High Performance Grid Environment for Parallel Multiple Biological Sequence Alignment

Plamenka Borovska
Computer Systems Department
Technical University of Sofia
Sofia, Bulgaria
pborovska@tu-sofia.bg

Veska Gancheva
Programming and Computer Technologies Department
Technical University of Sofia
Sofia, Bulgaria
vgan@tu-sofia.bg

Nikolay Landzhev
Computer Systems Department
Technical University of Sofia
Sofia, Bulgaria
landzhev@yahoo.com

***Abstract*— The parallel in silico simulations based on methods and algorithms for analysis of biological data using high-performance distributed computing is essential for accelerating the research and reduce the investment. This paper presents a high-performance Grid environment integrating various services and middleware to facilitate access to distributed resources for carrying out scientific experiments in the area of bioinformatics. This environment enables parallel computer simulations increasing the efficiency of the computations and allowing scientists easy and user friendly access. Web portal provides as services access and execution of parallel program implementation based on an algorithm for comparative analysis of biological data. An innovative algorithm called MSA_BG for multiple sequences alignment based on the concept of Artificial Bee Colony metaheuristics is designed. Experimental simulations on the basis of parallel implementation of MSA_BG algorithm for multiple sequences alignment through the high-performance Grid environment have been carried out for the case study of the influenza virus variability.**

Keywords-artificial bee colony; bioinformatics; GRID computing; multiple sequence alignment; portal.

I. INTRODUCTION

In silico biological sequence processing is a key investigation in the area of bioinformatics. This scientific area requires powerful computing resources for exploring large sets of biological data. The huge amount of biological sequences accumulated in the world nucleotide and protein databases requires efficient parallel tools for structural genomic and functional analysis. The parallel implementation of methods and algorithms for analysis of biological data using high-performance distributed computing is essential.

Multiple sequences alignment involves more than two biological sequences, generally protein, DNA, or RNA [1]. Multiple sequence alignment is computationally difficult and is classified as a NP-Hard problem [2][3][4]. ClustalW is a widely used multiple sequence alignment algorithms for Deoxyribonucleic acid (DNA) or proteins and implements a progressive method for multiple sequence alignment [5]. It calculates the best match for the selected sequences and lines them up so that the identities, similarities and differences can be seen. The basic algorithm behind ClustalW proceeds in three stages: pairwise alignment (PA), guide tree (GT) and multiple alignment (MA). The ClustalW phases are relatively independent. Each of the phases produces intermediate data which is used as an input for the next one, but the calculations are independent.

Several parallel ClustalW algorithms for multiple sequence alignment have been reported in recent years. ClustalW-MPI is a distributed and parallel implementation on distributed computer clusters and on the traditional parallel computers and uses a scheduling strategy called fixed-size chunking where batches of tasks of one fixed size are to be allocated to available processors [6]. The algorithm ClustalW_MPI is not so effective in the case of highly parallel implementations. The analysis of the ClustalW_MPI algorithm show that it has not regular structure, does not consist of replica modules and is characterized by significant computational imbalance, because it is based on the parallel algorithmic paradigm “master-slave” [7]. Increasing input file size is limited due to allocated memory size (Juqueen’s memory allocation per core is 1GB [8]). The maximum number of input sequences is approximately 10000 (depending on the sequences length).

The aim of our work is to provide a generic high level user friendly heterogeneous high-performance grid based environment for massively parallel in silico biosimulations. The solution allows scientists easy customization, parameterization, execution and analysis of complex simulation scenarios utilizing bioinformatics tools.

An innovative parallel algorithm MSA_BG for multiple sequences alignment based on the concept of Artificial Bee Colony (ABC) metaheuristics and the concept of algorithmic and architectural spaces correlation is designed.

The ABC algorithm is an optimization algorithm based on the intelligent foraging behaviour of honey bee swarm [9]. In the ABC model, the colony consists of three groups of bees: employed bees, onlookers and scouts.

The choice of the ABC algorithm is based on the fact that, in essence, it is a hybrid metaheuristics - a combination of methods based on populations (scouts generate a number of possible solutions simultaneously) and a method based on trajectories (employed bees perform the local searches around the decisions of the scouts, seeking to improve the decisions quality).

 ISBN: 978-1-61208-283-7

This paper is structured as follows. Section II explains the scientific application gateway. The design and the algorithmic framework of the algorithm MSA_BG are explained in Section III. Section IV presents the experimental framework. The experiments, performance evaluation and results analysis are discussed in Section V. We present the conclusion and future work aspects in Section VI.

II. SCIENTIFIC APPLICATION SPECIFIC GATEWAY

The objective of an academic scientific portal gateway is to allow a large number of users to have transparent access to available distributed advanced computing infrastructure, software tools, visualization tools and resources in the area of High Performance Computing (HPC).

The portal provides user-centric view of all available services and distributed resources. The Web-based environment is organized to provide a common user interfaces and services that securely access available heterogeneous computing resources, data and applications. It also allows many academic and scientific users to interact with these resources.

In order to achieve these requirements, the execution of computing tasks on these resources needs to be managed. Therefore, resource management and security services are needed. The required execution management can be described in terms of three classes of services: Resources, Job Management and Monitoring, and Resource selection. End-users (e-scientists) will be able to: enable the access to the various components of the e-science infrastructure via specific, user community views; execute the published applications with custom input parameters by creating application instances using the published applications as templates; store and share applications; observe and control application execution; support user's collaboration via sharing applications and databases. Application developers will be able to develop applications by the portal (application porting) and publish the completed applications into the infrastructure in order to end-users to use them. Portal developers will be able to develop user interfaces as portlets for end-users of the portal and associate the user interfaces to the portal using an API (Application Programming Interface).

The portal provides a customized application specific gateway, and can be summarized as follows:

- Identifying and clarifying the specific requirements, scenarios and real needs of target groups such as scientists, researchers, educators and students.
- Defining of specific views for the user community according to their requirements and development of user scenarios for the specific applications.
- Identifying software packages, testing these packages and selecting the suitable ones.
- Creating the relevant biological database and data extracting.
- Designing and developing application-specific portlets corresponding to high-level user scenarios in order to execute the user scenarios and to allow users to define their own input parameters, execute and monitor the experiments, gather the results and visualize them.
- Experimental studying and evaluating of the specific application portlets and defining the usability of portlets.

Depending on the user community needs, the scientific gateway may provide any of the following features:

- High-performance computation resources (local clusters).
- General or domain-specific analytic and visualization software.
- Collaborative interfaces for sharing interfaces and database.
- Job submission and monitoring tools and performance analysis tools.
- Specific user community views.

The major user scenarios can be summarized as follows:

- Supporting users with uniform access to heterogeneous distributed high-performance computation resources such as local clusters.
- Storing, sharing and parameterising of ready-to-use applications.
- Monitoring and controlling of the application execution, collecting and visualizing the results.
- File transfer (transfer based on user request and transfer based on subscription/notification).
- Accessing collaborative tools for data management, search tools, and analysis tools.
- Running new Distributed Computing Infrastructure (DCI) applications based on the developed gateway.
- Enable the integration of other services.
- Enable the store and share of half-made applications, application templates.

The basic contributions of the portal are as follows:

- Developing of application-specific gateway.
- Establishing and providing a production service for the user communities in the area of high performance computing.
- Increasing the number of specific portlets and applications in the portlet and application repositories, intended for user communities.
- Support user's community with uniform access to high-performance computation resources such as local clusters.

The proposed high-performance Grid-based infrastructure as services for carrying out remote simulations consists of HPC resources, learning management system, resource database, knowledge resources as services, software tools and web portal (Fig. 1). The Grid environment is based on the guse/ws-pgrade system [10]. Building the portal enables users to manage their tasks through the interface and administrators to easily manage users.

 ISBN: 978-1-61208-283-7

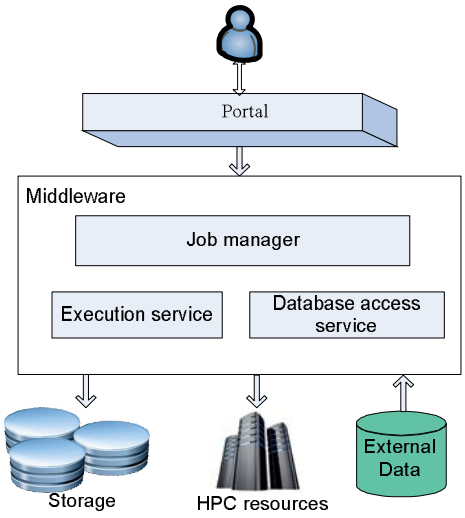

Figure 1. High performance infrastructure.

The web portal provides also: user profile management, e.g., different views for different user; personalized access to information, software tools and processes; getting information from local or remote data sources, e.g., from databases, transaction systems, or remote web sites; aggregated the information into composite pages to provide information to users in a compact and easily consumable form. In addition, the portal also includes applications like execution of some software tools, etc.

III. MULTIPLE SEQUENCE ALIGNMENT ALGORITHM MSA_BG BASED ON ARTIFICIAL BEE COLONY

The concept is to synthesize a parallel iterative algorithm with a regular computational and communication system based on data parallelisms and replica code, which is executed on all computing nodes. The parallel paradigm is Single Program Multiple Data (SPMD) and data decomposition. The granularity is hybrid - coarse granule computing for each node (multithreaded process) that runs multithreading (fine granule) of the cores within the computing node. In the case of hybrid granularity in order to effectively use the resources of supercomputers is appropriate to use hybrid parallel implementations.

A new parallel algorithm MSA_BG, in which each process simulates the behaviour of a beehive and each hive contains multiple swarms is proposed. The parallel algorithm is designed based on the methodology for the synthesis of parallel algorithms, which is based on the correlation of the parameters of the algorithmic and architectural spaces. The threads simulate the behaviour of many bees in the swarm.

Scout bees in the swarms round certain subregions in the searching space and construct a potential solution. Once the scout bees obtain possible (feasible) solution, they return to the hive and begin to dance in the hive.

Onlookers watch the dance of employed bees, choose one of their solutions and evaluate them.

The employed bees select one of the solutions and make attempts to improve it based on local search. The quality of obtained solutions is determined by the grade of sequences similarity. As much as this evaluation is greater, the quality of obtained alignment is higher, i.e., the criterion of optimality is a maximum similarity score.

In case of higher quality of solution generated by a scout the possibilities to include it in the list of elite solutions increased. As next step this solution will be improved based on a local searching by the employed bees.

A. *Metaphor*

The food source is presented by possible sequence alignments. The quality of solutions (nectar amount) is evaluation function of the similarity between the sequences for a particular alignment.

The proposed algorithm is based on the concept of swarms and hives. The allocation of computing supercomputer resources is as follows. The entire supersystem simulates the behaviour of a colony of beehives, and the number of hives is equal to the number of computing nodes. Each computing node simulates the behaviour of a hive. In a hive are included q swarms, where q is the segments number of the supersystem. The swarms within a hive work on common lists of best temporary solutions and elite solutions. Each hive has a queen bee, which gets the elite decision of all swarms on the hive. Scout 0 reads the sequences from the memory and stores it in sharing memory of the computational node (hive). Scouts generate initial solutions thought sequence alignment including gaps. Random number generator Mersenne Twister is used. Scouts evaluate quality of the alignment using the following method.

An assessment by columns is done – in case of nucleotide sequences the numbers of symbols – A, G, C and T are counted. The numbers of symbols are compared and are selected the nucleotides, which occur mostly in the different columns. It is called "nucleotide-favourite" sequence (f_{ij}). After the calculation of assessments in columns is formed the so-called sequence-favourite contained in each position the respective favourite in the column, i.e., the nucleotides form the so-called sequence-favourite. Compared sequences along with the sequence-favourite form the so-called working set sequences.

A scoring matrix is built up where for sequences (rows in the matrix) are stored (by columns) the values of the evaluation function S. For sequence (row) i in position j (column): nucleotide a_{ij} and nucleotide–favourite f_{ij}:

$S_{ij} = 0$ in case of a_{ij} = gap

$S_{ij} = 1$ in case of $a_{ij} = f_{ij}$

$S_{ij} = -1$ in case of $a_{ij} \neq f_{ij}$

The elements of the scoring matrix are calculated for each sequence $S_{p,n+1}$, where n is the sequences length and p – the number of aligned sequences.

The overall assessment of the quality of alignment S is formed as:

$$S = \sum_{i=1}^{p} \sum_{j=1}^{n} Si,j \quad (1)$$

The additional column (similarity counters) of the scoring matrix $S_{n,n+1}$ consists of the similarity evaluation for each sequence compared with the sequence-favourite, i.e.,

$$Si,n+1 = \sum_{j=1}^{n} Si,j \quad \text{i€[1,p]} \quad (2)$$

The next example shows the working set. The sequence-favorite is marked in red.

A	**G**	**T**	**C**	**A**	**A**	**T**
A	A	T	C	G	A	T
A	G	T	C	A	T	T
A	G	-	G	A	A	G

The next example illustrates the calculation of scoring matrix *S*. The scoring column of the counters is marked in red and consists similarity scores for each sequence and the sequence-favorite.

1	-1	1	1	-1	1	1	**3**
1	1	1	1	1	-1	1	**6**
1	1	0	-1	1	1	-1	**2**

The scouts write down the scores of the working sets in the list of working solutions that is sorted in descending order by the total alignment scores. For each sequence (row) is generated dynamic data structure containing the indexes of the gaps in the sequences.

The onlookers select the best quality working set and perform local search – make minor changes in the working set and evaluate the quality of the modified alignment. In case of quality improvement, the modification is accepted and the working set is stored in the list of "best temporary solutions". Otherwise, the new alignment is ignored.

The following approach for modification of the aligned working set of sequences is proposed. The column with counters of the scoring matrix *S* is reviewed and is selected the row (sequence) with the lowest counter value (sequence that differs most from the favourite). Using a random generator is selected an index for insertion a gap (INS) and an index for deletion a gap (DEL) from the list of empty positions (to keep the length of the sequence) by obligatorily INS ≠ DEL. Both indexes are compared. If DEL>INS, then all characters in positions between INS and DEL are shifted one position to the right (shift_right), otherwise if DEL>INS – are shifted one position to the left (shift_right).

An example for the sequence is shown below:

```
A-TGC-GGTA-CCGT-G
```

The list of gaps indexes is: 1, 5, 10, 15.

Based on random generator, the position for insertion INS=4 and the position for deletion DEL=15 are selected. In this case, DEL>INS, mandatory DEL ≠ INS, and the operation is shift_right.

```
A-TGC-GGTA-CCGT-G
A-TG-C-GGTA-CCGTG
```

The value of similarity counter for sequence compared with the sequence-favourite is calculated in case of improving the modification of the working set is accepted and is stored in the list of temporary best solutions; otherwise, the modification is ignored.

In case of inserting position INS=12, and deleting position DEL=5, DEL<INS and the operation is shift_left.

```
A-TGC-GGTA-CCGT-G
A-TGCGGTA-C-CGT-G
```

After number of modifications, employed bees suspend the processing of the current working set and write down the best solution in the list of elite solutions that is sorted in descending order.

The condition for termination of the parallel algorithm is the number of iterations; then, the mother bee shall inform the queen bee of the colony and sends it the quality of the best solution (elite solution).

B. Algorithmic framework for parallel multiple alignment

The algorithmic framework for parallel multiple alignment of biological sequences MSA_BG on the basis of ABC algorithm is shown in Fig. 2.

```
Parallel For
For m=1, Q                          // For every hive
   For r=1, q        // For every swarm (node)
      Input_sequences //Scout 0
      Parallel Sections
         Parallel For k=0,15    /*Scout_bees
            Generate_ Random_Alignment
            Evaluate_Score_By Columns
            Construct_Favorite_Sequence
            Evaluate_Column of Counters
            Save_Working Set_in_Optimization list
         End Parallel For
         Parallel For k=0,15    /*Onlookers_bees
            Select_Random_Working_Set_Out_of_Optimization_List
            Select_Sequence_of_MIN_Counter
            Select_INS(Random)
            Select_DEL, such that DEL≠INS (RANDOM)
            If DEL>INS then shift_right else shift_left
            Evaluate_Column of Counters
            If higher_quality then Save_In_Optimization_List
            else skip
         End Parallel For
      End Parallel For
   Sort_Optimization_List_by_Quality    // Scout 0
   Reduce_Elite_Alignment_to_Mother_of_Hive   // Scout 0
   Reduce_MAX_ Elite_Alignment_of_Hive   //Mother of Hive
   End Parallel For
   Reduce_MAX_ Elite_Alignment _of_Colony //Queen of Colony
   Output_Best_Alignment_Obtained_So_Far          //Queen of Colony
```

Figure 2. Algorithmic framework for parallel multiple alignment of biological sequences MSA_BG on the basic of ABC algorithm.

The conceptual model of the MSA_BG method for parallel multiple alignment of biological sequences on the basis of ABC algorithm is shown in Fig. 3.

 ISBN: 978-1-61208-283-7

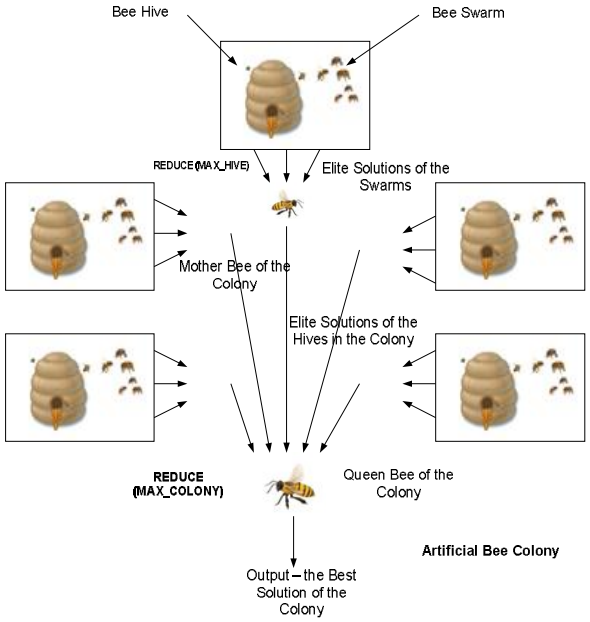

Figure 3. Conceptual model of the MSA_BG method for parallel multiple alignment of biological sequences on the basis of ABC algorithm.

The colony queen through collective communication reduction and operation MAX gets the quality of the elite decisions by hives mother bees. Thereafter the colony queen determines and sends the best one.

IV. EXPERIMENTAL FRAMEWORK

The experimental framework is based on a heterogeneous computer cluster of 10 nodes including 8 servers AMD Opteron 64 Dual Core Processor 1.8 GHz, RAM: 2GB 800 MHz, HDD: 2x160GB Hitachi SATA in RAID 0 and 2 servers CPU: 2x Intel Xeon E5405 Quad Core Processor 2 GHz, RAM: 4GB 800MHz, HDD: 2x 146 GB Hitachi 10000 RPM in RAID 0. All nodes are interconnected via gigabit Ethernet switch. The operating system is 64 bit Scientific Linux 5.3. Message passing is based on the MPICH2 1.1.1p2 distribution of the MPI standard. The resource manager is SLURM.

V. PARALLEL PERFORMANCE EVALUATION AND RESULTS ANALYSIS

The objective of the experiments is to estimate experimentally parallel performance parameters and make profiling of the program implementation on the basis of MSA_BG algorithm. Similarity searching between Ribonucleic acid (RNA) segments of various influenza viruses A/H1N1strains obtained from Genbank [11] has been carried out based on the parallel program MPI-based implementation of MSA_BG algorithm for multiple sequence alignment on heterogeneous distributed high-performance resources.

The experiments are conducted on grid infrastructure using various numbers of cores and various numbers of iterations. Experimental results are shown in Table I.

TABLE I. EXECUTION TIME OF MSA_BG ALGORITHM USING VARIOUS NUMBERS OF CORES AND VARIOUS NUMBERS OF ITERATIONS

Execution time					
Cores	*Iterations*				
	1000	*10000*	*100000*	*1000000*	*10000000*
1	4 sec	41 sec	6,42 min	62 min	624 min
2	2 sec	20 sec	3,22 min	31 min	312 min
8	1 sec	6 sec	0,51 min	8 min	82 min
16	1 sec	3 sec	0,26 min	4 min	42 min
28	1 sec	2 sec	0,15 min	2 min	24 min

The speedup is evaluated as a ratio of the sequential execution time to the parallel execution time. The experimental results for the speedup of MSA_BG algorithm on using various numbers of cores and various numbers of iterations are shown in Table II and Fig. 4.

The efficiency is evaluated as a ratio of the achieved speedup to the number of cores. The results for the efficiency of the MSA_BG algorithm gains in the case of various number of cores and number of iterations are shown in Table III and Fig. 5.

TABLE II. SPEEDUP OF MSA_BG ALGORITHM USING VARIOUS NUMBERS OF CORES AND VARIOUS NUMBERS OF ITERATIONS

Speedup					
Cores	*Iterations*				
	1000	*10000*	*100000*	*1000000*	*10000000*
2	2	2,05	1,99	2,00	2,00
8	4	6,83	12,59	7,75	7,61
16	4	13,67	24,69	15,50	14,86
28	4	20,50	42,80	31,00	26,00

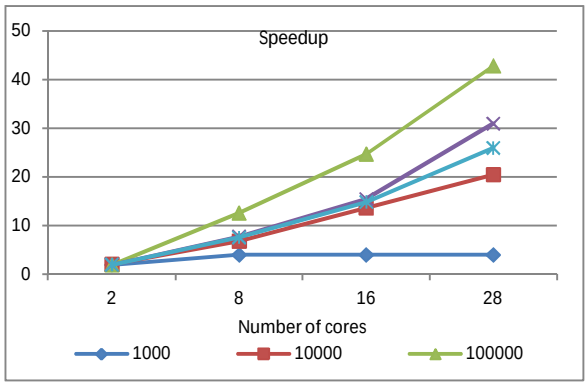

Figure 4. Speedup of MSA_BG algorithm using various numbers of cores and various numbers of iterations.

TABLE III. EFFICIENCY OF MSA_BG ALGORITHM USING VARIOUS NUMBERS OF CORES AND VARIOUS NUMBERS OF ITERATIONS

Efficiency					
Cores	*Iterations*				
	1000	*10000*	*100000*	*1000000*	*10000000*
2	1,00	1,03	1,00	1,00	1,00
8	0,50	0,85	1,57	0,97	0,95
16	0,25	0,85	1,54	0,97	0,93
28	0,14	0,73	1,53	1,11	0,93

 ISBN: 978-1-61208-283-7

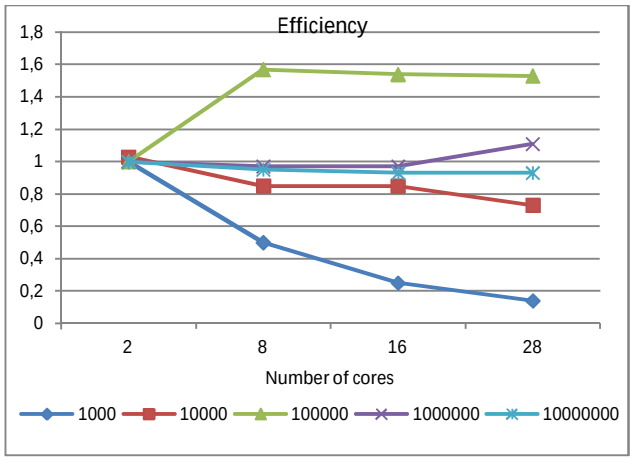

Figure 5. Efficiency of MSA_BG algorithm using various numbers of cores and various numbers of iterations.

Obviously, the parallel system shows good scalability with respect to both the cluster size and the workload size. In the case of small number of iterations the speedup obtained is not sufficient. Increasing the number of iterations results in speedup improvement because of the better utilization of the cores.

VI. Conclusion and Future Work

We presented an environment enabling secure access to the grid based services as follows: security, parallel program implementation execution and database access on distributed heterogeneous high-performance grid infrastructure. Web portal provides as services access and extraction of biological data and execution of parallel program implementations based on algorithm for comparative analysis of biological data. The proposed portal is verified experimentally for the case study of investigation the influenza virus variability.

An innovative parallel algorithm MSA_BG for multiple alignment of biological sequences that is highly scalable and locality aware has been designed. The MSA_BG algorithm is iterative and is based on the concept of Artificial Bee Colony metaheuristics and the concept of algorithmic and architectural spaces correlation. The metaphor of the ABC metaheuristics has been constructed and the functionalities of the agents has been defined. The conceptual parallel computationional model has been designed. The algorithmic framework of the designed parallel algorithm has been constructed.

MSA_BG algorithm has a hierarchical structure, which enables observing the principle of locality (independent calculations), and very high scalability (hives and swarms), so it is expected high efficiency implementations for petaflops supercomputers.

Parallel performance evaluation and profiling of multiple sequence alignment on the basis of MSA_BG algorithm heterogeneous distributed high-performance computation resources have been proposed in this paper. The case study is investigating viral nucleotide sequences and finding out consensus motifs and variable domains in the different segments of influenza virus.

Parallel performance parameters, such execution time and acceleration, have been estimated experimentally. The performance estimation analyses have shown that the parallel system is well balanced both in respect to the workload and machine size.

Acknowledgment

This work was supported in part by the Grant DFNI-M01/005 and in part by the Grant DCVP 02/1.

References

[1] H., Carrillo and D. Lipman, “The Multiple Sequence Alignment Problem in Biology,” SIAM Journal of Applied Mathematics, vol. 48, no. 5, 1988, pp. 1073-1082.

[2] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment," Journal of Computational Biology, vol. 1, no. 4, 1994, pp. 337–348, doi:10.1089/cmb.1994.1.337.

[3] W. Just, "Computational complexity of multiple sequence alignment with SP-score," Journal of Computational Biology, vol. 8, no. 6, 2001, pp. 615–23.

[4] S. Sze, Y. Lu, and Q. Yang, "A polynomial time solvable formulation of multiple sequence alignment," Journal of Computational Biology, vol. 13, no. 2, 2006, pp. 309–319, doi:10.1089/cmb.2006.13.309.

[5] J. Thompson, D. Higgins, and T. Gibson, “ClustalW: Improving the Sensitivity of Progressive Multiple Sequence Alignment through Sequence Weighting, Position-Specific Gap Penalties and Weight Matrix Choice,” Nucleic Acids Research, vol. 22, no. 22, 1994, pp. 4673-4680.

[6] K. Li, “ClustalW-MPI: ClustalW Analysis Using Distributed and Parallel Computing,” Journal of Bioinformatics, vol.19, no. 12, 2003, pp. 1585–1586.

[7] P. Borovska, V. Gancheva, and S. H. Ko, “Scaling of Parallel Multiple Sequence Alignment on the Supercomputer Juqueen,” the 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, 12-14 September 2013, Berlin, Germany, accepted.

[8] JUQUEEN – Configuration, http://www.fz-juelich.de/ias/jsc/EN/Expertise/Supercomputers/JUQUEEN/Configuration/Configuration_node.html, retrieved: 30.06.2013.

[9] D. Karaboga, “An Idea Based On Honey Bee Swarm for Numerical Optimization,” Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005, mf.erciyes.edu.tr/abc/pub/tr06_2005.pdf, retrieved: 20.05.2013.

[10] G. Lewis, G. Sipos, F. Urmetzer, V. Alexandrov, and P. Kacsuk, “The collaborative P-GRADE grid portal,” Proceeding of ICCS'05 Proceedings of the 5th international conference on Computational Science - volume part III, Springer-Verlag Berlin, Heidelberg, 2005, pp. 367-374.

[11] GenBank, http://www.ncbi.nlm.nih.gov/Genbank/, retrieved: 20.05.2013.

 ISBN: 978-1-61208-283-7