Accelerating Data Mining Workloads: Current Approaches and Future Challenges in System Architecture Design

Jayaprakash Pisharath [*] Joseph Zambreno[†] Berkin Özışıkyılmaz [†] Alok Choudhary [†]

Abstract

With the unstoppable growth in data collection, data mining is playing an important role in the way massive data sets are analyzed. Trends clearly indicate that future decision making systems would weigh on even quicker and more reliable models used for data analysis. In order to achieve this, current algorithms and computing systems have to be optimized and tuned to effectively process the large volumes of raw data to be seen in future. In this paper, we present a brief overview of the current approaches and challenges faced in system design. The paper starts out by highlighting the uniqueness of data mining applications, which actually makes current "generic" system designs unsuitable for mining large data. Subsequently, we summarize the current innovations and efforts made by researchers to design systems to efficiently process data mining workloads.

1 Introduction

Data collection and data storage rates are growing at an exponential pace. The "How Much Information" project at the University of California at Berkeley [23] have recently estimated that over five exabytes of data was stored across paper, film, magnetic and optical mediums during the year 2002. Also, this number represents a near doubling of the total data stored worldwide in the previous two years. This particular study has not yet been updated to represent more recent years, but it is reasonable to assume that this exponential trend has continued. Corporations like Intel realizes this as well, since they have introduced the looming "Era of Tera" where people will require teraflops of computing power, terabits per second of communications bandwidth and terabytes of data storage [6]. One can already purchase terabytes of data storage today. If the current trends continue, such an explosion of data would clearly outstrip our ability to make meaningful use of it. While this problem is partially algorithmic in nature, it is also an indictment of current system architectural practices. In recent years, system performance has not been scaling well with the amount of data. Overall system performance has not been keeping up with on-chip speed improvements from advances in transistor technology. For example, the off-chip memory latency of a Pentium 4 is more than 2 times that of a 386 (in terms of processor cycles). Clearly, processor speeds have scaled but not their data handling capabilities. This is especially true for data-driven applications (including data mining workloads) since their underlying operations are memory-intensive in nature.

In order to close the gap between data-intensive applications and computing systems, we propose the following two-phased approach:
(a) The first phase involves performing an in-depth study to clearly understand the system characteristics and bottlenecks, and also to enlist the future computing requirements of data mining applications
(b) The second phase consists of designing novel (or adapting existing) computer systems to cater to the primary demands of data mining workloads. On the other hand, the algorithms too have to be revised to suit the demands of new applications and architectures.

The rest of this paper describes the above approach in detail. We also consider several existing work, and discuss their relevance to this approach. Then, we present the quantitative facts to prove the uniqueness of data mining applications and its impact on the current approach to system design. The paper then presents a brief overview of some of the emerging architectures and systems that are designed specifically for handling data mining workloads. We also present the tradeoffs between the many designs. Overall, our research concludes that a thorough understanding of the characteristics of data mining applications from a systems perspective is completely essential in order to design high performance data mining systems.

2 Application Analysis

In the past, researchers have performed system level characterizations of data mining algorithms to reveal their core characteristics and bottlenecks [1, 7, 22, 27]. These studies primarily focus on studying the memory behavior of select data mining applications. Given the fact that data mining is still an evolving field, understanding the dynamic runtime characteristics of the applications is in reality a challenging task. A comprehensive system-wide study of data mining

[*]Architecture Performance and Projections Group, Intel Corporation, Santa Clara, CA 95054. Email: jayaprakash.pisharath@intel.com

[†]Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL 60208. Email: {zambro1, boz283,choudhar}@eecs.northwestern.edu

applications would be useful in understanding their dynamic behavior.

In our research work, we use non-traditional methods to identify the system bottlenecks and computational requirements of several data mining applications. A comprehensive framework has been set up to extract various architectural features and design implications of data mining applications. We studied a mix of several data mining applications using this framework and compiled a benchmark suite named NU-MineBench [2]. This benchmark suite has been built after extensive investigation of well-known data mining applications and their characteristics. We also analyze each application in detail and identify the core characteristics that make it unique. Although there has been previous work analyzing individual data mining applications, we believe that analyzing the behavior of a complete benchmarking suite will certainly give a better understanding of the underlying bottlenecks for data mining applications. We understand the indispensable need for a data mining benchmark suite since there are currently no mechanisms to review data mining algorithms and systems that run data mining applications. Benchmarks do not only play a role in measuring the relative performance of different systems. In order to perform an architectural exploration study, a data mining benchmark is a very essential tool. Such a benchmark would also aid programmers in the specific domain in various ways.

NU-MineBench consists of 15 scalable, high performance applications from several application domains. The applications are scalable (parallel versions of applications are included), representative and widely known. Table 1 shows the applications and their descriptions [2]. We analyzed the benchmark applications using a mixture of simulators, architecture modeling tools and system performance profiling tools [22, 32, 25]. The goal of our studies is to analyze and extract the several system wide characteristics of data mining applications. We believe such characteristics serve to be one of the key factors to designing new algorithms and systems for data mining applications.

In the following subsections, we highlight some of the key characteristics of data mining applications that we identified from our studies.

2.1 Uniqueness One distinct feature is the uniqueness of data mining applications. We compared NU-MineBench applications against applications from other prominent benchmark suites. Specifically, data mining applications were compared against compute intensive applications, multimedia applications, streaming applications and database applications to identify the core differences. Applications were taken from integer applications benchmark (SPEC INT from SPEC CPU2000 [26]), floating point applications benchmark (SPEC FP from SPEC CPU2000), multimedia applications benchmark (MediaBench from UCLA [21]) and decision support applications benchmark (TPC-H from Transaction Processing Council [29]).

We performed statistical analysis on the architectural characteristics of applications to identify their core differences. Specifically, we monitored the performance counters during execution using profiling tools (like Intel VTune analyzer [19]) for every application, and analytically studied their individual characteristics. A k-Means based clustering algorithm [16] was applied to the performance characteristics of these applications. The goal of this clustering is to find clusters of characteristics. That is, the assumption is that each benchmark is unique. This is intuitively true since the characteristic features of each benchmark is different. If data mining applications resemble any other domain, they both will belong to the same cluster, in which case, existing system optimizations from the corresponding field can be applied to data mining applications as well. Figure 1 shows the scatter plot of the final cluster configuration obtained from the results of the clustering method. Clearly data mining algorithms fall into different clusters, with a few of them sharing characteristics with other application domains. Our approach identifies data mining applications to be distinctly unique.

Table 2 shows the distinct characteristics of data mining applications as compared to other applications. One key attribute that signifies the differences is the number of data references per instruction retired. For data mining applications, this rate is 1.103, whereas for other applications, it is significantly less. The number of bus accesses originating from the processor to the memory (per instruction retired) verifies this fact as well. These results solidify the intuition that data mining is data-intensive by nature.

Another important difference is the fraction of total instruction decodes to the instructions retired. This measure defines the instruction efficiency of a processor. In our case, the results indicate that data mining applications are not well handled by the processor. Resource related stalls comprises of the delay that incurs from the contention of various processor resources, which include register renaming buffer entries, memory buffer entries, and also the penalty that occurs during a branch misprediction recovery. The above measures have a direct impact on the CPI (Cycles Per Instruction retired) of the system. A CPI of 1.54 for data mining applications is considered to nominal [19, 18]. Besides, the number of ALU operations per instruction retired is also high for data mining applications, which indicates the extensive amount of computations performed in data mining applications.

What makes the data mining applications unique is this combination of high data rates combined with high computation power requirements. Such a behavior is clearly not seen in other applications. In addition, data mining applications tend to oscillate between such data and compute phases reg-

Table 1: Overview of the NU-MineBench data mining benchmark suite

Application	**Category**	**Description**
`k-Means`	Clustering	Mean-based data partitioning method
`Fuzzy k-Means`	Clustering	Fuzzy logic-based data partitioning method
`HOP`	Clustering	Density-based grouping method
`BIRCH`	Clustering	Hierarchical data segmentation method
`Apriori`	ARM	Horizontal database, level-wise mining based on Apriori property
`Eclat`	ARM	Vertical database, equivalence class based method
`Utility`	ARM	Utility-based association rule mining
`ScalParC`	Classification	Decision tree classification
`Naive Bayesian`	Classification	Statistical classifier
`SNP`	Classification	Hill-climbing search method for DNA dependency extraction
`Rsearch`	Classification	RNA sequence search using stochastic context-free grammar
`SVM-RFE`	Classification	Gene expression classifier using recursive feature elimination
`GeneNet`	Structure Learning	Gene relationship extraction using microarray-based method
`SEMPHY`	Structure Learning	Gene sequencing using phylogenetic tree-based method
`PLSA`	Optimization	DNA sequence alignment using Smith-Waterman optimization method

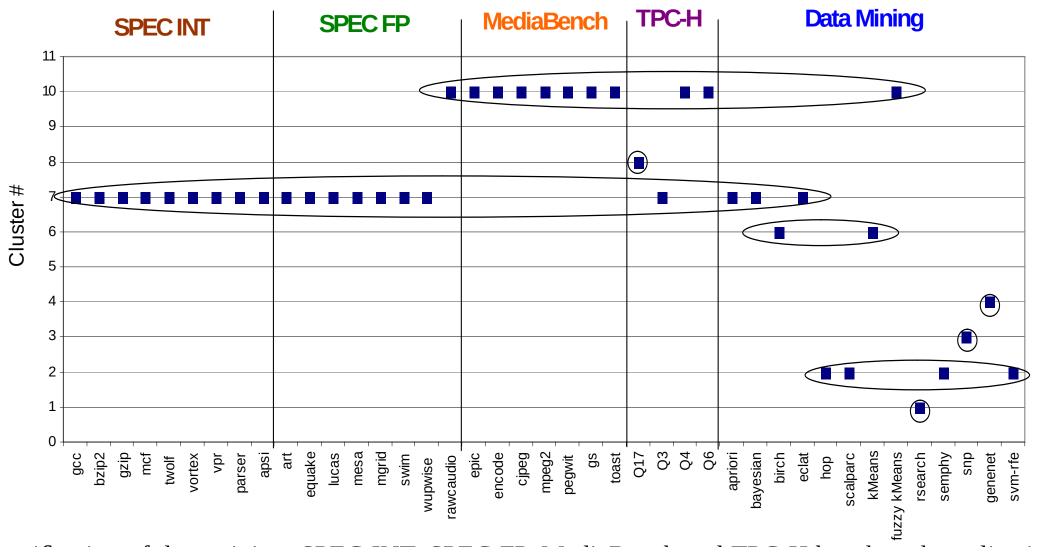

Figure 1: Classification of data mining, SPEC INT, SPEC FP, MediaBench and TPC-H benchmark applications based on their characteristics. A k-Means based clustering algorithm was used for this classification. Data mining applications tend to form unique clusters.

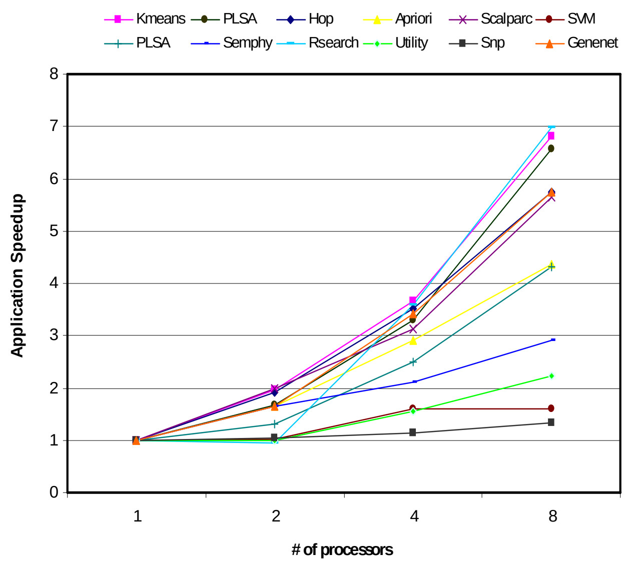

Figure 2: Scalability of NU-MineBench applications. The graph shows the speedup of data mining applications on a Intel Xeon based SMP machine for the 1, 2, 4 and 8 processor cases.

ularly, making the current processors and architectural optimizations mostly inadequate. Note that the CPI levels are nominal for these applications, which highlights the need for non-traditional system and architecture optimization techniques for data mining applications.

The L1 and L2 miss rates are considerably high for data mining applications. The reason for this is the inherent streaming nature of data retrieval, which does not provide enough opportunities for data reuse. This indicates that current memory hierarchy is insufficient for data mining applications. It should be noted that the number of branch instructions (and the branch mispredictions) are typically low for data mining applications, which highlights yet another unique behavior of data mining applications.

2.2 Scalability The core kernels of the NU-MineBench benchmark are extracted and scaled further by extending traditional data parallelism techniques widely available in the literature [31, 14, 4, 16]. Figure 2 shows the scalability of applications with the processors. Here, the scalability of the parallel applications is measured based on the execution times on an 8-way Intel Xeon based shared memory parallel machine.

Figure 2 shows the benchmark application execution speedups when executed on 1, 2, 4 and 8 processors. The performance numbers indicate that most applications show good scalability with a higher number of processors. For a few applications, the 2 processor case provides minimal or in some cases, no speedups. The best speedup, 6.98 on 8 processors, is seen for the Rsearch application. This is primarily due to the fact that the application data gets uniformly distributed on multiple processors, and the processors process their respective data concurrently. Every processor accesses only its respective data block in memory, synchronizing only occasionally. The clustering based applications follow Rsearch in terms of the achieved speedups. In general, it can be observed that clustering algorithms show better scalability than the remainder of the applications. The underlying reason for this observation is the highly parallelizable distance calculation routine, which is common to all the clustering algorithms. The worst scalability is observed for SNP and SVMRFE. For SVM-RFE, the problem arises due to unnecessary communication problems and locking of memory structures. This redundant locking is done to ensure the code works on distributed and shared memory machines. If the locks are removed (using other shared memory programming techniques), the program and its kernels scale better.

Overall, our studies clearly indicate that data mining applications are highly scalable. There are predominant compute-intensive kernels in data mining applications, which are highly scalable. Such kernels when parallelized using data parallelism methods, either at hardware [25] or traditional methods [22, 32], yield significant performance improvements in the applications. In a similar study done by Intel Corporation [4], researchers have proved that data mining applications are highly scalable. Their results show close to linear speedups for certain bioinformatics workloads for up to 8 processors and super-linear speedup for up to 16 processors on shared memory machines. They also prove the fact that when specific optimizations targeting the underlying system is applied to data mining applications, the applications provide super-linear speedups.

3 Architecture Design Projections: A System Perspective

The three approaches adopted by recent researchers to improve the system level performance of data mining applications are as follows:

1. Heavily system-optimized applications, packages, libraries and interfaces
2. Parallel and distributed system design
3. Custom-hardware based system design

Figure 3 shows the above approaches in terms of their complexity. Algorithmic and library based approaches have a quick turnaround time. Algorithmic optimizations for enhancing system performance are hard to conceive, but once conceived, they are easy to embed into the data mining application in the form of libraries or just by recompiling the application source using compilation flags. On the other hand, parallel and distributed versions of these algorithms

Table 2: Comparison of data mining application with other benchmark applications

Parameter[†]	*Benchmark of Applications*				
	SPECINT	**SPECFP**	**MediaBench**	**TPC-H**	**NU-MineBench**
Data References	0.81	0.55	0.56	0.48	1.10
Bus Accesses	0.030	0.034	0.002	0.010	0.037
Instruction Decodes	1.17	1.02	1.28	1.08	0.78
Resource Related Stalls	0.66	1.04	0.14	0.69	0.43
CPI	1.43	1.66	1.16	1.36	1.54
ALU Operations	0.25	0.29	0.27	0.30	0.31
L1 Misses	0.023	0.008	0.010	0.029	0.016
L2 Misses	0.003	0.003	0.0004	0.002	0.006
Branches	0.13	0.03	0.16	0.11	0.14
Branch Mispredictions	0.009	0.0008	0.016	0.0006	0.006

[†] *The numbers shown here for the parameters are values per instruction retired*

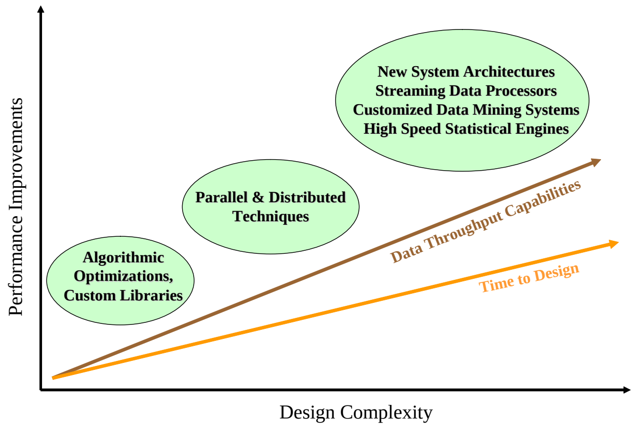
New System Architectures
Streaming Data Processors
Customized Data Mining Systems
High Speed Statistical Engines
Parallel & Distributed Techniques
Algorithmic Optimizations, Custom Libraries
Data Throughput Capabilities
Time to Design
Performance Improvements
Design Complexity

Figure 3: Tradeoffs between algorithmic optimizations, high performance techniques and custom hardware based techniques. The techniques are compared in terms of the design complexity, projected performance, time to market and data handling capabilities.

offer even faster execution times. They require a lot of programming in order to map the algorithm to the underlying parallel resources. To ensure accuracy of results and a quick turnaround time, extensive system optimizations are necessary in these parallel codes as well. Custom-logic and accelerator driven systems began to be expensive alternatives. But the prices of custom-logic is falling drastically in recent years [30], which make them a viable alternative for executing data mining applications in terms of cost. The challenging task lies in identifying the core computational components of data mining applications and mapping them efficiently to these logic-based accelerators. In the rest of this section, we elaborate on existing techniques based on the above three approaches to speed up data mining applications. We present a few novel approaches proposed by researchers in the area of system design and customization for data mining applications. We also present the advantages of each approach.

The first approach involves enhancing the application capabilities to adopt to the system on hand. This includes both system-specific optimizations at the algorithmic level and also the design of interfaces and libraries to allow seamless interaction of the application with the underlying architecture. Chen et al. propose an optimization at the algorithmic level to adapt a data intensive application to minimize the usage and caching of memory [3]. In another work, a dynamic memory allocation strategy based on the access patterns of certain data mining applications is proposed by Parthasarathy et al. [24]. They propose memory sharing techniques that exploit the data locality and false-sharing properties of shared memory data to significantly improve the overall application performance. On the other hand, there also exists several statistical libraries [11, 15] that have been optimized to minimize the interaction between applications and the underlying system architecture. Such interfaces also hide the complexities of system by providing a seamless access to the application. Govindaraju et al. extend existing algorithmic optimizations and the execution capabilities of graphics processors to data mining applications [13]. Specifically, they utilize an existing sliding window based sorting algorithm (optimized version) available in graphics processors and apply it to numerical statistics computation on data streams.

As verified by our studies (presented in the previous section), data mining applications do heavily favor scalable systems. There is tremendous amount of data and instructions that need to be processed. Researchers have offered several parallel and distributed versions of data mining algorithms [31, 4, 12]. The goal of such development methodologies is to fully utilize the computing power offered by large scale high performance setups. There are also scalable data mining packages and benchmarks for high performance systems [15, 20, 11].

On the other hand, one also has the option of designing customized systems for data mining applications. Researchers at the University of California at Santa Cruz built a custom-logic based system to accelerate algorithms from computational chemistry, computational biology, and other related fields. The system offers 20X to 40X speedups on a 520 node custom system [8]. The system is specially designed to effectively solve certain types of problems, such as gene sequence alignment and hidden Markov model training. Recently, TimeLogic has introduced a custom genome analysis system named DeCypher Engine [28]. This processing engine, based on Field-Programmable Gate Array (FPGA) logic, accelerates the computationally intensive modules within bioinformatics algorithms. Specifically, DeCypher Engine provides optimal solutions for BLAST, Smith-Waterman, Hidden Markov Model Analysis, and their custom gene modeling applications. Results report a 15X to 70X speed up on BLAST based applications. Compugen, a genomics-based drug and diagnostic discovery company, has introduced BioXL/H [5]. This is a high-end hardware accelerator for rigorous homology searches on protein and nucleic acid sequence databases. BioXL/H builds upon Compugen's Bioaccelerator and BioXL/P, which are the well-known commercial accelerators. Compugen designed the BioXL/H for higher throughput. A few of the accelerated applications include Smith-Waterman, ProfileSearch, Translated Searches, Frame and Profile-Frame. BioXL/H executes these algorithms up to 3X magnitude faster than a high end UNIX processor.

In [17], Hayashi et al. propose using a PRAM to perform the k-Merge process, a conventional method of performing data mining. A clustering algorithm based on k-Means methodology was mapped on to a reconfigurable hardware logic by Estlick et al. [10]. The above work require algorithmic modifications to the code. Another pattern matching (pure string matching) hardware has been proposed by Zhang et al. in [33]. They introduce a string matching logic on reconfigurable logic and prove that data mining applications that use string matching show significant speedups.

Recently, we proposed a hardware accelerator for density based clustering applications [25]. In this work, we first extract the inherent kernels of density-based data mining applications. The kernels are actually identified after extensive characterization and not based on pure algorithmic analysis. The logic (and the kernels) are generic and are not application-specific. Our methodology and design are applicable to any distance/density based algorithms. These core kernels are then accelerated using our proposed accelerator-based data mining system.

Figure 4 illustrates the generic design of our proposed accelerator based data mining system. The kernel accelerator exists as a coprocessor along with the existing general purpose processor. The decision to go with a coprocessor is because the kernels (that are accelerated by the accelerator) tend to run for a considerable amount of time, allowing the code to be offloaded to the coprocessor (ensuring good speedups that is worth the offload overhead) and thus, allowing the general purpose processor take up other tasks without getting blocked. Our accelerator is designed to accelerate the major kernels of most data mining applications. In the case of [25], we were able to accelerate the three core kernels of density based clustering algorithms, which include density calculation and the migration of points to denser regions. The accelerator works with the processor in speeding up the kernel execution, which results in application speedups close to 60X to that of a generic system.

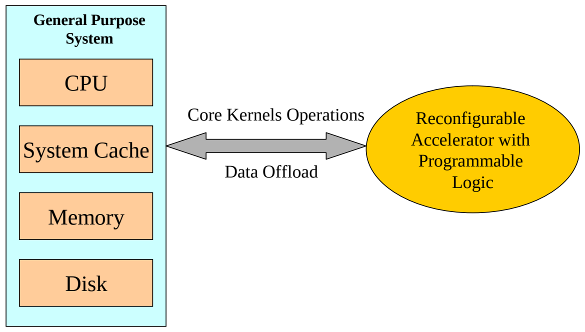

Figure 4: Design of the accelerator based data mining system.

3.1 Comparison of Approaches In order to understand the benefits from various approaches, we compare the speedups obtained from various approaches. We consider a cosmology application named HOP [9], and parallelize the kernels of this application using the different approaches. That way, we verify their scalability and also study the benefits of using each of the schemes. We extracted and parallelized the kernels on a variety of high performance setups. The core kernels of HOP include density calculation, the neighborhood search and gather process [25]. Figure 5 shows the performance based on the application speedups obtained on *(a)* distributed memory machines, *(b)* shared memory machine, and *(c)* reconfigurable logic based data mining system. The base execution times of each setup is close. On an eclectic distributed memory setup, the best speedup achieved is 25X on 64 processors. As the number of compute nodes increase, the communication overheads increase proportionally, thus, undermining the achieved application speedups. On a shared memory machine, the same application shows 6X speedup on 8 processors. Figure 5(b) shows the performance of the application on a shared memory setup consisting of an 8-way Intel Xeon machine. There is no communication in a shared memory model, but the overheads due to memory contention increase as the number of processors increase. Shared memory machines offer good scalability. However, the complexity of programming and the cost of building large scale shared memory machines could be excessive.

Since the application kernels are stand-alone components, we extracted and ported these kernels to a prototype data mining system. This data mining system is based on the architecture presented in Figure 4. Figure 5(c) shows the result of executing the application on this data mining system. Clearly, a reconfigurable logic based data mining system offers the best speedups. The reason for this is the fact that the coprocessor is pre-programmed to effectively accelerate the core kernels in the application [25]. As the kernel speedups improve, the overall application execution times decrease. The non-kernels execution times do not benefit from the techniques, which in turn partially mask the benefits obtained from the kernel speedups. However, our studies indicate that core kernels cover 86 to 99% of any data mining application [25], which basically is the reason for the remarkable speedups seen in Figure 5(c). Our methodology relies on extracting core (scalable) kernels and accelerating them on custom logic, thus, driving outstanding application speedups.

4 Conclusion

Data growth is an unavoidable. Future system design should consider this growth in data and also the enhanced expectations in data analysis. We use this fact as a motivation and present NU-Minebench, a diverse benchmark suite of data mining applications, to enable development of superior algorithms and systems for data mining applications. Using NU-MineBench, we verify the fact that data mining applications form a unique kind of workload, which makes current system capabilities insufficient to handle them. From an in-depth study of data mining applications, we conclude that they are very scalable in nature. Their memory behavior is different from other benchmark applications. They show high L1 and L2 data cache miss rates. Another important factor is their low instruction-level parallelism. We strongly believe that the architectural characteristics of data mining workloads need be considered before designing high performance data intensive systems.

We presented existing trends and efforts made by researchers to efficiently handle data intensive workloads. Introducing algorithmic optimizations is the most common and straight-forward approach. Novel high performance setups are also considered in our study. We found that if the core kernels of data mining applications are smartly extracted, high performance setups tend to offer the best speedups. The prices of custom-logic based reconfigurable systems have been plummeting in recent years. We use this fact as a motivation and propose a new accelerator-based data mining system. This system speeds up the core application kernels by providing custom logic for the underlying kernel operations. Our results indicate that such customized setups offer the best speedups for data mining applications. Overall, our results stress the need for a better understanding of the runtime characteristics of data mining applications, especially from a system perspective. For instance, our studies proved that data mining applications contain within them several highly scalable kernels, which in turn can be scaled using smart hardware techniques to offer better performance.

5 Acknowledgements

This work was supported by NSF grants CNS-0404341, IIS-0536994, CCF-0444405, DoE grant DE-FC02-01ER25485 and by a research grant from Intel Corporation. The authors would like to thank Pradeep Dubey and Carole Dulong from the Architecture Research Team of Intel Corporation for their valuable input and feedback.

References

[1] J. Bradford and J. Fortes. Performance and memory-access characterization of data mining applications. In *Workshop on Workload Characterization at the Annual International Symposium on Microarchitecture.*, 1998.

[2] Center for Ultra-scale Computing and Information Security (CUCIS), Northwestern Univeristy. NU-MineBench Version 2.0 - Scalable Data Mining Benchmark. Available at http://cucis.ece.northwestern.edu/projects/DMS/MineBench.html, 2005.

[3] S. Chen, P.B. Gibbons, T.C. Mowry, and G. Valentin. Fractal prefetching B+trees: optimizing both cache and disk performance. In *Proc. of ACM SIGMOD*, 2002.

[4] Y. Chen, Q. Diao, C. Dulong, W. Hu, C. Lai, E. Li, W. Li, T. Wang, and Y. Zhang. Performance scalability of data mining workloads in bioinformatics. *Intel Technology Journal*, 09(12):131–142, May 2005.

[5] CompuGen. Bioxl accelerator, 2003. http://www.cgen.com/products/.

[6] Intel Corporation. *Architecting the Era of Tera - Technical White Paper*, 2005. http://www.intel.com/technology/computing/archinnov/teraera/.

[7] A. Czezowski and P. Christen. How fast is fast? Performance analysis of KDD applications using hardware performance counters on UltraSPARC-III. Technical Report TR-CS-02-03, The Australian National University, Department of Computer Science, September 2002.

[8] D. Dahle, L. Grate, E. Rice, and R. Hughey. The ucsc kestrel general purpose parallel processor. In *Proc. of Int. Conference on Parallel and Distributed Processing Techniques and Applications*, 1999.

[9] D.J. Eisenstein and P. Hut. Hop: A new group finding algorithm for N-body simulations. *Journal of Astrophysics*, (498):137–142, 1998.

[10] M. Estlick, M. Leeser, J. Theiler, and J.J. Szymanski. Algorithmic transformations in the implementation of K-means clustering on reconfigurable hardware. In *Proc. of the International Symposium on Field Programmable Gate Arrays*, 2001.

[11] National Center for Biotechnology Information. Tools for data mining, 2005. http://www.ncbi.nlm.nih.gov/Tools/.

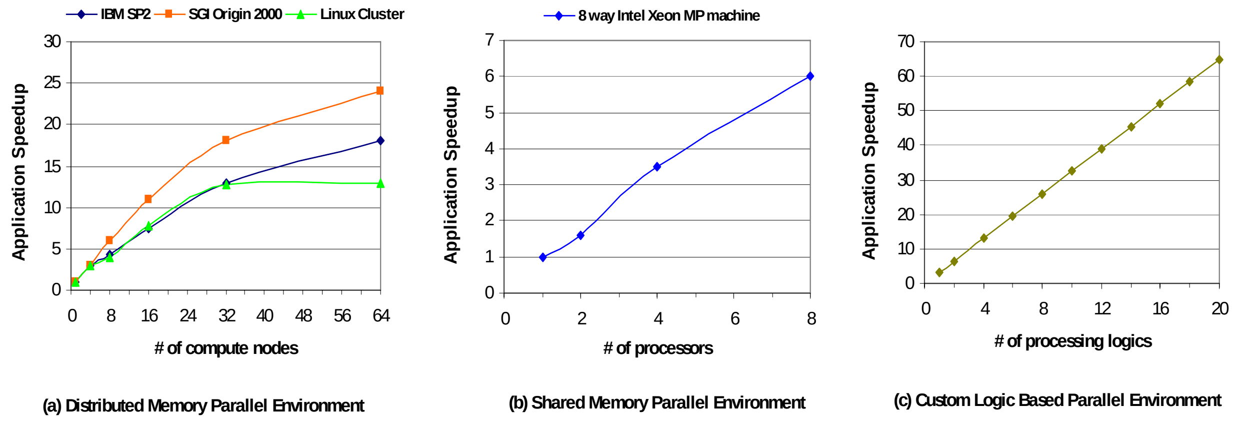

Figure 5: Comparison of speedups obtained using different high performance techniques. Speedups obtained for HOP when using a setup with (a) distributed memory machines, (b) shared memory machine, and (c) customized reconfigurable-logic based system.

[12] A.A. Freitas and S.H. Lavington. *Mining Very Large Databases with Parallel Processing*. Kluwer Academic Publishers, 1998.

[13] N.K. Govindaraju, N. Raghuvanshi, and D. Manocha. Fast and approximate stream mining of quantiles and frequencies using graphics processors. In *Proc. of the ACM SIGMOD international conference on Management of data*, pages 611–622, New York, NY, USA, 2005. ACM Press.

[14] A. Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing*. Addison Weslesy, 2003.

[15] Numerical Algorithms Group. Nag parallel library, 2005. http://www.nag.co.uk/numeric/fd/FDdescription.asp.

[16] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, August 2000.

[17] T. Hayashi, K. Nakano, and S. Olariu. Work-time optimal k-Merge algorithms on the PRAM. *IEEE Transactions on Parallel and Distributed Systems*, 9(3):275–282, 1998.

[18] J.L. Hennessy and D.A. Patterson. *Computer Architecture: A Quantitative Approach*. Morgan Kaufmann, third edition, 2002.

[19] Intel Corporation. Intel VTune performance analyzer 7.2, 2005. http://www.intel.com/software/products/vtune/vpa/.

[20] Los Alamos National Laboratory. mpiblast, 2005. http://mpiblast.lanl.gov/.

[21] C. Lee, M. Potkonjak, and W.H. Mangione-Smith. Mediabench: A tool for evaluating and synthesizing multimedia and communicatons systems. In *Proc. of the International Symposium on Microarchitecture*, 1997.

[22] Y. Liu, J. Pisharath, W.K. Liao, G. Memik, and A. Choudhary. Performance characterization of scalable data mining algorithms. In *Proc. of the International Conference on Parallel and Distributed Computing and Systems*, 2004.

[23] University of California at Berkeley. How much information? 2003. http://www.sims.berkeley.edu:8000/research/projects/how-much-info-2003/execsum.htm.

[24] S. Parthasarathy, M.J. Zaki, and W. Li. Memory placement techniques for parallel association mining. In *Proc. of the Intl. Conference on Knowledge Discovery and Data Mining*, 1998.

[25] J. Pisharath and A. Choudhary. Design of a hardware accelerator for density based clustering applications. In *Proc. of the International Conference on Application-Specific Systems, Architectures, and Processors*, 2005.

[26] Standard Performance Evaluation Corporation. Spec cpu2000 benchmark. http://www.spec.org/benchmarks.html.

[27] M. Thoennes and C. Weems. Exploration of the performance of a data mining application via hardware based monitoring. *The Journal of Supercomputing*, 26(1):25–42, 2003.

[28] TimeLogic. Decypher engine g4, 2006. http://www.timelogic.com/decypher_engine.html.

[29] Transaction Processing Performance Council. TPC-H Benchmark Revision 2.0.0, 2004.

[30] N. Tredennick and B. Shimamoto. The inevitability of reconfigurable systems. *ACM Queue*, 1(7), October 2003.

[31] M.J. Zaki and C.-T. Ho. *Large-Scale Parallel Data Mining, Lecture Notes in Artificial Intelligence, Vol. 1759*. Springer, 2000.

[32] J. Zambreno, B. Ozisikyilmaz, J. Pisharath, G. Memik, and A. Choudhary. Performance characterization of data mining applications using minebench. In *Proc. of the 9th Workshop on Computer Architecture Evaluation using Commercial Workloads*, 2006.

[33] Q. Zhang, R. Chamberlain, R.S. Indeck, B.M. West, and J. White. Massively parallel data mining using reconfigurable hardware: Approximate string matching. In *Proc. of the International Parallel and Distributed Processing Symposium*, 2004.