HPC & Parallel Programming

Overview

Kun Suo

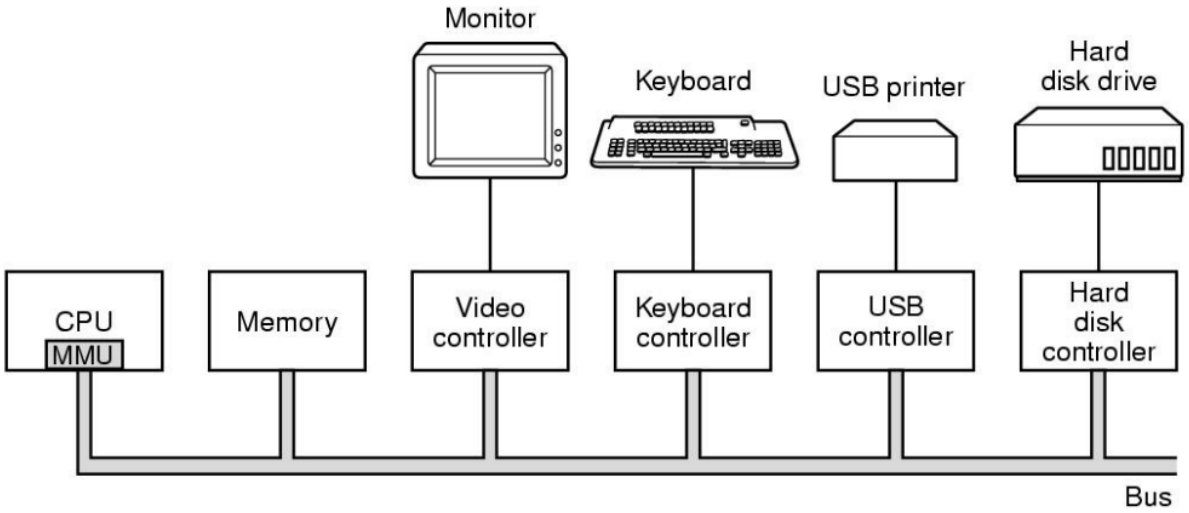
Computer Science, Kennesaw State University

https://kevinsuo.github.io/

Computer Hardware Review

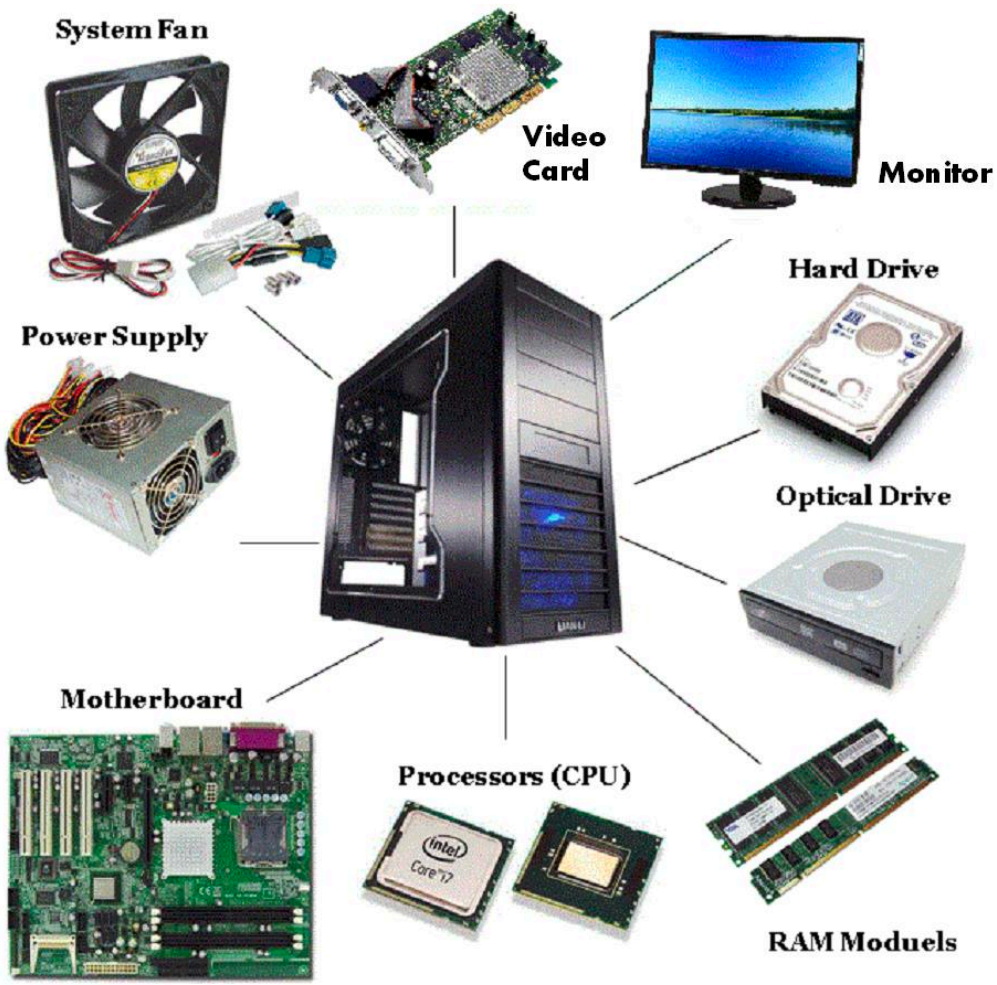
Computer Hardware Review

- Basic components of a simple personal computer

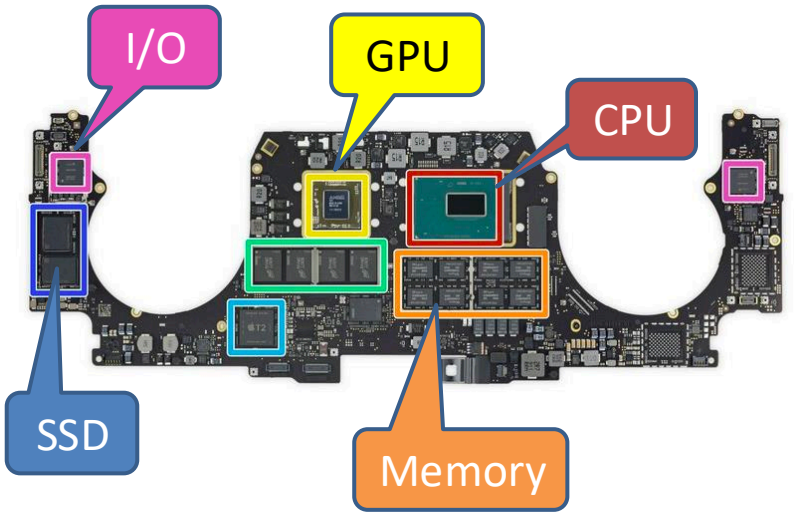


- CPU: data processing
- Memory: volatile data storage
- Disk: persistent data storage
- NIC: inter-machine communication
- Bus: intra-machine communication

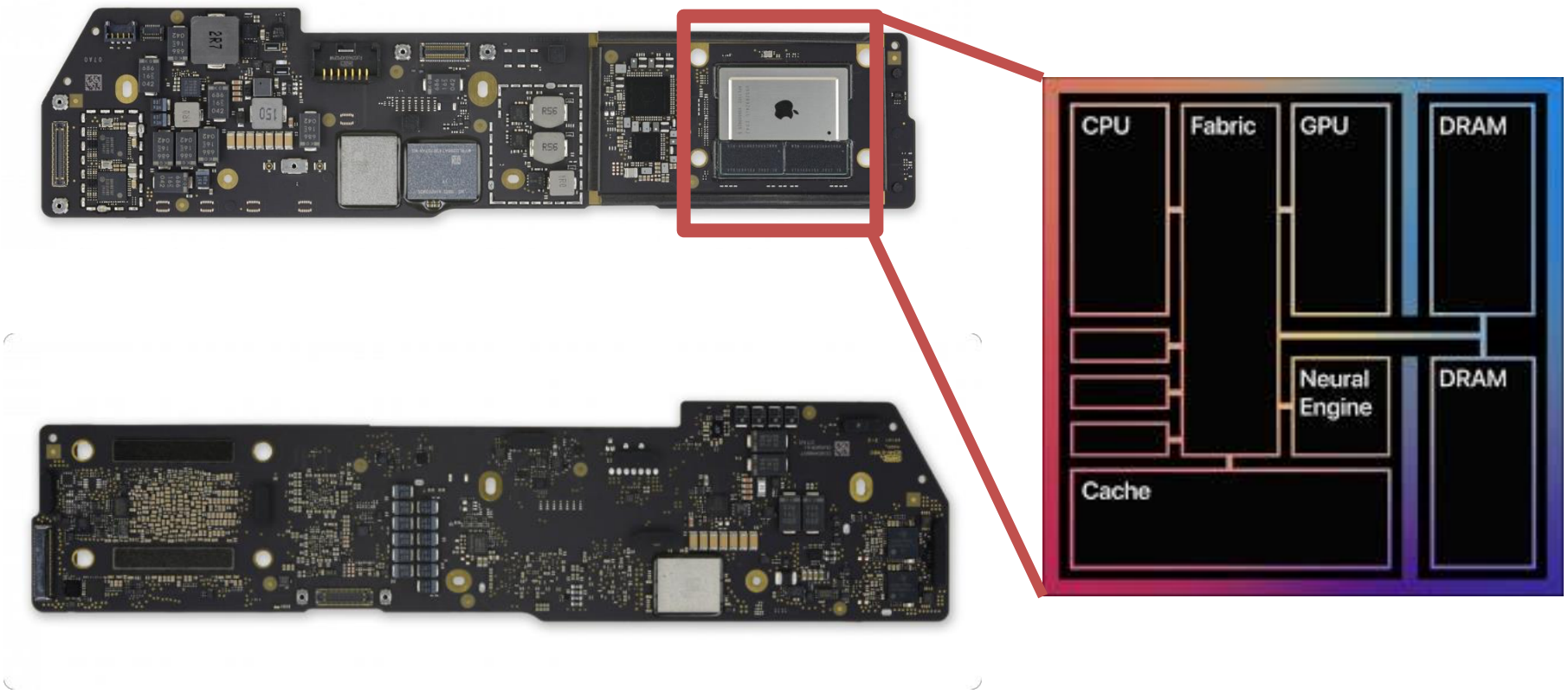
Computer Hardware Review



Computer Hardware Review

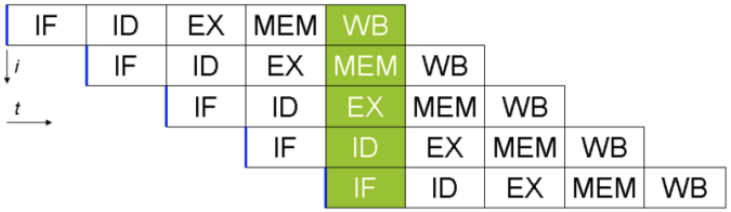


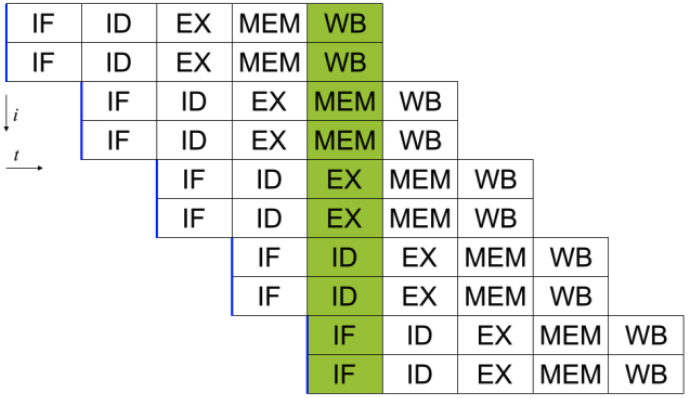
Computer Hardware Review

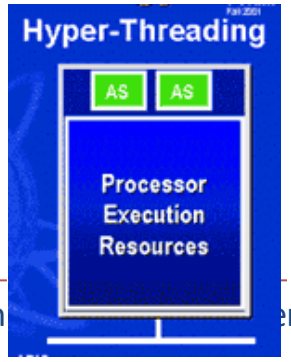


Central Processing Unit (CPU)

- 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

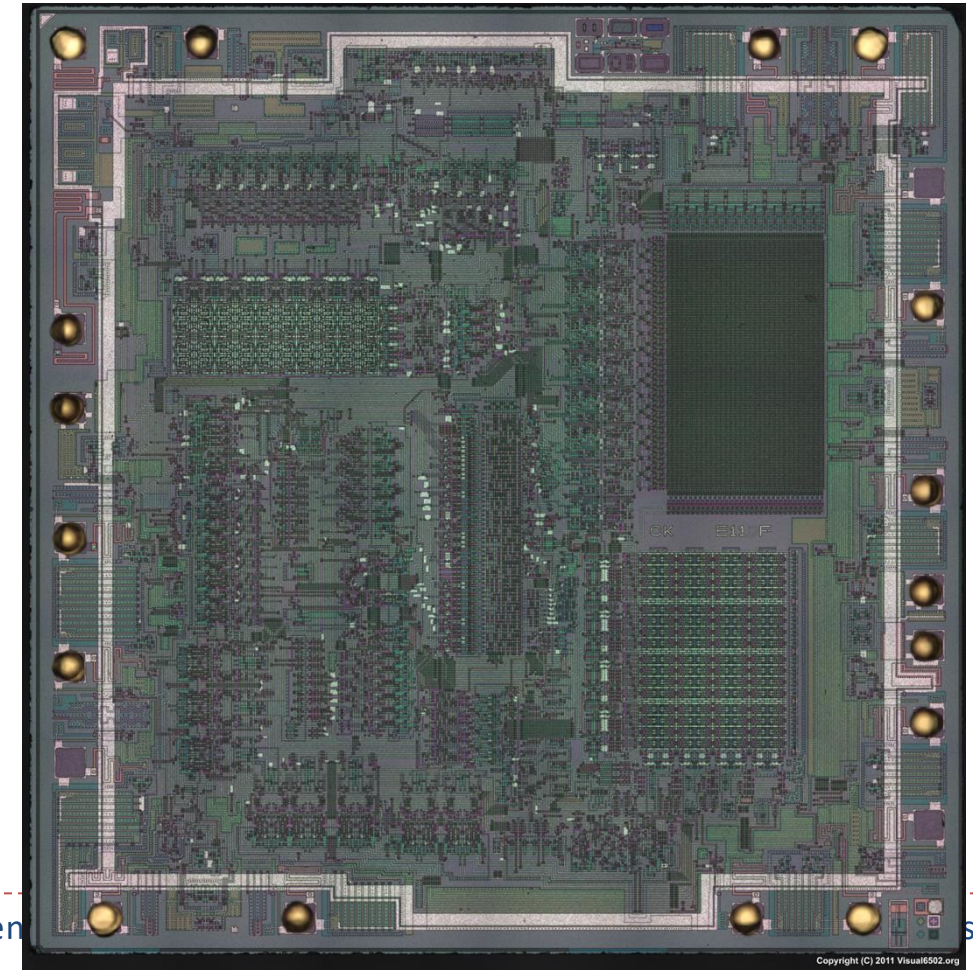






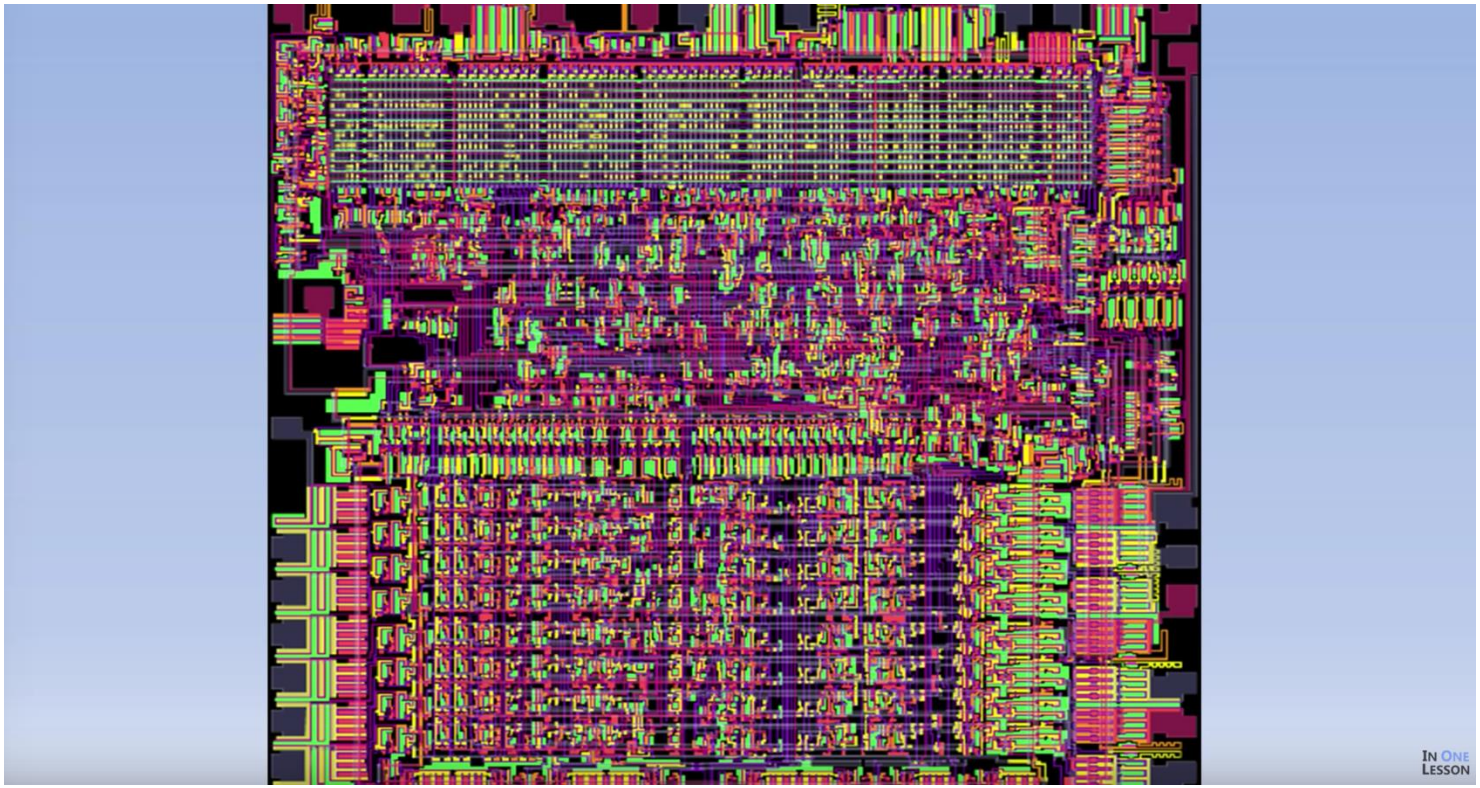
What's inside of CPU?

- http://www.visual6502.org/

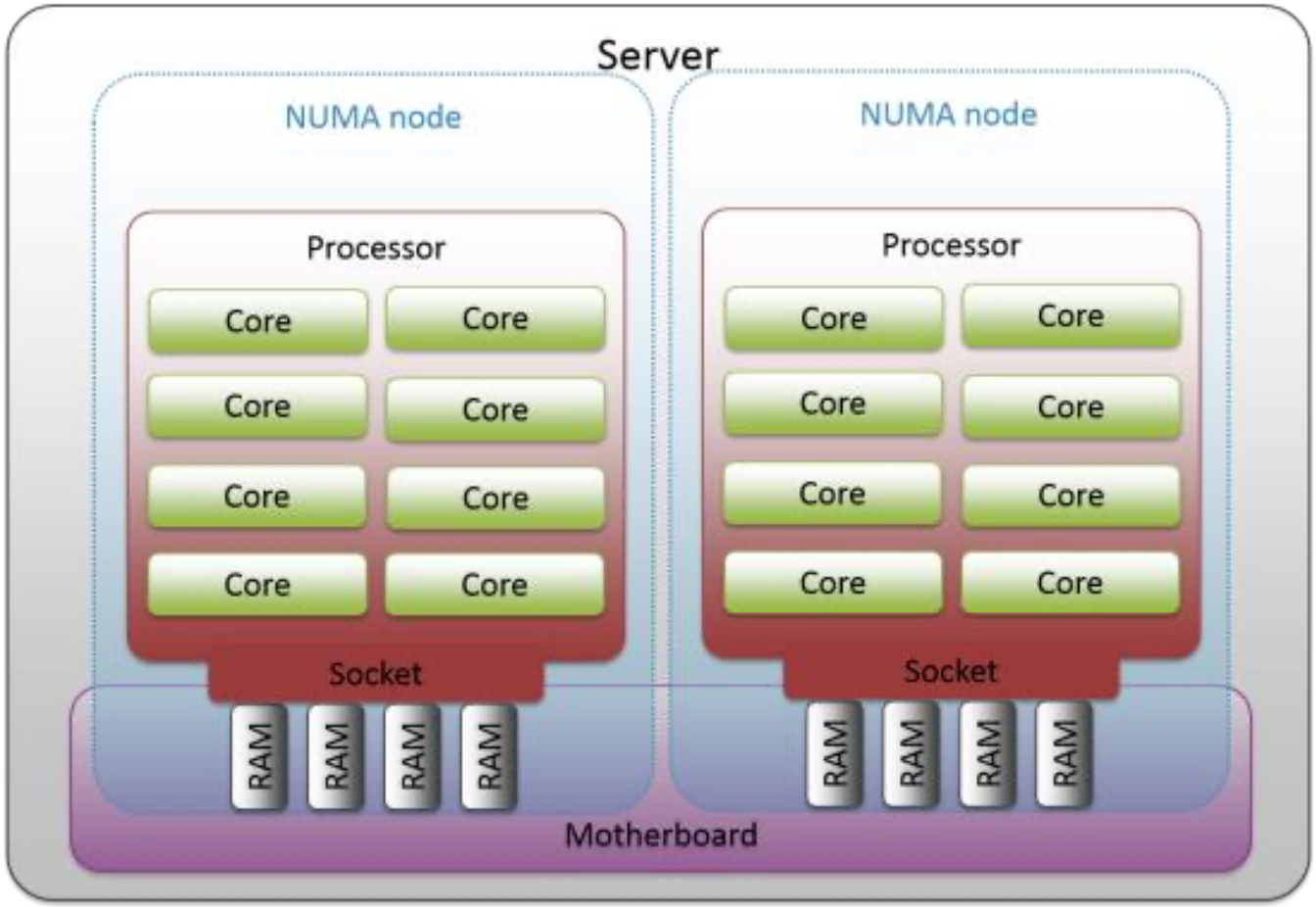


How CPU works?

- 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
```

Tell me your CPU:
1, clock rate?
2, parallelism (num of core)
3, producer & type

CPU information

- lscpu

Tell me your CPU:
1, clock rate?
2, parallelism (num of core)
3, producer & type

```
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

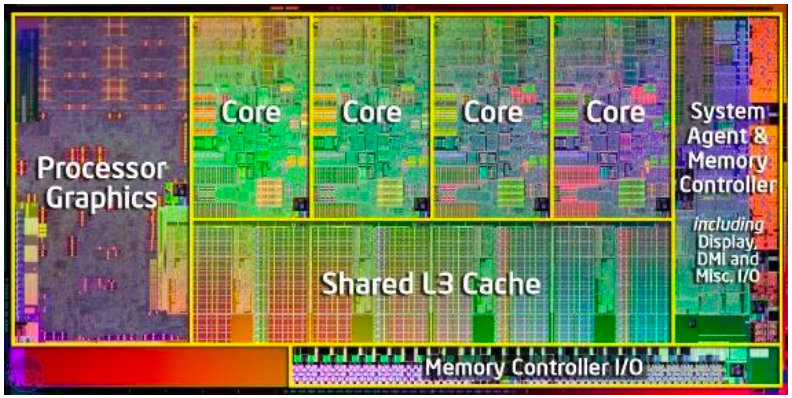
Tell me your CPU:
1, clock rate?
2, parallelism (num of core)
3, producer & type

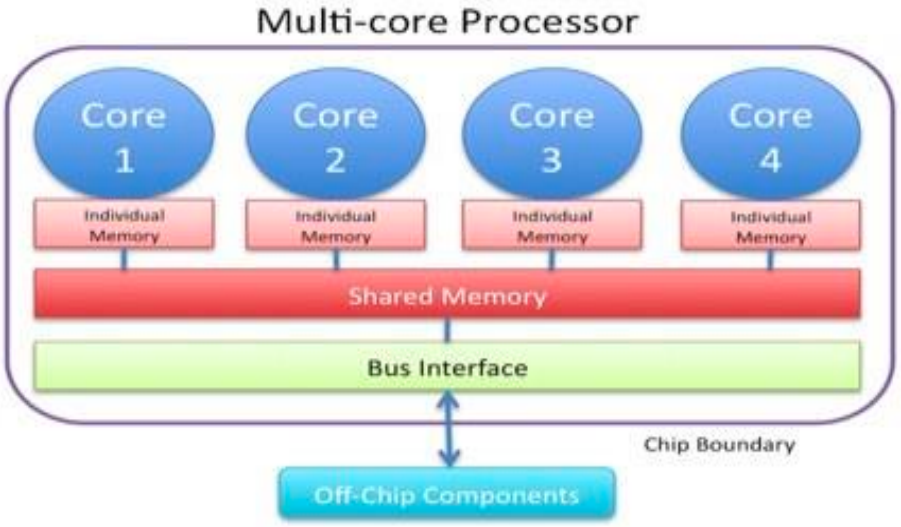
```
administrator@ubuntuvm-1604 ~> cat /proc/cpuinfo
processor       : 0
vendor_id       : GenuineIntel
cpu family      : 6
model           : 79
model name      : Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz
stepping        : 1
microcode       : 0xb000036
cpu MHz         : 2199.998
cache size      : 51200 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_sto
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

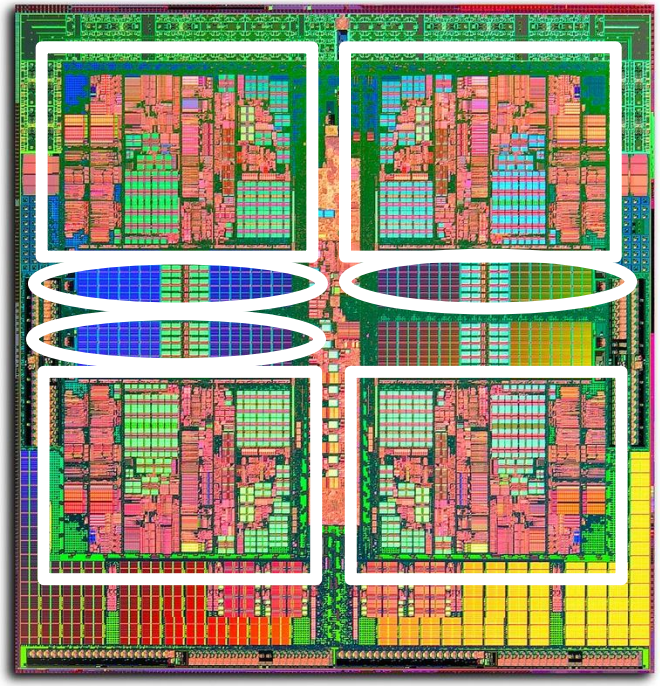


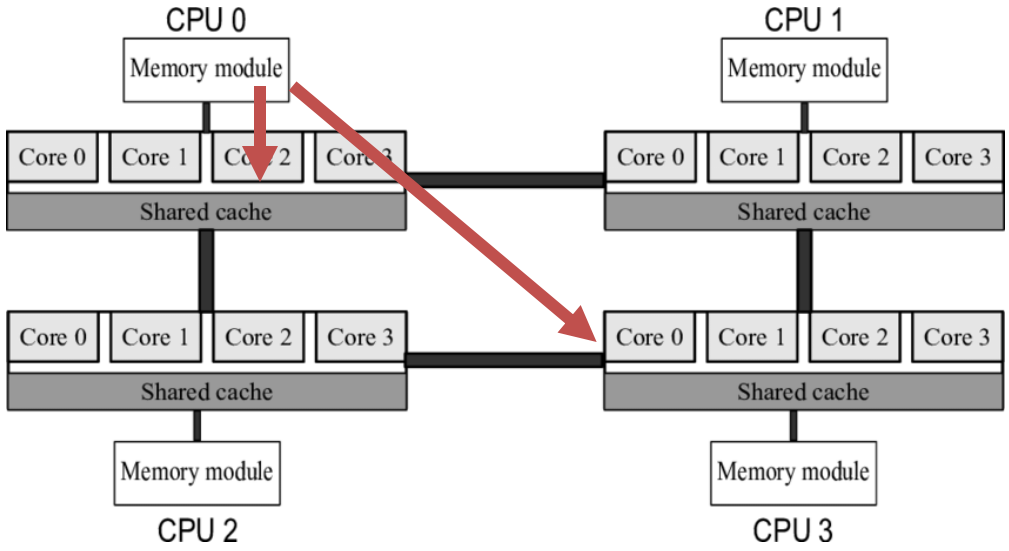


Symmetric multiprocessing (**SMP**)

Multi-Core Processors (NUMA)

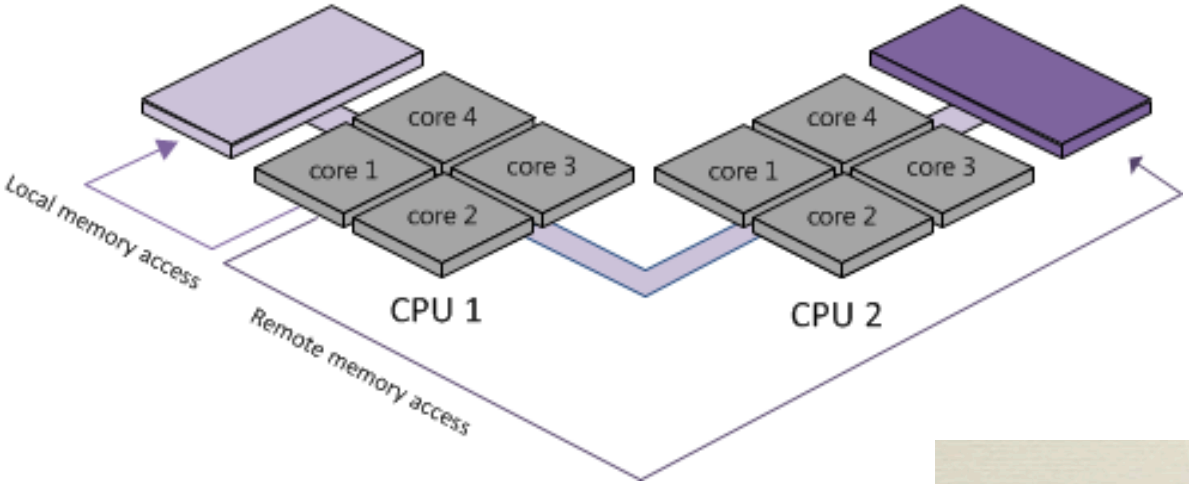
- Multiple CPUs on a single chip

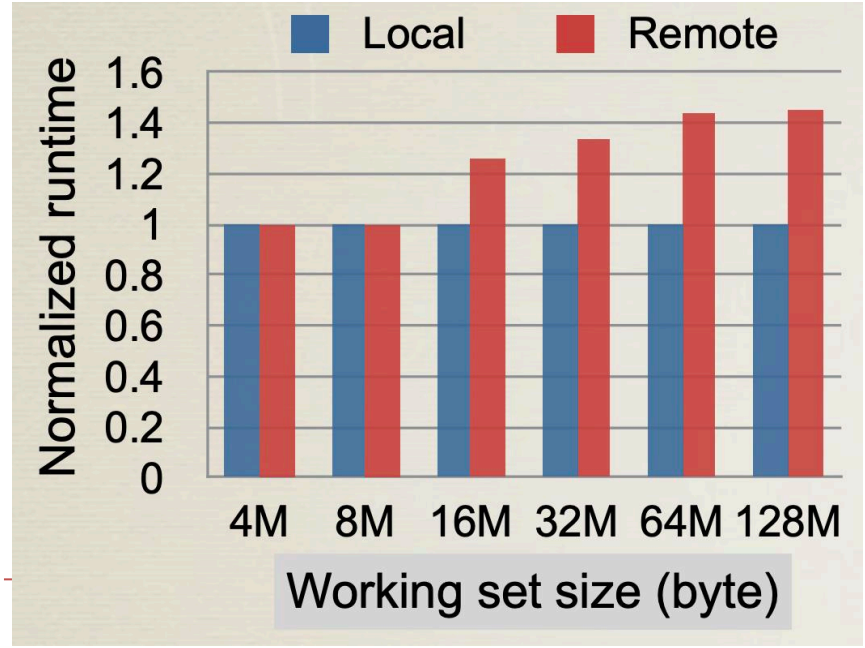




Non-uniform memory access (**NUMA**)

Multi-Core Processors (NUMA)





Check CPU topology

- $ likwid-topology –g

```
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 | |
| +--------+ +--------+ +--------+ +--------+ |
+-----------------------------------------+
```

NUMA example:

https://github.com/RRZE-HPC/likwid/wiki/TutorialNUMA

Benchmarking my CPU

- 7-Zip is a file compression tool that can do extreme levels of compression on files and store them in a reduced size 7z archive format.

- To install 7-Zip in Ubuntu, run command:
  - $ sudo apt install p7zip-full
- To run single threaded benchmark, use the command below:
  - $ 7z b -mmt1
- Multi-threaded benchmarking can be run by using the following command:
  - $ 7z b

Benchmarking my CPU

- Output

```
administrator@ubuntu1804vm ~> 7z b -mmt1

7-Zip [64] 16.02 : Copyright (c) 1999-2016 Igor Pavlov : 2016-05-21
p7zip Version 16.02 (locale=en_US.UTF-8,Utf16=on,HugeFiles=on,64 bits,2 CPUs Intel(R) X
6F1),ASM,AES-NI)

Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz (406F1)
CPU Freq:  1816  2664  2660  2661  2655  2674  2678  2676  2678

RAM size:    7962 MB,  # CPU hardware threads:   2
RAM usage:    435 MB,  # Benchmark threads:      1

                    Compressing  |                  Decompressing
Dict     Speed Usage    R/U Rating  |      Speed Usage    R/U Rating
         KiB/s     %   MIPS   MIPS  |      KiB/s     %   MIPS   MIPS

22:       3372   100   3289   3281  |      33177   100   2834   2833
23:       2899    92   3212   2954  |      33196    99   2891   2873
24:       3017   100   3245   3245  |      32716   100   2872   2872
25:       2858   100   3264   3264  |      32319   100   2877   2877
--------------------------------  | ------------------------------
Avr:              98   3253   3186  |             100   2868   2864
Tot:              99   3061   3025
```

The results are in million instructions per second (MIPS). You can compare these results with other CPU models from here.

Benchmarking my CPU

- S-tui is a CPU resource monitor that uses a terminal based graphical interface to display data and graphs.

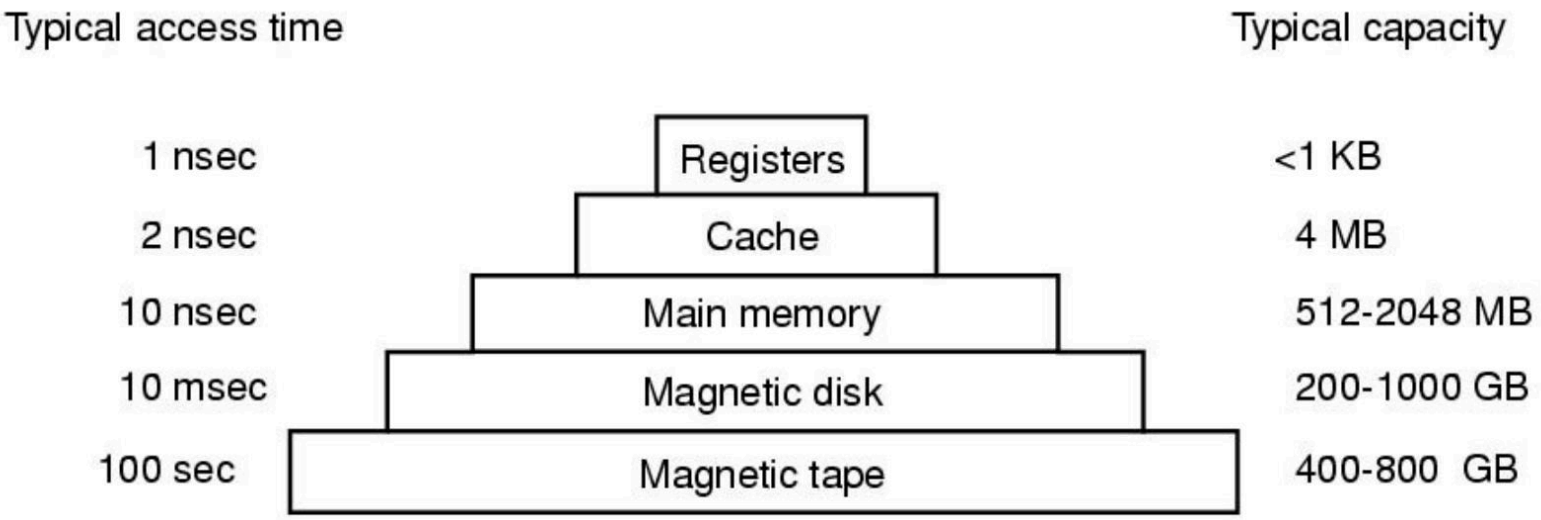
- It supports CPU stress testing using the Stress app listed above while giving you a nice bird's eye view on the various CPU parameters like frequency, temperature and utilization.

- Installation
  - $ sudo add-apt-repository ppa:amanusk/python-s-tui
  - $ sudo apt-get update
  - $ sudo apt install stress python3-distutils –y
  - $ sudo apt-get install python3-s-tui
- Run
  - $ s-tui

Demo: https://youtu.be/objk3eTN7jg

Memory

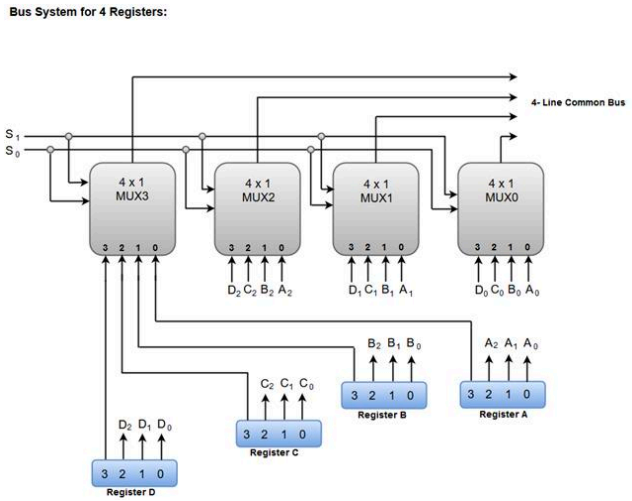
- A typical memory hierarchy

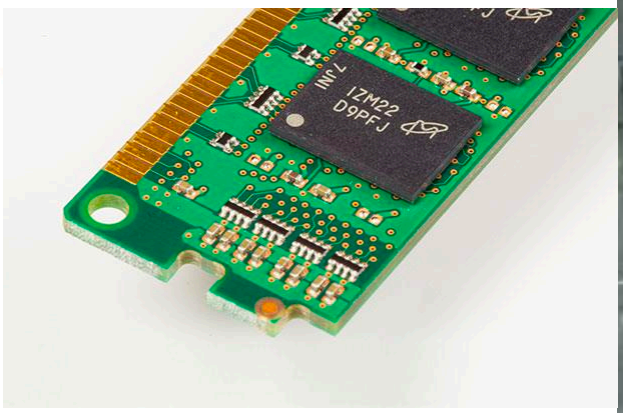


Minimize the access time vs. Cost

Memory

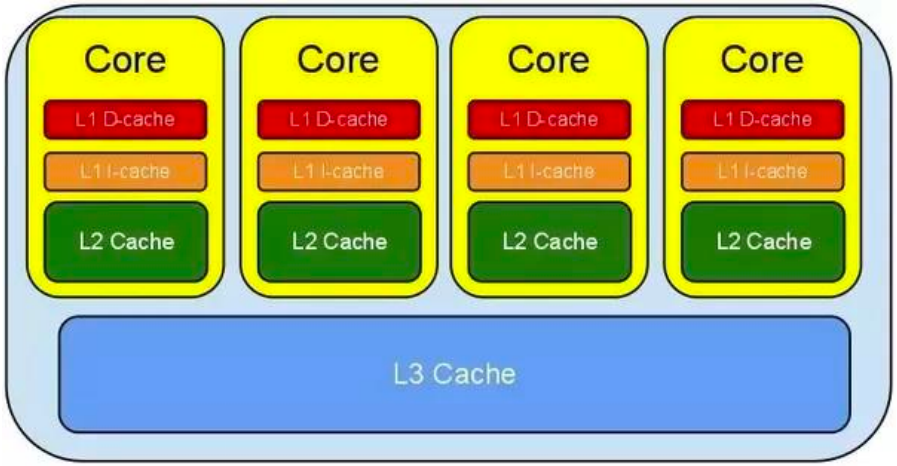
- A typical memory hierarchy

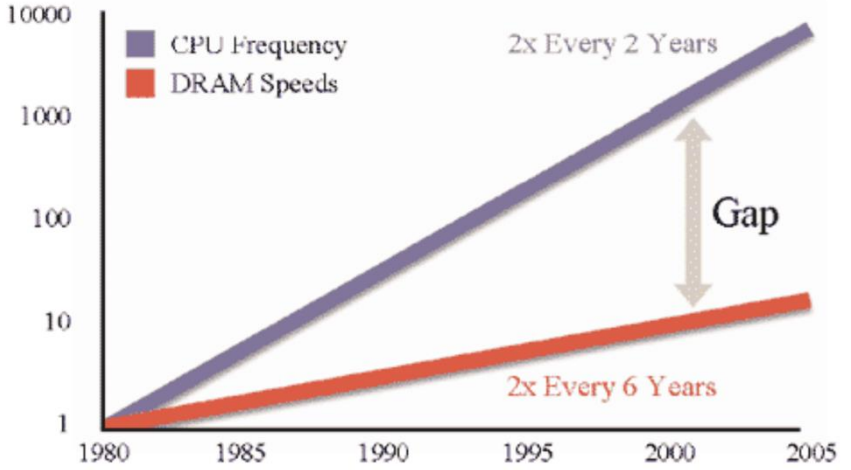




Cache

- 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-9630-57B5-89A2-5F2B85254DC1

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

- 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

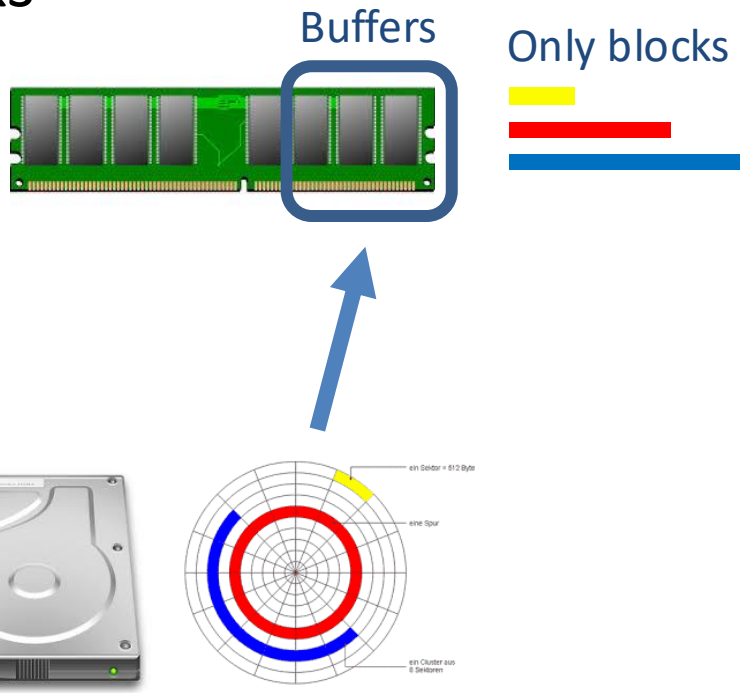
Memory information

- cat /proc/meminfo
  - MemTotal
  - MemFree
  - MemAvailable
  - Buffers
  - Cached
  - SwapCached
  - SwapTotal
  - SwapFree

```
administrator@ubuntuvm-1604 ~> cat /proc/meminfo
MemTotal:        8168756 kB
MemFree:         3034596 kB
MemAvailable:    6385744 kB
Buffers:          287636 kB
Cached:          3188812 kB
SwapCached:            0 kB
Active:          2523764 kB
Inactive:        2177968 kB
Active(anon):    1226312 kB
Inactive(anon):    96392 kB
Active(file):    1297452 kB
Inactive(file):  2081576 kB
Unevictable:          48 kB
Mlocked:              48 kB
SwapTotal:       998396 kB
SwapFree:        998396 kB
Dirty:              264 kB
Writeback:            0 kB
AnonPages:       1225344 kB
Mapped:           304212 kB
Shmem:             97424 kB
Slab:             312268 kB
SReclaimable:     278104 kB
SUnreclaim:        34164 kB
KernelStack:        8896 kB
PageTables:        31004 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     5082772 kB
Committed_AS:    5183160 kB
VmallocTotal:   34359738367 kB
VmallocUsed:           0 kB
VmallocChunk:          0 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
CmaTotal:              0 kB
CmaFree:               0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
DirectMap4k:      135040 kB
DirectMap2M:     5107712 kB
DirectMap1G:     5242880 kB
```

Cache vs Buffer

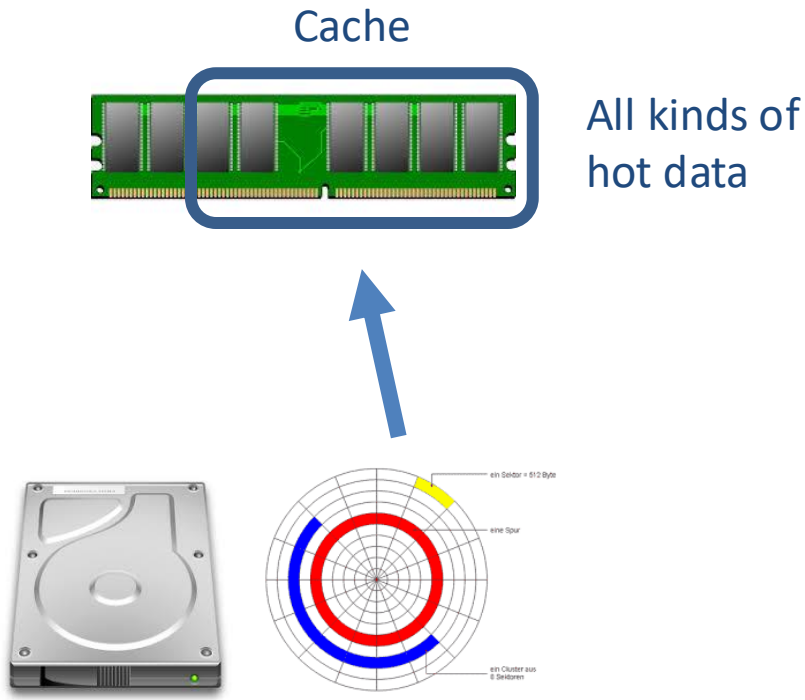
- "Buffers" represent how much portion of RAM is dedicated to cache disk blocks




```
administrator@ubuntuvm-1604 ~> cat /proc/meminfo
MemTotal:        8168756 kB
MemFree:         3034596 kB
MemAvailable:    6385744 kB
Buffers:          287636 kB
Cached:          3188812 kB
SwapCached:            0 kB
Active:          2523764 kB
Inactive:        2177968 kB
Active(anon):    1226312 kB
Inactive(anon):    96392 kB
Active(file):    1297452 kB
Inactive(file):  2081576 kB
Unevictable:          48 kB
Mlocked:              48 kB
SwapTotal:        998396 kB
SwapFree:         998396 kB
Dirty:               264 kB
Writeback:             0 kB
AnonPages:       1225344 kB
Mapped:           304212 kB
Shmem:             97424 kB
Slab:             312268 kB
SReclaimable:     278104 kB
SUnreclaim:        34164 kB
KernelStack:        8896 kB
PageTables:        31004 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     5082772 kB
Committed_AS:    5183160 kB
VmallocTotal:   34359738367 kB
VmallocUsed:           0 kB
VmallocChunk:          0 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
CmaTotal:              0 kB
CmaFree:               0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
DirectMap4k:      135040 kB
DirectMap2M:     5107712 kB
DirectMap1G:     5242880 kB
```

Cache vs Buffer

- "Cached": keep as much data as possible in memory, especially those data that we need to access frequently



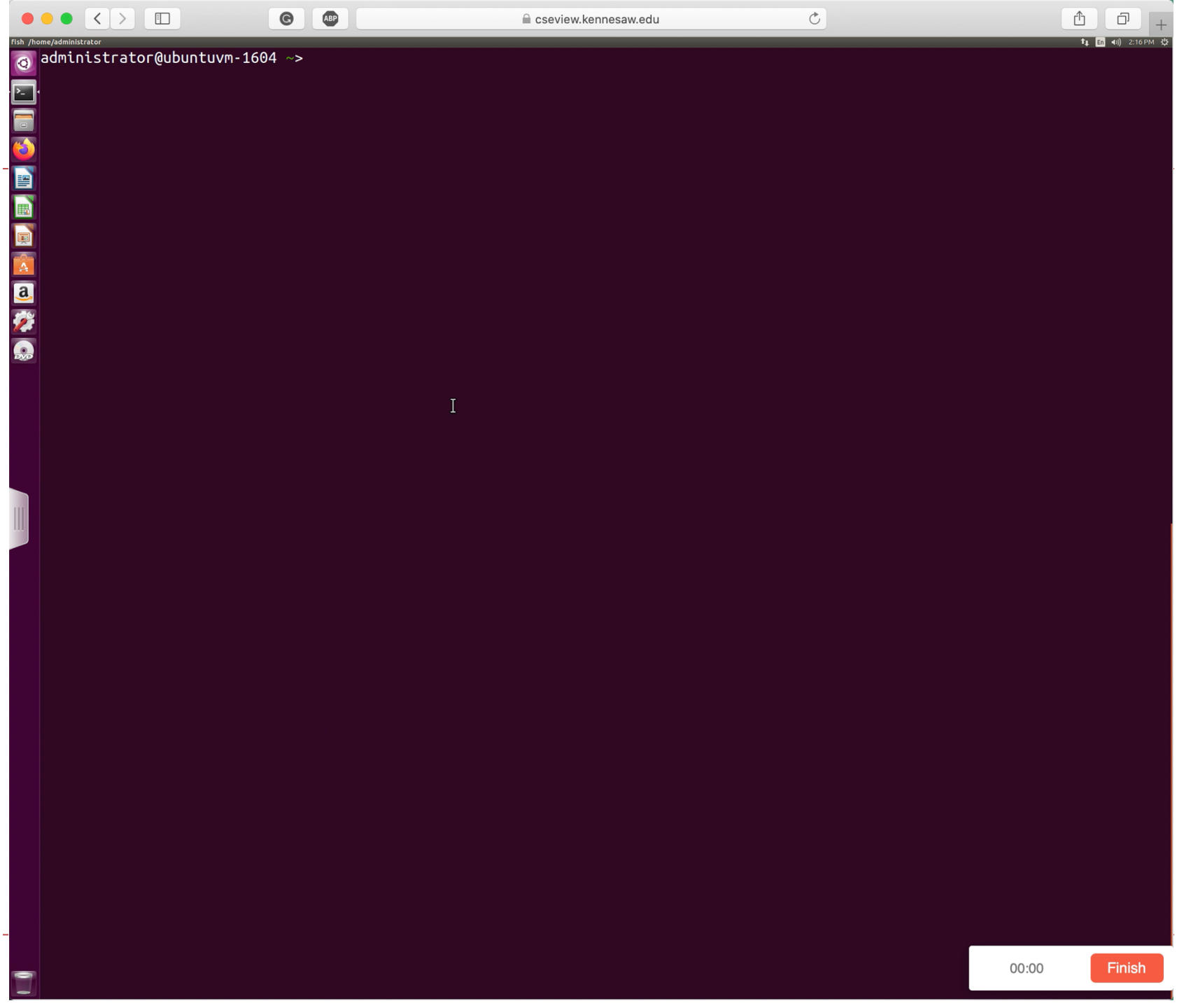

```
administrator@ubuntuvm-1604 ~> cat /proc/meminfo
MemTotal:        8168756 kB
MemFree:         3034596 kB
MemAvailable:    6385744 kB
Buffers:          287636 kB
Cached:          3188812 kB
SwapCached:            0 kB
Active:          2523764 kB
Inactive:        2177968 kB
Active(anon):    1226312 kB
Inactive(anon):    96392 kB
Active(file):    1297452 kB
Inactive(file):  2081576 kB
Unevictable:          48 kB
Mlocked:              48 kB
SwapTotal:       998396 kB
SwapFree:        998396 kB
Dirty:              264 kB
Writeback:            0 kB
AnonPages:       1225344 kB
Mapped:           304212 kB
Shmem:             97424 kB
Slab:             312268 kB
SReclaimable:     278104 kB
SUnreclaim:        34164 kB
KernelStack:        8896 kB
PageTables:        31004 kB
NFS_Unstable:          0 kB
Bounce:                0 kB
WritebackTmp:          0 kB
CommitLimit:     5082772 kB
Committed_AS:    5183160 kB
VmallocTotal:   34359738367 kB
VmallocUsed:           0 kB
VmallocChunk:          0 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
ShmemHugePages:        0 kB
ShmemPmdMapped:        0 kB
CmaTotal:              0 kB
CmaFree:               0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
DirectMap4k:      135040 kB
DirectMap2M:     5107712 kB
DirectMap1G:     5242880 kB
```

Memory information

- lstopo: show the CPU cache and logical CPU layout

- Install: $ sudo apt-get install hwloc

- Run: $ lstopo


cseview.kennesaw.edu
fish /home/administrator
2:16 PM
administrator@ubuntuvm-1604 ~>
00:00
Finish

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

Test: what is the speed of your memory?

- 1, install memory bandwidth app

  $ sudo apt-get install mbw

- 2, run

  $ mbw -q -n 100 256

n: run 100 times

256: means test 256MB in memory