IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 5, Ver. III (Sep.- Oct. 2017), PP 31-37
www.iosrjournals.org

Parallel DNA Global Alignmentimplementation For Cloud Computing Environment

[1]Mohamed Assal, [2]Ahmed Said, [3]Ahmed Ramadan,
[4]Mahmoud Mokhtar,[5]Mahmoud Zakaria
Faculty of Computers and Information,Modern university for information and technology, Cairo, Egypt

Abstract:*Nowadays Bioinformatics is a good and an upcoming technology for the recent researchers. Alignment and comparison of DNA and RNA, Genemapping on chromosomes, Protein structure prediction, gene finding from DNA sequences arevarious useful tasks of bioinformatics. In recent years, many techniques used for the DNAAlignment and Comparison. In the field of biologyDNA Alignment and Comparison plays a vital role in many biological areas. Needleman-Wunsch algorithm is the most famous algorithm for DNA global alignment. Awkwardly, it is based on sequential computing so it has a problem of being slow. Parallel Version of Needleman-Wunsch algorithm purposed aim to overcome this sequential computing limitation. This paper presents an implementation of this algorithm along with experiments in real parallel environment. Finally, the paper study the suitability of this algorithm to the grid-computing environment and provide it as a Cloud Service.*

Keywords:*DNA Computing, Computational biology, Cloud computing, Grid computing, Bioinformatics*

Date of Submission: 11-09-2017 Date of acceptance: 06-10-2017

I. Introduction

Multiple sequence alignments (MSA) are an essential and widely used computational procedure for biological sequence analysis in molecular biology, computational biology, and bioinformatics [1]. An ever-increasing number of biological modeling methods depend on the assembly of accurate MSAs. The accuracy ofMSA is of critical importance because many bioinformatics techniques and procedures are dependent on MSA results [1] [2].As the protein alignment problem studied for several decades; studies have shown considerable progress in improving the accuracy, quality, and speed of multiple alignment tools. Unfortunately, constructing accurate multiple sequence alignments is a computationally intense (classified as a NP-Hard problem) and biologically complex task and no current MSA tool is likely to generate a biologically perfect result. Sequences can be aligned across their entire length (global alignment) or only in certain regions (local alignment). Local sequence alignment plays a major role in the analysis of DNA and protein sequences[5][6] [7].Over the years, researcher efforts in finding different algorithms or mathematical models that require low computational cost as well as ensure accuracy. Therefore, this area of research is very active, aiming to develop a method, which can align thousands of sequences that are lengthy and produce high-quality alignments and in a reasonable time [3] [4].The two general sequence alignment models view the alignments in different ways: the first considers similarity across the full extent of the sequences (a global alignment); the second focuses on regions of similarity in parts of the sequences only (a local alignment). It is important to there is no value in performing a global similarity on sequences that have only local similarity[5][8].

Several algorithms developed for the sequence alignment problem based on dynamic programming, heuristic algorithm and probabilistic methods. From all the approaches, dynamic programming based implementations are more time consuming than heuristic based implementations. However, dynamic programming based approach provides a more accurate result. There exists two well-known dynamic programming based approach for sequence alignment. One for global sequence alignment, which is Needleman-Wunsch (NW) algorithm and the other, is for local sequence alignment, which is Smith-Waterman algorithm [4]. NW algorithm [9] considered one of the commonly used global sequence alignment algorithm. Given two input sequences, the algorithm calculates the dynamic programming matrix (D) and trace-back matrix (T). Two sequences written on each axis of the matrix and an extra row and column added to the matrix to allow alignment to begin with gap as shown in Table I. The corresponding dynamic programming matrix shown in Table II. After calculating dynamic programming and trace-back matrix, the highest score is searched, and a path is drawn, which corresponds to the best alignment [9].

This paper presents a multicore parallel implementation NW algorithm proposed in [10]. This implementation used to extend the capability of the Grid Developing System (GDS) [11]. Finally, An extension to the Grid Developing System (GDS) middleware, which enables the usage of the GDS computing resources as a cloud service.

Table 1 Initial Matrix

		A	T	C	G	G	T
	0	-8	-16	-24	-32	-40	-48
A	-8	-	-	-	-	-	-
T	-16	-	-	-	-	-	-
G	-24	-	-	-	-	-	-
C	-32	-	-	-	-	-	-
C	-40	-	-	-	-	-	-
A	-48	-	-	-	-	-	-

Table 2 Matrix after filling

		A	T	C	G	G	T
	0	-8	-16	-24	-32	-40	-48
A	-8	1	-7	-15	-23	-31	-39
T	-16	-7	2	-6	-14	-22	-30
G	-24	-15	-6	-14	-5	-13	-21
C	-32	-23	-14	-5	-13	-21	-29
C	-40	-31	-22	-13	-21	-29	-37
A	-48	-39	-30	-21	-29	-37	-45

II. Cloud Computing

Cloud computing is giving a start to new computing techniques in which local computers are not being used for computational processes as centralized facilities have overcome local computers which are being operated by cloud providers or by third-party computational and storage providers. Cloud Computing is being a suitable platform for the processing of huge amount of data in computer fields for various applications. It has emerged as a computing infrastructure that enables rapid delivery of computing resources as a utility in dynamically scalable, virtualized manner. There are various advantages of cloud computing over traditional computing which include: agility, lower entry cost, device independency, location independency, and scalability. There are many cloud-computing initiatives from IT giants such as Microsoft, IBM, Google, Amazon as well as start-up such as Parascale, Elastra and Appirio [12, 13].

Cloud computing is a computational process in which services are delivered over a network using computing resources. The name‘cloud’ symbolizes an abstraction for complex infrastructure it contains in system diagrams. The three main types of service modelsareInfrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). There are many other types such as Database as a Service (DBaaS), Recovery as a Service (RaaS) …etc.[14]. Cloud Computing has five essential characteristics, and four deployment models.

Those characteristics are *On-demand self-service*, *Broad network access*, *Resource pooling*, *Rapid Elasticity* and *Measured Service*. While Cloud Computing deployment models are: *Private cloud* when the cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units).Community Cloud when a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations) provisions the cloud infrastructure for exclusive use. *Public cloud* when The cloud infrastructure is provisioned for open use by the general public, and *Hybrid cloud* when The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities [15].

III. Parallel Needleman-Wunsch Algorithm For Grid [10]

In[10], The Authors presented a new global alignment algorithm that can be implemented using parallel computing (such as grid computing) to overcome the Needleman-Wunsch sequential computing limitation.

A parallel version of Needleman-Wunsch algorithm [9] implemented in [10]. The parallel implementation uses multiple processors or multicores for initializing, Calculating and filling the Matrix by dividing the Matrix into slices (diagonal rows). Each slice will only have data dependency on only its previous slice, so it enables us to use multiple processors on the same slice to fill its score and symbol in parallel. Fig. 1 shows the matrix slices.

The Dynamic Programming Matrix in the parallel version of Needleman-Wunsch algorithm is stored in the global or shared memory space among the processors or the cores. The parallel version follows the NW algorithm in the requirements, as the same Matrix is required, and the three assumptions Gap, Match, Mismatch values.

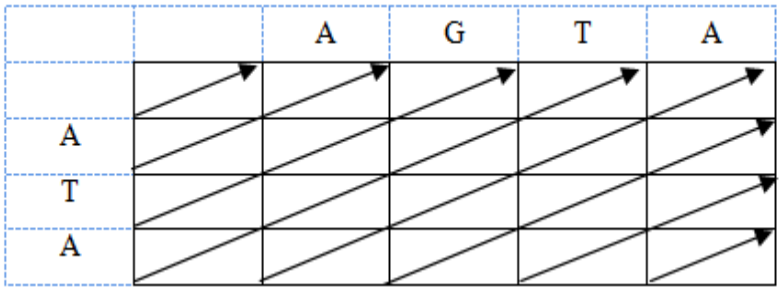

Fig.1 Parallel Needleman-Wunsch Slices

Parallel Needleman-Wunsch algorithm for Grid implemented using Microsoft .Net Framework 4.6.2 and C# 6.0.

The steps of the algorithm, which are as follows:

1) Initialization: Fill the first row and column initially with distance from origin multiplied by GAP value. Place the value in the trace-back matrix accordingly. Fig. 2 shows the pseudocode of the initialization process.

2) Matrix Fill: Fill other cells in the matrix from the values of its left, top and top left diagonal element. Initialize trace-back matrix according to the selected value. Fig. 3 shows the pseudocode of the filling process.

3) Trace-back: Reach at the bottom right corner and start tracing the arrows mentioned in the trace-back matrix until you reach at the first element in the matrix. Put the values of the GAP according to the direction traversed in the matrix into the new sequence that we generate during trace backing.

Initialization algorithm

```
Doing the following processes in parallel
fori = 1 to M do
        Matrix [0, i] = Matrix [0, i-1] + Gap
End for
forj = 1 to Ndo
        Matrix [j, 0] = Matrix [j-1, 0] + Gap
End for
Output: Initialized Matrix
```

Fig. 2InitializationAlgorithm

Filling algorithm

```
For equal length sequences
        Number of Slices = M + N +1
        For each CurrentSlice = 2 to Number of Slices {
                Parallel For each StartRow = 1 to CurrentSlice {
                        Matrix (StartRow, CuurentSlice - StartRow) = max {
                                Matrix (StartRow − 1, CurrentSlice - StartRow − 1) + (Match
                                (SeqA[CurrentSlice - StartRow − 1],SeqB[StartRow − 1]),
                                Diagonal)
                                Matrix (StartRow − 1, CurrentSlice) + Gap , Up
                                Matrix (StartRow, CurrentSlice − 1) + Gap , Left
                        }
                }
        }
For not equal length sequences
By assuming, the first sequence is the larger one.
        For each CurrentSlice = 2 to M {
                Parallel For each StartRow = 1 to CurrentSlice {
                Matrix (StartRow, CuurentSlice - StartRow) = max {
                        Matrix (StartRow − 1, CurrentSlice - StartRow − 1) + (Match
                        (SeqA[CurrentSlice - StartRow − 1],SeqB[StartRow − 1]), Diagonal)
                        Matrix (StartRow − 1, CurrentSlice - StartRow) + Gap, Up
                        Matrix (StartRow, CurrentSlice - StartRow − 1) + Gap, Left
                        }
                }
```

```
        }
        For each CurrentSlice = M to (M + N +1){
                Parallel For each StartRow = (CurrentSlice – M +1) to N +1 {
                        If CurrentSlice<StartRow) return;
                        Matrix (StartRow, CuurentSlice – StartRow + 1) = max {
                                Matrix (StartRow − 1, CurrentSlice - StartRow − 1) + (Match
                                (SeqA[CurrentSlice - StartRow ],SeqB[StartRow − 1]), Diagonal)
                                Matrix (StartRow − 1, CurrentSlice – StartRow + 1) + Gap, Up
                                Matrix (StartRow, CurrentSlice - StartRow ) + Gap, Left
                                }
                        }
        }
        *Match(letterA,letterB) {
        If letterA equals letter B
                returnMatchValue
        Else
                returnMismatchValeu
        }
Output: Matrix after filling
```

Fig3Filling Algorithm

Trace back algorithm

```
i=length of seq A
j=length of seq B
while(i>0 or j>0):
        ifptr[i,j]==0:
                Addi^th character of seqA to alnseqA
                Addj^th character of seqB to alnseqB
                Decrementi by 1
                Decrement j by 1
        else ifptr[i,j]==1:
                Addi_th character of seqA to alnseqA
                Add '-' to alnseqB
                Decrementi by 1
        Else:
                Add '-' to alnseqA
                Addj^th character of seqB to alnseqB
                Decrement j by 1
Output: Aligned Sequences
```

Fig. 4 Trace back algorithm

IV. Cloud GDS Implementation

GDS is a .NET based computational grid environment implemented in MTI University. The GDS grid-computing framework conceived with the aim of making grid construction and development of grid software as easy as possible without sacrificing flexibility, scalability, reliability and extensibility [11].GDS has practical capabilities of connecting up to 4096 workstations. In addition, GDS is hardware scalable in which workstations could be easily replaced with a high-end server through the GDS plug and play agent feature. The GDS will automatically utilize the new powerful resources in the new connected agents [16].An extension for the GDS implemented to allow the usage of the GDA DNS features as a cloud service over the internet. The service is available through a web portal and available freely for the users. In other words as a public cloud service. This extension also implemented using Microsoft .Net Framework 4.6.2 and C# 6.0.The user will submit two DNA Sequences to the web portal and will receive an id representing the alignment task for future reference. The Web Portal will communicate with the Grid backbone using a shred database and SignalR to place the new task in the GDS Task queue. SignalR gives the ability to define a communication path between the GDS and the website using a Hub, that Hub should contain only two functions one for the Web Portal for talking to the Grid and another for listening from the Grid. Fig. 5 shows the basic connection model for Microsoft SignalR Technology.The queue simply follows the FIFO mechanism to get the task complete. Once the grid resources do the task, alignment results will be available through the web portal and the user will be notified by email. The user can also check his previous tasks results using the task is or through his control panel provided by the web portal. Fig. 6 shows a couple of screenshots for the Web Portal, (a) shows the landing page of the web portal

which allow the user to navigate through the portal while (b) shows how the user can submit a new sequence alignment job to the GDS.

If there are many tasks in the GDS queue, the task dispatcher will following the recommended task-sending patch found in [17], which is four tasks to allocated tasks to the underlying agents. The Web portal offer number of services other than the Sequence alignment including *Cleaning Sequences* from errors, *Generating Sequences* for testing purposes and *Splitting lengthy sequences* to split long sequence into a number of smaller sequences.

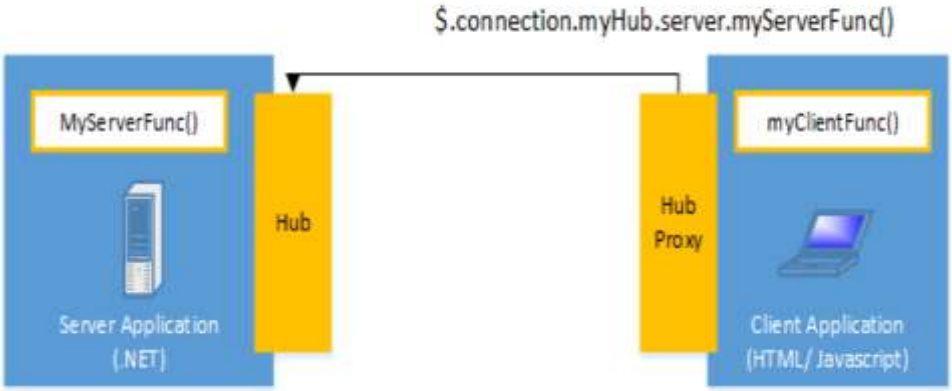

Fig. 5 SignalR Connection Model.

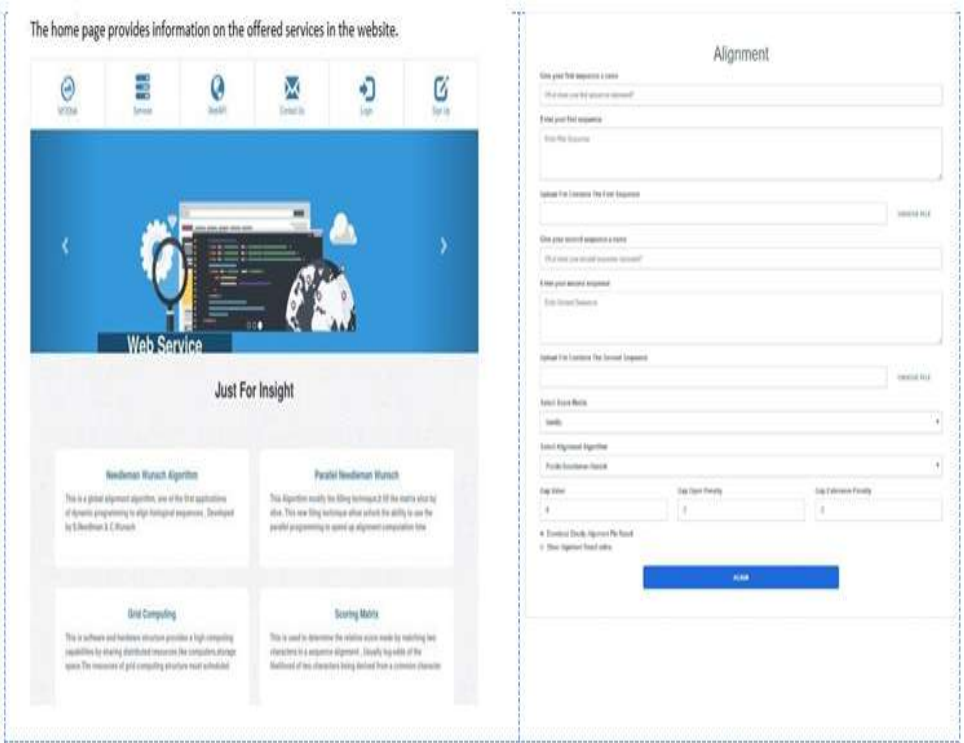

Fig. 6 Webportal Screenshots.

V. Experiments And Discussions

The authors of the parallel NW algorithm [10] did not prvovide any experiment results of their proposed algorithm. So, it has been decieded to put the algorithm to test. Before, experimenting our parallel implementation of the NW algorthim. The implementation should be validated against well known publicly available alignment algorithm such asMicrosoft Bio Library (BioNet) [18] and the European Bioinformatics Institute (EMBL-EBI) Web service [19]. Microsoft Bio Library has been chosen to validated the Parallel NW algorithm agaisnt using 3 public available datasets (Amebiasis, Fascioliasis and Haemophilia). Fig. 7 shows the validation results.

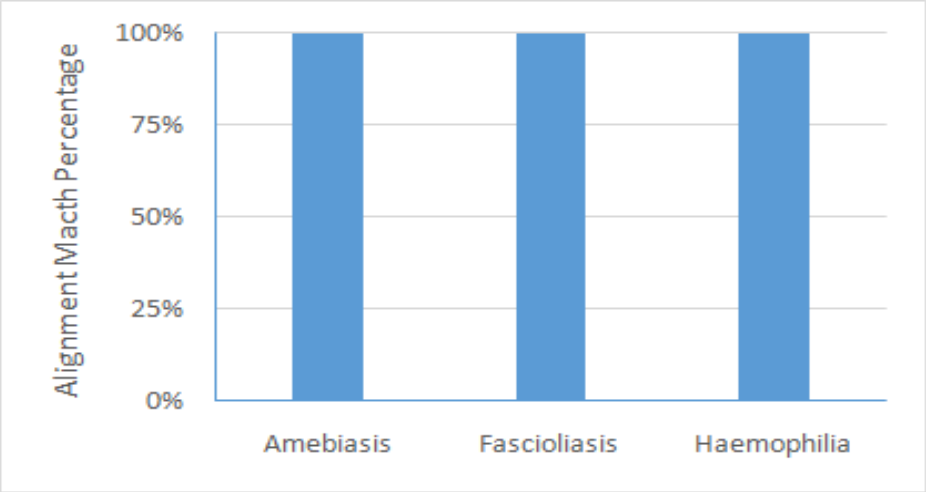

Fig. 7 Parallel NW Implementation Validation

After validating the parallel NW implemention, it has been decieded to test the parallel implementation in a real parallel environment before put them to test in the GDS grid. The implemented algorithm tested using a workstation with the specifications in Table 3.The experiments ran on a single core (sequential), 2 Cores, and 4 Cores. The experiments done on a 2 sequences of an equal variable lengths starting from 10,000x10,000 characters up to 40,000x40,000 characters. Table 4 and Fig. 8showsome of the performance results of the new implementation.

Table 1 Workstation Specifications

Specification	
Processor	Intel® Core™ i7-6700
Clock	3.4 GHz
Number of Cores	4 Physical
Memory	16 GB
Operating System	Windows 10 Version 1703 64 bit Professional

Table 2 Parallel NW Performance Results

Sequences Length	**Cores (in Seconds)**			
	Single	**2 (Dual)**	**3**	**4 (Quad)**
10000	8.74	6.41	5.011	4.831
15000	21.279	13.938	10.468	8.615
20000	46.68	29.429	20.977	15.272
25000	67.765	42.913	30.571	21.681
30000	103.382	62.606	48.831	32.495
32000	174.726	97.637	71.007	45.037
35000	185.161	112.224	76.069	48.62
37000	236.465	137.27	95.266	61.237
40000	273.928	162.428	118.5	76.416

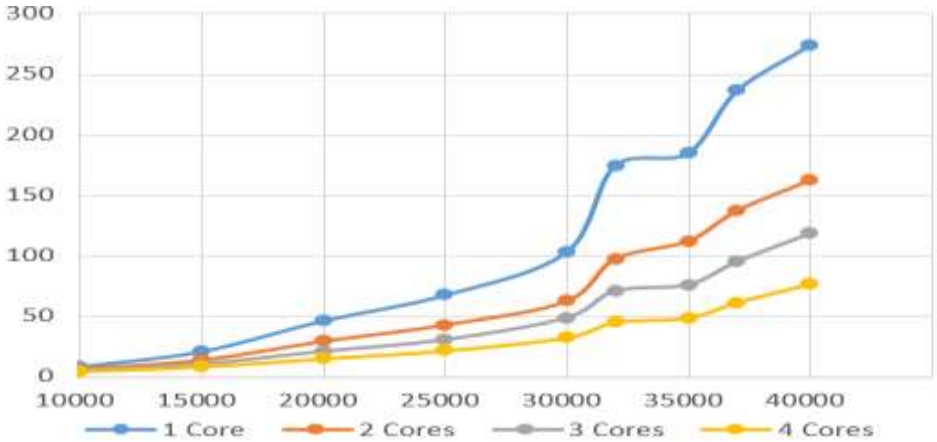

Fig.8Parallel NW Performance Results

The pervious expirements show that the parallel NW algorithm is completely suitable to run on parallel shared memory environment and achieve an outstanding performance results.

VI. Conclusion

The paper presented a parallel implementation of Needleman-Wunsch algorithm.The new algorithm has been validated against a couple of well-kown publicy available (MicrosoftBio Library). The parallel Needleman-Wunsch achieved an exceptional performance results in parallel environment. The algorithm used to extend to the capability of the GDS Grid computing environemnt to allow utilizing the multicore of the grid underlying agents. Finally, The GDS middleware has been elevated to provide the DNA algnment features as a public cloud serice over the internet.

References

[1]. J. Daugelaite, A. O' Driscoll and R. D. Sleator, "An Overview of Multiple Sequence Alignments and Cloud Computing in Bioinformatics," ISRN Biomathematics, vol. 2013, p. 14, 2013.

[2]. C. Kemena and C. Notredame, "Upcoming challenges for multiple sequence alignment methods in the high-throughput era," BIOINFORMATICS, vol. 25 , no. 19, p. 2455–2465, 2009.

[3]. R. C. Edgar and S. Batzoglou, "Multiple sequence alignment," Current Opinion in Structural Biology, vol. 16, no. 3, p. 368–373, 2006.

[4]. A. Chaudhary, D. Kagathara and V. Patel, "A GPU based implementation of Needleman-Wunsch Algorithm using Skewing Transformation," in IC3 '15 Proceedings of the 2015 Eighth International Conference on Contemporary Computing (IC3), USA, 2015.

[5]. M. Assal, A. Said, S. Souliman and A. Essam, "A Study to the different implementation approaches for the Grid YM-Algorithm DNA alignment," IOSR Journal of Computer Engineering (IOSR-JCE), vol. 18, no. 6, pp. 16-21, 2016.

[6]. L. V. Vinh, T. V. L. Lang, N. Thi Thu Du and V. H. B. Chau, "Multiple Sequence Alignment on the Grid Computing," International Journal of Computer Science and Telecommunications, vol. 3, no. 7, pp. 46-51, 2012.

[7]. L. Jiang and W. T., "On the complexity of multiple sequence alignment," Journal of Computational Biology, vol. 1, no. 4, p. 337–348, 1994.

[8]. S. Vasantharathna, A. Kunthavai and R. Karuppayya, "AGAligner – DNA Local Sequence Alignment Using Alchemi Grid," IRACST – Engineering Science and Technology: An International Journal (ESTIJ), ISSN: 2250-3498, vol. 2, no. 3, 2012.

[9]. S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," Journal of Molecular Biology, vol. 48, no. 3, pp. 443-453, 28 March 1970.

[10]. T. Naveed, I. S. Siddiqui and S. Ahmed, Parallel Needleman-Wunsch Algorithm for Grid, Islamabad, Pakistan: Bahria University, 2005.

[11]. A. Said, M. Assal and M. Bakr, "An Enhanced framework for Grid Computing Developing System (EGDS)," Managerial Research Journal, Consultancy Research & Development Center, 2012.

[12]. W.-T. Tsai, X. Sun and J. Balasooriya, "Service-Oriented Cloud Computing Architecture," Seventh International conference on Information Technology, IEEE, pp. 684-689, 2010.

[13]. S. B. Rout, B. S. P. Mishra and S. Dehury, "Hadoop Cloud Application In DNA Alignment And Comparison," International Journal of Engineering Research & Technology (IJERT), vol. 2, no. 7, pp. 102-107, 2013.

[14]. S. Khurana and A. G. Verma, "Comparison of Cloud Computing Service Models: SaaS, PaaS, IaaS," International Journal of Electronics & Communication Technology (IJECT), vol. 4, no. 3, pp. 29-33, 2013.

[15]. P. Mell and T. Grance, "The NIST Definition of Cloud Computing," National Institute of Standards and Technology (NIST) Special Publication 800-145, Gaithersburg, 2011.

[16]. A. Said, Design and Building of a Framework for Grid Computing Developing System, Cairo: Arab Academy for Science, Technology & Maritime Transport, December 2012.

[17]. M. Assal, A. Said, D. Mohamed and N. Osama, "A Study to the effect of task granulation for the DNA multiple sequence alignment on Grid Computing," IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS),, vol. 5, no. 3, 2015.

[18]. Microsoft, "Bioinformatics library for .NET," Microsoft, [Online]. Available: https://github.com/dotnetbio/bio. [Accessed 27 May 2017].

[19]. EMBL-EBI, "The European Bioinformatics Institute (EMBL-EBI)," EMBL-EBI, [Online]. Available: https://www.ebi.ac.uk/. [Accessed 27 May 2017].

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

Mohamed Assal. "Parallel DNA Global Alignmentimplementation For Cloud Computing Environment." IOSR Journal of Computer Engineering (IOSR-JCE), vol. 19, no. 5, 2017, pp. 31–37.