Scientific Journal Impact Factor (SJIF) : 3.518
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

Mobile Cloud-Data Processing And Energy Efficient Fault Tolerant Data storage

Sandip Kadam[1], Kanchan Doke[2] ,Vijaya Chavan[3]
[1,2,3]*Computer Engineering, BVCOENM*

Abstract— Smartphones have gained enormous popularity over the past few years. Resource-ntensive 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. The first to address these challenges in an integrated manner for both data storage and processing in mobile cloud, an approach is call k-out-of-n computing.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 it proves the feasibility of this approach. Extensive simulations demonstrate the fault tolerance and energy efficiency performance of this framework in larger scale networks. Keywords— Smartphones , Cloud ,Mobile Phone ,Mobile Cloud ,Real System

Keywords— Smartphones , Cloud ,Mobile Phone ,Mobile Cloud ,Real System

I. INTRODUCTION

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 works 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. Going beyond the traditional cloud-based scheme, recent research has proposed to offload processes on mobile devices by migrating a Virtual Machine (VM) overlay to nearby [1], [2], [3] infrastructures.This strategy essentially allows offloading any process or application, but it requires a complicated VM mechanism and a stable network connection. Some systems (e.g., Serendipity) even leverage peer mobile devices.In dynamic networks, e.g., mobile cloud for disaster response or military operations [5], when selecting remote servers, energy consumption for accessing them must be minimized while taking into account the dynamically changing topology.Serendipity and other VM based solutions considered the energy cost for processing a task on mobile devices and offloading a task to the remote servers, but they did not consider the scenario in a multi-hop and dynamic network where the energy cost for relaying transmitting packets is significant. Furthermore, remote servers are often inaccessible because of node failures, unstable links, or node-mobility, raising a reliability issue. Although Serendipity considers intermittent connections, node failures are not taken into account; the VM-based solution considers only static networks and is difficult to deploy in dynamic environments.In this system, propose the first framework to support fault-tolerant and energy-efficient remote storage and processing under a dynamic network topology, i.e., mobile cloud. This framework aims for applications that require energy efficient and reliable distributed data storage and processing in dynamic network. For example, military operation or disaster response.it is integrate the 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.In this framework, a data object is encoded and partitioned into n fragments, and then stored on n different nodes. As long as k or more of the n nodes are available, the data object can be successfully recovered. Similarly, another set of n nodes are assigned tasks for processing the stored data and all tasks can be completed as long as k or more of the n processing nodes finish the assigned tasks. The parameters k and n determine the degree of reliability and different (k, n) pairs may be assigned to data storage and data processing. System administrators select these parameters based on their reliability requirements.

In this solution following are some important concept:

- It 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.
- It presents an efficient algorithm for estimating the communication cost in a mobile cloud, where nodes fail or move, joining/leaving the network.
- It presents the first process scheduling algorithm that is both fault-tolerant and energy efficient.
- It presents a distributed protocol for continually monitoring the network topology.

II. RELATED WORKS

Recent research has proposed to offload processes on mobile devices by migrating a Virtual Machine (VM) overlay to nearby infrastructures recent research has proposed to offload processes on mobile devices by migrating a Virtual Machine (VM) overlay to nearby infrastructures. In dynamic networks, e.g., mobile cloud for disaster response or military operations [5], when selecting remote servers, energy consumption for accessing them must be minimized while taking into account the dynamically changing topology. Serendipity and other VM based solutions considered the energy cost for processing a task on mobile devices and offloading a task to the remote servers, but they did not consider the scenario in multi-hop and dynamic network where the energy cost for relaying transmitting packets is significant.

Furthermore, remote servers are often inaccessible because of node failures, unstable links, or node-mobility, raising a reliability issue. Although Serendipity considers intermittent connections, node failures are not taken into account, the VM-based solution considers only static networks and is difficult to deploy in dynamic environments. Think Air exploits the concept of smartphone virtualization in the cloud and provides method-level computation offloading. Advancing on previous work, it focuses on the elasticity and scalability of the cloud and enhances the power of mobile cloud computing by parallelizing method execution using multiple virtual machine (VM) images. 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. It present the design and implementation of a system, Serendipity, that enables a mobile computation initiator to use remote computational resources available in other mobile systems in its environment to speedup computing and conserve energy. It propose a simple but powerful job structure that is suitable for such a system. Serendipity relies on the collaboration among mobile devices for task allocation and task progress monitoring functions. Disaster responders require timely delivery of high volumes of accurate data to make correct decisions. To meet these needs, Distress Net, an ad hoc wireless architecture that supports disaster response with distributed collaborative sensing, topology-aware routing using a multichannel protocol, and accurate resource localization.

III. METHODOLOGY

A. ARCHITECTURE AND FORMULATIONS:

An overview of this framework is depicted in Fig.1. 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).As an example, an 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, will assume a data object is stored once, but will be retrieved or accessed for processing multiple times later.In this 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 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:

- K Topology Discovery and Monitoring,
- Failure Probability Estimation,
- Expected Transmission Time (ETT) Computation,
- k-out-of-n Data AllocationAnd k-out-of-n Data Processing.

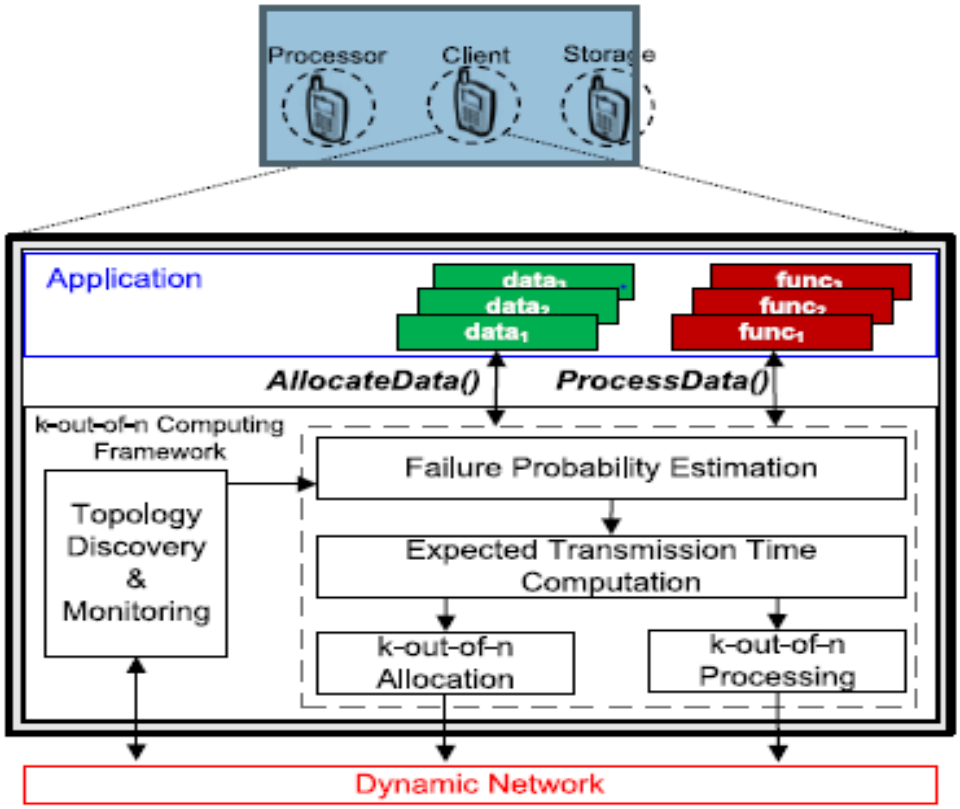

Fig. 1. Architecture for integrating the k-out-of-n computing framework for energy efficiency and fault-tolerance.

The framework is running on all nodes and it provides data storage and data processing services to applications, 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-out-of- 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.

B. ENERGY EFFICIENT AND FAULT TOLERANT DATA ALLOCATION AND ROCESSING

Topology Discovery

Topology Discovery is executed during the network initialization phase or whenever a significant change of the network topology is detected (as detected by the Topology Monitoring component). During Topology Discovery, one delegated node floods a request packet throughout the network. Upon receiving the request packet, nodes reply with their neighbor tables and failure probabilities. Consequently, the delegated node obtains global connectivity information and failure probabilities of all nodes. This topology Information can later be queried by any node.

Failure Probability Estimation

Fault model in which faults caused only by node failures and a node is inaccessible and cannot provide any service once it fails. The failure probability of a node estimated at time t is the probability that the node fails by time t þ T, where T is a time interval during which the estimated failure probability is effective. A node estimates its failure probability based on the following events/causes: energy depletion, temporary disconnection from a network (e.g., due to mobility), and application-specific factors.

Failure by Energy Depletion

Estimating the remaining energy of a battery-powered device is a well-researched problem. This adopts the remaining energy estimation algorithm in because of its simplicity and low overhead. The algorithm uses the history of periodic battery voltage readings to predict the battery remaining time. Considering that the error for estimating the battery remaining time follows a normal distribution, we find the probability that the battery remaining time is less than T by calculating the cumulative distributed function (CDF) at T.

Failure by Temporary Disconnection

Nodes can be temporarily disconnected from a network, e. g., because of the mobility of nodes, or simply when users turn off the devices.

Expected Transmission Time Computation

It is known that a path with minimal hop-count does not necessarily have minimal end-to-end delay because a path with lower hop-count may have noisy links, resulting in higher end-to-end delay. Longer delay implies higher transmission energy. As a result, when distributing data or processing the distributed data, this consider the most energy efficient paths with minimal transmission time. When path p is the shortest path from node i to node j, it imply that path p has the lowest transmission time (equivalently, lowest energy consumption) for transmitting a packet from node i to node j. The shortest distance then implies the lowest transmission time.

K-Out-of-n Data Processing

The k-out-of-n data processing problem is solved in two stages—Task Allocation and Task Scheduling. In the Task Allocation stage, n nodes are selected as processor nodes; each processor node is assigned one or more tasks; each task is replicated to n _ k þ 1 different processor nodes. An example is shown in Fig. 3a. However, not all instances Of a task will be executed—once an instance of the task completes, all other instances will be canceled

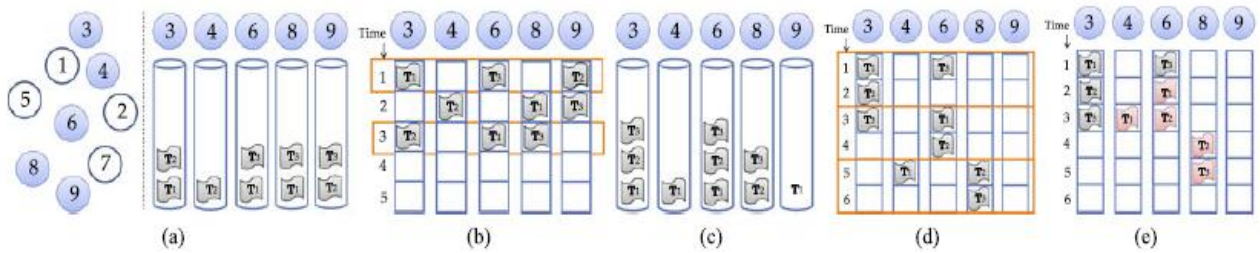

Fig. 3. (a) And (c) are two different task allocations and (b) and (d) are their tasks scheduling respectively. In both cases, node 3; 4; 6; 8; and 9 are selected as processor nodes and each task is replicated to three different processor nodes.

(e) Shows that shifting tasks reduce the job completion time from 6 to 5.

Topology Monitoring

The Topology Monitoring component monitors the network topology continuously and runs in distributed manner on all nodes. Whenever a client node needs to create a file, the Topology Monitoring component provides the client with the most recent topology information immediately. When there is a significant topology change, it notifies the framework to update the current solution.

REFERENCES

[1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," IEEE Pervasive Comput., vol. 8, no. 4, pp. 14–23, Oct.-Dec. 2009.

[2] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in Proc. 6th Conf. Comput. Syst., 2011, pp. 301–314.

[3] 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. IEEE Conf. Comput. Commun.,2012, pp. 945–953.

[4] C. Shi, V. Lakafosis, M. H. Ammar, and E. W. Zegura, "Serendipity: Enabling remote computing among intermittently connected mobile devices," in Proc. 13th ACM Int. Symp. Mobile Ad Hoc Netw. Comput., 2012, pp. 145–154.

[5] S. M. George, W. Zhou, H. Chenji, M. Won, Y. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah, "DistressNet: A wireless Ad Hoc and sensor network architecture for situation management in disaster response," IEEE Commun. Mag., vol. 48, no. 3, pp. 128–136, Mar.2010.