

Parallel Guided Dynamic Programming Approach for DNA Sequence Similarity Search

A. R. M. Nordin, M. S. M. Yazid, A. Aziz and M. T. A. Osman

Abstract $\frac{3}{4}$ Development of DNA sequence comparison technique is an active research activity in computational biology application. Commonly techniques studied are dynamic programming and heuristic algorithms. Exhaustive dynamic programming algorithm produces optimal result but requires longer time and bigger space. Heuristic algorithm gives approximate results with much faster processing. We have developed a new model that improves the speed of large scale DNA sequence similarity search and at the same time the best possible alignment result is retained. The model is known as a guided dynamic programming approach for DNA sequence similarity search (FRA-Search). Two approaches are used to complete the FRA-Search model: an automaton based exact string matching algorithm is employed to skip irrelevant database sequences from being computed for dynamic programming alignment processing and; the rough sets theory has been employed to classify and reduce the dataset. This paper discusses the parallel model for FRA-Search application. The parallel FRA-Search model is implemented on PC-based cluster system. It is developed on a single program multiple data (SPMD) architecture and MPJ Express software is used as a communication interface protocol between processors.

Key words — DNA sequence comparison, dynamic programming algorithm, SPMD architecture, parallel computing, MPJ Express.

I. INTRODUCTION

Computing DNA sequence data has received high attentions from scientists. The growing advancements in engineering computational biology requires for better framework and algorithms to deal with the complex and extensive biological data. DNA sequence comparison is one of most important algorithms considered in computational biology.

When a new DNA sequence (query) is found by the biologists, they need to search the sequences in the databases for its similarity or relationship. The biological information from the similar sequences found in the databases can give important indications for determining the structure and

function of the query. Since discovery activities in new genomic projects have been very active, the number of sequences is expected to increase dramatically every year. NCBI BLAST server processes over 105 queries a day, and this rate is growing by 10 – 15% per month [8].

The best choice of sequence comparison is using dynamic programming algorithms such as Smith-Waterman [17] and Needleman-Wunsch [10]. Since those algorithms are too slow for processing large sequence databases, fast heuristic algorithms have been developed such as BLAST [2] and PatternHunter [8][9]. Heuristic algorithms reduce sensitivity of alignment score and detail relationship between the compared sequences may not be established. Another approach to get optimal results of alignment process in shorter time is to utilize the benefits offered by High Performance Computing (HPC). Many approaches have been proposed for utilizing HPC techniques in the large scale sensitive sequence similarity search. HPC is a methodology to solve the high complexity problems such as computing the huge workload and data, and the intensive critical analysis. It reduces the computation time and, consequently the results are produced efficiently and decision making can be made much faster.

Previously, we have developed a new model that can speed-up the large scale DNA sequence similarity search and at the same time optimal alignment result is retained. The model is known as FRA-Search [12][13]. FRA-Search integrates automaton based filtering, rough sets theory based data classification and reduction and dynamic programming algorithm to give efficiency in producing the best possible DNA sequence similarity search results. This paper discusses the detailed on implementing the parallel FRA-Search model. The parallel FRA-Search model is implemented on the powerful PC-based cluster system. The MPJ Express software [3][4][5][6], a pure object-oriented Java Message Passing Interface (JMPI) is utilized to operate as communication interface protocol between processors.

The rest of the paper is organized as follows: The next subsection reviews a few related works on parallel DNA sequence similarity search. Section II discusses the developed FRA-Search model. Section III explains the detailed of the designing parallel FRA-Search model. Experimental results of the proposed model are presented in Section IV. Conclusion is placed in Section V.

A. Related Works

A number of techniques have been introduced to perform both fast and sensitive genomic sequence similarity search. A practical way to speedup the process of optimal similarity

Manuscript received April 1, 2009.

A. R. M. Nordin is with the Faculty of Informatics, Universiti Darul Iman Malaysia, KUSZA Campus, 21300 K Terengganu, Malaysia. (Phone: +6096653300; Fax: +6096673412)

M. S. M. Yazid and A. Aziz are with the Faculty of Science and Technology, Universiti Malaysia Terengganu, 21030 K Terengganu, Malaysia.

M. T. A. Osman is with the Kuliyyah of Information Technology and Communication, International Islamic University of Malaysia, 50728 Kuala Lumpur, Malaysia.

search is using parallel computing techniques. Both fine-grain and coarse-grain based parallel computing techniques are implemented in biological sequence similarity search. This section purposed to explain some of these techniques.

[11] propose parallel processing of optimal alignment between two sequences by exploiting parallel MPI/FORTRAN 90. The algorithm for optimal alignment is based on dynamic programming techniques. Two versions of algorithms have been developed: one versus one sequence alignment and one versus many sequence alignment. The second algorithm used "block" parallel dynamic programming algorithm and this technique will increase the amount of workloads done by each processor.

DSEARCH is a sensitive database sequence searching running under Java based distributed computing [14]. In DSEARCH, a sequence database is divided into dynamically sized units and sequence searching process is done over a client-server topology. Communication between clients and server is based on combination of Java RMI and ordinary Java sockets in which all machines (approximately 200 desktop PCs) in the systems are connected by a 100 Mbit/s speed network.

Grid computing is a technique for super scale processing speed. Combination of several PC clusters creates a hierarchical grid computing. [7] have implemented a DNA sequence alignment model under this hierarchical grid architecture. They used dynamic programming algorithm with linear space parallelism and is separated into two parts: parallelization of the similarity matrix and parallelization of the divide-and-conquer algorithm. Three clusters have been setup where each cluster has eight nodes. The clusters are connected by Ethernet switch where the bandwidth is about 8 MByte/s. Meanwhile the bandwidth between nodes in each cluster is about 190 Mbyte/s. The architecture of the software is based on two layers; upper layer uses MPICH-G2 and lower layer employs MPICH as a communication interface protocol.

FASTA is a heuristic based technique in sequence similarity search. Parallelization of FASTA has been implemented in the Grid Application Development Software (GrADS) project [18]. The GrADS adapts the master-worker paradigm, scheduling and rescheduling the tasks on an appropriate set of resources, launching and monitoring the execution. The GrADSoft scheduler makes a static schedule for its application where the whole or a portion of sequence databases are replicated on some or all of the grid nodes. The master will inform each worker which portions of database should be loaded into memory. The master also sends the input query sequence to each worker and collects the results from the workers.

II. THE FRA-SEARCH MODEL

The basic problem to be solved by the FRA-Search model

can be define as: Let $T = \{t_1, t_2, \dots, t_n\}$ be a collection of DNA sequences from a database and q is a DNA query sequence; let θ be a fixed score threshold and F be an alignment scoring function. By means of an optimal local alignment, A , of (q, t_i) , find $R \subseteq T$ where $\forall t_i \in R$ has score $F(A) \geq \theta$. The large volume of DNA sequence makes filtering process becomes as a prerequisite for efficient sequence similarity searching. Moreover, clustering the database is also believed can improve the efficiency of DNA sequence retrieval. The basic FRA-Search model consists of seven main stages: query initialization, patterns generating, patterns scanning, ranking, classification and reduction, optimal alignment and reporting. Specifically, generate patterns, patterns scanning and ranking stages are working as a filtering process. Figure 1 depicts the process flow of the FRA-Search model. The descriptions of stages in the model are as follows:

- Query initialization – biologist needs to input the new discovered DNA sequence (labeled as q).
- Patterns generating – a set of random patterns with length l characters are generated from q , and denoted as, $P = \{\rho_1, \rho_2, \dots, \rho_\delta\}$.
- Patterns scanning – by applying Aho-Corasick algorithm [1], patterns in P are scanned against the DNA sequences in T . Using BLOSUM62 scoring matrix, for each found pattern is given exact matching score. Once scanning all $\rho_i \in P$ in a t is completed, the total exact matching score (ϵ) for t is computed.
- Ranking – based on ϵ , DNA sequences in T are ranked using Quick-Sort algorithm.
- Classification and reduction – by using the indiscernibility relation provided by rough sets theory [15][16], the ranked DNA sequences are partitioned into equivalence classes. Based on lower and upper approximation condition, reduction technique is applied in order to reduce the size of DNA sequence dataset.
- Optimal alignment – based on the matching score, classification and reduction done before, the selected DNA sequences in T are aligned with q using Smith-Waterman algorithm [17]. The alignment score, W , and percent identity score (I) are generated for each pair of q and t .
- Reporting – an alignment report is generated for representing the similarity search results of query q .

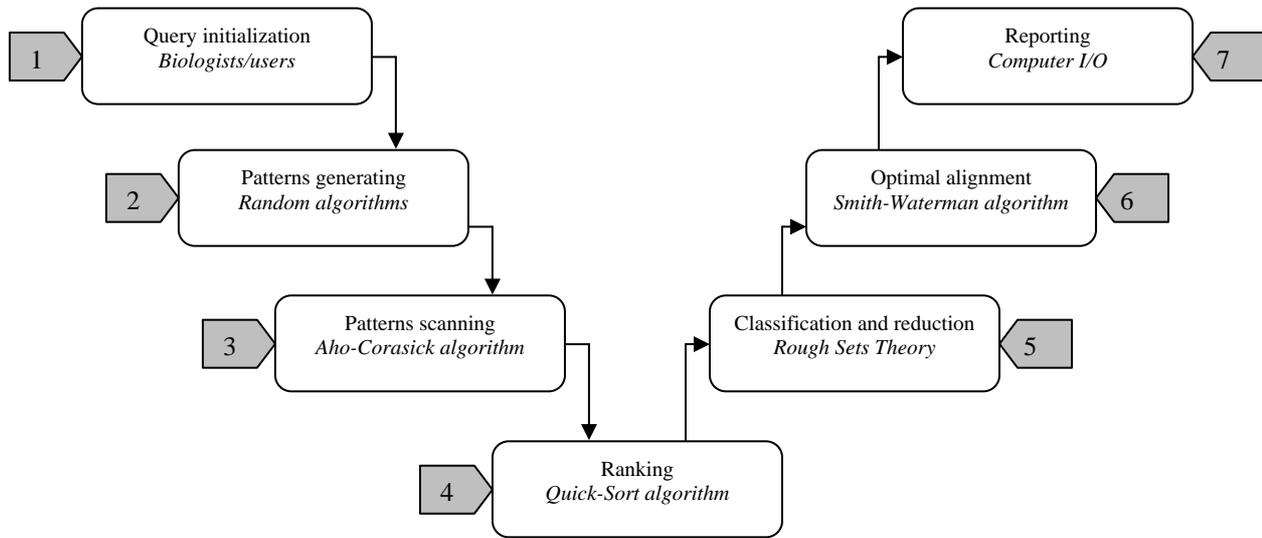


Figure 1: The process flow of FRA-Search model

In order to access and extract DNA database from GenBank or any other public DNA sequence databases, FRA-Search application utilized BioJava library [19]. BioJava is one of the bioinformatics projects that employs open source platform and is still under active development. The FRA-Search model is implemented using Java language, running under the Microsoft Windows XP Professional operating systems. The processor used was 1.86GHz Intel Core 2 with 1.0GB RAM. For the experiment purposes, a set of 30,000 DNA sequences are pickup randomly from five GenBank databases.

Table 1 shows the experiment result from 10 selected queries and λ is preset to 10 characters. The experimental results show that the proposed filtering technique can discard irrelevant DNA sequences from being computed in a dynamic programming based alignment process. Clustering the ranked sequences (generated by means of equivalence class) partitions the DNA database and therefore the efficiency and effectiveness of retrieval can be improved. The lower and upper approximation features are utilized to reduce the size of targeted DNA sequence dataset. This reduced dataset is known as the reduct dataset and it will represent the whole expected relevant DNA sequences in original dataset. This reduct dataset is then computed for optimal alignment using Smith-Waterman algorithm.

For instance in Query 2, the filtering process successfully groups the 3,359 relevance sequences to the query. However, after the classification and reduction process only 1,321 sequences are selected to be computed by dynamic programming based alignment. These 1,321 sequences

represent the characteristics of the entire relevant sequences in the generated classes. The combination of filtering process and rough sets theory technique effectively removed 28,679 database sequences (95.60%) from being computed in the rigorous optimal alignment process. Another example is from Query 8. After targeted DNA sequences are filtered, there are 26,607 (88.69%) have been determined for not relevant to query sequence. Therefore, those irrelevant targeted DNA sequences are skipped from being executed for Smith-Waterman algorithm. There are only 3,393 DNA sequences that are considered for clustering. However after those sequences have undergone for reduction process, 1,865 DNA sequences are left and considered as a reduct set. All members in this reduct set are considered for representing the entire relevant sequences in all equivalent classes. Finally, all DNA sequences in this reduct will be computed for optimal alignment process. The combination of filtering process and rough sets theory techniques successfully reduced the size of the original targeted DNA sequences dataset where 28,135 (93.78%) sequences are skipped from being executed for optimal alignment process.

Overall, the FRA-Search model gives high efficiency of DNA sequence similarity search process with low computational numbers for $O(n \times m)$ time and space complexity of Smith-Waterman algorithm. Hence, the time for retrieving a set of similar DNA sequences from a database to a query is minimized.

Table 1: Applying rough sets theory in F-R-A model experiment results

Query #	Automaton based filtering		Rough sets theory classification and reduction	
	Relevance	Smith-Waterman	Smith-Waterman	Total Smith-Waterman

	<i>sequence (T*)</i>	<i>reduced (from T)</i>	<i>reduced (form T*)</i>	<i>reduced (from T)</i>
Query 1	3,247	26,753 (89.18%)	1,652 (50.88%)	28,405 (94.68%)
Query 2	3,359	26,641 (88.80%)	2,038 (60.67%)	28,679 (95.60%)
Query 3	3,159	26,841 (89.47%)	1,698 (53.75%)	28,539 (95.13%)
Query 4	4,055	25,945 (86.48%)	2,501 (61.68%)	28,446 (94.82%)
Query 5	3,138	26,862 (89.54%)	1,532 (48.82%)	28,394 (94.65%)
Query 6	3,511	26,489 (88.30%)	2,108 (60.04%)	28,597 (95.32%)
Query 7	3,190	26,810 (89.37%)	1,657 (51.94%)	28,467 (94.89%)
Query 8	3,393	26,607 (88.69%)	1,528 (45.03%)	28,135 (93.78%)
Query 9	3,290	26,710 (89.03%)	1,751 (53.22%)	28,461 (94.87%)
Query 10	3,084	26,916 (89.72%)	1,494 (48.44%)	28,410 (94.70%)

III. THE PARALLEL FRA-SEARCH MODEL

The key crisis of large scale DNA sequence search algorithm is time and space complexity. The implementation of computational DNA sequences on a parallel computing system has developed a new application domain in computer science studies. Parallel computational biology application is always associated with HPC, where multiple computers are used to run the application. The real challenge in successful parallel execution is the handling of data and operations dependences. In other words, the parallel application must be engineered with suitable machine communication protocol and parallel architecture.

A. The Machines Communication

Parallel computation frequently uses a standard communication interface protocol to send instructions and exchange information or data between processors. In the parallel FRA-Search model, the MPJ Express software [3][4][5][6], a pure Java interface for MPI is used to enable messages to be exchanged between processors. The MPJ Express functions support point-to-point communication, group communications, synchronization etc. The parallel FRA-Search model can be viewed as a three packages interaction: FRA-Search application tool, BioJava library and MPJ Express library (Figure 2). The MPJ Express software can be run under the LINUX or Microsoft Windows

operating systems. The parallel FRA-Search model is configured under the Microsoft Windows XP professional.

B. The Architecture

The proposed parallel FRA-Search model has been mapped onto the Single Program Multiple Data (SPMD) architecture. Each processor in SPMD architecture has read and writes access to its own memory. In SPMD, there is no local control over parameters of the execution in each processor and all the processors have to be active at all times.

A PC-based cluster system is used to implement parallelization of the processes. In FRA-Search, the developed cluster can be viewed as a process-farm based distributed memory multiprocessors. The process-farm paradigm consists of a master processor which controls a set of worker processors. Figure 3 exhibits the parallel FRA-Search architecture where MPJ Express library supports for communication among processors. As a pilot implementation, the designed model employed a cluster of eight PCs with Intel Pentium Core 2 1.86GHz, DDRAM 1.0GB. The machines are connected with 16-port Gigabit 3Com 10/100/1000 switch. Master processor is installed with BioJava library, MPJ Express software and parallel source code as well. The DNA database is replicated to all processors.

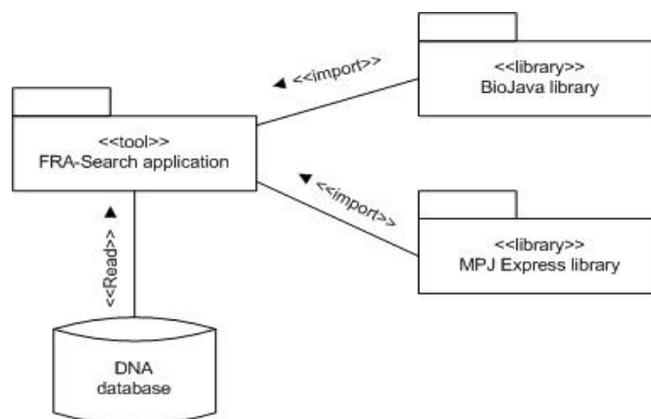


Figure 2: The packages interaction

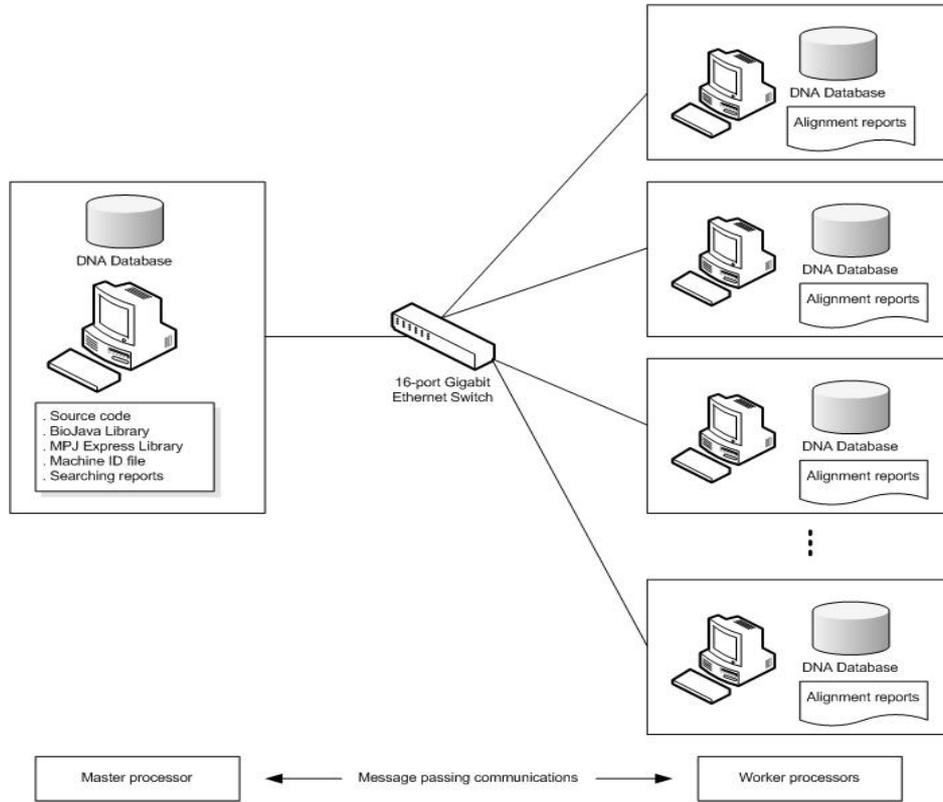


Figure 3: The parallel FRA-Search architecture

C. The Specifications and Algorithms

During the implementation, one PC is allocated for master processor and the rest are worker processors. Each processor is assigned a distinct process identification number upon a JMPI is initialized. Basically, two phases in FRA-Search model are considered to be run under parallel computation mode; patterns scanning and optimal alignment. All processes in those phases are scheduled by master processor and will be distributed to worker processors. In addition, the master processor is responsible for controlling of the application. Specifically, processes allocation scheme in the parallel FRA-Search model can be signed as:

Master := {read query, patterns generating, ranking, generate final report}
 Worker := {patterns scanning, optimal local alignment, generate alignment structure report}.

The DNA sequence database has been replicated to all processors local hard disk and therefore logical partitioning can be applied. Based on machine identification number, the logical partitioning process will be executed automatically by each worker processor. Using this approach, communication overhead between master processor and worker processors can be minimized. If a set of workers is $W = \{w_1, w_2, \dots, w_k\}$, then a divide and conquer approach is applied where DNA sequence database is partitioned into k , thus each worker will have their own dataset. If T is a sequence database, the partitioning process can be denoted as:

$$T = \sum_{i=1}^k t_i \quad (1)$$

where t_i is a block of DNA sequences for w_i . The first job of the master processor is to receive the query sequence from the users/biologists. This query sequence will be sent to worker processors and a set of patterns will be generated. Then each worker processor will calculate its own data partitioning parameters, $\langle S_i, E_i \rangle$ where $i = 1, 2, \dots, k$, S is the starting index of sequence block in the database, E is the ending index of sequence block in the database and k is the number of worker processors. According to parameters $\langle S_i, E_i \rangle$ each worker processor will read the DNA sequence database. For each sequence retrieved, patterns scanning process is executed and a total exact matching score is calculated. Finally, all the total exact matching scores are sent to master processor. Assume that C is exact matching process, and if Y is a set of exact matching scores generated by C process then the computation by each worker can be signed as,

$$C(t_i) = Y_i \quad (2)$$

After receiving all the exact matching scores from worker processors, the master processor will combine of those results. The combination process can be viewed as:

$$C(T) = \bigcup_{i=1}^k C(t_i) \quad (3)$$

where \bigcup denotes a combination operation. Based on the Y values, the ranking process will be executed by the master processor. Further on, master processor classified those ranked sequences based on indiscernibility relationship and a reduct set of DNA sequences is produced. In the next stage,

the master processor will send a pair of query and targeted DNA sequences from the reduct set to the idle worker processor. The worker processor executes optimal alignment process and generates alignment structure report as well. The process of sending jobs to worker processors will be iterated until a targeted number of similar DNA sequences to query are retrieved. If S is a set of alignment scores and D is a set of percent identity scores generated by an optimal alignment process, A , then the computational result by each worker can be represented as:

$$A(q, t_i) = \langle \Omega_i, I_i \rangle \quad (4)$$

where $\Omega_i \in S$ and $I_i \in D$. If T^{**} a reduct set of targeted DNA sequences, then the combination of those results can be viewed as operation:

$$A(q, T^{**}) = \bigcup_{i=1}^j A(q, t_i) \quad (5)$$

where \bigcup denotes a combination procedure and j is the number of iterations occurred. Based on this combination, the final report of similarity search process will be generated by the master processor. The general algorithms for main method, patterns scanning and optimal alignment processes in parallel FRA-Search model are depicted in Figure 4, Figure 5 and Figure 6 respectively.

Algorithm: *The main method for parallelization*

```
L1. Initialize for MPJ Express and acquiring for parallel variables
L2. Waits for all processors ready for executing jobs

// starts for exact patterns matching process
L3. exactMatching.patternsSearch();

L4. Waits for all processors ready for executing the next phase

// starts for optimal local alignment
L5. inexactMatching.alignmentProcess()

// master processor performs for result analysis
L6. appAnalysis.readRecords(numberOfSWPerformed)

L7. Terminate MPJ Express
```

Figure 4: The main method for parallel F-R-A model

Algorithm: *Parallel patterns scanning*

```
L1. If (master_processor)
    L2. Read query
    L3. Send query to all worker processors
L4. Receives patterns scanning results from all worker processors
    L5. Performs for ranking process
L6. Else if (worker_processor)
    L7. Receives query from master processor
    L8. Calculates data block parameters, <S, E>
    L9. Generates patterns from query
    L10. For a ← S to E do
        L11. ACMachine.find(ta)
        L12. Calculates total exact matching score
    L13. Sends result to master processor
```

Figure 5: Parallel patterns scanning algorithm

Algorithm: *Parallel optimal local alignment*

```
L1. If (master_processor)
    L2. Read the reduct DNA sequence dataset
L3. If (master_processor)
    L4. Sends the first block of sequences to the idle worker processors
L5. If (master_processor)
    L6. While (yesdata is true)
        L7. Receives alignment result from worker processors
        L8. Sends a new DNA sequence to worker processor
L9. End if (worker_processor)
    L10. Recieves DNA sequence from master processor
    L11. Performs optimal local alignment
    L12. Calculates alignment score and identity
    L13. Sends result to master processor
```

Figure 6: Parallel optimal local alignment algorithm

IV. THE EXPERIMENTS

We have evaluated the model with seven different sizes of dataset; 1,000 sequences (1K), 5,000 sequences (5K), 10,000 sequences (10K), 15,000 sequences (15K), 20,000 sequences (20K), 25,000 sequences (25K) and 30,000 sequences (30K). Throughout the performance evaluation, the model attempts to retrieve 10% DNA sequences in database that are satisfying the similarity score threshold value. Since all PCs in the cluster have the same technical specifications, we can assume that all machines will work at the same level of performance.

A. The Processing Time and Speedup

Table 2 shows the processing time for three selected queries. Excluding for two worker processors, the parallel FRA-Search model processing time is better overall. For all DNA sequence datasets, elapse time seem to exponential decrease when more worker processors are considered. For instance, Query 1, in case of 30,000 DNA sequences dataset, the serial FRA-Search model takes for 570 seconds to retrieve 3,000 sequences that similar to the query. The processing time is increased to 615 seconds when processed in parallel mode using two worker processors. However, the processing time becomes decreasing to 452 seconds, 347 seconds, 283 seconds, 248 seconds and 211 seconds with three, four, five, six and seven worker processors respectively.

The performance of the parallel computing model can be evaluated by calculating its speedup. Speedup shows how much of an improvement is practically possible in the best case, ignoring overhead and communication costs. Assume that the speed of the processors and network is constant; the speedup of k processor(s) is calculated using the following equation:

$$S(k) = \frac{T_1}{T_k} \quad (6)$$

where T_1 is the time required to execute an equivalent sequential program on one processor and T_k is the time required to execute the parallel version of the program on k processors. Using the experiment results tabulated in Table 2, Figure 7 shows the speedups of Query 3. The experimental results show that parallel FRA-Search model exhibits good

speedups when using three worker processors and above. Negative speedup occurred when two worker processors are used. Its happen because of the granularity of the work assigned to each worker processor is decreased. Here, longer time is spending by worker processors to get the input on which to work rather than actually working on it.

Table 2. Processing time

k	q	Database size						
		1K	5K	10K	15K	20K	25K	30K
1	1	36s	201s	358s	399s	461s	545s	570s
	2	50s	264s	465s	511s	594s	701s	744s
	3	43s	242s	420s	490s	573s	660s	727s
2	1	35s	176s	317s	405s	500s	586s	615s
	2	42s	224s	398s	504s	612s	726s	777s
	3	38s	210s	373s	475s	600s	703s	767s
3	1	25s	115s	223s	278s	343s	405s	452s
	2	28s	148s	266s	335s	414s	506s	550s
	3	28s	145s	257s	326s	403s	488s	546s
4	1	18s	88s	164s	217s	259s	310s	347s
	2	23s	112s	204s	264s	329s	380s	422s
	3	20s	108s	195s	248s	311s	377s	409s
5	1	14s	72s	134s	171s	211s	259s	283s
	2	19s	91s	163s	211s	265s	316s	339s
	3	17s	85s	159s	203s	257s	309s	331s
6	1	13s	60s	113s	148s	179s	217s	248s
	2	17s	79s	144s	177s	227s	291s	228s
	3	15s	73s	137s	172s	213s	256s	288s
7	1	11s	53s	101s	128s	160s	192s	211s
	2	14s	65s	121s	159s	192s	228s	255s
	3	13s	64s	116s	146s	185s	226s	248s

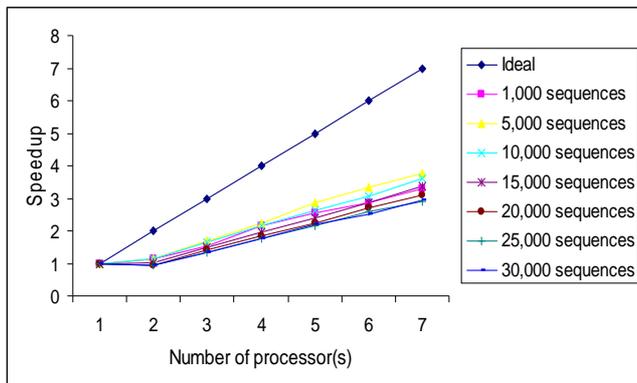


Figure 7: Query 3 – speedup

A. The Parallel System Efficiency

The efficiency of parallel communication can be calculated using the property of:

$$E = \frac{T_1}{k \times T_k} \times 100\% \quad (7)$$

where T_k denotes the processing time for k processors and T_1 indicates the processing time for single processor. The best possible for efficiency occurred when all worker processors are being used on the computation at all times. From the three queries before, Table 3 shows the efficiency results of the parallel FRA-Search model based on different number of worker processors where the size of dataset used is 30,000. Table 4 shows the efficiency of the parallel FRA-Search model for seven different sizes of dataset and the number of worker processor used is seven. From the tabulated results, most of the parallel systems efficiency values are below 50%. These figures indicate that many

worker processors are being used less than half the time on the actual computation. In other words, much of the time involved for completing the whole processes in parallel FRA-Search model are used for communication among processors. The results also show that the efficiency of parallel FRA-Search model decreases with the incremental of k and dataset size. It is happen due to the increasing of communication overhead. Definitely that with the increasing of k , the communication channel of master processor is also increased.

Table 3: The efficiency of parallel FRA-Search model – worker processors number

q #	Efficiency (%)					
	2 wkrs.	3 wkrs.	4 wkrs.	5 wkrs.	6 wkrs.	7 wkrs.
1	46.34	42.04	41.07	40.28	38.31	38.59
2	47.88	45.09	44.08	43.89	41.47	41.68
3	47.39	44.38	44.44	43.93	42.07	41.88

Table 4: The efficiency of parallel FRA-Search model – dataset size

q #	Efficiency (%)						
	1K	5K	10K	15K	20K	25K	30K
1	46.75	54.18	50.64	44.53	41.16	40.55	38.59
2	51.02	58.02	54.90	45.91	44.19	43.92	41.68
3	47.25	54.02	51.72	47.95	44.25	41.72	41.88

Communications between master processor and worker processors are rapidly occurred during optimal alignment process. For each alignment process, master processor interacts with worker processor twice; for sending the input and receiving the output. In cluster and distributed memory systems, the time for sending and receiving between processors depends on software parameters. Type of message transfer and structure of message stored in the machine memory are categorized as software parameters. If large size of messages involved in sending-receiving processes, the communication traffic is very dense. The algorithm in the FRA-Search model and MPJ Express software used an array data structure to store all information related to the primitive operations in message passing system. This data structure is considered as a static memory allocation and cannot be resized during execution time. In addition, sending query and targeted sequences represented by thousands base pairs (characters) and therefore a huge size of memory and time is required.

V. CONCLUSION

Massive volumes of biological sequences data are available worldwide and their size is still growing at exponential rate. This situation makes the sequence homology searching process becomes more complex and requires longer time and bigger space. Therefore, speeding up the process is an important problem in this application domain. Parallel computing techniques are used to solve the problem of time and space complexity. In addition, the independence of biological sequences in the databases makes parallel processing platform for similarity search very appropriate. We have developed a model that can perform efficient DNA sequence similarity search called FRA-Search. The distributed memory PC-based cluster system is used to implement this parallel version of FRA-Search model. To support this processing system a master-workers paradigm is

selected. The MPJ Express software is used as a communication interface protocol between processors. Experiments show that, the combination of the proposed filtering, classification and reduction and parallel computing techniques is very appropriate for DNA sequence similarity search.

There are two issues can be considered in order to improve the developed parallel FRA-Search model:

- A better load balancing strategy needs to be studied, and therefore parallel computation efficiency can be improved;
- The concept of multiple master processors can be employed and therefore the problem of master processor bottleneck can be avoided.

REFERENCES

- [1] Aho, A. V. and Corasick, M. J. "Efficient String Matching: An Aid to Bibliographic Search", *Communication of the ACM*, Vol. 18(6), 1975, 333 – 340.
- [2] Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs", *Nucleic Acids Research*, 25, 1997, 3389 – 3402.
- [3] Baker, M., Carpenter, B. & Shafi, A, "Cluster Computing and Grid 2005 Works in Progress: A Pluggable Architecture for High-Performance Java Messaging", *IEEE Distributed Systems Online*, vol. 6, no. 10, 2005.
- [4] Baker, M., Carpenter, B. & Shafi, A. "MPJ Express Meets Gadget: Towards a Java Code for Cosmological Simulations", *Special Session ParSim in Euro PVM/MPI 2006*, Bonn, Germany, 17-20 September 2006.
- [5] Baker, M., Carpenter, B. & Shafi, A. "An Approach to Buffer Management in Java HPC Messaging", In V. Alexandrov, D. van Albada, P. Sloot, and J. Dongarra, editors, *International Conference on Computational Science (ICCS 2006)*, LNCS. Springer, 2006
- [6] Baker, M., Carpenter, B. & Shafi, A. "MPJ Express: Towards Thread Safe Java HPC", *IEEE International Conference on Cluster Computing (Cluster 2006)*, Barcelona, Spain, 25-28 September, 2006.
- [7] Chen, C. and Schmidt, B. "An Adaptive Grid Implementation of DNA Sequence Alignment", *Future Generation Computer Systems*, 21, 2005, 988 – 1003.
- [8] Li, M. and Ma, B. "PatternHunter II: Highly Sensitive and Fast Homology Search" *Genome Informatics*, 14, 2003, 164 – 175.
- [9] Ma, B., Tromp, J. and Li, M. "PatternHunter: faster and more sensitive homology search", *Bioinformatics*, 18(3), 2002, 440 – 445.
- [10] Needleman, S. B. and Wunsch, C. D. "A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Sequences", *Journal of Molecular Biology*, 48, 1970, 443 – 453.
- [11] Nguyen, E. N. D., Nguyen, D. N., Nguyen, D. T. and Tungkahotara, S. "Comparing DNA Sequences by Dynamic Programming in Sequential and Parallel Computer Environments", *Proc. of the 2006 WSEAS International Conference on Mathematical Biology and Ecology*, 2006, 146 – 153.
- [12] Nordin, M. A. R., Yazid, M. M. S, Aziz, A. and Osman, A. M. T. "Automaton Based Filtering in Optimal DNA Sequence Similarity Search", *Proc. of the 1st Regional Conference On Computational Science and Technology*, 2007, 478 – 482.
- [13] Nordin, M. A. R., Yazid, M. M. S, Aziz, A. and Osman, A. M. T. "DNA Sequence Database Classification and Reduction: Rough Sets Theory Approach". *Proc. of the 2nd International Conference on Informatics*, 2007, 41 – 48.
- [14] Page, A. J., Keane, T. M. and Naughton, T. J. "Bioinformatics on a Heterogeneous Java Distributed System", *Proc. of the 19th IEEE International Parallel and Distributed Processing Symposium*, 2005, 184.1
- [15] Pawlak, Z. "Rough Sets". *International Journal of Computer and Information Sciences*, 11, 1982, 341 – 356.
- [16] Pawlak, Z., Grzymala-Buse, J., Slowinski, R. & Ziarko, W. "Rough sets". *Communication of the ACM*, 38(11), 1995, 88 – 95.
- [17] Smith, T. F. and Waterman, M. S. "Identification of Common Molecular Subsequences", *Journal of Molecular Biology*, Vol. 147, 1981, 195 – 197.
- [18] YarKhan, A. and Dongarra, J. J. "Biological Sequence Alignment on the Computational Grid Using the GrADS Framework", *Future Generation Computer Systems*, 21, 2005, 980 – 986.
- [19] BioJava Web Site, <http://biojava.org>, Accessed on January 2008