IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-6, Issue-3 E-ISSN: 2347-2693

Design of an Efficient Memory Management Model for Mobile Device Using C-Ram

Rajeev Kumar Bedi [1*], Vandana [2], Sunil Kumar Gupta [3]

[1*]Department of Computer Science and Engineering, Beant College of Engineering and Technology, Gurdaspur, India
[2]Department of Computer Science and Engineering, Beant College of Engineering and Technology, Gurdaspur, India
[3]Department of Computer Science and Engineering, Beant College of Engineering and Technology, Gurdaspur, India

**Corresponding Author: rajeevbedi12@gmail.com, Tel.:7696168094*

Available online at: www.ijcseonline.org

Received: 17/Feb//2018, Revised: 23/Feb2018, Accepted: 18/Mar/2018, Published: 30/Mar/2018

Abstract-Cloud computing demand is expanding because of which it is essential to redress benefits within the sight of faults too. The Resources in cloud computing can be progressively scaled that too in a manner that is cost effective. Adaptation to non-critical failure is the way toward discovering faults and failures in a system. Mobile gadgets are portrayed by constrained and non-adaptable assets, for example, low handling power, inadequate essential memory and short battery life. Asset concentrated applications, for example, increased reality, counterfeit consciousness, simulated vision, protest following, picture handling and common dialect preparing are getting to be famous. These applications cannot be executed adequately on asset requirement mobile gadgets. Over the most recent couple of years impressive research has been done on Mobile Cloud Computing to expand the abilities of mobile gadgets. Different models and structures have been proposed to offload asset escalated parts of uses to cloud for productive execution. The system should work appropriately even if there is chance of failure happens or hardware failure or programming failure. Failure ought to be overseen viably for dependable Cloud Computing. It will likewise guarantee accessibility and robustness. This paper depicts a profile based approach for energy efficient mobile cloud computing environment.

Keywords- Cloud Computing, Faults, Failures, Energy Consumption

I. INTRODUCTION

In cloud computing the assets are gotten remotely here and there which makes faults amid that procedure. To amend the defective segments adaptation to non-critical failure techniques are utilized and adjust them. Because of its remote access there are bunches of odds of mistakes so to accomplish unwavering quality progressively cloud computing. Adaptation to internal failure is accomplished by mistake handling having two constituent stages. These stages are "viable mistake forms" i.e. prior to the event of blunder and "dormant mistake handling" implies mistake won't happen once more. [1]

Service models of cloud computing

The various service models that cloud providers offered are mainly categorized in three categories: [1], [2]

1. Software as a Service (SaaS)
   [3]This model provides a 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. Various service provider like Google, salesforce , Microsoft provides Saas facility.

2. 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. Google's App Engine, Force.com is providing a platform to users. [4]

3. 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. To use the services the user has install own software over the infrastructure. Amazon and GoGrid are some example of IaaS. [5]

Diverse adaptation to non-critical failure methods and systems, we order faults as occur within cloud computing field as under

- Proactive adaptation to internal failure: The Proactive adaptation to non-critical failure intends to recognize the issue before it really happens. In this instrument, foresee the blame and maintain a strategic distance from recuperation from blame, blunders and failure. It keeps the parallel running applications from getting influenced because of failure of influenced single segments. [6]

- Reactive adaptation to non-critical failure Reactive adaptation to non-critical failure system decreases the exertion of failures, when failure happens. This On-request adaptation to non-critical failure strategy makes framework more robust. [7]
- Adaptive adaptation to internal failure: The versatile adaptation to non-critical failure methods are done naturally as indicated by the circumstance. Adaptive Fault Tolerance (AFT) can guarantee unwavering quality even under basic circumstances. [8]

Faults, Areas and Most Common strategies for handling faults

Faults are abnormal states occurring as result of falsifying hardware or software state. Faults may not necessarily cause deviation from actual results. In case actual result is not achieved, failure takes place. Thus, fault and failure correlate with each other. Error results as fault aggravates. The order of fault errors and failure is given as under[9]–[11]

Faults---→Errors--→Failures

Failure introduces inability to the system to perform desire operation. Cloud is prone to faults so handling and preventing faults is critical. Fault tolerance strategies thus incorporated within cloud environment. Cloud environments are frequently categorized as under

Mobile Cloud Computing

Authors discuss issues and challenges present within mobile cloud computing. Mobile cloud computing offloading strategies commonly employed for transfer of workload towards host machine. Faults at any stage could render overall operation. Although existing application partitioning algorithms allow an adaptive execution of the application between the mobile devices and the cloud servers, they still do not provide any solution on how to utilize and benefit from the elastic resources in the clouds. [12] Further one work proposed mobile cloud offloading to empower mobile devices with the computational capabilities. Decentralized approach followed within offloading process enhances storage and fetching operation within mobile cloud computing. Failure in offloading is commonly due to node failure. Tackling such a failure require higher node discovery where nodes discovered has large amount of energy associated with them. Nodes discovered in such manner tackle the issue of energy dissipation node failure by migration of resources to fittest node. It assigns data fragments to nodes such that other nodes retrieve data reliably with minimal energy consumption. It also allows nodes to process distributed data such that the energy consumption for processing the data is minimized. [13]

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

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. 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 areas 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. [17], [23]

* 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. [24], [25]

* 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. [23][26]

SHADOW RAPLICATION

This is the mechanism which follows both hardware and time redundant techniques. The process level replication is performed in this case. Large scale distributed system requires self configuration of resources with minimum human interference. Fault tolerance mechanisms for mobile agents to cope with the mobile server crash can be accomplished through shadow replication. The approach required in this case is known as dynamic shadow replication[14].

The basic concept of shadow replication is to associate with each main process a shadow. The size of shadow depends upon criticality of application. Shadow replication mechanism is described through the following steps.

- A process is given a responsibility to execute a task by allocating it exclusive core processor.
- Many Shadows are associated with the main process.
- Every shadow has equivalent speed and process allocation space as the original process.
- In case of failure main process is shifted to the shadow and shadow becomes main processor on which process executes[15].

Shadow replication suffers from the drawback of failure in shadow cores. But this problem can be rectified by the use of checkpoint in this scheme. [21], [22]

Section I contains the introduction of the work and further Section II carries the related work. Section III explains the methodology used for carrying out the work whereas Section IV explores the results that are obtained. Finally Section V has the brief conclusion.

II. RELATED WORK

Background analysis is conducted to analyze the efficient fault tolerant strategies used existing literature and to draw pros and cons of each for future enhancements.

Authors proposed a fault tolerance techniques in cloud computing along with its challenges and implementation. It also proposes HAproxy based cloud virtualized system. This system deals with faults that occur in system along with server application in cloud. If there occurs fault in any server over a connection then it automatically redirects to another one. Also data replication technique utilized to achieve high reliability. [16] Anthers proposed technique which is reliability based, for fault tolerance. The technique is AFTRC model that keep track of all nodes reliability. This technique deletes the node that has less reliability and the checkpoints are used. If any fault occurs during the transformation of data then backward recovery has been done by these checkpoints. [17] Proposed availability based fault tolerance technique that uses FTM model. This technique uses replication of users and when replica fails it uses the algorithm like gossip based protocol. [18] gives two visions for management of fault in cloud computing platforms. In the first one it provide management over the cloud provider and second one will share responsibilities over two participants over cloud. This paper has result that shows the improvement over the cloud. Also involve exclusive and collaborative fault tolerance techniques. [19] A paper described the study of different fault tolerant technique and describes different types of failure. It describes how fault can be occurs in system and which action should be taken if fault could occur. [3]Some proposed a FTC technique for fault tolerance which uses check pointing. This is good mechanism for fault that occurred in cloud computing over the infrastructure. The advantage of method is forward recovery that is provided with the help of checkpoints. The results show that this scheme has given reliability and less chances of failure. [20]

A work described model for fault tolerance and also compare them on the basis of metrics for fault tolerance. There are many techniques for fault tolerance in cloud computing and these techniques use different criteria's for finding faults. But still there are challenges that need some concern to speed up fault tolerance. [21] Another described technique that is based on scheduling using CAN in mobile computing. The CAN (content address networking) logically manages the locations of various devices the uses clouds for storing their data. Also malicious user filtering is done that ensure no unauthorized user could access the cloud. [22] One described the system that overcome the problem of energy efficiency and fault tolerance. Each fragment is assigned to node so that energy consumption is less during data retrieval. The result shows that implementation of the system gives feasible solutions. [23] Another work presented four crucial abstractions in the space of fault tolerant disseminated frameworks. Message Abstractions address the rightness of individual messages sent and got. Fault Abstractions address the sorts of faults conceivable and also their belongings in the framework. Fault-Masking Abstractions address the sorts of nearby calculations forms make to cover faults. At long last, Communication Abstractions address the sorts of information conveyed and the properties required for correspondence to prevail within the sight of faults. [24] A paper suggested that the capacity of fault tolerance is to safeguard the conveyance of anticipated administrations in spite of the nearness of fault-caused blunders inside the framework itself. Mistakes are identified and amended, and lasting faults are found and expelled while the framework keeps on conveying worthy administration. This objective is achieved by the utilization of fault recognition calculations, fault analysis, recuperation calculations and additional assets determination. [25]

1. COMPARISON TABLE

The comparison table indicate the techniques used in existing literature along with pros and cons of each. The comprehensive comparison suggests best possible technique in terms of parameters which can be used for future enhancements.

TABLE1: Comparison of Techniques, parameters and future enhancements

Paper / Parameters	Technique	Energy Efficiency	Parameters	Key Aspect	DATA PORTABILITY	FAIL OVER AND BACK UP
[15]	Shadow Replication	Fault tolerance is used and least work is done towards energy efficiency	Reliability Fault tolerance	Shadow replication is used primarily to achieve fault tolerance	Yes	Yes
[26]	Data Replication	Energy efficiency is considered	Energy Efficiency Fault Tolerance	Data replication of critical or sensitive data	Yes	Yes
[27]	Model Driven Replication	Not Used	Storage Availability	Model driven replication for enhancing replication	Data from data centers can be transferred to other data enters	Yes
[28]	Energy Efficient approach replication for cloud storage	Energy efficiency is achieved by disallowing topology breakage	Energy efficiency Fault tolerance	Topology breakage along with different shortage path algorithm for optimization	Data from data centers can be transferred to other data enters	Backup server is used
[29]	Energy efficient load balancing technique within cloud	Achieved through server inaccessible for idle resources	Energy efficiency Load balancing	Energy-aware computing in a cluster of servers running data-intensive workloads	Inter platform data transfer is supported	Back Server is used
[22]	Fault tolerance through QoS scheduling	Energy Efficiency is not considered	Execution Time Error Rate Reliability	Scheduling mechanism is proposed to achieve fault tolerance	Data portability is provided	Not used
[30]	Redundancy mechanism to achieve energy efficiency	Energy Efficiency is achieved	Energy Consumed Hit Ratio	In and on disk redundancy mechanism is used to achieve energy efficiency	Interdependence is used	Proxy servers are used for backup
[31]	Shadow Computing	Energy efficiency is achieved	Buffer Size Energy Efficiency	Shadow replication is used to copy sensitive information while non critical inform	Data moved between client and server	Provided with backup server

				ation is given less preference to conserve space		
[32]	Resource management through VM Migration	Not Used	Migration Time Down time	The focus of this article is to present the details of virtual machine migration techniques and their usage toward dynamic resource management in virtualized environments	Data moved between client and server	Provided with backup server
[33]	Benchmark mechanism instead of Keys is used	Analysis in terms monitoring and certification agencies	Sharing	Certification and monitor nodes are used	Data movement is supported	Provided with backup server
[34]	Fitness enabled technique is used	Mutation is used as a update mechanism	Sharing	Offspring mechanism is used for the same	Data is moved and altered depending upon fitness	Not specified
[35]	Dynamic auditing technique is used	Verification mechanism as access right	Auditing Execution Time	Administrator conduct Auditing	Data movement between source and destination	Provided with backup server

Comparison of techniques and parameters suggest there is a room for improvement in terms of fault tolerance rate, energy efficiency , reliability and availability.

III. METHODOLOGY

Profile the application functions by scanning source code. Split function of application which can be sent to cloud or executed on mobile on basis of memory usage implement enhanced snapshot maintaining techniques. Create and save the partitions. Now call the compiler to create application for device and its clone. Evaluate the application performance using the benchmark methods. The flow chart for implementation of the proposed approach is.

Flow Chart

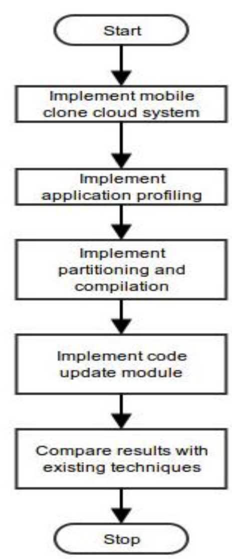

Fig. 4.4 Implement the mobile clone cloud system

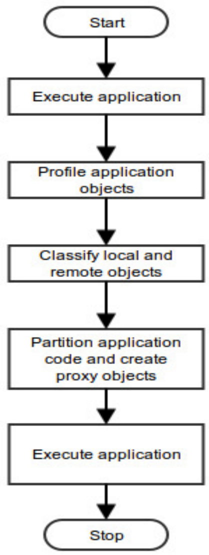

Fig 4.5.Execute the mobile clone cloud system

Proposed Algorithm:

Set NoC = number of classes, class = 0,
localClassList = empty list, remoteList = empty list

1. Get list of classes
2. For each class:
   a. Get list of function being called within class
   b. If function uses:
      i. Native library method
      ii. Camera/speakers/mic functions
   c. Then add class to localClassList
   d. Else if add class to remoteList
3. Next
4. For each class in localClassList:
   a. Create list of calls to remote classes
   b. Create list of shared variables between local and remote classes
5. For each class in remoteList:
   a. Create list of calls from remote to local classes
   b. Create list of calls dependent on native methods
6. Create proxy classes for remote and local method calls
7. Recreate classes with proxy connections
8. Recompile application

IV. RESULTS AND DISCUSSION

Below the results of the compile-time tests performed on the C-RAM are presented. We measured memory allocation and amount of time required to compile, for both the Fair play and compilers. In the latter case, we have data for compiling to and from the language. Our complete compiler is referred to as CBIR in this section.

Table 5.1 Comparing parameters value

Parameter	Existing value	Enhanced Value
Energy Efficiency	6.2	7.2
Response time	1.50	1.12
Failure overhead (in seconds)	30	25

5.1.1 Energy Efficiency

Device energy: It shows the improvement in efficiency of device energy with enhanced algorithm.

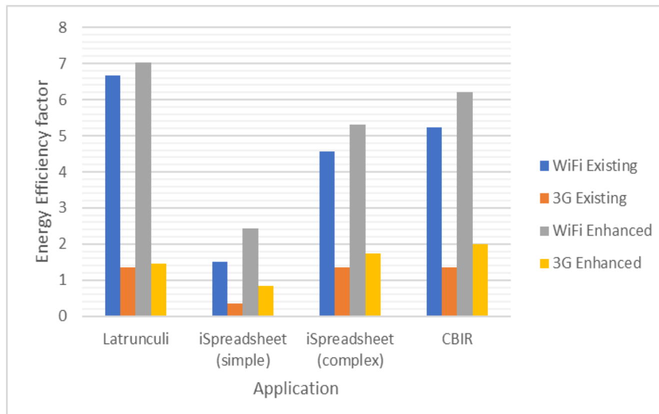

Fig.5.1 Device Energy

Response time: These graphs show response time with network types like Wifi and 3G. The graphs show improvement in enhanced.

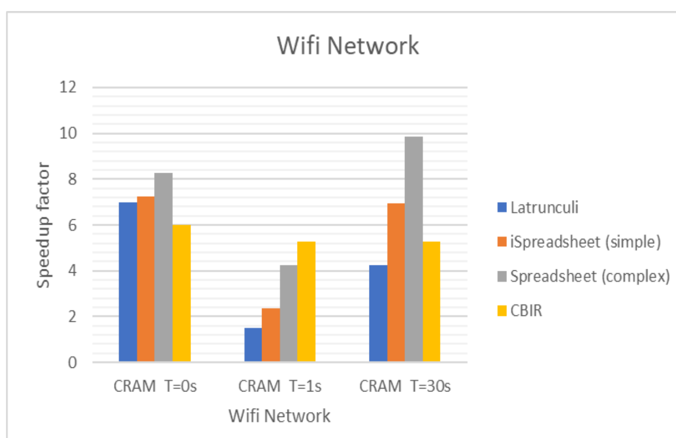

Fig.5.2 Response Time for Wifi Network

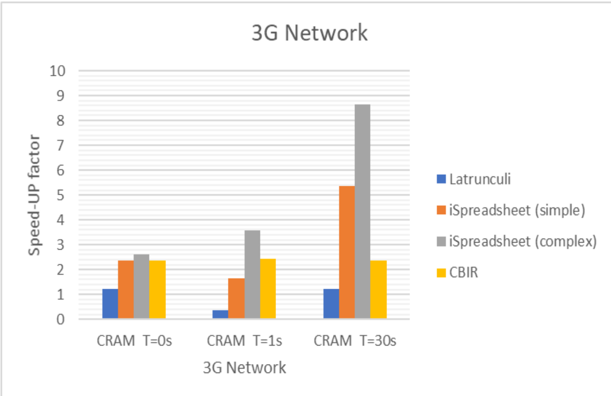

Fig.5.3 Response Time for 3G Network

5.1.3 Memory Utilization

Discussion: Memory utilization shows the usage of memory on cloud and device when AI, CSV processing or CBIR execution search is executed on the device.

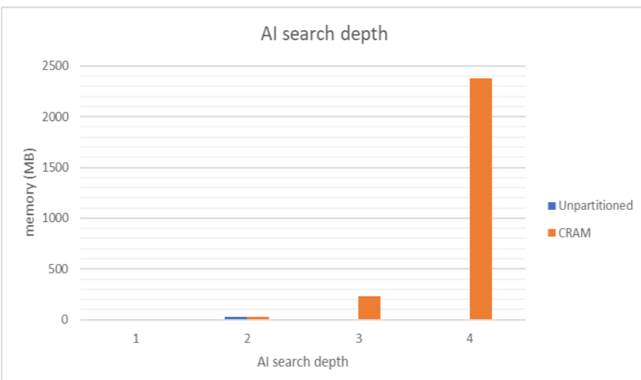

Fig.5.4 Unpartitioned AI Search Depth v/s CRAM AI Search Depth

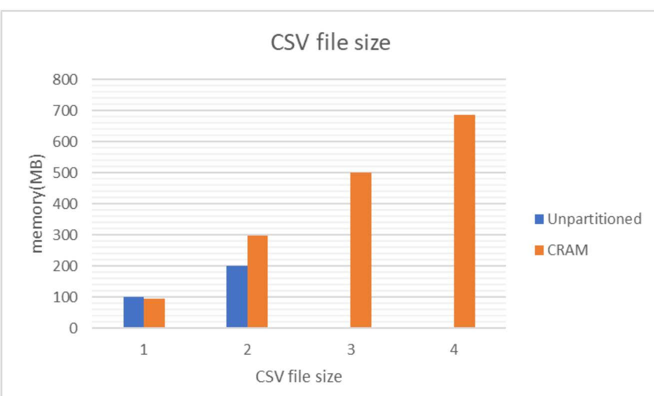

Fig.5.5 Unpartitioned CSV file size v/s CRAM CSV file size

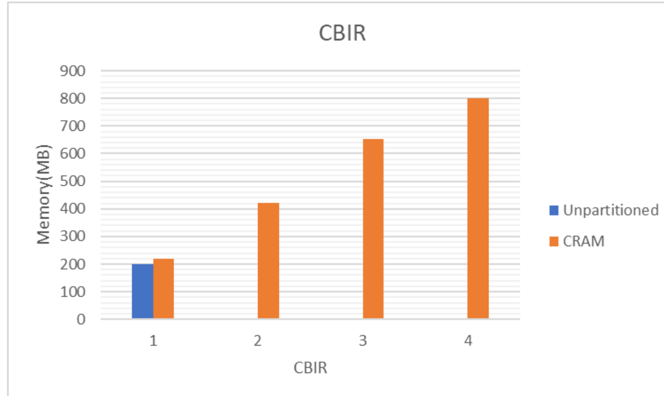

Fig.5.6 Unpartitioned CBIR File v/s CRAM CBIR File

Comparison with pure virtual memory approach: Next we compare C-RAM against an approach that uses only C-RAM's virtual memory scheme to allow an application to consume more than the available device memory. Virtual memory with on-demand paging carries a fundamental overhead, mostly due to the time to load and store application objects from/to flash memory during execution. This is a reason why current mobile platforms do not offer support for on-demand paging to work around memory limitations. Although C-RAM primarily aims to extend the memory of mobile devices, it can also save energy. However, similar to the speed-ups observed for execution, energy savings are not guaranteed for all applications—they are instead a consequence of C-RAM's state partitioning approach, which tries to mask the associated network overhead by distributing execution accordingly.

V.CONCLUSION AND FUTURE SCOPE

The energy efficiency and fault tolerance is need of the hour and is state of the art problem required to be tackled while using any concerned field associated with cloud computing. Mobile cloud computing (MCC) has turn out to be the must for today's Android phones and their composite functions. Remote implementation on cloud carries confronts like failure of network, issues of scalability and partitioning technique that require to be handled suspiciously. Each procedure has little drawbacks that must to be taken in consideration throughout the research. This study analysis the scope, challenges, solutions and approaches in the field of Mobile Cloud Computing (MCC). This study focuses on Energy saving in mobile devices, migration problems, application advance platforms and the various mobile cloud computing (MCC) applications. In future shadow replication with core computing can be used to reduce effect of fault on high performance systems.

REFERENCES

[1] 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.

[2] 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.

[3] Z. Amin, "Review on Fault Tolerance Techniques in Cloud Computing," vol. 116, no. 18, pp. 11–17, 2015.

[4] 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.

[5] B. Nicolae, "BlobCR : Efficient Checkpoint-Restart for HPC Applications on IaaS Clouds using Virtual Disk Image Snapshots."

[6] B. H. Oh and J. Lee, "Constraint-based proactive scheduling for MPTCP in wireless networks," *Comput. Networks*, vol. 91, pp. 548–563, 2015.

[7] T. Hirofuchi, H. Nakada, S. Itoh, and S. Sekiguchi, "Reactive Consolidation of Virtual Machines Enabled by Postcopy Live Migration," pp. 11–18, 2011.

[8] J. Hursey, J. M. Squyres, T. I. Mattox, and A. Lumsdaine, "The Design and Implementation of Checkpoint / Restart Process Fault Tolerance for Open MPI ∗."

[9] B. Schroeder and G. a. Gibson, "Disk failures in the real world: What does an MTTF of 1,000,000 hours mean to you," *Conf. File Storage Technol.*, no. September, pp. 1–16, 2007.

[10] G. Aupy, A. Benoit, M. E. M. Diouri, O. Glück, and L. Lefèvre, "Energy-Aware Checkpointing Strategies," pp. 279–317, 2015.

[11] W. Chen, S. Member, R. Ferreira, E. Deelman, and T. Fahringer, "Dynamic and Fault-Tolerant Clustering for Scientific Workflows," vol. 7161, no. FEBRUARY, pp. 1–14, 2015.

[12] K. Akherfi, M. Gerndt, and H. Harroud, "Mobile cloud computing for computation offloading : Issues and challenges," *Appl. Comput. Informatics*, 2017.

[13] X. Chen, "Decentralized Computation Offloading Game for Mobile Cloud Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.

[14] J. Xu and S. Pears, "A Dynamic Shadow Approach to Fault-Tolerant Mobile Agents in an Autonomic Environment," *Real-Time Syst.*, vol. 32, no. 3, pp. 235–252, Mar. 2006.

[15] X. Cui, B. Mills, T. Znati, and R. Melhem, "Shadow replication: An energy-aware, fault-tolerant computational model for green cloud computing," *Energies*, vol. 7, no. 8, pp. 5151–5176, 2014.

[16] A. Tchana, L. Broto, and D. Hagimont, "Fault Tolerant Approaches in Cloud Computing Infrastructures," no. c, pp. 42–48, 2012.

[17] A. Bala and I. Chana, "Fault Tolerance-Challenges , Techniques and Implementation in Cloud Computing Fault Tolerance- Challenges , Techniques and Implementation in Cloud Computing," no. January 2012, 2015.

[18] D. Terry, "Toward a New Approach to IoT Fault Tolerance," *Computer (Long. Beach. Calif).*, vol. 49, no. 8, pp. 80–83, 2016.

[19] 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.

[20] V. M. Sivagami, "Survey on Fault Tolerance Techniques in Cloud Computing Environment," no. 9, pp. 419–425, 2015.

[21] G. A. Kaur and S. Sharma, "Fault Tolerance in Cloud Computing : A Review," vol. 8491, pp. 94–98, 2017.

[22] S. Choi, K. Chung, and H. Yu, "Fault tolerance and QoS scheduling using CAN in mobile social cloud computing," 2013.

[23] C. A. Chen, M. Won, R. Stoleru, and G. G. Xie, "Energy-efficient fault-tolerant data storage and processing in mobile cloud," *IEEE Trans. Cloud Comput.*, vol. 3, no. 1, pp. 28–41, 2015.

[24] R. Jhawar, V. Piuri, and M. Santambrogio, "A Comprehensive Conceptual System-Level Approach to Fault Tolerance in Cloud Computing," pp. 0–4, 2012.

[25] S. K. Suman, "Effective Fault Tolerance Management in Cloud Computing," vol. 6, no. 3, pp. 192–198, 2016.

[26] D.-W. Sun, G.-R. Chang, S. Gao, L.-Z. Jin, and X.-W. Wang, "Modeling a Dynamic Data Replication Strategy to Increase System Availability in Cloud Computing Environments," *J. Comput. Sci. Technol.*, vol. 27, no. 2, pp. 256–272, 2012.

[27] K. Ranganathan, A. Iamnitchi, and I. Foster, "Improving data availability through dynamic model-driven replication in large peer-to-peer communities," *2nd IEEE/ACM Int. Symp. Clust. Comput. Grid, CCGrid 2002*, pp. 0–5, 2002.

[28] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," *Globecom Work. (GC Wkshps), 2013 IEEE*, pp. 446–451, 2013.

[29] W. Lang, J. M. Patel, and J. F. Naughton, "On energy management, load balancing and replication," *ACM SIGMOD Rec.*, vol. 38, no. 4, p. 35, 2010.

[30] J. Wang, X. Yao, and H. Zhu, "Exploiting in-memory and on-disk redundancy to conserve energy in storage systems," *IEEE Trans. Comput.*, vol. 57, no. 6, pp. 733–747, 2008.

[31] B. Mills, T. Znati, R. Melhem, and K. B. Ferreira, "Shadow Computing An Energy-Aware Resiliency Scheme for High Performance Computing."

[32] M. Mishra and A. Das, "Dynamic resource management using virtual machine migrations," *IEEE Commun. Mag.*, vol. 50, no. September, pp. 34–40, 2012.

[33] K. Hwang, X. Bai, Y. Shi, M. Li, W.-G. Chen, and Y. Wu, "Cloud Performance Modeling with Benchmark Evaluation of Elastic Scaling Strategies," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 1, pp. 130–143, Jan. 2016.

[34] H. Wang, Z. Kang, and L. Wang, "Performance-Aware Cloud Resource Allocation via Fitness-Enabled Auction," *IEEE Trans. Parallel Distrib. Syst.*, vol. 27, no. 4, pp. 1160–1173, Apr. 2016.

[35] K. Yang and X. Jia, "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 9, pp. 1717–1726, Sep. 2013.

Authors Profile

Rajeev Kumar Bedi is currently working as Assistant Professor in Beant College of Engineering and Technology, Gurdaspur (Punjab). He has completed B.Tech and Post graduation M.Tech from Punjab Technical University, Jalandhar. He is currently persuing Ph.D from Punjabi University, Patiala. He has published papers in reputed journals including Springers, Elsevier etc. His main Research work focusses on Cloud Computing, Mobile Cloud Computing and Wireless Netwok.

Mrs. Vandana is currently pursuing M.Tech (CSE) from Beant College of Engineering and Technology (BCET), Gurdaspur under IKG Punjab Tachnical University, Jalandhar, Punjab.

Dr. Sunil Kumar Gupta did B.E. in Computer Science from Gorakhpur University, Gorakhpur, India in 1988, and M.S. in 1991 and completed Ph.D. in Computer Science from Kurukshetra University, Kurukshetra, India. He possesses 28 years of teaching experience. He has worked as teaching faculty in many reputed institutions in India including N.I.T., Hamirpur (HP). Presently, he is working as Associate Professor in Computer Science & Engg. Department at Beant College of Engineering and Technology, Gurdaspur (India).He has more than 40 research publications. His work is published and cited in highly reputed journals of Elsevier, Springer, Taylor and Francis and IEEE. His areas of interest include database management systems, distributed systems, cloud computing and mobile computing and security.

