Speedup, scalability, and more OpenMP

Troels Henriksen

Inspired by slides by Randal E. Bryant and David R. O'Hallaron.

Scalability

Speedup

Definition (Speedup in latency)

If T_1, T_2 are the runtimes of two programs P_1, P_2, then the *speedup in latency* of P_2 over P_1 is

$$\frac{T_1}{T_2}$$

Definition (Speedup in throughput)

If Q_1, Q_2 are the throughputs of two programs P_1, P_2, then the *speedup in throughput* of P_2 over P_1 is

$$\frac{Q_2}{Q_1}$$

Scalability

Definition (Strong scaling)

How the runtime varies with the number of processors for a fixed problem size.

Definition (Weak scaling)

How the runtime varies with the number of processors for a fixed problem size *relative to the number of processors*.

Definition (Amdahl's Law)

If p is the proportion of execution time that benefits from parallelisation, then $S(N)$ is maximum theoretical speedup achievable by execution on N threads, and is given by

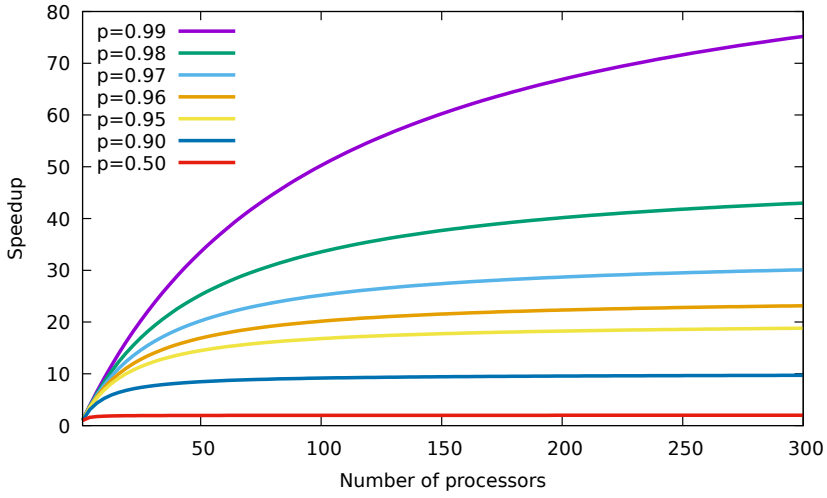
$$S(N) = \frac{1}{(1-p) + \frac{p}{N}}$$

Note that

$$S(N) \leq \frac{1}{1-p}$$

- Potential speedup by optimising part of system is bounded by proportion of part in overall runtime.
- **We should optimise the parts that take a long while to run.**
- Predicts *strong scaling*.

Amdahl's law is pessimistic



Gene Amdahl

By Pkivolowitz at English Wikipedia, CC BY 3.0,
https://commons.wikimedia.org/w/index.php?curid=4041015

Gustafson's Law

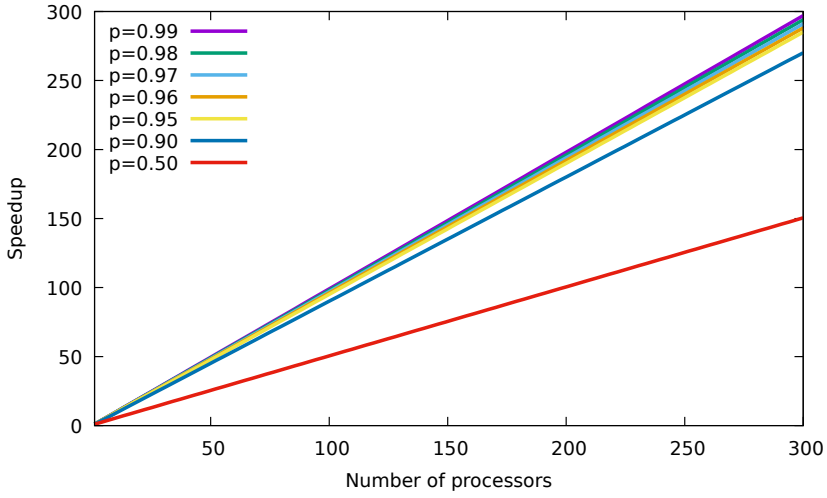
Definition (Gustafson's Law)

If $s = 1 - p$ is the proportion of execution time that must be sequential, then $S(N)$ is maximum theoretical speedup achievable by execution on N threads, and is given by

$$S(N) = N + (1 - N) \times s$$

- Predicts *weak scaling*.

Gustafson's law is optimistic



John L. Gustafson

By Davisourus at English Wikipedia, CC BY-SA 3.0,
https://commons.wikimedia.org/w/index.php?curid=67553743

Summary

- Compare performance of programs by computing *speedup*.
- For parallel programs, *scalability* is the ability to take advantage of greater machine resources.
- Use Amdahl's Law to predict the potential of parallelisation of a program on a *fixed* problem.
- Use Gustafson's Law to predict the potential of parallelisation of a program on a *growing* problem.