Cluster Computing of Nucleotide Sequence by Fractal Analysis

Deepa Mary Mathews, Research Scholar, Dr M.G.R.Educational & Research Institute, Chennai
Dr.K.S.M.Panicker, Professor, Federal Institute of Science and Technology [FISAT], Angamaly, Kerala

Abstract— The nucleotide sequence has capacity to represent information. Biological DNA represents the information which directs the functions of a living thing. Sequences can be read from the biological raw material through DNA sequencing methods. Cluster computing for nucleotide sequence by fractal analysis can be useful for identifying certain diseases. Very long sequences of nucleotide were checked with parallel search methods of cluster system. This paper uses box counting algorithm for fractal analysis. The implementation of fractal analysis in cluster system is done by parallel computing load balancing method. Finally performance of fractal analysis is tested. This paper, proposing new enhancement that can be included for nucleotide sequence analysis application programme by implementing MPI communication on clusters.

Index Terms— Box Counting method, Cluster Computing, Fractals, Fractal dimension, Message Passing Interface, Nucleotide sequence, Parallel Computing

1 INTRODUCTION

Understanding the relationship between genetic variation and biological function on a genomic scale will be helpful in providing fundamental new insights into biology, evolution and the path physiology of human diseases. DNA sequencing is the process of determining the exact order of nucleotides within a DNA molecule. It includes any method or technology that is used to determine the order of the four bases—adenine, guanine, cytosine, and thymine—in a strand of DNA. The sequence of the DNA of a living thing encodes the necessary information for that living thing to survive and reproduce. Therefore, determining the sequence is useful in fundamental research into why and how organisms live, as well as in applied subjects. DNA sequencing may be used to determine the sequence of individual genes, larger genetic regions (i.e. clusters of genes or operons), full chromosomes or entire genomes. Depending on the methods used, sequencing may provide the order of nucleotides in DNA or RNA isolated from cells of animals, plants, bacteria, or virtually any other source of genetic information. Because of the importance of DNA to living things, knowledge of a DNA sequence may be useful in practically any biological research. For example, in medicine it can be used to identify, diagnose and potentially develop treatments for genetic diseases. Similarly, research into pathogens may lead to treatments for contagious diseases.

Fractal analysis is a new tool that is being applied to surface science so that we can enhance our ability to work with surfaces. Fractal analysis involves finding the order within a disordered system and then describing the system in terms of non-integral dimensions. Fractal analysis is assessing fractal characteristics of data. It consists of several methods to assign a fractal dimension and other fractal characteristics to a dataset which may be a theoretical dataset or a pattern or signal extracted from phenomena including natural geometric objects, sound, market fluctuations, heart rates, digital images, molecular motion, networks, etc. In the area of computer graphics we use fractal methods to generate displays of natural objects and visualizations of various mathematical and physical systems. We can describe the amount of variation in the object detail with a number called the fractal dimension which is topological dimension value of the object.

Fractal methods have proven useful for modeling a very wide variety of natural phenomena, so fractals can be used to analyze many biologic structures not amenable to conventional analysis. There are many complex biological structures that cannot be easily modeled by simple shapes. The analysis of Trabecular bone, Regional distribution of pulmonary blood flow, pulmonary alveolar structure, mammographic parenchyma pattern as a risk for breast cancer, regional myocardial blood flow heterogeneity, fractal surfaces of proteins, distribution of arthropod body lengths are often fractal processes in nature.

Cluster computing method is an efficient form of information processing which emphasizes the exploitation of concurrent events in the computing process. Concurrency can be achieved by parallelism, simultaneity and pipelining method. This level requires the development of parallel processable algorithms, depending on the type of applications. Data dependency analysis is often performed to reveal parallelism among instructions. Vectorization is required among scalar operations within each instruction

Even though, the speed up that can be achieved by a cluster computer with n identical processors is at most n times faster than a single processor, in practice, speed up is much less, since some processors are idle at a given time because of conflicts over memory access or communication paths. Our algorithm to parallel search clusters tries to minimize these conflicts over memory access by certain mutual exclusion methods.

2 Box Counting Method For Fractals Computation

Nowadays fractals can be computed by fractal calculators, with front end and Java or .Net languages. Several methods exist for fractal computations. Similarity methods, geometric method and box counting methods are some of them. Similarity method and geometric method require you to measure the size of the object; in most of the situations measurement of the size of the object will not be possible.

In the box counting method, the Euclidean space [1] containing the image is covered with a grid, and then count how many boxes of the grid are covering part of the image. Then we do the same thing but using a finer grid with smaller boxes. By shrinking the size of the grid repeatedly, we end up more accurately capturing the structure of the pattern. In box counting method, instead of finding the exact size of the fractal, we count the number of boxes that are not empty. Let this number be N. Making the boxes smaller gives you more details, which is same as increasing the magnification. The magnification e is equal to 1/h. With this method, we can change the fractal dimension D = log N / log (1/h). Making the h smaller will make the dimension more accurate. For 3D fractals, you can do the same with cube instead of squares and for 1D fractals, you can use line segments. Fractal calculators (FDC) use this formula to calculate dimension. FDC can be freely downloaded for evaluation. It is fully functional except that offset sampling (a critical feature for good estimate of fractal dimension) is not enabled. Cluster based fractal dimension calculation has not studied yet. Actually very limited research groups works on the area of fractal computation in spite of its wide range of applications.

3 Design Of Cluster Algorithm For Nucleotide Fractals

Plotting a 2D graph for the given sequence was the first step for values of Adenine (A), Thymine (T), Guanine (G) and Cytosine (C) were entered from the terminal after considering various aspects of A, T, C & G such as molecular weight, electronic charge, strength of chemical bonds, Vander-Wall force etc. Then place an arbitrary grid over the plotted graph and count how many boxes in the grid are filled by the fractal structure. The process is then repeated with a grid half the size of the previous one. Now, the data from repeated operation are tabulated and plotted on a log-log plot with log (1/box size) as X-axis and log (no of boxes) on Y-axis. A linear regression is done to find the best fit. The slop of this line is used to calculate the fractal index.

Correlation Fractal Dimension Algorithm:
Compute fractal dimension D of a dataset A (box count approach)

Input: Normalized dataset A (N rows, with E dimensions/attributes each)
Output: Fractal dimension D

```
Begin
For each desirable grid-size r = 1/2^j , j= 1,2,…l
    For each point of the data set
          Decide which grid cell it falls in
           (say, the i-th cell)
         Increment the count Ci
            ( 'Occupancy')
        Compute the sum of occupancies
             S(r) = Σ Ci²
Print the values of log(r) and log(s(r))
          generating a plot;
Return the slop of the linear part of the
        plot as the fractal dimension D of
        the dataset A
End
```

Now, implementation of box counting on computer clusters:

Implementing a 500 character length nucleotide for fractal analysis with a P4 HT PC will take more than 25 minutes. So you can image the amount of time required for a nucleotide of 20,000 more needed for fractal analysis. So we decided to implement fractal analysis by a parallel computation method of a cluster. We have designed a load balancing algorithm for each processor.

It is mainly proposed for a cluster based super computing system where the communication cost is not very large as resources are connected through a high band width network. Here, at every status exchange time periods Ts, each Pi communicates its status (queue length, estimate of the arrival rate) to all its buddy processors. At each estimation instant Te, every processor calculates the queue length on buddy processors using the estimated arrival rate and exact service rate of a buddy processor. Pi will make a decision of job migration if its queue length is greater than the average queue length in its buddy set.

In this design, each Pi estimates its arrival rate, service rate and the load at each status exchange instant. At each estimation instant, P_i calculates the load on processors. Based on this calculated buddy load, each processor calculates the average load in its buddy set. P_i will make a decision of job migration if its load is greater than the average load in its buddy set and will try to distribute its load such that load on all buddy processors get finished at almost the same time, taking in to account the node's heterogeneity in terms of processor speed. This average buddy load can be calculated using the following rela-

tionships.

Let S_i denote the weight of a processor P_i which is a normalized measure of its speed. Therefore, a value of 2 for S_i means that P_i will take half amount of time to execute a job than the time taken by the reference processor having a value of 1 for (Normalized speed measure from P_i) S_i. Here, each P_i will calculate the average normalized buddy load using the value of Estimated load on buddy processor P_k calculated by P_i at time T ($L_{k,i}$ (T)) and S_i by the following equation:

$$NBL_{avg}=\Sigma_{\ k \in \text{buddyset i}} S_k * L_{k,i}(T)/\Sigma_{\ k \in \text{buddyset j}} S_k$$

NBL_{avg} indicates the average load for a reference processor. P_i is considered as a sender processor if $NBL_{avg} < S_i * L_i(T)$, where $L_i(T)$ is estimated load on P_i at time T. Now, P_i will try to transfer its extra load to all receiver processors P_k such that they receive extra load based on their current load ($L_{k,i}$ (T)) and processor weight (S_k).

4 Implemenation Of Algorithm On Dhakshina Cluster Series-I

Dhakshina Cluster Series I is a High Performance Computing System developed by the FISAT faculty & Free Software Cell using the Beowulf Architecture, with a peak speed of 180 Giga Flops. Linux clusters network topology, programming environments, batch and interactive computations, are key factors for the implementation of fractals computation. The control node provides services such as Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), and Network File System (NFS). In addition, the portable batch system (PBS) and the scheduler are usually installed on this machine. Redundancy is required on hardware and data because; unlike in the case of compute nodes, the failure of the control node may affect the availability of the entire cluster. The use of redundant fans, redundant power supply and RAID (to protect the data) is common. Cluster manager takes control of the nodes and collects Simple Network Management Protocol (SNMP) alarms. It is difficult to store large amounts of data on a single server or on one of the cluster nodes; there is a need for a SAN (Storage Area Network), or dedicated servers. The cluser nodes in the Dhakshina are interconnected using the Gigabit Ethernet LAN. The data transfer takes place at the speed of around 990 Megabit per second among the servers and around 665 Mb per second among the client nodes.

4.1 The Application Programming Interface

Dhakshina uses the Message Passing Interface (MPI) to enable the application of nucleotide sequence analysis to communicate among nodes. MPI is equally suitable for large parallel machines and for smaller environments such as a group of workstations. Since clusters of workstations are readily available at many institutions, it has become common to use them as a single parallel computing resource running MPI programs. The MPI standard facilitates portability and platform independent computing. As a result, users can enjoy cross-platform development capabilities as well as transparent heterogeneous communication without much difficulty.

MPI's goals are high performance, scalability, and portability. Generally considered to have been successful in meeting these goals, it is a crucial part of fractal analysis of nucleotide sequence. Most MPI implementations consist of a specific set of routines (API) callable from Fortran, C, or C++ and from any language capable of interfacing with such routine libraries. Bio-perl and mpipearl library functions of the high level object oriented programming language Perl helped this project work to identify fractional dimension. MPICH is a freely available, portable robust and flexible implementation of the MPI (Message Passing Interface), a standard for message-passing libraries. It implements both MPI-1 and MPI-2. MPICH is a developed program library. MPICH is a multi-platform, configurable system (development, execution, libraries, etc) for MPI. It can achieve parallelism using networked machines or using multitasking on a single machine. LAM/MPI is a high-quality open-source implementation of the Message Passing Interface specification, including all of MPI-1.2 and much of MPI-2. From its beginnings, it was designed to operate on heterogeneous clusters. Several transport layers, including Myrinet, are supported by LAM/MPI. With TCP/IP, LAM imposes virtually no communication overhead, even at gigabit Ethernet speeds.

5 Peformance Evaluation

The benchmark tests were conducted using HPLinpack 1.0a (High Performance Linpack), a common benchmarking utility for high performance systems, developed by Innovative Computing labs, UTK. Initially, the hpl.conf file was modified. The configuration was fine-tuned to extract maximum output from the nodes. The program with sequence analysis for fractal computation as proposed by algorithm -1 was then run on the cluster using the Message Passing Interace (MPI).

5.1 Test Results

The results of the benchmark test (for 32 nodes) are provided below:

==========

HPLinpack 1.0a -- High-Performance Linpack benchmark -- January 20, 2004

Written by A. Petitet and R. Clint Whaley, Innovative Computing Labs., UTK

==========

An explanation of the input/output parameters follows:

T/V: Wall time / encoded variant.

N: The sequence of characters of A,T,C,G in some order.
NB : The partitioning blocking factor.
P : The number of process sequence.
Q : The number of process log values for calculating fractals.
Time : Time in seconds to solve the linear system.
Gflops : Rate of execution for solving the linear system.
The following parameter values will be used:
N : 27386
NB : 170 175 180
PMAP : line by line sequence mapping
P : 400 Q : 80
PFACT : Left
NBMIN : 2
NDIV : 2
RFACT : Left
BCAST : 1ring
DEPTH : 0
SWAP : Mix (threshold = 640)
L1 : transposed form
U : transposed form
EQUIL : no
ALIGN : 8 double precision words

- The 2D graph is generated for each test.
- The following scaled residual checks will be computed:
  1) ||Ax-b||_oo / (eps * ||A||_1 * N)
  2) ||Ax-b||_oo / (eps * ||A||_1 * ||x||_1)
  3) ||Ax-b||_oo / (eps * ||A||_oo * ||x||_oo)
- The relative machine precision (eps) is taken to be 1.110223e-16
- Computational tests pass if scaled residuals are less than 16.0

```
==========================================
T/V             N    NB     P     Q         Time          Gflops
------------------------------------------------------------------------------
WR00L2L2      27386   170    400    80         244.22         5.607e+01
------------------------------------------------------------------------------
||Ax-b||_oo / ( eps * ||A||_1  * N  ) = 0.0021722 ....   PASSED
||Ax-b||_oo /( eps*||A||_1*||x||_1  ) =  0.0039217.  PASSED
||Ax-b||_oo /(eps*||A||_oo*||x||_oo ) = 0.0006798  PASSED
==============================================
T/V             N    NB     P     Q         Time          Gflops
------------------------------------------------------------------------------
WR00L2L2      27386   175    400    80         242.51         5.647e+01
------------------------------------------------------------------------------
||Ax-b||_oo /( eps *||A||_1 * N) =      0.0021881 ..... PASSED
||Ax-b||_oo / ( eps *||A||_1* N ) =     0.0021881 ..... PASSED
||Ax-b||_oo/(eps* ||A||_1*||x||_1) =0.0039504 ..... PASSED
||Ax-b||_oo/(eps*||A||_oo*||x||_oo ) =0.0006847... PASSED
==============================================
T/V             N    NB     P     Q         Time          Gflops
------------------------------------------------------------------------------
WR00L2L2      27386   180    400    80         266.51        5.138e+01
------------------------------------------------------------------------------
||Ax-b||_oo /( eps * ||A||_1  * N) =   0.0023373 ..... PASSED
||Ax-b||_oo /(eps*||A||_1*||x||_1)= 0.0042198 ..... PASSED
||Ax-b||_oo/(eps*||A||_oo*||x||_oo)= 0.0007314 ... PASSED
==============================================
Finished     3 tests with the following results:
             3 tests completed and passed residual checks,
             0 tests completed and failed residual checks,
             0 tests skipped because of illegal input values.
------------------------------------------------------------------------------
End of Tests
```

6 Conclusions

We have executed an application programme for fractals analysis by parallel programming technique on cluster system and executed it in successful manner. However, we have to check the various codon of DNA sequence and should keep the corresponding fractal values in a database. By comparing the fractal values of the part of available sequence, we can predict the characteristic of the species. Also, we can determine the possibility of occurring certain hereditary diseases. Depending on the generation of information by biologists using fractal analysis method we can predict various cases of fractal dimension. Commodity processor-based clusters have rapidly become the compute systems of choice for computational modeling and simulation in many scientific fields. Many biological computations such as genomic and protein analysis are increasingly being performed on large-scale clusters based on Linux.

7 Acknowledgments

This research was supported by Centre for High Performance Computing Lab at Federal Institute of Science & Technology [FISAT], Cochin, India. Bio-Informatics Centre of University of Kerala contributed significant tools under the sponsorship of University Grants Commission [UGC], India.

References

[1] Caetano Traina Jr, Agma Traina, Leejay wu, Christos Falouts, Dept. of Computer Science & Statistics-University of Sao Paulo, Brazil, "*Fast feature selection using fractal dimension, Fractal Analysis – Statistical Method*", Scitech Publishers, Page 243-245.

[2] Donald Hearn & Pauline Baker, Computer Graphics, Prentice Hall of India, 2003, page 362-375

[3] Daniel Osman, David Newitt, '*Fractal based image analysis of Human Trabecular bone using the box counting algorithm*' , Fractals Vol.6, No.3 (1998)275-283, World Scientific Publishing Company

[4] Achuthsanker S Nair, " *Computational Biology & Bioinformatics: A Gentle Overview*", Communications of Computer Society of India, 2007, Page 21

[5] Ana L.N. Fred, Member, IEEE, and Anil K. Jain, Fellow, IEEE, Combining Multiple Clusterings Using Evidence Accumulation, IEEE Transactions On Pattern Analysis And Machine Intelligence, VOL. 27, NO. 6, JUNE 2005, Page 835.

[6] Dinesh Kadamuddi and Jeffrey J.P. Tsai, Fellow, IEEE, Clustering Algorithm for Parallelizing Software Systems in Multiprocessors Environment, IEEE Transactions On Software Engineering, VOL.

26, NO. 4, APRIL 2000, Page 340

[7] Hyo Jung Song, Member, IEEE, and Andrew A. Chien, Senior Member, IEEE Computer Society, ' Feedback-Based Synchronization in System Area Networks for Cluster Computing', IEEE Transactions On Parallel And Distributed Systems, VOL. 16, NO. 10, October 2005, Page 908

[8] Jian Yu, Member, IEEE, 'General C-Means Clustering Model' IEEE Transactions On Pattern Analysis And Machine Intelligence, VOL. 27, NO. 8, August 2005 Page, 1197

[9] Alexander Marquardt, Vaughn Betz, and Jonathan Rose, 'Speed and Area Tradeoffs in Cluster-Based FPGA Architectures', IEEE Transactions On Very Large Scale Integration (VLSI) Systems, VOL. 8, NO. 1, February 2000, Page 84.

[10] Johan Vromans 'Programming PEARL-Quick Reference Guide' O'REILLY Publishers, 2004

[11] Ruchir Shah, Bhardwaj Veeralli, Senior member IEEE, 'On the Design of Adaptive and Decentralized Load balancing algorithms with Load estimation for computational grid environments', IEEE Transactions on parallel and distributed systems, Vol18, No.12, Dec 2007.

[12] Herbert F. Jelinek , School of Community Health Charles Sturt University, Eduardo Fernandez, Instituto de Bioingeniera,Universidad Miguel Hernandez ,Elche , Spain, Neurons and fractals: how reliable and useful are calculations of fractal dimensions.

[13] Chaos, Solitons and Fractals 11 (2000) 825±836, Fractals related to long DNA sequences and complete genomes, Bai-lin Hao , H.C. Lee , Shu-yu Zhang

[14] Turanov AA, Lobanov AV, Fomenko DE, Morrison HG, Sogin ML, Klobutcher LA, Hatfield DL, Gladyshev VN (January 2009). "Genetic code supports targeted insertion of two amino acids by one codon". Science 323 (5911): 259–61. doi:10.1126/science.1164748.