# A Tree of Life Approach for Multidimensional Data

**Dr. Kavita Pabreja**
*(Associate Professor and Head - Department of Computer Science,*
*Maharaja Surajmal Institute, GGSIP University, New Delhi, India )*

***ABSTRACT:*** *With the recent exponential growth of ICT, there is explosive growth of data of varied nature. Human effort is always to efficiently store the data for the current and future usages. In the present day, we create data with a speed that 90% of the total data in the world today has been created in the last few years alone, as explained by IBM[1]. There are many problems and challenges to handle this big data i.e. heterogeneity, scale, timeliness, complexity, and associated privacy. One has to develop suitable and flexible strategies to derive knowledge and value from this price less data. Data analysis, organization, retrieval, and modeling are foundational challenges. Of the three main characteristics of Big data i.e. Volume, Velocity and Variety; this study is a step towards fast processing of the data that is generated at a fast speed. The traditional OLAP systems use static data cube approach or partial materialization of cuboids in order to ensure fast query performance which does not provide an up-to-date data warehouse for decision support systems and requires a lot of memory space. This paper describes Big Data and philosophy of its retrieval through tree of life approach that uses the power of Multi-core processing for efficient parallel computing in real time.*

***Keywords -*** *Big Data, OLAP, Parallel processing, Multi-core processors, data cube, tree of life*

## 1. INTRODUCTION

1.1 Background of Big Data

Big data is a massive volume of both structured and unstructured data that is so large that it's difficult to process with traditional database and software techniques [2].

One has to develop suitable and flexible strategies to derive knowledge and value from this price less data. The problems start right away during data acquisition, when the data tsunami requires us to make decisions, currently in an ad hoc manner, about what data to keep and what to discard, and how to store what we keep reliably with the right metadata. Much data today is not natively in structured format; for example, tweets and blogs are weakly structured pieces of text, while images and video are structured for storage and display, but not for semantic content and search: transforming such content into a structured format for later analysis is a major challenge. The value of data explodes when it can be linked with other data, thus data integration is a major creator of value. Since most data is directly generated in digital format today, we have the opportunity and the challenge both to influence the creation to facilitate later linkage and to automatically link previously created data. Data analysis is a clear bottleneck in many applications, both due to lack of scalability of the underlying algorithms and due to the complexity of the data that needs to be analyzed. Finally, presentation of the results and its interpretation by non-technical domain experts is crucial to extracting actionable knowledge.

1.2 History of Data Management

Meteorological, archeological & other geosciences were creating data since last two hundred years or more. They were designing their own methods to store & retrieve the information. There has been some standardization e.g. GRIB, GRIdded Binary or General Regularly-distributed Information in Binary form, a mathematically concise data format commonly used in meteorology to store historical and forecast weather data [3]. It is standardized by the World Meteorological Organization's Commission for Basic Systems.

During the last over four decades, data management principles such as physical and logical independence, declarative querying and cost-based optimization have led to a multi-billion dollar industry. More importantly, these technical advances have enabled the first round of business intelligence applications and laid the foundation for managing and analyzing Big Data today. Traditional business intelligence and data warehouse solutions are not engineered for this type of dynamic and unstructured data. Different approaches were proposed like network databases, hierarchical databases & relational databases. Relational databases became popular because of its simplicity to use. But none of these data bases were found to be useful by the big data users' community like meteorologists.

Multi dimensional data base approach is a good choice. A multidimensional data schema is represented by using the concept of a cube. A cube is a logical organization of multidimensional data as shown in Fig. 1. A cube is derived from a fact table. Edges of cube are referred to as Dimensions that categorize a cube's data. Each dimension is a grouping of common or related columns from one or more tables into a single entity [4][5]. Dimensions group multiple columns from one or more tables into one single entity organized around one or more hierarchies and ordered by levels. These objects can then be used to provide a very simple query interface.

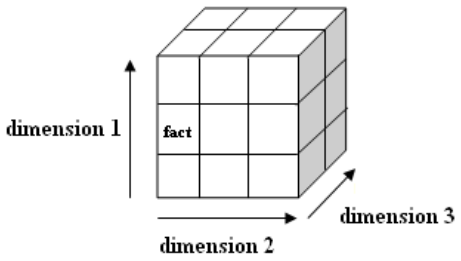


*Figure 1 A multidimensional cube having 3 dimensions*

A cube contains measures that share the same dimensionality. Measures are just like arrays and are automatically associated to the physical fact table column and related dimension tables. This transformation from fact table column to measure, insulates the user from the complexity of the underlying schema and from the need to understand how the various parts of the schema are joined together. Within an Online analytical processing (OLAP) environment, it is extremely easy to create new measures. OLAP server allows users to "drill down" or "roll up" on hierarchies, "slice and dice" particular attributes, or perform various statistical operations such as ranking and forecasting. Fig.2 illustrates the basic model where the OLAP server represents the interface between the data warehouse and the reporting and display applications available to end users [4].

To support this functionality, OLAP relies heavily upon a classical data model known as the data cube [6]. Conceptually, the data cube allows users to view organizational data from different perspectives and at a variety of summarization levels. It consists of the base cuboid, the finest granularity view containing the full complement of d dimensions (or attributes), surrounded by a

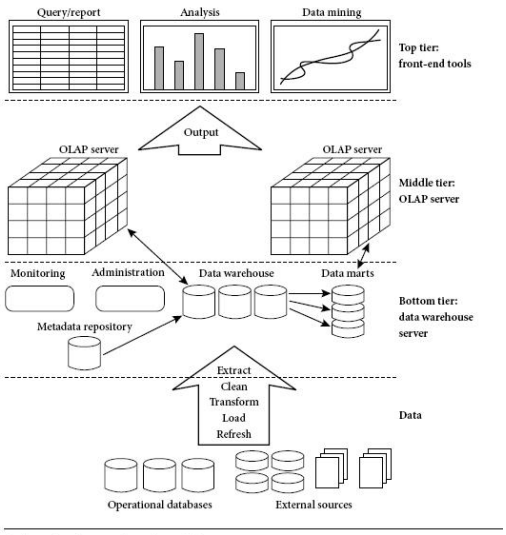


*Figure 2. A three-tier data warehousing architecture*

collection of $2^d - 1$ subcubes/cuboids that represent the aggregation of the base cuboid along one or more dimensions. Fig. 3 illustrates a five-dimensional data cube that might be associated with the meteorological environment, to analyze Rainfall along five dimensions namely Time, Gridded-Location, Low Pressure System, River catchment area and district, has been implemented [7] as shown in Fig. 3. The implementation of this hypercube is specific to Microsoft OLE DB Provider for SQL Server.

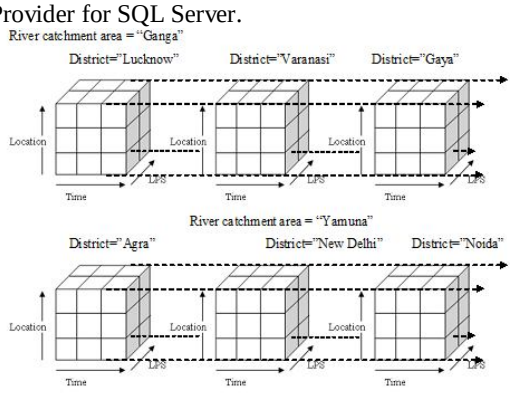


*Figure 3 A 5-D data cube representation of rainfall data, according to dimensions time, gridded-location, Low Pressure system, river catchment area and district*

Snowflake schema has been used to model these datasets, as shown in Fig. 4. The access of the fact is very fast as the internal storage of this static cube is a tree structure as shown in Fig. 5. If the total number of dimensions is n, the various unique

combinations of different dimensions are $2^n$ which is equal to the number of unique cuboids. A cuboid can be 1-D, 2-D, 3-D,….. n-D if number of dimensions is n. The time taken to access a fact across any particular combination of dimensions is $O(\log_2 2^n)$. In the particular case of 5-D data hypercube, the access time of any cuboid against any combination of dimensions has been reduced to $O(\log_2 2^5)$ i.e. at the most 5 accesses which otherwise would have been 32 if relational database management system was used.

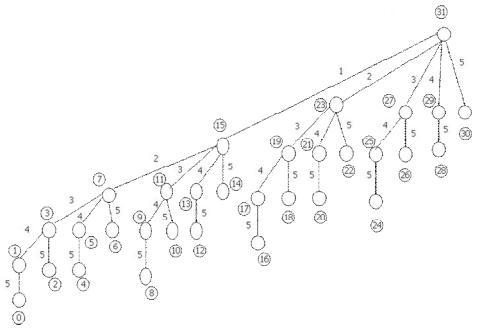

*Figure 5 Search of a node corresponding to cuboid in a 5 dimensional data hypercube*

1.3 Need of concurrent access real time OLAP

Building the data cube is a massive computational task, and significant research has been published on sequential and parallel data cube construction methods (e.g. [6], [8], [9], [10], [11], [12]). However, the traditional static data cube approach has several disadvantages. The OLAP system can only be updated periodically and in batches, e.g. once every week. Hence, latest information can not be included in the decision support process. The static data cube also requires massive amounts of memory space and leads to a duplicate data repository that is separate from the on-line transaction processing (OLTP) system of the organization. Several practitioners have therefore called for an integrated OLAP/OLTP approach with a real-time OLAP system that gets updated instantaneously as new data arrives and always provides an up-to-date data warehouse for the decision support process (e.g. [13]).

In this paper, we present a parallel concurrent access real time OLAP that is based on tree of life approach as described next.

## 2. MANAGEMENT OF BIG DATA

As stated earlier so far we are dealing with only fraction of the current data. We are soon landing into data sizes which are tending close to our concept of infinity. That has its own challenges. If we like to consider some big data, meteorological data stored by the meteorological community around the world comes out to be a great example. Meteorological communities have been archiving their data in various media & format since last over 200 hundred years. International standards came up after 1961 when World Meteorological Organization (W.M.O) established "World Weather Watch system (WWW, So term WWW was coined by WMO much before ICT community coined World Wide Web).Weather data was, and continues to be observed through various observing systems. If we consider the data available since 1960s, it would be an astronomical number. In ICT, we have been talking of kilo, mega, giga, tera, peta, exa, zeta, yotta ($10^{24}$) etc. to represent growing data & storage capabilities. But there are various systems of numeration found in various ancient Vedic literatures of India. Table 1 gives one such system used in the Valmiki Ramayana[14]. But if one considers the size of weather related data from 1960 onwards, we might need to coin a new representation.

It seems our ancestors also had thought of such problem of Big Data. Historians trace modern numerals to the Brahmi numerals, which were in use around the middle of the 3rd century BC, as discussed in [15]. The place value system, however, evolved later. As shown in table 1, the knowledge of big numbers of order of $10^{62}$ called mahaugha/ mahā-ogha (in Vedic numbering system) or one hundred novemdecillion (in Arabic number system), existed even hundreds of years ago. Such big numbers are capable of representing large and may be many different data values and needs specialized methods of storage and retrieval.

## 3. TREE OF LIFE APPROACH

To represent big data, it is interesting to fall back on our ancient philosophy. Lord Krishna has provided a way to represent whole creation, animate and inanimate, of universe. In this regards, it is interesting to refer to the Chapter XV of The Bhagavad Gita [16], there is a tree of life which is a well thought-out representation of complete knowledge of Vedas & universe. The huge data

*Table 1 Various systems of numeration*

| **Sanskrit** | **Indian figure** | **Power Notation** | Short scale (Arabic) |
|---|---|---|---|
| एक *(eka)* | 1 | 100 | one |
| दश *(dasha)* | 10 | 101 | ten |
| शत *(shata)* | 100 | 102 | hundred |
| सहस्र *(sahasra)* | 1,000 | 103 | one thousand |
| अयुत *(ayuta)* | 10,000 | 104 | ten thousand |
| लक्ष *(lakṣa)* **one lakh** | 1,00,000 | 105 | one hundred thousand |
| कोटि *(koṭi)* **one crore** | 1,00,000 śata | 107 | ten million |
| शङ्कु *(śaṅku)* | 1,00,000 koṭi | 1012 | one trillion |
| महाशङ्कु *(mahā-śaṅku)* | 1,00,000 śaṅku | 1017 | one hundredquadrillion |
| वृन्द *(vṛnda)* | 1,00,000 mahā-śaṅku | 1022 | ten sextillion |
| महावृन्द *(mahā-vṛnda)* | 1,00,000 vṛnda | 1027 | one octillion |
| पद्म *(padma)* | 1,00,000 mahā-vṛnda | 1032 | one hundrednonillion |
| महापद्म *(mahā-padma)* | 1,00,000 padma | 1037 | ten undecillion |
| खर्व *(kharva)* | 1,00,000 mahā-padma | 1042 | one tredecillion |
| महाखर्व *(mahā-kharva)* | 1,00,000 kharva | 1047 | Onehundred quattuordecillion |
| समुद्र *(samudra)* | 1,00,000 mahā-kharva | 1052 | ten sexdecillion |
| ओघ *(ogha)* | 1,00,000 samudra | 1057 | oneoctodecillion |
| महौघ *(mahaugha/mahā-ogha)* | 1,00,000 ogha | 1062 | onehundred novemdecillion |

available in modern times can also be organized and mapped to a similar kind of representation.

This Chapter explains that the wise people speak of the indestructible Peepul tree as having roots above and branches below, whose leaves are the Vedas; he who knows it is alone the Veda-knower. In the Purāna also we have "It sprouts from the root in the form of the unmanifest, it grows through the sturdiness of that very One." And it has abundance of intelligence as its trunk and the apertures of the organs as hollows. The great elements are the boughs; nourished by the Gunas; so also, it has the objects of perception as its leaves. It has virtue and vice as its beautiful flowers. This eternal tree presided over by Brahma is a means of livelihood to all creatures i.e. from the roots of the tree, energy to each and every leaf is supplied and a pipeline from roots through bough to leaves can provide nourishment to many leaves. Similarly, when we have big data i.e. leaves required to be managed, such philosophy can provide a great solution".

*Figure 6 Tree of life*

The application areas of meteorology, social networking sites, genomics [17], connectomics, complex physics simulations [18] and biological and environmental research [19] have big data that spans three dimensions viz. velocity, volume and variety. As discussed, it is a great challenge to handle these datasets using traditional data processing applications. Based on the approach followed in "Tree of Life" and handling each path from root to branch to leaf independently in a parallel manner, one can solve the problems of big data.

These days Multi-core processing is a growing industry trend as single core processors rapidly reach the physical limits of possible complexity and speed. This processor is a single computing

component with two or more independent actual central processing units (called "cores"), which are the units that read and execute program instructions as explained by [20]. The instructions are ordinary CPU instructions such as add, move data, and branch, but the multiple cores can run multiple instructions at the same time, increasing overall speed for programs amenable to parallel computing. As we have seen that the tree of life is a symbol depicting provision of nourishment to each and every path towards leaves in a parallel manner, similarly the multiple cores of these Multi-core processors can each be assigned the job of handling access towards each cuboid (leaf) of the Multi-dimensional Database model shown in Fig. 6.

We believe tree approach as depicted above may be an ideal approach to efficiently store the big data for optimum utilization for the benefit of humanity.

Appropriate strategies to organize Big Data in tree structure will lead to a new wave of fundamental technological advances that will be embodied in the next generations of Big Data management and analysis platforms, products, and systems.

## 4. AN ALGORITHM BASED ON TREE OF LIFE APPROACH

The algorithm that has been developed works on the concept of "No materialization of cuboids" and the advantage is that the real time updates are supported. The algorithm is in relation to a 5-D lattice of cuboids as in Fig. 7, but it can be extended to a large number of dimensions as large as the number of cores available in the Multi-core processor.

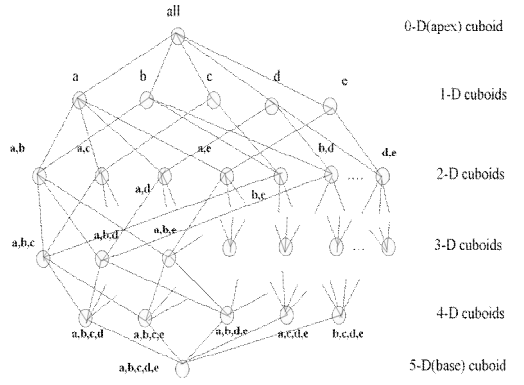


Figure 7 A 5-dimensional lattice of cuboids

It is very clear from the diagram that there are 5 dimensions viz. a,b,c,d and e. The total number of cuboids is $2^5$ i.e. 32. The description as per different dimensioned cuboids is mentioned below:-

1. The number of 0-D cuboids is ${}^5C_0 = 1$
2. The number of 1-D cuboids is ${}^5C_1 = 5$
3. The number of 2-D cuboids is ${}^5C_2 = 10$
4. The number of 3-D cuboids is ${}^5C_3 = 10$
5. The number of 4-D cuboids is ${}^5C_4 = 5$
6. The number of 5-D cuboids is ${}^5C_5 = 1$

Following conclusions can also be drawn by observing the fig. 7.

1. Each 1-D cuboid is required to use four 2-D cuboids for calculating the value of the fact across a single dimension.
2. Each 2-D cuboid is required to use three 3-D cuboids for calculating the value of the fact across the given two dimensions.
3. Each 3-D cuboid is required to use two 4-D cuboids for calculating the value of the fact across the given three dimensions.
4. Each 4-D cuboid is required to use one 5-D cuboid for calculating the value of the fact across the given four dimensions.

So, in order to make best use of the multiple cores of the CPU, an algorithm has been developed that is based on these conclusions and uses multiple cores in order to calculate the fact across any number of dimensions in real time i.e. there is no materialization of cuboids.

4.1 About Algorithm

The variables used by the algorithm that follows are explained below:-

1. Let n denote the total number of dimensions of the Multidimensional Data Cube.
2. For a multidimensional query that uses 2 dimensions in its group by expression i.e. it is a 2-D cuboid, so let d denote the number of dimensions in the cuboid.

**Algorithm : Calculatefact(n,d)**

1. q = d
2. Repeat
3. Initiate a new thread
4. Num = n-q // number of cuboids required to calculate the fact
5. Dim = q+1 // dimensions of the cuboids mentioned in point 4.
6. Initiate Num threads to calculate the required cuboids
7. q = q + 1
8. Until (q=n)

Hence, for each lower numbered dimension cuboid, higher numbered dimension cuboids are to be

called with the initiation of new thread that uses a new core, which is the essence of the above algorithm.

## 5. DISCUSSIONS AND CONCLUSIONS

As we have seen from this algorithm that each cuboid uses the fact values from the cuboids below it, this approach is clearly indicating parallel concurrent calculations handled by multiple cores. This is same as the tree of life approach which is a symbol depicting provision of nourishment to each and every path towards leaves in a parallel manner. Although, for large data warehouses, pre-computed cuboids still outperform real-time data structures but of course with the major disadvantage of not allowing real-time updates. The main contribution of this paper is the design of a parallel tree using the new parallel access method that has the potential to enable OLAP systems that are real-time and efficient for large databases.

We believe tree approach as depicted above may be an ideal approach to efficiently retrieve the big data for optimum utilization for the benefit of humanity. The recent invention Tianhe-2, developed by China's National University of Defense Technology, the world's fastest supercomputer [21] has 32,000 Ivy Bridge Xeon CPUs and 48,000 Xeon Phi accelerator boards for a total of 3,120,000 compute cores, which are decked out with 1.4 petabytes of RAM. Such powerful machines can certainly compute the fact across multiple dimensions in fractions of a second.

## 6. Acknowledgements

The author wishes to place on record hearty thanks and gratitude to Prof. (Dr.) Rattan K. Datta, Former Advisor, Department of Science and Technology, for his continuous guidance and support for this research.

## REFERENCES

[1] IBM website, Available from: http://www-01.ibm.com/software/in/data/bigdata/

[2] About Big Data, Available from: Tavo De León BigDataArchitecture.com

[3] About GRIB standard FM 92 GRIB , I.2-GRIB Reg. - 1 (Edition 2 - Version 2 - 05/11/2003) Available from: http://www.wmo.int/pages/prog/www/DPS/FM92-GRIB2-11-2003.pdf

[4] J. Han, M. Kamber, Data Mining Concepts and techniques, Morgan Kaufmann Publisher, 2006

[5] A. Berson, S.J. Smith Data Warehousing, Data Mining and OLAP, Tata McGraw-Hill Publishing Company Limited, New Delhi, 2005

[6] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Cube: A Relational Aggregation Operator Generalizing Group-By,Cross-Tab, and Sub-Totals, Data Min. Know. Disc., vol. 1,pp. 29–53, 1997.

[7] K. Pabreja, Mapping of spatio-temporal relational databases onto a multidimensional data hypercube, Proceedings of Einblick – Research Paper Competition held during Confluence 2010 organized by Amity University in association with EMC data storage systems (India) Pvt. Ltd., Noida, UP, India, 2010, Jan. 22-23, 127-133.

[8] Y. Chen, F. Dehne, T. Eavis, and A. Rau-Chaplin, PnP: sequential, external memory, and parallel iceberg cube computation, Distributed and Parallel Databases, vol. 23,no. 2, pp. 99–126, Jan. 2008. Available from: http://www.springerlink.com/index/10.1007/s10619-007-7023-y

[9] F. Dehne, T. Eavis, and S. Hambrusch, Parallelizing the data cube, Distributed and Parallel Databases, vol. 11, pp. 181–201, 2002. Available from: http://www.springerlink.com/index/BGN4YJUMUBPELXK0.pdf

[10] Z. Guo-Liang, C. Hong, L. Cui-Ping, W. Shan, and Z. Tao, Parallel Data Cube Computation on Graphic Processing Units, Chines Journal of Computers, vol. 33, no. 10, pp. 1788–1798, 2010. Available from: http://cjc.ict.ac.cn/eng/qwjse/view.asp?id=3197

[11] R. T. Ng, A. Wagner, and Y. Yin, Iceberg-cube computation with PC clusters, ACM SIGMOD, vol. 30, no. 2, pp. 25–36, Jun. 2001. Available from: http://portal.acm.org/citation.cfm?doid=376284.375666

[12] J. You, J. Xi, P. Zhang, and H. Chen, A Parallel Algorithm for Closed Cube Computation, IEEE/ACIS International Conference on Computer and Information Science, pp.95–99, May 2008. Available from: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4529804

[13] R. Bruckner, B. List, and J. Schiefer, Striving towards near real-time data integration for data warehouses, DaWaK, vol.LNCS 2454, pp. 173–182, 2002. Available from: http://www.springerlink.com/index/G5T567NVR9AA96XQ.pdf

[14] Valmiki Ramayana Yuddha Kanda

[15] J. John O'Connor and Edmund F Robertson, Indian numerals, The MacTutor History of Mathematics archive, November 2000

[16] Srimad Bhagavad Gita, Ved Vyasa, Chapter XV, The Supreme Self

[17] Community cleverness required, Nature, International Weekly journal of Science, 455 (7209):1, 4September 2008. doi:10.1038/455001a.

[18] Sandia sees data management challenges spiral, HPC Projects, 4 August 2009. Available from: http://www.scientific-computing.com/news/news_story.php?news_id=922

[19] O.J. Reichman, M.B. Jones, M.P. Schildhauer, Challenges and Opportunities of Open Data in Ecology, Science 331 (6018): 703–5. doi:10.1126/science.1197962.

[20] M. Rouse, Definition: multi-core processor, TechTarget, March, 2007, Retrieved March 6, 2013.

[21] About Tianhe-2, Available from: http://www.top500.org/blog/lists/2013/06/press-release/