Proposal for a C++ module to facilitate Distributed Computing of Parallelisable Tasks

Ashley Flynn – Academy of Interactive Entertainment – 2018
INFT567 - Complex System Project Brief

Overview

A distributed system is a model in which networked computers communicate and coordinate their actions in order to achieve a common goal.

This has many practical applications in the games industry such as providing a single seamless MMO world managed by multiple servers, render farms generating 3D graphic assets or processing large scale neural networks for advanced machine learning, and so on.

The proposed module will be a .DLL and .LIB, along with header files, which will facilitate the distributed computing of parallelisable tasks. The module will allow any parallelisable task to be broken down into discreet work packages which can then be shared among multiple computers on a network (known as nodes).

The module will be able to operate as either a host or client node, with a single host managing the distribution of work packages to clients. The host will collate the data resulting from the completed work, which can then be used by a program using the module.

A host will optionally act as its own client, performing a share of the work that it divides among its connected client nodes.

Similar Systems

Similar systems include Apache Hadoop https://en.wikipedia.org/wiki/Apache_Hadoop, HTCondor https://en.wikipedia.org/wiki/HTCondor, the Open Source Cluster Applications Resources https://www.csm.ornl.gov/oscar/ and Beowulf Clusters https://en.wikipedia.org/wiki/Beowulf_cluster

Dependencies

The module will be written in C++14 to take advantage of its advanced multi-threading capabilities to gain the maximum performance from each client node.

The networking component of SFML https://www.sfml-dev.org/ will provide a robust and well supported communications core.

Compression of data sent over the network will be provided by Zlib https://zlib.net/

Examples, Test and Benchmarking

An automated test and benchmark suite will perform empirical testing of the module and a connected cluster using a standardised set of test data.

An example of practical use of the module will take the form of a Mandelbrot fractal calculator, able to visualise and zoom into fractals in real time. The Mandelbrot set is mathematically intensive to compute and provides a large set of data for the cluster to work on, with the results visualised in an interesting and interactive way.