Improved Fault Tolerant Elastic Scheduling Algorithm for Cloud Computing

Sukhbir Singh and Raman Kumar

Abstract— The paper focus on Fault Tolerance, a long standing problem in cloud computing by extending Primary Backup model to include cloud features such as virtualization and elasticity. Fault tolerance is a challenging work in Cloud Computing as virtual machines are the basic computing instances rather than hosts that enable virtual machines to migrate to other hosts. The on demand provisioning of resources makes the cloud elastic thus resources can be added on demand and when no longer requirement exists, the resources can be released. Prevailing fault tolerant scheduling algorithms which studied PB model approach in cloud computing investigated single Host crashes. The proposed algorithm, improved Fault Tolerant Elastic Scheduling Algorithm identified as IFTESA, is able to handle two host crashes concurrently. IFTESA attempts to boost performance by involving virtual machine migration and overlapping technique. The constraints have been investigated to full fill fault tolerance needs.

Index Terms— Cloud, elasticity, fault tolerant scheduling, GR, HAT, primary-backup model, RTH.

1. INTRODUCTION

The cloud is a need based delivery of software applications, platform and infrastructure on payment as per requirement which can be increased or decreased any time. We can submit the job in the cloud to get the processed information back. A job or cloudlet is a particular task which is to be executed on the virtual machine. Whenever we try to execute the job, the cloudsim [2] framework do following things first cloud information service (CIS) component is created; this is a kind of Registry which contains the information about the resources that are available in the cloud. Resources are data centers and each data center will have hosts and each host may have set of virtual machines. The CIS takes care of the registry of the data centre when created. Each data centre has some characteristics and these are the characteristics that you specify for your host. Virtualization says that the host will be virtualized into a number of virtual machines and every virtual machine has some characteristics. Scheduling – a well planned perspective to attain fault tolerance by assigning many jobs on unique virtual machines. A scheduler or broker submits the job in data center. This scheduler speaks to the CIS and retrieves the resource information (Data Center characteristics) which is registered with the CIS. The set of jobs will be submitted to the scheduler who has the details of the data center, now the scheduler interacts with the data center directly and assign these jobs to the vm on host in the data center.

- *Sukhbir Singh is currently pursuing Master of Technology in Computer Science and Engineering in IKG PTU Jalandhar, Punjab, India E-mail: sukhbir@live.in*
- *Dr. Raman Kumar is currently Assistant Professor in Department of Computer Science and Engineering in DAV Institute of Engineering and Technology, Jalandhar, Punjab, India E-mail: er.ramankumar@aol.in*

2. RELATED WORK

A number of researchers have studied fault tolerant scheduling in which PB model has been investigated wherein every job has Primary and Backup which executes on unique computing machines to attain fault tolerance [3]. A large number of researches were carried out to come up with effectual scheduling algorithm while considering fault tolerance in conventional distributed systems like clusters and grids [4],[5],[6],[7],[8],[9]. In [10] the author proposed an adaptable scheduling algorithm deployed considering virtual machine combination which retain energy. In [11] the author proposed HARMONY, heterogeneity concerned resources allocation and job scheduling algorithm. The referred literature does not examine fault tolerance at the time of finding optimal solution wherever fault tolerance is examined; virtualization has not been taken into consideration. According to our perception very less researchers examined elastic resource assignment that follows fault tolerance. In [1] the authors proposed a novel algorithm FESTAL investigated elastic resource assignment that considers fault tolerance to perfectly allocate the resources. The algorithm examines and fixes the problem of reliable, elastic and scheduling in virtual clouds. The algorithm can withstand single host crashes. Therefore, we have designed IFTESA that is able to handle two Host crashes concurrently.

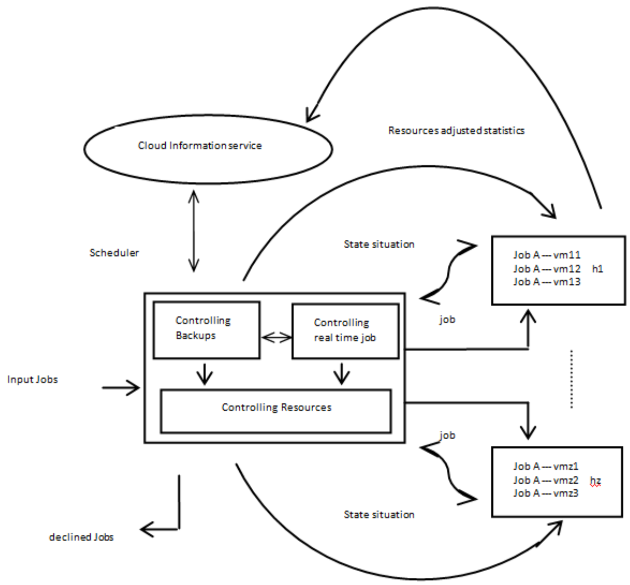

Figure 1: Scheduler Design

3. RESEARCH GAPS

Scheduling design is given in Figure 1. The scheduler contains Controlling Backups, Controlling real time job and controlling resources. The scheduler assigns all the jobs to the virtual machine on the hosts and continuously monitoring the state situation information which is directly provided by the hosts to the scheduler. The active hosts and virtual machines (which act as computing instances) status is registered with the cloud information service (CIS) which is referenced by the scheduler while assigning the job to the virtual machine. When new job enters the queue, the controlling backups controller creates backkup1 and backup2. The controlling backups supplies triplicates of the job to the controlling real time job which investigates by getting information from the CIS that if three copies of the job can finish within specified time. If not the controlling real time job intimates the controlling resources to add more number of resources. If the schedule is not available to meet the timing constraint requirement of the job even if new resources are added, the job is dismissed. The controlling resources controller in consultation with CIS supervises the resource status. When less number of jobs are in execution and some of the virtual machines are idle for longer time, controlling resources controller determines if few virtual machines can be dropped to boost the resource usage. If the primary job completes execution without errors, the victory information is sent to the Controlling backups controller to cancel the execution of backup1 and backup2. Otherwise, when one host or two hosts fail, the fault finding procedure will mark the crash and inform the controlling backups controller. Now, the controlling backups controller will not intimate the virtual machine to stop the execution of the backup jobs. The backup executes successfully thus achieving fault tolerance.

4. FAULT TOLERANT MECHANISM

We have included the features of clouds in the PB model for fault tolerance. In conventional PB model every job is executed as two sub jobs, first one is Primary job executing one processor/ virtual machine and second is Backup job executing on another processor/ virtual machine while considering the fault tolerance. In order to allow multiple Host crashes we have made an attempt in which every submitted job will complete as three jobs, first is Primary job second is Backup1 Job and third is Backup2 Job which shall execute the job on three virtual machines thus achieving the fault tolerance. We introduce the Backup-Backup overlapping technique and Virtual Machine migration technique in clouds. We shall be investigating BB overlap constraint and Virtual Machine migrating constraint in cloud environment.

4.1 Backup-Backup Overlapping

In traditional distributed systems, when primary jobs are executing on two different Hosts, backups can overlap with each other. However, this situation becomes difficult in virtualized clouds environment. The Backup-Backup technique follows basic principle that primary job will not be scheduled on virtual machine on one Host.

Theorem 1: if Host(tk_i^p) = Host(tk_j^p) = Host(tk_k^p) , and Host(tk_i^{b1}) = Host(tk_j^{b1}) = Host(tk_k^{b1}) then tk_i^{b2} , tk_j^{b2} , tk_k^{b2} can not overlap, regardless of whether vm(tk_k^p) = vm(tk_j^p) = vm(tk_k^p) or not.

Contradictory Proof: Let the Host(tk_i^p)= Host(tk_j^p)= Host(tk_k^p) = h1, Host(tk_i^{b1}) = Host(tk_j^{b1}) = Host(tk_k^{b1}) = h2 and tk_i^{b2} , tk_j^{b2} , tk_k^{b2} overlaps. When h1 and h2 fails, all the virtual machines in h1 and h2 fails, thus crashes the jobs on these vm's including tk_i^p , tk_j^p , tk_k^p and tk_i^{b1} , tk_j^{b1} , tk_k^{b1} . Thus tk_i^{b2} , tk_j^{b2} , tk_k^{b2} must execute. Since tk_i^{b2} , tk_j^{b2} , tk_k^{b2} execute simultaneously in the same vm, conflicting time exists. Contradictory situation. Therefore tk_i^{b2} , tk_j^{b2} , tk_k^{b2} not allowed to overlap.

We assume that tk_j followed by tk_k are new jobs to execute and triplicates of job tk_i are executing namely, tk_i^p, tk_i^{b1} , tk_i^{b2}. In tk_i^b, different cases should be

ISSN 2229-5518

discussed respectively, that is st(tk_i^{b1}) = st(tk_i^{b2}) = passive and st(tk_i^{b1}) = st(tk_i^{b2}) = active.

Case 1: st(tk_i^{b1}) = st(tk_j^{b1}) = st(tk_k^{b1}) = passive, st(tk_i^{b2}) = st(tk_j^{b2}) = st(tk_k^{b2}) = passive and Host(tk_i^{p}) ≠ Host(tk_j^{p}) ≠ Host(tk_k^{p}) , vm(tk_i^{b2}) = vm(tk_j^{b2}) = vm(tk_k^{b2}).

According to the assumption, at most two Hosts may encounter a failure at one time instant, thus tk_i^{b2} , tk_j^{b2} , tk_k^{b2} can overlap.

Theorem 2: if st(tk_i^{b1}) = st(tk_j^{b1}) = passive, st(tk_k^{b1}) = active then tk_i^{b1} , tk_j^{b1} , tk_k^{b1} cannot overlap.

Contradictory Proof: Suppose tk_i^{b1} , tk_j^{b1} , tk_k^{b1} overlaps. If Host(tk_j^{p}), Host(tk_k^{p}) crashes, tk_j^{b1}, tk_k^{b1} are invoked. However, st(tk_j^{b1}) = passive, st(tk_k^{b1}) = active, both try to execute simultaneously in the same virtual machine. Contradiction happens. Thus tk_i^{b1} , tk_j^{b1} , tk_k^{b1} cannot overlap.

Case 2: st(tk_i^{b1}) = st(tk_j^{b1}) = st(tk_k^{b1}) = active, st(tk_i^{b2}) = st(tk_j^{b2}) = st(tk_k^{b2}) = active and Host(tk_i^{p}) ≠ Host(tk_j^{p}) ≠ Host(tk_k^{p}) , vm(tk_i^{b2}) = vm(tk_j^{b2}) = vm(tk_k^{b2}). tk_i^{b2} , tk_j^{b2} , tk_k^{b2} will be overlapping if given below condition is followed.

Theorem 3: Suppose job tk_j^{b} has Earliest Starting Time ErSTkl(tk_j^{b}). In vm vm_{kl}, vm(tk_i^{b})= vm(tk_j^{b}) = vm_{kl}, if st(tk_i^{b}) = active, st(tk_j^{b})=passive and tk_i^{b} $overlaps$ tk_j^{b}, then ErSTkl(tk_j^{b})≥$\int_i^p + €$, here € is execution cancel time of tk_i^{b} if tk_i^{p} finished without crashing.

Contradictory Proof: Let $s_j^b < \int_i^p$, and tk_i^{b} $overlaps$ tk_j^{b}, so tk_i^{by} overlaps tk_j^{b}. When Host (tk_j^{p}) crashes, tk_j^{b} is invoked. But tk_i^{by} is executing, and cannot be interrupted, a conflicting time happens into tk_i^{by} and tk_j^{b}. Hence Contradicts. Therefore ErSTkl(tk_j^{b}) must be later than $\int_i^p + €$.

4.2 Virtual Machine Migration

Theorem 4: Suppose $\int H_{kl}$ be the set of all Hosts. $\int H_{kl}$= { Host (tk_i^{p})‖¥tk_i€T, vm(tk_i^{b}) = v_{kl}} U { Host(tk_i^{b}) ‖¥tk_i€T, vm(tk_i^{p}) = v_{kl}}. v_{kl} will not be able to migrate to h_j, ¥h_j€ $\int H_{kl}$

Contradictory Proof: Suppose that vm_{kl} migrates to ht_j, ¥h_j€ $\int H_{kl}$. It means the primary job and the backup job executing on the same Host. Suppose the Host crashes, primary job and backup job will also crashes. Therefore, fault tolerance is not achieved. Hence Contradicts. Migrating VM not feasible.

Theorem 5: ¥tk_i€T, vm(tk_i^{p}) = vm_{kl}, if tk_i^{b} $overlaps$ tk_j^{b}, then vm_{kl} is not able to migrate to Host(tk_j^{p}).

Contradictory Proof: Let vm_{kl} migrates to Host (tk_j^{p}). Then tk_i^{p} and tk_j^{p} are executing on the same Host while tk_i^{b} overlaps tk_j^{b} , which is in contradiction to Theorem 1. Hence, migration is not feasible.

5. Improved Fault Tolerant Elastic Scheduling Algorithm for Cloud Computing (IFTESA)

IFTESA includes a resource provisioning mechanism which is elastic means can adjust dynamically the resource provisioning on the basis of resource request. It consists of a mechanism of scaling-up of resources and scaling-down of resources mechanism. When it is not possible to execute the duplicates of jobs on available resources, scaling-up of resources mechanism is executed to meet the resources demand. In the mean time, if some resource is not used for some time, when the system is operating, scaling-down of resources module will be executed to delete/ release the idle resources, thus improving the resource utilization. If job tk_i is not able to be executed on the available virtual machine, scaling-up resources module is called which will create a new virtual machine which will be added into the system. The new virtual machine's processing power satisfies the expression given below:

$$rd_{new(tk_i)+tks_i/pp_{new}+defer} < d_i \qquad (1)$$

where $rd_{new}(tk_i)$ indicates the time to ready job tk_i on new virtual machine, tks_i/pp_{new} shows the task size divided by processing power of the new machine which gives us the approx. execution time, defer indicates creating timing of new virtual machine, virtual machine migration and booting timing of new active Hosts, d_i is the deadline time.

ISSN 2229-5518

Algorithm 1: To schedule scalingUpResources() in IFTESA

1 New virtual machine is to be selected which will satisfies equation (1);
2 Active Hosts are to be sorted in decreasing order by the remaining MIPS (Million Instructions per second);
// NEW VIRTUAL MACHINE ON EXISTING ACTIVE HOST WILL BE ASSIGNED
3 for each Host htk in Ha (Active Hosts) do
4 if htk can accommodate the new virtual machine next
5 new virtual machine will be created on htk.
6 return new vm;
// SOME VIRTUAL MACHINES WILL BE MIGRATED AMONG ACTIVE HOSTS
7 for each Host in the set of active Hosts do
8 Virtual machine with minimal MIPS in hs is to be migrated to alternative hosts; fault-tolerating need must be followed in Theorem4 and Theorem5;
9 if the Host hs can accommodate new virtual machine next
10 Create the new virtual machine on Host hs;
11 return new virtual machine;
// SLEEP STATUS HOSTS WILL BE TURNED ON
12 Now, Turn on a Host htnew in H – Ha (Total set of Hosts minus the set of active Hosts);
13 if the MIPS of htnew are satisfied by the new virtual machine next
14 Create the new virtual machine on htnew;
15 return new virtual machine;
16 else
17 return nothing/ INVALID;

Algorithm 2: To schedule scalingDownResources() in IFTESA

1 for each virtual machine in vmkl in the cloud do
// REACHED THRESHHOLD
2 if it reached the time vmkl.TTcancel next
3 Remove vmkl from htk and cancel it;
//PROCESSING POWER UTILIZATION FALLS BELOW //Umin THEN //MIGRATE VM FROM THIS HOST
4 if htk.utilization ≤ Umin next
5 offTagg ←TRUE;
6 for each vmkl in htk do
7 migTagg← NOT TRUE;
8 for each hti in Hta except htk do
9 if hti could serve vmkl & the migrating requirement is met of fault-tolerance of Theorem4 and Theorem5 next
10 migTagg←TRUE;
11 stop;
//CANNOT MIGRATE
12 if migTagg==NOT TRUE next
13 offTagg←NOT TRUE;
14 stop;
//SWITCH OFF WHEN ALL MIGRATE
15 if offTagg next
16 virtual machine in htk is to be migrated to the designated Hosts;
17 Change htk to sleeping status, delete the host from the set of active Hosts Hta;
18 else
19 Stop the migration operation;

Algorithm 3: To schedule Primary Job in IFTESA

1 The set of all the Hosts in active status are sorted in ascending counting of primary jobs scheduled;
// FIND THE DESIGNATED HOSTS WITH FEWER PRIMARIES SCHEDULED
2 Htcand ←upper β% Hosts in Hta;
3 finding ←NOT TRUE; Earliest Finish Time (ErFT) ←+∞; vm ←INVALID;
4 while not every Host in active status (Hta) is examined do
// WHERE THE ERFT OF THE JOB IS SMALLEST IS CHOOSEN TO //EXECUTE ON PRIMARY
5 for each htk in Htcand do
6 for each vmkl in htk.vmListing do
7 ErFT of job primary 1 is calculated ErFTkl(tk_i^p);
8 if ErFTkl(tk_i^p) ≤ di (deadline time) next
9 finding ← TRUE;
10 if ErFTkl(tk_i^p) ≤ ErFT next
11 ErFT← ErFTkl(tk_i^p);
12 vm← vmkl;
13 if finding == NOT TRUE next
14 Htcand ←next upper β % Hosts in the set of active Hosts (Ha);
15 else
16 stop;

// VIRTUAL MACHINE IS ASSIGNED TO THE NEWLY ARRIVED JOB
17 if finding == NOT TRUE next
18 vm← scalingUpResources();
19 if finding == TRUE next
20 Assign tk_i^p to vm;
21 Update the TTcancel of vm;
22 else
23 Dismiss tk_i^p;

Algorithm 4: To schedule Backup Job in IFTESA

1 Htcand ← are the set of Hosts on which primary jobs are set to be executed;
2 Htprimary ←Sorting Hta - Htcand counting by executing primary in ascending order;

3 finding← NOT TRUE; vm← INVALID; T ←MAXIMUM; Latest Starting Time (LTST) ←0;
4 while not every Host in Htprimary is examined do
// ASSIGN BACKUP1 AND BACKUP2 ON THE VIRTUAL MACHINE //ON DIFFERENT HOSTS WITH MAXIMUM TTcancel AND THE //STARTING TIMING OF BACKUP IS LATEST AMONGEST THE VM
5 for each htk in Htcand do
6 for each vmkl in htk.VmListing do
7 Latest Starting Time is to be calculated LTSTkl(tk_i^b);
8 if LTSTkl(tk_i^b) + etkl (tk_i) ≤ di next
9 finding ← TRUE;
10 if vmkl.TTcancel < T ∥ vmkl.TTcancel == T & LTSTkl(tk_i^b) >LTST next
11T← vmkl.TTcancel;
12 LTST ← LTSTkl(tk_i^b);
13 vm← vmkl;
14 if finding == NOT TRUE next
15 Htcand ←next upper β % Hosts in Htprimary;
16 else
17 stop;
18 if finding == NOT TRUE next
19 vm← scalingUpResources();
// FIND 1 AND FIND 2 ARE TRUE ON TWO DIFFERENT HOSTS
20 if finding == TRUE next
21 Assign tk_i^b to vm;
22 Updating TTcancel of vm;
23 else
// DISMISS JOBS PRIMARY, BACKUP1, BACKUP2
24 Dismiss tk_i^p; Dismiss tk_i^b;

6. RESULTS & DISCUSSION

The analysis is based on three performance evaluation matrices:

1) GR - Guarantee Ratio
2) HAT – Hosts Active Time
3) RTH – Ratio of total execution of all the jobs/tasks over total time of hosts in active state

GR-It is defined as the number of jobs which are guaranteed to complete execution amid all the accepted jobs.
HAT-It is defined as total time of all hosts in active state indicating the resource usage of the system.
RTH-It is defined as total time taken by all the tasks to execute over total time of all hosts in active state indicating the resource usage of the system

In order to analyze the new algorithm IFTESA, we will check the performance on the following:

1) Effect of job count on performance
2) Effect of arrival rate on performance
3) Effect of deadline time on performance

CASE [1] - Effect of job count on performance
With increasing job count both algorithms preserve high Guarantee Ratio that can be ascribed to infinite number of resources available in the cloud.

With the increase in the job count, the new request can be satisfied by scaling up of resources (add new resources). In spite of the availability of infinite number of resources, there exist some jobs submitted to the cloud that are rejected. The reason is extra time required for creation of new virtual machine and turning on time of new host due to which jobs are not able to start in time thus deadlines are missed.

FESTAL is able to handle single host failures only i.e. if primary of a task fails, backup can always execute. If suppose the host on which backup is scheduled also fails the guarantee ratio decreases whereas IFTESA comes to the rescue in this situation as we have triplicates of the job i.e primary, backup1 and backup2. When both Primary and Backup1 fails we still have backup2 executing on a different computing instance thus increasing the overall guarantee ratio.

To ensure higher GR the IFTESA needs to keep virtual machines and hosts active for longer time interval as compared to FESTAL

Keeping in view the infinite number of resources in the cloud, it makes little impact for IFTESA as GR increase and same is the impact on RTH

CASE [2] - Effect of arrival rate on performance
When arrival time varies the GR of FESTAL will keep on increasing but GR or IFTESA will be more than FESTAL because of less number of rejected tasks and ability to handle concurrent crashes.

With the increase in the arrival rate IFTESA needs more number of active vm and hosts in order to ensure higher GR therefore again FESTAL will outperform IFTESA in terms of HAT.

RTH will be higher in IFTESA since more number of tasks will be active and executing thus less chances of failures, overall increasing the GR and consequently the RTH.

CASE [3] - Effect of deadline time on performance

When base Deadline is prolonged, GR increases. This increase will be more in IFTESA. Compared to time of arrival of job and count of jobs, the effect of deadline in GR is much greater because strict deadline timing requirement makes it meaningless to add new resources, reason boot time is extra. If deadline time is flexible then all jobs are accepted and executed in IFTESA and FESTAL. Again if more than one host crashes the GR of IFTESA will increase.

When base Deadline is increased, HAT is likely to first increase and then may keep stable values. The reason is that deadline changes from strict to flexible, more jobs are accepted by the system and thus more hosts keep active to accommodate these tasks. When almost all the tasks are accepted by the system, HAT keeps stable values but even then IFTESA will require more number of resources as each job will execute as triplicates. Hence FESTAL outperform in terms of HAT. Even due to prolonged base Deadline FESTAL performs better but in case of two concurrent crashes IFTESA outperform in terms of RTH. The following are the observations found:

a) IFTESA has higher GR as compared to FESTAL
b) IFTESA needs higher number of resources to maintain higher GR
c) FESTAL uses less number of resources so performs better in terms of HAT
d) IFTESA has higher GR so higher RTH as compared to FESTAL

On average IFTESA outperforms FESTAL

7. Conclusions and Future Scope

We investigated Fault Tolerance, a long standing problem in cloud computing by extending PB model to include cloud features. Meeting Fault Tolerance in Cloud Computing is a challenging job as virtual machines are the computing instances rather than Hosts. Prevailing fault tolerant scheduling algorithms which studied PB model approach in cloud computing investigated single Host crashes. The proposed algorithm, improved fault tolerant elastic scheduling algorithm identified as IFTESA is able to handle two Host crashes concurrently, IFTESA attempts to boost performance by involving virtual machine migration and overlapping technique. The constraints have been investigated to full fill fault tolerance needs. Future effort will be 1) To test the proposed algorithm using simulator. 2) Propose a generalized algorithm which can handle multiple host crashes. 3) To consider communication time and other features of hosts for further improvement the system. 4) IFTESA can be implemented in a real cloud environment to further test its performance.

ACKNOWLEDGMENT

The authors thank the anonymous referees for their suggestions to improve this paper

REFERENCES

[1] Ji Wang, Weidong Bao, Xiaomin Zhu, Member, IEEE, Laurence T. Yang, Senior Member, IEEE, and Yang Xiang, Senior Member, IEEE “FESTAL: Fault-Tolerant Elastic Scheduling Algorithm for Real-Time Tasks in Virtualized Clouds” IEEE TRANSACTIONS ON COMPUTERS, VOL. 64, NO. 9, pp. 2545–2558 SEPTEMBER 2015

[2] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, “CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” Softw.: Pract. Exp., vol. 41, no. 1, pp. 23–50, 2011.

[3] S. Ghosh, R. Melhem, and D. Moss_e, “Fault-tolerance through scheduling of aperiodic tasks in hard real-time multiprocessor systems,” IEEE Trans. Parallel and Distrib. Syst., vol. 8, no. 3, pp. 272–284, Mar. 1997.

[4] X. Qin and H. Jiang, “A novel fault-tolerant scheduling algorithm for precedence constrained tasks in real-time heterogeneous systems,” Parallel Comput., vol. 32, no. 5, pp. 331–356, 2006.

[5] Q. Zheng, B. Veeravalli, and C.-K. Tham, “On the design of faulttolerant scheduling strategies using primary-backup approach for computational grids with low replication costs,” IEEE Trans. Comput., vol. 58, no. 3, pp. 380–393, Mar. 2009.

[6] J. Balasangameshwara and N. Raju, “Performance-driven load balancing with a primary-backup approach for computational grids with low communication cost and replication cost,” IEEE Trans. Comput., vol. 62, no. 5, pp. 990–1003, May 2013.

[7] R. Al-Omari, A. K. Somani, and G. Manimaran, “Efficient overloading techniques for primary-backup scheduling in real-time systems,” J. Parallel Distrib. Comput., vol. 64, no. 5, pp. 629–648, 2004.

[8] G. Manimaran and C. S. R. Murthy, “A fault-tolerant dynamic scheduling algorithm for multiprocessor real-time systems and its analysis,” IEEE Trans. Parallel Distrib. Syst., vol. 9, no. 11, pp. 1137–1152, Nov. 1998.

[9] X. Zhu, X. Qin, and M. Qiu, “QoS-aware fault-tolerant scheduling for real-time tasks on heterogeneous clusters,” IEEE Trans. Comput., vol. 60, no. 6, pp. 800–812, Jun. 2011.

[10] P. Graubner, M. Schmidt, and B. Freisleben, “Energy-efficient management of virtual machines in eucalyptus,” in Proc. IEEE Int.Conf. Cloud Comput., 2011, pp. 243–250.

[11] Q. Zhang, M. F. Zhani, R. Boutaba, and J. L. Hellerstein, “Harmony: Dynamic heterogeneity-aware resource provisioning in the cloud,” in Proc. IEEE Int. Conf. Distrib. Comput. Syst., 2013, pp. 511–519.

[12] Raman Kumar and Harsh Kumar Verma, "A Channel Analysis in Network Security - Using a Covert Approach", International Journal on Information Sciences and Computing, IJSAC, Vol. 2, Issue 1, 2008.

[13] Raman Kumar, "A New Break-Flow Technique for Encryption and Decryption of JPG/BMP/DIB Images", International Journal of Secure Digital Information Age, IJSDIA, Vol. I No. II, 2009.

[14] Harsh Kumar Verma, Abhishek Narain Singh and Raman Kumar, "Robustness of the Digital Watermarking Techniques against Brightness and Rotation Attack", International Journal of Computer Science and Information Security, IJCSIS, Vol. 5, No. 1, pp. 36-40, 2009.

[15] Harsh Kumar Verma, Ramanpreet Kaur and Raman Kumar, "Random of the Audio Watermarking Algorithm for Compressed Wave Files", International Journal of Computer Science and Network Security, IJCSNS", Vol. 9, No. 11, pp. 283-287, 2009.

[16] Raman Kumar and Harsh Kumar Verma, "Secure Threshold Proxy Signature Scheme Based on RSA for Known Signers", Journal of Information Assurance and Security, JIAS, Vol. 5, Issue 1,pp. 319-326, 2010.

[17] Raman Kumar, Asmita Puri, Vivek Gautam and Saumya Singla, "An Image Based Authentication System - Using Needham Schroeder Protocol", International Journal of Computer Science and Network Security, IJCSNS, Vol. 10, No. 11, pp. 135-140, November 2010.

[18] Raman Kumar, Saumya Singla, Sagar Bhalla and Harshit Arora, "Optimization of work flow execution in ETL using Secure Genetic Algorithm", International Journal of Computer Science and Information Security, IJCSIS, Vol. 8, No. 8, pp. 213-222, November 2010.\

[19] Raman Kumar, Roma Jindal, Abhinav Gupta, Sagar Bhalla and Harshit Arora, "A Secure Authentication System - Using Enhanced One Time Pad Technique", International Journal of Computer Science and Network Security, IJCSNS, Vol. 11, No. 2, pp. 11-17, February 2011.

[20] Raman Kumar, Naman Kumar and Neetika Vats, "Protecting the Content - Using Digest Authentication System and Other Techniques", International Journal of Computer Science and Information Security, IJCSIS, Vol. 9, No. 8, pp. 76-85, August 2011.

[21] Raman Kumar and Rajesh Kochher, "Information Technology Empowers by Women", International Journal of Science and Emerging Trends with latest Trends, IJSETT, Vol. 1, No. 1, pp. 1-5, June 2011.

[22] Raman Kumar, Harsh Kumar Verma and Renu Dhir, "Security Analysis and Performance Evaluation of Enhanced Threshold Proxy Signature Scheme Based on RSA for Known Signers", International Journal of Computer Network and Information Security, IJCNIS, Hong Kong, Vol. 4, No. 9, pp. 63-76, 2012.

[23] Jaskanwal Minhas and Raman Kumar, "An analysis on Blocking of SQL Injection Attacks by Comparing Static and Dynamic Queries", International Journal of Science and Emerging Trends with latest Trends, IJSETT, Vol. 3, No. 1, pp. 46-53, 2012.

[24] Jaskanwal Minhas and Raman Kumar, "Blocking of SQL Injection Attacks by Comparing Static and Dynamic Queries", International Journal of Computer Network and Information Security, IJCNIS, Hong Kong, Vol. 5, No. 2, pp. 1-9, 2013.

[25] Anil Gupta and Raman Kumar, "Design and Analysis of an algorithm for Image Deblurring using Bilateral Filer", International Journal of Science and Emerging Trends with latest Trends, IJSETT, Vol. 5, No. 1, pp. 28-34, 2013.

[26] Raman Kumar, Harsh Kumar Verma and Renu Dhir, "Cryptanalysis and Performance Evaluation of Enhanced Threshold Proxy Signature Scheme Based on RSA for Known Signers", Mathematical Problems in Engineering, Vol. 2013, Article ID 790257, 24 pages, 2013.

[27] Ashwani Malhotra and Raman Kumar, "The Comparison of Various Multi Proxy Multi Signature Schemes with Shared Verification", International Journal of Computer Science and Technology, IJCST, Vol. 4, Issue 1, Ver. 4, pp. 633-637, April 4, 2013.

[28] Raman Kumar and Anil Gupta, "Comparative Study of Various Image Restoration Techniques in the Basis of Image Quality Assessment Parameters", International Journal of Computer Science and Technology, IJCST, Vol. 4, Issue 2, Ver. 2, pp. 274-279, April - June, 2013.

[29] Anshu Sharma and Raman Kumar, "The obligatory of an Algorithm for Matching and Predicting Crime - Using Data Mining Techniques", International Journal of Computer Science and Technology, IJCST, Vol. 4, Issue 2, Ver. 2, pp. 289-292, April - June, 2013.

[30] Ashwani Malhotra and Raman Kumar, "Performance and Security Analysis of Improved Hsu et. al's Multi Proxy Multi Signature Scheme With Shared Verification", International Journal of Computer Science and Technology, IJCST, Vol. 4, Issue 2, Ver. 4, pp. 659-664, April - June, 2013.

[31] Anshu Sharma and Raman Kumar, "Analysis and Design of an Algorithm Using Data Mining Techniques for Matching and Predicting Crime", International Journal of Computer Science and Technology, IJCST, Vol. 4, Issue 2, Ver. 4, pp. 670-674, April-June, 2013.

[32] Navneet Kaur and Raman Kumar, "Review of Restorable Routing Algorithm in Optical Networks", International Journal of Computer Applications - IJCA (0975-8887), Vol. 83, No. 13, pp. 23-27, December 2013.

[33] Raman Kumar, "Security Analysis and Implementation of an Improved CCH2 Proxy Multi-Signature Scheme", International Journal of Computer Network and Information Security, IJCNIS, Hong Kong, Vol. 6, No. 4, pp. 46-54, March 2014.

[34] Raman Kumar and Nonika Singla, "Cryptanalytic Performance Appraisal of Improved CCH2 Proxy Multisignature Scheme", Mathematical Problems in Engineering, Vol. 2014, Article ID 429271, 13 pages, 2014.

[35] Navneet Kaur and Raman Kumar, "Design of Restorable Routing Algorithm in Optical Networks", International Journal of Computer Applications - IJCA (0975-8887), Vol. 91, No.2, pp. 39-45, April 2014.

[36] Divya Jyoti and Raman Kumar, "Review of Security Analysis and Performance Evaluation of an Enhanced Two-Factor Authenticated Scheme", International Journal of Electrical, Electronics and Computer Engineering, Vol. 3, No.1, pp. 218-222, June 2014.

[37] Divya Jyoti and Raman Kumar, "Security Analysis and Performance Evaluation of an Enhanced Two-Factor Authenticated Scheme", International Journal of Computer Applications - IJCA (0975-8887), Vol. 101, No.8, pp. 28-33, September 2014.

[38] Raman Kumar, Harsh Kumar Verma and Renu Dhir, "Analysis and Design of Protocol for Enhanced Threshold Proxy Signature Scheme Based on RSA for Known Signers", Wireless Personal Communications – An International Journal - Springer ISSN: 0929-6212 (Print) 1572-834X (Online), Volume 80, Issue 3 (2015), Page 1281-1345.

[39] Raman Kumar and Manjinder Singh Kahlon, "Power, Bandwidth and Distance Optimization in WMAN - Using Cross Layer Approach", International Journal of Computer Science and Technology, IJCST, Vol. 5, Issue 4, Ver. 4, pp. 144-147, Oct-Dec, 2014.

[40] Navneet Kaur, Manjinder Singh Kahlon and Raman Kumar, "AN IMPROVED DETERMINISTIC HEER PROTOCOL FOR HETEROGENEOUS REACTIVE WSNS – USING NETWORK LIFETIME AND STABILITY PERIOD", International Journal of Scientific & Engineering Research, IJSER, Vol. 5, Issue 9, pp. 650-655, September 2014.

[41] Raman Kumar, Arjun Sahni and Divanshu Marwah , "Real Time Big Data Analytics Dependence on Network Monitoring

Solutions using Tensor Networks and its Decomposition", Network and Complex Systems International Institute for Science Technology and Education-International Knowledge Sharing Platform: ISSN (Paper)2224-610X ISSN (Online)2225-0603, Vol. 5, No. 2, pp. 30-38, 2015.

[42] Harsh Kumar Verma, Kamalpreet Kaur and Raman Kumar, "A Comparison of Threshold Proxy Signature Scheme", The 2008 International Conference on Security and Management, USA, 14-17 July, 2008.

[43] Raman Kumar, "Electronic Information Security Paradigm: Using Security Engineering Strategies and Cryptography Techniques", International Conference on Computing Updates and Trends, JIM, India, 6 February, 2009.

[44] Raman Kumar and Harsh Kumar Verma, "An Advanced Secure (t, n) Threshold Proxy Signature Scheme Based on RSA Cryptosystem for known Signers", IEEE 2nd International Advance Computing Conference, TU, India, 19-20 February 2010.

[45] Raman Kumar and Rajesh Kochher, "Motivational Aspects of Information Security Metrics in an Organization", International Conference on Industrial Competitiveness, GGI, India, 10 April, 2010.

[46] Raman Kumar, "Organization of CHORD Peer-To-Peer Overlay: Using Self Organizing Approach", International Conference on Industrial Competitiveness, GGI, India, 10 April, 2010.

[47] Raman Kumar, "Ethical Issues for Scientists and Engineers: Using Cyber Laws and Information Security Techniques", International Conference on Challenges of Globalization & Strategy for Competitiveness, ICCGC, India, ISBN 978-81-920451-2-2, 287-294, 14-15 January 2011.

[48] Raman Kumar, Manjinder Singh Kahlon and Amit Kalra, "A Secure Authentication System- Using Enhanced Serpent Technique", AICTE and DST Government of India sponsored International Conference on Advancements in Computing and Communication, ICACC, BBSBEC, Fatehgarh Sahib, India, 409-415, 23-25, February 2012.

[49] Raman Kumar and Manjinder Singh Kahlon, "Ethical Issues for Scientists and Engineers", AICTE and DST Government of India sponsored International Conference on Advancements in Computing and Communication, ICACC, BBSBEC, Fatehgarh Sahib, India, ISBN 978-81-920451, 422-426, 23-25 February, 2012.

[50] Raman Kumar and Nonika Singla, "Cryptanalysis of Improved CCH2 Proxy Multi-Signature Scheme", 7th International Conference on Advanced Computing and Communication Technologies, ICACCT 2013, APIITD SD, India, ISBN 978-93-83083-38-1, 267-275, 16 November, 2013.

[51] Raman Kumar and Navneet Kaur , "Restorable Routing Algorithm in Optical Networks by Reducing Blocking Probability", IEEE Recent Advances in Engineering and Computational Sciences, RAECS 2014, UIET, PU, Chandigarh, India, 6-8 March, 2014.

[52] Raman Kumar, Arjun Sahni and Divanshu Marwah , "Real Time Big Data Analytics Dependence on Network Monitoring Solution using Tensor Networks and its Decomposition", International Conference on Modelling, Simulation and Optimizing Techniques, ICMSOT-2015, DAV College, Jalandhar, India, 12-14 February, 2015

[53] .Raman Kumar, "Power Conservation in Wireless Networks: Cross Layered Approach", Magazine, GNDU, India, 2007.

[54] Raman Kumar and Harsh Kumar Verma, "The Wireless Application Software Platform", AICTE sponsored National Seminar on “Enterprise Information Systems”, Apeejay, India, 24 May, 2008.

[55] Raman Kumar, "An improved Method of Generating a Threshold Proxy Signature Scheme on a Confidential Message", National Conference, 12th PSC, PAU Ludhiana, India, 7-9 February 2009.

[56] Raman Kumar, "Security Management Issues in Automatic Password Protection Mechanism", National Conference, 12th PSC, PAU Ludhiana, India, 7-9 February 2009.

[57] Raman Kumar and Arvind Kumar, "A New Proposed Model for Secure E-Business Intelligence System", National Conference on Modern Management Practices and Information Technology, DAVIET, Jalandhar, India, 17-18 April, 2009.

[58] Raman Kumar, Harsh Kumar Verma, Renu Dhir and Pankaj Yadav, "Information Security Vulnerability Issues in Intrusion Detection Systems", National Conference on Education & Research, JUIT, M.P., India, 6-7 March 2010.

[59] Raman Kumar, "The Lifecycle of Secure Application Development in a Business Environment", National Conference on Business Innovation, Apeejay, India, 27 February 2010.

[60] Raman Kumar, "Cyber Laws: Ethical Issues for Scientists and Engineers", Magazine, DAVIET, India, 2010.

[61] Raman Kumar and Amit Kalra, "India: Needs to Focus on the Security Engineering Strategies and Cryptographic Techniques", National Conference on “Innovations: A way to keep ahead, JIM, 17 April, 2010.

[62] Raman Kumar and Amit Kalra, "Role of Dynamic Learning in Establishment of Efficient Education System", National Conference, KC Institutions, Nawanshahr, India, 20 April, 2010.

[63] Raman Kumar and Rajesh Kochher, "Empowerment of women through Information Technology", National Conference on Role of Women in I.T - Today and Tomorrow, SD, India, 12 October, 2010.

[64] Raman Kumar and Amit Kalra, "A Survey on Sustainability Performance of Technology Sector: Using Green Engineering and Technology", National Seminar on Green Technology: Opportunities and Challenges, KC Institutions, Nawanshahr, India, 8-9 April, 2011

[65] Sukhbir Singh and Raman Kumar, “Improved Fault Tolerant Elastic Scheduling Algorithm (IFESTAL),” International Journal of Scientific and Engineering Research (IJSER), Accepted, December 2015.