

2.1: The Amdahl's law

Introduction

The section introduces the learners to Amdahl's law; this law introduces the concept of how to find the maximum expected improvement to an overall system when only part of the system is improved.

Activity Details

Amdahl's law is also known as Amdahl's argument. It is used to find the maximum expected improvement to an overall system when only part of the system is improved. It is often used in parallel computing to predict the theoretical maximum speed up using multiple processors.

The law is named after computer architect Gene Amdahl.

Amdahl's law: is law used to find the maximum expected improvement to an overall system when only part of the system is improved. It is often used in parallel computing to predict the theoretical maximum speed up using multiple processors.

The speedup of a program using multiple processors in parallel computing is limited by the time needed for the sequential fraction of the program. E.g. if a program needs 20 hours using a single processor core, and a particular portion of the program which takes one hour to execute cannot be parallelized, while the remaining 19 hours (95%) of execution time can be parallelized, then regardless of how many processors are devoted to a parallelized execution of this program, the minimum execution time cannot be less than that critical one hour. Hence, the theoretical speedup is limited to at most 20X.

Equation

A task which can be parallelized can be split up into two parts:

- A non-parallelizable part (or serial part) which cannot be parallelized;
- A parallelizable part which can be parallelized.

Example 2.1.1

A program that processes files from disk, A small part of that program may scan the directory and create a list of files internally in memory. After that, each file is passed to a separate thread for processing. The part that scans the directory and creates the file list cannot be parallelized, but processing the files can.

The time taken to execute the whole task in serial (not in parallel) is denoted T . The time T includes the time of both the non-parallelizable and parallelizable parts. The portion of time to execute in serial the parallelizable part is denoted P . The portion of time to execute in serial the non-parallelizable part is then $1 - P$.

From this follows that : $T = (1 - P)T + PT$

It is the parallelizable part P that can be sped up by executing it in parallel. How much it can be sped up depends on how many subtasks are executed in parallel. The theoretical execution time of the parallelizable part on a system capable of executing N subtasks in parallel.

Amdahl's law gives the theoretical execution time $T(N)$ of the whole task on a system capable of executing N subtasks in parallel:

Consequently, the best (with an infinite number of subtasks) theoretical execution time of the whole task is

In terms of theoretical overall speedup, Amdahl's law is given as

and the best theoretical overall speedup is

As an example, if P is 90%, then $1 - P$ is 10%, and the task can be sped up by a maximum of a factor of 10, no matter how large the value of N used. For this reason, parallel computing is only useful for small numbers of processors and problems with very high values of P (close to 100%): so-called embarrassingly parallel problems. A great part of the craft of parallel programming consists of attempting to reduce the component $1 - P$ to the smallest possible value.

In parallel computing, P can be estimated by using the measured speedup $S(N)$ on a specific number of processors N using

P estimated in this way can then be used in Amdahl's law to predict speedup for a different number of processors.

Conclusion

This unit introduced the learner to the Amdahl's law. Examples were used to teach the learner how to find the maximum expected improvement to an overall system when only part of the system is improved.

1. Briefly describe Amdahl's law on parallel computing

In computer architecture, Amdahl's law (or Amdahl's argument) gives the theoretical speedup in latency of the execution of a task at fixed workload that can be expected of a system whose resources are improved. It is named after computer scientist Gene Amdahl, and was presented at the AFIPS Spring Joint Computer Conference in 1967.

Amdahl's law can be formulated the following way:

where

- S is the theoretical speedup in latency of the execution of the whole task;
- s is the speedup in latency of the execution of the part of the task that benefits from the improvement of the resources of the system;
- p is the percentage of the execution time of the whole task concerning the part that benefits from the improvement of the resources of the system before the improvement.

This page titled [2.1: The Amdahl's law](#) is shared under a [CC BY-SA](#) license and was authored, remixed, and/or curated by [Harrison Njoroge \(African Virtual University\)](#).