ISSN 2348 – 7968

A Survey on Efficient Privacy Preserving and Increased Quality of Service (QoS) Parameters in Mobile Cloud

J.Arul Keerthi Dhaya[1] , S. Raja Mohamed[2]

[1]Computer Science Engineering, Kalaignar Karunanidhi Institute of Technology, Coimbatore , Tamil Nadu , India.
[2]Computer Science Engineering, Kalaignar Karunanidhi Institute of Technology, Coimbatore , Tamil Nadu , India.

Abstract

In MANETs, how to conserve the limited power resources of sensors to extend the network lifetime of the network as long as possible while performing the sensing and sensed data reporting tasks, is the most critical issue in the network design. In many applications, the node closer to the sink are over burdened with huge traffic load as the data from the entire region are forwarded through them to reach to the sink.
In this Paper, we proposed the clustering and routing algorithm for cluster path relocation. By cluster path relocation we can form the larger. By using the Parallel Machine intelligence the actor nodes are created and it is used for the path selection during the process of communication.

Keywords: *MANET, Cluster and routing algorithm , Parallel Machine intelligence.*

1.Introduction

Clouds are large-scale distributed computing systems built around core concepts such as computing as utility, virtualization of resources, on demand access to computing resources, and outsourcing computing services.

Personal mobile devices have gained enormous popularity in recent years. Due to their limited resources (e.g., computation, memory, energy), however, executing sophisticated applications (e.g., video and image storage and processing, or map-reduce type) on mobile devices remains challenging. As a result, many applications rely on offloading all or part of their w orks to “remote servers” such as clouds and peer mobile devices. For instance, applications such as Google Goggle and Siri process the locally collected data on clouds. Wireless sensor networks often consists of a large number of low-cost sensor nodes that have strictly limited sensing, computation, and communication capabilities. Due to resource restricted sensor nodes, it is important to minimize the amount of data transmission so that the average sensor lifetime and the overall bandwidth utilization are improved. Data aggregation is the process of summarizing and combining sensor data in order to reduce the amount of data transmission in the network. As wireless sensor networks are usually deployed in remote and hostile environments to transmit sensitive information.

Cloud computing is a new paradigm in which computing resources such as processing, memory, and storage are not physically present at the user’s location. Instead, a service provider owns and manages these resources, and users access them via the Internet. For example, Amazon Web Services lets users store personal data via its Simple Storage Service (S3) and perform computations on stored data using the Elastic Compute Cloud (EC2). This type of computing provides many advantages for businesses—including low initial capital investment, shorter start-up time for new services, lower maintenance and operation costs, higher utilization through virtualization, and easier disaster recovery—that make cloud computing an attractive option. Reports suggest that there are several benefits in shifting computing from the desktop to the cloud.1,2 What about cloud computing for mobile users? The primary constraints for mobile computing are limited energy and wireless bandwidth. Cloud computing can provide energy savings as a service to mobile users, though it also poses some unique challenges. This dissertation deals with problems in reliability and lifetime data analysis. The rest part focuses on the study of graphical estimators from probability plots with right censored data. The second part deals with reliability inference for repairable systems.

Mobile cloud computing is the next big thing. Recent market research predicts that by the end of 2014 mobile cloud
applications will deliver annual revenues of 20 billion dollars. Although it is hard to validate precise predictions, this is hardly implausible: mobile devices as simple as phones and as complex as mobile Internet devices with various network connections, strong connectivity especially in developed areas, camera(s), GPS, and other sensors are the current computing wave, competing heavily with desktops and laptops for market and popularity. Connectivity offers immediate access to available computing, storage, and communications on commercial clouds, at nearby wireless hot-spots equipped with

computational resources, or at the user's PC and plugged-in laptop.
Smart phones are becoming increasingly popular, with approximately 550,000 new Android devices being activated
World wide every day 1. These devices have a wide range of capabilities, typically including GPS, WiFi, cameras, gigabytes of storage, and gigahertz-speed processors. As a result, developers are building ever more complex smart phone applications such as gaming, navigation, video editing, augmented reality, and speech recognition, which require considerable computational power and energy. Unfortunately, as applications become more complex, mobile users have to continually upgrade their hardware to keep pace with increasing performance requirements but still experience short battery lifetime.

2.Related Works

2.1 In [1] Byung-Gon Chun Presented a mobile device delivers a small Virtual Machine (VM) overlay to a cloudlet infrastructure and lets it take over the computation. Similar works that use VM migration are also done in *CloneCloud.* Our system overcomes design and implementation challenges to achieve basic augmented execution of mobile applications on the cloud, representing the whole-sale transfer of control from the device to the clone and back. We combine partitioning, migration with merging, and on-demand instantiation of partitioning to address these challenges. Our prototype delivers up to 20x speedup and 20x energy reduction for the simple applications we tested, without programmer involvement, demonstrating feasibility for the approach, and opening up a path for a rich research agenda in hybrid mobile-cloud systems.

2.2 In [2] Sokol Kosta presents a ThinkAir, framework for offloading mobile computation to the cloud. a mobile device delivers a small Virtual Machine (VM) overlay to a cloudlet infrastructure and lets it take over the computation. Similar works that use VM migration are also done in *ThinkAir*. Using Think Air requires only simple modifications to an application's source code coupled with use of our ThinkAir tool-chain. Experiments and evaluations with micro benchmarks and computation intensive applications demonstrate the benefits of ThinkAir for profiling and code offloading, as well as accommodating changing computational requirements with the ability of on-demand VM resource scaling and exploiting parallelism. Key components of ThinkAir: we have ported Android to Xen allowing it to be run on commercial cloud infrastructure, and we continue to work on improving programmer support for parallelizable applications.

2.3 In [3] M. Satyanarayanan, Cloud computing in a small-scale network with battery-powered devices has also gained attention recently. *Cloudlet* is a resource-rich cluster that is well connected to the Internet and is available for use by nearby mobile devices we can apply synthesis recursively to generate a family of overlays. Creating a launch VM would then involve pipelined application of these overlays, with intermediate results cached for reuse. Earlier stages of the pipeline tend to involve larger overlays that are more widely used across applications and are hence more likely to be found in a persistent cache. Conceptually, we seek a "wavelet"-like decomposition of VM state into a sequence of overlays that decrease in size but increase in specificity. A trade-off is that each overlay introduces some delay in pre-use infrastructure customization. The cost of generating overlays isn't a factor because it occurs offline. Another deployment challenge relates to the assumption that a relatively small set of base VMs will suffice for a large range of applications. A mobile device with an overlay generated from a base VM that's too old might not be able to find a compatible cloudlet. This problem could be exacerbated by the common practice of releasing security patches for old OS releases. Although the patch's effect could be incorporated into the overlay, it would increase overlay size. A different approach would be to trigger generation of new overlays when security patches are released, which mobile devices would then have to download.

2.4 In [4] Dimakis et al. proposed several erasure coding algorithms for maintaining a distributed storage system in a dynamic network. The results about the problem of reducing repair traffic in distributed storage systems based on erasure coding. Three versions of the repair problems are considered: *exact repair*, *functional repair* and *exact repair of systematic parts*. In the exact repair model, the lost content is exactly regenerated; in the functional repair model, only the same MDS-code property is maintained before and after repairing; in the exact repair of systematic parts, the systematic part is exactly reconstructed but the non-systematic part follows a functional repair model. Finally small finite-field constructions require further investigation. While many of the constructions presented require a large finite-field size, practical storage systems would benefit from efficient binary operations. Recently Zhang et al. suggested a scheme for repairing Even odd code, which are binary codes with $n = k+2$. While the proposed scheme does not match the cut-set bound it improves on the naive repairing method of reconstructing all the data blocks. Constructing regenerating codes for small finite fields or designing repair algorithms for existing codes will be of significant practical interest.

2.5 In [5] Cong Shi presents a MAUI determines which processes to be offloaded to remote servers based on their CPU usages. Serendipity considers using remote computational resource from other mobile devices. this paper we have developed and evaluated the Serendipity system that enables a mobile device to remotely access computational resources on other mobile devices it may encounter. The main challenge we addressed is how to model computational tasks and how to perform task allocation under varying assumptions about the connectivity environment. Through an emulation of the Serendipity system we have explored how such a system has the potential to improve computation speed as well as save energy for the initiating mobile device. We have also reported on a preliminary prototype of our system on Android platforms.

2.6 In [6] Aguilera et al. proposed a protocol to efficiently adopt erasure code for better reliability Erasure codes are powerful alternatives to replication for storage, a they provide better space efficiency and _ner control over the redundancy level. However, they create complications due to complexity and cohesion of data, especially with concurrent updates and failures. Here, we propose a new protocol to address these complications. The protocol has features to make it broadly applicable, and its efficiency is reasonable as demonstrated by experiments. Our protocol allows the use of highly-efficient erasure codes with large n and k, and small n - k.

2.7 In [7] Chien-An Chen, Despite the advances in hardware for hand-held mobile devices, resource-intensive applications (e.g., video and image storage and processing or map-reduce type) still remain off bounds since they require large computation and storage capabilities. Recent research has attempted to address these issues by employing remote servers, such as clouds and peer mobile devices. For mobile devices deployed in dynamic networks (i.e., with frequent topology changes because of node failure/unavailability and mobility as in a mobile cloud), however, challenges of reliability and energy efficiency remain largely unaddressed. To the best of our knowledge, we are the first to address these challenges in an integrated manner for both data storage and processing in mobile cloud, an approach we call *k-out-of-n computing*. In our solution, mobile devices successfully retrieve or process data, in the most energy-efficient way, as long as k out of n remote servers are accessible. Through a real system implementation we prove the feasibility of our approach. Extensive simulations demonstrate the fault tolerance and energy efficiency performance of our framework in larger scale networks. We presented the first *k*-out-of-*n* framework that jointly addresses the energy-efficiency and fault-tolerance challenges. 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. Through system implementation, the feasibility of our solution on real hardware was validated. Extensive simulations in larger scale networks proved the effectiveness of our solution.

Conclusions

We presented the first *k*-out-of-*n* framework that jointly addresses the energy-efficiency and fault-tolerance challenges. 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. Through system implementation, the feasibility of our solution on real hardware was validated. Extensive simulations in larger scale networks proved the effectiveness of our solution.

References

[1] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, “CloneCloud: elastic execution between mobile device and cloud,” in Proc. of EuroSys, 2011.

[2] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, “ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading,” in Proc. Of INFOCOM, 2012.

[3] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, “The case for VM-based cloudlets in mobile computing,” *Pervasive Computing, IEEE*, vol. 8, pp. 14–23, 2009.

[4] A. G. Dimakis, K. Ramchandran, Y. Wu, and C. Su, “A survey on network codes for distributed storage,” Proc. Of the IEEE, vol. 99, no. 3, pp. 476–489, 2010.

[5] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, “Serendipity: enabling remote computing among intermit- tently connected mobile devices,” in *Proc. of MobiHoc*, 2012.

[6] M. Aguilera, R. Janakiraman, and L. Xu, “Using erasure codes efficiently for storage in a distributed system,” in *Proc. of DSN*, 2005.

[7] Chien-An Chen, Myounggyu Won, Radu Stoleru, and Geoffrey G. Xie , “Energy-Efficient Fault-Tolerant Data Storage &
Processing in Mobile Cloud,” Proc. Of the IEEE, vol. 3, no. 1, pp. 476–489, 2014.