

Samuel J. Sully

**Voxel Populi:
A Decentralised
Peer-to-Peer Voxel-Based
World**

Computer Science Tripos

Robinson College

2019-20

Proforma

Name:	Samuel John Sully
College:	Robinson College
Project Title:	Voxel Populi: A Decentralised Peer-to-Peer Voxel-Based World
Examination:	Computer Science Tripos – Part II, July 2020
Word Count:	¹
Project Originator:	Samuel John Sully
Supervisor:	Prof. Jon Crowcroft
Director of Studies:	Prof. Alan Mycroft
Overseers:	Prof. Marcelo Fiore & Dr. Amanda Prorok

Original Aims of the Project

My project aimed to create a peer-to-peer 3D world using a distributed hash table (DHT), namely Kademlia [1]. I aimed to explore this decentralised, peer-to-peer approach for Massively Multiplayer Online games (MMOs) to see if such an approach is viable. This was motivated by the advantages of the decentralised approach, such as better load balancing and longevity for the game.

Work Completed

I have completed all the work set out in my proposal, the three parts of my project are all functioning correctly. I implemented Kademlia with some modifications to better suit the virtual world application; I implemented the game server to run above the DHT and process the computation for an individual chunk for the world and I implemented the graphical client in Unity which connects to the world and allows a user to move around and interact with it. I also completed the test client which was used in the evaluation stage.

¹This word count was computed by `command?`

Special Difficulties

None.

Declaration

I, Samuel John Sully of Robinson College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

I, Samuel John Sully of Robinson College, am content for my dissertation to be made available to the students and staff of the University.

Contents

1	Introduction	1
1.1	Project Summary	1
1.2	Motivation	1
1.3	Related Works	2
2	Preparation	3
3	Implementation	5
4	Evaluation	7
5	Conclusion	9
	Bibliography	11
A	Proposal	13

List of Figures

Chapter 1

Introduction

1.1 Project Summary

My project explores a peer-to-peer architecture for MMOs or large scale simulations. This is in contrast to the more commonly used centralised approach. My project is build upon a distributed hash table which is used to locate in the peer-to-peer network the server responsible for handling any particular part of the world.

My project consists of three parts: the distributed hash table which is a modified version of the Kademlia [1] specification; the game server which runs the computation for certain segments of the world and the Unity client used to interact with the world. All these have been completed in adherence to the success criteria in my project proposal, as well as the evaluation client used in the evaluation stage. The project culminated in a large scale test using Amazon Web Services.

1.2 Motivation

The Massively Multiplayer Online Game (MMO) genre is very popular¹ in modern gaming, as an increasing proportion of the populace have access to high speed broadband the prevalence of these games continues to increase. Most of these games employ a centralised client-server mode where the creators of the MMO have a relatively small number of expensive and powerful machines which they use to handle all players.

This centralised approach often requires some form of ‘sharding’ [2], whereby players are separated into separate, independent instances (‘shards’) of the same world. Meaning that players can only interact with others connected to the same shard. The centralised approach also means that the game creators have total authoritative control over the game.

¹World of Warcraft – a popular MMO – had 7.7 million subscribers in 2019.

An alternative approach is a decentralised, peer-to-peer approach which I explore in this project. In this approach the world is separated into segments (or ‘chunks’) and each peer in the network is responsible for handling the load for a number of chunks. This approach implicitly performs load balancing and is highly failure tolerant, as a node failure can be dealt with by simply having another take over.

This has a number of advantages over the centralised, sharded approach. One significant advantage is that the world is able to be explicitly mutable (such as the voxel-based world I have implemented), with the sharded approach if a player makes a change in one shard then we may need some way of propagating these changes to the other shards while maintaining consistency. However, in my approach there is only one server which is authoritative for the state of any part of the world so there is no need for complex consensus mechanisms.

A further advantage is that the system has improved longevity. When large-scale MMOs cease to be profitable or useful for the developers, who operate the centralised servers, they often shut them down, as recently happened with the popular MMO Club Penguin [3] in 2017. With my approach, if we allow individuals to create their own servers to join the peer-to-peer network then, provided there exists a community dedicated to keeping the MMO running, it can continue to exist at no cost to the developers. It would even be possible to have multiple, separate networks running or even networks running modified versions of the game.

1.3 Related Works

There are very few large-scale, peer-to-peer MMOs, likely due to the security issues I will present in the evaluation chapter and due to the fact that it limits the ability for the developers to monetize the MMO post-release. However, it is possible that techniques similar to mine may be used behind the scenes on a number of large-scale MMOs.

One similar piece of work is SpatialOS [4], this is a platform for managing online games or simulations in the cloud. It works in a similar way to my project, by splitting up the world into segments which are administrated by separate servers. SpatialOS is produced by the startup Improbable and is still fairly new, however, it is being used in the development of a number of games.

Chapter 2

Preparation

Chapter 3

Implementation

Chapter 4

Evaluation

Chapter 5

Conclusion

Bibliography

- [1] Maymounkov, P. and Mazières, D. Kademlia: A Peer-to-peer Information System Based on the XOR Metric. <https://pdos.csail.mit.edu/~petar/papers/maymounkov-kademlia-lncs.pdf>. Accessed: 2019-10-16.
- [2] “Sharding” on Wikipedia. [https://en.wikipedia.org/wiki/Shard_\(database_architecture\)](https://en.wikipedia.org/wiki/Shard_(database_architecture)). Accessed: 2019-10-15.
- [3] “Club Penguin is shutting down”. <https://techcrunch.com/2017/01/31/club-penguin-is-shutting-down/>. Accessed: 2019-10-15.
- [4] SpatialOS by Improbable. <https://improbable.io/spatialos>. Accessed: 2020-03-20.

Appendix A

Proposal