Serial Performance Bottlenecks
08/19/2020
Serial Computer Architecture

EDVAC : First Binary Computer

Serial Computer Architecture

CPU

- Control Unit: Instructions executed
- Arithmetic Logic Unit: Calculations on data

Memory: Instructions and data stored as bits

Input / Output: Communication with users
Serial Computer Architecture

CPU

Instructions executed

Calculations on data

Von Neumann bottleneck
Control and arithmetic units must continuously be fed instructions and data from memory

Memory

Instructions and data stored as bits

Input / Output

Communication with users
One Fix: Cache

- Main memory:
  - Large
  - Far from CPU
  - Slow

- Cache:
  - Smaller
  - Closer to CPU
  - Faster
Cache-Based Microprocessor
Cache-Based Microprocessor

Three little components perform all operations

Main memory

L2 unified cache

L1 data cache

Memory interface

L1 instr. cache

INT/FP queue

INT reg file

FP reg file

shift mask

INT op

LD

ST

FP mult

FP add

Floating point multiplication

Floating point addition

Operations on integers

Everything else used to efficiently move data and instructions
Cache-Based Microprocessor

Data and instructions start in main memory.

Instructions are moved into the queues to be executed.

Data is moved into registers to be operated upon.

Registers directly access “work” components (previous slide).

Main memory

L2 unified cache

Memory interface

L1 data cache

L1 instr. cache

INT/FP queue

Memory queue

INT reg file

FP reg file

shift mask

INT op

LD

ST

FP mult

FP add

INT op

LD

ST

FP mult

FP add
Cache-Based Microprocessor

1.) After data is used moved to L1 data cache
2.) Once an L1 cache is full some data/instr are moved to L2 cache
3.) All other data and instr. stored in main memory

After instructions are executed moved to L1 instr. cache
Multicore processors have multiple cores that share main memory, and maybe a level of cache.
Multicore Processor

- Cores may have own cache
- May share some levels of cache
- All cores share main memory

**Shared Memory Programming**
Will come back to this next week
Superscalar Processor

- Capable of executing more than one instruction per cycle
- Typical quality of today’s computers

Some details:

- Multiple instructions fetched, decoded concurrently
- Fast caches: > 1 load or store per cycle
- Multiple floating-point pipelines run in parallel (explanation to follow)
Pipelining: Instruction-level parallelism

- Parallelism occurs in serial computers

- To complete an instruction:
  - Fetch data from memory
  - Decode instruction
  - Execute operation

- A program will have many, many operations.

- Can pipeline the fetch - decode - execute process
Pipelining

- Similar to assembly lines
- Workers only need to know about their specific task
- Each worker executes his/her task over and over on successive objects
- Each object is then moved to the next worker
If you want to learn more...

- “Introduction to High Performance Computing for Scientists and Engineers” by Georg Hager and Gerhard Wellein

- We will talk about caches more in depth next lecture