Parallel Processors and Programs

Concurrency and Parallelism
Can you spot the error?
A Mips processor running two different tasks every 5ms and 20 ms are concurrent programs running serially.

Flynn's taxonomy

Concurrency and Parallelism
Can you spot the error?
A Mips processor running two tasks every 5ms and 20 ms is a sequential program running in parallel.

Concurrency and Parallelism
- Software is concurrent/sequential
- Hardware is serial/parallel
Concurrent programs can be run in serial, concurrent, sequential programs can be run in parallel.

Amdahl's law
Remember: for true speedup, T(ALL) should be the running time for the sequential code.
Parallel Processors and Programs

Concurrent and Parallelism

Can you spot the errors?
A MIPS processor running two different tasks every 5ms and 20 ms are concurrent programs running serially.

Concurrent and Parallelism

Can you spot the errors?
A MIPS processor running two tasks every 5ms and 20 ms is a sequential program running in parallel.

Flynn's taxonomy

- Single Instruction Single Data (SISD)
  - Simple instruction, single data
  - Examples: CPU instructions
- Single Instruction Multiple Data (SIMD)
  - Single instruction, multiple data
  - Examples: SIMD instructions
- Multiple Instruction Single Data (MISD)
  - Multiple instructions, single data
  - Examples: Instruction and memory addressing
- Multiple Instruction Multiple Data (MIMD)
  - Multiple instructions, multiple data
  - Examples: Multiple instruction sets

Concurrent and Parallelism

- Software is concurrent/sequential
- Hardware is serial/parallel

Concurrent programs can be run in serial hardware.
Sequential programs can be run in parallel.

Measuring speedup

Comparing a sequential and parallel version of the same algorithm/program
- All benchmarking should be done on the same machine
- Use the same compiler version / options (if possible)
- Disable all hardware parallelism (multiple cores, hyperthreading) and add them one by one for each test.

Amdahl's law

\[ \text{Speedup} = \frac{T_{	ext{serial}}}{T_{	ext{parallel}}} = \frac{T_{	ext{serial}}}{T_{	ext{serial}} - T_{\text{improved}}} \]

Remember: for true speedup, T before should be the running time for the sequential code!
Flynn's taxonomy

**SISD**
*Single Instruction, Single Data*
- Example: MIPS uniprocessor

**SIMD**
*Single Instruction, Multiple Data*
Data parallelism - one operation on multiple pieces of data
- Example:
  - Intel AVX extension
  - GPU:s

**MISD**
*Multiple Instruction, Single Data*
Very rare - No examples today

**MIMD**
*Multiple Instruction, Multiple Data*
- Example:
  - Intel i7
  - Exynos 5 Octa (ARM)
  - GPU:s

Concurrency and Parallelism
**Measuring speedup**

Comparing a sequential and parallel version of the same algorithm/program

- All benchmarking should be done on the same machine
- Be aware of true speedup vs relative speedup!
- Disable all hardware parallelism (multiple cores, hyperthreading) and add them one by one for each test.
Amdahl's law

\[
\text{Speedup} = \frac{T_{\text{before}}}{T_{\text{after}}} = \frac{T_{\text{before}}}{\frac{T_{\text{affected}}}{N} + T_{\text{unaffected}}}
\]

Remember - for true speedup, \( T_{\text{before}} \) should be the running time for the sequential code!
One issue, 5-stage pipelined MIPS, conditional branch (beq) computed in the decode stage assuming branch-not-taken.

- What is the IPC when N = 5?

- How could we optimize the code for a two-issue MIPS processor?

- What is the IPC for the two-issue processor when N = 5?

- What speedup do we achieve by using a two-issue processor?
Parallel Processors and Programs

Concurrent and Parallelism

A Mips processor running two different tasks every 5ms and 20 ms are concurrent programs running serially.

Concurrent and Parallelism

A Mips processor running two tasks every 5ms and 20 ms is a sequential program running in parallel.

Flynn's taxonomy

- SIMD: Single Instruction Single Data
  - Example: MIPS processor

- MISD: Multiple Instruction Single Data
  - Example: CMOS processor

- MIMD: Multiple Instruction Multiple Data
  - Example: Memory-mapped processors

Concurrent programs can be run in parallel, sequential programs can be run in parallel.

Amdahl's law

\[
\text{Speedup} = \frac{T_{	ext{serial}}}{T_{	ext{parallel}}} = \frac{1}{(1 - p) + \frac{p}{N}}
\]

where
- \(T_{	ext{serial}}\) is the time required for the serial code
- \(T_{	ext{parallel}}\) is the time required for the parallel code
- \(p\) is the percentage of the code that can be parallelized
- \(N\) is the number of processors

Remember - for true speedup, \(T_{	ext{before}}\) should be the running time for the sequential code!

Measuring speedup

Comparing a sequential and parallel version of the same algorithm/program

- All benchmarking should be done on the same machine
- The amount of true speedup vs relative speedup
- Disable all hardware parallelism (multiple cores, hyperthreading) and add them one by one for each test.
Concurrency and Parallelism

- **Software** is concurrent/sequential
- **Hardware** is serial/parallel

Concurrent programs can be run in serial hardware, Sequential programs can be run in parallel
Concurrent and Parallelism

Can you spot the errors?

A MIPS processor running two tasks every 5ms and 20 ms is a sequential program running in parallel.
Concurrency and Parallelism

Can you spot the errors?

A MIPS processor running two different tasks every 5 ms and 20 ms are concurrent programs running serially.