CSCI 540 - Advanced Databases Course Project Progress Report for November 14th

Larry Lynn
Liessman Sturlaugson
Cole Schock

November 13, 2011

Abstract

This report documents the progress of the team tasked with implementing iMinMax. The tasks for this past week can be divided into four categories: 1) setting up a team development server, 2) evaluating the B^+-Tree implementations, 3) identifying requirements for iMinMax implementation, and 4) initial work on the final paper.

1 Team Development Environment

To aid in the coordinated development of the iMinMax algorithm, the team has set up a development server, including a git repository and SSH access. The server provides a consistent development environment and a source code repository for all team members. Furthermore, the team has identified the need for hosting the source code files at the conclusion of the project to facilitate the scientific goal of reproducibility.

The server and developing environment currently have the following specifications, although it is foreseeable that the software will later include additional libraries:

1.1 Hardware

- MSI Wind Nettop 100
- 1.6 GHz Intel Atom Dual Core N330 Processor
- 2 GB DDR2 RAM (200-Pin SODIMM Laptop Memory)
- 2 TB SATA hard drive (Samsung Spinpoint F4 EcoGreen 2 TB Serial ATA / SATA 3.0 Gbps 3.5 Inch Hard Drive, 5400RPM) of which 814GB is free

1.2 Build Environment

- OS: Debian GNU/Linux
- OS Version: Squeeze (Debian 6.0.3)

1.3 Software

- C++ compiler: g++ (Debian 4.4.5-8) 4.4.5
- GNU C Library (Debian EGLIBC 2.11.2-10) stable release version 2.11.2, by Roland McGrath et al.
- libc6 version 2.11.2-10

2 B^+-Tree Implementation Evaluation

The team has begun evaluating the two proposed B^+-Tree implementations:

- STX B^+-Tree (idlebox.net/2007/stx-btree/)
- ScalingWeb.com B^+-Tree (www.scalingweb.com/bplus_tree.php)

The team has identified the following criteria for evaluation:

- Number of lines of code
- Number of dependencies
- Ease of incorporating tree statistics variables required by the project
- Tree save/load capabilities
- Quality of documentation

As of yet, the team has not chosen one over the other, as evaluations are still ongoing.

3 Algorithm Implementation

The team is currently examining [4], the iMinMax extension for decreasing radius KNN queries, to see how the implementation may need to be modified from the original algorithm described in [3].

The team has identified some initial considerations for implementing the iMinMax algorithm:

- A function for converting a d-dimensional vector into its 1-dimensional index
- A function performing a point query by traversing the B^+-Tree

- A function for performing a 1-dimensional subquery, callable by both the range and KNN query functions
- A function performing a range query by calling the subquery function (at most) d times
- A function performing a KNN query by calling the subquery function (at most) d times
- A function to refine the candidate set at each step of a range query
- A function to refine the candidate set at each step of a KNN query

The team is also looking into C++ implementations for aiding in:

- Command line arguments parsing
- Parsing input CSV files
- Writing output to CSV files for charts and graphs

4 Paper

The team has also begun preparing for the final paper, setting up the IEEE LaTeX template and performing some initial literature review.

4.1 IEEE LaTeX Template

The team has downloaded the LaTeX template for IEEE Conference Proceedings (www.ieee.org/conferences_events/conferences/publishing/templates.html) and begun creating an outline for the various sections required for the paper.

4.2 Literature Review

In preparation for the final paper, the team has begun their literature review for high-dimensional indexing strategies supporting point, range, and KNN queries. In addition to the Pyramid-Technique [1], the iMinMax [3], and the iDistance [2, 7] papers, the team is currently looking at [4], which extends iMinMax to support KNN queries using a decreasing radius KNN search strategy.

Furthermore, for providing a full background of the iMinMax scheme, [6] has been identified as providing an *approximate* KNN extension to iMinMax. Other developments include [5], which adapts the VA-file with iMinMax for an approximation algorithm that decreases the number of leaf nodes in the B^+-Tree that must be scanned.

References

[1] Stefan Berchtold, Christian Böhm, and Hans-Peter Kriegal. The pyramid-technique: towards breaking the curse of dimensionality. *SIGMOD Rec.*, 27:142–153, June 1998.

[2] H. V. Jagadish, Beng C. Ooi, Kian L. Tan, Cui Yu, and Rui Zhang. iDistance: An adaptive B^+-tree based indexing method for nearest neighbor search. *ACM Trans. Database Syst.*, 30(2):364–397, June 2005.

[3] Beng Chin Ooi, Kian-Lee Tan, Cui Yu, and Stephane Bressan. Indexing the edgesa simple and yet efficient approach to high-dimensional indexing. In *Proceedings of the nineteenth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, PODS '00, pages 166–174, New York, NY, USA, 2000. ACM.

[4] Q. Shi and B. Nickerson. Decreasing Radius K-Nearest Neighbor Search Using Mapping-based Indexing Schemes. Technical report, University of New Brunswick.

[5] Shuguang Wang, Cui Yu, and Beng Ooi. Compressing the index - a simple and yet efficient approximation approach to high-dimensional indexing. In X. Wang, Ge Yu, and Hongjun Lu, editors, *Advances in Web-Age Information Management*, volume 2118 of *Lecture Notes in Computer Science*, pages 291–302. Springer Berlin / Heidelberg, 2001.

[6] Cui Yu, Stéphane Bressan, Beng Chin Ooi, and Kian-Lee Tan. Querying high-dimensional data in single-dimensional space. *The VLDB Journal*, 13:105–119, May 2004.

[7] Cui Yu, Beng C. Ooi, Kian-Lee Tan, and H. V. Jagadish. Indexing the Distance: An Efficient Method to KNN Processing. In *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases*, pages 421–430, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.