ISSN 2047-3338

Multiple Sequence Alignment on the Grid Computing using Cache Technique

Le Van Vinh[1], Tran Van Lang[2], Nguyen Thi Thu Du[2] and Vo Hong Bao Chau[3]

***Abstract*—Multiple sequence alignment is an important problem and popular in the molecular biology. This is a basic problem that its solution could be used to proof and discover the similarity of the new sequence with other exist sequences; to define the evolution process of the family's sequences; as well as to support the protein structure prediction, etc. In this paper, we consider and improve the global progressing algorithm by using the cache storage technique to make the best of the previous alignment results. This algorithm and cache technique were been developed on the distributed system and Grid computing environment in order to decrease the algorithms execution time, as well as increase the quantity and size of input sequences.**

***Index Terms*—Biological Sequences, DNA, Protein, Grid Computing and Distributed Computing**

I. INTRODUCTION

MULTIPLE Sequence Alignment (MSA) is a sequence alignment of three or more biology sequences such as DNA, RNA, or protein. The result of the task can be used to infer sequence homology and conduct phylogenetic analysis to assess the sequences shared evolutionary origins [2], [3]. The accuracy and execution time are major factors requiring the attention of researchers. Some popular approaches used for MSA algorithms are exact solution, progressive methods, iterative methods, or methods based on Hidden Markov Models. Each method has its advantages and disadvantages. Biologists are the persons who decided suitable method to process their biological data.

The multiple sequence alignment is the problem with exponential complexity. Over the years, researcher efforts in finding different algorithms or mathematical models that require low computational cost as well as ensure accuracy. A lot of popular algorithms were proposed such as ClustalW, T-Coffee. However, when the number of sequences increases, the software's processing speed becomes slow because the limitation of a single computer. Thus, parallel and distributed systems are used to gain the efficiency for the programs.

[1]HCM City University of Technical Education, Vietnam Ministry of Education and Training, vinhlv@fit.hcmute.edu.vn

[2] Institute of Applied Mechanics and Informatics, Vietnam Academy of Science and Technology, {tvlang,ntthudu}@vast-hcm.ac.vn

[3] Lac Hong University, Vietnam Ministry of Education and Training

In the field of applying parallel, Grid or Cloud computing to solve the MSA problem, last decade witnessed many contributions. ClustalW-MPI [14] is the parallel implementation of ClustalW software, using MPI (Message Passing Interface) library. It can be deployed on workstation clusters with distributed memory architecture. MAFFT [15] is also modified to parallelized version by using the POSIX Threads library [16]. Other researchers aimed to get the usefulness of Grid or Cloud computing for their improvement [17], [18].

Cache technique is another strategy applied in MSA algorithms. It helps programs to reuse the results of previous works so that they could decrease execution time. Zola [5]is the first person use the technique to improve MSA algorithms. He proposed an MSA program and developed it with CaLi cache library [7]. Xun Tu et al. [19] also proposed a new caching technique with two novel cache replacement policies and did some experiments in progressive algorithms. However, their approach was not deployed on distributed system.

This paper presents the research results on applying cache technique for the global progressing algorithm to reduce computational cost. In addition, it also describes the algorithm and cache technique on Grid computing environment to decrease execution time.

The paper's content includes 4 sections. In the first and second section, the introduction and related works were described; the Pair-wise Sequence Alignment (PSA) and Multiple Sequence Alignment algorithms were discussed more detail. The main content of paper is the section 3, where the reasons and works were done to improve the global progressing algorithms was presented. The last sections show the experimental results and the discussions.

II. MULTIPLE SEQUENCE ALIGNMENT ALGORITHM

The sequence alignment algorithms can be classified into two types: the Pairwise Alignment and the Multiple Alignment. The Pairwise Alignment algorithm is commonly used as the first step in MSA algorithm.

A. Pairwise Sequence Alignment

In biology, the evolution of organisms is caused by many different factors. However, the major factor is the change in the structure of DNA/RNA including insertion or deletion

mutation, leading to protein structural change. The Pairwise Sequence Alignment algorithms can help biologists discover the changes. Thus, they can find out the relationship between two individual organisms. This problem is fundamental for many other problems in bioinformatics such as finding homologous sequences, motif finding or MSA.

For example, there are two sequences with different length:

```
CGACTAAATCAAGCCGCCTTTATTGCCTCACGCCCAGGGGTCTTTTCG
CGACTTCATCAAACCATTTATTGCTACTCGTCCGGGAGTTTTTACG
```

The one possible alignment can be seen as follows:

```
CGACTAAATCAAGCCGCCTTTATTGCCTCACGCCCAGGGGTCTTTT-CG
CGACTTCATCAAAC--CATTTATTGCTACTCGTCCGGGAGT-TTTTACG
```

The Pairwise Sequence Alignment algorithm arranges and finds similar regions between two sequences. In some regions, the special characters, called gap "−", are added. There are two groups of those algorithms such as the global alignment methods (Fig. 1) and the local alignment methods (Fig. 2).

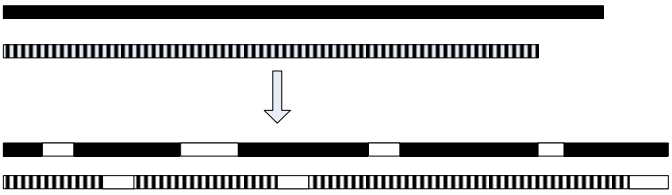


Fig. 1. Global Alignment Algorithm

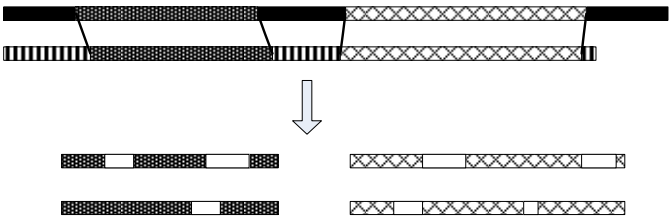


Fig. 2. Local Alignment Algorithm

The global alignment algorithms aim to align two sequences in their entirety how to get the largest number of matched elements and the least number of gaps. Some of the popular global sequence alignment algorithms are Needleman-Wunch [8], GLASS [12]. In contrast, the local alignment algorithms [1] attempt to find the best similar subsequence within two sequences. Some the local alignment algorithms are well-know such Smith-Waterman, FASTA, and BLAST.

B. Multiple Sequence Alignment

The problem of Multiple Sequence Alignment are defined as follows:

Given n sequences S_1, S_2, …, S_n, multiple sequence alignment is obtained by inserting gap characters ("−") in each sequence to make them all the same length l.

Thus, the sequences can be stored to an array with n row and l column. Each item of array is the character standing for the four nucleic acid bases (A, C, G, T) or gap character ("−"). There exist several strategies are used in MSA algorithms:

C. Exact algorithms

The algorithm is based on the Needleman-Wunsch algorithm [8]. It always produces the optimal alignment by using a dynamic backtracking algorithm. However, the weakness of the algorithms is that the amount of time and memory requiring tends to grow exponentially with the number of sequences. There are some MSA algorithms uses this strategy including MSA, DCA.

D. Progressive algorithms

The progressive algorithm (such as ClustalW) is known as the simplest and most effective strategy in case of aligning sequences in little time and with little memory[3], [6]. The algorithm obtains to find the sequences closely related by using guide tree. It was initially proposed by Hogeweg [10] and later improved by Feng-Dolittle [9] and Taylor [11]. The algorithm's idea is as follows: The first step is to perform pairwise sequence aligning for all pairs of sequences. In the next step, guide tree is constructed and used in the processing step. In this step, all sequences are aligned based on guide tree. This algorithm is used in our research to illustrate the effectiveness of the use of the cache and Grid computing technique.

E. Iterative algorithms

The algorithms work similarly to progressive strategy but repeatedly align the initial sequences and adding new sequence to grow the number of sequence in alignment process. It aims to reduce the inherent errors occurring in progressive method. This method is commonly used for local alignment algorithms. MUSCLE is known as the popular multiple sequence alignment using iterative method.

F. HMM-based algorithms

Hidden Markov Model is method using probabilistic model can produces both global and local alignment. The algorithms offer significant improvements in computational cost. Several HMM-based MSA software are popular such as POA, SAM.

III. CACHE TECHNIQUE AND GRID COMPUTING FOR PROGRESSIVE ALGORITHMS

In this paper, the progressive algorithm was improved by appling cache technique in order to reduce computational cost.

A. Progressive Algorithm

There are three basic steps as follows in this algorithm:

- *Step 1: Pairwise sequence alignment*

This step aims to align all pairs of sequences and determine the similarity between those sequences. Some pairwise sequence alignment algorithms can be used in this step, for instance, Sum of pairs, Linear – Sankoff. For each pair of sequences, the algorithm produces a score value that shows the distance between two sequences. A distance matrix $n \times n$ (n is the number of sequences) is used to store all values.

- *Step 2: Construct a guide tree*

Create guide tree by using a bottom-up clustering method NJ (Neighbor-Joining) proposed by Saitou and Nei. This

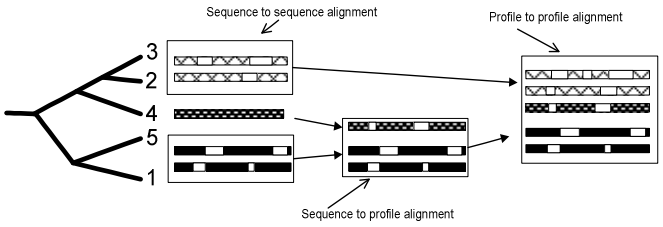


Fig. 3. Progressing

phonetic tree shows the relationship between all given sequences.

- *Step 3: Progressing*

Using guide tree, group the sequences closely relative together first. Then, combine the group closely relative together. The process repeated until all sequences are grouped. We have three kinds of grouping (Fig. 3): Sequence to sequence, sequence to profile, and profile to profile. (Profile is a group of sequences closely relative).

B. Cache Technique

The cache is a collection of copies of objects accessed frequently by users. The objects are memory pages or the site, the records of the database or files. Users may be operating systems or computing program. One common application of this technique is web cache. By using proxy servers or browser cache, the documents (or data) located temporal locality or spatial locality are brought closer to users. Thus, it reduces response time required by users significantly. However, one of the major factors that affecting the system's performance is cache replacement policy when the cache is full. The Cache replacement policy aims to get optimal parameters such as ratio of document found, latency time, total cost.

The other important feature of the cache system is data synchronization so that the users always access the latest documents (or data). This technique is similar to the one used to synchronize files or data of distributed database systems. The main feature is concerned in data synchronization process is data consistency. It is categorized into two types: weak consistency and strong consistency [4].

The weakness of cache technique is that it uses large amount of storage space. So depending on the system's capabilities and users' needs, they will decide whether to use the technique or not.

In this modified algorithm, the progressing step (step 3) reuses the result of the Pairwise Sequence Alignment step (step 1). For instance, in the first test there are five sequences 1, 2, 3, 4, 5. In step 1, the following pairs of sequence are aligned: (1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4), (2, 5), (3, 4), (3, 5), (4, 5). Then the guide tree is constructed (Fig. 3). In step 3, two pairs of sequence (1, 5), (2, 3) are realigned.

In addition, to align six sequences 1, 2, 3, 4, 5, 6 (sequences from 1 to 5 have aligned in the first test) there are following steps: Step 1 needs to align 15 pairs of sequences: (1,2), (1,3), (1,4), (1,5), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5), (1,6), (2,6), (3,6), (4,6), (5,6). In this case, the 10 pairs of sequences have to be realigned: (1,2), (1,3), (1,4), (1,5), (2,3), (2,4), (2,5), (3,4), (3,5), (4,5).

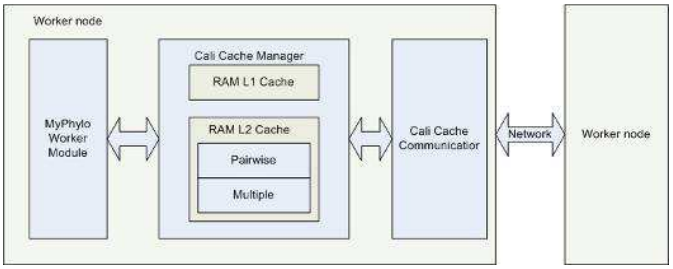


Fig. 4. The Cache system [7]

Two mentioned above examples give a motivation to use cache technique so that it reuses the already aligned pairs in the previous steps or executed application.

To demonstrate above ideas, the algorithms were modified from available source code, using CaLi library [7] and MSA program using the cache technique proposed by Zola [5]. Moreover, we developed for global progressing algorithm, and deployed them the grid computing system.

The cache system architecture is described in Fig. 4. Each worker has a CaLi cache manager that manages its cache and communicates with other cache manager through CaLi cache communicator. The CaLi cache is deployed using MPI-2 (Message Passing Interface standard 2) library, so it could be run on both the parallel system and the Grid computing system. The CaLi cache uses hash table structure to store and retrieve data. There are two level of cache: L1 cache and L2 cache. Whereas the L1 cache allows program to store and retrieve data on internal memory, the L2 cache allows program to do on external memory. Depending on the system's capability, users will choose to use L1 cache, L2 cache or both of them.

C. Grid computing

Grid computing is known as a distributed system that connects many computer systems having different hardware and platforms (operating system, system software). It allows applications to run in parallel on multiple machines, clusters, or systems (VO - Virtual Organization). The system is suitable for solving the problems that require a large amount of computation as well as storage capacity. In our research, the Grid system is used for execution bioinformatics problems.

IV. EPERIMENTAL RESULTS

In this paper, the multiple sequence alignment using cache technique on IOIT-HCM Grid System. This system is connected to PRAGMA system (Fig. 5). There are two clusters with two head nodes: venus.ioit-hcm.ac.vnand moon.ioit-hcm.ac.vn. Each cluster includes 5 nodes (1 head node and 4 worker nodes).

In these experiments, the data from the Data Bank NCBI (National Center for Biotechnology Information) were used. The system provides resources for researcher in the field of Bioinformatics and computational biology.

A. First data

In this test pattern, the program for 60 sequences was executed. Each sequence contains 800 characters. The result shows as follows (Fig. 6a, 6b):

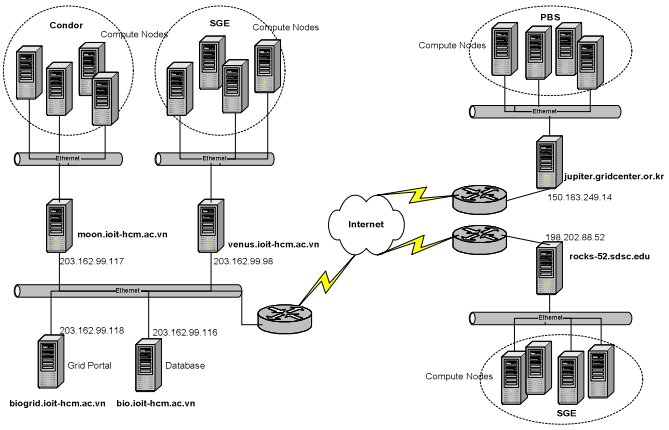


Fig. 5. IOIT-HCM Grid System

Using parallel algorithm without cache, the time is 344,42 sec ≈ 5,74 min.

Using parallel algorithm with cache, the time is 270,17 sec ≈ 4,5 min.

B. *Second Data*

In the second test data, the program for 90 sequences with 800-character length was executed. 60 sequences in this test have used in the first data.

```
Score : 4281.5
  Score: 9672.500000Gia tri i,
Vao vong lap lan thu 112 , 113
Score : 6398.5
Score : 9672.5
  Score: 16557.000000Gia tri i,
Vao vong lap lan thu 103 , 114
Score : 16557
Score : 2672
  Score: 19783.500000Gia tri i,
Vao vong lap lan thu 107 , 115
Score : 19783.5
Score : 3834.5
  Score: 24180.500000Gia tri i,
Vao vong lap lan thu 111 , 116
Score : 24180.5
Score : 3732
  Score: 28519.500000Gia tri i,
Vao vong lap lan thu 104 , 117
Score : 28519.5
Score : 2725.5
  Score: 31879.500000Gia tri i,

------------Ket thuc Processing
Ket thuc Progressive !!!
TIME CHECK:
----------
total: 344.421s
       0h:5m:44s
```

Fig. 6a. Parallel execution

```
Duong dan Cache: 5f/0001A12111.23.
.1166931938.0
  Score: 23589.500000Gia tri i, j:
Vao vong lap lan thu 107 , 117

Step 6 -----CACHE_L2_1-----:

Step 7:
Score : 23589.5
Score : 4309

Luu vao Cache

Duong dan Cache: 2a/0001A12111.51.
1166931942.0
  Score: 28457.500000Gia tri i, j:

------------Ket thuc Processing---
Ket thuc Progressive !!!
TIME CHECK:
+ L1 cache initialize: +0.273986s
- L1 cache initialize: +0.03466s
+ L1 replica: +197.819s
- L1 replica: +2.01737s
+ L2 cache initialize: +0.012555s
- L2 cache initialize: +0.011217s
----------
total: 270.17s
       0h:4m:30s
```

Fig. 6b. Parallel execution with cache

```
8.5 7 7 7.5 14 7
A C G T -
9.5 10.5 9.5 8.5 16.5 8.5
A C G T -
8.5 6.5 6.5 8.5 13.5 6.5
A C G T -
7 7.5 6.5 5 13 5
A C G T -
10.5 9.5 10.5 9.5 16 9.5
A C G T -
6.5 8 7 6.5 13.5 6.5
A C G T -
10 10 10 10 6 6
A C G T -
10 10 10 10 6 6
A C G T -
6 6 6 6 2 2
A C G T -
7.5 8.5 7.5 8.5 9.5 7.5
A C G T -
6 6.5 6.5 5 7 5
  Score: 47386.000000Gia tri i,

------------Ket thuc Processing
Ket thuc Progressive !!!
TIME CHECK:
----------
total: 727.784s
       0h:12m:7s
[guser@moon MyPhylo_MPI]$
```

Fig. 7a. Parallel execution

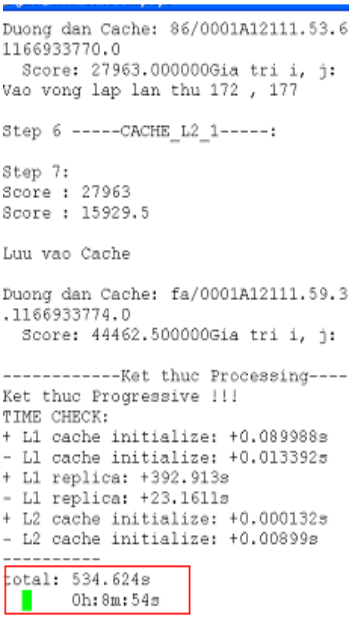


Fig. 7b. Parallel execution with cache

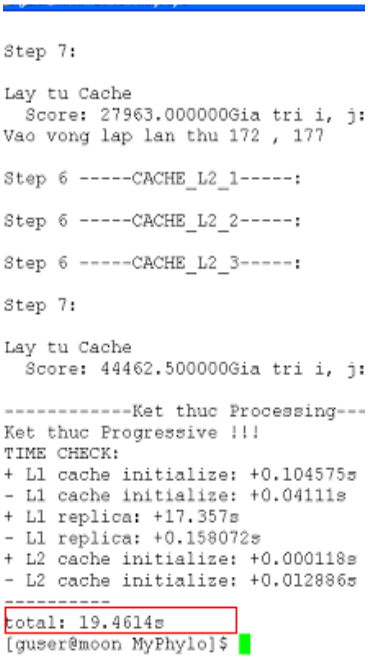


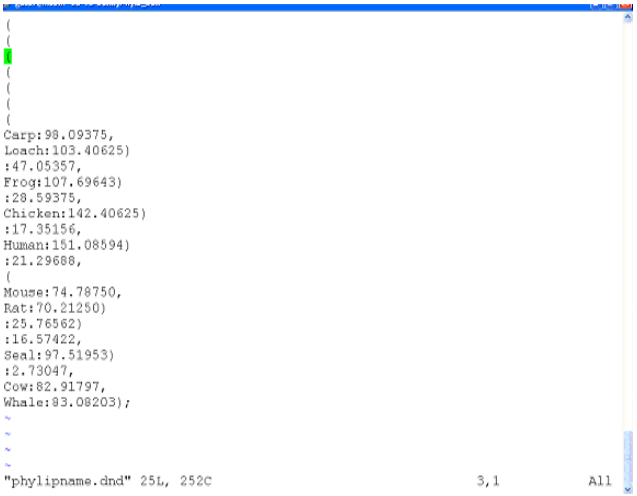
Fig. 7c. Parallel execution with cache

The execution time decreases significantly in the second test as follows:

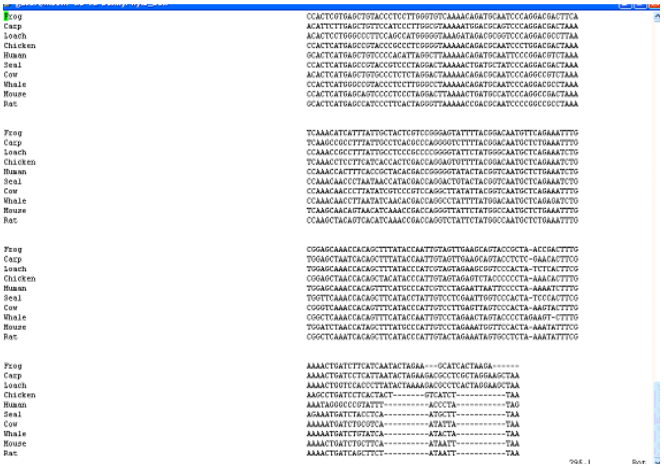
- Using parallel algorithm without cache: 727,78 sec≈ 12,13 min
- Using parallel algorithm with cache: 534,62 sec≈ 8,91 min
- Using parallel algorithm with cache the second time (for the same data): 19,46 sec≈ 0,32 min.

The output of program is guide tree and list of aligned sequences:

C. *Guide Tree*



D. *List of aligned sequences*



V. CONCLUSIONS AND FUTURE WORKS

The improvement of algorithm using cache technique allows us to reduce computational cost because the results calculated in previous times not to be calculated in the progressive process. Besides, Grid computing is also an effective environment to deploy the Multiple Sequence Alignment program. The combination of two techniques brings us many benefitsto increase program's execution speed. However, the parallel cache requires large amount of shared storage space, and network bandwidth is also a factor affecting the processing speed of the program. Our future work is studying suitable cache techniques to solve next-generation sequencing problems. In addition, we continue investigating the usefulness of parallel and distributed systems to boost the performance of the bioinformatics programs.

ACKNOWLEDGMENT

Thank to the research group of Czestochowa University of Technology Poland and Zola. They have developed CaLi library [7] and MSA program using the cache technique with their proposed methods [5].We used their library and source codes for our modified algorithms.

REFERENCES

[1] Waqar Haque, Alex Aravind, Bharath Reddy. *Pairwise Sequence Alignment Algorithms – A Survey*. ISTA'09, ACM 978, 2009.

[2] Yang, Hong Zhao, Ding-Yuan. *Research on Multiple Sequences Alignment Algorithm*. IEEE ICCIS, 2011.

[3] Cedric Notredame. *Recent revolutions of Multiple Sequence Alignment Algorithms: A survey*. PLoS Computational Biology. Volume 3, 2007.

[4] Balamash A., Krunz M.. *An overview of web caching replacement algorithms*. IEEE Comm. Surv. & Tut., 6(2):44–56, 2004.

[5] Jaroslaw Zola, *Parallel Server for Multiple Sequence Alignment*, L'Institut National Polytechnique de Grenoble, France, 12/2005.

[6] Grasso C, Lee C. *Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems*. Bioinformatics 20, 2004.

[7] Cali library: http://icis.pcz.pl/~zola/CaLi/index.html.

[8] NeedlemanS. B. and WunschC. D., *A general method applicable to the search for similarities in the aminoacid sequence of two proteins*. Journal of Molecular Biology, vol. 48, pp. 443-453, 1970.

[9] Feng D.F., *Doolittle R.F.,Progressive sequence alignment as a prerequisite to correct phylogenetic trees*. J. Mol. Evol. 25, 351-360, 1987.

[10] Hogeweg P, Hesper B.,*The alignment of sets of sequences and the construction of phylogenetic trees*. An integrated method. J. Mol. Evol. 20, 175-186, 1984.

[11] Taylor W.R.,*A flexible method to align large numbers of biological sequences*. J. Mol. Evol. 28, 161-169, 1988.

[12] L. Batzoglou, J. Pachter, B. Mesirov, B. Berger, and E. S. Lander, *Human and mouse gene structure: comparative analysis and application to exon prediction*. RECOMB 00: Proc of the 4th Int'l Conference on Computational Molecular Biology, 2000, pp. 46-53.

[13] The library for parallel programming MPI: http://www.mpi-forum.org/docs/docs.html

[14] Kuo-Ben Li. ClustalW-MPI: *ClustalW analysis using distributed and parallel computing*. Bioinformatics, 2003.

[15] Katoh, Ketal. *MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform*. NucleicAcidsRes., 2003.

[16] Kazutaka Katoh, *Hiroyuki Toh. Parallelization of the MAFFT multiple sequence alignment program*. Bioinformatics, 2010.

[17] G. Sudha Sadasivam, G. Baktavatchalam. A novel approach to Multiple Sequence Alignment using Hadoop Data Grids. MDAC'10, 2010.

[18] K. Arumugam, Y.S. Tan, B.S. Lee, R. Kanagasabai, *Cloud-enabling Sequence Alignment with Hadoop MapReduce: A Performance Analysis*. IPCBEE, 2012.

[19] Xun Tu, Kajal T. Claypool and Cindy X.Chen. *ACache: Using Caching to Improve the Performance of Multiple Sequence Alignments*. IEEE SSDBM'06, 2006.