ISSN : 2394-2975 (Online)
ISSN : 2394-6814 (Print)

Energy-Efficient Fault-Tolerant Data Storage and Processing in Cloud Environments

[I]R.Saranya, [II]V.P.Muthukumar
[I]M.Phil Full-time Research Scholar, [II]M.Sc.,M.Phil., Assistant Professor
[I,II]Dept. of Computer Science and Applications, Vivekanandha College of Arts and Sciences for Women (Autonomous), Elayampalayam, Tiruchengode, Tamil Nadu, India.

Abstract
In 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 because it need to maintain power consumption. Due to lacking of power in cloud computing of mobile devices, if any node becomes failure then, entire network of mobile communication may disordered. In this 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 the method proves the feasibility of the proposed approach. Extensive simulations are demonstrated with fault tolerance and energy efficiency performance of the framework in larger scale networks The integrate k-out-of-n reliability mechanism into distributed computing in mobile cloud formed by only mobile devices. K-out-of-n, a well-studied topic in reliability control, ensures that a system of 'n' components operates correctly as long as k or more components work. More specifically, we investigate how to store data as well as process the stored data in mobile cloud with k-out of-n reliability such that: 1) The energy consumption for retrieving distributed data is minimized; 2) The energy consumption for processing the distributed data is minimized and 3) Data and processing are distributed considering dynamic topology changes.

Key words
Personal mobile devices, power consumption, fault tolerances, M-K-Out-Data Allocation, M-K-out-Data Processing

I. Objectives

- Under utilization: Typical deployments have very low utilization of total computing capacity. Users want to run only a few applications per computer to obtain better response time. As a result, many computers are under-utilized.
- Security: Desktops are of ten managed by individual users and they have to regularly apply security patches. Otherwise, the computers can become vulnerable.
- Operational costs: The total cost of ownership can grow rapidly for supporting increasing numbers of desktops and laptops, and for upgrading and up- dating software. Moreover, these computers may waste power as they are often kept on 24 h.
- The objective of this problem is to find n nodes in V as process n nodes such that energy consumption for processing a job of M tasks is minimized.

II. Problem Definition

The existing presents a mathematical model for both optimizing energy consumption and meeting the fault tolerance requirements of data storage and processing under a dynamic network topology. This paper presents an efficient algorithm for estimating the communication cost in a mobile cloud, where nodes fail or move, joining/leaving the network.

This work proposed a first process scheduling algorithm that is both fault-tolerant and energy efficient. It presents a distributed protocol for continually monitoring the network topology, without requiring additional packet transmissions. The evaluation of this proposed framework processed through a real hardware implementation and large scale simulations.

The framework, running on all mobile nodes, provides services to applications that aim to: (1) store data in mobile cloud reliably such that the energy consumption for retrieving the data is minimized (k-out-of-n data allocation problem); and (2) reliably process the stored data such that energy consumption for processing the data is minimized (k-out-of-n data processing problem).

The proposed work includes all the existing system approach which covers multiple cloud service provider environments with different execution cost for same task. In general, each node may have different energy cost depending on their energy sources; e.g., nodes attached to a constant energy source may have zero energy cost while nodes powered by battery may have relatively high energy cost.

The applications generate data and our framework stores data in the network. For higher data reliability and availability, each data is encoded and partitioned into fragments; the fragments are distributed to a set of storage nodes. In order to process the data, applications provide functions that take the stored data as inputs.

Each function is instantiated as multiple tasks that process the data simultaneously on different nodes. Nodes executing tasks are processor nodes; we call a set of tasks instantiated from one function a job. Client nodes are the nodes requesting data allocation or processing operations. A node can have any combination of roles from: storage node, processor node, or client node, and any node can retrieve data from storage nodes. It considered Topology Discovery and Monitoring, Failure Probability Estimation, Expected Transmission Time (ETT) Computation, k-out-of-n Data Allocation and k-out-of-n Data Processing.

In general, each node may have different energy cost depending on their energy sources; e.g., nodes attached to a constant energy source may have zero energy cost while nodes powered by battery may have relatively high energy cost. For simplicity, we assume the network is homogeneous and nodes consume the same amount of energy for processing the same task. As a result, only the transmission energy affects the energy efficiency of the final solution. We leave the modeling of the general case as future work

In existing work, it is assumed the network is homogeneous and nodes consume the same amount of energy for processing the same task. As a result, only the transmission energy affects the energy efficiency of the final solution. But in proposed system, different execution cost for same task scenario is considered which resembles the real time case.

III. Review of Literature

A. Related Works

Clone Cloud: Elastic Execution Bettheyen Mobile Device And Cloud

Authors:
BYUNG-GON CHUN, SUNGHWAN IHM , PETROS MANIATIS, MAYUR NAIK, ASHWIN PATTI

In this paper, the authors explained that mobile applications are becoming increasingly ubiquitous and provide ever richer functionality on mobile devices. At the same time, such devices often enjoy strong connectivity with more powerful machines ranging from laptops and desktops to commercial clouds. This paper presents the design and implementation of Clone Cloud, a system that automatically transforms mobile applications to benefit from the cloud. The system is a flexible application partitioner and execution runtime that enables unmodified mobile applications running in an application-level virtual machine to seamlessly off-load part of their execution from mobile devices onto device clones operating in a computational cloud.

Serendipity: Enabling Remote Computing Among Intermittently Connected Mobile Devices

Authors
CONG SHI, VASILEIOS LAKAFOSIS, MOSTAFA H. AMMAR, ELLEN WZEGURA

Mobile devices are increasingly being relied on for services that go beyond simple connectivity and require more complex processing. Fortunately, a mobile device encounters, possibly intermittently, many entities capable of lending it computational resources. At one extreme is the traditional cloud-computing context where a mobile device is connected to remote cloud resources maintained by a service provider with which it has an established relationship. In this paper they consider the other extreme, where a mobile device's contacts are only with other mobile devices, where both the computation initiator and the remote computational resources are mobile, and where intermittent connectivity among these entities is the norm.

As mentioned previously they envision Serendipity as developed here to enable an extreme of a spectrum of remote computation possibilities that are available to mobile devices. Their future work will consider extending their investigation to enable hybrid remote computation where the use of cloud or cloudlet resources is augmented with the use of resources on other mobile devices.

A Survey on Network Codes For Distributed Storage

Authors:
ALEXANDROSG.DIMAKISKANNAN,RAMCHANDRAN,Y UNNANWU, HANGHOSUH,

Distributed storage systems often introduce redundancy to increase reliability. When coding is used, the repair problem arises: if a node storing encoded information fails, in order to maintain the same level of reliability they need to create encoded information at a new node. This amounts to a partial recovery of the code, whereas conventional erasure coding focuses on the complete recovery of the information from a subset of encoded packets. The consideration of the repair network traffic gives rise to new design challenges. Recently, network coding techniques have been instrumental in addressing these challenges, establishing that maintenance bandwidth can be reduced by orders of magnitude compared to standard erasure codes. This paper provides an overview of the research results on this topic.

The issues of security and privacy are important for distributed storage. When coding is used, errors can be propagated in several mixed blocks through the repair process and an error-control mechanism is required. A related issue is that of privacy of the data by information leakage to eavesdroppers during repairs.

Network-Aware Service Placement In A Distributed Cloud Environment

Authors:
MORITZ STEINER, BOB GAGLIANELLO, VIJAY GURBANI, VOLKER HILT, W. D. ROOME, MICHAEL SCHARF, AND THOMAS VOITH

They consider a system of compute and storage resources geographically distributed over a large number of locations connected via a wide-area network. By distributing the resources, latency to users can be decreased, bandwidth costs reduced and availablility in-creased. The challenge is to distribute services with varying characteristics among the data centers optimally. Some services are very latency sensitive, others need vast amounts of storage, and yet others are computationally complex but do not require hard deadlines on execution. They propose efficient algorithms for the placement of services to get the maximum benefit from a distributed cloud systems. The algorithms need input on the status of the network, compute resources and data resources, which are matched to application requirements.

our demonstration shows the tight coupling of the network with a distributed cloud solution using a system that integrates several of their own software components addressing individual challenges. They demonstrate that in case of network or data center problems migrating a service to another data center can mitigate the performance issues.

IV. Energy Architecture Model

The framework, running on all mobile nodes, provides services to applications that aim to:

- Store data in mobile cloud reliably such that the energy consumption for retrieving the data is minimized (k-out-of-n data allocation problem);
- Reliably process the stored data such that energy consumption for processing the data is minimized (k-out-of-n data processing problem).

The application running in a mobile ad-hoc network may generate a large amount of media files and these files must be stored reliably such that they are recoverable even if certain nodes fail. At later time, the application may make queries to files for information such as the number of times an object appears in a set of images. Without loss of generality, we assume a data object is stored once, but will be retrieved or accessed for processing multiple times later. They first define several terms. As shown in Fig. 1, applications generate data and our framework stores data in the network. For higher data reliability and availability, each data is encoded and partitioned into fragments; the fragments are distributed to a set of storage nodes. In order to process the data, applications provide functions that take the stored data as inputs. Each function is

ISSN : 2394-2975 (Online)
ISSN : 2394-6814 (Print)

instantiated as multiple tasks that process the data simultaneously on different nodes. Nodes executing tasks are processor nodes; we call a set of tasks instantiated from one function a job.

Client nodes are the nodes requesting data allocation or processing operations. A node can have any combination of roles from: storage node, processor node, or client node, and any node can retrieve data from storage nodes. As shown in Fig. 1, our framework consists of five components: Topology Discovery and Monitoring, Failure Probability Estimation, Expected Transmission Time (ETT) Computation, k-out-of-n Data Allocation and k-out-of-n Data Processing. When a request for data allocation or processing is received from applications, the Topology Discovery and Monitoring component provides network topology information and failure probabilities of nodes.

The failure probability is estimated by the Failure Probability component on each node. Based on the retrieved failure probabilities and network topology, the ETT Computation component computes the ETT matrix, which represents the expected energy consumption for communication between any pair of node. Given the ETT matrix, our framework finds the locations for storing fragments or executing tasks.

The k-out-of-n Data Storage component partitions data into n fragments by an erasure code algorithm and stores these fragments in the network such that the energy consumption for retrieving k fragments by any node is minimized. K is the minimal number of fragments required to recover a data. If an application needs to process the data, the k-outof-n Data Processing component creates a job of M tasks and schedules the tasks on n processor nodes such that the energy consumption for retrieving and processing these data is minimized. This component ensures that all tasks complete as long as k or more processor nodes finish their assigned tasks.

The Topology Discovery and Monitoring component continuously monitors the network for any significant change of the network topology. It starts the Topology Discovery when necessary.

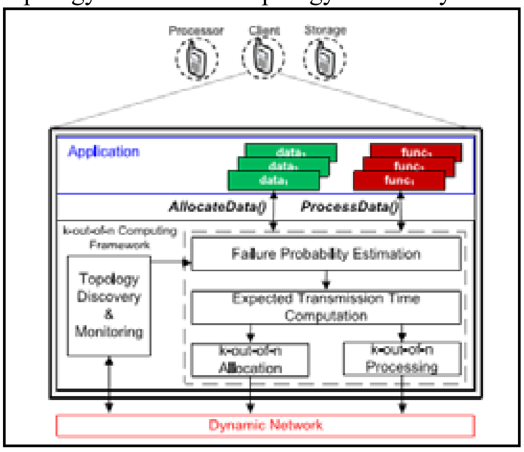


Fig. 1 : Overview Of Our Proposed Framework

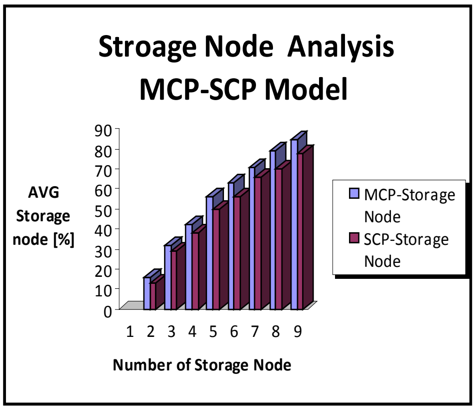
V. Experimental Results

The **Table 1.1** represents experimental result for Single K-Out-N-Algorithm and M-K-Out-N-Algorithm model. The table contains finding Data storage node count in K-Out-N and M-K-Out-N- Algorithm model within 1000 sec time interval details as shown.

Table 1.1 : Selection of Storage Node Analysis

S.NO	Number of Data Storage node (Count)	M-K-OUT-N-ALGORITHM		K-OUT-N-ALGORITHM	
		MCP-Storage Node	Selection of Storage Node	SCP-Storage Node	Selection of Storage Node
1	100	16	22	13	17
2	200	32	35	29	30
3	300	42	49	38	44
4	400	56	62	50	56
5	500	63	72	56	65
6	600	71	83	66	80
7	700	79	89	70	85
8	800	85	92	78	86

The **Figure 1.1** represents experimental result for Single K-Out-N-Algorithm and M-K-Out-N-Algorithm model. The figure contains finding Data storage node count in K-Out-N and M-K-Out-N- Algorithm model within 1000 sce time interval details as shown.



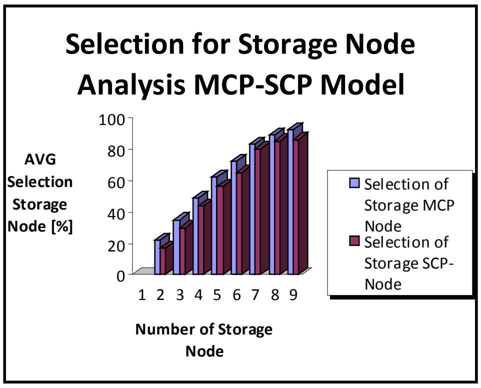


Fig. 1.1 : Selection Storage Node MCP-SCP Model

The **Table 1.2** represents experimental result for Single K-Out-N-Algorithm and M-K-Out-N-Algorithm mode time analysis. The table contains Data storage node allocation time analysis for K-Out-N and M-K-Out-N- Algorithm model within 1000 sec time interval details as shown.

Table 1.2 : Data storage Node- Time Analysis

S. No	Data Storage Node [n]	K-OUT-N [sec]	M-K-OUT-N [sec]
1	30	20	25
2	40	30	35
3	50	40	45
4	60	50	55
5	70	60	65
6	80	70	75
7	90	80	85
8	100	90	95

The **Fig 1.2** represents experimental result for Single K-Out-N-Algorithm and M-K-Out-N-Algorithm mode time analysis. The figure contains Data storage node allocation time analysis for K-Out-N and M-K-Out-N- Algorithm model within 1000 sec time interval details as shown.

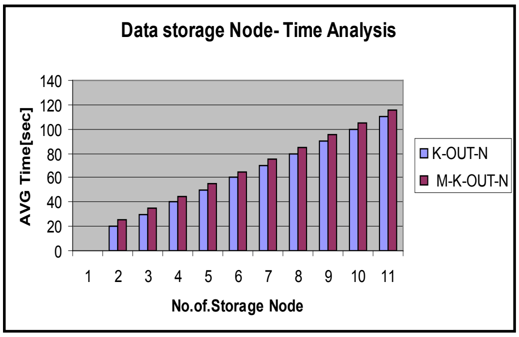


Fig 1.2 : Data storage Node- Time Analysis

VI. Conclusion and Future Works

In this thesis work presented the first k-out-of-n framework that jointly addresses the energy-efficiency and fault-tolerance problem overcome. 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 this thesis solution.

In this 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 the method proves the feasibility of the proposed approach. Extensive simulations are demonstrated with fault tolerance and energy efficiency performance of the framework in larger scale networks with multi cloud provider.

In the future, to utilize the inferred information and extend the framework for efficient and effective Cloud services monitoring and application design. The new system become useful if the below enhancements are made in future.

- The application can be web service oriented so that it can be further developed in any platform.
- The application if developed as web site can be used from anywhere.
- The algorithm can be further improved so that cost of the path can be further reduced
- The energy updation automatically reconfigured the storage node
- Currently the scheme has a slightly less memory overhead, while in the more complex applications; the scheme may utilize more memory. The future study can be in the area of more significant memory savings.

The new system is designed such that those enhancements can be integrated with current modules easily with less integration work. The new system becomes useful if the above enhancements are made in future. The new system is designed such that those enhancements can be integrated with current modules easily with less integration work.

References

[1]. *[Cuervo 2010] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl. MAUI: Making smart phones last longer with code offload. In MobiSys, 2010.*

[2]. *[Tolia 2006] N. Tolia, M. Kaminsky, D. G. Andersen, and S. Patil. An architecture for Internet data transfer. In NSDI, 2006.*

[3]. *L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. Mao, and L. Yang. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In Proc. Int. Conf. Hardware/Software Codesign and System Synthesis, 2010.*

[4]. *A. Arora et al., "ExScal: Elements of an Extreme Scale Wireless Sensor Network," Proc. 11th IEEE Int'l. Conf. Embedded Real-Time Comp. Sys. Apps., Aug. 2005, pp. 102–8.*

[5]. *T. He et al., "VigilNet: An Integrated Sensor Network System for Energy-Efficient Surveillance," ACM Trans. Sensor Net., vol. 2, no. 1, 2006, pp. 1–38.*

[6]. *A. Wood et al., "Context-Aware Wireless Sensor Networks for Assisted Living and Residential Monitoring," IEEE Network, vol. 22, no. 4, July 2008, pp. 26–33.*

[7]. *Dept. Homeland Security, National Incident Management System, Mar. 2004.*

[8]. *A. Jiang, BNetwork coding for joint storage and transmission with minimum cost,[in Proc. Int. Symp. Inf. Theory, Jul. 2006, pp. 1359–1363.*

[9]. *S.-Y. R. Li, R. W. Yeung, and N. Cai, BLinear network coding,[IEEE Trans. Inf. Theory, vol. 49, no. 2, pp. 371–381, Feb. 2003.*

[10]. *A. G. Dimakis, P. G. Godfrey, Y. Wu, M. J. Wainwright, and K. Ramchandran, BNetwork coding for distributed storage systems,[IEEE Trans. Inf. Theory, vol. 56, no. 9, pp. 4539–4551, Sep. 2010.*