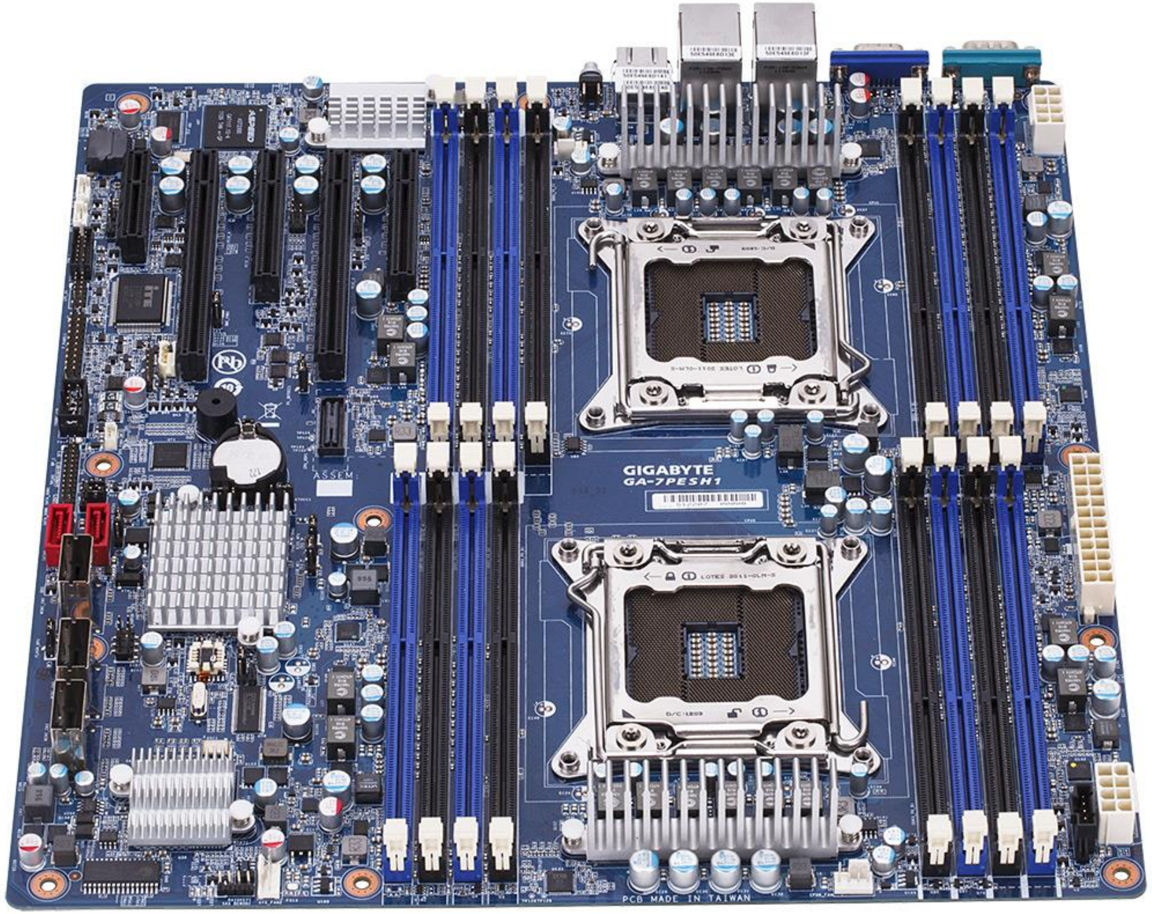
# HYPERTHREADING

DAVID PERRONE AND RUSSELL ROYALITY

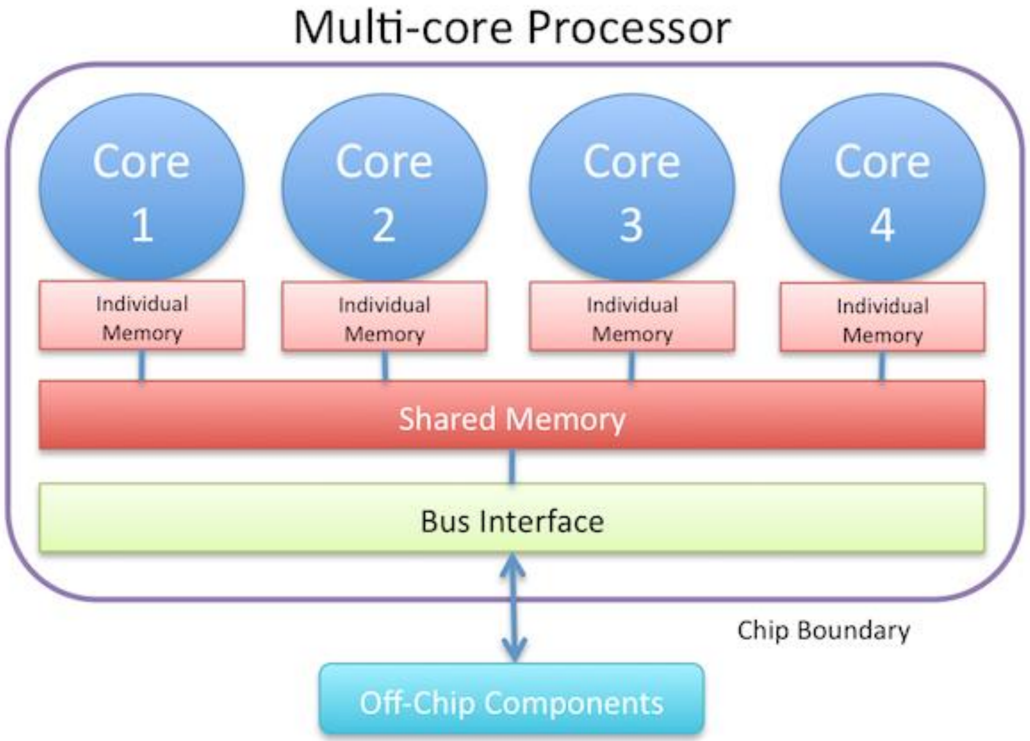
# CPU

- Receives instructions from software and hardware
- More CPUs = more processing power
  - Expensive, requires space

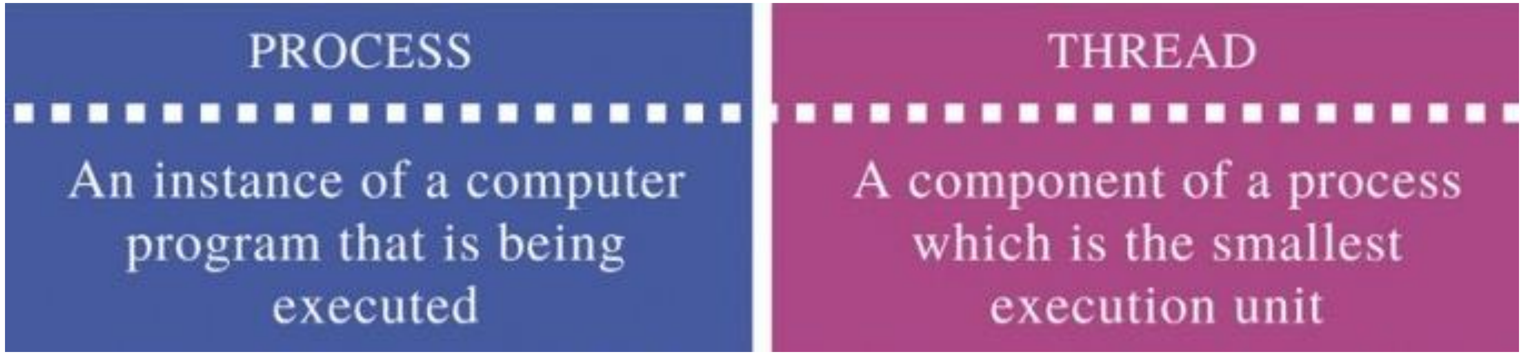


# CORE

- Hardware Component
- Execution unit inside the CPU that receives and executes instructions.
- Contains an ALU, control unit, and registers
- One CPU can contain multiple cores
    - Each core can run its own process



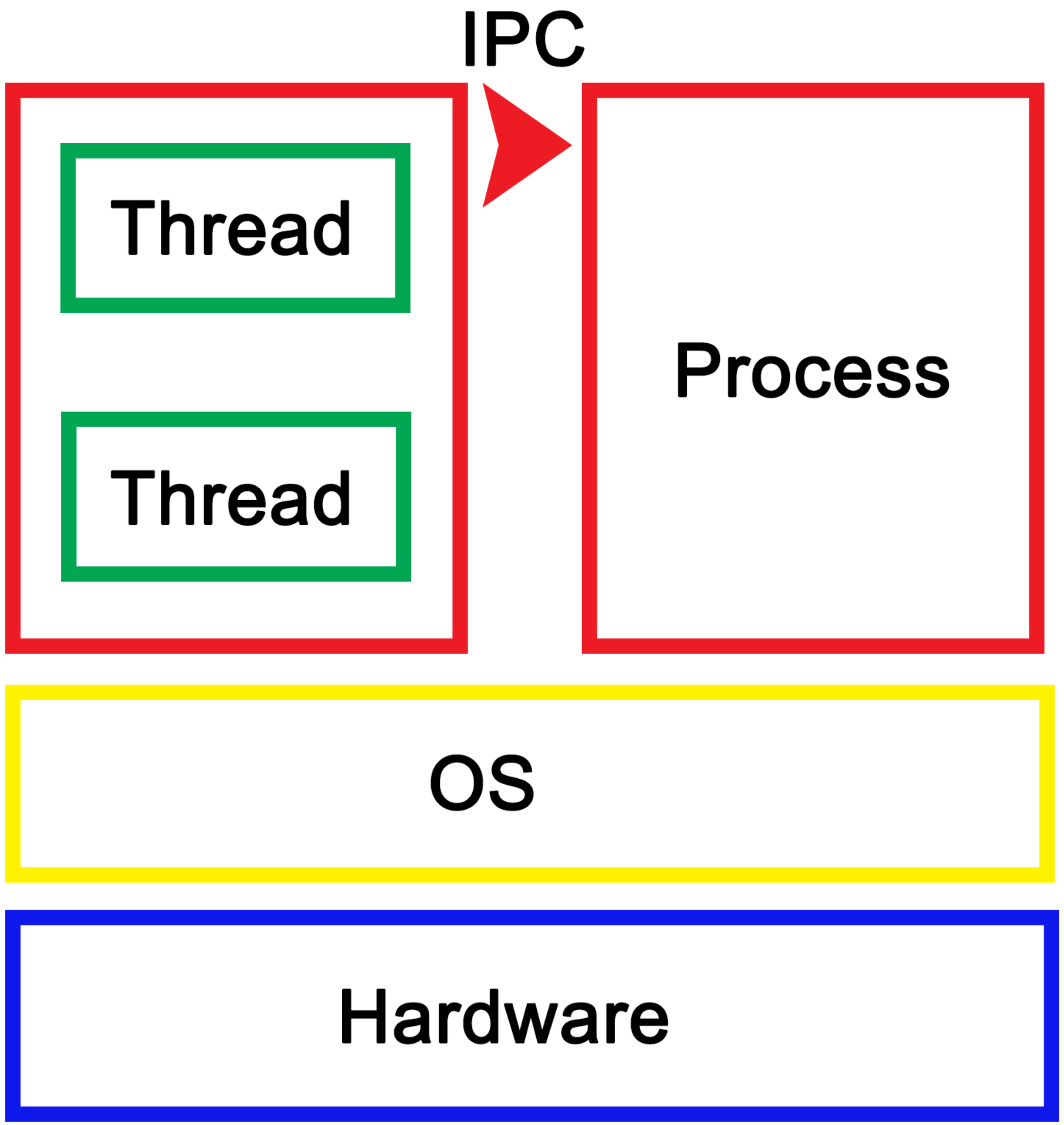
# PROCESS VS THREAD



| PROCESS | THREAD |
|---|---|
| An instance of a computer program that is being executed | A component of a process which is the smallest execution unit |

- A process includes one or more threads
- A thread includes:
  - Program counter (Instruction Pointer)
  - Register state
  - A stack

# PROCESS VS THREAD



# DEFINING MULTITHREADING

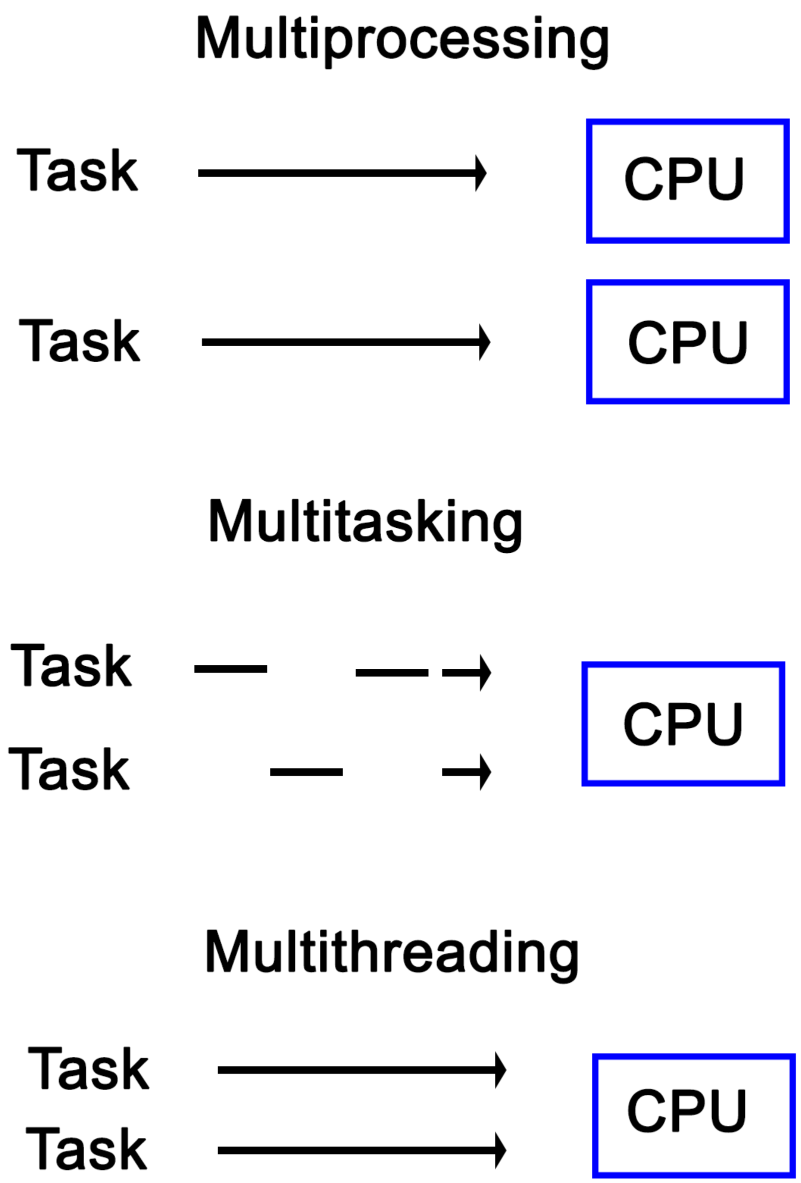
## Multiprocessing

One task per processor with multiple, physical processors

## Multithreading

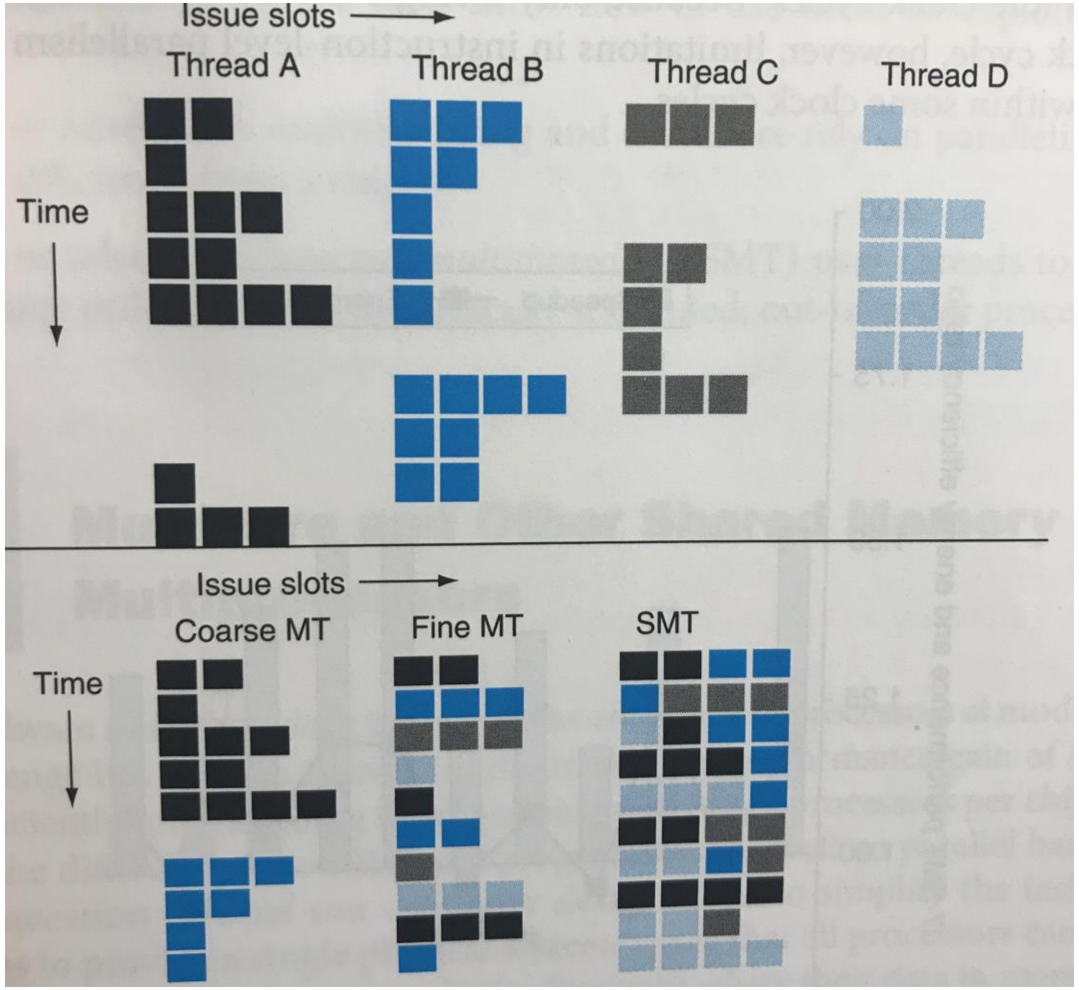
Multiple tasks on a singular processor, using multiple threads

# DEFINING MULTITHREADING



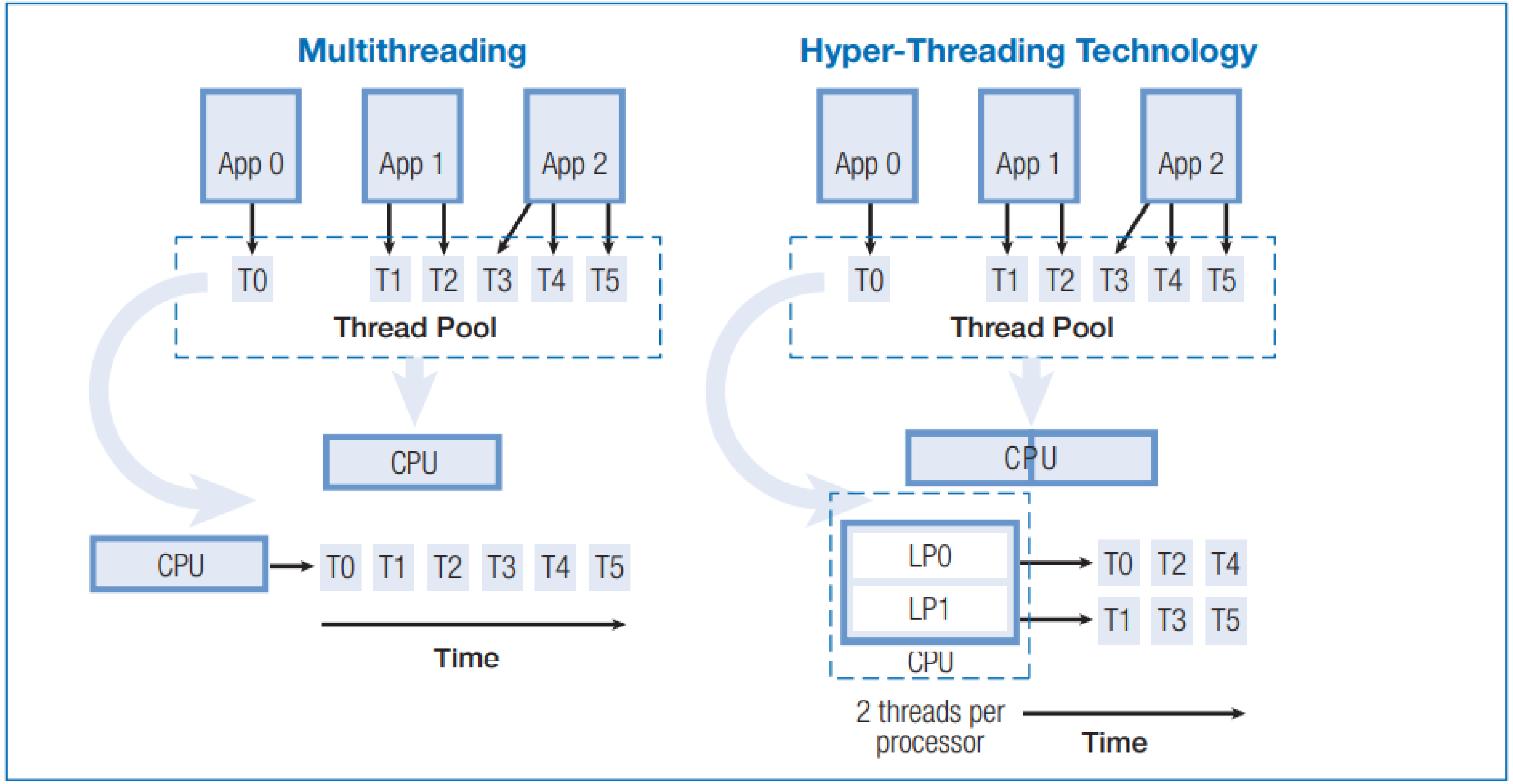
# MULTITHREADING

- Three different types
  - Fine-grained, course-grained, simultaneous
- Course-grained
  - Switch threads only on stall (cache miss)
- Fine-grained
  - Switch threads every clock cycle
- Simultaneous
  - Uses Idle threads to execute instructions



# LOGICAL PROCESSORS

- Instructions are dispatched for execution by processor core
- Processor core executes the two threads concurrently, using out-of-order instruction



# HYPERTHREADING

Uses dynamically scheduled pipelined processor

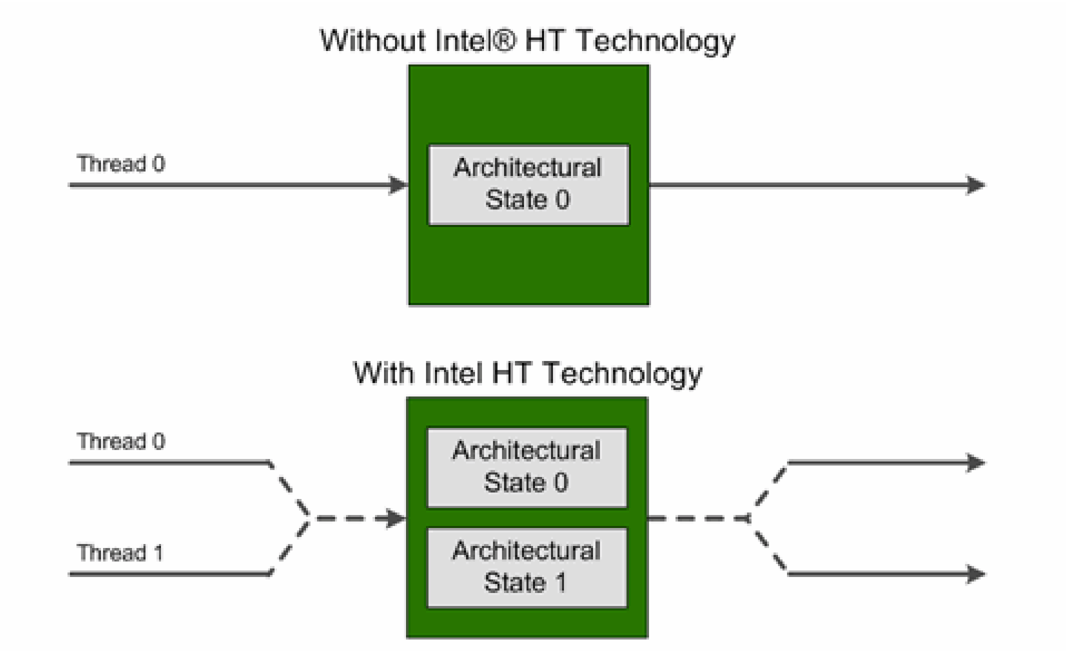
Out-of-order execution avoids hazards and stalls

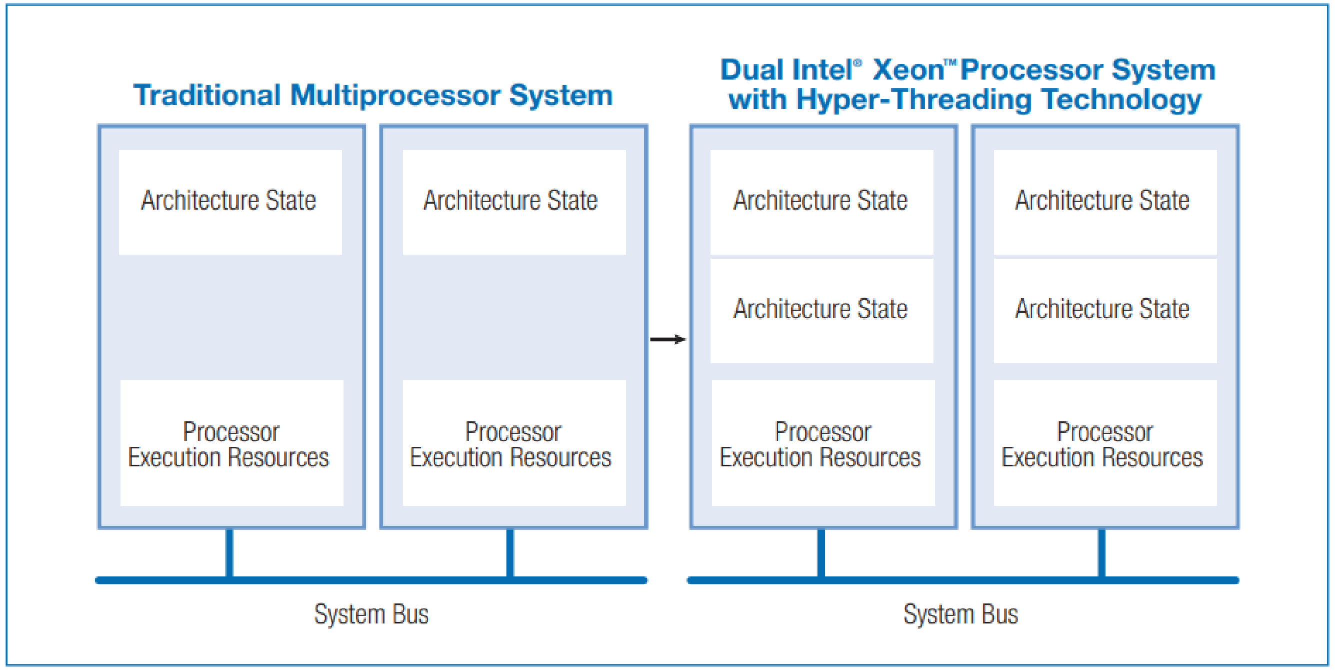
Utilizes both instruction and thread level parallelism

Processors often have more resources than one single thread can use

# HYPERTHREADING

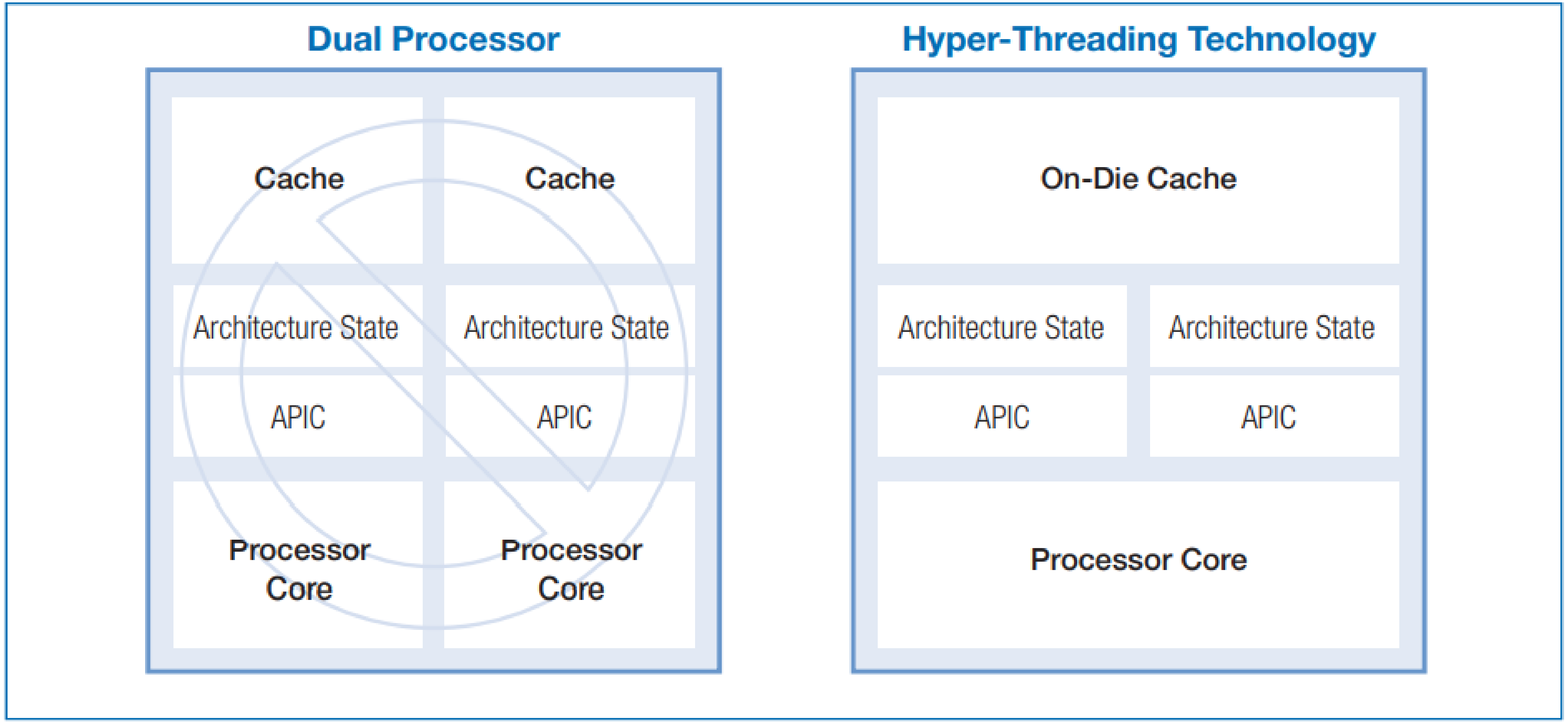
- CPU core "split" into two
  - Uses second set of registers
- Access two logical processors from one core
  - Process idle threads
  - Share resources



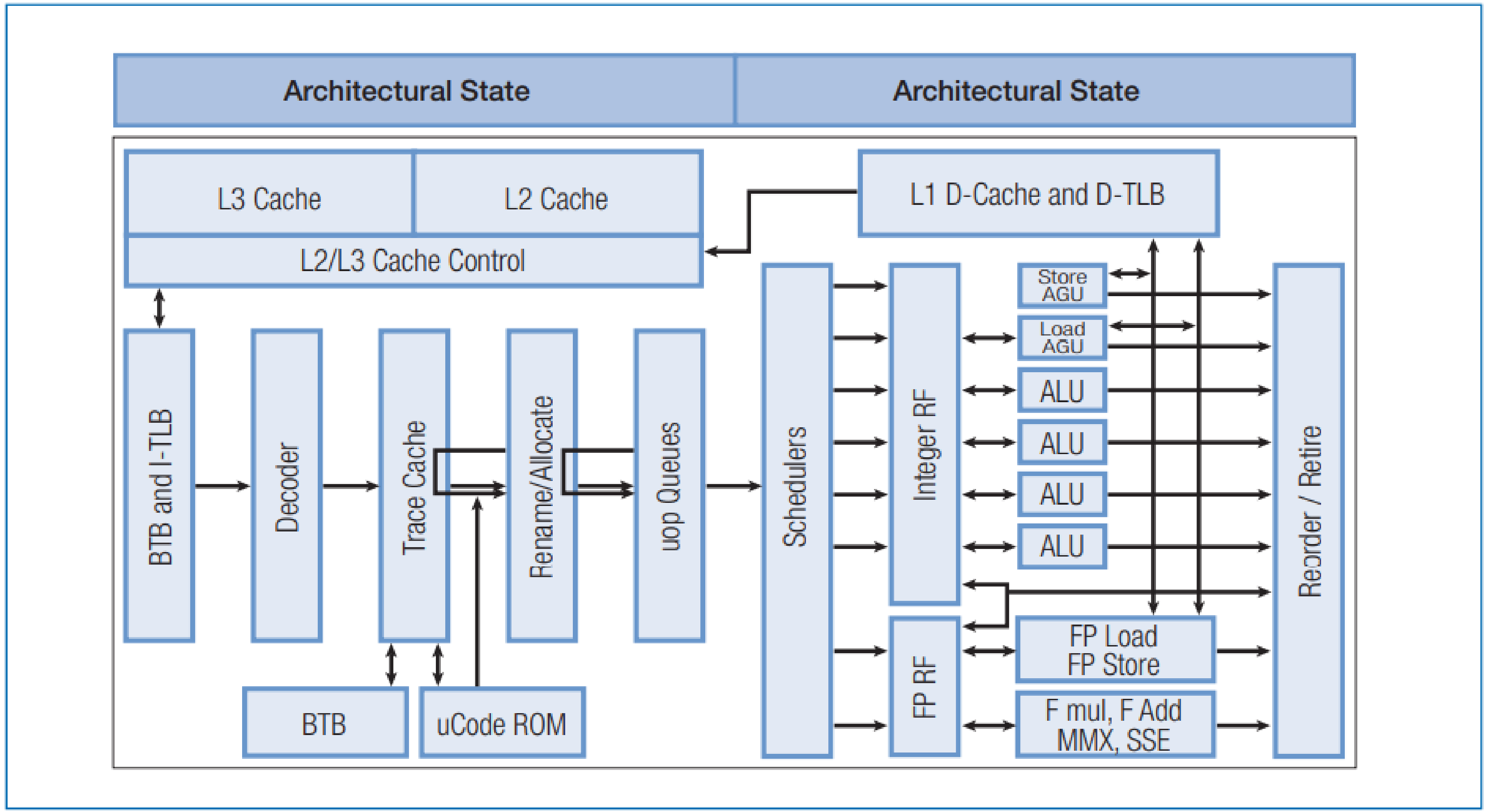

Traditional Multiprocessor System
Dual Intel® Xeon™ Processor System with Hyper-Threading Technology
Architecture State
Architecture State
Processor Execution Resources
Processor Execution Resources
Architecture State
Architecture State
Architecture State
Architecture State
Processor Execution Resources
Processor Execution Resources
System Bus
System Bus

- A processor with Hyper-Threading technology consists of two logical processors, each with its own architectural state
- Shared resources: Execution engine, caches, system-bus interface, firmware

# HYPERTHREADING



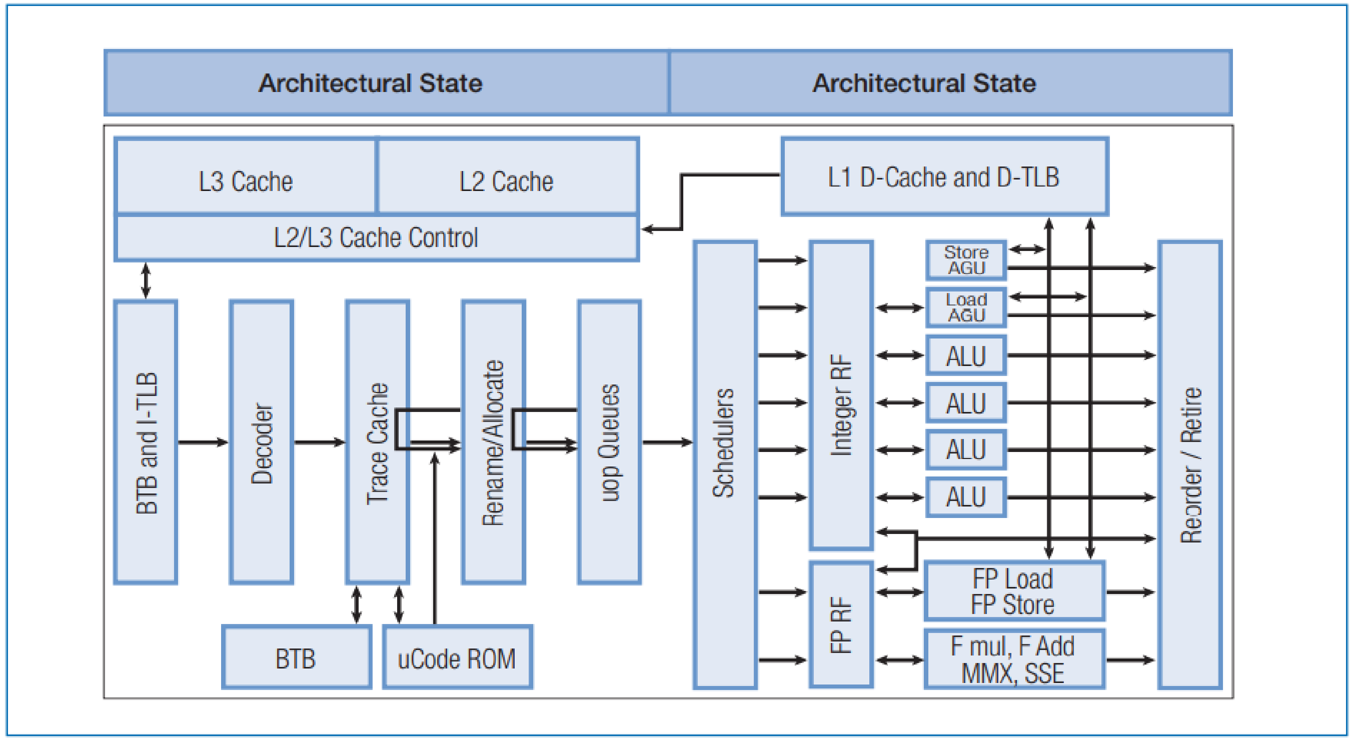
# LOGICAL PROCESSORS != PHYSICAL CORE



# SHARED RESOURCES

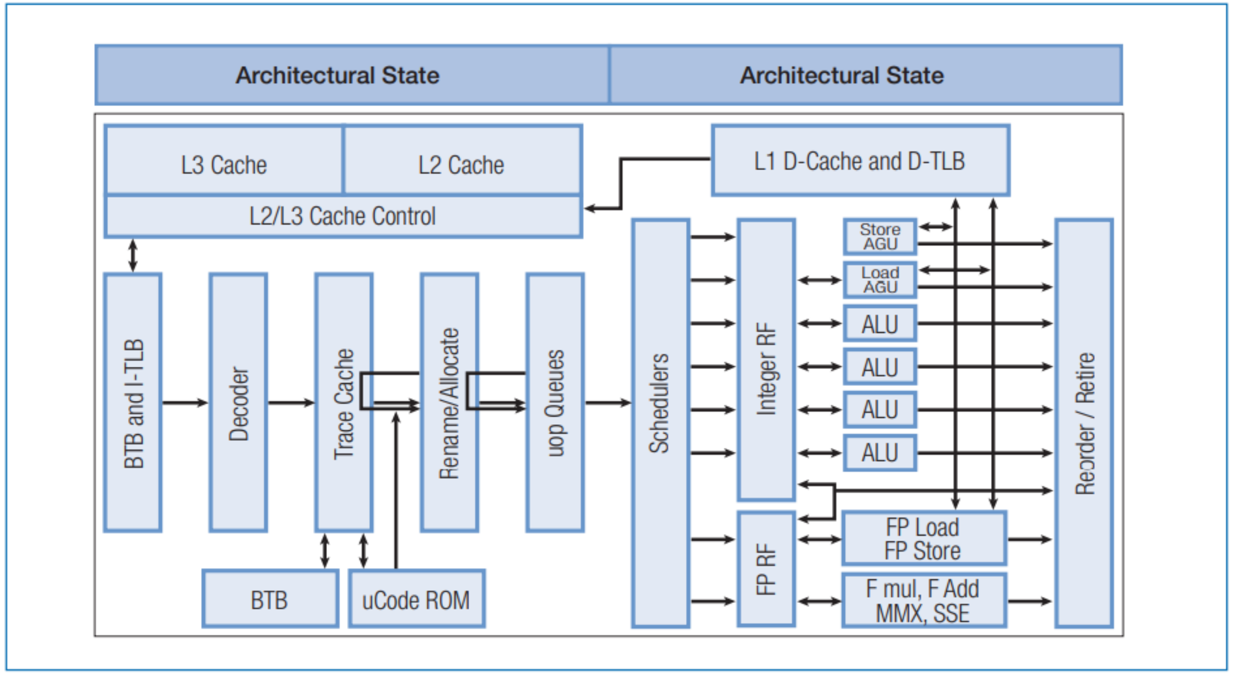
# SIMULTANEOUS INSTRUCTION EXECUTION

- Floating point arithmetic, integer arithmetic
- Bottleneck if both need floating point
- If one thread is performing lots of pointer arithmetic other threads cannot
    - Revert to time slices

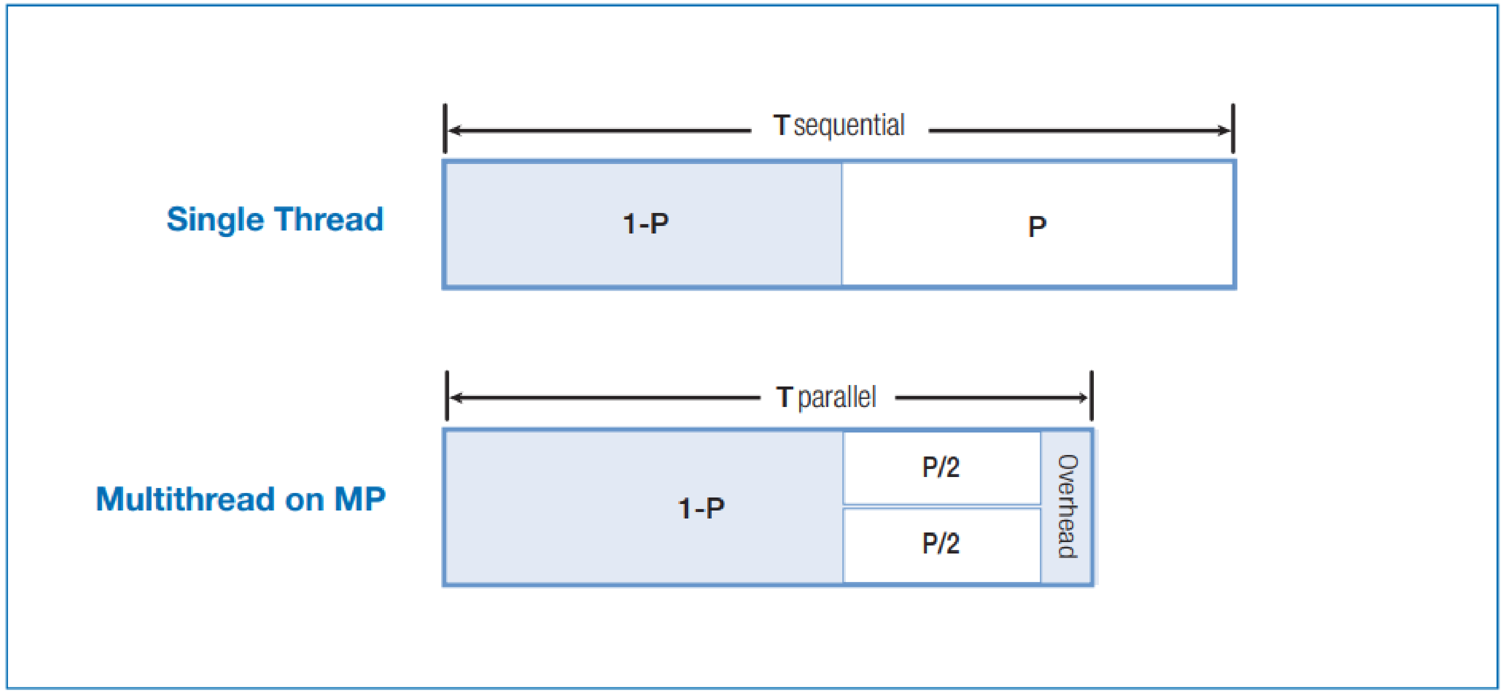


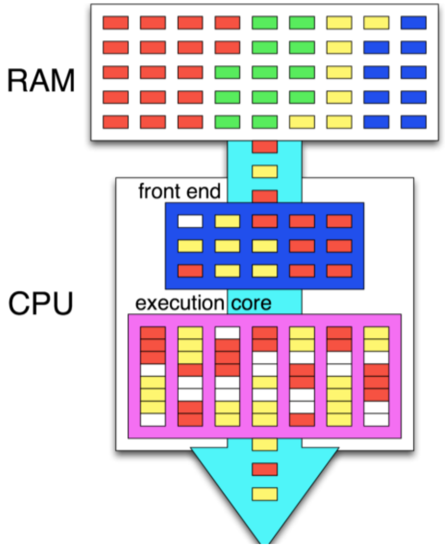
# SHARED CACHE

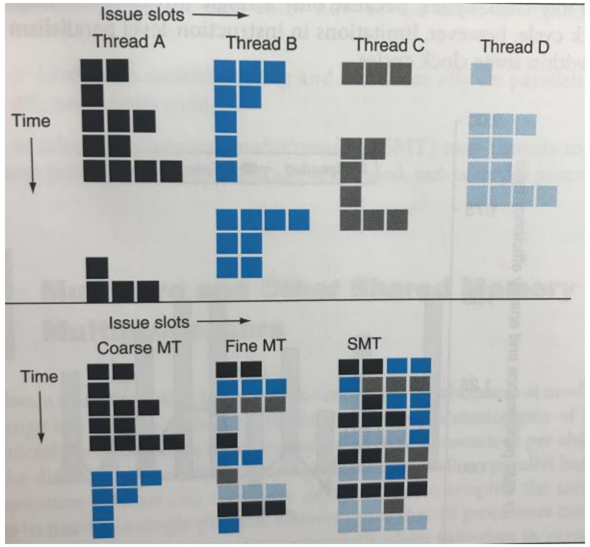
- Thread from processor 0 can access data cached by processor 1
- Cache can't tell the difference between one thread or another
- Locality – Threads can be accessing different areas of memory
- Cache is forced to fetch data for each thread – reducing performance



# OVERHEAD


T sequential
Single Thread
1-P
P
T parallel
Multithread on MP
1-P
P/2
P/2
Overhead





# STRENGTHS

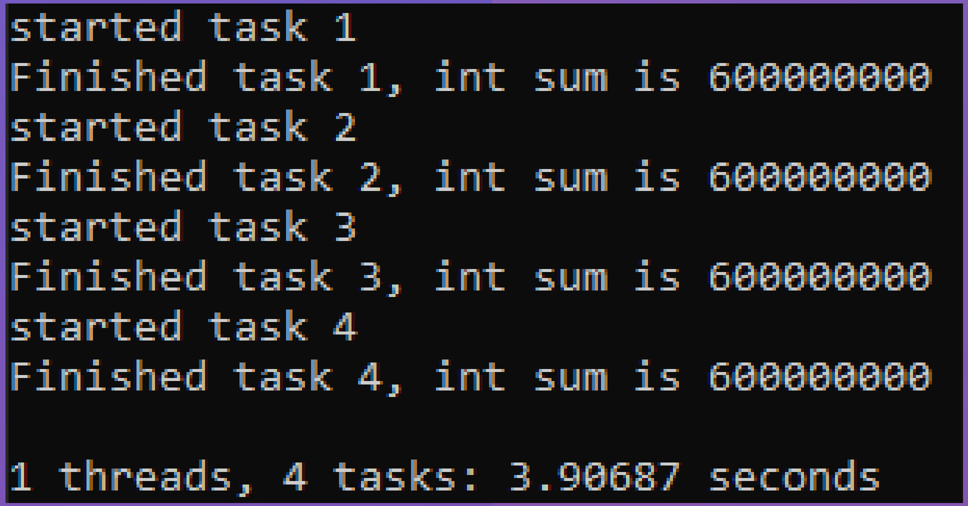
- Maximum flexibility in scheduling
- Use all available execution resources – keeping core busy

# CONS

- Potential to hinder performance
- Possibly less secure
    - Access shared cache
    - Jump from virtual core to virtual core
    - More susceptible to Spectre attacks

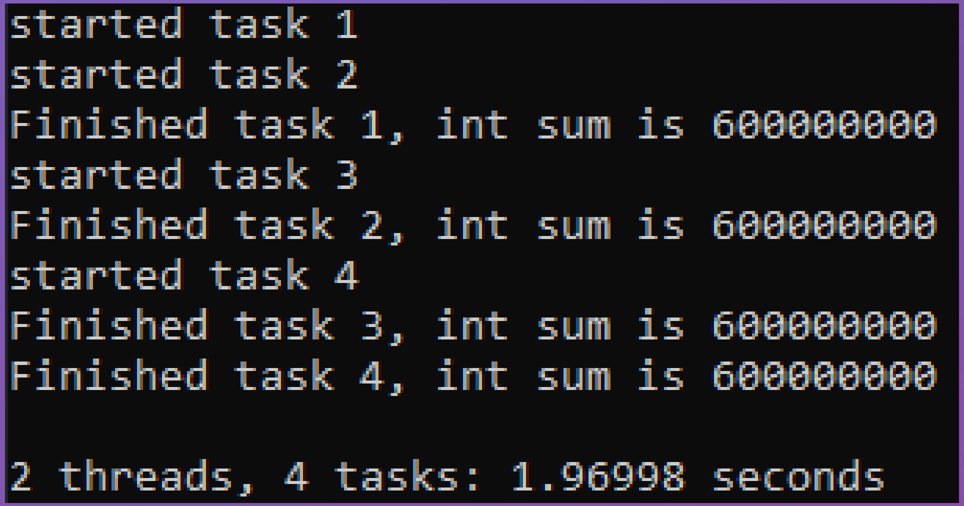
# BENCHMARKING MULTITHREADING

1 thread

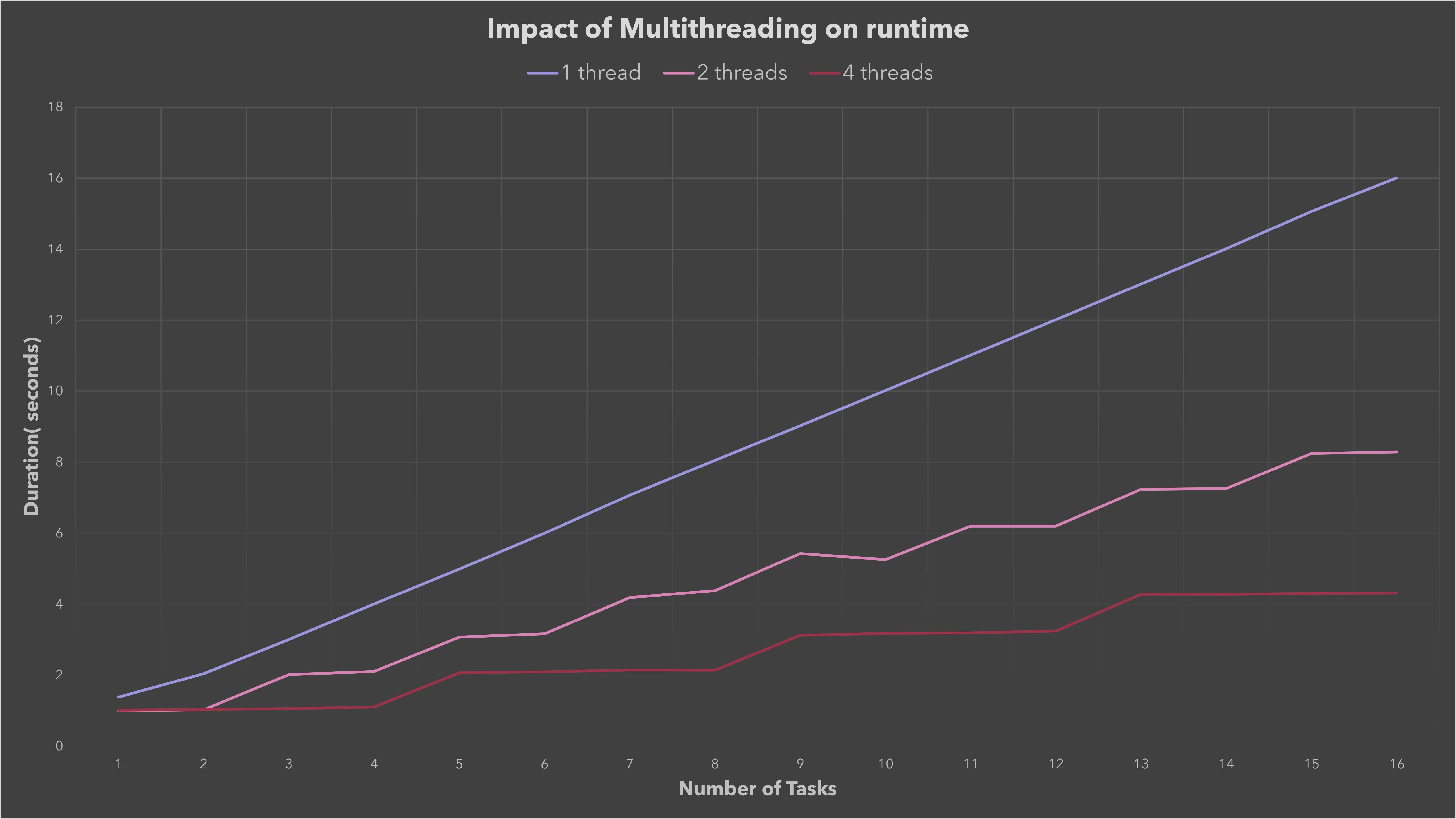


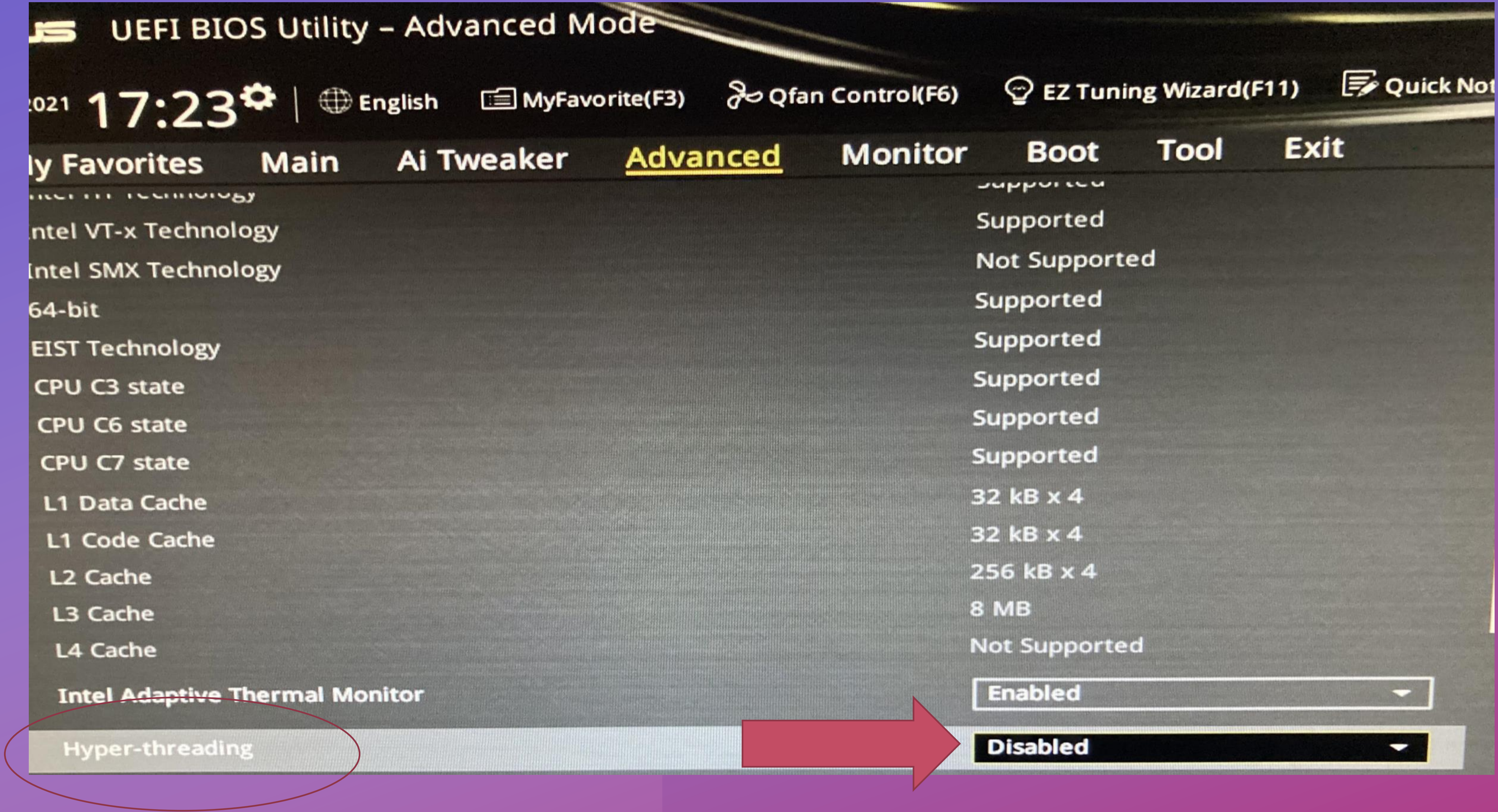
```
started task 1
Finished task 1, int sum is 600000000
started task 2
Finished task 2, int sum is 600000000
started task 3
Finished task 3, int sum is 600000000
started task 4
Finished task 4, int sum is 600000000

1 threads, 4 tasks: 3.90687 seconds
```

2 threads



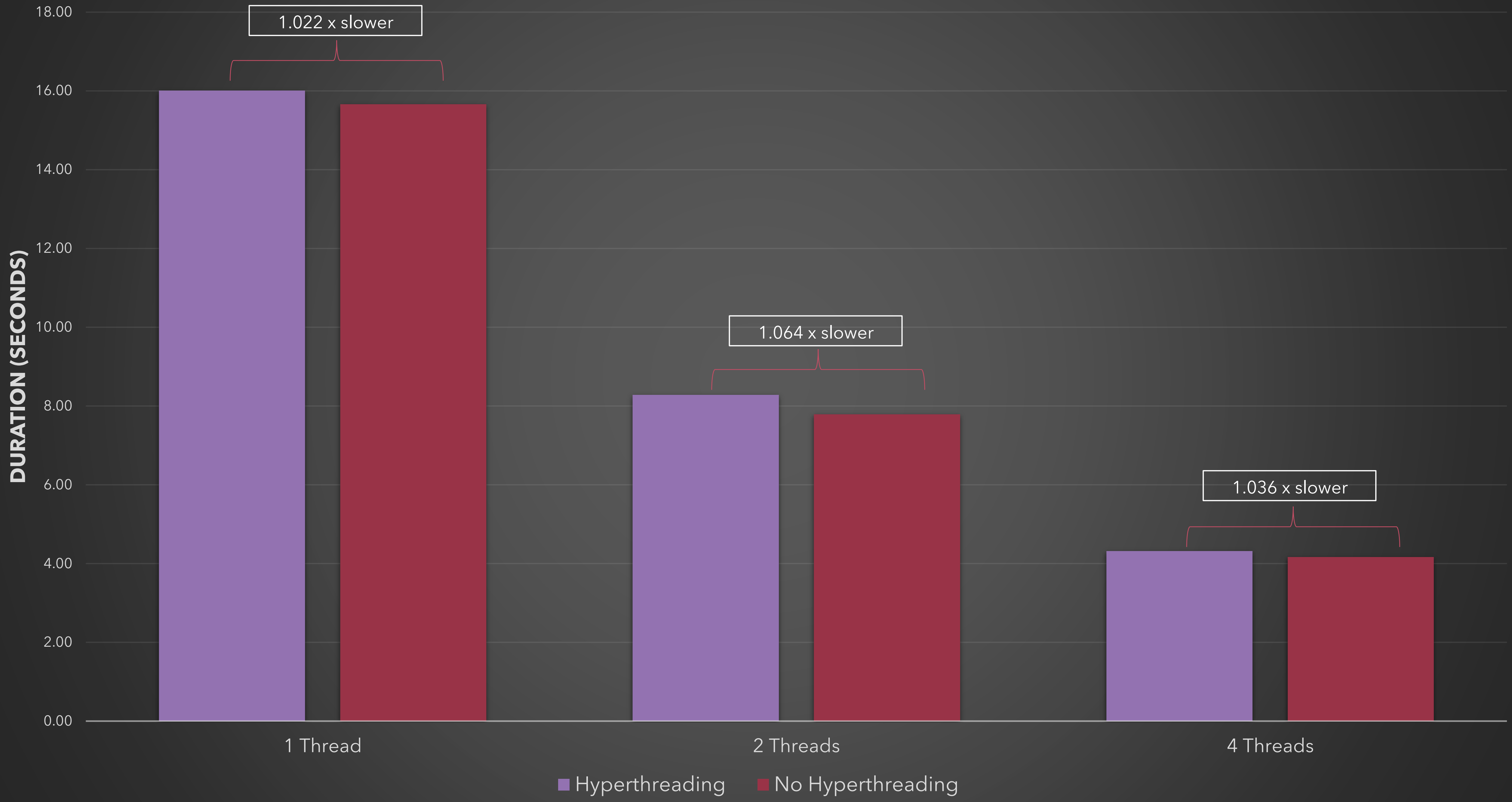
```
started task 1
started task 2
Finished task 1, int sum is 600000000
started task 3
Finished task 2, int sum is 600000000
started task 4
Finished task 3, int sum is 600000000
Finished task 4, int sum is 600000000

2 threads, 4 tasks: 1.96998 seconds
```


Impact of Multithreading on runtime
1 thread
2 threads
4 threads
Duration( seconds)
0
2
4
6
8
10
12
14
16
18
Number of Tasks
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

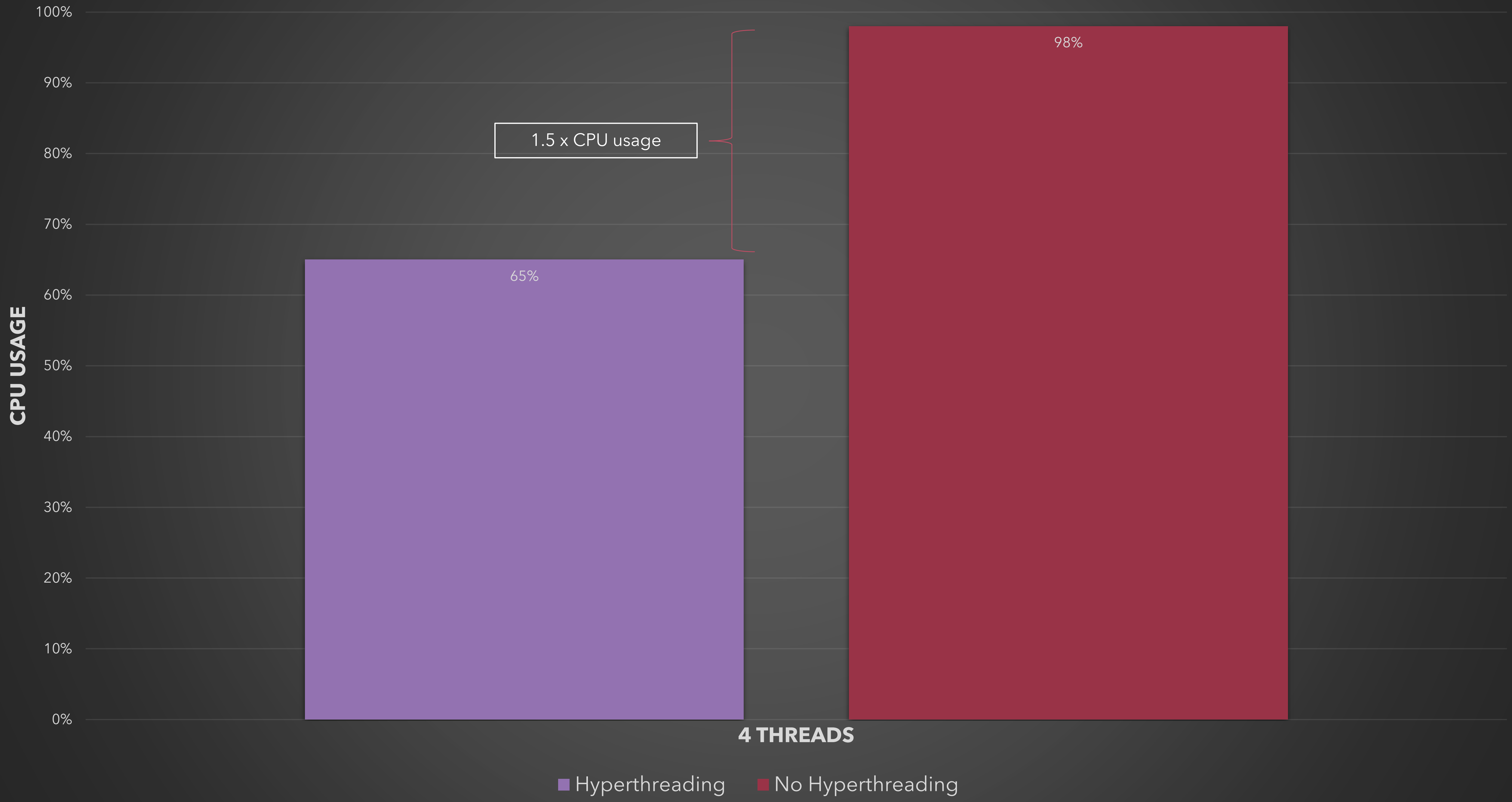


| | |
|---|---|
| ntel TXT Technology | Supported |
| ntel VT-x Technology | Supported |
| Intel SMX Technology | Not Supported |
| 64-bit | Supported |
| EIST Technology | Supported |
| CPU C3 state | Supported |
| CPU C6 state | Supported |
| CPU C7 state | Supported |
| L1 Data Cache | 32 kB x 4 |
| L1 Code Cache | 32 kB x 4 |
| L2 Cache | 256 kB x 4 |
| L3 Cache | 8 MB |
| L4 Cache | Not Supported |
| Intel Adaptive Thermal Monitor | Enabled |
| Hyper-threading | Disabled |

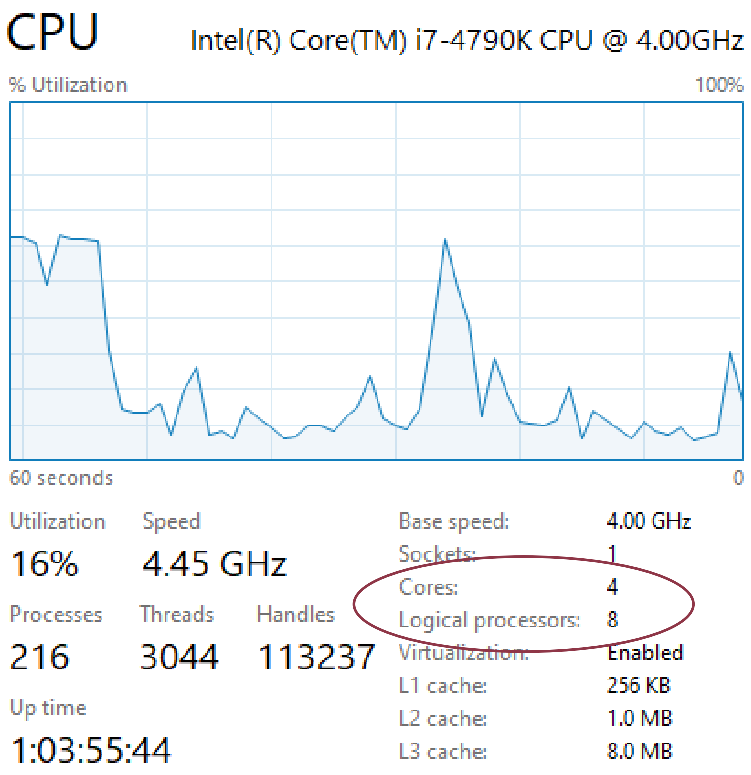
# Impact of Hyperthreading on Runtime



# Impact of Hyperthreading on CPU Usage



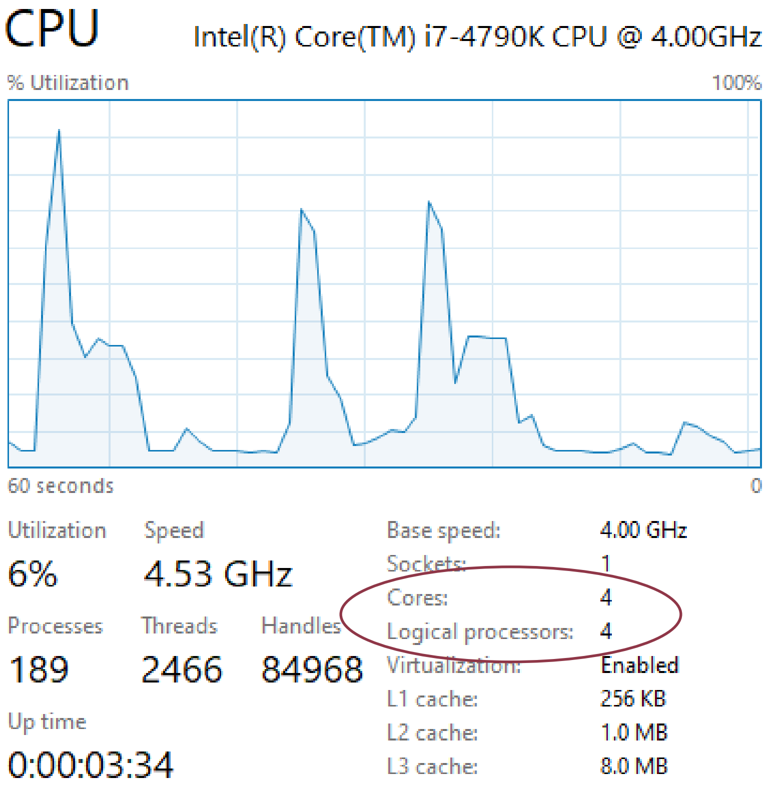
# LOGICAL PROCESSORS



CPU Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz

| Utilization | Speed | |
|---|---|---|
| 16% | 4.45 GHz | |
| Processes | Threads | Handles |
| 216 | 3044 | 113237 |
| Up time | | |
| 1:03:55:44 | | |

| | |
|---|---|
| Base speed: | 4.00 GHz |
| Sockets: | 1 |
| Cores: | 4 |
| Logical processors: | 8 |
| Virtualization: | Enabled |
| L1 cache: | 256 KB |
| L2 cache: | 1.0 MB |
| L3 cache: | 8.0 MB |

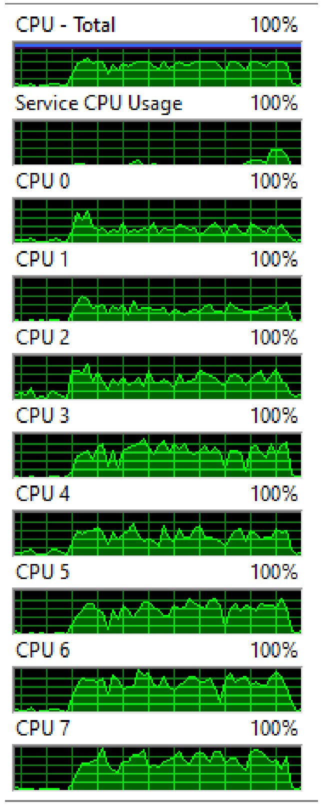


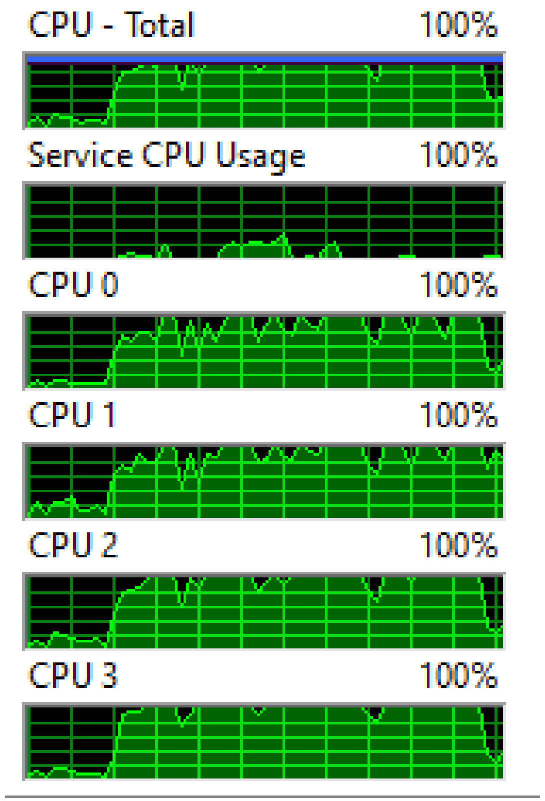
CPU Intel(R) Core(TM) i7-4790K CPU @ 4.00GHz

| Utilization | Speed | |
|---|---|---|
| 6% | 4.53 GHz | |
| Processes | Threads | Handles |
| 189 | 2466 | 84968 |
| Up time | | |
| 0:00:03:34 | | |

| | |
|---|---|
| Base speed: | 4.00 GHz |
| Sockets: | 1 |
| Cores: | 4 |
| Logical processors: | 4 |
| Virtualization: | Enabled |
| L1 cache: | 256 KB |
| L2 cache: | 1.0 MB |
| L3 cache: | 8.0 MB |

# CPU USAGE (DURING TEST)





# SOURCES

- Posch, Maya. *Mastering C++ Multithreading: Write Robust, Concurrent and Parallel Applications*. Packt, 2017.
- Patterson, David A, and John L. Hennessy. *COMPUTER ORGANIZATION AND DESIGN MIPS EDITION: the Hardware/Software Interface*. MORGAN KAUFMANN PUBLISHER, 2020.
- Stokes, Jon. "Introduction to Multithreading, Superthreading and Hyperthreading." *Arstechnica*, Oct. 2002, arstechnica.com/features/2002/10/hyperthreading/.
- *Intel Hyper-Threading Technology: Technical User's Guide*. Intel Corporation, 2003.
- Lithmee. "Difference Between Process and Thread." *Pediaa.Com*, Pediaa, 8 July 2018, pediaa.com/difference-between-process-and-thread/.
- Gigabyte GA-7PESH1 Visual Inspection, Board Features - Gigabyte GA-7PESH1 Review: A Dual Processor Motherboard through a Scientist's Eyes (anandtech.com) – (motherboard Image)
- Multicore processors terminology (microcontrollertips.com) – (multicore cpu image)

# GITHUB SOURCE CODE

- https://github.com/djperrone/Hyperthreading.git