IJITE INTERNATIONAL JOURNAL OF INNOVATIVE TRENDS IN ENGINEERING (IJITE) ISSN: 2395-2946
VOLUME 18, NUMBER 02, 2016

Fault Tolerance over Cloud with Task Migration

Shivam Choudhary, Prof. Sanjay Gupta

Department of Computer Science Engineering, M. Tech (CTA) VITS Jabalpur (M.P.) India

Abstract- Cloud computing may be an approach of computing wherever service is provided across the web using the models and levels of abstraction. Several analysis problems area unit needed to be absolutely addressed in cloud like Fault tolerance, advancement management, advancement programming, security, etc. Fault Tolerance in cloud may be a major concern to ensure the supply and dependability of essential services further as application execution. So as to attenuate failure impact on the system and application execution, failures ought to be anticipated and proactively handled. The task of providing fault tolerance as a service needs the service supplier to appreciate generic fault tolerance mechanisms specified the client's applications deployed in virtual machine instances will transparently get fault tolerance properties. To this aim, we tend to outline ft−unit because the basic module that applies a coherent fault tolerance mechanism to a continual system failure at the graininess of a VM instance. The notion of ft−unit is predicated on the observation that the impact of hardware failures on client's applications will be handled by applying fault tolerance mechanisms directly at the virtualization layer than the applying itself.

Keywords – Cloud Computing, Cloud design, Virtualization, Distributed Computing.

I. INTRODUCTION

Cloud computing may be a comprehensive way delivers IT as a service. It's a web based large computing mechanism wherever shared resources are provided like electricity. The pliability of cloud computing may be a work of allocating resource on demand. Cloud computing is that the combination of grid computing and utility computing. Cloud computing has the potential to form irreversible changes in whatever numbers of computers are used around the world. It's a delivery of computing and storage capability as a service to a community of end-to-end recipients. [1]

Cloud computing may be an approach of computing wherever service is provided across the web using the models and levels of abstraction. Several analysis problems area unit are absolutely addressed in cloud like Fault tolerance, advancement management, advancement programming, security, etc. Grid computing may be a federation of resources from multiple body domains to succeed in a typical goal in an exceedingly single task to resolve the grand challenge drawback like biological process, monetary modeling etc. Grid computing is sharing of coordinated resources in an exceedingly dynamic atmosphere during which multi-institutional organizations concerned. [1]

The major layers in cloud design associate the various components of the cloud applications. The components of the cloud includes computer, mobile or alternative handheld devices accustomed hook up with the cloud over net, varied servers that area unit accustomed settle for the shopper requests and supply services to them from the cloud, the tools specific to the varied cloud applications like info, hardware resources, applications etc. and at last an information center and broker applications which give the authentication, authorization, confidentiality and sharing of resources to the varied users of the cloud. [6]

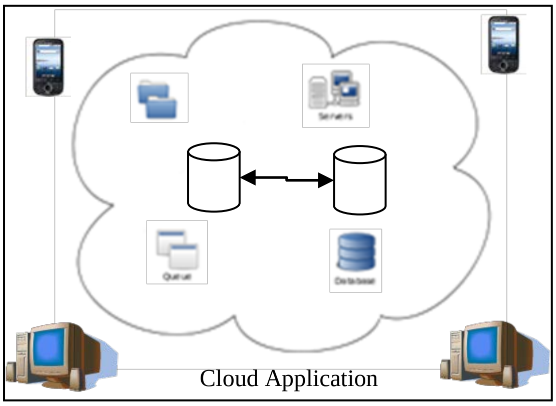


Figure 1: Cloud Computing

According to the usage of the cloud it's either public, non-public or hybrid cloud. Once a Cloud is formed on the market in an exceedingly pay-per-use manner to the general public, we tend to decision it a Public Cloud and therefore the service being sold-out is Utility Computing. Once the services area unit reserved for a few specific organization then the cloud is taken into account as non-public cloud and it works for specific organization. A number of the user oriented applications like shopping carts, banking services etc needs each behavior of Public and Private Clouds, such clouds are termed as Hybrid Clouds. [6]

Cloud computing may be a wide space network based computing, wherever shared resources like software package, and knowledge area unit provided to computers and alternative devices whenever a shopper demands them either as paid or free services.

On the idea of the higher than discussion, platforms like YouTube, Vimeo, Flickr, Slideshare and Skype are enclosed in an exceedingly large list of cloud applications – platforms that hold your information (images, video, shows, voice) and manages all thus you don't got to worry regarding them.

A. Merits & Demerits of Cloud Computing

Merits of cloud computing is too many to enlist, nearly a couple of big merits of the cloud computing are as follows:

1. Cloud computing facilitates the equal flow of knowledge between the outsourced and outsourcing services.

2. Data Center idea permits for centralized information processing and therefore all the users get equal quantity of updated information.

3. The simple flow of knowledge permits the host organization ensure a small degree assurance to the staff regarding their work and data management.

The major issues with cloud computing is as follows:

1. Ends up in management issues.

2. Disagreement at intervals the data technology departments.

3. The webmasters got to know about new systems for managing the conflict.

4. Further communication system and its configurations area unit needed, so there's another company concerned within the business may not get affected.

5. Businesses that traumatize responsive information are involved regarding safety of their mechanism [2]

II. ARCHITECTURE OF CLOUD COMPUTING

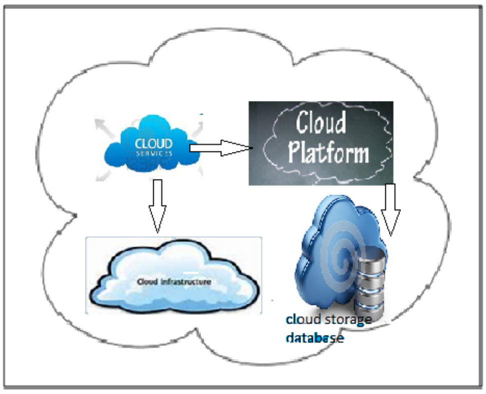


Figure 2: Architecture of Cloud

Cloud design is consisting of multiple resources operating for cloud altogether with one another having loose coupling between them so the system won't have direct dependencies and any of the half will be additional, updated or modified just in case of requirement/failure while not moving the remainder of the system. It involves each hardware and software package applications. [1,2,3]

For loose coupling between the varied applications over the cloud messages queues area unit used so dependencies between them are manageable.

The Cloud Computing design is that the structure of the system, that consists of on-premise and cloud resources, services, middleware, and software package elements, their geo-location, their outwardly visible properties and therefore the relationships between them. Within the space of cloud computing, protection depends on having the correct design for the correct application. Organizations should perceive the individual needs of their applications, and if already employing a cloud platform, perceive the corresponding cloud design.

A cloud computing design consists of a face and a face. They hook up with one another through a network, typically the web. The face is that the aspect the pc user, or client, sees. the rear finish is that the "cloud" section of the system. [2,3]

III. EXISTING SYSTEM

There are a unit varied faults which might occur in cloud computing. Supported the fault tolerance policies varied fault tolerance techniques will be used like advancement level and task level.

A. Proactive fault tolerance: The principle of proactive fault tolerance policies is to avoid recovery from faults, errors and predict the failure and proactively replace the suspected elements from alternative operating elements. a number of the techniques supported these policies area unit Pre-emptive migration and software package Rejuvenation. a. Pre-emptive Migration: Proactive fault tolerance mistreatment pre-emptive migration depends on a circuit management mechanism wherever application is continually monitored and analyzed. b. software package Rejuvenation: it's a method that styles the system for periodic reboots. It restarts the system with clean state.

B. Reactive fault tolerance: Reactive fault tolerance policies cut back the result of failures on application execution once the failure effectively happens. There area unit varied techniques that area unit supported these policies like checkpoint/Restart, Replay and rehear. a. Check pointing/Restart: once a task fails, it's allowed to be restarted from the recently checked pointed state instead of from the start. it's associate degree economical task level fault tolerance technique for running applications. b. Replication: Replication based mostly technique is one in every of the favored fault tolerance techniques. Reproduction suggests that multiple copies. Replication may be a method of maintaining completely different copies of an information item or object. In replication techniques, request from shopper is forwarded to 1 of reproduction among the set of replicas. Varied task replicas area unit run on completely different resources, for the execution to succeed until the whole replicated task isn't crashed. Replication adds redundancy within the system. It will be enforced mistreatment tools like HAProxy, Hadoop and AmazonEC2.

Consistencies among reproduction, reproduction management, reproduction on demand, and degree of reproduction etc area unit some vital problems in replication based mostly fault tolerance technique. A replication protocol should make sure the consistency among all replicas of a similar object.

C. Task Resubmission: it's the foremost wide used fault tolerance technique in current scientific advancement systems. Whenever a failing task is detected, it's resubmitted either to the same or to a unique resource at a runtime.

Fault Tolerance may be a configuration that forestalls a laptop or network device from failing within the event of sudden drawback or error like hardware failure, link failure, unauthorized access, variations within the configuration of various systems and system running out of memory or disc space. The combination of fault tolerance measures with programming gains a lot of importance. Scientific workflows use distributed heterogeneous resources in cloud interface area unit typically onerous to program. This paper explains task replication technique and simulation of cloud computing systems. [1]

The collection and prompt analysis of synchrophasor measurements may be a key step towards sanctionative the longer term sensible grid, during which grid management applications would be deployed to observe and react showing intelligence to ever-changing conditions. The potential exists to slash inefficiencies and to adaptively reconfigure the grid to require higher advantage of renewable, coordinate and share reactive power, and to scale back the danger of ruinous large-scale outages. However, to appreciate this potential, variety of technical challenges should be overcome. we tend to describe a ceaselessly active, timely observation framework that we've created, architected to support a large vary of grid-control applications in an exceedingly normal manner designed to leverage cloud computing. Cloud computing systems bring vital blessings, as well as associate degree elastic, extremely on the market and efficient reckon infrastructure well-suited for this application. We tend to believe that by showing however challenges of dependability, timeliness, and security will be addressed whereas investing cloud standards, our work opens the door for wider exploitation of the cloud by the sensible grid community. This paper characterizes a PMU-based state-estimation application, explains however the required system maps to a cloud design, identifies limitations within the normal cloud infrastructure relative to the requirements of this use case, and so shows however we tend to adapt the fundamental cloud platform choices with subtle technologies of our own to attain the desired levels of usability, fault tolerance, and correspondence. [2]

The most common benchmarks for cloud computing area unit the Terasort benchmark and therefore the YCSB benchmark. Though these benchmarks area unit quite helpful, they weren't designed for information warehouse systems and connected OLAP technologies. During this paper, first, we tend to gift cloud computing and information warehouse systems. Then, we tend to argue that TPC-H benchmark -the most distinguished benchmark for call web, mismatches cloud principle (scalability, elasticity, pay-per-use, fault-tolerance features) and client Relationship Management principle (end-user satisfaction, Quality of Service features). Finally, we tend to gift new needs for implementing a benchmark for information warehouse systems within the cloud. The planned needs ought to permit a good comparison of various cloud systems providers’ offerings. [3]

The increasing quality of Cloud computing as gorgeous different to classic IP systems has exaggerated the importance of its correct and continuous operation even within the presence of faulty elements. During this paper, we tend to introduce associate degree innovative, system-level, standard perspective on making and managing fault tolerance in Clouds. we tend to propose a comprehensive high-level approach to shading the implementation details of the fault tolerance techniques to application developers and users by suggests that of an ardent service layer. Particularly, the service layer permits the user to specify and apply the required level of fault tolerance, and doesn't need data regarding the fault tolerance techniques that area unit on the market within the visualized Cloud and their implementations. [4]

SLAs area unit common suggests that to outline specifications and needs of cloud computing services, wherever the secure availableness is one in every of the foremost vital parameters. Fulfilling the stipulated availableness could also be dear, because of the value of failure recovery software package, and therefore the quantity of physical instrumentality required to deploy the cloud services. Therefore, a relevant question for cloud suppliers is: a way to guarantee the SLA availableness in an exceedingly value economical way? This paper studies completely different fault tolerance techniques on the market within the market, associate degreed it proposes the employment of an hybrid management to own full management over the SLA risk, mistreatment solely the required resources so as to stay a value economical operation. This paper shows a way to model the likelihood distribution of the accumulated period, and the way this may be employed in the planning of hybrid policies. Mistreatment specific case studies, this paper illustrate a way to implement the planned hybrid policies, and it shows the obtained value saving by mistreatment them. This paper takes advantage of the cloud computing flexibility, and it opens the door to the employment of dynamic management policies to succeed in performance objectives in ICT systems. [5]

V. PROPOSED WORK

This outline work elaborates the potential reasons of faults occurring within the cloud ADPS. Fault tolerance mechanism being applied in cloud is said with task migration techniques.

There area unit varied faults which might occur in cloud computing. Supported the fault tolerance policies varied fault tolerance techniques will be used like advancement level and task level.

A. Proactive Fault Tolerance: The principle of proactive fault tolerance policies is to avoid recovery from faults, errors and predict the failure and proactively replace the suspected elements from alternative operating elements. A number of the techniques supported these policies are Pre-emptive migration and software package Rejuvenation.

B. Reactive Fault Tolerance: Reactive fault tolerance policies cut back the result of failures on application execution once the failure effectively happens. There are a unit varied techniques that area unit supported these policies like checkpoint/Restart, Replay and rehear.

c. Task Resubmission: it's the foremost wide used fault tolerance technique in current scientific advancement systems. Whenever a failing task is detected, it's resubmitted either to the same or to a unique resource at a runtime.

The complete fault tolerance shall be preventively performed in following steps:

Step 1: many virtual machines shall be created over the cloud created using Cloudsim software.

Step 2: every virtual machine shall be allotted with several applications through the cloudlets.

Step 3: Load on virtual machines shall be tracked using the infinitely running thread.

Step 4: If the load will increase to a threshold worth then the cloudlets shall be migrated to a different virtual machine that is comparatively free.

Step 5: Track of the virtual machine shall be done through a global resource that is shareable to all or any of the virtual machines.

Step 6: Task shall be migrated with the most effective match technique shall be used for migration to avoid the quickly occurring on a specific virtual machine.

Step 7: Threshold shall be set by pursuit the speed of load being applied over the cloud and can be adjusted dynamically.

Best Fit Algorithm

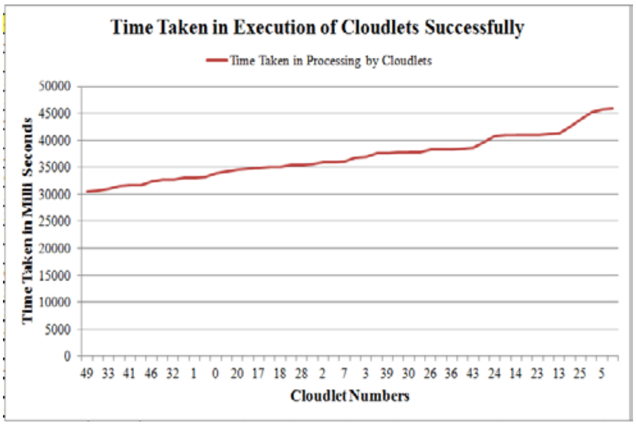
```
Algorithm for allocate (n)
{
        size (block) = n + size(header)
        Scan free list for smallest block with nWords >=
size(block)
        If block not found
                Failure (time for garbage collection!)
        Else if free block nWords >= size(block) +
threshold*
                Split into a free block and an in-use
block
                Free block nWords = Free block nWords
- size(block)
                In-use block nWords = size(block)
                        Return pointer to in-use block
        Else
                Unlink block from free list
                Return pointer to block
        *Threshold must be at least size(header) + 1 to
leave room for header and link
        Threshold can be set higher to combat
fragmentation
}
```

Allocation time is O(K) (K = number of blocks in free list)

VI. RESULTS & DISCUSSION

Table: Cloud lets and the time taken by them during the simulation

Cloudlet ID	Time Taken in Processing in Millis Seconds
49	30424
10	30680
33	31030
37	31482
41	31626
12	31744
46	32336
6	32620.8
32	32665.6
22	32968
1	32992.4
16	33151.16
0	33932.76
29	34174.36
20	34461.94
8	34767.94
17	34944.34
42	34982.38
18	35020.72
38	35357.92
28	35404.32
48	35531.92
2	35944.78
19	35951.56
7	35993.14
4	36669.94
3	36831.54
44	37513.12
39	37638.76
47	37677.16
30	37710.76
9	37785.56
26	38278.36
27	38291.16
36	38309.56
40	38375.56
43	38555.94
21	39592.38
24	40813.14
31	40985.18
14	41006.74
11	41025.58
23	41039.14
34	41189.14
13	41363.14
35	42403.94
25	43871.14
45	45105.94
5	45749.14
15	45910.34



Inference: From the above graph it is clear that the time taken in processing of cloudlets is average time and do not affect the performance of the system due to proactive fault tolerance mechanism. This also shows that the amount of time required to evaluate best fit cloudlet do not affect the working of the cloud very much.

Table: Cloud lets and the file size processed by them during the simulation

Cloudlet ID	File Size processed in bytes
49	1358
10	2904
33	7713
37	5256
41	5590
12	4403
46	1816
6	5172
32	6100
22	6195
1	2076
16	495
0	8127
29	1112
20	6151
8	255
17	3374
42	4017

18	2277
38	8767
28	1920
48	8231
2	3801
19	7346
7	1264
4	5173
3	5276
44	4539
39	5875
47	8041
30	7823
9	3228
26	8586
27	1617
36	7954
40	6692
43	2610
21	3229
24	796
31	7778
14	4866
11	6451
23	5281
34	8361
13	5829
35	2491
25	7160
45	858
5	8718
15	3479

Inference: The above graph shows that the file size applied is different for different cloudlets and each one of them is processed with same average time as shown in the graph. This concludes that the cloudlets with different sizes also not affect the working of the proposed system and algorithm.

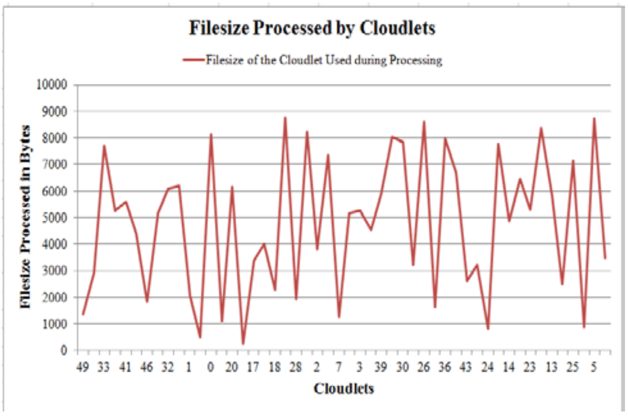
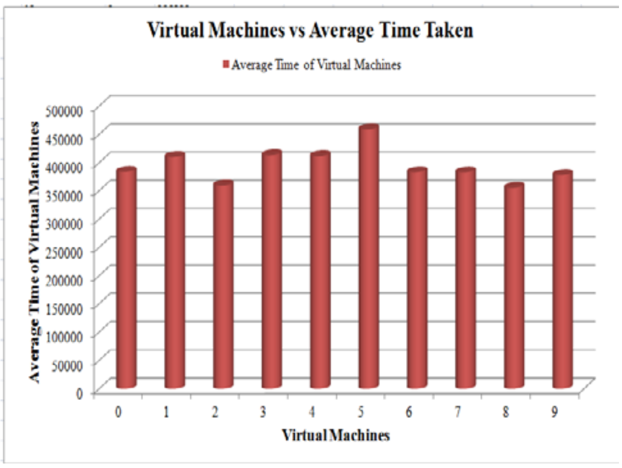


Table: Virtual Machine and their average processing time during the simulation

Virtual Machine ID	Average Time Taken in Millis Seconds
0	383756
1	410256
2	359448
3	413632
4	411892
5	459104
6	383096
7	382912
8	355320
9	377856



Inference: The above graph has been drawn by taking the average time required on each virtual machine having different cloudlets running on them. This indicates that the average time for which any virtual machine is working is almost same and there are no exceptions during the processing. This also helps in concluding that the proposed work is not overloading the system or any particular virtual machine.

Comparison with Existing Work

SNO	Feature	Base Paper Work	Proposed Work
1.	Mechanism	reactive fault tolerance algorithm	Proactive Fault Tolerance
2.	Algorithm	Task Replication Technique	VM Migration Technique using Best Fit Algorithm
3.	Algorithm Complexity	O(e*p) where e = Dset is the number of Task	O (n * log n) where n is number of virtual machines. Log n

		Dependencies and p is number of resources	is the time required for searching best fit cloudlet on any virtual machines.
4.	Resource Requirement	High as the replication will occupy double space	Low as best fit requires to migrate means only one occurrence remains in memory
5.	Results Obtained	94% success rate as the system is reactive	100% Success Rate as the system is proactive

VII. CONCLUSION

The proposed work in this paper is to provide a mechanism of fault tolerance in cloud environment. The task of providing fault tolerance as a service needs the service supplier to appreciate generic fault tolerance mechanisms specified the client's applications deployed in virtual machine instances will transparently get fault tolerance properties.

The planned ft−unit is that the basic module, that applies a coherent fault tolerance mechanism to a continual system failure at the graininess of a VM instance. The notion of ft−unit is predicated on the observation that the impact of hardware failures on client's applications will be handled by applying fault tolerance mechanisms directly at the virtualization layer than the applying itself. Proactive Fault Tolerance is usually desirable approach of handling applications however complexities concerned within the same at the side of the performance compromises makes it value analysis space.

The planned work is predicted to perform the most effective because the resources are a managed at one location and every migration is ready to find next VM and therefore the migration time won't be exaggerated for looking succeeding on the available resource.

REFERENCES

[1] K.Ganga, Dr S.Karthik, "A Fault Tolerent Approach in Scientific Workflow Systems based on Cloud Computing," Proceedings of the 2013 International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME) February 21-22 978-1-4673-5845-3/ ©2013 IEEE

[2] Ketan Maheshwari, Marcus Lim, Lydia Wang, Ken Birman, Robbert van Renesse, "Toward a reliable, secure and fault tolerant smart grid state estimation in the cloud" 978-1-4673-4896-6/ ©2013 IEEE

[3] Rim Moussa, "Benchmarking Data Warehouse Systems in the Cloud," 978-1-4799-0792-2/ ©2013 IEEE

[4] Ravi Jhawar, Graduate, Vincenzo Piuri, Marco Santambrogio, "Fault Tolerance Management in Cloud Computing: A System-Level Perspective" IEEE SYSTEMS JOURNAL, VOL. 7, NO. 2, JUNE 2013, 1932-8184 © 2012 IEEE

[5] Andres J. Gonzalez and Bjarne E. Helvik, "Hybrid Cloud Management to Comply Efficiently with SLA Availability Guarantees", 2013 IEEE 12th International Symposium on Network Computing and Applications, 978-0-7695-5043-5 © 2013 IEEE DOI 10.1109/NCA.2013.32

[6] Qi Zhang · Lu Cheng · Raouf Boutaba, "Cloud computing: state-of-the-art and research challenges" , Springer, J Internet Serv Appl (2010) 1: 7–18 DOI 10.1007/s13174-010-0007-6

[7] Eman M.Mohamed, Hatem S. Abdelkader, Sherif EI-Etriby, "Enhanced Data Security Model for Cloud Computing", The 8th International Conference on INFOrmatics and Systems (INFOS2012) - 14-16 May, 2012

[8] http://www.cloudcomputinginsights.com/management/cloud-computing-advantages-and-disadvantages/?mode=featured

[9] Borko Furht, Armando Escalante, "Handbook of Cloud Computing," ISBN 978-1-4419-6523-3, e-ISBN 978-1-4419-6524-0, DOI 10.1007/978-1-4419-6524-0, Springer

[10] Manop Phankokkruad, "Implement of Cloud Computing for e-Leaming System" 2012 International Conference on Computer & Information Science (ICCIS), 978-1-4673-1938-6/12 ©2012 IEEE

[11] Antonina Litvinova, Christian Engelmann and Stephen L. Scott, "A Proactive Fault Tolerance Framework for High Performance Computing," 2009.