# ENHANCED APPROACH TO OPTIMIZED CLOUD RELIABILITY BY USING REDUNDANCY HANDLING AND OPTIMIZED VM SELECTION POLICY

**[1]Bharti Bedi, [2]Gaurav Kumar**

[1]Research Scholar, [2]Assistant Professor
Computer Science & Engineering,
G.I.M.E.T, Amritsar, India

***Abstract:*** **The developing reliance of clients via web-based networking media, media transmission administrations, portable applications and other cloud services requires an arrangement that mitigates inescapable failures and guarantees the dependably on access to these services. So there must be a way to handle these failures during migration and adoption of cloud without any redundancy. This paper proposes migration way to deal with managing benefit conveyance upon a sudden failure, a virtual machine (VM) over-burden, or support. It builds up a combined redundancy handling FTC programming model that limits the movement downtime that is based on VM memory pages and the reliable situation of the VM. Experimental results shows that proposed system is much reliable than existing systems. The results are calculated in terms of make span, energy consumption and fault tolerance rate during VM migration resource allocation.**

***Index Terms*****: VM migration, pre copy, post copy, fault tolerance, redundancy handling**

## I. INTRODUCTION

Cloud computing provides a long term benefit of computing as a utility that has started new era involving distributed computing. It transformed the IT industry completely and reshaped the purchase of IT Hardware and software[1]. Resources such as networks, Servers, storage, application and data can be accessed with great ease using Cloud computing[2]. Virtualized environment of cloud allow user to access resources as if ownership is held by the user. Services are delivered to the user on pay per use basis. Services are provided using high speed internet labeled as X as a Service. X is partition into segments as application, platform and infrastructure. The aim of these segments is to provide more reliable service transparently at ever cheaper price[3].

### 1.1 Service Models of Cloud Computing

The various service models that cloud providers offered are mainly categorized in three categories[4, 5] :-

- **Software as a Service (SaaS)**
  This model provides complete application to the user on demand. It can handle multiple users at a time but in background only single application of the service is executed. The single application facilitate the users, they need not to go for any advance payment [6]. Various service provider like Google, sales force , Microsoft provides SaaS facility.
- **Platform as a Service(PaaS)**
  This model provides environment as a service to user for constructing own application that run on service provider base. It also provides predefined application server combined with OS to user [7]. Google's App Engine, Force.com is providing a platform to users.
- **Infrastructure as a Services(IaaS)**
  This model provides services to fundamental storage and computing capability. In order to manage workload the shared resources are utilized among various users [8]. To use the services the user has install own software over the infrastructure. Amazon and Go Grid are some example of IaaS.

Mobile cloud computing used heavily and hence faults required to tackled in this critical area of cloud computing.

### 1.2 Live VM Migration

Live VM migration is optimization strategy in which load from current virtual machine is migrated to healthier machine without shutting down any machine [9]. Both the machine are working during migration hence the name live VM migration. Several techniques are available to be used under Live VM Migration. These techniques are as discussed as under:-

**Pre Copy**

In a Pre copy approach the pages from source machine is copied to destination machine iteratively. The iterative nature of the pre-copy approach is due to dirty pages. This approach is useful in a situation where modification to the pages is limited. In case of heavy changes in pages this approach considerably slow down the migration process hence downtime and migration time is substantially increases.

**Post Copy**

In post copy approach first of all the processing or execution is transferred and then the memory. This means that execution immediately starts after memory resources are transferred to destination machine. Downtime is considerable reduced in this case. Migration time depends upon length of memory resources along with number of CPU required to execute the job.

**Hybrid Approach**

This approach is optimal among pre and post copy approach. This approach uses optimal features of pre and post copy. This approach contains processor state information along with lots of useful information about state. This information is termed as working set.

### 1.3 Energy Efficiency within Cloud

Energy consumption within the cloud is an issue required to be tackled. The primary reason for high degree of energy consumption is increased demand from cloud regular users that enhance load on cloud datacenters. Energy consumption reduces the resource utilization and decreases the reliability as degradation of machines takes place. To tackle this issue [10] proposed energy efficient scheduling within cloud. The scheduling is done to build a schedule that can optimize the execution of jobs on the resources provided by making jobs execute on maximum powered resources first. [11] proposed a smart and energy aware mechanism for large scale applications. Advance prediction scheme for predicting the energy consumption during the job execution is built. The reliability of advance prediction is questionable. [12] Suggested a batch formation mechanism for forecasting energy consumption in cloud based system. The web based workloads are assigned to datacenters for determining energy consumption. The information about the energy consumption is stored within the stable storage. This information about the energy consumption is used in the future workload assignment on the datacenters within cloud based systems. [13] Proposed a virtualization mechanism for energy conservation. VM placement strategy is considered to optimization energy consumption. The VM list is sorted before assignment of cloudlets. The VM with the highest MIPS value is selected for allocation of jobs hence jobs is completed well within time thus conserving energy.

### 1.4 Fault Tolerance Strategies

Fault tolerance strategies are critical for the performance enchantment of cloud. Cloud computing popularity is enhanced and the regular cloud users' demands and expectations from cloud is also enhanced. As the failure occurs, the popularity of cloud degrades.[14] Fault tolerance strategies thus play a critical part in ensuring the cloud performance to be stable if not enhanced. **The fault tolerance is divided into following categories:-**

**Proactive Fault tolerance**

The proactive fault tolerance mechanism is proposed by [15] in high performance cloud. Proactive fault tolerance is the mechanism in which faults and errors are avoided by replacing the component which is suspicious. Some of the techniques under proactive fault tolerance include preemptive migration, load balancing and software rejuvenation.

**Reactive Fault tolerance**

Reactive fault tolerance mechanism is proposed by [16, 17] with post copy migration for handling availability issue. In reactive fault tolerance action is taken after the failure occurs. Techniques under reactive fault tolerance include check pointing, replay and restart.

## II. Related Work

The proposed High Performance Computing (HPC) that focuses development of computing resources to give all the more figuring energy to their clients, we watch a desire for the union between Cloud Computing (CC) and HPC stages, with the business would like to see CC foundations to in the long run supplant in-house offices [18].

In the event that we prohibit the execution perspective where numerous past reviews highlight a non-insignificant overhead incited by the virtualization layer at the heart of each Cloud middleware when running a HPC workload, the subject of the genuine cost effectiveness is regularly left aside with the instinct that, most presumably, the cases offered by the Cloud suppliers are focused from a cost perspective [3, 19].

With explosive development of cell phones including PDAs, PDAs, and tablet PCs and the applications introduced in them, the portable Internet will keep up the improvement development slant as 4G correspondence system is broadly elevated to our lives [20]. What clients of the cell phones and applications need is that portable Internet can give them the administration which is easy to understand, high speed, and consistent. Dynamic Resource allocation proposed by is distributed computing. It enables business clients to scale here and there their asset use in light of requirements. Large portions of the touted picks up in the cloud show originated from asset multiplexing through virtualization innovation.

In this paper, we display a framework that utilizations virtualization innovation to distribute server farm assets progressively in view of use requests and bolster green processing by upgrading the quantity of servers being used [22].

In [23, 24] proposes the idea of "skewness" to quantify the unevenness in the multidimensional asset use of a server. Mostly this scheme of thing is effective in detecting attacks. By limiting skewness, we can join distinctive sorts of workloads pleasantly and enhance the general usage of server assets. We build up an arrangement of heuristics that anticipate over-burden in the framework

successfully while sparing vitality utilized. Follow driven recreation and test comes about show that considered calculation accomplishes great execution.

Distributed computing as proposed by [26] is efficient in order for the jobs to execute efficiently in Cloud based system. Distributed computing offers utility-arranged IT administrations to clients around the world. In light of a compensation as-you-go display, it empowers facilitating of unavoidable applications from shopper, logical, and business spaces. Be that as it may, server farms facilitating Cloud applications devour colossal measures of electrical vitality, adding to high operational expenses and carbon impressions to the earth. Subsequently, we require Green Cloud figuring arrangements that can limit operational expenses as well as decrease the natural effect. In this paper, we characterize an engineering system and standards for vitality proficient Cloud registering. In view of this engineering, we show our vision, open research difficulties, and asset provisioning and allotment calculations for vitality proficient administration of Cloud figuring situations.

Effective Grid computing as suggested by [27] is used for job allocation in advanced computing system. Advanced computing involves legions of nodes or VMs of different configurations, Heterogeneous environment form grids. Nodes of same configuration are grouped into same batch known as cluster.

## III. PROPOSED METHODOLOGY

Proposed approach uses allocation of resources parallel with backtracking mechanism in place for allocating resources by finding out spare virtual machines with sufficient resources. This section describes proposed system with parameters evaluation

### 3.1 Flowchart of Proposed Work:

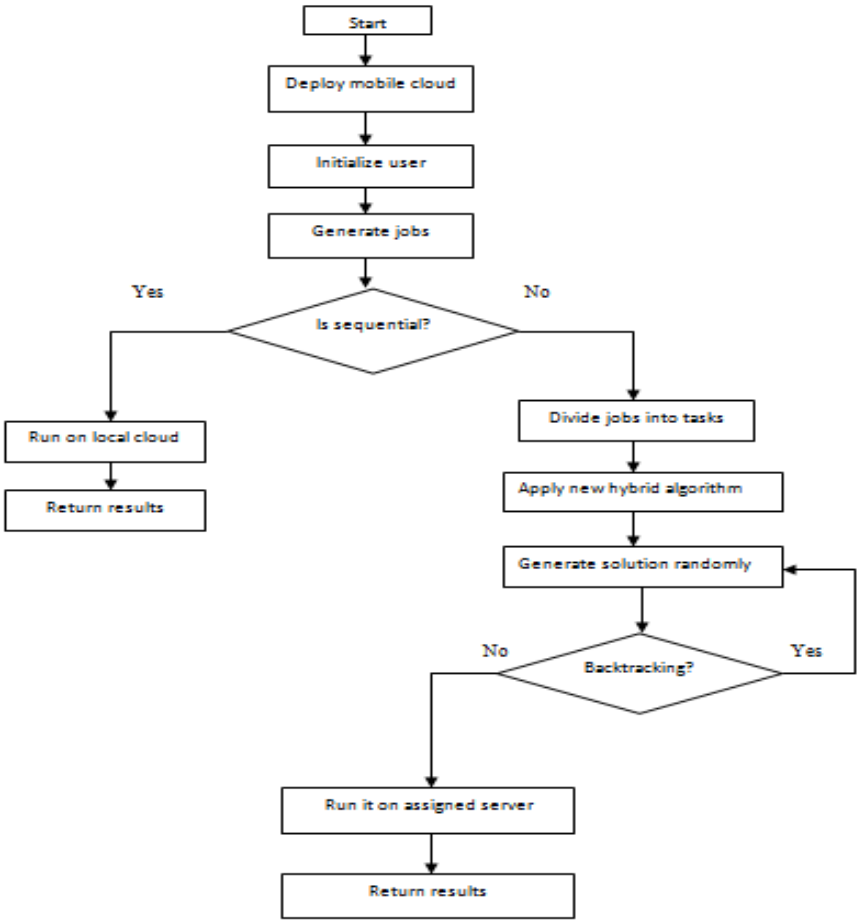


**Figure 1: Proposed methodology**

### 3.2 Proposed Algorithm

**Algorithm for REDUNDANCY HANDLING FTC is as under:-**

1. Deploy mobile cloud
   - Initialize Data centres, and allocate configuration to VMs
   - Arrange VMs with maximum power first and least power at last place.
2. Initialize Users
   - Assign users of mobile cloud with username and password

- Validate the users according to authentication
- If validated move to 3 step otherwise perform 2 step again

3. Load jobs externally
   - Jobs are loaded from external file containing processes and burst time
4. Divide jobs into tasks.
   - $\text{Tasks}_i = \text{Jobs}_i/\text{VM}_{\text{capacity}}$
5. If available(resources)
   - Execute job sequentially selecting VMs according to resource requirement and obtain make span

   Else
   - Execute jobs parallel and obtain make span

   End of if
6. If(optimal(Make span))
   Result=Make span
   Else
   Perform Backtracking
   End of if
7. Obtain result in terms of Make span

### 3.3 Phases of Redundancy Handling FTC

There are following phases associated with proposed literature:-

#### 3.3.1 Initialization

The jobs are partitioned into tasks. These tasks are referred to as[28, 30] chromosomes. These chromosomes are in hunt of resources acting as food for chromosomes. Initially these chromosomes are set according to order presented through cloudlets. These cloudlets considered for proposed operation includes jobs with burst time and arrival time.

$$\text{Chromosomes} = \text{Tasks}(t_1, t_2, ----, t_n) \quad (1)$$

Equation 1: Chromosomes in terms of tasks

#### 3.3.2 Objective Function

The objective function consists of Make span. The BSA has the implication in reducing Make span. The existing gives Make span which serves as baseline or threshold value. Main objective thus is to minimize the Make span. The schedule is executed again and again until Make span lesser than Make span obtained through RFTC is obtained.

$$\text{obj}_{\text{function}} = \min(\text{makspan}_{\text{BSA}}) \quad (2)$$

Equation 2: objective function

#### 3.3.3 Reliability Tackling Phase

As the initialization is complete, process of allocating resources to task begins. Make span is evaluated at each step. If the Make span is more than Make span obtained through RFTC then mutation with probability 0.5 is made. Allocation is done again and Make span is evaluated. This process continues until optimal schedule of allocation is accomplished [28, 31].

## IV. RESULTS AND DISCUSSION

Performance analysis is done by evaluating the existing system and comparing it with Redundancy Handling FTC. The performance analysis and results are describe in terms of the following :-

### 4.1 Parameter of Make Span

Table 4.1: Make Span

| Size | Existing (ms) | Proposed (ms) |
|---|---|---|
| 50 MB | 12 | 10.5 |
| 100MB | 18 | 15 |
| 150MB | 19.5 | 16.5 |

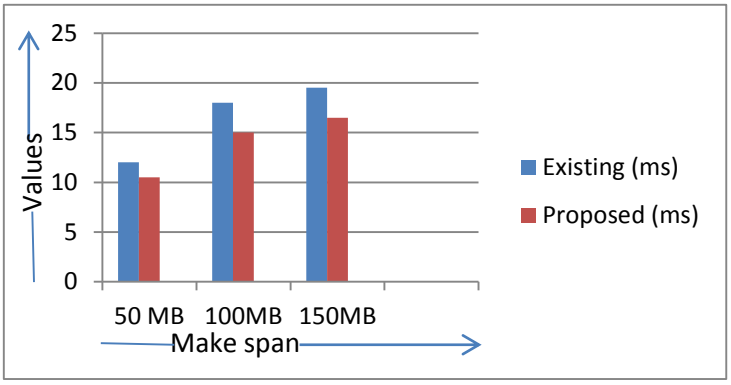


**Figure 2: Make Span of existing and proposed system**

### 4.2 Parameter of Fault Tolerant Rate

Table 4.2: Fault Tolerant Rate of Existing and Proposed System

| Size | Existing | Proposed |
|---|---|---|
| 50 MB | 0.2 | 0.6 |
| 100MB | 0.4 | 0.7 |
| 150MB | 0.5 | 0.88 |


**Figure 3: Fault Tolerant Rate of existing and proposed system**

### 4.3 Parameter of Energy Consumption

Table 4.3: Energy consumed by existing and proposed system

| Size | Existing (joules) | Proposed (joules) |
|---|---|---|
| 50 MB | 14 | 12 |
| 100MB | 16 | 14 |
| 150MB | 20 | 16 |

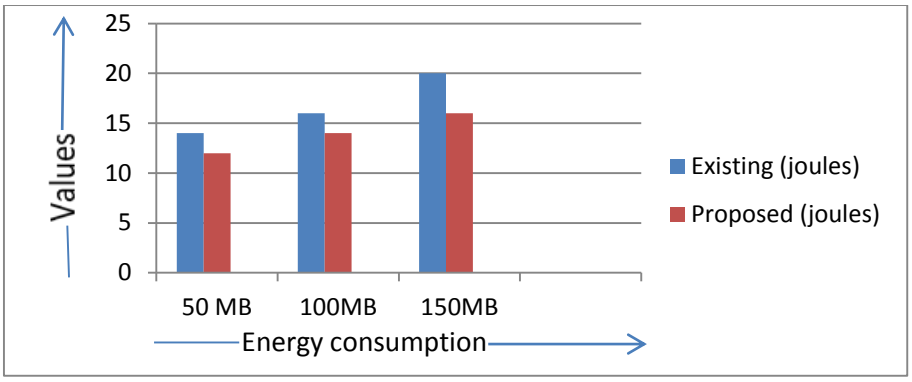


**Figure 4: Energy Consumption through existing and proposed system**

## V. CONCLUSION AND FUTURE SCOPE

The existing systems provide energy efficient approach for allocating jobs to virtual machines. However as resource decreases and requirement increases existing system fails. Individual resource searching criteria is required under such situation. For this purpose Redundancy FTC comes into picture which is hybridized with pre copy in proposed literature. Performance evaluation indicates significant improvement in results in terms of Make span, energy consumption and fault tolerant rate. The work can be further extended by using ant colony along with RFTC for enhancement in distinct parameters especially Make span.

## REFERENCES

**[1]** M. Armbrust et al., "A view of cloud computing," Commun. ACM, vol. 53, no. 4, p. 50, 2010.
**[2]** D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud : A Packet-level Simulator of Energy- aware Cloud Computing Data Centers," J. Supercomput., vol. 62, no. 3, pp. 1263–1283, 2012.
**[3]** Z. Xiao, W. Song, and Q. Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment," IEEE Trans. Parallel Distrib. Syst., vol. 24, no. 6, pp. 1107–1117, Jun. 2013.
**[4]** P. Zhang, S. Hu, J. He, Y. Zhang, G. Huang, and J. Zhang, "Building cloud-based healthcare data mining services," Proc. - 2016 IEEE Int. Conf. Serv. Comput. SCC 2016, pp. 459–466, 2016.
**[5]** T. H. Noor, Q. Z. Sheng, L. Yao, S. Dustdar, and A. H. H. Ngu, "CloudArmor: Supporting Reputation-Based Trust Management for Cloud Services," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 2, pp. 367–380, Feb. 2016.
**[6]** Z. Amin, "Review on Fault Tolerance Techniques in Cloud Computing," vol. 116, no. 18, pp. 11–17, 2015.
**[7]** D. Singh, J. Singh, and A. Chhabra, "High availability of clouds: Failover strategies for cloud computing using integrated checkpointing algorithms," Proc. - Int. Conf. Commun. Syst. Netw. Technol. CSNT 2012, pp. 698–703, 2012.
**[8]** B. Nicolae, "BlobCR : Efficient Checkpoint-Restart for HPC Applications on IaaS Clouds using Virtual Disk Image Snapshots."
**[9]** C. Clark et al., "Live Migration of Virtual Machines," no. Vmm, pp. 273–286.
**[10]** D. Kliazovich, P. Bouvry, and S. U. Khan, "DENS: Data center energy-efficient network-aware scheduling," Cluster Comput., vol. 16, no. 1, pp. 65–75, 2013.
**[11]** M. E. M. Diouri, O. Gl??ck, and L. Lef??vre, "Towards a novel smart and energy-aware service-oriented manager for extreme-scale applications," 2012 Int. Green Comput. Conf. IGCC 2012, 2012.
**[12]** J. Subirats and J. Guitart, "Assessing and forecasting energy efficiency on cloud computing platforms," Futur. Gener. Comput. Syst., 2014.
**[13]** N. Kord and H. Haghighi, "An Energy-Efficient Approach for Virtual Machine Placement in Cloud Based Data Centers," pp. 44–49, 2013.
**[14]** A. Tchana, L. Broto, and D. Hagimont, "Fault Tolerant Approaches in Cloud Computing Infrastructures," no. c, pp. 42–48, 2012.
**[15]** I. P. Egwutuoha, S. Cheny, D. Levy, B. Selic, and R. Calvo, "Energy efficient fault tolerance for high performance computing (HPC) in the cloud," IEEE Int. Conf. Cloud Comput. CLOUD, pp. 762–769, 2013.
**[16]** T. Hirofuchi, H. Nakada, S. Itoh, and S. Sekiguchi, "Reactive Consolidation of Virtual Machines Enabled by Postcopy Live Migration," pp. 11–18, 2011.
**[17]** P. Gupta and S. Banga, "Topic - Review of Cloud Computing in Fault Tolerant Environment With Efficient Energy Consumption," vol. 1, no. 4, pp. 251–254, 2013.
**[18]** L. A. Tawalbeh, "Mobile Cloud Computing Model and Big Data Analysis for Healthcare," vol. 3536, no. c, 2016.
**[19]** Z. Zhu, G. Zhang, M. Li, and X. Liu, "Evolutionary Multi-Objective Workflow Scheduling in Cloud," IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 5, pp. 1344–1357, May 2016.
**[20]** J. Emeras, S. Varrette, V. Plugaru, and P. Bouvry, "Amazon Elastic Compute Cloud ( EC2 ) vs . in-House HPC Platform : a Cost Analysis," vol. 7161, no. c, pp. 1–14, 2016.
**[21]** Meenakshi Sharma, Dr. Himanshu Aggarwal, "Mobile based application for prediction of diabetes mellitus: FHIR Standard" , *International Journal of Engineering and technology,vol.7,issue-2,e-ISSN:3347-3696 on 2018.*
**[22]** R. Deng, R. Lu, S. Member, and C. Lai, "Optimal Workload Allocation in Fog-Cloud Computing Towards Balanced Delay and Power Consumption," IEEE Access, vol. X, no. X, pp. 1–11, 2016.

**[23]** T. Scholl, R. Kuntschke, A. Reiser, and A. Kemper, "Community Training: Partitioning Schemes in Good Shape for Federated Data Grids," in Third IEEE International Conference on e-Science and Grid Computing (e-Science 2007), 2007, pp. 195–203.
**[24]** S. Moradi, "A distributed method based on mobile agent to detect Sybil attacks in wireless sensor networks," pp. 276–280, 2016.
**[25]** Meenakshi Sharma and Dr. Himanshu Aggarwal, "Methodologies of legacy clinical decision support system - A review" *International Conference on Recent Innovations in Computer Science and Information Technology*.2017 May.
**[26]** A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Futur. Gener. Comput. Syst., vol. 28, no. 5, pp. 755–768, 2012.
**[27]** M. B. Qureshi et al., "Survey on Grid Resource Allocation Mechanisms," J. Grid Comput., vol. 12, no. 2, pp. 399–441, 2014.
**[28]** J. B. Kaper, H. Lockman, M. M. Baldini, and M. M. Levine, "A Recombinant Live Oral Cholera Vaccine," Bio/Technology, vol. 2, no. 4, pp. 345–349, Apr. 1984.
**[29]** Meenakshi Sharma , Dr.Himanshu Aggarwal, "Evaluation factors for testing and validation of Clinical Reporting System" , *International Journal of Computer Science and Engineering,*vol.6,issue-2,e-ISSN:2347-2693 on 2018.
**[30]** S. D. Zahedani and G. Dastghaibyfard, "A hybrid batch job scheduling algorithm for grid environment," Proc. 4th Int. Conf. Comput. Knowl. Eng. ICCKE 2014, pp. 763–768, 2014.
**[31]** J. Xie, Y. Zhou, and H. Chen, "A novel bat algorithm based on differential operator and Levy flights trajectory," Comput. Intell. Neurosci., vol. 2013, 2013.