

Architecture for Analysis of Multi-Dimensional Data using Standalone Desktop based Cluster

Suja P Mathews¹, Sunu George²

¹Department of Computer Science
Jyoti Nivas College, Bangalore, India
suja4sam@gmail.com

²Department of Computer Science
CJM Assumption HSS, Varandarappilly, India
sunugeorge1976@gmail.com

Abstract: *The critical parameters of multi-dimensional data analysis are its data size and the number of computations. The data size is typically large and the numbers of computations are immense. The complexity increases further when the dimensions and the resulting intersection space are larger. The processing involves data access, required computations and aggregations considering all dimensions with the multi-level hierarchical grouping. The large amount of data and resulting computations require huge processing power and memory, which are usually found in Symmetric Multi-Processing (SMP) machines. Here we present architecture to analyze multidimensional data using a cluster of regular desktop computers which do not share anything between them. All the tasks like building the multidimensional cube, storage of data, data query and the computations are distributed across the standalone nodes of the cluster. Thus, we can achieve infinite scalability and excellent performance.*

Keywords: Multidimensional data analysis, Messaging based IPC, Desktop Cluster

1. Introduction

Multi-dimensional analysis is used for decision support systems and statistical inference to find interesting information [7] from large database. Multidimensional databases [6] are suitable for OLAP [4] [5] and data mining [3] since these applications require dimension-oriented operations on data. Typically, multidimensional arrays are used by the traditional multidimensional databases [8] for data storage and for performing analytical operations. The above storage method is good for dense data. But most data sets are sparse for most practical scenarios. This warrants other efficient storage schemes.

The huge data size and immense computation are the main challenges of multidimensional data analysis [9]. When more dimensions and the corresponding intersection space are put into the context, the complexity becomes much more complex. All the data need to be accessible, should be usable in computation and should be aggregated with all the applicable dimensions and with multi-level hierarchical grouping. In symmetric multi-processing computers we can find the required huge processing power and memory requirement. Here we present Architecture for the analysis of multidimensional data [10] using shared nothing-desktop server [1]. The architecture enables running of applications on independent desktop servers, sharing nothing between them, and forming a cluster environment. All the core processor and memory intensive operations like multi-dimensional cube building, data storage, data query and the required computations are distributed across the cluster nodes which share nothing among them.

2. Multi-Dimensional Aggregation OLAP Server

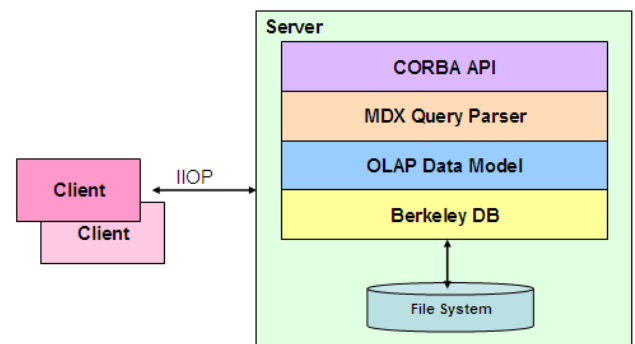


Figure 1: Multi-Dimensional Aggregation Architecture

The main challenge of any multi-dimensional aggregation server is the data volume involved and the required processing power and memory requirement. Modern multi-processor computers can address this challenge but the costs of such machines are exorbitant.

Another economical way to address this scalability issue is to make use of the processing power and memory of commodity desktop machines. Since these machines are independent machines (shared nothing), harnessing their capability is a big challenge. By making use of Message Passing Interface (MPI) we can build parallel applications which can run on many shared nothing desktop computers which are connected by normal network.

3. Proposed Distributed Architecture

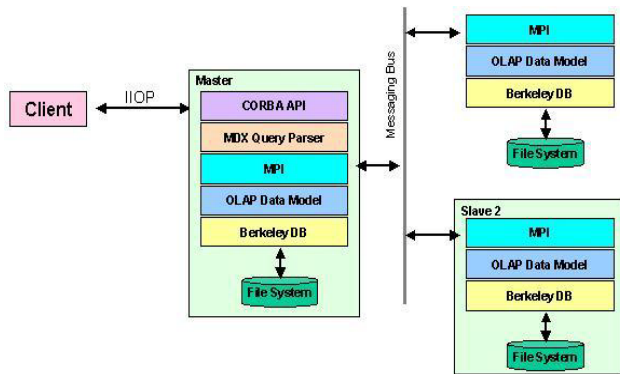


Figure 2: Distributed Architecture for Multidimensional OLAP Server

Here we are presenting a parallel architecture for Multi-Dimensional Aggregation Server by using Message Passing Interface.

The proposed architecture has the following properties:

- The architecture is based on M-OLAP (multi-dimensional) engine.
- Since aggregate data is neither stored, nor pre-computed, it can handle data explosion.
- It should be able to handle sparsity very well.
- It has low memory foot print.
- It aggregates data at query time.
- Cube and dimension management can be supported using SQL-like syntax
- Basic MDX (Multi-Dimensional Expression) [2] is supported for query
- Server is a parallel engine, and uses MPI middleware.
- Multi-user environment is supported
- Uses client-server architecture
- Data management is done using embedded Berkeley DB, which ensures high efficiency.
- Commands can be executed using a command line utility
- Applications can be integrated using APIs provided.

4. Distributed Strategy

As indicated elsewhere in this paper, the major challenge in multi-dimensional data handling and processing is its huge data size and large number of computations. The complexity grows exponentially as the dimensions and the resulting intersection space increase. The tasks that are involved are data access, computations, and data aggregation, along with all dimensions with their multi-level hierarchical grouping. The mentioned requirements ask for large computing power and memory availability. These are found in Symmetric Multi-Processing (SMP) computers.

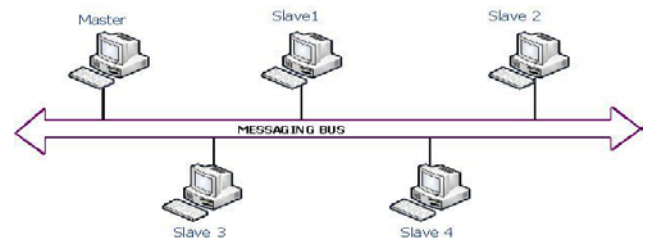


Figure 3: Message Passing Interface (MPI) -Distributed Architecture

Parallel computers, for e.g. Scalable Parallel Computers use Message Passing as their programming paradigm. They have distributed memory and run as a cluster of workstations. The basic concept of Inter-process communications using messages is well understood, though there are variations in this methodology. Several applications have been designed and implemented using this paradigm, though the variants vary with vendors. It has been demonstrated that a highly portable and efficient system can be implemented using message passing paradigm.

SMP computers have few disadvantages:

- High cost per computation
- Scalability is difficult and expensive

SMP has certain advantages as well:

SMP supports accessing shared memory for the parallel processors. This results in lower latency and thus better performance.

This architecture allows applications to run on fully independent desktop servers which form a cluster environment. These desktop servers share nothing between them. Building of multi-dimensional cube, storage of data, querying of data and computations are distributed across the shared-nothing independent desktop cluster servers (nodes). This gives large scalability and performance.

The issues of network latency introduced due to interconnection between the nodes in the above approach can be resolved by the approaches explained below:

- Minimize data communication between the cluster nodes through a distribution strategy
- Implement concurrent file access and data partitioning.
- Utilize asynchronous disk I/O, which are concurrent with computations. This allows hiding the latency.
- Shared Nothing Architecture

In Shared-Nothing (SN) Architecture, every node of the distributed computing system is a fully independent and self-sufficient computer. There is no single point of contention in the complete distributed system. This is in contrast to systems which centrally store large amount of state information in a database, application server or any

other single point of contention. The SN concept is essentially based on the concept of web.

In a web environment, the whole architecture is based on SN, due to its scalability. We can simply add nodes – inexpensive standalone computers - to a pure SN system to achieve infinite scalability. In such a architecture, there is no single bottleneck to slow down the system. The data is partitioned among many nodes, assigning different computers to deal with different users and queries. It may also require that every node maintains its own copy of the application data, and co-ordinate between nodes using a protocol.

5. Multi-Dimensional Aggregation Engine

The architecture is for multi-dimensional data aggregation using Bit Encoded Storage Scheme (BESS) which addresses sparsity and data explosion in an efficient manner. Sparse data is stored in chunks using a data structure which use bit encodings for dimension indices. This structure is called Bit Encoded Sparse Structure. This supports fast OLAP query operations on sparse data utilizing bit operations. This does not require exploding the sparse data into multi-dimensional array. This allows for high dimensionality and large dimension sizes. Sparsity is handled by using compressed chunks using a bit encoded sparse structure (BESS). Data is read from a relational data warehouse which provides a set of tuples in the desired number of dimensions from which data cube is computed.

The architecture makes use of MOLAP concepts but does not store the data in memory which causes large memory foot print. Data management is achieved by Berkeley DB database library which helps in storing, retrieving and updating the data using file system. Berkeley DB is a library that links directly into your application. Your application makes simple function calls, rather than sending messages to a remote server, eliminating the performance penalty of client-server architectures. Berkeley DB stores data in application native format, as simple key/value pairs, eliminating the need for translation or mapping.

Multi-Dimensional Expression (MDX) interface is proposed to enable multi-dimensional querying of data. Multidimensional queries can be described using this MDX language. The term MDX stands for “Multi-Dimensional extensions. This is an extension of SQL, and thus its structure is similar to SQL, and the same keywords serve similar functions. The key difference is that MDX builds multi-dimensional view of the data while the SQL builds a relational view of the data.

The aggregation services are exposed as interfaces to enable Client Server architecture. CORBA is used as middle ware for exposing the interfaces. CORBA enables language independence through the concept of language mapping.

6. Conclusion

The objectives of providing scalable Online Analytical Processing capabilities like aggregation, dicing and slicing of multidimensional data is achieved using the proposed architecture. It enables to build parallel software which can run on economical commodity desktop machines without any specialized hardware. These machines are deployed in a cluster environment which makes it highly scalable and economical. Maintenance and upgradability are easily achieved without impacting system availability.

This architecture is the parallel daemon software which runs across multiple shared nothing desktop servers (or on SMPs) and executes the commands. Web Service is the future implementation possibility and will be using client SDK to interact with server.

7. Acknowledgement

If words are considered as symbols of approval and tokens of Acknowledgement, then let the words play the heralding role of expressing my gratitude. First and foremost, we are thankful to the Almighty, for his blessings in the successful completion of this paper. Our heart full thanks to Mr. Radhakrishnan Maniyani, Chief Technology Officer, True Influence for his help for the successful completion of this paper. We would like to express our sincere and hearty thanks to Mr. M J Samuel, M.S BITS Pilani, Engineering Manager, Celstream Technologies, for his splendid help for the successful completion of our paper.

References

- [1] George Spofford, Sivakumar Harinath, Chris Webb, Dylan Hai Huang, Francesco Civardi: MDX-Solutions: With Microsoft SQL Server Analysis Services 2005 and Hyperion Essbase. Wiley, 2006, ISBN 0-471-74808-0
- [2] Mosha Pasumansky, Mark Whitehorn, Rob Zare: Fast Track to MDX. ISBN 1-84628-174-1
- [3] Jiawei Han and Micheline Kamber: Data Mining Concepts and Techniques.
- [4] Surajit Choudhary: Data Warehousing and OLAP technology.
- [5] Umeshwar Dayal: An overview of data warehousing and technology
- [6] S. Agrawal. On the Computation of Multidimensional Aggregates. Proceeding Very Large Data Base Conference. Morgan Kaufmann, San Francisco. 1996, pp. 506-512.
- [7] "Pan for Gold in the Clickstream," Information Week, March 12, 2001
- [8] White paper: "Building Profitable Customer Relationships with Data Mining," in PDF format (156 KB)
- [9] "Mining Large Databases -- A Case Study" (written for IBM), in PDF format (250 KB)
- [10] Clifton, Christopher (2010). "Encyclopedia Britannica: Definition of Data Mining".

Author Profile

Suja P Mathews completed MCA in 2000 and M.Phil in Computer Science in 2006 from Madurai Kamaraj University. He started the career as a lecturer in Computer Science Department in SCT Institute of Technology, Bangalore in 2005. Promoted as the Head of the Department in the same Institution and continued for 3 years. Joined St.Joseph's Arts & Science College, Bangalore in 2010 & worked for 2 years. He is currently working as Associate professor in Jyoti Nivas College, Bangalore.

Sunu George received MCA and M.Phil in Computer Science from Madurai Kamaraj University in 2000 and 2006 respectively. In year 2002, was appointed as a Higher Secondary Teacher in St. Thomas College HSS Thrissur, a reputed and Government Aided institution under Corporate Educational Agency of Arch Diocese of Thrissur, Kerala. Currently working as Senior Teacher in Assumption HSS Varandarappilly, Thrissur, another institution under same Management.