EXTENDIBLE MULTIDIMENSIONAL SPARSE ARRAY REPRESENTATIONS FOR DATA WAREHOUSING

Catherine Ann Honegger

A masters proposal submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Master of Science in Engineering.

Johannesburg, September 2017

The financial assistance of the National Research Foundation (NRF) towards this research is hereby acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Declaration

I declare that this masters proposal is my own, unaided work, except where otherwise acknowledged. It is being submitted for the degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

Signed this ____ day of ____________ 20__

____

Catherine A. Honegger

Abstract

Data warehousing is a method used to combine multiple varied datasets into one complete and easily manipulated database as a decision support system. Research over several years has concluded that multidimensional representation of data in data warehousing not only gives a good visual perspective of the data to the user, but also provides a storage scheme for efficient processing. Since data in a data warehouse grows dynamically the multidimensional representation should also be expanded dynamically. Efficient dynamic storage schemes for storing dense, extendible, multidimensional arrays by chunks have been developed [7, 8]. However in data warehousing the corresponding multidimensional arrays are predominantly sparse arrays. Currently no storage or mapping techniques allow for the extendibility of multidimensional sparse arrays [12]. This research proposes to improve the modelling and representation of extendible multidimensional sparse arrays so that useful compression and storage efficiency can be determined. Our work addresses extendibility of the sparse array only, and does not concern the shrinking of the sparse array.

Acknowledgements

I would like to thank my research supervisor, Professor Ekow Otoo, for taking me on as a masters student. My special thanks also go to Professor Ken Nixon for his help and support in early stages of the proposal.

I am grateful to the University of the Witwatersrand for awarding me a Postgraduate Merit Award as financial assistance during my studies.

I am also grateful to the National Research Foundation (NRF) South Africa for their financial support for my study.

Contents

List of Figures

List of Tables

List of Symbols

The principal symbols used in this thesis are summarised below, and the first equation in which each symbol appears is given.

Nomenclature

2D	two dimensional
3D	three dimensional
4D	four dimensional
BESS	Bit-Encoded Sparse Storage
BxCCS	Bit-Encoded Compression Column Storage
BxCCR	Bit-Encoded Compression Row Storage
CCS	Compressed Column Storage
CRS	Compressed Row Storage
GB	Giga Byte
GCC	GNU Compiler Collection
HDF	Hierarchical Data Format
HDFS	Hadoop Distributed File System
NetCDF	Network Common Data Form
OLAP	On-Line Analytical Processing
OS	Operating System
PTCS	PATRICIA Trie Compressed Storage
SQL	Structured Query Language
TM	Trademark

xCCS	Extended CCS
xCRS	Extended CRS

Chapter 1

Introduction

This chapter introduces the research and provides a detailed motivation for the significance of the research.

1.1 Problem Motivation

Over recent years with the increase in smartphone use, internet use, and virtually every activity leaving a digital trace, society has been able to store and collect more data, leading to the emergence of huge databases with quintillion bytes of data being produced everyday [1]. Data has been dubbed "The world's most valuable resource" in 2017 as economists believe it is "the oil of the digital era" with the five most valuable listed firms in the world for 2017 being data titans[1].

By combining different data sets, useful information can be retrieved from the stored data. Information resources are incredibly important and valuable in business management and decision-making [2, 3]. By collecting lots of data, companies are able to improve their products and entice more clients with these improvements [1]. Due to their significance, these data assets must be stored properly and accessed easily [2]. Traditional data analysis tools are not able to handle such growing quantities of data. Thus various applications and technologies for managing big-data and high volume data-streams in data intensive computing is becoming essential. Further big-data, data-mining and machine learning are now of major concern in several scientific domains. Thus the field of Big Data research has materialised from the need to store such large quantities of information. Data analysis in all societal domains involves the storage, retrieval, processing and visualisation of huge databases [4].

Big data is a rapidly growing on a global scale. However there is a bottleneck of technology that is required to extend the capabilities of the field. Several big data technologies are required that currently don't exist, including: architectures, algorithms, and techniques [5, 6].

1.2 Significance

A new phenomenon, known as data warehousing, was developed as a result of the large quantities of data that need to be stored in recent years [2]. Data warehousing is a method used to combine multiple varied datasets into one complete and easily manipulated database as a decision support system. On-Line Analytical Processing (OLAP) can then be performed on these databases by an organisation in order to analyse the data, complete data-mining and determine trends. The organisation is then able to build business intelligent decisions by utilising the analysed data. Data warehousing has many applications in medical informatics and public-health, smart cities, mining, energy, physics and financial systems to name a few.

The storage of the datasets influences the accuracy, speed and performance of the data analysis and thus it is crucial to assess how this information is stored and accessed. Research over several years has concluded that multidimensional representation of data in data warehousing not only gives a good visual perspective of the data to the user, but also provides a storage scheme for efficient processing. Several research advancements have been made in multidimensional arrays in order to ensure that the datasets are efficient and inexpensive. Multi-dimensional arrays when used for their selection, aggregation, summation and other range queries are processed more efficiently than their SQL counter part in huge database queries. Since data in a data warehouse grows dynamically the multidimensional representation should also be expanded dynamically. These multidimensional datasets are continuously increasing by appending new data to the dataset as new information is added [4].

Recent breakthroughs in extendible multidimensional arrays have led to a new area of study in data warehousing, which requires further research and development. Efficient dynamic storage schemes for storing dense, extendible, multidimensional arrays by chunks have been developed [7, 8]. However in data warehousing the corresponding multidimensional arrays are predominantly sparse arrays. It is thus important to analyse extendible multidimensional sparse arrays. There have been a number of advances made in the performance and efficiency of storage schemes for representing multidimensional sparse arrays [9, 10, 11]. However, currently no storage or mapping techniques allow for the extendibility of multidimensional sparse arrays [12].

With computation and storage costs being the most expensive part of data warehousing and data-mining, being able to utilize extendible multidimensional sparse arrays will extend the capabilities and domains for big-data analysis.

1.3 Problem Statement

The question raised is whether multidimensional sparse arrays can be stored in such a way that new elements and hyper-plane dimensions can be added. The problem that we are focused on requires the use of new storage techniques to investigate the extendibility of multidimensional sparse arrays with regards to computer performance and storage efficiencies.

An illustrative example of where data warehousing is used is given in Table 1.1. Here we will have a fact table displaying the number of items sold on a particular day. This can be further extended into a data repository consisting of items sold per day per store as a relational table. In order to analyse the data to improve a company's sales, they might want to know the number of items sold, or which item is sold more on which day. I.e more carrots are sold on Wednesdays. In order to summarise these values and find patterns in the data these fact tables are converted into multidimensional arrays as shown in XXX. Using these arrays, roll-in and roll-out queries can be conducted.

1.4 Application

NB - Still working on this section

There are several problems that currently exist in the world that would utilize Big Data. A major example would be the frequency and quality of demographic statistics produced by countries as well as demographic statistics deficits. Such applications are particularly of major demand in developing countries.

- There are 46 African countries operating without a complete birth and death registration system.
- Population censuses have not been conducted since 2010 in several African countries.

Table 1.1: A typical data warehousing example.

Item	**Time**	**Amount**
Apple	Monday	12
Orange	Monday	10
Carrot	Monday	9
Banana	Monday	9
Grapefruit	Monday	10
Apple	Tuesday	7
Orange	Tuesday	8
Banana	Tuesday	18
Apple	Wednesday	5
Banana	Wednesday	3
Carrot	Wednesday	10

- Several African countries have not conducted an agricultural census in the last ten years. South Africa in particular held their last agricultural census in 2007.

Having up-to-date, reliable demographic statistics enables people within the population to partake in highly important activities. These include gaining quality education, formal employment, voting in elections, access to financial services, obtaining passports and obtaining IDs to name but a few. Reliable data also enables governments to budget better and improve the delivery of public goods and services [13].

1.5 Early Results from Literature

NB - Still working on this section

1.6 Contribution to Field

NB - Still working on this section

As data is constantly growing, there needs to be a method to provide for this extendibility.

The aim of the proposed study is to improve the modelling and representation of extendible multidimensional sparse arrays so that useful compression and storage efficiency can be determined. The representation of the extendible multidimensional sparse arrays must include the characteristics of an array format, such that the array can take on any designed multiple hyper-plane dimensions. The characteristics of the array allow for easy data access. In addition, this allows for both drill-in and roll-out queries. Where drill-in queries allow for detailed data access and roll-out queries allow for summary data access.

The modelling of the extendible multidimensional sparse arrays will be able to contribute to developing algorithms that can process data at higher speeds, use less physical computational resources and improve the usage of computational power.

Our work addresses extendibility of the sparse array only, and does not concern the shrinking of the sparse array.

1.7 Organisation of the Proposal

Chapter 2 provides a background on data warehousing and OLAP, extendible multidimensional dense arrays and multidimensional sparse arrays. Chapter 3 provides the proposed methodology for developing extendible multidimensional sparse array representations for data warehousing. The experimental setup is provided in Chapter 4. Chapter 5 details the preliminary results of the research. Expansion on the preliminary results is discussed in Chapter 6. Possible issues that could arise during the research as well as their proposed solutions are presented in Chapter 7. A schedule for the completion of the research is outlined in Chapter 8. A summary of the proposal is given in Chapter 9.

Chapter 2

Background

This chapter provides a background on data warehousing and OLAP, extendible multidimensional dense arrays and multidimensional sparse arrays.

2.1 Methodologies

In order to develop a model for extendible multi-dimensional sparse arrays, the current representations of extendible multi-dimensional arrays must be analysed. The study of the current representations will give a clear indication of how to integrate the representation of extendible multi-dimensional sparse arrays, so that the model can easily be integrated into the current system, preventing unnecessary additional costs. Once a model has been developed, compression of data as well as computing speed will be assessed.

2.2 Sparse Array Representations

There are numerous studies that have been conducted on the efficiency of different sparse array representation schemes [3, 10]. In order to accurately understand these models, we must first understand exactly what is mean by a sparse array. If an array is sparse, it has very few non-zero elements relative to the product of the cardinalities [3].

There are three main methods used for 2-dimensional sparse array representations, namely:

1. Compressed Row (or alternatively Column) Storage (CRS/CCS)
2. Linked list

3. Triplicate

There are six other methods that are used for multidimensional representations, namely:

1. Bit-Encoded Sparse Storage (BESS)
2. Extended CRS/CCS known as xCRS and xCCS
3. PATRICIA Trie Compressed Storage (PTCS)
4. Bit-Encoded Compressed Row (or Column) storage (BxCRS/BxCCS)
5. Hybrid approach (Hybrid) that combines BESS with xCRS

BESS is the simplest low level storage scheme that chops every point into bit block dimensions. BESS has decent storage, but is traditionally static with the maximum number of chunks being determined algorithmically.

The PTCS is unique and allows for extendibility with regards to data warehousing operations.

The data warehousing operations that are used include: Get, Insert, Addition, Multiplication, Subtraction, Division, Slices and range or box access .

2.3 Data Warehousing

In Chapter 1 it was discovered how important information is in the current world [2]. It was also discussed that due to the vast amounts of data being created and stored every day and the inability for traditional database methods (e.g. SQL database systems) to deal with such large volumes of data, data warehousing was created for decision making.

Bill Inmon defines data warehousing as an architecture that is a subject-oriented, non-volatile, integrated, time variant collection of data created for the purpose of managements decision making.

Ralph Kimball defines a data warehouse as a copy of transaction data specifically structured for query and analysis.

2.4 Big Data Representations

Big Data characterization

- Volume - Scalability, storage to grow - investigate using a parallel file system
- Velocity - Cope with speed, rate at which data comes in - times series
- Variety - Data types - different formats i.e multimedia databases for geographic location - GI, Spacial, Moving Targets, structured, unstructured/text, legal documents - Databases that capture almost everything.
- Veracity - validity
- Value

- Capable of holding very large amounts of data.
- Hold the data in inexpensive storage devices.
- Processing is done by the Roman census method.
- Data is stored in an unstructured format.
- Hierarchical Data Format (HDF5)
  - Data model, library, and file format for storing and managing data
  - Supports an unlimited variety of datatypes, and is designed for flexible and efficient I/O and for high volume and complex data.
  - Portable and extensible, allowing applications to evolve in their use of HDF5.
  - Tools and applications for managing, manipulating, viewing, and analyzing data in the HDF5 format.
  - Organized in a hierarchical structure, with two primary structures: groups and datasets.
    - group: a grouping structure containing instances of zero or more groups or datasets, together with supporting metadata.
    - dataset: a multidimensional array of data elements, together with supporting metadata.

- Hadoop
  - Distributed storage and processing of very large data sets.
  - Consists of computer clusters built from commodity hardware.
  - Assumption that hardware failures are common occurrences and should be automatically handled by the framework
  - Hadoop can run parallel queries over flat files. This allows it do basic operational reporting on data in its original form.
  - Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data in parallel
  - Hadoop framework includes following four modules:
    - Hadoop Common: These libraries provides filesystem and OS level abstractions and contains the necessary Java files and scripts required to start Hadoop.
    - Hadoop YARN: This is a framework for job scheduling and cluster resource management.
    - Hadoop Distributed File System (HDFS): A distributed file system that provides high-throughput access to application data.
    - Hadoop MapReduce: This is YARN-based system for parallel processing of large data sets.
- Network Common Data Form (NetCDF)
  - Set of software libraries and self-describing, machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data.

2.5 TileDB storage manager

- Used for data which is represented as multi-dimensional arrays.
- Sparse arrays significantly impact storage and application performance
- when storing data cells that are accessed together should be co-located - global cell orders in data tiles in sparse arrays include row-major vs column-major
- bookkeeping is very important for locating the correct data as exact distribution of the non-empty cells is unknown

2.6 Impact

2.7 New Results

Chapter 3

Methodology

Based on the literature review in Chapter 2, it is evident that the extendibility of multidimensional sparse arrays is an incredibly important topic that has been overlooked, specifically with regards to data warehousing and OLAP. In order to make a contribution to this field, an extendible multidimensional sparse array model must be developed, implemented and evaluated. This chapter presents the proposed model design and experimental procedures for the research study.

3.1 Extendible Multi-Dimensional Sparse Arrays

In order to represent a data warehouse in a multidimensional space a formula needs to be defined to increase the dimensionality of the storage scheme without changing the mapping function f(). As sparse data warehouses consist of large quantities of information with few non-zero elements, an efficient storage scheme that represents these non-zero elements must be chosen. Furthermore a method for extendibility of the multi-dimensional sparse array representation must be chosen.

3.1.1 Multi-Dimensional Sparse Array Representation

In order to improve the storage efficiency of multi-dimensional sparse arrays, a good storage scheme must be chosen. A comparison of the different sparse array storage techniques detailed in Section 2 is carried out.

CRS and CCS are both only used for 2-dimensional arrays. In the work of Goil et al. [10] it is clear that BESS out performs Offset-Value pair. In the work of Wang [3] it is evident that although PTCS has a better storage ratio and xCRS/xCCS and BxCRS have faster element retrieval times, BESS has a faster construction time and a faster multi-dimensional aggregation time than all of the other storage schemes.

BESS is a very simple and efficient multi-dimensional sparse array representation as it maps a multi-dimensional sparse array space to a one-dimensional array space, resulting in a bit encoded key index [3].

A multi-dimensional sparse array data model using a BESS array representation scheme as well as a model using a CRS representation scheme will be analysed.

3.1.2 Extendible Multi-Dimensional Sparse Array Representation

NB - Still working on this section

By making use of chunking techniques on a BESS array representation scheme as well as a CRS array representation scheme. The proposed extendibility model can increase the dimensionality of the storage scheme by either increasing the density of the array or increasing the dimension of the indices or bounds of the array. When increasing a sparse array by density, the bounds and structure remain the same, new elements are slot into a chunk where there was a free space. Two algorithms will need to be developed and implemented in software in order to determine which algorithm performs better.

To assess the performance of the extendible multi-dimensional models, a measure of the storage efficiency and order of retrieval of queries shall be conducted using basic construction, low level retrieval and partial match query array operations.

The two algorithms will be tested on a 2D array (both CRS and BESS), a 3D array(only BESS) and a 4D array(Only BESS) to asses their limitations on different dimensionalities.

Figure 3.1 shows the main steps in the proposed methodology.

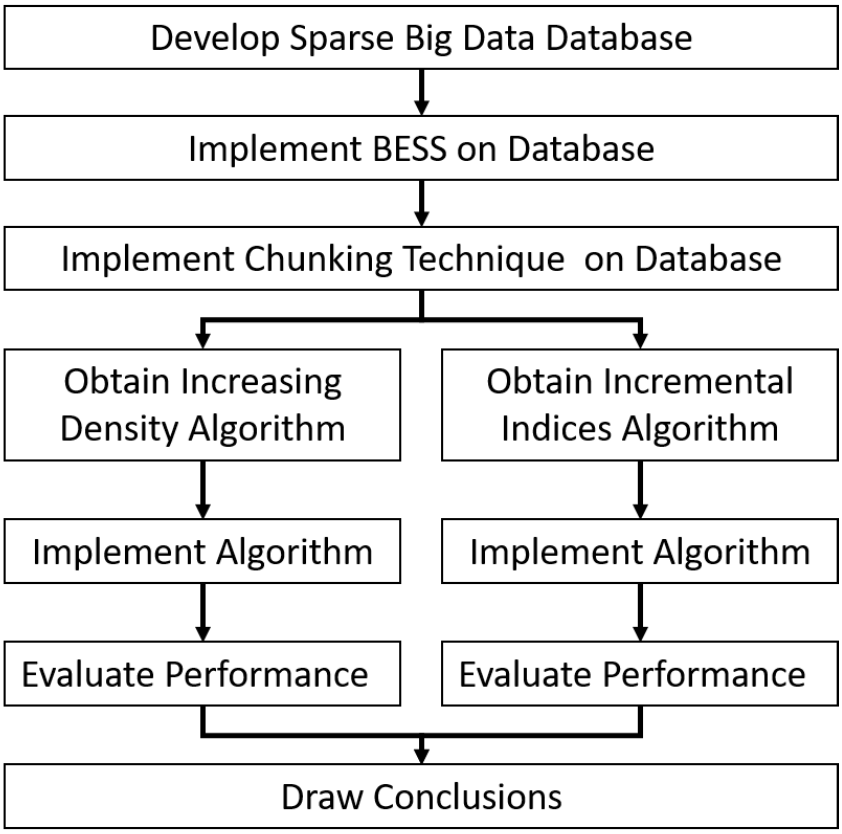
Develop Sparse Big Data Database
Implement BESS on Database
Implement Chunking Technique on Database
Obtain Increasing Density Algorithm
Obtain Incremental Indices Algorithm
Implement Algorithm
Implement Algorithm
Evaluate Performance
Evaluate Performance
Draw Conclusions

Figure 3.1: Proposed Methodology

Chapter 4

Experimental Setup

This chapter details the computational environment and the experimental data in order to allow for independent reproducibility and confirmation of the conducted experiments.

4.1 Computational Environment

4.1.1 Experimental Computational Model and Software Environment

The experiments will be run on a laptop with an Intel(R) Core(TM) i5-2450 multi-core processor at 2.50 GHz and 4GB of memory running Ubuntu 64-bit Linux 16.04.3 LTS.

The experiments will be implemented in C using GNU Compiler Collection (GCC) 7.2.

4.2 Experimental Data

NB - Still working on this section

4.3 Software Engineering Practice

NB - Still working on this section

Chapter 5

Preliminary Results

Chapter 6

Future Work

Chapter 7

Risk Management

Chapter 8

Schedule and Time-Line for Completion

This chapter details the research tasks and their proposed date of completion.

8.1 Proposed Task Completion Time-Line

The research is expected to be completed by April 2018. Regular meetings are held with the supervisor once a week. There is an open door policy enabling casual meetings to be arranged ensuring that there is constant feedback throughout the lifespan of the proposed research.

Table 8.1 presents a set of major tasks along with their preliminary completion dates for the proposed work.

Table 8.1: Proposed Task Completion Times.

Task Description	**Proposed Completion Date**
Begin Research Project	January 2017
Literature Review of Data Warehousing	March 2017
Literature Review of Sparse Array Representation	May 2017
Literature Review of Dynamic Dense Array Representation	July 2017
Documentation of the Methodology	September 2017
Submit Research Proposal for Approval	October 2017
Visual Presentation of the Research Proposal	October 2017
Develop and Implement Storage and Chunking Techniques	December 2017
Develop Dynamic Expansion Algorithms	February 2018
Evaluate Performance and Analyse Results	April 2018
Submit Research Dissertation for Approval	April 2018

Chapter 9

Summary

Bibliography

[1] T. Economist. "The World's Most Valuable Resource." *The Economist*, vol. 423, no. 9039, p. 7, May 2017.

[2] M. Golfarelli and S. Rizzi. *Data Warehouse Design: Modern Principles and Methodologies*, chap. 1, pp. 1–42. India: McGraw-Hill Professional, first ed., Jan. 2009.

[3] H. Wang. *Sparse Array Representations And Some Selected Array Operations On GPUs*. MSc Dissertation, University of the Witwatersrand, Johannesburg, South Africa, Jul. 2014.

[4] E. J. Otoo and D. Rotem. "Efficient Storage Allocation of Large-Scale Extendible Multidimensional Scientific Datasets." In *18th International Conference on Scientific and Statistical Database Management*, pp. 179–183. IEEE, 2006.

[5] B. Twala. "Big Data for Africa." presentation, Feb. 2017.

[6] N. E. Moukhi, I. E. Azami, and A. Mouloudi. "Data Warehouse State of the art and future challenges." In *2015 International Conference on Cloud Technologies and Applications (CloudTech)*. Marrakech, Morocco: IEEE, Jun. 2015. URL `http://ieeexplore.ieee.org/abstract/document/7337004/`.

[7] G. Nimako, E. J. Otoo, and D. Ohene-Kwofie. "Chunked Extendible Dense Arrays for Scientific Data Storage." In *41st International Conference on Parallel Processing Workshops*, pp. 38–47. 2012. URL `http://ieeexplore.ieee.org/document/6337461/`.

[8] P. Pedereira. "Cubrick: A Scalable Distributed MOLAP Database for Fast Analytics." In *VLDB 2015 PhD Workshop*. Very Large Data Base Endowment Inc., Hawaii: Springer-Verlag, Jul. 2015. URL `http://www.vldb.org/2015/wp-content/uploads/2015/07/pedreira.pdf`.

[9] E. J. Otoo, H. Wang, and G. Nimako. "Multidimensional Sparse Array Storage for Data Analytics." In *IEEE 18th International Conference on High Performance Computing*

and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems, pp. 1520–1529. 2016.

[10] S. Goil and A. Choudhary. "BESS: A Sparse Storage Structure of Multi-dimensional Data for OLAP and Data Mining." Tech. Rep. CPDC-TR-9801-005, Centre for Parallel and Distributed Computing, Northwestern University, 1997.

[11] H. W. E.J. Otoo, G. Nimako. "New Approaches to Storing and Manipulating Multi-Dimensional Sparse Arrays." In *SSDBM 2014*, 41. ACM, 2014. URL `http://dl.acm.org/citation.cfm?id=2618281`.

[12] G. Nimako. *Chunked Extendible Arrays and its Integration with the Global Array Toolkit for Parallel Image Processing*. PhD Dissertation, University of the Witwatersrand, Johannesburg, South Africa, Oct. 2016.

[13] M. I. Foundation. "Strength in Numbers: Africa's Data Revolution." online, 2015.

[14] C. H. Roth and L. John. United States of America: Cengage Learning, third ed., 2016.