Volume 4 Issue IV, April 2016
ISSN: 2321-9653
International Journal for Research in Applied Science & Engineering Technology (IJRASET)

KNF Computing for Energy-Efficiency Fault-Tolerance Data Storage And Processing in Mobile Cloud

Pankaja Bgade[1], Prof. R.R.Badre[2]
[1]Pune University, MIT Academy of Engineering
Alandi Road, Pune 412105, Pune

Abstract: The advances on hardware for hand-held mobile devices, resource-intensive application still remain off bounds since they require large data processing and storage capabilities. These issues are addressed by employing remote servers, such as clouds and peer mobile devices. However, for mobile devices deployed in dynamic networks, challenges of reliability and energy efficiency remain largely unaddressed. These challenges are addressed by an approach called as k-out-of-n computing. In this solution, mobile devices successfully retrieve or process the reliable data, in the most energy-efficient way, as long as k-out-of-n distance servers are accessible. Through a virtual system implementation, the feasibility of the approach can be proved and the fault tolerance and energy efficiency performance will be improved.
Keywords: Mobile Computing, Cloud Computing, k-out-of-n Computing

I. INTRODUCTION

Core concepts like computing as utility, virtualization of resources, on demand access to computing resources, and outsourcing computing services, clouds are large-scaled distributed computing systems are built around these concepts. In recent years mobile devices have gained vast popularity this result in many applications rely on offloading all or part of their works to remote servers like peer mobile devices and clouds [6][7]. Mobile devices derive the energy required for their operation from batteries [10]. Many consumer consume electronics hand-handled devices, especially mobile phones, battery capacity is severely restricted due to constraints on weight and size of the device. This implies that energy efficiency of these devices is very important to their usability [12]. Cloud computing is a new paradigm in which at the users location computing resources such as processing, memory, and storage are not physically present. Instead, a service provider owns and manages these resources, and users access them via the Internet. While power is an important constrained relative to tethered devices in mobile devices, advances in battery and power management technology this will enable mobile devices to manage longer lived computations with fewer burdens on available power consumption [14]. Fig.1 shows the architecture

for cloud computing. Cloud services allow individuals and businesses to use software and hardware that are managed by arbitrator at remote locations. Via the internet the cloud is reliable as it enables access to applications and documents anywhere in the world. Cloud computing is often considered efficient because it allows organizations to free up resources to

focus on innovation and product development.

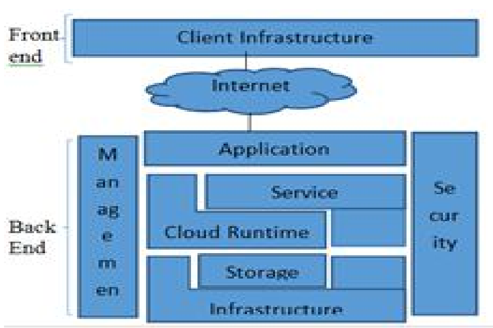

Figure 1: Cloud Computing Architecture

The rest of this paper is organized as follows. Section 2 introduces related work. Section 3 introduces the proposed work for energy consumption and fault-tolerant. Algorithm for rearrangement and topology monitoring is in section 4. Section 5 and 6 is the brief analysis of Mathematical model and Results respectively. Finally, conclusions is drawn in section 7.

II. RELATED WORK

M. Satyanarayanan [1], Small scale network in Cloud computing has gained devotion recently. A resource-rich cluster that is Cloudlet is well connected to the Internet and is available for use by nearby mobile devices; it can apply synthesis recursively to generate a family of overlays. Launching VM would 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, it seek a wavelet like decomposition of VM state into a sequence of overlays that decrease in size but increase in specificity. Another deployment challenge relates to the assumption that a relatively small set of base VMs will suffice for a large range of applications. The problem could be intensified by the common practice of releasing security patches for old OS releases. Although the patches effect could be incorporated into the overlay, it would increase overlay size.

Byung-Gon Chun [2]A small Virtual Machine (VM) overlay is delivered to mobile devices to a cloudlet infrastructure and lets it take over the computation is presented. Similar works that use VM migration are also done in CloneCloud. The 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. They combine partitioning, migration with merging, and on-demand instantiation of partitioning to address these challenges. The prototype delivers up to 20x speedup and 20x energy reduction for the simple applications

they tested, without programmer involvement, demonstrating feasibility for the approach, and opening up a path for a rich research agenda in hybrid mobile cloud systems.

III. PROPOSED WORK

Energy efficiency and fault tolerant is well studied problem in wireless sensor networks since from last decade successfully. But in mobile devices communication with data storage and mobile cloud processing systems, this problem yet remains to be addressed fully. The resource intensive applications such as video, audio or image processing and storage still remains headed away even if advanced hardware based mobile hand held devices introduced. Due to the requirement of large storage as well as computation costs for such applications, recent methods are proposed by using remote servers like clouds, peer mobile devices etc. However the mobile devices those are used for communication for such flexible networks like mobile cloud are suffered from the problems of energy efficiency and reliability. It is further extending the existing work by adding technique to improve the energy consumption based on type communication as well as improve the communication time, throughput. The technique which are introducing here is based on fuzzy rule algorithm. The aim of the algorithm is to check the possibility of excessive energy loss of mobile nodes and node failure conditions in order to improve the reliability and energy efficiency of mobile cloud data storage processing. The practical analysis of proposed will be done through the extensive simulations.

An overview of the proposed framework is depicted in Fig.2 The framework, consecutively on all mobile nodes, provides facilities to applications [13]. In Fig.2, applications create data where it is encoded and partitioned for higher data reliability and availability respectively into the data, as inputs functions that take store data are provided by applications. Each function is directed as multiple tasks that process the data concurrently on different nodes. Nodes executing tasks are processor nodes; call a set of tasks directed from one function a job [6]. The appeal for data allocation or processing operations are done by

Client nodes. As shown in Fig.2, the framework involve 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. failure probabilities of nodes and network topology information is provided by the Topology Discovery and Monitoring component, when from application appeal is received for data allocation or processing is received. On each node the failure probability is assessed by the Failure Probability component [12].

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

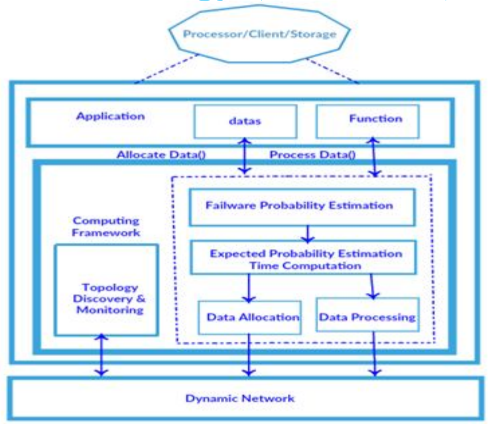

Figure 2: Amalgamation Architecture for Energy Efficiency and fault tolerance in the k-out-of-n Computing framework.

The retrieved information from failure probabilities and network topology helps in computation of ETT matrix that is done by ETT Computation component. Which signifies the expected energy consumption for communication between any pair of nodes. In the component, k-out-of-n Data Storage, the energy consumption for retrieving k fragments by any node is minimized, this is done by partitioning data into n fragments by an erasure code algorithm and stores these fragments in the network. When application prerequisite process the data and to minimize the energy consumption in retrieving and processing the stored data, the component k-out-of-n Data Processing creates a job of K tasks and schedules the tasks on n processor nodes. Therefore the component ensures that all tasks complete as long as k or more processor nodes finish their assigned tasks. The Topology Discovery and Monitoring component constantly monitors the network for any substantial change on the network topology.

IV. PROPOSED ALGORITHM

A. Re-arrangement of Task

When tasks are done scheduling the tasks are rearranged based on the free time slots and no same task is executed on other node simultaneously. With probability 1 the k-out-of-n data processing guarantee k or more functional processing nodes to complete all tasks of a job. It is possible that the subset of processing nodes that are of size less than k complete all assigned tasks.

1) Algorithm 1: Schedule Re-arrangement

a) *Z* = last time slot
b) for time t = 2 → *Z* do
c) for each scheduled task K in time t do
d) *n* ← processor node of task K
e) while *n* is idle at t - 1 and
f) K is not scheduled on any node at t - 1 do
g) Move K from t to t - 1
h) t = t - 1
i) end while
j) end for
k) end for

B. Topology Monitoring

The Topology Monitoring component monitors the network topology constantly and runs in circulated manner on all nodes. The term *s* refers to a state of a node, which can be either *C* where a node state is drastically changed and *NC* a node keeps the state. Let *e* be the number of entries in the node has changed. A set ID contains the node IDs with e greater than τ_1, a threshold parameter for a significant local topology change. The Topology Monitoring component is simple yet energy-efficient as it does not incur significant communication overhead it simply piggybacks node ID on a beacon message as the protocol is depicted in Algorithm 2. It predefine one node as a topology delegate M_{del} which is responsible for maintaining the global topology information. If e of a node is greater than the threshold τ_1, the node changes its state to *C* and piggybacks it's ID on a beacon message otherwise it changes its state back to *NC* and puts ID in a beacon message. After receiving a beacon message, nodes check the IDs in it. For each ID, nodes add the ID to set *ID* if the ID is positive; otherwise, remove the ID. If a client node finds that the size of set *ID* becomes greater than τ_2, a threshold for significant global topology change, the node notifies M_{del}; and M_{del} executes the Topology Discovery protocol. To reduce the amount of traffic, client nodes request the global topology from M_{del}, instead of running the topology discovery by themselves. After M_{del} completes the topology update, all nodes reset their status variables back to *NC* and set $e = 0$.

1) Algorithm 2: Distributed Topology Monitoring

a) At beacon interval
b) if $e > \tau_1$ and $s \neq C$ then
c) $s \leftarrow C$
d) Put +ID to a beacon message
e) end if
f) if $e \leq \tau_1$ and $s = C$ then
g) $s \leftarrow NC$
h) Put -ID to a beacon message
i) end if
j) After receiving a beacon message on M_i
k) for each ID in the received beacon message do
l) if $ID > 0$ then
m) $ID \leftarrow ID \cup ID$
n) else
o) $ID \leftarrow ID \setminus ID$
p) end if
q) end for
r) if $|ID| > \tau_2$ then
s) Notify M_{del} and M_{del} initiate topology discovery
t) end if
u) Add the s ID in M_i^s is beacon message.

V. MATHEMATICAL MODEL

Consider a dynamic network with N nodes denoted by a set $V = v_1, v_2, \ldots, v_N$. Assume nodes are time synchronized. Let use i and v_i interchangeably hereafter. The network is modeled as a graph G = (V,E), where E is a set of edges indicating the communication links among nodes.

Relationship Matrix A is a N * N matrix defining the relationship between storage nodes and nodes. More precisely, each element A_{ij} is a binary variable if A_{ij} is 0, node i will not retrieve data from storage node j; if A_{ij} is 1, node i will retrieve fragment from storage node j. Storage node list R is a binary vector containing storage nodes, i.e., $R_i = 1$, v_i is a storage node.

The Expected Transmission Time Matrix T is defined as a N * N matrix where element T_{ij} corresponds to the ETT for transmitting a fixed size packet from node i to node j considering the failure probabilities of nodes in the network, i.e., multiple possible paths between node i and node j. For estimating transmission time between two nodes in one hop, ETT metric has been widely used. Assign each edge of graph G a positive estimated transmission time. Then, the path with the shortest transmission time between any two nodes can be found. Because of the dynamic topology the shortest path for any pair of nodes may change over time. ETT, represents the expected transmission time, or expected transmission energy between two nodes because of

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

multiple paths due to nodes failures.

Scheduling Matrix C is an X * N * M matrix where element C_{xij} = 1 indicates that task j is scheduled at time r on node i; otherwise, C_{xij} = 0. x is a relative time referenced to the starting time of a job. Since all tasks are instantiated from the same function, assume they spend approximately the same processing time on any node. Given the terms and notations, it is ready to formally describe the k-out-of-n data allocation and k-out-of-n data processing problems.

A. Formulation of k-out-of-n Data Allocation Problem

It is intended to find n storage nodes denoted by X = s1; s2; :::sn , X _ V such that the total expected transmission cost from a node to its k closest storage nodes in terms of ETT is minimized. Formulate the problem as an ILP in Equations 1 - 5.

$$A_{opt} - arg_A \, min \sum_{i=1}^{N} \sum_{j=1}^{N} T_{ij} A_{ij} \qquad (1)$$

$$\text{Subject to: } \sum_{i-1}^{N} R_j = n \qquad (2)$$

$$\sum_{i=1}^{N} A_{ij} - k \; \forall i \qquad (3)$$

$$R_{ij} - A_{ij} \geq 0 \; \forall i \qquad (4)$$

$$R_j \text{and } A_{ij} \in 0,1 \; \forall i,j \qquad (5)$$

The first constraint (2) selects exactly n nodes as storage nodes; the second constraint (3) indicates that each node has access to k storage nodes; the third constraint (4) ensures that jth column of A can have a non-zero element if only if R_j is 1; and constraints (5) are binary requirements for the decision variables.

B. Formulation of k-out-of-n Data Processing Problem

The intention is to find nodes *n* in V as processor nodes such that energy consumption for processing a job of K tasks is minimized. Before formulating the problem, define some functions: (i) f_1(i) returns 1 if node i in C has at least one task; otherwise, it returns 0; (ii) f_2(j) returns the number of instances of task j in S; and (iii) f_3(z; j) returns the transmission cost of task j when it is scheduled for the z^{th} time. Now formulate the k-out-of-n data processing problem as shown in Equation 6 - 11. x represents the time slot of executing a task; i is the index of nodes in the network; j is the index of the task of a job. The D^r, the Data Retrieval Time Matrix, is a N * M matrix, where the element D^r_{ij} corresponds to the estimated time for node i to retrieve task j. Then D^r is computed by summing the transmission time from node i to its k closest storage nodes of the task.

$$\text{minimize } \sum_{x}^{X} \sum_{i}^{N} \sum_{j}^{M} C_{ij} D_{ij}^{r} \qquad (6)$$

$$\text{Subject to: } \sum_i f_1(1) = n \qquad (7)$$

$$f_2 = n - k + 1 \; \forall j \qquad (8)$$

$$\sum_{x=1}^{L} C_{xij} \leq 1 \; \forall i,j \qquad (9)$$

$$\sum_{i=1}^{N} C_{xij} \leq 1 \; \forall i,j \qquad (10)$$

$$\sum_{j-1}^{M} f(z_1, j) \leq \sum_{j-1}^{M} f(z_2, j) \; \forall z_1 \leq z_2 \qquad (11)$$

The first constraint (7) ensures that n nodes in the network are selected as processor nodes. The second constraint (8) indicates that each task is replicated n - k +1 times in the schedule such that any subset of k processor nodes must contain at least one instance of each task. The third constraint (9) states that each task is replicated at most once to each processor node. The fourth constraint (10) ensures that no duplicate instances of a task execute at the same time on different nodes. The fifth constraint (11) ensures that a set of all tasks completed at earlier time should consume lower energy than a set of all tasks completed at later time.

C. Failure Probability Estimation

Failure probability of a node examined at time t is the probability that the node fails by time t + T, where T is a time interval during which the estimated failure probability is nominal. A node estimates its failure probability based on the following events/causes: energy depletion, temporary disconnection from a network, and application-specific factors. Let f_i be the event that node i fails and let f_i^A , f_i^B and f_i^C be the events that node i fails due to application specific factors, energy depletion, temporary disconnection from a network respectively. The failure probability of a node is as follows:

$$P[f_i] = (1-P[f_i^A])(1-P[f_i^B])(1-P[f_i^C])$$

VI. CONCLUSION

In the paper, the challenges in executing the sophisticated application and in the use limited resource on hand-handled devices are addressed. The k-out-of-n computing is used for energy minimization during assigning data fragments to nodes that leads for energy consumption in storing reliable data and retrieving the data reliably. It provides a crystal clear idea about the wide dimensions of energy consumption for retrieving and processing the distributed data and to support the fault-tolerant under the dynamic network topology.

REFERENCES

[1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, The case for VM-based cloudlets in mobile computing, Pervasive Computing,, IEEE, vol. 8, pp. 1423, 2009.
[2] 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.
[3] Chien-An Chen, Myounggyu Won, Radu Stoleru, and Geofrey, G. Xie, Energy-Efficient Fault-Tolerant Data Storage Processing in Mobile Cloud, Proc. Of the IEEE, vol. 3, no.1, pp. 476489, 2014.
[4] Imre Kelnyi, Jukka K. Nurminen, Cloud Torrent Energy-Efficient BitTorrent Content Sharing for Mobile Devices via-Cloud Services.
[5] Eric Chen, Satoshi Ogata, Keitaro Horikawa, Off-loading Android Applications to the Cloud without Customizing Android,Eighth IEEE PerCom Workshop on Pervasive Wireless Networking 2012.
[6] Jasbir Kaur, Supriya Kinger , Resource allocation for energy-efficient k-out-of-n system in mobile ad hoc networks , International Journal of Computer Science and Information Technologies, Vol. 5 (5) , 2014.
[7] C. Chen, M. Won, R. Stoleru, and G.Xie, Resource allocation for energy efficient k-out-of-n system in mobile ad hoc networks, in Proc. of ICCCN, 2013.
[8] C. A. Chen, M. Won, R. Stoleru, and G.Xie, Energy-efficient-fault tolerant data storage and processing in dynamic network,, in Proc. of MobiHoc, 2013.
[9] Kuo-Zuo-koon, Chapter7:The k-out-of-n system model.
[10] S.M.George, W.Zhou, H. Chenji,M.Won, Y. Lee, A. Pazarloglou, R. Stoleru, and P. Barooah , Distress- Net: a wireless AdHoc and sensor network architecture for situation management in disaster response, IEEE Communications Magazine, vol. 48, no. 3, 2010.
[11] D. Shires, B. Henz, S. Park, and J. Clarke, Cloudlet seeding: Spatial deployment for high performance tactical clouds, International Journal on Computer Networks (Elsevier), WorldComp, 2012.
[12] D. Neumann, C. Bodenstein, O. F. Rana, and R.Krishnaswamy, STACEE: enhancing storage clouds using edge devices, WACE, 2011.
[13] D. Huang, X. Zhang, M. Kang, and J. Luo, MobiCloud: Building secure cloud framework for mobile computing and communication, SOSE, 2010.
[14] Sun Microsystems, Inc. Introduction to Cloud Computing Architecture, White Paper 1st Edition, June 2009.

AUTHOR PROFILE

Pankaja Bagde received the B.E. degree in Information Technology from Rashtrasant Tukadoji Maharaj Nagpur University in 2013. Currently she is PG Scholar, perceiving M.E. in Computers Engineering from MIT Academy of Engineering, Pune. Areas of interests are computer networking and cloud computing.

Prof. R.R.Badre received her M.E. degree in Computer Science and Engineering from Shivaji University, Kolhapur. She is currently working as an Associate Professor in the Department of Computer Engineering in MIT Academy of Engineering, Pune. She has published more than 8 papers in both International journals and conferences. She is also a member in Indian Society of Technical Education.