A Bio-Informatics Model for Task assignment and Fault Removal in Cloud Architecture

[1]Umesh Dwivedi, [2]Dr. Harsh Dev

Abstract

Cloud computing is an emerging field and currently is being used in almost all the economical and manufacturing fields. Main reason of attraction towards this field is its number of attractive and beneficial features. Cloud architecture is designed in a fashion, that it can execute the tasks provided by the client in minimum time, minimum cost and without any fault. These assigned tasks need some resources for their execution. These resources are to be provided by the data centrers with help of virtual machines available there. Proper assignment of virtual machines to each task is very difficult. If this is not done properly, there is huge wastage of resources. Number of tasks are left unexecuted just because they are not getting their required resources or they have to wait for much longer time. Even if virtual machines get assigned to the tasks, there is no guarantee of proper resource assignment. Either resources are provided in bulk or there is a scarcity of resources for the task. It is also an arduous process. This paper uses two biotechnology algorithms named FASTA and BLAST for handling this proper resource assignment problem. These algorithms work in biological field for DNA mapping. They promptly map the complete strip in parallel. This paper uses these biological algorithms for resource assignments for the tasks to be executed. This paper explains about these biological algorithms and how they are nifty in resource assignment problem and its solution in very less amount of time. Finally, error occurring conditions which may arise while approximation adoption of result are explained and solved in detail.

Keywords: *cloud computing, fault tolerance, FASTA, BLAST, VM, Task, Resources.*

I. INTRODUCTION

Now a days, cloud computing has become a demanding field. Every organisation is shifting towards digital world whereas every digital organisation also shifting towards cloud computing. Cloud computing can be defined as the new method of using all computing resources as Hardware, software, storage in a remote manner. Actually, they are located at remote locations and provide services with the help of internet. [1][2] .This is immensely popular and acceptable because of its unique features

[1] Ph.D. Scholar, Dr. A.P.J. Abdul Kalam Technical University, Lucknow,Uttar Pradesh, India.
[2] Professor and Dean in Research Wing, Pranveer Singh Institute of Technology, Kanpur

Received: 22 Feb 2020 | Revised: 28 Mar 2020 | Accepted: 30 Apr 2020

like money saving, no personal infrastructure is required, no personal hardware or software installation is required, no platform bounding is there, open source software development environment is available plus all these services are also available on payment basis and payment is done on the basis of its use. Cloud architecture is highly scalable, so any new development is easily accepted by this architecture.

In spite of all these qualities, still there are many research issues to be fully addressed[2]. A single fault can damage the whole architecture. This gave rise to a new field called fault tolerance. Foremost and required benefits of implementing fault tolerance in cloud computing are failure recovery, reliability towards architecture, cost reduction and large scale availability.[3][4]. When a fault occurs, the fault tolerance techniques provide appropriate mechanisms to the software system to prevent system failure.[2]. Many researchers have given their important outcome in this field. This paper also tries to act up short in the same direction. In the field of biotechnology FASTA and BLAST are the two searching algorithms which are used in the current paper for the task assignment and fault removing in this architecture.

This paper is dived into sections: Section 2: explores the fault tolerance in cloud computing field. Section 3: makes a review of the research work done in the fault tolerance field. Section 4 focuses on the proposed model using FASTA and BLAST algorithms. Section -5 explains the algorithm proposed with the help of FASTA and BLAST and error occurring conditions and their solutions. Section 6: Explains the conclusion of this paper based on the algorithms and future scope and direction of research related to the developed model.

II. FAULT TOLERANCE IN CLOUD COMPUTING

Client gives his identification to the service provider to get help at the time of fault occurrence. Service provider creates a list of all connected clients and gives them some identification number. At the time of any fault service provider helps to the client and tries to make the transaction successful[5].

Role of service provider is very much important while assigning the resources systematically throughout the request of each client, the service provider has to keep uniformity among all computing systems and the connected clients. In this situation, resource manager has to constantly observes the running state of physical and virtual resources. He manages with the help of a database of catalogue and graph created by himself which represents the actual state of resources and the connected clients[5].

At the time of resource utilization,first thing is to keep a watch on load and resource utilization. This is measured with the Hardware capacity such as CPU capacity and bandwidth. All this is to be handled by the resource manager. The performance of the system totally depends on fault tolerance method and on the end- to-end quality of service can be maintained collectively with these steps because quality of service is needed in both the situations at the time of failure and failure free periods. [6]. The resource manager is fully dedicated towards the stability of cost of resource, performance and fault model quality for the service providers [4]. Two new techniques of biotechnology field are being used here for the purpose of searching the faulty node and providing it proper sequence so that node can be executed without any fault. These two techniques are FASTA(Fast Alignment) and BLAST(Basic local alignment search tool).

III. REVIEW OF PREVIOUS RESEARCH WORK

3.1 FASTA and BLAST technique Analysis

BLAST stands for Basic Local Alignment Search Tool. It is one of the most used bioinformatics software. It was developed in 1990 and since then has been easily accessible to everyone through official website of NCBI. In this software, the sequence to be compared is submitted in FASTA format and the output can be obtained in plain text, html or XML.

BLAST works on the principle of searching for localized similarities between the two sequences. It searches for neighbourhood similarities.

It is used for many purposes like DNA mapping, comparing two identical genes in different species and for creating phylogenetic tree.

FASTA is a DNA and protein sequence alignment software package which was originally developed for searching similarities in protein sequences. It was described as FASTP by its developers David J. Lipman and William R. Pearson in 1985.

When a given nucleotide or amino acid sequence is submitted in FASTA format, it searches for the corresponding sequences in the available database by using “local sequence alignment” to find similar sequences. The FASTA programs find regions of local or global similarity between protein or DNA sequences, either by searching protein or DNA databases, or by identifying local duplication with a sequence. It can be used to explore functional and evolutionary relationships between sequences and to identify members of gene families.

Basic Local Alignment Search Tool (BLAST) searches known nucleotide or amino acid sequence in the database and it searches which is most similar to an unknown input sequence. It searches those sequences which are closest in biological similarity to the query [8,9].

3.2 Approaches of BLAST and FASTA fault tolerance:

Many papers describe the BLAST algorithm and concepts for similarity searching in detail. Scot E Dowd, Joaquin Zaragoza[7] describes a software application, termed Windows .NET Distributed Basic Local Alignment Search Toolkit (W.ND-BLAST), which enhances the BLAST utility by improving usability, fault recovery, and scalability in a Windows desktop environment. This software provides several layers of fault tolerance and fault recovery to prevent loss of data if nodes or master machines fail.

BLAST can be parallelized to achieve linear or even super-linear speedups. In the database segmentation model[10], a large sized database is break up in several database with equal size. Every node of database stores one separate part of the database. Independent parts of the database are searched on each processor or node, and results are collated into a single output file. NCBI-BLAST implements database segmentation by multi-threading[10,11,12,13].

Query Segmentation splits the set of query sequences such that each node in a cluster or CPU searches a fraction of the query sequences. Several BLAST searches can execute in parallel on different queries. BLAST searches using query segmentation on a cluster which replicate the entire database on each node's local storage system. If the database is larger than core memory, query-segmented searches suffer the same adverse effects of disk I/O as traditional BLAST[10,11,12,13].

One popular program for constructing a PSSM and comparing it with a database of sequences is Position-Specific Iterated BLAST (PSI-BLAST).Alejandro A. Schaffer[14] described new software package, IMPALA. That is designed for the complementary procedure of comparing a single query sequence with a database of PSI-BLAST-generated PSSMs. Paper illustrate the use of IMPALA to search a database of PSSMs for protein folds, and one for protein domains involved in signal transduction. IMPALA employs a more refined analysis of statistical significance than PSI-BLAST. It guarantees the output of the optimal local alignment by using the rigorous Smith–Waterman algorithm. Also, it is considerably faster when run with a large database of PSSMs than is BLAST or PSI-BLAST when run against the complete non-redundant protein database.

IV. PROPOSED MEHTODOLGY

A fault we can say is an erroneous state of hardware or software. Result of which is physical defect, design flaw, or operation error. An error is next state of a fault. All faults are not metamorphosed into errors. When fault is converted to the error we can say that fault is activated. Otherwise it is masked.. All errors do not lead to failure, but they may provoke failure[15].

In the system of cloud computing, dependency of one operation may be on another operation and other operation may dependent on third one. There can be a chain of these type of snags, for example, faults may be because of sequencing errors, the database may be corrupted that contains sequence information, errors in the form of wrong analysis results or misleading statistical values assigned to the result, and failure because of wrong conclusions.

A fault-tolerant system should continue its service even if there are faults [16]. Fault injection is a deliberate introduction of faults into the system to discover errors and failure modes. By injecting faults on purpose, we can identify the errors that the system produces and make appropriate improvements to the system.

The faults considered here are sequencing errors, which are divided into three parts, substitution, insertion, and deletion.

Substitution occurs when a base is misidentified as another.

Insertion occurs when an extra base is added.

Deletion occurs when a base is omitted.

The model emulates those sequence errors. It explains that

1. faults occur in a random fashion at each base,
2. no contiguous faults are injected but can occur,
3. The faulty base position follows a uniform distribution,

So we can say that there are three kinds of faults (substitution(F_s), deletion(F_d), insertion(F_i)).

Total fault rate is defined as

Ft=Fs+Fd+Fi

Actual cloud architecture consists following main entities as shown in figure1:

1. Task/Cloudlets
2. Cloud Information Service
3. Datacenter Broker
4. Datacenter
5. Host Virtual Machine (VM)

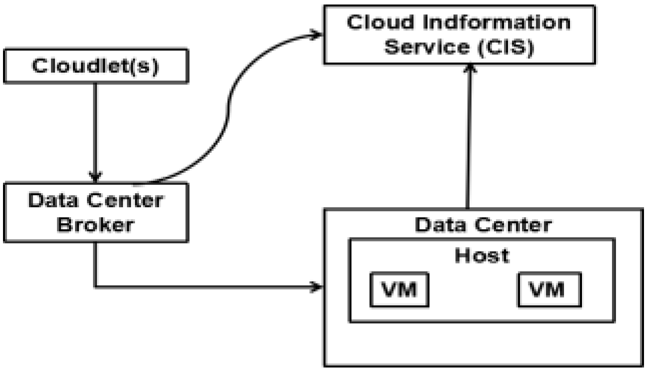

Figure 1

A datacenter consists of many hosts. Each host can have multiple Virtual Machines. Every host in datacenter takes the virtual machine on the basis of their hardware specifications (processing power, memory and bandwidth etc.) Processing power is calculated in MIPS(Milion Instruction Per Second).

Once a datacenter is created it has to be registered on a CIS.

When a user requests a service, cloudlet is (task) send to the datacenter broker which collects the information of available resources form the CIS, and then as per the scheduling policy defined in the datacenter broker, it allocates the tasks to each Virtual Machine.

Task assignment to proper virtural machines is a big problem. Every individual task needs some CPU time to be executed, some specific bandwidth needed and some part of memory is also required. In each case, task has to be assigned to proper virtual machine having similar properties of CPU, network bandwidth and memory. If resources are very much greater then the requirement of task then there is a wastage of resources. In other case if resources are less then the requirements of task then it will not be able to execute it. Proper matching is required so that the task gets executed in minimum amount of time and resources without wasting the resources.

Individual resource matching is very difficult and time consuming task and if proper matching has not been done, there may be a possibility of fault which may be converted to an error.

To solve this problem two algorithms FASTA and BLAST are being used in this paper which are basically made up for the field of Biotechnology. BLAST stands for Basic Local Alignment Search Tool. In this software the sequence to be compared is submitted in FASTA format and the output can be obtained in plain text, html or XML.

BLAST works on the principle of searching for localized similarities between the two sequences. It searches for neighbourhood similarities.

It is used for many purposes like DNA mapping, comparing two identical genes in different species and for creating phylogenetic tree.

When a given nucleotide or amino acid sequence is submitted in FASTA format, it searches for the corresponding sequences in the available database by using “local sequence alignment” to fond similar sequences. The FASTA programs find regions of local or global similarity between protein or DNA sequences, either by searching protein or DNA databases, or by identifying local duplication with a sequence. It can be used to explore functional and evolutionary relationships between sequences and to identify members of gene families.

This property of FASTA and BLAST is used in the cloud computing. Apart from comparing individual resource with the required resource of the task we create a strip of resources needed by the task as follows.

CPU cycle time eded	Bandwidth needed	Memory space eded

Figure 2

This strip in short can be seen as shown in figure 3

Figure 2

Every virtual machine details are first stored in table as shown in figure 4.

VM 1			
VM 1			
VM 1			
VM 1			

VM 1			
VM 1			

.

.

.

.

.

.

Figure 4

Every VM strip(Resources detail) is compared with the strip of task with the help of BLAST which compares in parallel. Individual strip will be compared with the individual virtual machine resources in parallel.

In every case complete match is not possible so a threshold value is decided up to that mark virtual machine is said to be matched and task is assigned to that virtual machine. If matching is beyond the threshold value then the VM is not said to be matched as shown in figure 5.

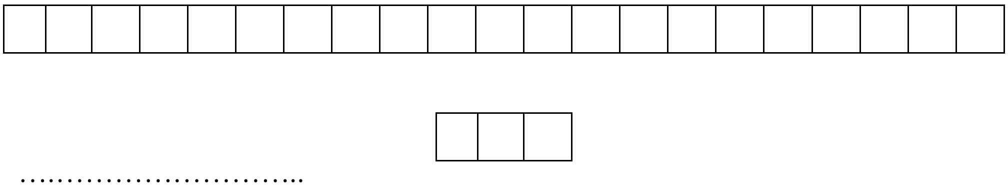

Figure 5

These informations are also stored in table with different virtual machines as shown in figure 6.

Number of VM	**CPU peed gigahertz (GHz)**	**Bandwidth required (MBPS)**	**Memory required (MB)**
VM 1	1.48	12	120
VM 1	1.54	10	125
VM 1	1.72	15	122
VM 1	1.98	14	132
VM 1	1.23	14	125

Figure 6

V. ALGORITHM AND FAULT OCCURING CONDITION

5.1 Algorithm for proper task assignment using BLAST and FASTA

Algorithmic representation of whole process is as follows.

```
f_s = size of each strip
f_vm=size of each strip of virtual machine stored in table
Seq_str=total number of virtual machine  strips stored in table
func resource_match
Input: f_s, f_vm, , seq_str
Local variable: r
for i = 1 … seq _str..lengt * f_s
do
Comp f_s with f_vm  up to the limit of threshold value
while (visited(r))
setVisited(r)
If f_s□ not f_vm
F_s  f_vm+ upto  seq_str.length*f_s

Output: f_vm with its index number
```

5.2 Fault occurring condition and their solution

There can be two type of errors false negative (FN) and false positive (FP). A FN occurs when a match is there up to negative threshold limit so it is acceptable.. A FP occurs when a match is there up to positive threshold limit so it is also acceptable. TP(True Positive) is the area where exact match is there between virtual machine and task to which resource to be assigned. TN(True Negative) is the area where no match is there and virtual machine can not be assigned to the task. as shown in figure 6

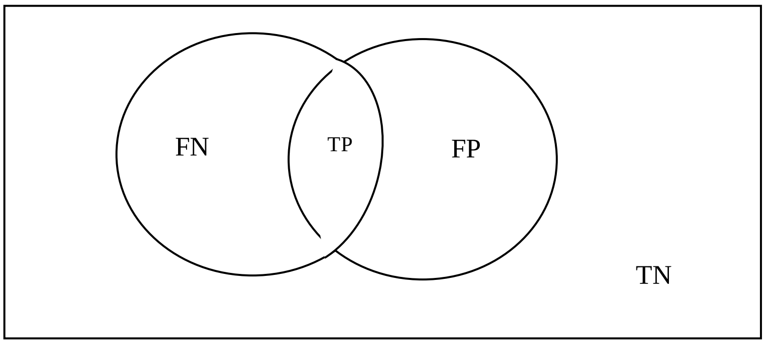

Figure 7

The false negative error rate rFN(F) and false positive error rate rFP(F) with the set of fault rate F are defined respectively as:

r_{FN}(F)= false negative hits/hits in left circle outside the TP

r_{FP}(F)=false positive hits/hits in right circle outside the TP

VI. RESULT ANALYSIS

Algorithm implementation is done with the help of cloudsim simulator. Implementation is done on the basis of two parameters: Time consumption and Resource utilization.

6.1 Time consumption basis:

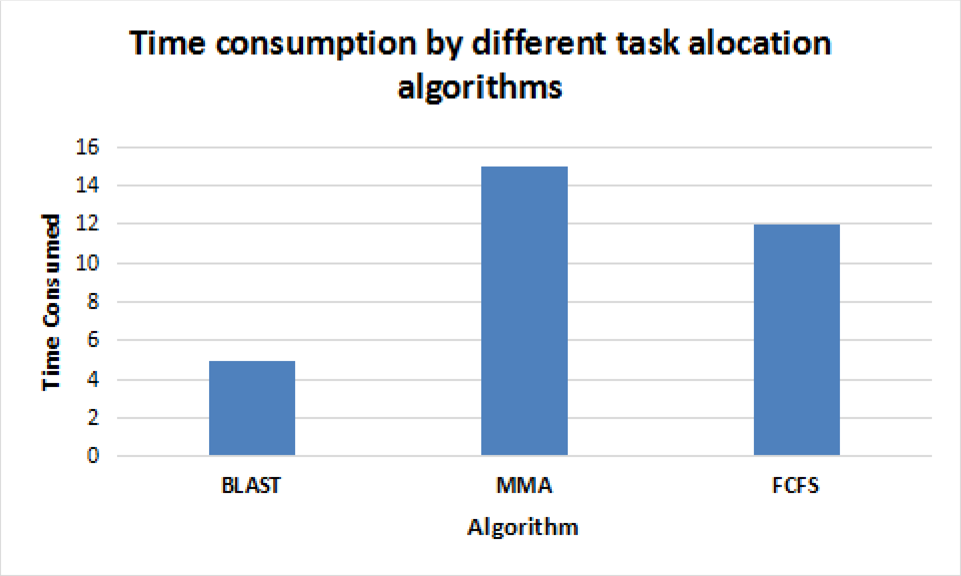

Figure 8

Received: 22 Feb 2020 | Revised: 28 Mar 2020 | Accepted: 30 Apr 2020

On the basis of time consumption different algorithms are compared. Figure 8 shows the result. It clearly shows that time consumed by BLAST algorithm is much less than other algorithms.

6.2 Resource utilization:

Resource utilization is also compared between three algorithms with the help of cloudsim simulator.

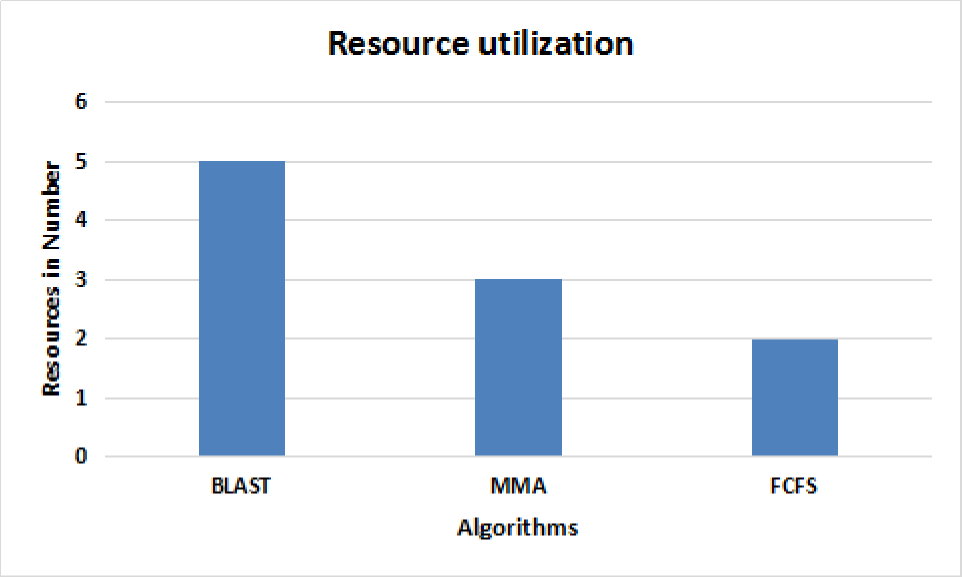

Figure 9

Result shown in figure 9 explains that resource utilization is maximum in case of BLAST algorithm.

This all shows that BLAST algorithm gives better results as compared to other algorithms on the basis of time consumption and resource utilization.

VII. CONCLUSION

The objective of this research paper is to represent a review of previous work related to previous fault tolerance techniques. We focus on resource assignment issue and its proper allocation with minimum time and least resource loss. This algorithm is explained with considering some differences between the resources needed by the task and the resources provided by the virtual machines. Some faulty conditions which can arise while considering approximate outputs which are also explained. FASTA and BLAST techniques are not used prior in cloud computing field for task assignment at proper virtual machines. It is a lengthy task and proper assignment of resources can not be done because many parameters are to be matched in parallel and collective result decides whether the task can be assigned to the virtual machine or not. The problem is gratifyingly handled by these two biotechnology algorithms.

VIII. FUTURE DIRECTIONS

This algorithm works with the strip of tasks simultaneously. This feature of this algorithm makes it much faster the other algorithms.In future these algorithms can also be used to find the error and error occurring conditions in a cloud architecture in parallel fashion by checking number of virtual machines simultaneously.

REFERENCES

[1] Sun Microsystems, Inc. “Introduction to Cloud Computing Architecture” White Paper 1st Edition, June 2009

[2] Swati Pawar, Jayoti Vidyapeeth, “A Survey on Fault Tolerance and its Techniques in Cloud Computing”, International Journal of Engineering Technology and Computer Research (IJETCR), Volume 3; Issue 3; May-June 2015; Page No. 116-120, Available Online at www.ijetcr.org

[3] AnjuBala, Inderveer Chana,” Fault Tolerance□Challenges, Techniques and Implementation in Cloud Computing” IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012 ISSN (Online): 1694-0814 www.IJCSI.org

[4] Y.M. Teo, B.L. Luong, Y. Song, T. Nam, ”Cost□Performance of Fault Tolerance in Cloud Computing, International Conference on Advanced Computing and Applications, (Special Issue of Journal of Science and Technology, Vol. 49(4A), pp. 61-73), Ho Chi Minh, Vietnam, October 19-21, 2011.

[5] Ravpreet Kaur, Manish Mahajan , “Fault Tolerance in Cloud Computing”, International journal of Science Technology & Management (IJSTM) ISSN: 2229-6646 Presented in National Conference on RTICCN-2015 at CGC-COE , Landran , Mohali(Punjab) on 26-27th March 2015.

[6] Jhawar, Ravi, Vincenzo Piuri, and Marco Santambrogio., "Fault tolerance management in cloud computing: A system-level perspective." Systems Journal, IEEE 7.2 (2013): 288-297.

[7] Scot E Dowd, Joaquin Zaragoza, Javier R Rodriguez,, Melvin J Oliver, Paxton R Payton, “Windows .NET Network Distributed Basic Local Alignment Search Toolkit (W.ND-BLAST)”, BMC Bioinformatics 2005, 6:93,Pg 1-14.

[8] Altschul S, Gish W, Miller W, Myers E, Lipman DJ, Basic Local Alignment Search Tool. Journal of Molecular Biology 1990,215:403-410.

[9] Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ, “ Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Res 1997, 25:3389-402.

[10] R. Bjornson, A. Sherman, S. Weston, N. Willard and J. Wing. Turboblast, “A parallel implementation of blast based on the turbohub process integration architecture”, IPDPS 2002 Workshops, April 2002.

[11] R. Braun, K. Pedretti, T. Casavant, T. Scheetz, C. Birkett, and C. Roberts, “Parallelization of local BLAST service on workstation clusters”, Future Generation Computer Systems 17(6)},745-754, April 2001

[12] .J. D. Grant, R. L. Dunbrack, F. J. Manion and M. F. Ochs,”BeoBLAST: distributed BLAST and PSI-BLAST on a Beowulf cluster”, Bioinformatics, 18(5)}:765-766, 2002.

[13] M. Dumontier, and Christopher WV Hogue , NBLAST: a cluster variant of BLAST for NxN comparisons", BMC Bioinformatics, 3:13, 2002,

[14] Alejandro A. Schaffer, Yuri I. Wolf, Chris P. Ponting, Eugene V. Koonin, L. Aravind, Stephen F.Altschul, "IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices", National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD 20894, USA and Department of Biology, Texas A&M University, Biological Sciences Building West, College Station, TX 77843, USA Received on March 19, 1999 ; revised on July 28, 1999; accepted on August 4, 1999, Vol 15.

[15] D. J. Sorin, "Fault Tolerant Computer Architecture," Synthesis Lectures on Computer Architecture, vol. 4, no. 1, p. 2, 2009.

[16] A. Avizienis, J. C. Laprie, B. Randell and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," Dependable and Secure Computing, IEEE Transactions on, vol. 1, no. 1, pp. 11-33, 2004.