A New Framework for High Performance Processing of Voluminous Multisource Datasets

Rania M. Kilany 1, Reda. Ammar 1, S. Rajasekaran 1 and Wala. Sheta 2,
rania.kilany@engr.uconn.edu, reda@engr.uconn.edu, rajasek@engr.uconn.edu, wsheta@mucsat.sci.eg.
1 Computer Science & Engineering Dept,The University of Connecticut, Storrs, Connecticut 06269,USA
2 Mubarak City for Scientific Applications and Technology, Borg Elarab, Alexandria, Egypt

Abstract
In this paper we present a new framework to process high-volumes of data generated from heterogeneous sources with different formats (text, image's features …etc.). The framework consists of three phases. The first phase selects appropriate data reduction technique that closely preserves all of the relevant information in the original data set. The second phase determines the suitable algorithm to apply the selected data reduction technique. The third phase integrates the reduced datasets and makes it ready to fit into different models (Visualization, Reports, Decision making, and predictions). This framework is ideal for knowledge management of data-intensive applications.

Keywords: High Performance processing, Data Mining, Data Reduction.

1. Introduction

Discovery and learning require the detection and identification of novel and important phenomena from voluminous datasets followed by a detailed analysis to measure and verifies the significance of the phenomena. The sheer volume of data involved often requires enormous amounts of computational resources, and processing it in real-time, may be unattainable in practice. Hence researchers are facing a new class of applications that require innovative approaches for storing, retrieving, processing and manipulating the available datasets.

Right now, each research community that has voluminous data to deal with uses a set of techniques that are intuitively selected with no scientific basis. For instance, association-rules mining algorithms are popular in dealing with market data. Internet data are processed using graph analysis (for example, to rank different nodes). Sequence analysis tools are employed for biological data. Statistical techniques, such as Bayesian, are used for atmospheric data. Existing techniques suffer from many drawbacks. Several important drawbacks of selecting algorithms in an ad hoc manner are: (1) they are slow; most of the applications of interest to the society (such as interactive visualization, medicine, weather forecasting, fraud detection, etc.) need real-time or very nearly real-time performance; (2) they do not necessarily generate the most accurate results; and (3) they are limited to one source of data and/or one specific data format.

2. Proposed Framework

In this paper we propose a new methodology for high performance processing voluminous datasets that are generated from multiple sources and may have different formats. (Figure 1) describes the proposed methodology that consists of the following phases:

1. The first phase pertains to employing data reduction techniques that closely preserve all of the relevant information in the original data sets. In an earlier study, we have broadly categorized data reduction techniques into two [1, 2, 3], namely those that reduce the number of points in the input and those that reduce the underlying dimension of the input points. One of major challenges is assessing available data reduction techniques and developing an algorithm to map a given dataset to the appropriate subset of these techniques. The selection depends on a set of qualitative and quantitative metrics (such as suitability, accuracy, and data reduction ability) to assess the coupling between the source and format of a dataset with each data reduction technique.
2. Each data reduction technique may have a number of algorithms that are either sequential or parallel. The question that always remains is which algorithm should be applied to each dataset. Should we execute the algorithm on a sequential machine or a multiprocessor system? In this phase, existing algorithms should be assessed to identify their advantages, special features, and shortcomings. Evaluation metrics include suitability to the data format, scalability, accuracy and performance (execution time, used resources, etc). These metrics are then composed into a weighted objective function(s) that guides mapping every dataset to the best matching algorithm.
3. The integration of the most efficient data reduction techniques/algorithms selected for each given dataset will form a hybrid approach to generate a compact unified reduced dataset that keeps the relevant information of the given application. The integration may be either architecture-based or semantic-based. In either case, the outcomes should be represented in a unified format for fast storing and retrieving. The reduced data can then be archived in a database, retrieved, processed and visualized as need.

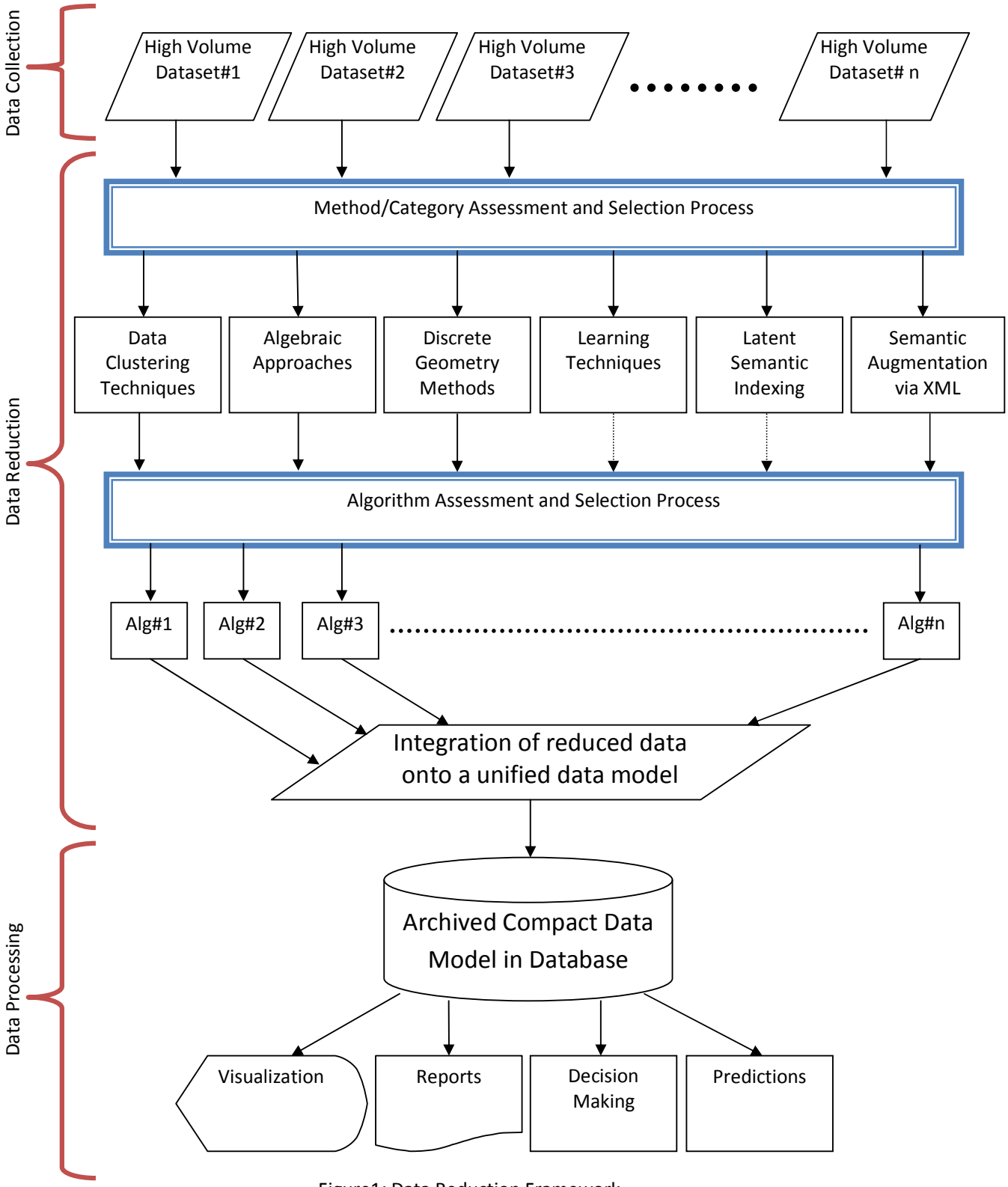

Figure1: Data Reduction Framework

In next sections, we describe different data reduction techniques, an assessment procedure of their algorithms and different methods to integrate the reduced datasets into a unified data model.

3. Phase I: Data Reduction Techniques

The amount of data generated in many of today's applications is voluminous. In order to process them in real time, we propose to employ various data reduction techniques [5-30]. Any data reduction technique reduces the amount of data drastically preserving the information content as much as possible. Popular data reduction techniques include Data Clustering, Algebraic Approaches, Discrete Geometry Methods, Learning Techniques, Latent Semantic Indexing, and Semantic Augmentation via XML. In our past work we have worked on nearly all of these techniques. For example, we have devised the best known approximation algorithm for the k-medians clustering problem [5] and the best known parallel algorithm for hierarchical clustering [6]. We have worked on PAC learning [25], the light bulb problem, etc. In this paper we investigate appropriate data reduction techniques for each data type/format. A data reduction technique that is suitable for images may be different from those that can be employed to text data. Furthermore, some of these techniques may require long processing time for voluminous data. Hence a part of this work is to find the highest performing approach for each data type if there is more than one technique. If necessary we will utilize supercomputers to implement the selected data reduction algorithm. Our prior experience in this area will prove very valuable. Next we briefly summarize six approaches to data reduction.

1. Data Clustering:

Clustering, also called unsupervised classification, is to partition a given set of data points into groups where each group has similar points [2, 7-11]. In general, clustering approaches can be grouped into five categories: *partitioning clustering, hierarchical clustering, density-based clustering, model-based clustering and fuzzy clustering.* In Partitioning Clustering, data is partitioned into several clusters such that objects in a cluster are more similar to each other than to objects in other clusters. This is achieved by minimizing an objective function iteratively. k-means and k-medoids are two representatives for this form of clustering. In k-means problem, given a set $P \subset Rd$ of n data points and a number k, we try to partition P into k subsets (clusters). Each such cluster has a center defined by the centroid (i.e., mean) of the points in the cluster. The clustering should minimize $\sum_{x \in P} \|x - K(x)\|^2$, where K(x) denotes the nearest center to the point x. Hierarchical Clustering builds a dendrogram. This allows exploring different granularity levels of the data set. The running time is O(n2). It is grouped into agglomerative clustering and divisive clustering. Agglomerative clustering proceeds in a bottom-up fashion while divisive clustering uses a top-down approach. For instance, agglomerative clustering starts with single point clusters and recursively merges the most similar two clusters until the requested number of clusters is reached. In Density-Based Clustering, a cluster is defined as a connected dense component. This clustering can produce arbitrary shape clusters and has protection against outliers. Its disadvantage is that adjacent clusters with different densities but bigger than a threshold cannot be separated. Model Based Clustering and Fuzzy Clustering are also in use.

2. Algebraic Approaches:

Algebraic approaches [12, 13] have made an impressive contribution to information retrieval and space reduction research. Some of the algebraic space reduction approaches are Singular Value Decomposition (SVD), Discrete Fourier Transform (DFT), Discrete wavelet Transform (DWT), Piecewise Aggregate Approximation (PAA) and Adaptive Piecewise Constant Approximation (APCA). Here we briefly summarize one of these techniques.

The basic idea of SVD is to reduce the original N-dimensional data to a k-dimensional subspace through the origin, where ($k < N$). Using the entire data, the SVD-transform matrix is chosen. However, to apply the reduction, the first k-dimensions are chosen as they contain most of the information. The reduction is done, by applying the transform matrix on each individual vector of the coordinates in the first k dimensions. If there is a set X of N-dimensional vectors, to reduce the dimension of the data set from N to k the ($N - k$) non-significant singular values of X are eliminated. SVD requires very heavy computational effort. On the other hand, SVD achieves quite high a precision compared to other dimensionality reduction transforms. The main disadvantage of SVD is the long running time. However, there have been some attempts to parallelize the algorithm. In our prior work [13] we have developed a parallel algorithm for SVDs that outperforms an algorithm developed at MIT.

3. Discrete Geometry Methods:

Discrete Geometry Methods [14-20] have recently emerged as a powerful approach for dimensionality reduction. Examples include Random Discrete Geometry Methods such as Random Projections (RP), Fast Map, Metric Map, Boost Map and Locally Linear Embedding.

Johnson-Lindenstrauss lemma [17] has laid the foundation for Random Projections. Random Projections aim to project the original n-dimensional data into a k-dimensional subspace, where (k < n). Basically, a random matrix R of size k × n whose columns have unit lengths is employed to achieve this projection. Choosing the random Matrix R is a challenge. The power of Random Projection comes from preserving distances closely. Moreover, RP doesn't require heavy computational efforts. Another approach is introduced by Achlioptas [14] who replaces the Gaussian distribution that is normally employed to form the elements of R, with a much simpler distribution. The computations in this scheme can be performed using integer arithmetic. Random projections are used as a preprocessing stage before data mining, image processing and clustering algorithms.

4. Learning Techniques:

Numerous learning techniques such as neural networks [21] and probably approximately correct learning [22-23] can be found in the literature. Learning techniques are relevant in the context of data reductions for the following reason. As a simple example, we could devise learning algorithms for classifying (and clustering) data. Representative(s) from each cluster can then be used (instead of the original input set) for processing. In this section we briefly describe neural networks, probably approximately correct learning, and Bayesian networks.

Neural networks have been devised getting the inspiration from how neurons function in the human brain. Each neuron can be thought of as a simple processing element. Millions of neurons work cohesively, with the help of the interconnection network, to produce impressive results. A neural network is a connected leveled graph where each node corresponds to a (simple) processing element and the (directed) edges correspond to communication links.

Probably Approximately Correct (PAC) learning can be described as follows [23]: If C is any concept that we are interested in learning and if G is the concept that has been learnt, we define the error in learning, error(G) as the probability that $C(x) \neq G(x)$ for an arbitrary element x of the universe under concern. For example, if C is a Boolean formula on n variables, one way of specifying C is with the set C' of satisfying assignments to C. The distance between C and G (or error(G)) can be defined as $\frac{|C'-G'|+|G'-C'|}{2^n}$. We say a learning algorithm is capable of learning a concept C probably approximately with parameters ε and δ if the probability that error(G) is greater than ε is at most δ. Here ε is known as the accuracy parameter and δ is called the confidence. These parameters can either be user-specified or set to default values. In our past work we have developed efficient PAC algorithms in the context of designing bikes for the physically challenged [25].

5. Latent Semantic Indexing:

It assumes that in any document, there is an underlying semantic structure involving the "words" of the text, and that this structure, can be captured and described, allowing the resulting indexed document to be searched and queried. In a given application, it is clear that having information reduced (using one of the previous techniques) and then indexed via LSI techniques (with a minimum of information loss), we can greatly diminish the processing time that is required to access the documents while still retaining the meaning.

The classical LSI approach, Latent Semantic Analysis (LSA) [26], is a method which is based on the singular value decomposition (SVD) algorithm. The premise is that there is a collection of documents that need to be indexed, for which there is an associated set of terms to be found. To accomplish this, for each document, there can be a vector (of length the number of terms to be indexed), with the vector entry indicating its absence (value is 0) or its presence (frequency – value greater than or equal to 1). Collectively, for all the documents that are indexed, there is a large term-by-document matrix X, with each position xij corresponding to the term (row i) in a document (column j). LSI works by making the assumption that there exists an underlying semantic structure of the use of words throughout the collection of documents. By doing so, the resulting document space that is represented by the matrix X can be reduced (via SVD) to a smaller space approximated by the matrix X', which has a lower rank k. The value k represents a threshold that is used to maintain the most significant structural aspects of the document collection while still excluding noise or trivial values as needed to improve retrieval performance. To augment classical LSI, Probabilistic Latent Semantic Indexing (PLSI) [27] employs a statistical model that targets domain-specific synonymy and polysemy. The intent of this model is to more precisely characterize the content of the documents (based on the indexes), but, as claimed by [27], more robust and achieves better precision over the classical LSI method.

In addition to these works, the Unitary Operators for Fast Latent Semantic Indexing (UOFLSI) [28] reduces the computation cost in SVD. UOFLSI utilizes a unitary transformation, which is memory efficient, and can be computed in linear to sub-linear time. The claim is that UOFLSI

can preserve the cohesive nature of the document content and reduce the dimension of the document content, with less computation. A second approach, Polynomial Filtering for Latent Semantic Indexing (PFLSI) [29], is a framework for LSI that utilizes polynomial filtering to assist in the calculation of the vector and matrix content. In this case, the claim is that matrix decomposition and its computational cost and storage requirements are substantially less as compared to traditional implementations of LSI. Clearly, UOFLSI and PFLSI, represent newer generations of LSI, which may enhance the document search process.

Lastly, distributed LSI seeks to address issues related to scalability to a more realistic environment as the quantity of documents increases – while still attempting to maintain quality of returned documents [30]. The objective is to have a better match between the user's query (and its meaning) and the document collection. Distributed LSI (DLSI) addresses scalability by partitioning information sources with respect to different conceptual domains (e.g., counter-intelligence, intercepted communications, terrorist activity, etc.), indexing each derived subcollection with LSI. Queries can then be performed over domains or the entire space, depending on the scope and breadth of the desired results. For counter-terrorism purposes, the partitioning may improve performance, and allow more focused queries to be posed and answered.

In summary, as a representation/search technology, LSI has the potential to represent an impressive technique for unstructured data which can also provide the means for semantic indexing. As such, there are many different applications of LSI, including IR, feedback on document relevance, archivist's assistant, automated writing assessment, textural coherence, information filtering, cross-language retrieval, etc.; these have been touted by many of the papers cited throughout this section. In addition, it is hoped that LSI can provide the means for queries that attain a higher precision (polysemy) and recall (synonymy) of the terms and their contexts (documents).

6. Semantic Augmentation via XML

The Extensible Markup Language (XML) has emerged as a standard for information exchange in many different settings, including: web-based applications, database interoperability, common software tool formats, etc. In a web-based setting, XML is the predecessor to HTML to allow information content to be hierarchically organized and tagged to highlight important and relevant content. The tags are intended to capture not only the content, but for our purposes, to try to represent the meaning of the information (semantics). There has been an emphasis on XML extensions for information retrieval (IR). Namely, the Initiative for the evaluation of XML Retrieval (INEX), established in April 2002, supports research within the XML retrieval community for large-scale evaluation of content-based access methods to XML. INEX supports both content-only and content-and-structure queries, to provide a more sophisticated means to find information. These queries operate over data of various times, including: plain text, numbers, date and time, etc. An XML document can have tags that are semantic (represent what the data means) and non-semantic (present the data content), and both of these situations must be supported.

XML can be combined with LSI, allowing document searches that transcend syntax to include semantic content. The conventional techniques used for information retrieval systems include stop lists, word stems, and frequency tables. Many of the LSI techniques maintain only the "most significant" content of a frequency table, which contains the frequency of terms in a document collection. There are a number of ways XML can be utilized to supplement LSI. First, suppose that XML documents are to be indexed via LSI. In this case, the LSI techniques can be augmented and expanded to leverage the content of the XML document (the tags and the meaning of each tag). If the XML document has more content, then the resulting indexing via LSI may yield a more accurate and robust frequency table, and may allow polysemy and synonymy to be addressed. Second, from an implementation perspective, perhaps the frequency table (or similarity matrix) used by the various LSI techniques can be stored as XML documents. The tags can be used to represent the indexes, and perhaps there can be an XML frequency table for each document; combining multiple documents would mean merging their respective XML frequency tables.

In summary, there are six different categories of data reduction techniques. Each one of these works differently and hence it may be amenable to a specific data source and/or format. The main challenge in this phase is selecting the most appropriate data reduction technique for a given dataset. Currently researchers use the semantics of the given dataset and/or the data format to make their selection. For example: association-rules mining algorithms are popular in dealing with market data. Internet data are processed using graph analysis (for example, to rank different nodes). Sequence analysis tools are employed for biological data. Statistical techniques, such as Bayesian, are used for atmospheric data. Our framework will motivate researchers to develop additional metrics

to map a dataset to the most appropriate data reduction category.

4. Phase II: Selecting the data reduction algorithm

As described above, every data reduction category may have several algorithms to implement its technique. The challenge is which one of these algorithms is the most efficient to execute the selected data reduction technique. This requires assessing these algorithms analytically to generate their different performance metrics as a function of the dataset's parameters. This assessment process can be implemented using the Hierarchical Performance Analysis (HPM) methodology (Figure 2). HPM was introduced over a decade ago [31], and numerous approaches have subsequently been reported [32, 33]. The hierarchical approach [34] provides an integrated scheme for interrelating the numerous variables in a composite modeling framework, and encapsulates the layers of the model to focus upon specific functional regions of an algorithm. The layers of modeling involve the architectural level as a foundation for determining the starting behavior of a software routine's execution, progress upward through the multiple layers of the software subroutines calls, through the tightly coupled multiple processor activities (in case of parallel algorithms), and finally over a loosely coupled network which forms the cluster/grid communications network. Each layer's model inherits the performance results of the lower, related layers as input properties for the model of the focus layer. In summary, HPM [34] provides a degree of accuracy that cannot be achieved with single layer models. Thus, quantitative performance assessment of an algorithm comprising of hardware, software and communication is provided. Such information provides the necessary bases for selecting the most efficient algorithm of a given dataset.

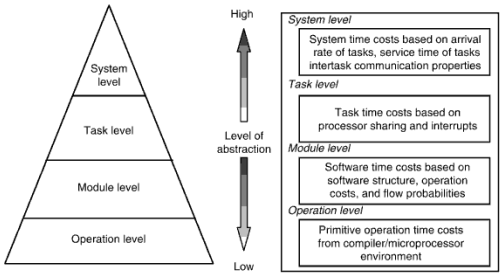

Figure 2: Hierarchical Performance Model

Figure 3 shows a detailed example for How the HPM work, In the *a) System Level* queuing networks can be used to model the behavior of the system, it concerned with input arrivals and the interaction between software processes. *b) Task level* models the interaction between software modules executing concurrently, the output of this level will be the communication costs (between processors) and interrupt delays. *Module level* calculates time-cost expressions for components of software (procedures and functions). *Operation level* provides time-cost measurements for primitive operations, built-in functions, function calls, and argument passing (dataset size, machine configurations); it determined the interactions of the primitive computer instructions with the underlying processor architecture.

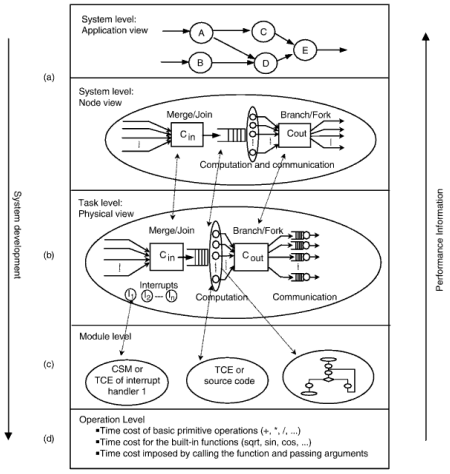

Figure 3: Detailed view of Hierarchical Performance Model

5. Phase III: Data Integration

Data integration is the process of combining data generated from different data reduction algorithms and providing a unified view of these datasets. There are several approaches to implement this phase [35-38]. These approaches can be classified into two categories: architecture integration or semantic integration. One of the architecture integration approaches is providing a uniform query interface over a mediated schema. Each reduced dataset is considered as a different view over the mediated schema. The information from the reduced datasets is extracted, transformed and then loaded into the database. To access the information, a given query is compiled into specialized queries over the original datasets. An alternate model of architecture integration is one where the mediated schema is designed to be a view over the reduced datasets. However, architecture integration has a drawback. The view for mediated schema needs to be rewritten whenever a new dataset is to be integrated and/or an existing dataset changes. On the other hand, semantic integration focuses on solving semantic conflicts. A common strategy for the resolution of such problems is the use of ontology which explicitly defines schema terms and thus helps to resolve semantic conflicts. The most appropriate approach, functional and performance wise, to integrate the reduced datasets to form a unified compact data model depends on the application, type of data, required processing functions and their desired performance.

6. Conclusion

In summary we have briefly described three phases for high performance processing of voluminous multi-sources datasets. Data reduction starting from data collection considering different data formats, in phase I, we map a given dataset to the most suitable data reduction category or technique (six different categories of data reduction techniques were described). In phase II, we analytically assess different algorithms that implement the selected data reduction technique. HPM will be utilized to evaluate the performance of each algorithm and compare their merits. Finally reduced datasets are integrated and then archived in a compact database in a way that makes it easy to be retrieved, processed and visualized as needed. This framework is ideal for knowledge management of data-intensive applications.

References

[1] R. Ammar, S. Demurjian, I. Greenshields, K. Pattipati, and S. Rajasekaran, "Analysis of Heterogeneous Data in Ultrahigh Dimensions," to appear in Emergent Information Technologies and Enabling Policies for Counter Terrorism. R. Popp and J. Yen (eds.), IEEE Press, Apr. 2006. S. Rajasekaran, R. Ammar, S. Demurjian, A. Abdel-Raouf, T. Doan, J. Lian, M. Song, and A. Mohamed, Strategies to Process High Volumes of Data in Support of Counter-terrorism, IEEE Aerospace Conference, March 2005.

[2] S. Rajasekaran, R. Ammar, S. Demurjian, A. Abdel-Raouf, T. Doan, J. Lian, M. Song, and A. Mohamed, Strategies to Process High Volumes of Data in Support of Counter-terrorism, IEEE Aerospace Conference, March 2005.

[3] S. Demurjian, S. Rajasekaran, R. Ammar. I. Greenshields, T. Doan, and L. He, Applying LSI and Data Reduction to XML for Counter Terrorism, Proc. 27th IEEE Aerospace Conference, Big Sky, MT, March 2006.

[4] LBornaz, L., Rinaudo F. Terrestrial laser scanner data processing. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Science Congresss, Istanbul, Turkey, Vol. XXXV, Part B5, pp. 514-519.

[5] M. Song and S. Rajasekaran, Fast k-Means Algorithms with Constant Approximation, Proc. International Symposium on Algorithms and Computations (ISAAC), Springer-Verlag LNCS 3827, 2005, pp. 1029-1038.

[6] Sanguthevar Rajasekaran, Efficient Parallel Hierarchical Clustering Algorithms, IEEE Transactions on Parallel and Distributed Systems 16(6), June 2005, pp. 497-502.

[7] L. Kaufman and P.J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, New York, 1990.

[8] J. T.-L. Wang et al. Evaluating a class of distance-mapping algorithms for data mining and clustering. In Knowledge Discovery and Data Mining, pages 307--311, 1999.

[9] S. Chandrasekaran, B.S. Manjunath, Y.F. Wang, J. Winkeler and H.Zhang. "An eigenspace update algorithm for image analysis". CVGIP, 1997.

[10] J. Bernardo, and A. Smith, Bayesian Theory. John Wiley and Sons, New York, 1994.P.

[11] Cheeseman, and J. Stutz, "Bayesian classification (AutoClass): Theory and results". In Fayyad, U., Piatesky-Shapiro, G., Smyth, Uthurusamy, R., editors. Advances in Knowledge Discovery and Data mining, pages 153-180. AAAI Press, Menlo Park, CA, 1995.

[12] E. Keogh, M. Pazzani "Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases", Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2000.

[13] S. Rajasekaran and M. Song, A Novel Scheme for the Parallel Computation of SVDs, Proc. International Conference on High Performance Computing and Communications (HPCC), Springer-Verlag Lecture Notes in Computer Science 4208, 2006, pp. 129-137.

[14] D. Achlioptas. "Database-friendly random projections", In Proceedings of ACM Symposium on the Principles of Database Systems, pages 274–281, 2001.

[15] V. Athitsos, J. Alon, S. Sclaroff, G. Kollios, "BosstMap: A Method for Efficient Approximate Similarity Rankings", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2004.

[16] E. Bingham and H. Mannila, "Random Projection in Dimensionality Reduction: Applications to Image and Text Data", Proceedings of the 7th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining, pp: 245-250, 2001.

[17] W. Johnson and J. Lindenstrauss. "Extensions of Lipshitz mapping into Hilbert space." In

Conference in Modern Analysis and Probability, Vol. 26 of Contemporary Mathematics, pages 189–206. Amer. Math. Soc., 1984.

[18] C. Papadimitriou, P. Raghavan, H. Tamaki, and S. Vempala. "Latent semantic indexing: A probabilisticanalysis", In Proceedings of the17th ACM Symposium on the Principles of Database Systems, pages 159–168, 1998.

[19] S. Roweis and L. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science, 290(5500), 2000, pp. 2323-2326.

[20] Tešić, S. Newsam, and B.S. Manjunath, "Challenges in Mining Large Image Datasets," in IPAM Short Program on Mathematical Challenges in Scientific Data Mining, Los Angeles, CA, Jan. 2002.

[21] K. I. Diamantaras, and S.Y. Kung, Principal Component Neural Networks: Theory and Applications, NY: Wiley, 1996.

[22] B. B. Eisenberg, On the Sample Complexity of PAC-Learning Using Random and Chosen Examples, M.SC Thesis, Massachusetts Institute of Technology, 1991.

[23] L. G. Valiant, "A theory of the learnable", Comm. ACM, 27 , pp. 1134–1142, 1984.

[24] N. Vapnik. The Nature of Statistical Learning Theory. Springer, New York, 1995.

[25] Trumbower, S. Rajasekaran, and P. Faghri, Identifying Offline Muscle Strength Profiles Sufficient for Short-Duration FES-LCE Exercise: A PAC Learning Model Approach, Journal of Clinical Monitoring and Computing, 20, 2006, pp. 209-220.

[26] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T.K. and Harshman, R. "Indexing by Latent Semantic Analysis", Journal of American Society for Information Science and Technology, Vol.41, 391–407, 1990.

[27] T. Hofmann, "Probabilistic Latent Semantic Indexing", Proceedings of ACM SIGIR 99, 50-57, 1999.

[28] E. Hoenkamp, "Unitary operators for fast latent indexing", Proceedings of 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 400-401, New York, 2001.

[29] Kokiopoulou, E. and Saad, Y. "Polynomial filtering in latent semantic indexing for information retrieval", Proceedings of the 27th annual international conference on Research and development in information retrieval, 104 – 111, 2004.

[30] Bassu, D. and Behrens, C. "Distributed LSI: Scalable Concept-based Information Retrieval with High Semantic Resolution", Proceedings of the 3rd SIAM International Conference on Data Mining (Text Mining Workshop), San Francisco, CA, May 3, 2003.

[31] Smith, C.U., Performance Engineering of Software Systems. Addison-Wesley, Boston, Mass.,1990.

[32] Smarkusky, D., Ammar, R., and Sholl, H., "A Framework for Designing Performance-Oriented Distributed Systems", Proceedings of the 6th IEEE Symp. on Computers and Communications, July 2001, pp. 92-98.

[33] Woodside, M., Hrischuk, C, Selic, B., Bayarov, S., "A Wideband Approach to Integrating Performance Prediction into a Software Design Environment", Proceedings of the First International Workshop on Software and Performance, Santa Fe, New Mexico, pp. 31-41, October 1998.

[34] Ammar, Reda "Hierarchical Performance Modeling and Analysis of Distributed Software", Chapter 12, Handbook of Parallel Computing: Models, Algorithms, and Applications, edited by S. Rajasekaran and J.H. Reif, Chapman & Hall/CRC Press, December 2007.

[35] Maurizio Lenzerini (2002). "Data Integration: A Theoretical Perspective". PODS 2002: 233-246.

[36] Patrick Ziegler and Klaus R. Dittrich (2004). "Three Decades of Data Integration - All Problems Solved?". WCC 2004: 3-12.

[37] Alon Y. Halevy (2001). "Answering queries using views: A survey". The VLDB Journal: 270-294.

[38] Alon Y. Halevy, Naveen Ashish, Dina Bitton, Michael J. Carey, Denise Draper, Jeff Pollock, Arnon Rosenthal, Vishal Sikka (2005). "Enterprise information integration: successes, challenges and controversies". SIGMOD 2005: 778-787.