COMP4321
Search Engine for Web and Enterprise Data
Team Project Phase I

[Group members]
CHAN Wing Yan Vannesa (20212130, wyvchan)
O Pui Wai (20198827, pwo)
TSUI Ka Wai (20197524, kwtsuiaa)

I. Database Specification

Class: MappingIndex			
Mapping Key -> Value			
Attribute		**Data type**	
key		String	
value		int	
Instance: WordMappingIndex		Instance: URLMappingIndex	
Mapping word -> wordID		URL -> Page-ID	
Attribute	**Data type**	**Attribute**	**Data type**
word	String	url	String
wordID	int	paegID	Int

Class: InvertedIndex		
wordID ->{Page-ID, <word positions>}		
Attribute	**Data/Object type**	
wordID	int	
HashMap<int, Posting>	Class : Posting(pageID, posting)	
	pageID	pageID
	wordPosList	wordPosList

Class: PageProperty	
pageID -> pagesize, title, modDate, size	
Attribute	**Attribute**
pageID	pageID
title	String
url	String
modDate	Date
size	Int

Class: ForwardIndex	
pageID -> {wordID}	
Attribute	**Data type**
pageID	int

II. JDBM Schema

MappingIndex (key, value)
InvertedIndex (wordID, HasMap<int, Posting>
PrageProperty (pageID, title, url, modDate, size)
ForwardIndex (pageID)

III. Reason for structures

The database has adopted Hash Tree structure that imported from JDBM library. Hash Tree structure provided a fast retrieval by storing keys.

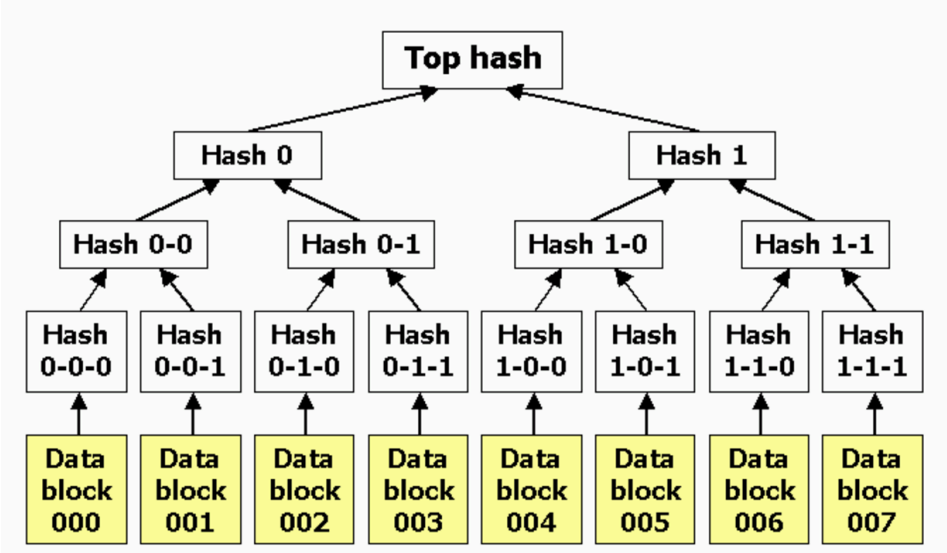

figure III.1 The Concept of hash tree

Advantages of Hash Tree

A tree hash has an advantage that allow programmer to compute the hash of both a portion of a file and the entire file anytime. Hashing the file chunk and the entire file separately. With a tree hash, the hash of the chunk is used to compute the hash of the file, it takes no extra work to compute both hashes. As a result, the time of retrieval is fast.

Structure of Hash Tree

Hash tree is suitable data structure for creating search engine by putting key value pair into the tree. This structure fits our needs perfectly since our project specified to put key and value in a data structure for data input and retrieval. This data structure is believed to be an ideal solution to our project.