EXACT STRING MATCHING FOR MS/MS PROTEIN IDENTIFICATION USING THE CELL BROADBAND ENGINE

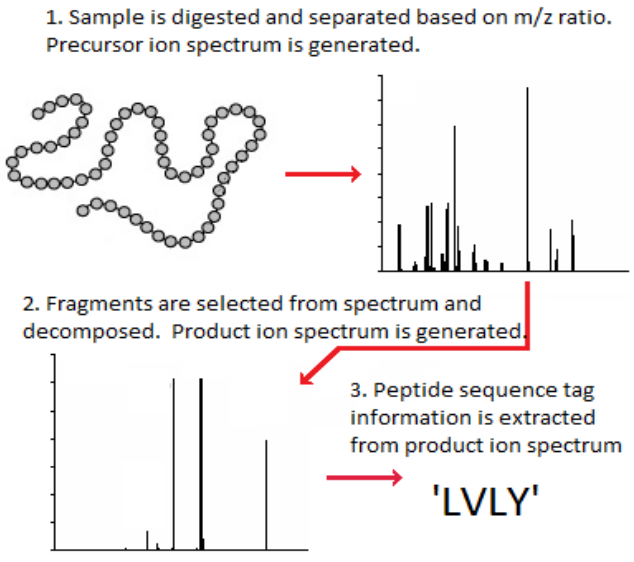
R.J. Peace, H. Mahmoud, J.R. Green

Departments of Systems and Computer Engineering, Carleton University, Ottawa, Ontario, Canada

I. INTRODUCTION

Tandem mass spectrometry (MS/MS) is an analytical technique which identifies proteins based on their amino acid sequence. Proteins are digested using a proteolytic enzyme such as trypsin, and the resulting peptide fragments are ionized and separated based on their mass-to-charge (m/z) ratio, resulting in a spectrum known as a precursor ion spectrum. Fragments of a specific m/z ratio (which correspond to a peak in the precursor ion spectrum) are extracted from the protein sample after initial separation, further decomposed, and separated into a second spectrum -- known as a product ion spectrum -- from which peptide sequence tag data can be determined. Typically, this second decomposition is performed repeatedly for the most abundant peaks in the precursor ion spectrum. In a typical mass spectrometer, peptide sequence tag data are generated approximately once per second. Figure 1, below, demonstrates the tandem mass spectrometry pipeline.

Figure 1: The Tandem Mass Spectrometry Pipeline



Recent advances in computational mass spectrometry have largely been geared toward improvement of off-line data analysis [1-6]. Throughout this time period, focus has been placed on three areas of computational mass spectrometry: offline statistical protein identification [7], offline quantification of relative protein abundance [8], and visualization of analysis outcomes [9]. Each of these avenues of computational mass spectrometry are fundamentally limited, as results of off-line data analysis are only as good as the collected data, and no amount of offline data analysis can improve the data acquisition process.

An additional avenue of computational mass spectrometry that has not yet been thoroughly explored is hypothesis-driven tandem mass spectrometry (hdMS/MS) [10]. Here, online or 'in the loop' data analysis tools guide a mass spectrometer's data acquisition strategy in real-time, producing mass spectrometry data which is more likely to uniquely identify proteins.

Unfortunately, hdMS/MS requires that data analysis be performed under strict real-time deadlines which are imposed by the mass spectrometry hardware; typical iterations of a mass spectrometer are in the range of one to three seconds, therefore real-time data analysis for hdMS/MS must produce a result within this short window in order to be effective. As discussed below, we propose that careful parallelization of analysis algorithms is the only way to achieve the necessary computational acceleration to meet these strict deadlines.

The concept of hypothesis-driven MS/MS, also known as directed MS/MS or non-redundant MS/MS, has been discussed previously [11][12]. However, the implementations of hypothesis-driven MS/MS presented in these papers require that data acquisition be temporarily suspended in order to complete the analysis of data collected so far, and therefore are limited to specific types of MS instrumentation. Zerck et al have explored the potential benefits of a fully real-time hdMS/MS system through simulation of mass spectrometry hardware [11], but have not attempted to actually implement a real-time hypothesis-driven MS/MS system. Recent developments in parallel computing, including field programmable gate arrays (FPGA), general-purpose computing on graphics processing units (GPGPU), and the Cell B/E architecture, provide the potential to achieve the level of computational acceleration required to implement a true real-time hdMS/MS system.

One of the crucial elements of the hdMS/MS system is exact single string matching. Peptide fragments must be searched against large databases of known proteins as the peptide fragments are detected by the mass spectrometry hardware. Thus, as a step toward a complete hdMS/MS system, we have developed an extension of the Parabix string searching algorithm [13] which is capable of rapidly searching peptide sequence tags against proteomic databases using the Cell B/E architecture. The Cell B/E is a heterogeneous multicore architecture which utilizes 8 powerful synergistic processing elements (SPEs). Each SPE has single-instruction-multiple-data (SIMD) capabilities as well as an independent direct memory access (DMA) engine for overlapping memory transfers with computations in order to hide memory latencies. Through careful algorithm design, these features can be leveraged to achieve significant acceleration of scientific computing [14].

Current parallel string matching literature does not describe algorithms which are well suited to the string matching problem presented by hypothesis-driven MS/MS. Existing string matching efforts on the Cell B/E and GPU architectures are almost exclusively implementations of the Aho Corasick algorithm [15], which uses a deterministic finite automaton to match multiple query strings against a stream of data. These algorithms can be classified into two main categories: Exact matching [16-18] and regular expression matching [19,20]. The Aho Corasick algorithm is not well suited to hypothesis-driven MS/MS, however, as it does not take advantage of database pre-processing and is not ideal for single query strings.

The Parabix approach to string searching [13] transposes characters into bitstreams prior to searching, an optimization which takes advantage of SIMD operations to increase search throughput significantly at the cost of pre-processing time. Since proteomic databases very rarely change, pre-processing time for these databases is not an issue. As a result, the Parabix approach is an ideal candidate for hypothesis-driven MS/MS database searching.

Here, we present the results of a study in which we have evaluated the search throughput of various string matching algorithms, including the Boyer-Moore [21] and Rabin-Karp [22] methods, and the Parabix approach, and the Orthogonal Parabix approach – our extension of the Parabix approach. The goal of this study is to determine the optimal string matching algorithm for the purpose of hdMS/MS.

II. METHODS

A. Existing String Matching Algorithms

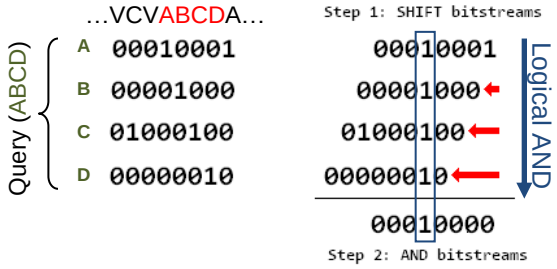
As a first step toward building a string matching module for a hdMS/MS system, we have implemented and optimized several state of the art string matching algorithms on the Cell B/E in order to determine the suitability of these algorithms for hdMS/MS. We have implemented the *shifting substring* (naive), Boyer-Moore [18], Rabin-Karp [19], and Parabix [13] string matching algorithms, adapting them for the Cell B/E's multicore architecture and applying advanced optimization techniques such as loop unrolling, application of SIMD vector operations, data parallelism, and multi-buffering techniques. Our implementations of the Boyer-Moore and Rabin-Karp algorithms are the first Cell B/E implementations of these algorithms.

B. The Orthogonal Parabix Approach

The Parabix approach, which transposes characters from 8-bit ascii format into parallel bitstreams, appears well-suited to the Cell B/E architecture. However, the Parabix approach is designed for streaming XML data, a problem space in which offline database pre-processing is not possible and it is optimized for simultaneously searching for multiple, potentially complex, query strings. Our problem space involves effectively unlimited pre-processing time and a very small subset of regular expressions in its queries – only exact matches and wildcards are possible, and the character set consists of only 20 amino acids. Therefore adapting the Parabix approach to suit our problem space has allowed us to optimize the approach significantly. The Orthogonal Parabix approach extends the Parabix approach, using the concept of parallel bit streams, while maximizing the amount of pre-processing done to the database in order to minimize online computation.

As a database pre-processing step, the Orthogonal Parabix algorithm generates 20 orthogonal bitstreams, one for each of the 20 amino acids. Each of the bitstreams has a 1 at bit *n* iff the database contains the corresponding amino acid at position *n*. This database pre-processing step allows for efficient exact string matching at the expense of an increase in database size of 250% relative to a standard 8-bit database encoding. The impact of this database expansion is limited during the online search portion of the Orthogonal Parabix algorithm, since only bitstreams relating to each character in the particular seach query are fetched and examined. During the online portion of the algorithm, starting at query position 0, bitstreams are shifted left by their position in the query. Then, all of the query bitstreams are ANDed resulting in an output bitstream in which bit *n* is 1 iff the database contains the query starting in the *n*th position. Figure 2 demonstrates the online shift-AND

Figure 2: Online shift-AND procedure for query ABCD in database containing "...VCVABCDA..."



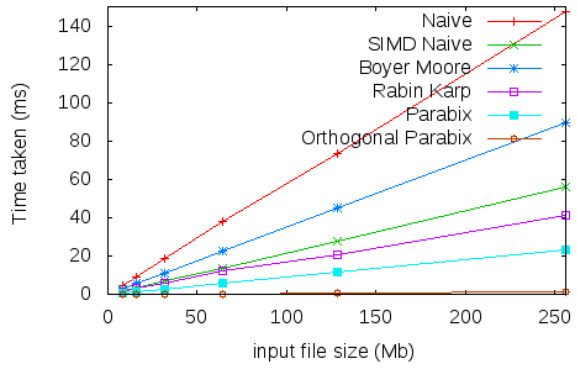
procedure of the Orthgonal Parabix algorithm. Using this shift-AND procedure and bitwise SIMD vector operations, the Orthogonal Parabix algorithm is capable of searching 128 database locations in 2*n*-1 operations, where *n* is the length of the query. The *shifting substring* algorithm, when fully optimized and using SIMD operations, requires a comparable number of operations to search only 16 database locations.

III. RESULTS

In order to determine the optimal string matching algorithm for hdMS/MS, optimized Cell B/E implementations of all previously detailed string matching algorithms were applied to a sample proteomic database and the resulting search times were compared. Each of the tests were performed on a QS22 Cell B/E blade server. The naive string matching algorithm appears twice in these results – with and without the use of SIMD vector operations – in order to demonstrate the increase in performance achieved through optimization on the Cell B/E.

Figure 3 demonstrates that each of the algorithms, as implemented, scale linearly with proteome size. In addition, Figure 3 demonstrates that the Orthogonal Parabix algorithm is extremely well-suited to the exact string matching problem which hdMS/MS presents – its runtime line is indistinguishable from the x-axis for most file sizes. When searching a 256Mb input file for

Figure 3: Search time under varying database size



approach achieves a sustained throughput of 215.4Gbps, representing a speedup of 19.6x over the Parabix approach and a speedup of 124.0x over a naive string searching implementation.

Figure 4 demonstrates the speedup which is obtained when data parallelism is employed to distribute the database searching task concurrently among multiple SPEs (256Mb database, 4 character query). For all algorithms, the speedup for *n* SPEs is approximately linear. By extension, the Orthogonal Parabix algorithm retains its speedup over other algorithms regardless of the number of SPEs used.

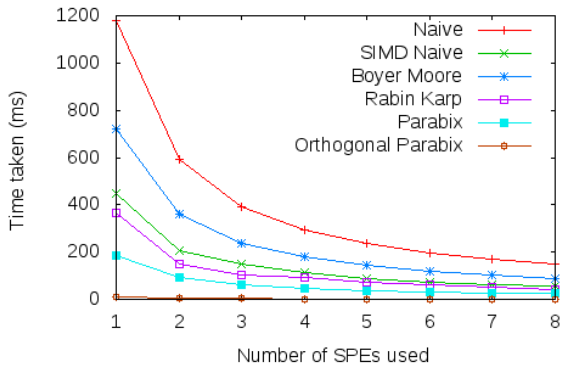
IV. DISCUSSION

To our knowledge, this study represents the first comparison of string matching algorithms for searching protein databases using the Cell B/E heterogeneous multicore processor. While the Boyer-Moore and Rabin-Karp algorithms proved to be suboptimal for the present application, the highly optimized Cell B/E implementations developed here may be useful for other studies.

While the results obtained thus far are promising, the creation of a proteomic string matching algorithm represents only a first step toward a complete hypothesis-driven MS/MS system. A complete hdMS/MS system will require several modules, all of which must function together within the real-time deadlines arising from mass spectrometry hardware. Future work on the hdMS/MS system includes, potentially:

- Creation of a parallel *de novo* sequencing algorithm which generates peptide sequence tags to be used as queries by the string matching algorithm. Alternately, a parallel algorithm which rapidly compares observed spectra against a database of spectra using cross-correlation techniques could be used.
- a decision-making system in which evidence gathered during string matching and spectral

Figure 4: Search times under varying SPE usage



alignment modules is used to determine an optimal future data acquisition strategy.

In addition, work must be performed in order to coordinate the Cell B/E processor with mass spectrometry hardware, so that the Cell B/E may guide mass spectrometry data acquisition.

IV. CONCLUSIONS

We have successfully implemented, optimized and benchmarked several single exact string matching algorithms on the Cell B/E hardware. We have implemented the Boyer-Moore and Rabin-Karp algorithms for the first time on the Cell B/E, and created an extension of the Parabix approach – called the orthogonal Parabix approach – which is ideal for the string searching problem presented by hdMS/MS.

Running on a Cell B/E blade server, the Orthogonal Parabix string matching approach was able to achieve a sustained search throughput of 215.4 Gbps under typical peptide fragment search parameters. Considering that the size of the human proteome is roughly 0.6Gb and a typical mass spectrometer obtains one sample per second, this throughput suggests that the Orthogonal Parabix approach running on the Cell B/E hardware is capable of searching proteomic databases within the real-time deadlines imposed by hdMS/MS.

The Orthogonal Parabix implementation as presented will be incorporated with further work, with the goal of providing a complete hdMS/MS system capable of directing data collection by the mass spectrometry hardware in real-time.

REFERENCES

[1] R. Aebersold and M. Mann, “Mass spectrometry-based proteomics,” *NATURE*, 422:198-207, 2003.

[2] A. Nesvizhskii and R. Aebersold, “Analysis, statistical validation and dissemination of large-scale proteomics datasets generated by tandem ms,” *Drug Discovery Today*, 9(4):173-181, 2004.

[3] R. Johnson, M. Davis, J. Taylor, and S. Patterson, “Informatics for protein identification by mass spectrometry,” *Methods*, 35(3):223-236, 2005.

[4] A. Nesvizhskii, O. Vitek, and R. Aebersold, “Analysis and validation of proteomic data generated by tandem mass spectrometry,” *Nature Methods*, 4:787-797, 2007.

[5] R. Sadygov, D. Cociorva, and J. Yates III, “Large-scale database searching using tandem mass spectra: Looking up the answer in the back of the book,” *NATURE Methods*, 1(3):195-202, 2004.

[6] E. Deutsch, H. Lam, and R. Aebersold, “Data analysis and bioinformatics tools for tandem mass spectrometry in proteomics,” *Physical Genomics*, 33:18-25, 2008.

[7] E. Kapp *et al*, “An evaluation, comparison, and accurate benchmarking of several publicly available ms/ms search algorithms: sensitivity and specificity analysis,” *Proteomics*, 5(13):3475-3490, 2005.

[8] L. Mueller, M. Brusniak, D. Mani, and R. Aebersold, “An assessment of software solutions for the analysis of mass spectrometry based quantitative proteomics data,” *Journal of Proteomics Research*, 7(1):51-61, 2008.

[9] M. Sturm and O. Kohlbacher, "TOPPView: an open-source viewer for mass spectrometry data.," *Journal of Proteome Research*, vol. 8, 2009, pp. 3760-3.

[10] A.T. Alex, M. Dumontier, J.S. Rose, and C.W. Hogue, "Hardware-accelerated protein identification for mass spectrometry," *Rapid Communications in Mass Spectrometry*, 19:833-7, 2005.

[11] A. Zerck, E. Nordho, A. Resemann, E. Mirgorodskaya, D. Suckau, K. Reinert, H. Lehrach, and J. Gobom, “An iterative strategy for precursor ion selection for lc-ms/ms based shotgun proteomics,” *Journal of Proteome Research*, 8(7):3239-3251, 2009.

[12] A. Scherl, P. Francois, V. Converset, M. Bento, J. A. Burgess, J.C. Sanchez, D. F. Hochstrasser, J. Schrenzel, and G. L. Corthals, “Non-redundant mass spectrometry: A strategy to integrate mass spectrometry acquisition and analysis,” *PROTEOMICS*, 4(4):917-927, 2004.

[13] R. Cameron, K. Herdy, and D. Lin, “High performance xml parsing using parallel bit stream technology,” In *Proceedings of the 2008 conference of the Center for Advanced Studies on collaborative research: meeting of minds*. ACM, 2008.

[14] S. Williams, J. Shalf, L. Oliker, S. Kamil, P. Husbands, and K. Yelick, "The potential of the Cell processor for scientific computing," *Proceedings of the 3rd conference on Computing frontiers - CF '06*, 2006.

[15] A. Aho and M. Corasick, “Efficient string matching: An aid to bibliographic search,” *Communications of the ACM*, 8(6):333-340, 1975.

[16] D. P. Scarpazza, O. Villa, and F. Petrini, “Peak-performance DFA-based string matching on the Cell processor,” In *Parallel and Distributed Processing Symposium*, 2007, page 1-8. IEEE, 2007.

[17] D. P. Scarpazza, O. Villa, and F. Petrini, “Exact multi-pattern string matching on the Cell/B.E. processor,” In *Proceedings of the 5th conference on computing frontiers*, pages 33-42. ACM, 2008.

[18] C. Schatz and C. Trapnell, “Fast exact string matching on the GPU,” Technical report, Center for Bioinformatics and Computational Biology. May 8, 2007.

[19] Vasiliadis G, M. Polychronakis, S. Antonatos, E. Markatos, and S. Ioannidis, “Regular expression matching on graphics hardware for intrusion detection,” In *Proceedings of the 12th International Symposium On Recent Advances In Intrusion Detection*, 2009.

[20] D. P. Scarpazza and G. Russell, “High-performance regular expression scanning on the Cell/B.E. processor,” In *Proceedings of the 23rd international conference on supercomputing*, pages 14-25. ACM, 2009.

[21] R. S. Boyer, and J. S. Moore, “A Fast String Searching Algorithm,” *Communications of the ACM*, 20(10):762-772. ACM, 1977.

[22] R. M. Karp, and M. O. Rabin, “Efficient Randomized Pattern-Matching Algorithms,” *IBM Journal of Research and Development*, 31(2):249-260, 1987.