International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 03 | June-2015 www.irjet.net p-ISSN: 2395-0072

ENHANCING FAST RETRANSMISSION AND FAST RECOVERY IN CLOUD MOBILE MEDIA

B.Raveendar[1], Dr.P.Marikannu[2]

[1] *III-M.tech(IT) , Department of Information technology, Regional Centre, Anna University, Coimbatore, India*
[2] *Head of the Dept, Department of Information technology, Regional Centre, Anna University, Coimbatore, India*

Abstract - ***In cloud media we propose two techniques. End-to-end view, Layered view. Using end-to-end view increase the fasten transformation between the mobile device and back-end content delivery system. Using layered view develop the security of the system with a effective techniques. Here we introduce a efficient security system to monitor the intruders. And also using the cloud centric media platforms system architecture, cloud media platform service, interaction applications. We proposed cloud centric media platform (CCMP) for giving extra features for this project. Earlier we used either cloud centric or media centric but now using mobile media cloud centric technique to achieve a good result. Summaries of existing research in this area are organized according to the layered service framework: i) cloud resource management and control in infrastructure-as-a-service (IaaS), ii) cloud-based media services in platform-as-a-service (PaaS), and iii) novel cloud-based systems and applications in software-as-a-service (SaaS).***

Key Words: *cloud centric, media centric*

1. Introduction

Increasing of mobile device with wireless internet has fueled an increasing user demand on rich media expectance on to go. This will cause the growth of mobile traffic, dominated by video contents. And mobile video will increasing 25-fold between 2011 and 2016, accounting for over 70% of total mobile data traffic by 2016. Increase scalability, heterogeneity, reliability, usability, and security. It will manage cost control, revenue marketing, privacy, security and trust. And also improve the performance, lower cost consumption for implementation, better QOS/QOE, human centric social media maintenance.

In resource management and controlling, use cloned virtual machine environment to execute the mobile application inside cloud. Clone Cloud used to combine static analysis and dynamic profiling to participate the application automatically at a fine granularity. ThinkAir technique used to provide a framework for migrating smart phone application to the cloud by means of virtualization and method-level computation offloading. Using authors formulate intra-cloud resource problem based on the queuing model of the systems.

A unified optimization framework, objective is to minimize the total cost of ownership for a cloud media network including both upfront cost and re-occurring cost. In this system, will reduce the processing time of the system by introducing the cloud centric mobile media technique for users according to their requirements. This approach not only reduces the processing overhead, it also provides results with higher Quality Of Service. Since, in this system no need of searching the optimal data planning for each user, because it automatically takes the corresponding value of the device which is held by the consumer.

1.1 Foundations of Cloud Computing

A hybrid cloud model is a combination of private clouds with public clouds. Private and public clouds mainly differ on the type of ownership and access rights that they support. Access to private cloud resources is restricted to the users belonging to the organization that owns the cloud. On the other hand, public cloud resources are available on the Internet to any interested user under pay-as-you-go model. Hence, small and medium enterprises (SMEs) and governments have started exploring demand-driven provisioning of public clouds along with their existing computing infrastructures (private clouds) for handling the temporal variation in their service demands.

This model is particularly beneficial for SMEs and banks that need massive computing power only at a particular time of the day (such as back-office processing, transaction analysis). However, writing the software and developing application provisioning techniques for any of the Cloud models – public, private, hybrid, or federated – is a complex undertaking. There are several key challenges associated with provisioning of applications on clouds: service discovery, monitoring, deployment of virtual machines and applications, and load-balancing among others. The effect of each element in the overall Cloud operation may not be trivial enough to allow isolation, evaluation, and reproduction.

Cloud computing holds a promise to deliver large-scale utility computing services to a wide range of consumers. Through the creation of hybrid clouds, one can use this internal infrastructure with public cloud resources. Grid computing provides the solution for large-scale problems. Various scheduling algorithm are used in hybrid and grid computing. The genetic and List scheduling algorithm proved to be efficient.

Multiprocessor Scheduling has been source of challenging problem, the general problem of multiprocessor scheduling can be stated as scheduling as set of partially ordered computational task. In Workflow scheduling, workflow management system defines manage and executes workflows on computing resources. In static scheduling which is usefully done at compile time the characteristics of parallel program are before program execution. DA G can be effectively used in static schedulsing. Task scheduling ready to run task to a host based on information into only about the task. It describes CPOP (Critical Path on a Processor), PETS (Performance Effective Task Scheduling) algorithms.

In this paper, we proposed dynamic critical path for effective workflow scheduling. Regardless of the usage of resources afford in grid computing which is not adequate to use. So as a remedy for this, Cloud computing becomes visible in a way that provides on-demand resources to the users, so as to proceed locally available computational power, delivering new computing resources when necessary. The Cloud computing environment provides resources dynamically whenever demanded by the user. The resource is utilized by the user without having enough knowledge about the technical details involved in the resource provider. We see Cloud Computing as a computing model, not a technology. In this model "customers" plug into the "cloud" to access IT resources which are priced and provided "on-demand". Over the last several years, virtual machines have become a standard object used.

Virtualization further enhances elasticity because it abstracts the hardware to the point where software stacks can be deployed and redeployed without being tied to a specific physical server. Virtualization provides a dynamic datacenter where servers provide a pool of resources that are united as needed, and where the relationship of applications to compute, storage, and network resources changes dynamically in order to meet both workload and business demands. With application deployment decoupled from server deployment, applications can be deployed and scaled rapidly, without having to first procure physical servers.

Virtual machines have become the prevalent abstraction and unit of deployment because they are the least-common denominator interface between service providers and developers. Using virtual machines as objects used is adequate for 80 percent of usage, and it helps to satisfy the need to rapidly deploy and scale applications. Virtual appliances, virtual machines that include software that is moderately or fully configured to perform a specific task such as a Web or database server, further develop the ability to create and deploy applications swiftly. The combination of virtual machines and appliances as standard deployment objects is one of the key features of cloud computing.

Adjusting the runtime of the job to cover both the time between the submission of the job and the desired reservation start time, and the duration of the reservation itself. Unlike scheduler-based reservations, probabilistic reservations do not require special support from resource providers. However, probabilistic reservations are not guaranteed because the actual queue delay may exceed the predicted delay, and the final cost of a probabilistic reservation is difficult to predict because the actual

2. Related Research

2.1. Auditing to Keep Online Storage Services Honest

Third-party auditing is an accepted method for establishing trust between two parties with potentially different incentives. Auditors assess and expose risk, enabling customers to choose rationally between competing services. Over time, a system that includes auditors reduces risk for customers: when combined with a penalty or incentive mechanism, auditing gives incentives to providers to improve their services. Penalty and incentive mechanisms become supportable when risks are well understood.

Auditing of OSPs is not feasible yet. First, customers are not yet sophisticated enough to demand risk assessment. Second, OSPs do not yet provide support for third party audits.

One way is to rely on a trusted third-party auditor, who has sufficient access to the provider's environment. An auditor understands the service level agreement (SLA) between a customer and a provider and quantifies the extent to which a provider might not meet the SLA. The auditor has expertise and capabilities that the customer does not. Auditors understand the threats posed, know best practices, and have the resources to check for process adherence and service quality. They perform these checks through well-defined interfaces to the service. With proper

safeguards, auditors can investigate providers who serve multiple customers without fear of information leakage.

Providers will not offer auditing interfaces unless there is motivation to do so. Mechanisms to provide such motivation are more likely to be social than technical, but we should keep them in mind when trying to design the system interfaces that support auditing. Generally, these behavior-changing mechanisms either use penalties or incentives, or a combination. For example, regulations (and the associated fines), laws (and the threat of incarceration), or loss of reputation (which can put a provider out of business) are penalty-based mechanisms. Market forces (i.e., the ability to charge a premium for better service), or the need to obtain cost effective insurance, can create incentives.

A growing number of online service providers offer to store customers' photos, email, file system backups, and other digital assets. Currently, customers cannot make informed decisions about the risk of losing data stored with any particular service provider, reducing their incentive to rely on these services. It argue that third party auditing is important in creating an online service oriented economy, because it allows customers to evaluate risks, and it increases the efficiency of insurance based risk mitigation.

2.2 PORs: Proofs of Retrievability for Large Files

Improving network bandwidth and reliability are reducing user reliance on local resources. Energy and labor costs as well as computing-system complexity are militating toward the centralized administration of hardware. Increasingly, users employ software and data that reside thousands of miles away on machines that they themselves do not own. Grid computing, the harnessing of disparate machines into a unified computing platform, has played a role in scientific computing for some years. Similarly, software as a service (SaaS)— loosely a throwback to terminal/mainframe computing architectures—is now a pillar in the internet-technology strategies of major companies.

In this work develop a new cryptographic building block known as a proof of retrievability (POR). A POR enables a user (verifier) to determine that an archive (prover) "possesses" a file or data object F. More precisely, a successfully executed POR assures a verifier that the prover presents a protocol interface through which the verifier can retrieve F in its entirety. Of course, a prover can refuse to release F even after successfully participating in a POR. A POR, however, provides the strongest possible assurance of file retrievability barring changes in prover behavior.

It introduce a POR protocol in which the verifier stores only a single cryptographic key—irrespective of the size and number of the files whose retrievability it seeks to verify—as well as a small amount of dynamic state (some tens of bits) for each file. (One simple variant of our protocol allows for the storage of no dynamic state, but yields weaker security.) More strikingly, and somewhat counter intuitively, our scheme requires that the prover access only a small portion of a (large) file F in the course of a POR. In fact, the portion of F "touched" by the prover is essentially independent of the length of F and would, in a typical parameterization, include just hundreds or thousands of data blocks.

Briefly, our POR protocol encrypts F and randomly embeds a set of randomly-valued check blocks called sentinels. The use of encryption here renders the sentinels indistinguishable from other file blocks. The verifier challenges the prover by specifying the positions of a collection of sentinels and asking the prover to return the associated sentinel values. If the prover has modified or deleted a substantial portion of F, then with high probability it will also have suppressed a number of sentinels. It is therefore unlikely to respond correctly to the verifier. To protect against corruption by the prover of a small portion of F, it also employ error-correcting codes. Let F˜ refer to the full, encoded file stored with the prover.

2.3 Provable Data Possession at Untrusted Stores

Verifying the authenticity of data has emerged as a critical issue in storing data on untrusted servers. It arises in peer-to-peer storage systems, network file systems, long-term archives, web-service object stores, and database systems. Such systems prevent storage servers from misrepresenting or modifying data by providing authenticity checks when accessing data. However, archival storage requires guarantees about the authenticity of data on storage, namely that storage servers possess data. It is insufficient to detect that data have been modified or deleted when accessing the data, because it may be too late to recover lost or damaged data. Archival storage servers retain tremendous amounts of data, little of which are accessed. They also hold data for long periods of time during which there may be exposure to data loss from administration errors as the physical implementation of storage evolves, e.g., backup and restore, data migration to new systems, and changing memberships in peer-to-peer systems. Archival network storage presents unique performance demands.

Previous solutions do not meet these requirements for proving data possession. Some schemes provide a weaker guarantee by enforcing storage complexity: The server has to store an amount of data at

least as large as the client's data, but not necessarily the same exact data. Moreover, all previous techniques require the server to access the entire file, which is not feasible when dealing with large amounts of data.

In this work formally define protocols for provable data possession (PDP) that provide probabilistic proof that a third party stores a file. Introduce the first provably-secure and practical PDP schemes that guarantee data possession. Implement one of our PDP schemes and show experimentally that probabilistic possession guarantees make it practical to verify possession of large data sets. Our PDP schemes provide data format independence, which is a relevant feature in practical deployments, and put no restriction on the number of times the client can challenge the server to prove data possession. Also, a variant of our main PDP scheme offers public verifiability.

2.4 Privacy-Preserving Audit and Extraction of Digital Contents

A growing number of online services, such as Amazon, Yahoo!, Google, Snapfish, and Mozy.com, aim to profit by storing and maintaining lots of valuable user data. Example uses of this storage include online backup, email, photo sharing, and video hosting. Many of these service offer a small amount of "teaser" storage for free, and charge for larger, upgraded versions of the service. Studies of deployed large-scale storage systems show that no storage service can be completely reliable; all have the potential to lose or corrupt customer data.

Today, a customer that wants to rely on these services must make an uneducated choice. He has only negative newsworthy anecdotes on which to base his decision, and service popularity or "brand name" is not a positive indicator of reliability. To know if his data is safe, he must either blindly trust the service or laboriously retrieve the hosted data every time he wants to verify its integrity, neither of which is satisfactory. Unfortunately, to date, there are no fair and explicit mechanisms for making these services accountable for data loss.

Our proposed solution to provide storage service accountability is through independent, third party auditing and arbitration. The customer and service enter into an agreement or contract for storing data in which the service provides some type of payment for data loss or failing to return the data intact, e.g. free prints, refunds, or insurance. In such an agreement, the two parties have conflicting incentives. The service provider, whose goal is to make a profit and maintain a reputation, has an incentive to hide data loss. On the other hand, customers are terribly unreliable, e.g. casual home users. Customers can innocently (but incorrectly) or fraudulently claim loss to get paid. Thus, it involves an independent, third party to arbitrate and confirm whether stored and retrieved data is intact.

3. Architecture

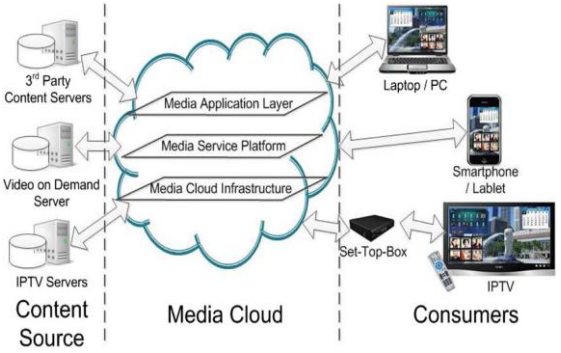

Figure 2

It shows the complete work flow of the proposed system and illustrate that how the system artichecture has been created for detecting the hard queries with cloud media for better performance and produes the quality result for the given query.

4. Implementation and Results

4.1. Algorithm : Find the path

Input: Node, Distance
Output: *Optimize Path For Substation.*
1: Select the initial node; //*The node needs the data or going to send data*
2: **FOR** check the path
3:**IF** evaluate the nearby node avail; // Find the efficient nodes
4: **FOR** show path **DO**
5: **FOR** calculate path distance **DO**
6: **IF** # path is not valid leave it **THEN**
7: Select the path;
8:Path distance is compared with everyone;
9: Fix the effective perfect path;
10:Evaluate the path in every time sending data;
11:**RETURN** *Distance and Perfect Path*; // Solution for issue

The Backup Link Mutual Exclusion Algorithm (BLME Algorithm), which computes the exact path to send data in fast transmission. Here we can get the available link for the fast transmission so we can easily send to next node to get efficient computation. This node division is mainly based on the density of the population over the

mobile media, so the system get collusion toward the transmission. The problems the system get traffic and it will be jammed over a rush time data forwarding. To avoid this kind of operation we need to use the BLME algorithm to define a path if not avail then produce a backup path for the system to resolve the system very efficiently. Here also the distance of the node is been calculated among path defined using this we can choose the better result on fast transmission.

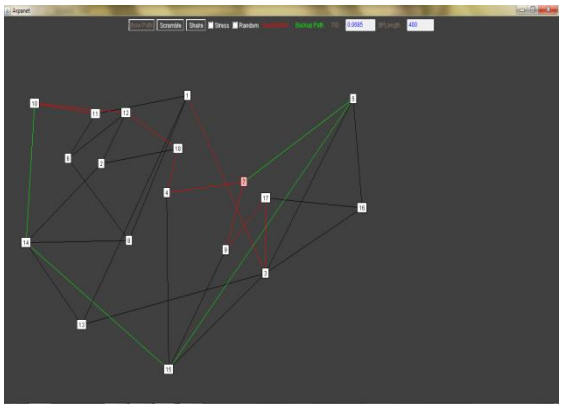

figure 3

This shows find the path between the substations.

4.2. Module Description

Joint Mobile-Cloud Resource Management

The most straightforward way is to execute mobile applications inside the cloud in a cloned virtual machine environment. Clone Cloud, for instance, combines static analysis and dynamic profiling to partition applications automatically at a fine granularity. The system aims to optimize execution time and energy use for a target computation and communication environment.

Similarly, Think Air provides a framework for migrating smart phone applications to the cloud by means of virtualization and method-level computation offloading. A single application is partitioned into multiple components called "weblets". "Weblets" are either executed on the mobile device, or migrated to the cloud. The intelligent decision is based on the status of the device, including CPU load, memory, battery level, network connection quality, and user preferences. The work in further studies the best strategy to choose between mobile execution and cloud execution within an energy-minimized framework.

Backup-Link Mutual Exclusion Algorithm

Solution for the BLME problem is developed using two approaches by formulating the backup path selection as an integer linear program, and developing a polynomial time heuristic based path routing. Integer linear program formulation

The BLME problem is formulated as an Integer Linear Program (ILP) using undirected links. The objective function is set to minimize the sum of the backup path lengths of all links, or equivalently the average backup path length of a link under a single link failure. The average backup path length under single link failure, denoted by H, is computed as:

$$\overline{H} = \left(\frac{1}{|\mathcal{L}} \sum_{\ell,\ell' \in \mathcal{L}} \alpha_{\ell\ell'} \right) - 1.$$

Heuristic Algorithm

The performance of the ILP and heuristic algorithm developed in this paper are evaluated by applying them to six networks. (a) ARPANET; (b) NSFNET;(c) Node-16; (d) Node-28; (e) Mesh-4×4; and (f) NJ-LATA. First, link resources such as conduit or duct are shared by multiple links for ease of layout. Such sharing of resources is typically limited to links that are close to each other, such as adjacent links. Hence, dual-link failure scenarios under such shared resource failure typically affect only nearby links. The second case of dual-link failure scenario is due to the time required to repair a failed link. Before a failed link is repaired, another link in the network could fail; however, such failures are typically rare.

Performance Metrics

The performance metrics considered specifically for the ILP solutions are: 1) solution time and 2) optimality bound. The optimality bound is relevant in scenarios where the ILP could not obtain optimal solution, but has a feasible solution with a known bound on optimality.

A Unified Optimization Framework

On-going works in resource control and management for mobile cloud computing often share a similar mathematical formulation, as a constrained optimization problem. In this subsection we propose to abstract various formulations into a unified optimization framework for resource management and control in mobile cloud media. Specifically, its objective is to minimize the total cost of ownership for a cloud media network including both upfront cost (i.e., CAPEX) and re-

occurring cost (i.e., OPEX). In addition, the optimization is subject to two categories of constraints including:

- *Capacity constraints*
- *QoS/QoE constraints*

5. Conclusion

In the existing work, analyzes the characteristics of cloud mobile media to transfer the data in a fast and very efficient manner but here the security level is been minimized, so the overall data may be corrupted or else it will be stolen by the intruders. However, in this system numbers of issues are there to address. They are, searching quality is lower than the other system and reliability rate of the system is lowest. In order to overcome these drawbacks, we are performing the layered architecture to perform the better result on the security.

This proposed system is well enhancing the reliability rate of fast and secure data transform system. In other words, this work is support these operators for efficient result. From the experimentation result, we are obtaining the proposed system is well effective than the existing system in terms of accuracy rate, quality of result.

6. References

1. Yonggang Wen., Xiaoqing Zhu., Joel J.P.C. Rodrigues. and Chang Wen Chen. (2014), 'Cloud Mobile Media: Reflection and Outlook', IEEE Transactions on Multimedia, Vol. 16, no.4. pp. 885-902

2. Satyanarayanan M., Bahl P., Caceres R. and Davies N. (2009), 'The case for VM-based cloudlets in mobile computing', IEEE Pervasive Comput., Vol. 8, no. 4, pp. 14–23.

3. Wendell P. and Freedman M.J. (2011), 'Going viral: Flash crowds in an open CDN, in Proc. 2011 ACM SIGCOMM Conf. Internet Measurement Conf. (IMC '11), NewYork, NY, USA, pp. 549–558.

4. Endo P., Gonçalves G., Kelner J. and Sadok D. (2010), 'A survey on opensource cloud computing solutions,' in Proc. VIII Workshop em Clouds Grids e Aplicações, pp. 3–16.

5. Hua X.-S., Hua G. and Chen C.W. (2010), 'ACM workshop on mobile cloud media computing,' in Proc. ACM MCMC'10, Firenze, Italy.

6. Tan M. and Su X. (2011), 'Media cloud: When media revolution meets rise of cloud computing,' in Proc. 6th IEEE Int. Symp. Service Oriented System Engineering (SOSE).

7. Zhu W., Luo C., Wang J. and Li S. (2011), 'Multimedia cloud computing,' IEEE Signal Process. Mag., Vol. 28, no. 3, pp. 59–69.

8. Dey S. (2012), 'Cloud mobile media: Opportunities, challenges, and directions,' in Proc. Int. Conf. Computing, Networking and Communications (ICNC) pp. 929–933.

9. Hoelzle U. and Barroso L.A. (2009), 'The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines', 1st ed. SanRafael, CA, USA: Morgan and Claypool.

10. Fernando, Niroshinie, Loke, Seng W., Rahayu and Wenny (2013), 'Mobile cloud computing: A survey', Future Gener. Comput. Syst., Vol. 29, no.1, pp. 84–106.

