Performance

How do we measure it?

Joseph Kehoe[1]

[1]Department of Computing and Networking
Institute of Technology Carlow

CDD101, 2017

- Latency
  - Total time it takes to compute a single result
  - Measured in units of time
- Throughput
  - The rate at which a series of results can be computed
  - Measured in units of work per unit time
- Power Consumption
  - The amount of power required to performa computation
  - Measured in Work per power unit

Compares latency in solving an indentical problem on one hardware unit versus P hardware units

$Sp = \frac{T1}{Tp}$

Where S is Speed and T is time

ABSOLUTE SPEEDUP Sequential Algorithm Time used for T1

RELATIVE SPEEDUP Parallel algorithm used for T1

Efficiency

- Speedup divided by number of workers
- $\frac{Sp}{P}$
- $\frac{T1}{P \times Tp}$
- Ideal efficiency is 1
  - aka 100% efficiency

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

- Parallel Algorithm makes better use of cache
- Parallel Algorithm is simply a better algorithm
- Parallel Algorithm with multiple threads uses cache better than single parallel algorithm execution

Amdahl's Law

Work is of two types:

- Serial Work that cannot be parallelised
- Parallel Work that can be done in parallel
- $T_1 = Work_{serial} + Work_{parallel}$
- $T_p >= Work_{serial} + \frac{Work_{parallel}}{P}$

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE *of* TECHNOLOGY CARLOW
At the Heart of South Leinster

If f is the fraction of the total work that is serial then (1-f) is the fraction that is parallel

- $Work_{serial} = f \times T_1$
- $Work_{parallel} = (1 - f) \times T_1$
- $S_p \leq \frac{1}{[f + \frac{(1-f)}{p}]}$
- As $P \rightarrow \infty$ then $S_\infty \leq \frac{1}{f}$

Institiúid Teicneolaíochta Cheatharlach
INSTITUTE of TECHNOLOGY CARLOW
At the Heart of South Leinster

Gustafson-Barsis' Law

- Amdahl's Law viewed program size as fixed and the processor count as variable
- But program size can increase as processor count increases
- Sometimes the serial part remains constant as the program size increases
- Thus allowing for greater speedup

Work-Span Model

- Tasks form an acyclic graph
  - Ignores communication and memory access costs
  - Assumes task scheduling is greedy
- T_1: time required for serial version of algorithm to run
  - Known as Work
- T_∞: time required for algorithm to run on ideal computer with infinite processors
  - Known as Span

Work-Span Model

- Span is equivalent to the length of the critical path
  - Aka depth, step complexity
- Superlinear speedup is impossible in the Work-Span Model
  - $S_p = \frac{T_1}{T_p}$
  - $\frac{T_1}{T_p} \leq \frac{T_1}{(\frac{T_1}{P})} = P$
  - Therefore $S_p \leq P$

Work-Span Model

- Adding new processors never slows down an algorithm
  - Assuming greedy scheduling
  - $S_p = \frac{T_1}{T_p} \leq \frac{T_1}{T_\infty}$
- Brent's Lemma
  - $T_p \leq \frac{T_1 - T_\infty}{P} + T_\infty if T_\infty << T_1$
- When working on parallelism focus on the span
- Increase work only if it decreases the span