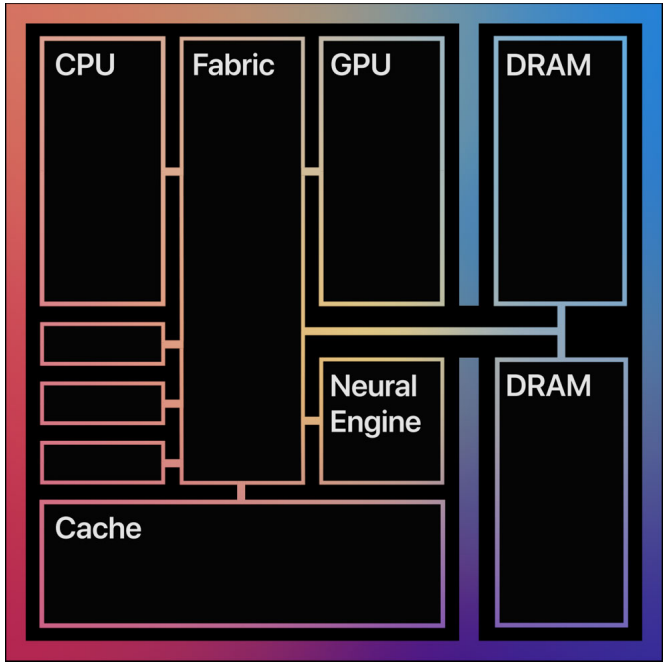
Computer Hardware Review

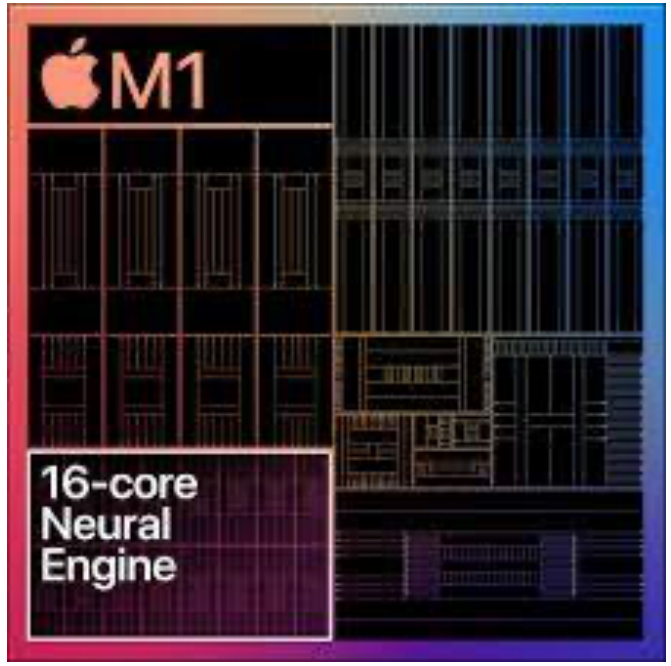


Computer Hardware Review

- Apple M1 chip

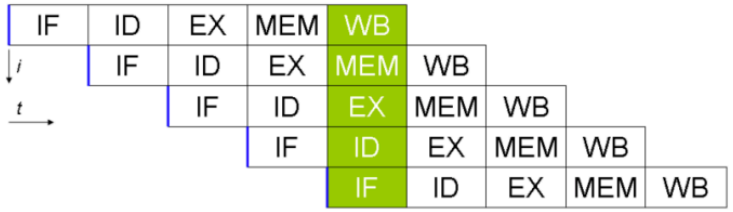


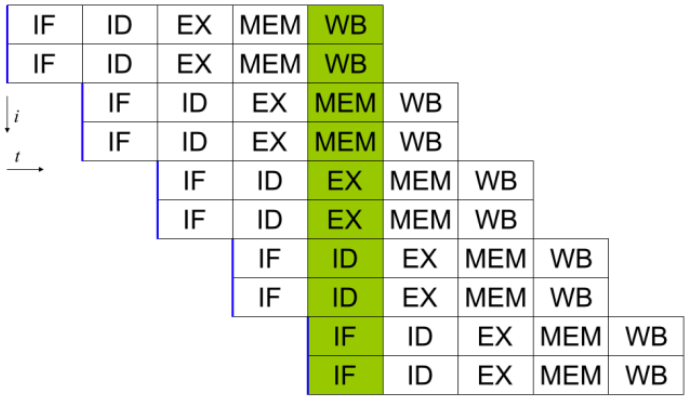


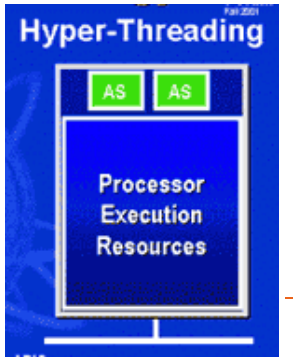


Central Processing Unit (CPU)

- 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

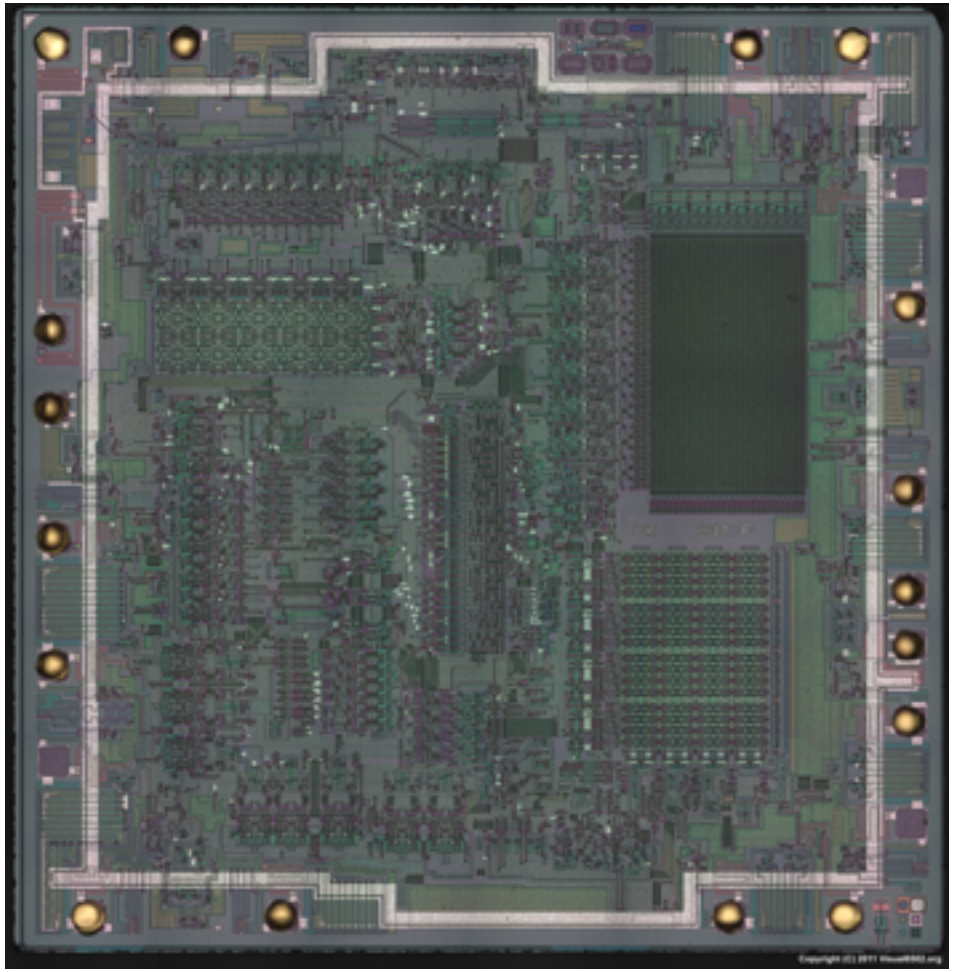






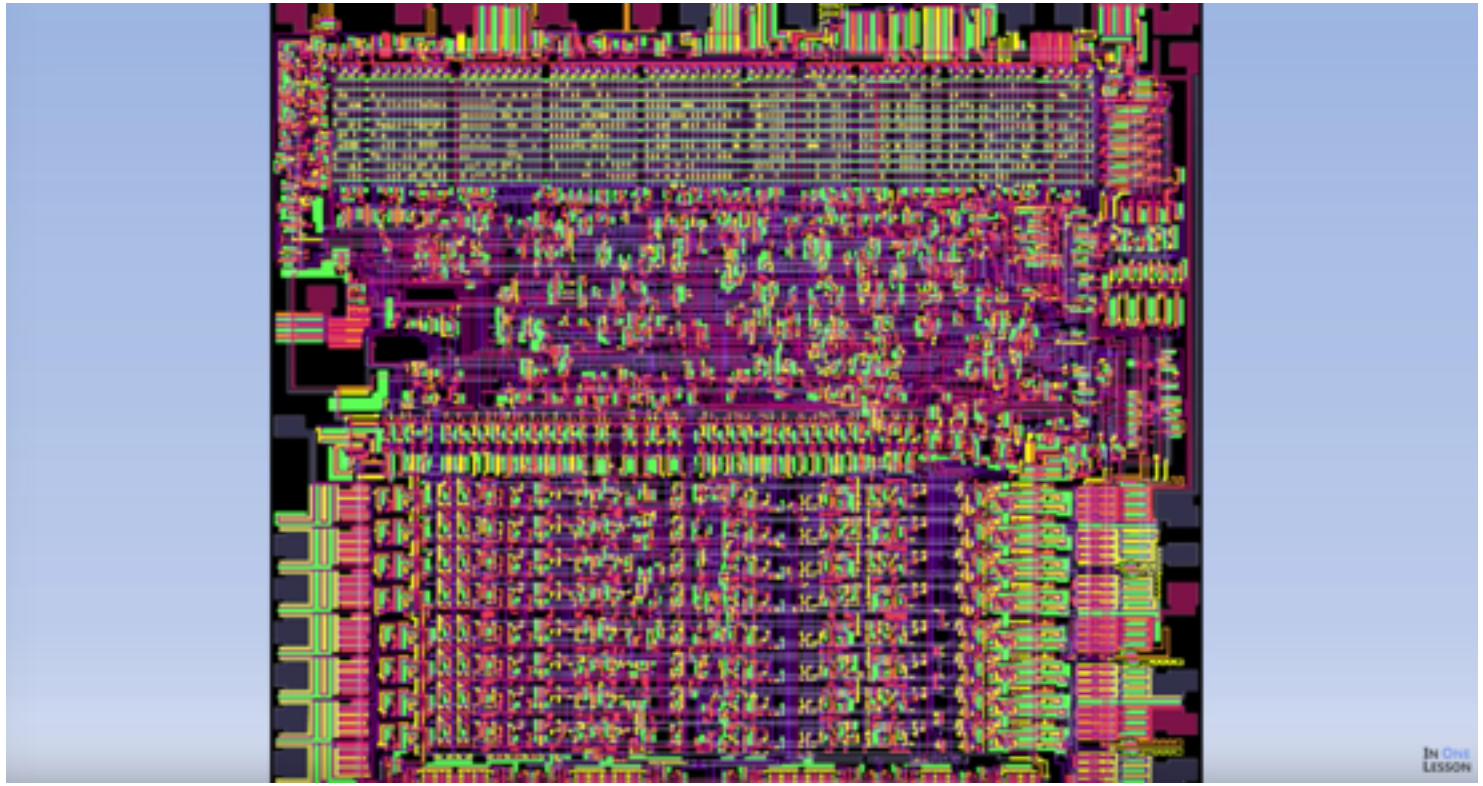
What's inside of CPU?

- http://www.visual6502.org/

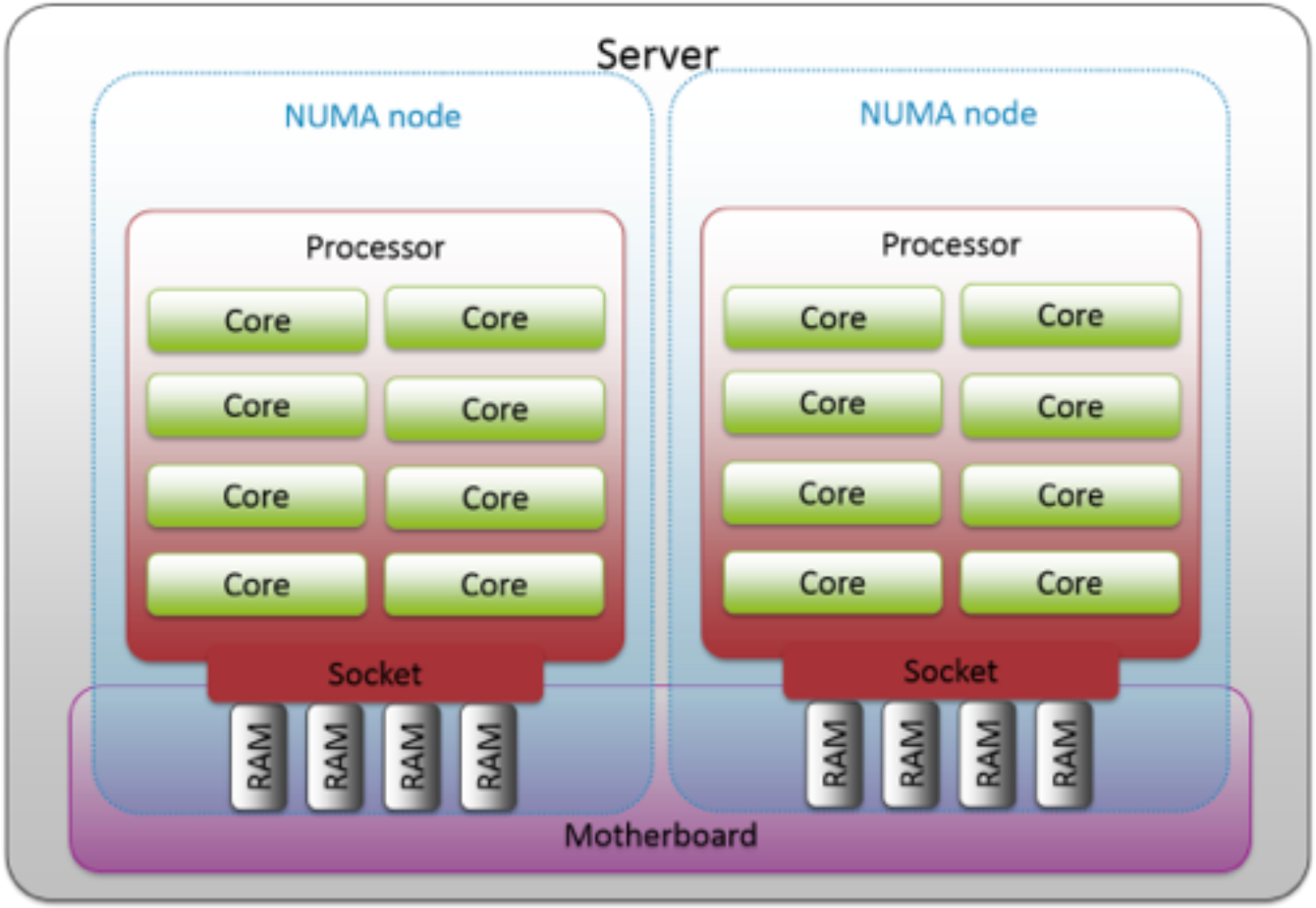


How CPU works?

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



NUMA node vs Socket vs Core relationship



CPU information

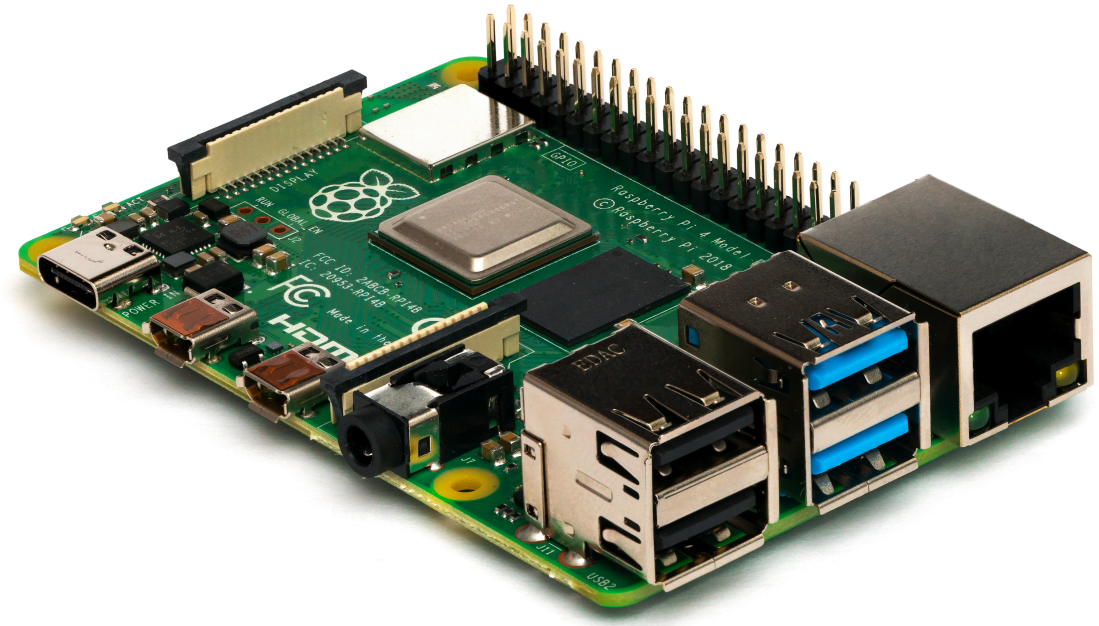
- lscpu

```
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
 flush_l1d arch_capabilities
```

CPU information

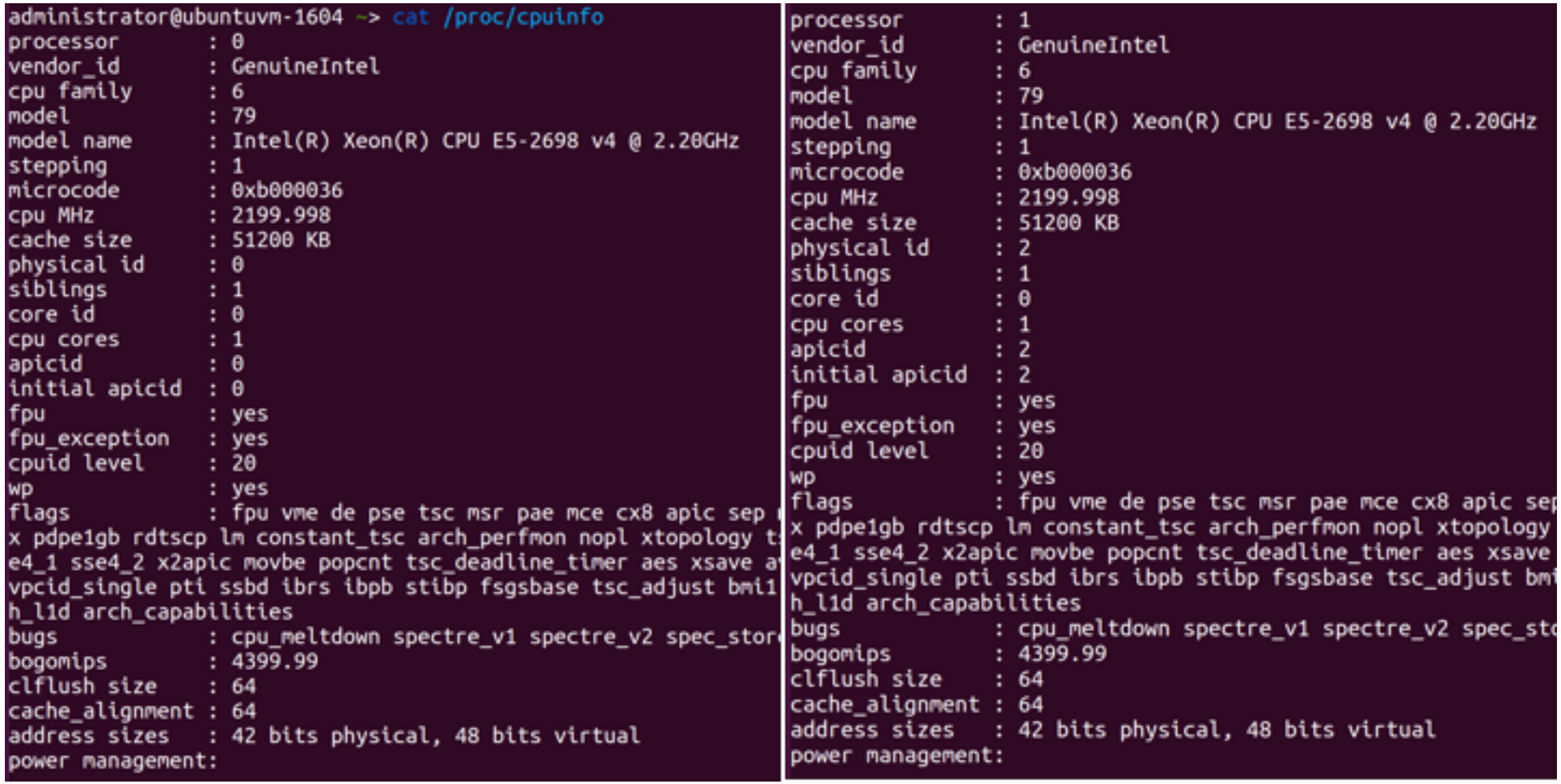
- lscpu

```
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

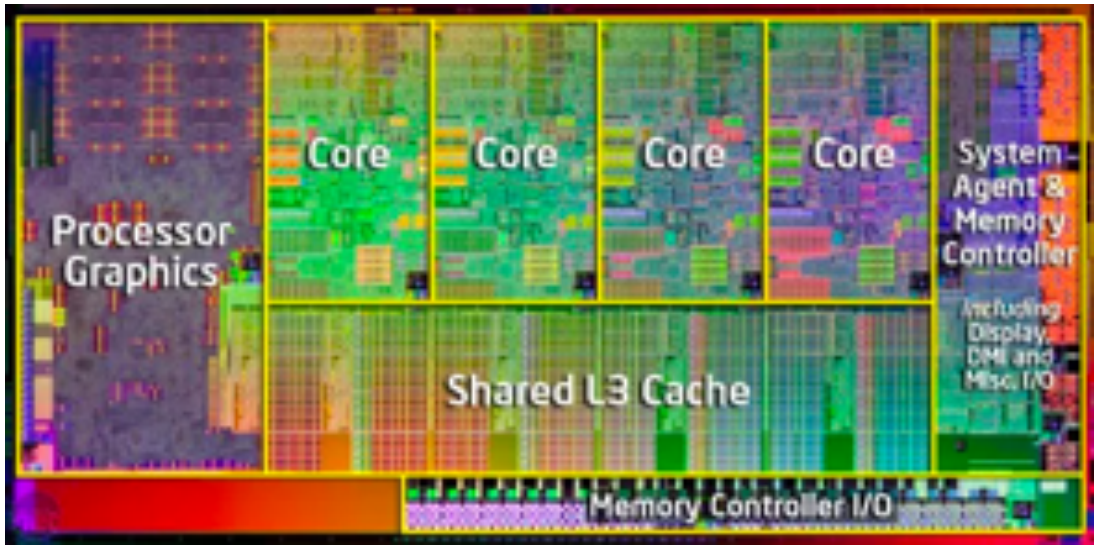


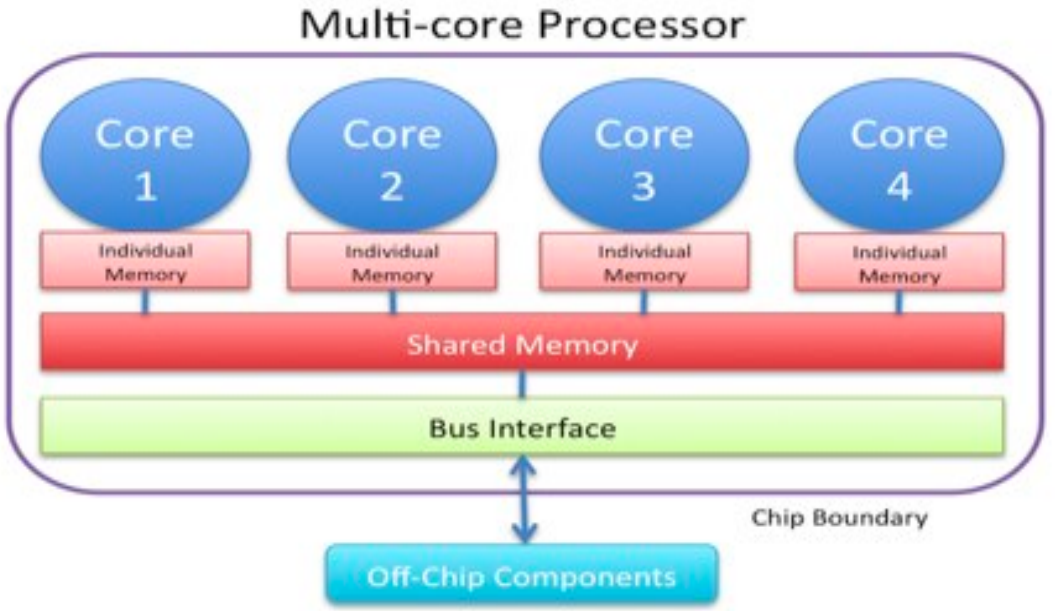
```
administrator@ubuntuvm-1604 ~> cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 79
model name      : Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz
stepping        : 1
microcode       : 0xb000036
cpu MHz         : 2199.998
cache size      : 51200 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_l1d 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       : 1
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     : 2
siblings        : 1
core id         : 0
cpu cores       : 1
apicid          : 2
initial apicid  : 2
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
e4_1 sse4_2 x2apic movbe popcnt tsc_deadline_timer aes xsave
vpcid_single pti ssbd ibrs ibpb stibp fsgsbase tsc_adjust bmi
h_l1d arch_capabilities
bugs            : cpu_meltdown spectre_v1 spectre_v2 spec_st
bogomips        : 4399.99
clflush size    : 64
cache_alignment : 64
address sizes   : 42 bits physical, 48 bits virtual
power management:
```

Multi-Core Processors (SMP)

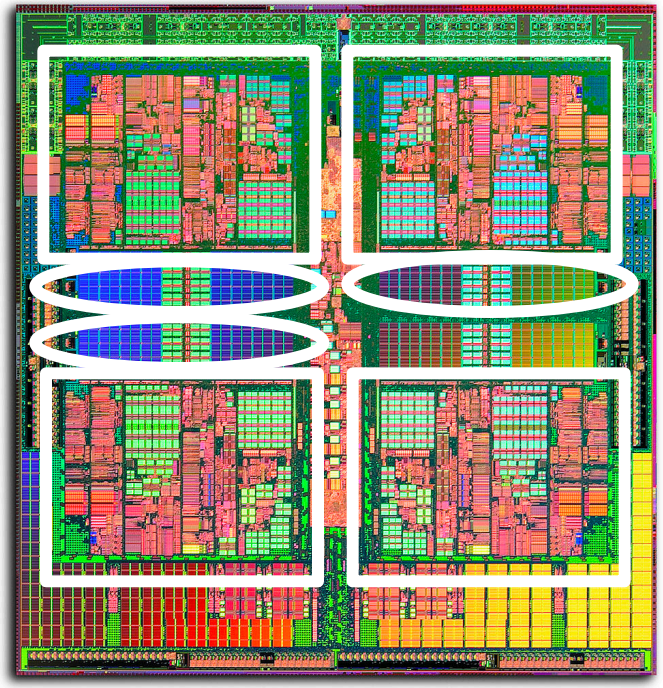
- Multiple CPUs on a single chip

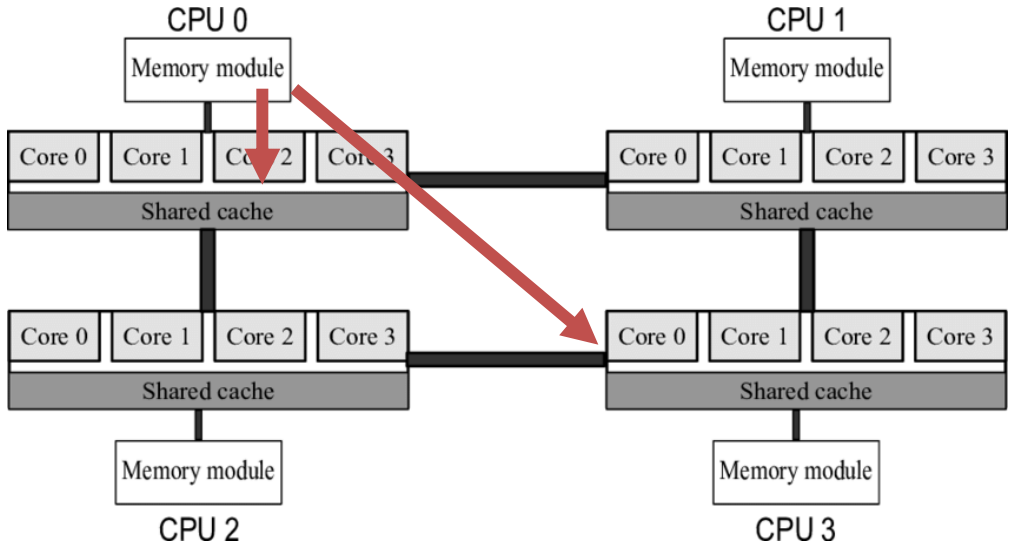




Multi-Core Processors (NUMA)

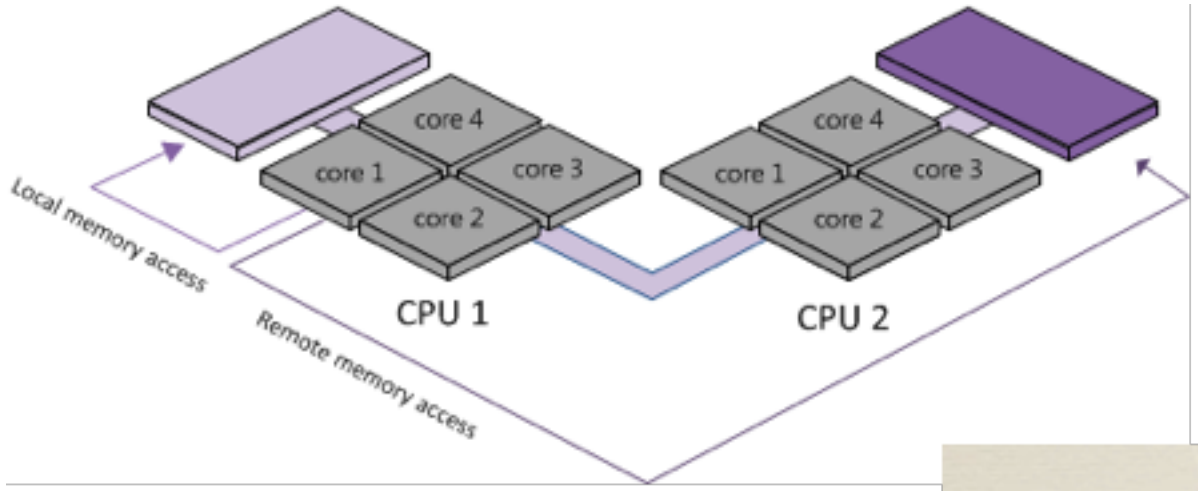
- Multiple CPUs on a single chip

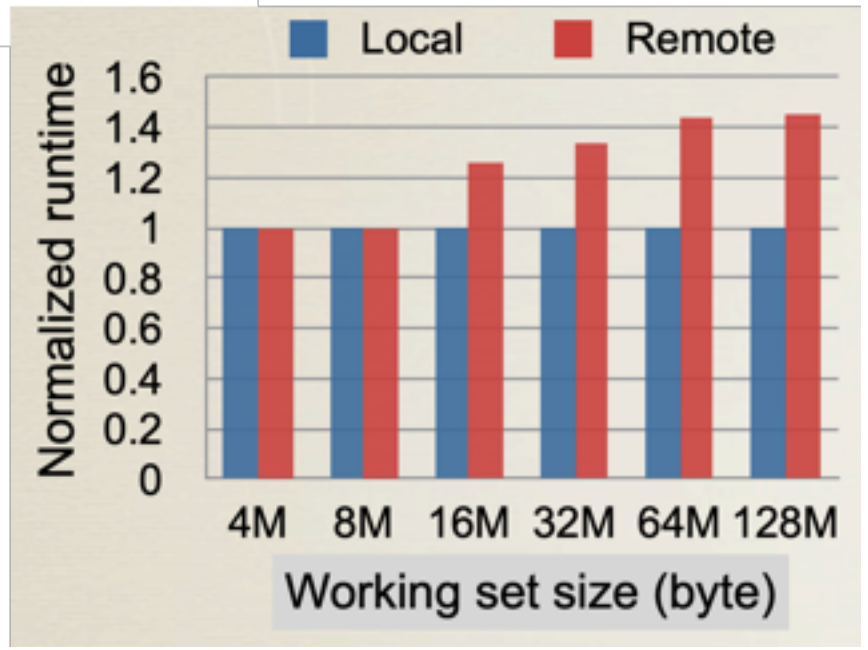




Non-uniform memory access (**NUMA**)

Multi-Core Processors (NUMA)





Check CPU topology

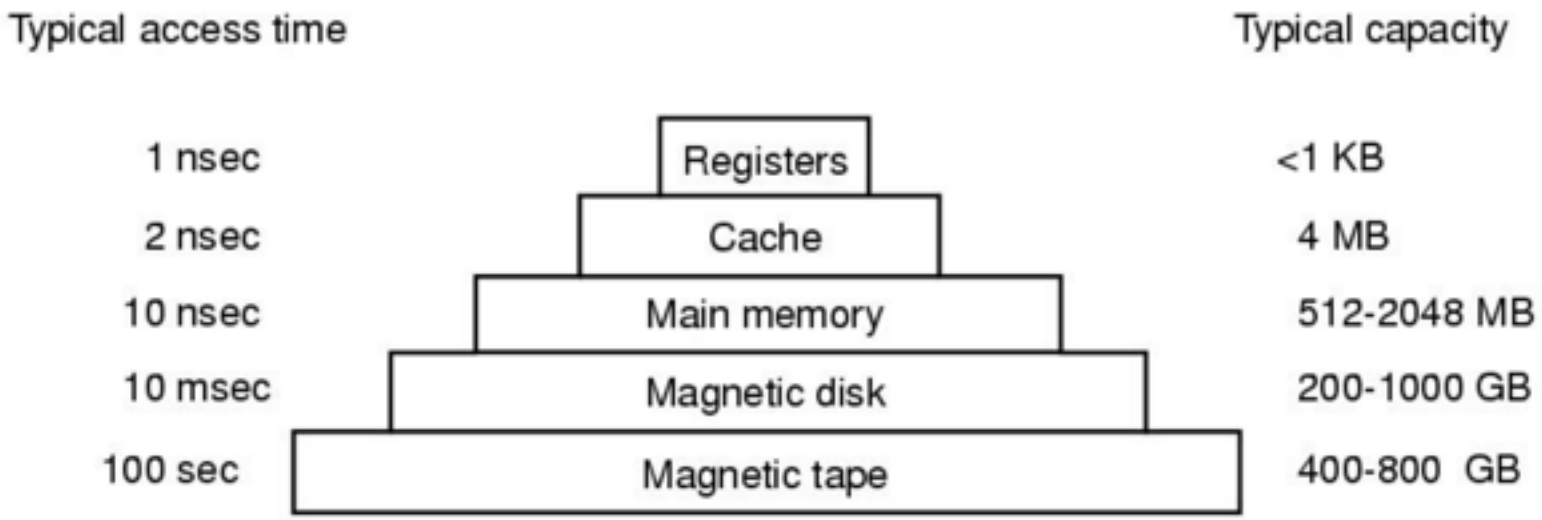
- $ likwid-topology –g

- NUMA example: https://github.com/RRZE-HPC/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

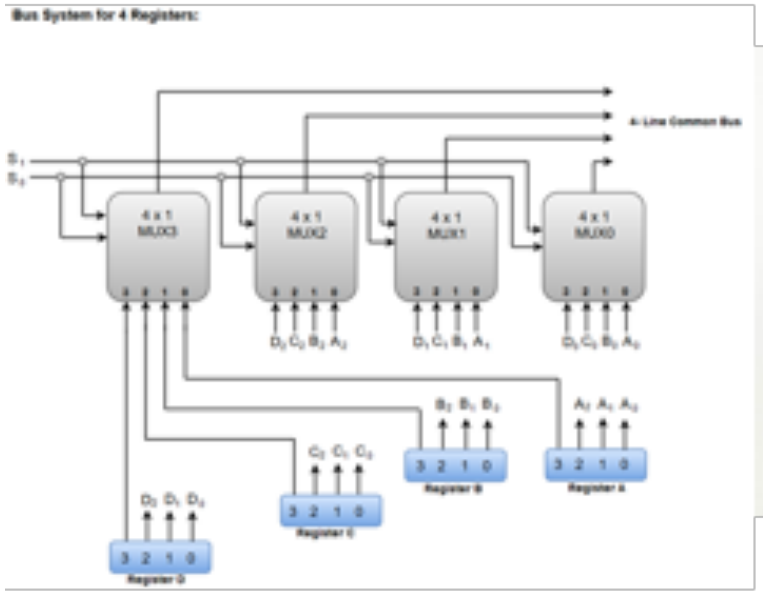
- A typical memory hierarchy

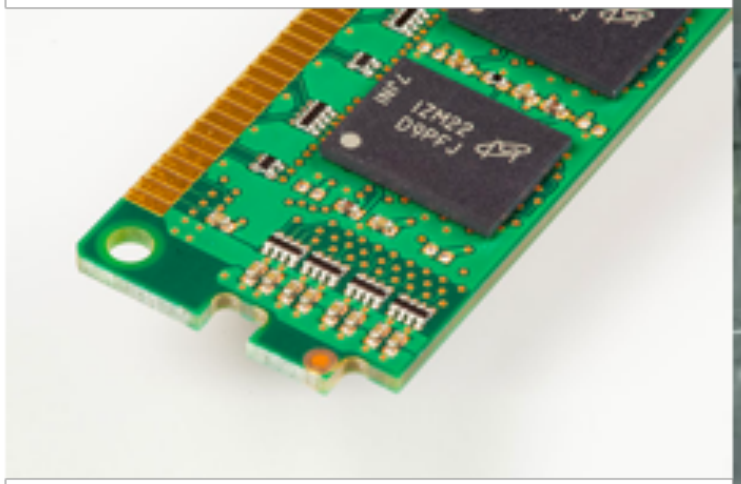


Minimize the access time vs. Cost

Memory

- A typical memory hierarchy





Memory information

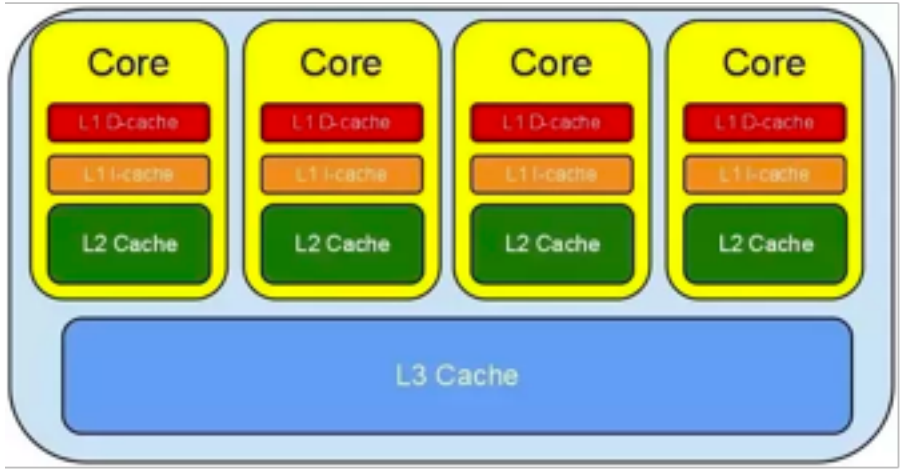
- free

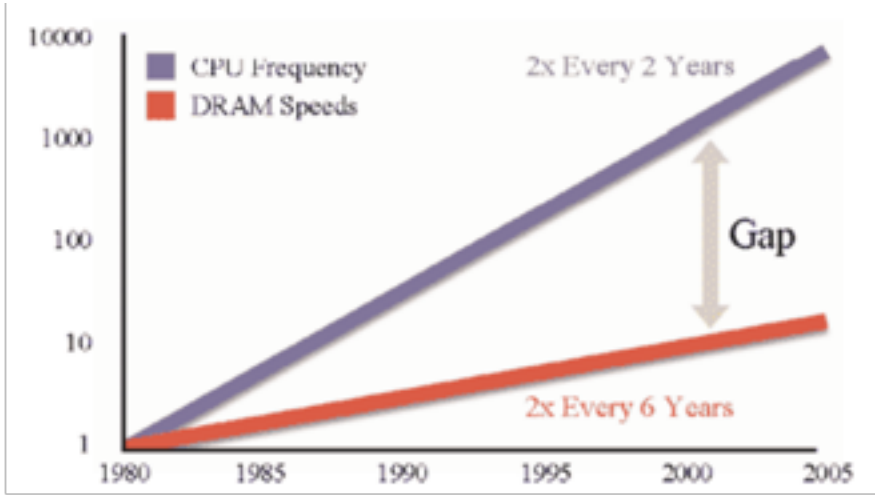
```
administrator@ubuntuvm-1604 ~> free
              total        used        free      shared  buff/cache   available
Mem:        8168756     1348160     3031888       97424     3788708     6383036
Swap:        998396           0      998396
```

- 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

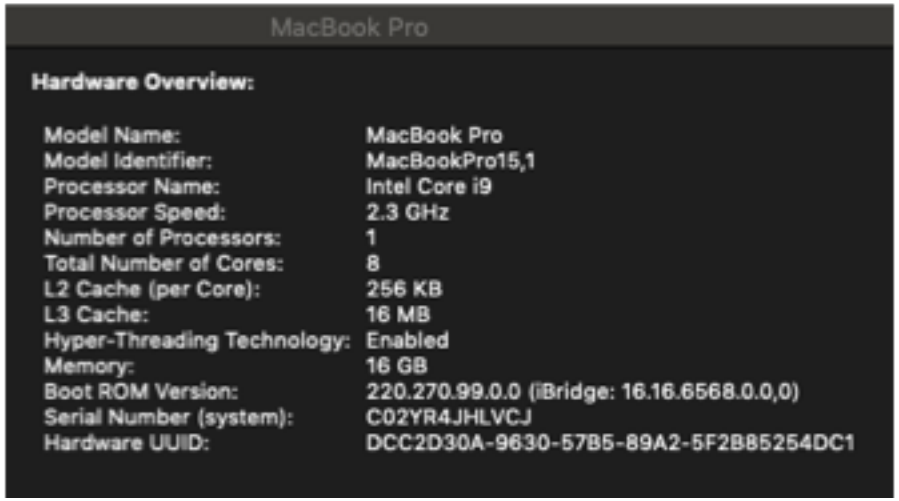
Cache

- Why Cache is important?





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



Check Cache info

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

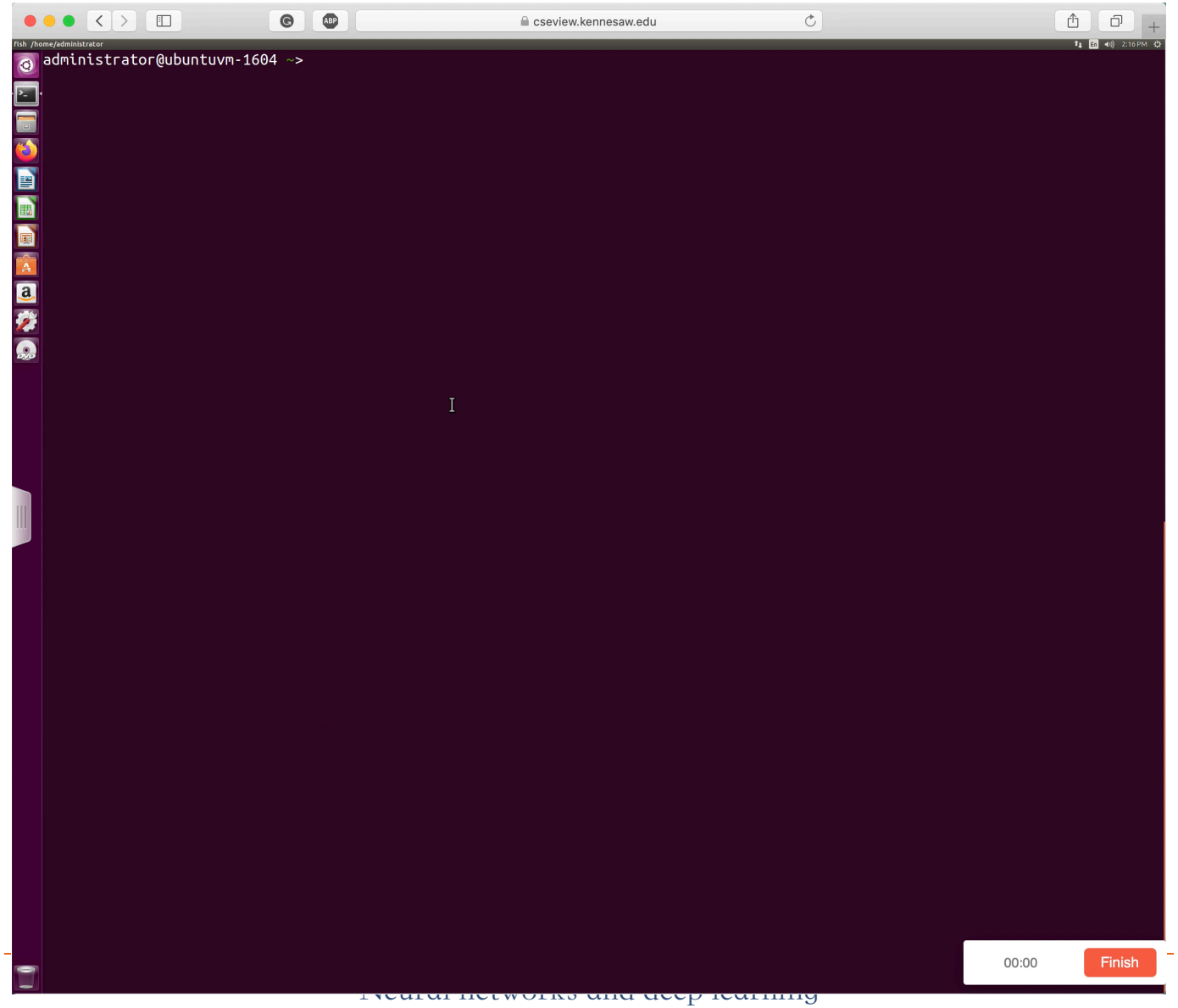
```
********************************************************************************
Cache Topology
********************************************************************************
Level:                  1
Size:                   32 kB
Cache groups:           ( 0 ) ( 1 ) ( 2 ) ( 3 )
-------------------------------------------------------------------------------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

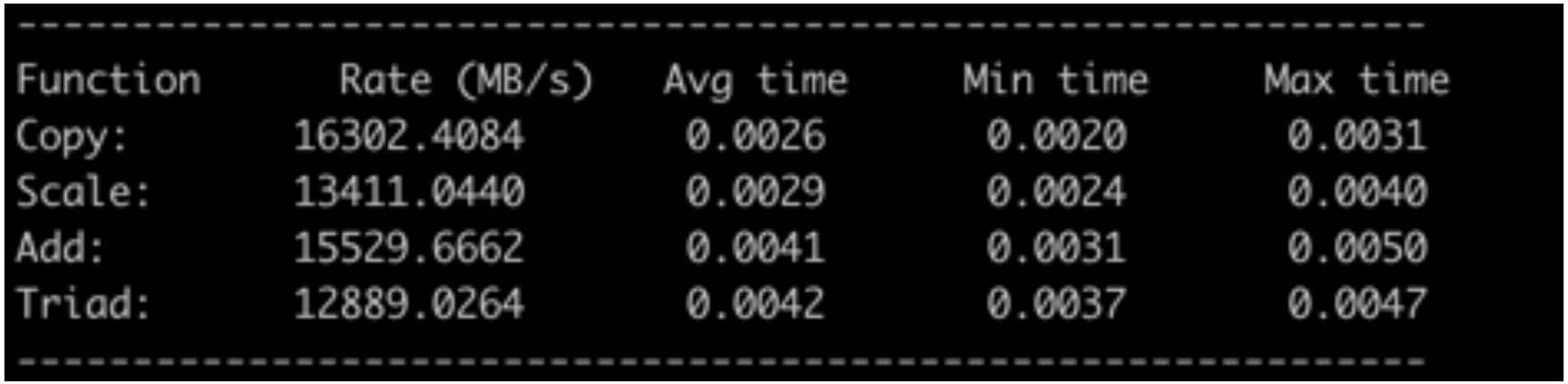
- lstopo: show the CPU cache and logical CPU layout
- Intall: $ sudo apt-get install hwloc
- Run: $ lstopo

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

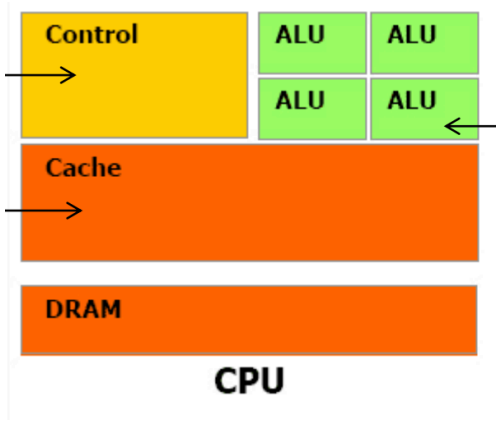
Test: what is the speed of your memory?

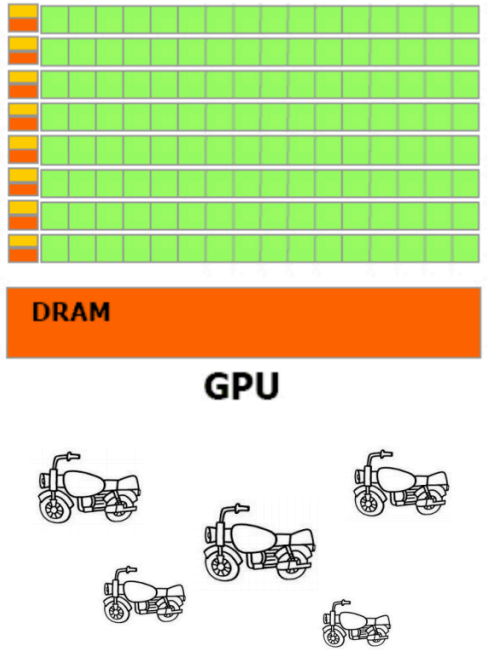
- 1, Download STREAM Benchmark
  - $ wget https://www.nersc.gov/assets/Trinity--NERSC-8-RFP/Benchmarks/Jan9/stream.tar

- 2, Compile & run
  - $ tar xvf stream.tar
    $ gcc stream.c -o stream
    $ ./stream

GPU

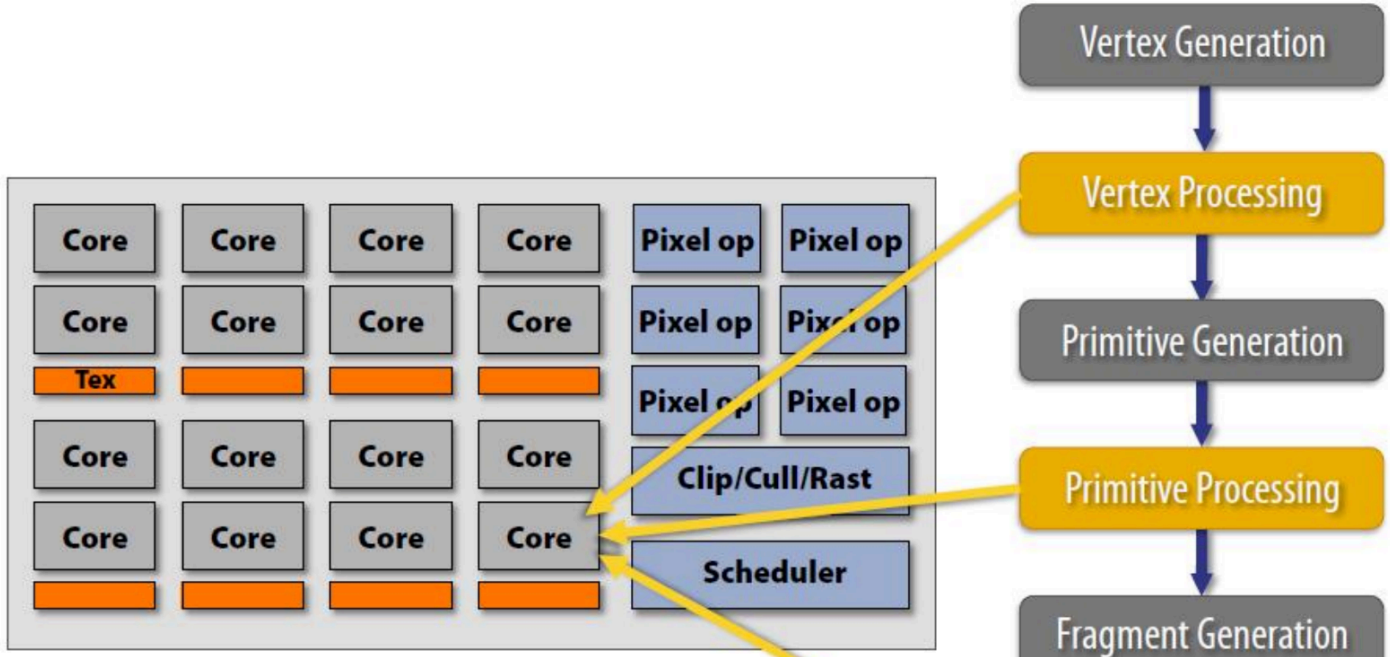
- Graphical Processing Unit
  - The task is to composite and display images with millions of pixels on the screen
  - Good at large numbers, simple, and inherently difficult to do complex calculations (logic, process control)
- GPU has many cores





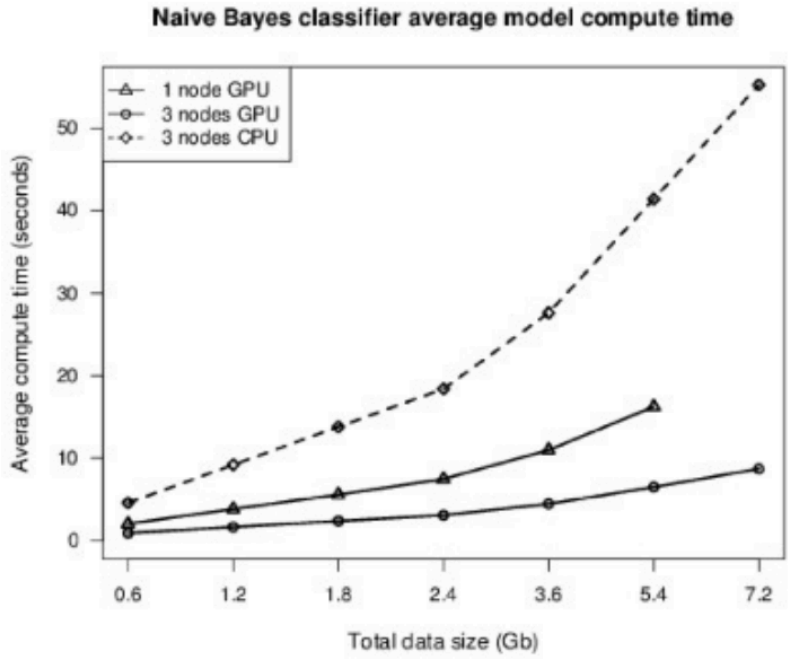
GPU

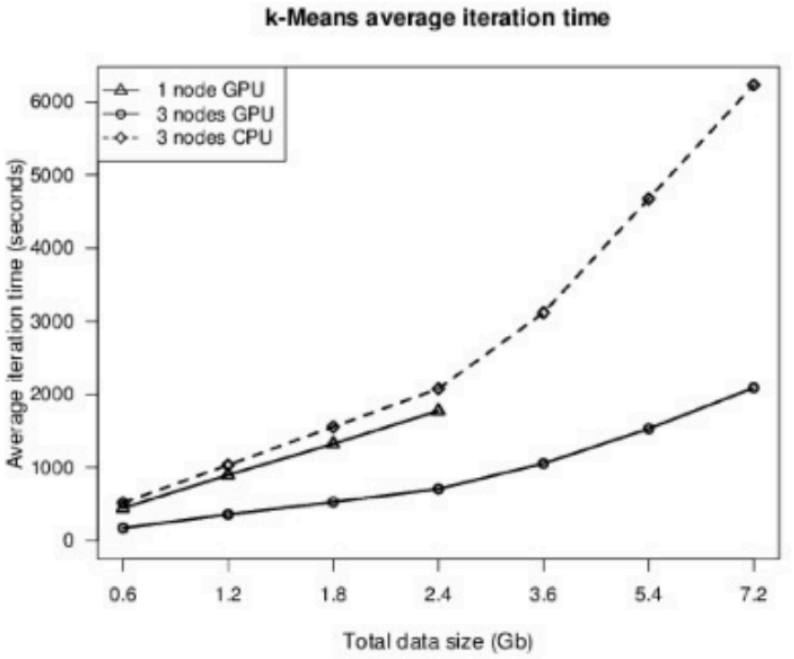
- Each pixel can be given to A CORE (core) processing



GPU

SPEEDUPS



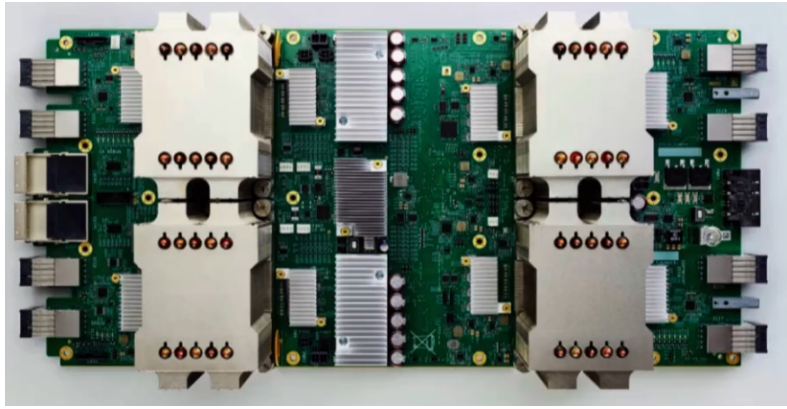


Up to 12x speedup for a same task when nodes are equipped with GPU

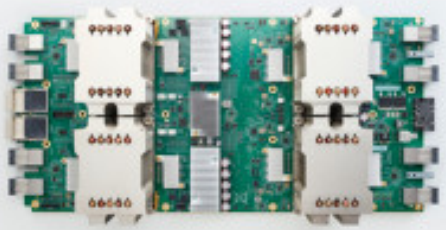
GPU

TPU: Tensor Processing Unit

- Proposal: Design a custom ASIC 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



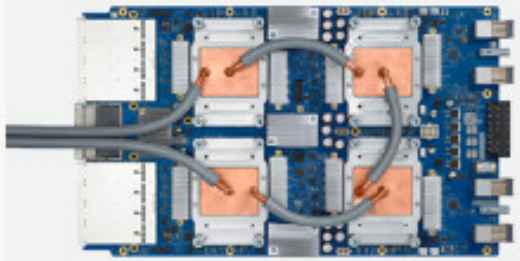
TPU: Tensor Processing Unit



Cloud TPU v2

180 teraflops

64 GB High Bandwidth Memory (HBM)



Cloud TPU v3 Alpha

420 teraflops

128 GB HBM

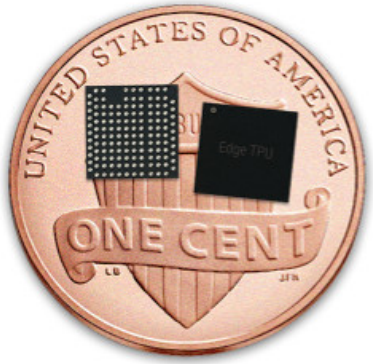
Cloud TPU v2 Pod Alpha

11.5 petaflops

4 TB HBM

2-D toroidal mesh network

Edge TPU



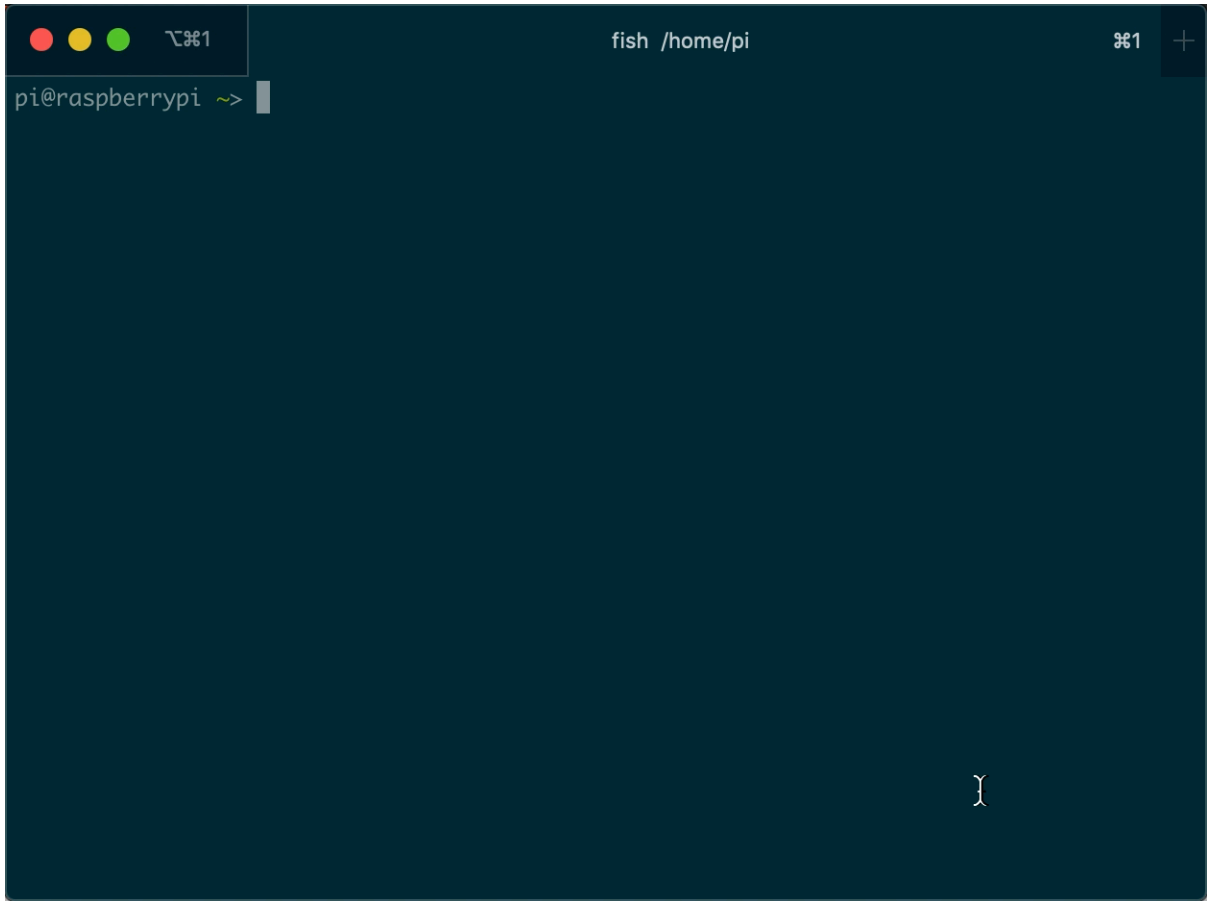
An interesting video introducing hardware

Test: Your VM hardware specs?

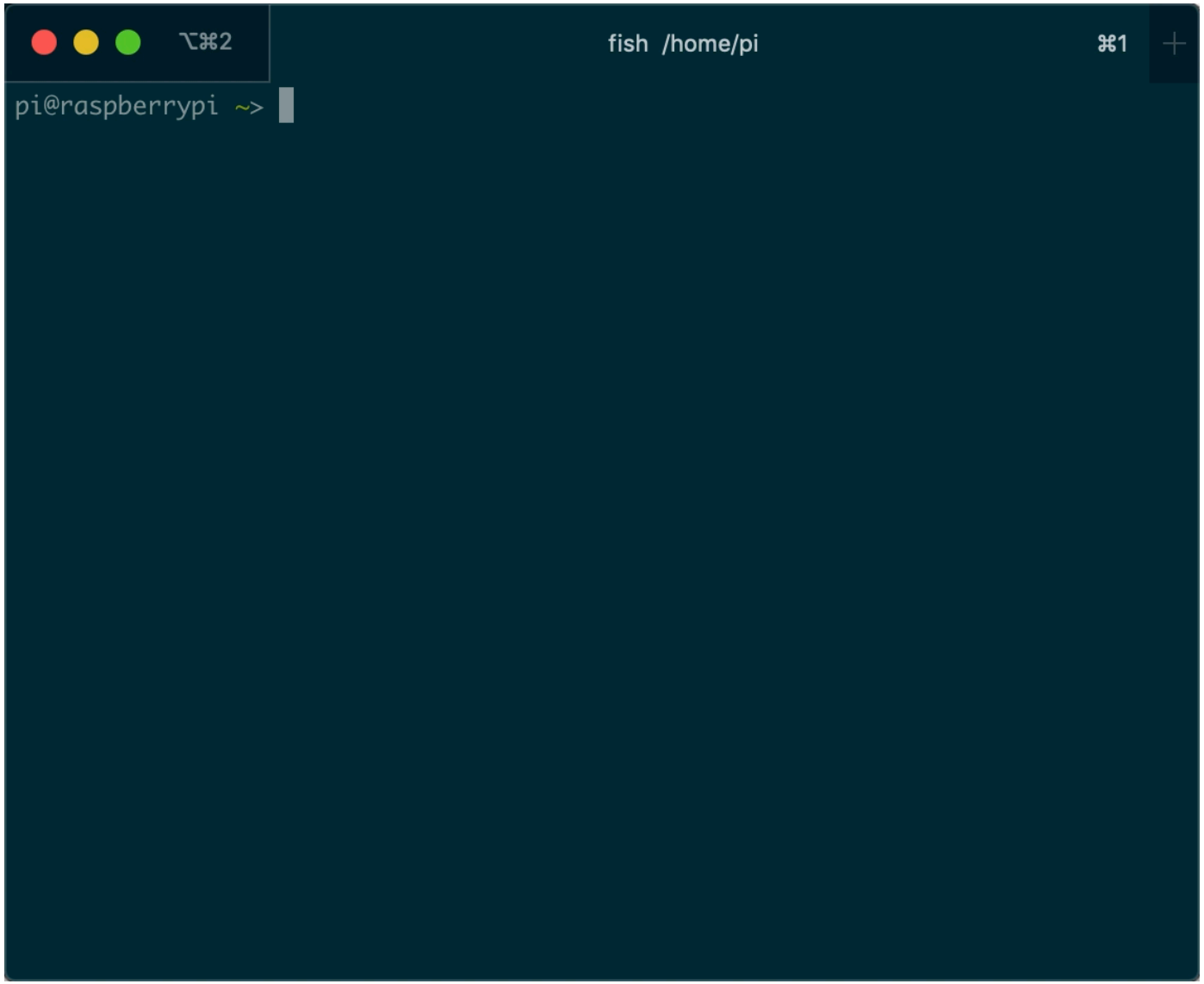
- CPU? → How many cores? Frequency? Topology?
- Cache? → Size?
- Memory? → Size? Speed?
- ~~Disk? → Size?~~
- ~~Network? → Latency? Bandwidth?~~

How to get my machine spec?

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



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



AI benchmark

- AI Benchmark Alpha is an open-source python library for evaluating AI performance of various hardware platforms, including CPUs, GPUs and TPUs. The benchmark is relying on TensorFlow machine learning library and is providing a lightweight and accurate solution for assessing inference and training speed for key Deep Learning models.

- http://ai-benchmark.com/

- https://arxiv.org/pdf/1910.06663.pdf

AI benchmark

- In total, AI Benchmark consists of 42 tests and 19 sections provided below:
  - MobileNet-V2 [classification]
  - Inception-V3 [classification]
  - Inception-V4 [classification]
  - Inception-ResNet-V2 [classification]
  - ResNet-V2-50 [classification]
  - ResNet-V2-152 [classification]
  - VGG-16 [classification]
  - SRCNN 9-5-5 [image-to-image mapping]
  - VGG-19 [image-to-image mapping]
  - ResNet-SRGAN [image-to-image mapping]
  - ResNet-DPED [image-to-image mapping]
  - U-Net [image-to-image mapping]
  - Nvidia-SPADE [image-to-image mapping]
  - ICNet [image segmentation]
  - PSPNet [image segmentation]
  - DeepLab [image segmentation]
  - Pixel-RNN [inpainting]
  - LSTM [sentence sentiment analysis]
  - GNMT [text translation]

Installing AI benchmark

```
apt update -y

apt-get install python3-pip python3-dev nmon fish

pip3 install tensorflow
# or
pip3 install tensorflow-gpu

#check version
python3 -c 'import tensorflow as tf; print(tf.__version__)'  # for Python 3

pip3 install ai-benchmark
```

Run AI benchmark

- Create test.py file as:

```
from ai_benchmark import AIBenchmark
benchmark = AIBenchmark()
results = benchmark.run_micro()
```

- python3 test.py
  - https://www.youtube.com/watch?v=OlbUcSlc0Uc&ab_channel=TonyJ

Monitor the AI-benchmark and visualize the resource consumption

- $ nmon -fTt -s 1 -c 10000

- When benchmark finished, killall nmon. On the disk, you will see file similar as raspberrypi_190927_2015.nmon

- Install nmonchart
  - $ git clone https://github.com/aguther/nmonchart.git

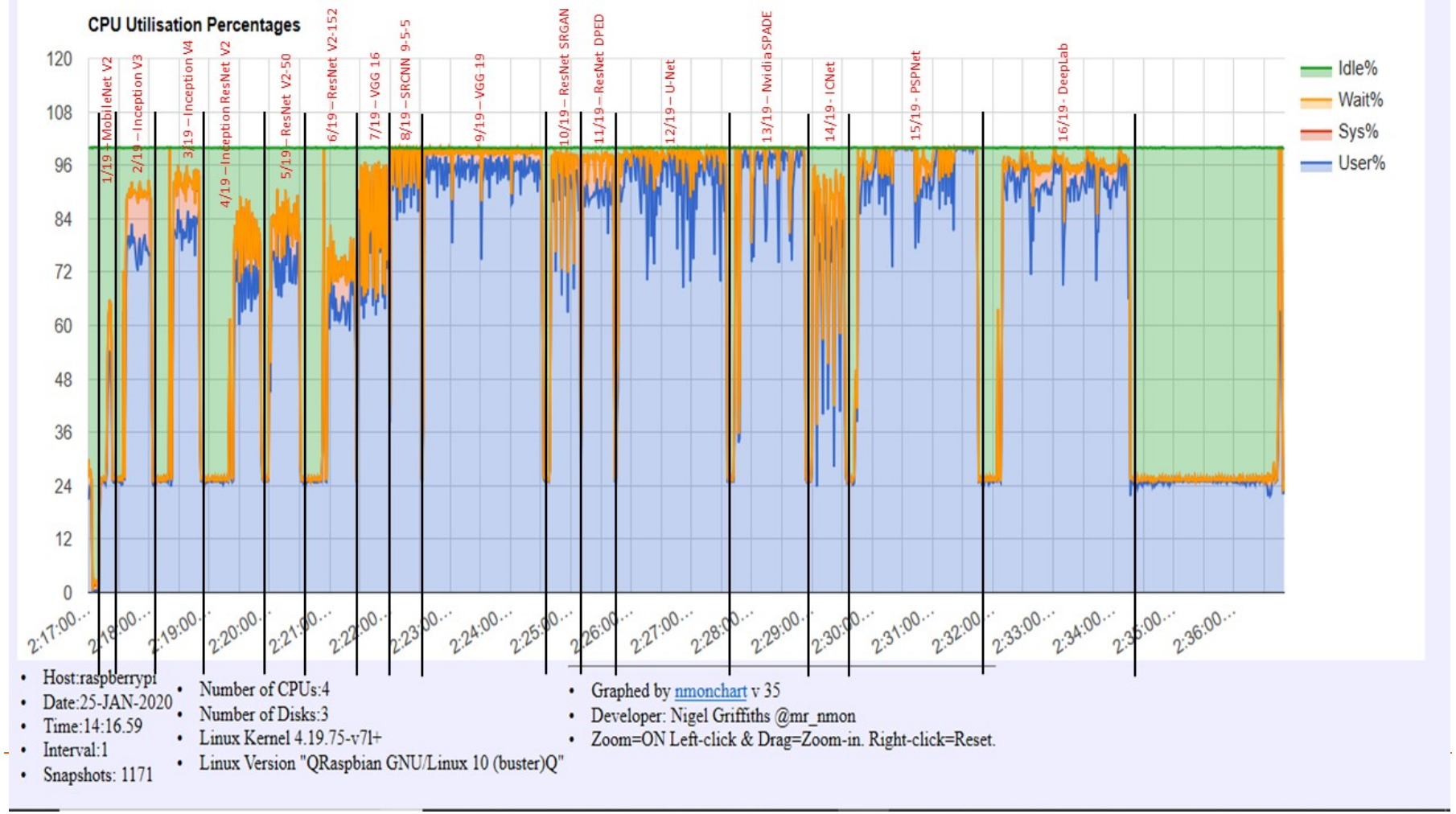
- Change nmonchart to executable
  - $ sudo chmod +x ./nmonchart/nmonchart

- Install ksh
  - $ sudo apt-get install ksh -y

- Transfer nmon file to html file
  - $ nmonchart raspberrypi_190927_2015.nmon raspberrypi_190927_2015.html

Output file

- https://kevinsuo.github.io/code-lab-files/results/rsbp4/2-raspberrypi_200125_1416.html



Output file

- https://kevinsuo.github.io/code-lab-files/results/rsbp4/2-raspberrypi_200125_1416.html