Cluster-Computing and Parallelisation for the Multi-Dimensional PH-Index

Master Thesis

Bogdan Aurel Vancea
<bvancea@student.ethz.ch>

Prof. Dr. Moira C. Norrie
Tilmann Zaeschke
Christoph Zimmerli

Global Information Systems Group
Institute of Information Systems
Department of Computer Science
ETH Zurich

5th March 2015

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

globis

Abstract

Here comes the abstract.

Contents

1 Introduction

This chapter briefly describes the context and the motivation of the thesis and presents the objectives that are to be achieved. Finally, section 1.3 provides an overview of this thesis.

1.1 Motivation

Multi-dimensional data is widely used today, especially in domains like database management systems, geographic information systems, computer vision and computational geometry. When all of the dimensions of the data hold numerical values, this data can be viewed as a collection of points in higher dimensional spaces. Due to this nature, multi-dimensional numerical data provides the possibly of posing more complex queries based on the distance between these points in space. For example, in the context of a geo-information system, one could query for all of the points that fall inside a specific hyper-rectangle or attempt to find the nearest neighbours of an arbitrary query point.

Several point-based multi-dimensional indexing solution have been developed in the latest years, the most prominent being kD-trees and quadtrees. This type of data structures store the multi-dimensional data such that more complex operations, like range and nearest neighbour queries are executed efficiently. The PhTree is a new multi-dimensional data structure based on the quadtree. In addition to providing support for complex queries, the PhTree is also space-efficient, as its space requirements are sometimes even lower than those of multi-dimensional arrays.

As technology advances and the world becomes more connected, multi-dimensional data becomes easier to acquire and store. Because of this, it is necessary that multi-dimensional data structure need to store and manage more data than would fit a single machine. However, traditional multi-dimensional indexes like the kD-tree and quad-tree do not cover this use case as they are designed to run on a single machine.

Additionally, in the last few years the processor speed has reached the power wall and processor designers cannot increase the CPU frequency by increasing the number of transistors. Recent advances in processor design have been made by adding more cores on CPU's rather than increasing the processing frequency. Therefore, it is important that contemporary data structures be adapted to multi-core architectures by allowing them to support concurrent accesses. As with the case of the increase storage requirements, traditional multi-dimensional data structures do not support concurrent write operations.

This thesis attempts to provide a solution to these two issues by extending the PhTree multi-dimensional index to run on distributed cluster of machines. Furthermore, the PhTree is also updated to support concurrent access.

1.2 Objectives

Meta - will be removed after editing. Describe the objectives of this thesis: the distribution of the index and the addition of the concurrency support

1.3 Thesis outline

Meta - will be removed after editing. Give an overview of what each chapter will contain

2 Background

Meta - will be removed after editing. This section should provide an overview of the PhTree. It should also present important previous work concerning distributed indexes and parallel data structures.

This is an example of how to cite a scientific publication [1] from your bibliography (BibTeX[1] file). And this example shows how you create links within your documents, e.g. link to section 1.3.

2.1 The PhTree

Meta - will be removed after editing. Provide on overview of the PhTree. Should not go into too many details here, refer to PhTree paper.

2.2 Related work

Meta - will be removed after editing. Present the relevant related work.

2.2.1 Distributed Indexes

2.2.2 Concurrent data structures

[1] http://en.wikipedia.org/wiki/BibTeX

3
Index distribution

Meta - will be removed after editing. This chapter should focus on how the distributed index was implemented: how the data was split across the cluster nodes, the manner in which the queries are executed and how the entry load balancing is performed.

3.1 Challenges

Meta - will be removed after editing. Present the callenges of implementing a distributed system : scalability, load balancing, etc. Should not focus on issues like security, availability as those are not relevant to this report.

3.2 Distibution strategies

Meta - will be removed after editing. Present the possible ways in which the entries can be distributed across the cluster nodes. Talk about the advantages and disaadvantages of each approach. Say which approach was chosen and why.

3.2.1 Hashing

3.2.2 Spatial splitting

3.2.3 Z-Order curve splitting

3.3 Algorithms

Meta - will be removed after editing. This section should explain how the queries should be executed on the distributed system. Present the load balancing algorithm.

3.3.1 Point queries

3.3.2 Range queries

3.3.3 Nearest neighbour queries

3.3.4 Entry load balancing

3.4 Architecture

Meta - will be removed after editing. This section should explain how the queries should be executed on the distributed system. Present the load balancing algorithm.

4
Concurrency

Meta - will be removed after editing. Present the concurrency strategies that could be added to the PhTree and explain the consistency model associated with each strategy.

The PhTree does not currently support concurrent write operations. There are several strategies that could be employed to add concurrent writes.

4.1 Challenges

4.2 Concurrency strategies

4.2.1 Copy-on-Write

4.2.2 Locking

5 Implementation

Meta - will be removed after editing. Present the implementation architecture and the techologies used.

5.1 System description

Meta - will be removed after editing. Describe the system, include class/deployment diagrams.

5.2 Technologies

Meta - will be removed after editing. Describe the technologies used, the reasons for which these technologies were chosen and any alternatives.

6
Evaluation

Meta - will be removed after editing. Explain how the system should be evaluated, present and explain the benchmarks

7
Conclussions

Meta - will be removed after editing. Conclude the report. This should reiterate the main points of the report and try to mirror the introduction

8
Future work

Meta - will be removed after editing. Present the points that were not tackled by the thesis and talk about possible future work.

List of Figures

List of Tables

Acknowledgements

Bibliography

[1] Alfonso Murolo. Designing wordpress themes by example. Master's thesis, ETH Zurich, 2013.