

CS618: Group 08

Distributed B+ Tree

Proneet Verma	Saurav Kumar
Student ID 44	Student ID 45
Roll Number 12520	Roll Number 12641
<code>proneetv@iitk.ac.in</code>	<code>ksaurav@iitk.ac.in</code>
Dept. of MSE	Dept. of CSE
Indian Institute of Technology, Kanpur	

Initial Report
March 21, 2015

Abstract

B+ Trees are efficient and effective in indexing data stored in databases. This project aims to efficiently implement and analyze performance of general queries in B+ Trees distributed over a multi-node computer network. We plan to consider several factors for *data distribution among servers*, such as geographical location of the node, bandwidth of network connecting the root and the node, hardware configuration of the node, and *likelihood of data* being queried in order to arrive at a comprehensive analysis.

1 Introduction and Problem Statement

The problem consists of 2 parts:

- Implementation of distributed database which uses B+ Tree to efficiently index the database
- Analysis of response time on general queries

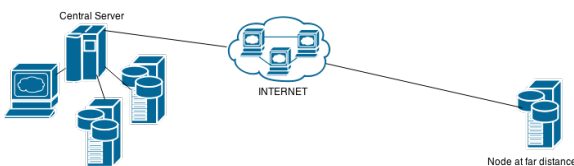


Figure 1: Distributed Database

Score based system: For this problem, we will use a score-based technique to distribute data among the nodes. Servers will have their scores computed based on their average response time which depends on network response time, processing speed and Disk I/O time. This score will roughly determine how much data should be kept on which servers.

Likelihood: Since some data (keys) may be returned more often than others depending on distribution of data and query responses, we will try to find an optimum strategy to keep more frequent data in servers with better score. To test the effectiveness of the strategy we will train the model with some distribution of queries and test with same distribution of query data. For actual practise, we propose online migration of nodes between servers based on query and response analytics

1.1 Related Material

<http://dl.acm.org/citation.cfm?id=1566451>

2 Algorithm or Approach

3 Results

4 Conclusions

References