RESEARCH ARTICLE OPEN ACCESS

Implementing K-Out-Of-N Computing For Fault Tolerant Processing In Mobile and Cloud Computing

Mr. R. Kiran Babu[1], Mr. J.Sagar Babu[2]
Assistant Professor Head, Department of Information Technology Princeton College of Engineering & Technology Ghatkesar Hyderabad.
Associate Professor Head, Department of Computer science & engineering Princeton College of Engineering & Technology Ghatkesar Hyderabad.

ABSTRACT
Despite the advances in hardware for hand-held mobile devices, resource-intensive applications (e.g., video and imagestorage 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 andmobility as in a mobile cloud), however, challenges of reliability and energy efficiency remain largely unaddressed. To the best of ourknowledge, we are the first to address these challenges in an integrated manner for both data storage and processing in mobilecloud, an approach we call k-out-of-n computing. In our solution, mobile devices successfully retrieve or process data, in the mostenergy-efficient way, as long as k out of n remote servers are accessible. Through a real system implementation we prove the feasibilityof our approach. Extensive simulations demonstrate the fault tolerance and energy efficiency performance of our framework in largerscale networks.

Index Terms—Mobile computing, cloud computing, mobile cloud, energy-efficient computing, fault-tolerant computing.

I. INTRODUCTION

Mobile Cloud Computing (MCC) is the combination of cloud computing, mobile computing and wireless networks to bring rich computational resources to mobile users, network operators, as well as cloud computing providers. The ultimate goal of MCC is to enable execution of rich mobile applications on a plethora of mobile devices, with a rich user experience. MCC provides business opportunities for mobile network operators as well as cloud providers. More comprehensively, MCC can be defined as "a rich mobile computing technology that leverages unified elastic resources of varied clouds and network technologies toward unrestricted functionality, storage, and mobility to serve a multitude of mobile devices anywhere, anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-use principle.MCC uses computational augmentation approaches by which resource-constraint mobile devices can utilize computational resources of varied cloud-based resources.In MCC, there are four types of cloud-based resources, namely distant immobile clouds, proximate immobile computing entities, proximate mobile computing entities, and hybrid (combination of the other three model). Giant clouds such as Amazon EC2 are in the distant immobile groups whereas cloudlet or surrogates are member of proximate immobile computing entities. Smartphone's, tablets, handheld devices, and wearable computing devices are part of the third group of cloud-based resources which is proximate mobile computing entities.

II. CLOUD COMPUTING

Cloud computing is the delivery of computing services over the Internet. Cloud services allow individuals and businesses to use software and hardware that are managed by third parties at remote locations. Examples of cloud services include online file storage, social networking sites, webmail, and online business applications. The cloud computing model allows access to information and computer resources from anywhere that a network connection is available. Cloud computing provides a shared pool of resources, including data storage space, networks, computer processing power, and specialized corporate and user applications.

Characteristics

The characteristics of cloud computing include on-demand self service, broad network access, resource pooling, rapid elasticity and measured service. On-demand self service means that customers (usually organizations) can request and manage their own computing resources. Broad network access allows services to be offered over the Internet or private networks. Pooled resources means that customers draw from a pool of computing resources, usually in remote data centres. Services can be scaled larger or smaller; and use of a service is measured and customers are billed accordingly. The cloud computing service models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In a Software as a Service model, a pre-made application, along with any required software, operating system, hardware, and network are provided. In PaaS, an operating system, hardware, and network are provided, and the customer installs or develops its own software and applications. The IaaS model provides just the hardware and network; the customer installs or develops its own operating systems, software and applications. Deployment of cloud services are the Cloud services are typically made available via a private cloud, community cloud, public cloud or hybrid cloud. Generally speaking, services provided by a public cloudare offered over the Internet and are owned and operated by a cloud provider. Some examples include services aimed at the general public, such as online photo storage services, e-mail services, or social networking sites. However, services for enterprises can also be offered in a public cloud. In a private cloud, the cloud infrastructure is operated solely for a specific organization, and is managed by the organization or a third party.

In a community cloud, the service is shared by several organizations and made available only to those groups. The infrastructure may be owned and operated by the organizations or by a cloud service provider.Ahybrid cloudis a combination of different methods of resource pooling (for example, combining public and community clouds). Cloud services are popular because they can reduce the cost and complexity of owning and operating computers and networks. Since cloud users do not have to invest in information technology infrastructure, purchase hardware, or buy software licences, the benefits are low up-front costs, rapid return on investment, rapid deployment, customization, flexible use, and solutions that can make use of new innovations. In addition, cloud providers that have specialized in a particular area (such as e-mail) can bring advanced services that a single company might not be able to afford or develop. Some other benefits to users include scalability, reliability, and efficiency. Scalability means that cloud computing offers unlimited processing and storage capacity. The cloud is reliable in that it enables access to applications and documents anywhere in the world via the Internet. Cloud computing is often considered efficient because it allows organizations to free up resources to focus on innovation and product development.

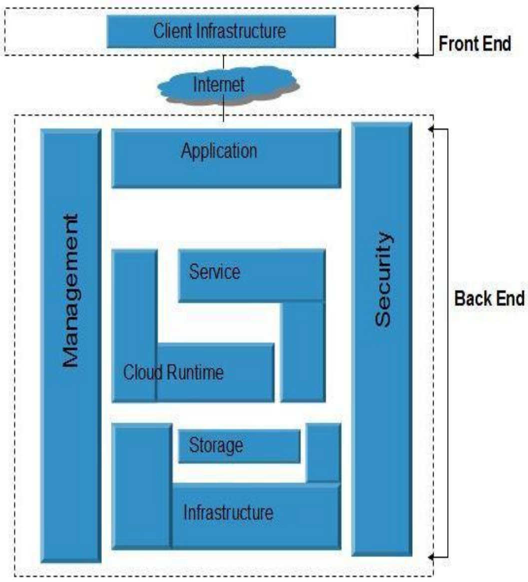


Fig 1.0 Cloud computing architecture

III. FORMULATION OF K-OUT-OF-N DATA PROCESSINGPROBLEM

The objective of this problem is to find n nodes in V as processornodes such that energy consumption for processing ajob of M tasks is minimized. In addition, it ensures that thejob can be completed as long as k or more processors nodesfinish the assigned tasks. Before a client node starts processinga data object, assuming the correctness of erasure coding,it first needs to retrieve and decode k data fragmentsbecause nodes can only process the decoded plain dataobject, but not the encoded data fragment.In general, each node may have different energy costdepending on their energy sources; e.g., nodes attachedto a constant energy source may have zero energy costwhile nodes powered by battery may have relativelyhigh energy cost. For simplicity, we assume the networkis homogeneous and nodes consume the same amount ofenergy for processing the same task. As a result, onlythe transmission energy affects the energy efficiency of thefinal solution. We leave the modeling of the general caseas future work.

IV. ENERGY EFFICIENT AND FAULT TOLERANT DATA ALLOCATION AND PROCESSING

This section presents the details of each component in ourframework.

A Topology Discovery

Topology Discovery is executed during the networkinitialization phase or whenever a significant change ofthe network topology is detected (as detected by theTopology Monitoring component). During Topology Discovery,one delegated node floods a request packetthroughout the network. Upon receiving the requestpacket, nodes reply with their neighbor tables and failureprobabilities. Consequently, the delegated node obtainsglobal connectivity information and failure probabilitiesof all nodes. This topology information can later be queriedby any node.

B Failure Probability Estimation

We assume a fault model in which faults caused only bynode failures and a node is inaccessible and cannot provideany service once it fails. The failure probability of a nodeestimated at time t is the probability that the node fails bytime t þ T, where T is a time interval during which the estimatedfailure probability is effective. A node estimates itsfailure probability based on the following events/causes:energy depletion, temporary disconnection from a network(e.g., due to mobility), and application-specific factors. Weassume that these events happen independently.

C Failure by Energy Depletion

Estimating the remaining energy of a battery-powereddevice is a well-researched problem [8]. We adopt theremaining energy estimation algorithm in [8] because of itssimplicity and low overhead. The algorithm uses the historyof periodic battery voltage readings to predict the batteryremaining time. Considering that the error forestimating the battery remaining time follows a normal distribution[9], we find the probability that the batteryremaining time is less than T by calculating the cumulativedistributed function (CDF) at T.

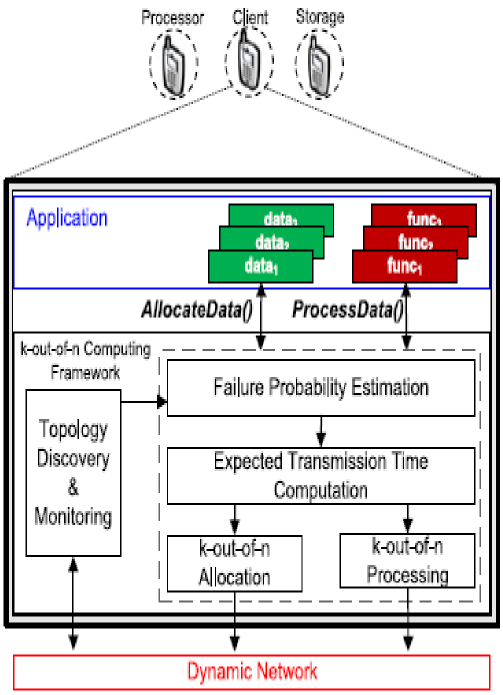


Fig 2.0 K-out of N computing

V. OUT-OF-N DATA PROCESSING

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

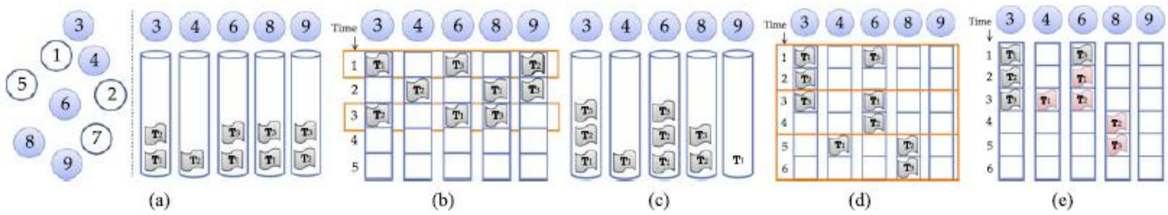


Fig 3.0 K-out-of-n data processing

task completes, all other instances will be canceled. The task allocation can be formulated as an ILP as shown inEqs. In the formulation, Rijis a N _M matrix which predefines the relationship between processor nodes and tasks; each element Rij is abinary variable indicating whether task j is assigned toprocessor node i. X is a binary vector containing processornodes, i.e., Xi ¼ 1 indicates that vi is a processornode. The objective function minimizes the transmissiontime for n processor nodes to retrieve all their tasks.

$$\overline{R_{opt}} = \underset{\overline{R}}{arg\,min} \sum_{i=1}^{N} \sum_{j=1}^{M} T_{ij}^{r} \overline{R_{ij}}$$

$$\text{Subject to:} \quad \sum_{i=1}^{N} \overline{X_i} = n$$

$$\sum_{i=1}^{N} \overline{R_{ij}} = n - k + 1 \; \forall j$$

$$\overline{X_i} - \overline{R_{ij}} \geq 0 \; \forall i$$

$$\overline{X_j} \text{ and } \overline{R_{ij}} \in \{0, 1\} \forall i, j$$

Once processor nodes are determined, we proceed to theTask Scheduling stage. In this stage, the tasks assigned toeach processor node are scheduled such that the energy andtime for finishing at least M distinct tasks is minimized,meaning that we try to shorten the job completion timewhile minimizing the overall energy consumption. Theproblem is solved in three steps. First, we find the minimalenergy for executing M distinct tasks in Rij. Second, we find a schedule with the minimal energy that has the shortestcompletion time. These two steps are repeatedn-k+1 times and M distinct tasks are scheduled upon eachiteration. The third step is to shift tasks to earlier time slots. A task can be moved to an earlier time slot as long as noduplicate task is running at the same time, e.g., in Fig3.0,task 1 on node 6 can be safely moved to time slot 2 becausethere is no task 1 scheduled at time slot 2.

Algorithm 1: Schedule Re-Arrangement

L ¼ last time slot in the schedule
for time t ¼ 2 ! L do
for each scheduled task J in time t do
n processor node of task J
while n is idle at t _ 1 AND
J is NOT scheduled on any node at t _ 1 do
Move J from t to t _ 1
t ¼ t _ 1
end while
end for
end for

VI. SIMULATION RESULTS

We conducted simulations to evaluate the performance ofour k-out-of-n framework (denoted by KNF) in larger scalenetworks. We consider a network of 400_400 m2 where upto 45 mobile nodes are randomly deployed. The communicationrange of a node is 130 m, which is measured on oursmartphones. Two different mobility models are tested—Markovian Waypoint Model and Reference Point GroupMobility (RPGM). Markovian Waypoint is similar to RandomWaypoint Model, which randomly selects the waypointof a node, but it accounts for the current waypointwhen it determines the next waypoint. RPGM is a groupmobility model where a subset of leaders are selected; eachleader moves based on Markovian Waypoint model andother non-leader nodes follow the closest leader.

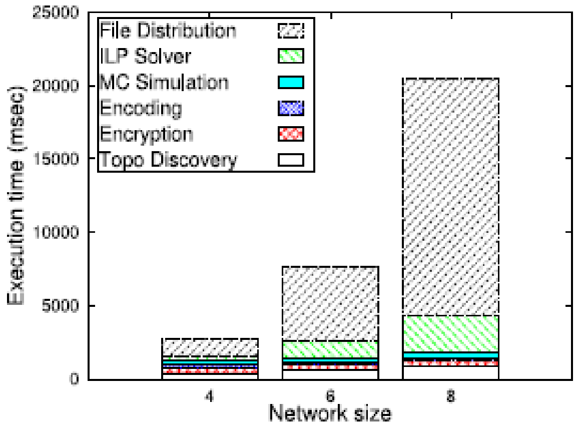
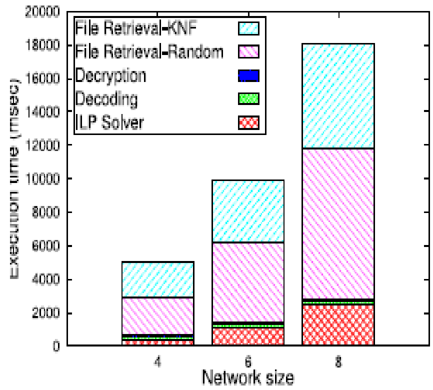


Fig 4.0 Execution time for different components



VII. CONCLUSION

We presented the first k-out-of-n framework that jointlyaddresses the energy-efficiency and fault-tolerance challenges.It assigns data fragments to nodes such that othernodes retrieve data reliably with minimal energy consumption.It also allows nodes to process distributed data suchthat the energy consumption for processing the data is minimized. Through system implementation, the feasibility ofour solution on real hardware was validated. Extensive simulationsin larger scale networks proved the effectiveness ofour solution.

REFERENCES

[1] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "*The case for VM-based*

cloudlets in mobile computing," IEEE PervasiveComput., 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. MobileAd 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.

[6] D. W. Coit and J. Liu, "*System reliability optimization with k-outof- n subsystems*," Int. J. Rel., Quality Safety Eng., vol. 7, no. 2, pp. 129–142, 2000.

[7] D. S. J. D. Couto, "*High-throughput routing for multi-hop wireless networks*," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., MIT, Cambridge, MA, USA, 2004.

[8] Y. Wen, R. Wolski, and C. Krintz, "*Online prediction of battery lifetime for embedded and mobile devices*," in Proc. 3rd Int. Conf.Power-Aware Comput. Syst., 2005, pp. 57–72.

[9] A. Leon-Garcia, Probability, Statistics, and Random Processes for Electrical Engineering. Englewood Cliffs, NJ, USA: PrenticeHall, 2008.

[10] S. Huchton, G. Xie, and R. Beverly, "*Building and evaluating a kresilient mobile distributed file system resistant to device compromise*," inProc. Mil. Commun. Conf., 2011, pp. 1315–1320.

Mr.R.KiranBabu is Asst. Professor & Head, Department of Information Technology, Princeton College of Engineering & Technology, Ghatkesar, R.R-Dist. He has Working experience in teaching field since 2012.His qualification is B.Tech in Information Technology from Prakasam Engineering College, Kandukur, Prakasam-Dist in 2007.M.Tech in Computer Science & Engineering from Princeton College of Engineering & Technology, Ghatkesar,R.R--Dist. Completed In 2012.

Mr.J.SagarBabu is Associate Professor & Head, Department of Computer science & engineering, Princeton College of Engineering & Technology, Ghatkesar, R.R-Dist, He has Working experience in teaching field since 2008.His qualification is B.Tech in Computer Science & Engineering from Dr.Samuel George Institute of Engineering & Technology, Markapur, Prakasam-Dist in 2005.M.Tech in Computer Science & Engineering from QIS College of Engineering & Technology, Ongole,Prakasam-Dist. Completed In 2008.