International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016 [ICSSCCET 2016]

ISBN	978-81-929866-6-1
Website	icssccet.org
Received	25 – February – 2016
Article ID	ICSSCCET050

VOL	02
eMail	icssccet@asdf.res.in
Accepted	10 - March – 2016
eAID	ICSSCCET.2016.050

An Efficient Power Aware Mobile Job Execution in Public Mobile Cloud

S Ganapathi Ammal[1], M Suresh[2], S Mani[3], S Senthil kumar[4], M Senkottaiyan[5]
[1]Assistant Professor, Department of Information Technology, Karpagam Institute of Technology, Coimbatore
[2]Assistant Professor, [3,4,5] UG Scholar, Department of Information Technology, Selvam College of Technology, Namakkal

Abstract: *Mobile application moves the computing power and the storage away from the mobile device. Centralized computing platform located in clouds. Collaborative task are sequential execution of mobile device or cloud. Tasks are uploaded to the base station by offloading policy using static network. When the cloud environment is connected to the base station the task are executed. The throughput returns the task from cloud to the mobile device. Execution of process takes place using shortest path algorithm. Each task executes under one-climb policy. This policy is efficient and saves time. In existing system we use the enumeration algorithm to find optimal solution. In this project, the concept is to reduce the power consumption and minimizes the cost of workflow while meeting the deadline. LARAC (Lagrange Relaxation based Aggregated Cost) algorithm is used to bring efficiency for simulation results. In order to achieve optimized solution the task is carried out in simulation environment. In addition to that it reduces the power consumption.*

Index Terms: *Collaborative task execution, mobile cloud computing, scheduling policy, Markov decision process, stochastic Wireless channel.*

1. INTRODUCTION

Computation offloading is Important for handheld devices. By offloading modules, we can achieve, at most, 75% Savings in execution time and 56% in battery usage. Processor of limited capacity and limited amount of main memory mobile cloud computing the main purpose is Energy consumption for mobile devices. The clone cloud model supports the augmented execution technique that offloads parts of the Application [2],[3]. Execution to the nearby cloud infrastructure Genetic algorithm produces an optimization solution in polynomial time using various evolutionary techniques from a large search space. The PCP (Partial Critical Path) will initially hold the parent of one of the exit task which was not already scheduled [4]. If it was already scheduled, the task with the latest finish time will be chosen and added to the Partial Critical Path (PCP).

The state transition and the immediate reward are often random in nature, and hence the cumulative reward X may be inherently stochastic. Classical approach deals with the maximization of the expected value of X [6], [7], which implicitly Assumes that the decision maker is risk-neutral. Lagrange Relaxation based Aggregated Cost (LARAC) execution technique proposed, that cloud Belongs to the second category. The LARAC algorithm provides a polynomial heuristic solution for the DCLC problem. The parameters of the MDP, Namely the transition probability and the reward, are not precisely known. While the decision maker is risk-neutral to internal randomness performance fluctuation due to the stochasticity of the state transition, immediate reward and possible randomness of [9],[10]the action, risk averse to the performance deviation due to the parameter uncertainty. a common NP-complete problem is the process of mapping the workflow tasks to the appropriate resources with the help of efficient scheduling algorithms in a way to satisfy the Quality of Service (QoS) of the user along with cost optimization and deadline constraint. The set of policies that depend on the time horizon, the current state, and the accumulated reward up-to-now, which we called "Pseudo-Markovian" policy.

Cite this article as: S Ganapathi Ammal, M Suresh, S Mani, S Senthil kumar, M Senkottaiyan. "An Efficient Power Aware Mobile Job Execution in Public Mobile Cloud". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 267-273. Print.

Grouping flows that belong to guaranteed delay service [8], may result in lower resource requirement than handling all the flows separately. The highest constraint is exceeded, or there is no more possibility of improving the paths. Since this extension uses delay instead of hop count, which is a continuous metric, the routing table which contains entries for all possible delays can be very large, even of unrealizable size. As we mentioned the CBF algorithm has an exponential running time. Widyono have not analyzed it conclusively, but stated that despite of its exponential nature, the performance of the algorithm was reasonable in practice. Distribution networks describe the flow of a commodity from a source, where it is produced, to sinks. Such networks can model many scenarios such as liquids flowing through a pipe, parts through assembly lines or current through a power network. Each edge in this network is a conduit for The commodity and has a certain capacity which caps the amount of flow through it. In addition, it may incur a cost when the commodity flows through it. Vertexes are the conduit junctions or sink which may consume some of the incoming flow and forward the rest on a subset of its incident edges. This property is also called flow conservation and is equivalent to Kirchhoff's current law when the commodity is electric current. Additionally, the flow from the source to sinks must take the form of a tree i.e. acyclic, also called feeder tree in the context of power distribution networks.

The optimal flow configuration of distribution networks, an essential problem in managing power distribution networks. Experiments demonstrate the efficiency and scalability.

2. System Architecture

The rest of the paper is organized as Section is provides the related work. It presents system models and a mathematical formulation for the scheduling policy of the collaborative task execution. Below that show the fig 2.1

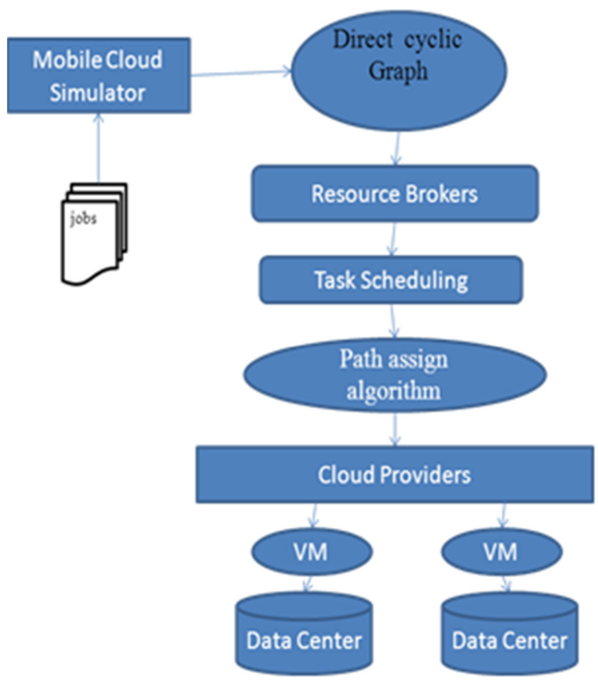

Fig 2.1 Logical System flow architecture

The one-climb policy indicates that there will occur only once task migration for the optimal solution of policy during the entire application execution if ever. For example by the role of one-climb policy, execution in not optimal. Task are loaded the base station of offloading policy.

3. Related Work

Computation offloading is a procedure that takes resource intensive computations from a mobile device to the resource rich cloud or server to get it processed there and returns back the results to the device [1]. Computation Offloading is Important for handheld devices. Certain applications are too resource demanding to run on handheld devices. One way to use those application programs is to offload all or part of the computation to a more powerful. One of the performance based application development model is the Clone Cloud Model. Clone cloud model as the name suggests creates a clone of the mobile device in the cloud infrastructure. The Handheld device and the clone need to be synchronized properly for the smooth functioning.

Cite this article as: S Ganapathi Ammal, M Suresh, S Mani, S Senthil kumar, M Senkottaiyan. "An Efficient Power Aware Mobile Job Execution in Public Mobile Cloud". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 267-273. Print.

The clone cloud model supports the augmented execution technique that offloads parts of the Application execution to the nearby cloud infrastructure. At the time of execution the data is transferred to the clone on the cloud and the handheld goes in the sleep mode till the processed data is returned from the clone of the device from the cloud. The advantage of this model is that when a smart phone is lost or destroyed, the clone can be used as a backup for the recovery of data and applications. Clone Cloud augments execution of the smart phone applications on the cloud by performing a code analysis for application partitioning, taking into consideration the offloading cost and constraints.

4. The Description of the Algorithm

Application execution in a Mobile system is as the application is either executed by the CPU, Co-processor or in the cloud. The data Input as shown in the figure 1 represents the data stored on the memory that is to be processed to get the required output Output. So here the data to be processed is transmitted to the CPU, Co-processor or in the cloud. So in our model if the data is supposed to consume more energy and time on the device we will tram it's the data to the cloud. For the execution. So here it takes some Time to transmit the data to the cloud. Suppose the time required by the cloud is Cloud. We will divide the time required. The first part is the transmission time trams and the second part execution time taxes that give the exact execution time by the cloud. Consider the speed of the network to transmit the data to the cloud is "A". Consider Cloud as the power consumed by the network interface to transmit and receive the data from the cloud. Now the total time by the cloud is given by the above formula we can give the total amount of time by the cloud to execute the given data. So as the time required is reduced the performance of battery increases gradually. So time required is directly proportional to the performance as well as the battery consumption. So using this designed model by reducing the response time will help us to reduce the energy consumption of the device. We have designed a Matrix multiplication application.

The above proposed model that reduces the response time and offloaded the calculations to the cloud to test the amount of energy that can be preserved on various stable and unstable internet connections such as Edge, 3G, Wi-Fi of various Bandwidths. We have performed this experiment on Samsung Galaxy S3 I9300. We adopted Samsung galaxy S3 I-9300, which is a popular and powerful smart phone, as our Device under test. I-19300 is equipped with a Quad-core 1.4 GHZ cortex A9 processor, a 1GB RAM, and a Wi-Fi IEEE 802.11 a/b/g/n interface. The operating system used in I-9300 is Android 4.3 Jellybean. It has a standby time of up-to 590 hours on 2G and up-to 790 hours on 3G. It has a data speed of 21Mbps on HSDPA, and 5.76 Mbps on HSUPA and a Li-Ion 2100 mAh battery. We will show that the collaborative task execution is more energy-efficient and flexible than the approach of offloading an entire application to the cloud [12]. A probabilistic constraint of time deadline into the optimization problem.

5. The Optimality of the Path

The cloud clone of network variation number of times, and found that there is a lot of variations in the execution times on the Edge and 3G connections as these connections are not stable because there are network variations with respect to place. In contrast to this we found that there were negligible variations on Wi-Fi connections as these connections are Stable in terms of Bandwidth and speed as well[5]. As a further development of the CBF algorithm, using a kind of scaling technique, the authors of gave fully polynomial time approximation schemes for the DCLC problem, namely they proved. a polynomial time algorithm that is able to find a path satisfying the delay constraint with cost no greater than a factor of 1 from the optimum.

The running time of the best known approximation scheme is unfortunately; in practical cases the running time of these methods for sufficiently small _ will be worse than even of the CBF algorithm, which makes these results rather theoretical[7],[8]. All other algorithms try to use some kind of heuristics or approximation. Does not give optimal paths, but provides a simple heuristic solution to the DCLC problem [10]. The Fallback routing algorithm assumes that there are ranked metrics in the network. First, the routing algorithm calculates the shortest path for the first metric (i.e. cost), and then checks whether it can guarantee all the other Qus requirements. If the path failed, the algorithm tries to find another one for the next metric, until an appropriate path.

$\Pr(X\pi u \geq V) = \_\pi \in \Pi h$
$\Pr(X\pi u \geq V \mid \pi ukim \sim \pi) \Pr(Pu \sim \pi)$
$= \_\pi \in \Pi h$
$\Pr(X\pi \geq V) \Pr(\pi u \sim \pi)$.

The routing fails for all the metrics. This algorithm is very simple, fast and always gives an appropriate solution if it exists, but there is no guarantee of finding the optimal route and we do not know anything about the quality of the found path. The algorithm proposed by Milton [7] improves the previous idea by combining paths calculated on the different metrics, By first calculating the paths based on the metrics from the source node to the others and from all the nodes to the destination, and then trying all the possible combinations of them.

There are two more algorithms improving the idea of Fallback. Ishida [8] proposed a distributed algorithm in which the nodes always choose the least cost path until it fulfills the delay requirements and from that point they choose the path with the least delay. The

Cite this article as: S Ganapathi Ammal, M Suresh, S Mani, S Senthil kumar, M Senkottaiyan. "An Efficient Power Aware Mobile Job Execution in Public Mobile Cloud". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 267-273. Print.

Delay Constrained Unicast Routing (DCUR) algorithm proposed by fallback [9] is similar to it, but it can choose between the least cost and least delay paths independently from the choice of the previous node. These three algorithms have higher but still reasonable running time, and it is more likely that they find a solution near the optimal than it was in case of the Fallback algorithm, but neither these algorithms have guarantees for the optimality of the path.Cheng and Nahrstedt [10] give an algorithm to find a path that meets two requirements in polynomial time.

The algorithm reduces the original problem to a simpler one by modifying the cost function, based on which the problem can be solved using an extended shortest path algorithm. The clone cloud model supports the augmented execution technique that offloads parts of the Application execution to the nearby cloud infrastructure. At the time of execution the data is transferred to the clone on the cloud and the handheld goes in the sleep mode till the processed data is Returned from the clone of the device from the cloud. The advantage of this model is that when Smartphone is lost or destroyed, the clone can be used as a backup for the recovery of data and applications. Clone Cloud augments execution of the smart phone applications on the cloud by performing a code analysis for application partitioning, taking into consideration the offloading cost and constraints. The running time of the best known approximation scheme. Tasks are loaded to the base station of mobile device and when the cloud is connected to the base station task is executed.

6. Pseudo-Polynomial Algorithms

The probabilistic goal MDP can be solved in polynomial time. In this section we develop a pseudo-polynomial algorithm to handle this question. Recall that the running time of a pseudo polynomial algorithm is polynomial in the number of parameters and the size of the parameter, as opposed to polynomial algorithms whose running time is polynomial in the number of parameters and poly algorithmic in the size of the parameters.

_i∈Iwi ≤ W; _i∈Ivi ≥ V.

The LARAC algorithm found almost the same paths as CBF, while DCUR found paths with higher cost, and the worst cost was achieved by Fallback. The result of Fallback meets our expectations, however we could not tell before that DCUR is worse than LARAC, although it could be seen that DCUR can easily find paths with higher cost than the optimal. LARAC's lower bound for the optimal path cost is very close to the results of CBF.

7. Energy Based Application Development Model

Maximizing the battery lifetime is one of the most desired things for a smart phone. The cloud model [5] One of the energy based application model for mobile cloud computing. The advantages and disadvantages of this model are as it supports self contained application components that are decoupled from each other. [1] Disadvantage: It requires skilled programmers to develop the application components that are later used. Secondly in cloud a single application partition can only execute on one orchestrator at a time. Thirdly, the data that is exchanged between the components is not provided with any security.

8. Performance Based Application Development Model

Enhancing the performance of the mobile device is the need of the hour to run the heavy resource demanding applications. So such models only deal with offloading the tasks to improve the performance without caring about the energy efficiency. One of the performance based application development model is the Clone Cloud Model [4]. Clone cloud model as the name suggests creates a clone of the mobile device in the cloud infrastructure. The Handheld device and the clone need to be synchronized properly for the smooth functioning.

The clone cloud model supports the augmented execution technique that offloads parts of the Application execution to the nearby cloud infrastructure. At the time of execution the data is transferred to the clone on the cloud and the handheld goes in the sleep mode till the processed data is returned from the clone of the device from the cloud. The advantage of this model is that when a Smartphone is lost or destroyed, the clone can be used as a backup for the recovery of data and applications. Moreover, Clone Cloud augments execution of the Smart phone applications on the cloud by performing a code analysis for application partitioning, taking into consideration the offloading cost and constraints. The model is only capable of migrating at points in the execution where no native heap state is collected. Moreover, Clone Cloud requires the development of cost model for every application under different partitions, where each partition is executed separately on the mobile device and the cloud.

9. Energy and Performance Based Application Development Model

These are the application models that are concerned with the Performance as well as Energy Efficiency of the Mobile cloud application. One of the Multi-Purpose application models is MAUI [6]. This model provides offloading application with minimum possible intervention of the programmer. This is a plus point as programming to offload is not preferable in the cases of mobile phones. MAUI is more focused towards the energy efficiency so it offloads all the heavy tasks to the cloud or nearby infrastructure to save energy. MAUI works on the History based approach while making decisions to offload the task. It uses a profiler to make Decisions. Bellow shows the fig9.1 energy and performance model.

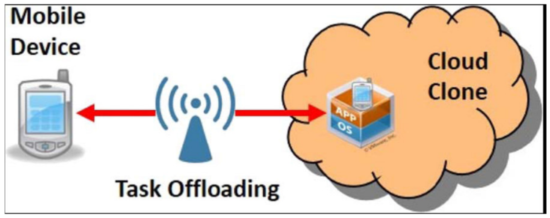

Fig9.1: Energy and Performance Based Application Model.

Profiler is an optimization engine that takes the decision whether offloading is more efficient or performing on the local resource is more efficient. It takes these decisions based on the History based approach i.e. based on the some previous database about the performance and energy requirements [6]. The application partitioning is dynamic and the offloading is done on the basis of methods instead of complete application modules to minimize the offloading delay. MAUI creates two versions of smart phone application, for local and remote execution using Microsoft .NET Common Language Runtime (CLR). Independent methods can be marked for remote execution in MAUI programming environment. Secondly it uses dynamic methods to partition in order to reduce the burden on the programmers [1], [6].MAUI uses history for the offloading decisions. If a new independent task comes for execution it is not considered by MAUI. Secondly MAUI Profilers consumes energy, processing power and Memory that are already less in smart phones.

10. Ant Colony Optimization Algorithms

Workflow scheduling using Ant Colony Optimization is explained in by W. N. Chen, J. Zhang and Yang [6]. The ant colony optimization algorithm is one of the most vital algorithms to solve computational problems and it uses heuristics. Complex problems are reduced into simpler ones to find out the good optimized path through the graphs. This algorithm is based on the fact of how ants find a path between their colony and the source of food. An ant colony optimization algorithm is a Meta-heuristic one. The implementation of this algorithm involves a no of concurrent steps. The heuristic information has to be initialized first, followed by the generation of ants. Next we need to map the ants with the path and the objective function has to be evaluated.

11. Genetic Algorithms

The use of genetic algorithms to schedule scientific workflows with budget and deadline constraints [7]. Genetic algorithm belongs to the class of evolutionary algorithms. It produces an optimization solution in polynomial time using various evolutionary techniques from a large search space. The fitness function plays a key role in this type of algorithm which determines how much an organism can reproduce before it dies. Genetic algorithms make use of operations like mutation, recombination and selection. A problem which need an optimized solution and that has to be implemented within the specified deadline depends widely on genetic algorithms.

12. Particle Swarm Optimization (pso) Algorithms

The describes how Particle swarm Optimization is being used efficiently to schedule workflows. This is a population based stochastic optimization algorithm developed by Dr. Beernaert and Dr. Kennedy in 1995 given great ideas from social behavior of bird travelling in (a large group) or fish schooling. PSO shares many homogenous attributes with genetic algorithms, but it doesn't have any evolutionary operators. The existing workflow scheduling algorithms on deadline constraints have been tabulated.

$\Pr(X\pi u \geq V) = \_\pi\in\Pi h$

This optimization is a case of domination based pruning as described in define a new operator in highlight node.

Current optimum prarticles consumes offloading.

$\Pr(X\pi u \geq V \mid \pi u \sim \pi) \Pr(\pi u \sim \pi)$
$= \_\pi\in\Pi h$
$\Pr(X\pi \geq V) \Pr(\pi u \sim \pi)$.

The PSO algorithm works by initializing a population (swarm of) candidate solutions (known as particles). Each particle will keep track of its best solution named as personal best (pbest) and also the best value of any particle called as global best (guest). The particles fly through the quandary space by following the current optimum particles. Every particle changes its position based on its current position, speed. To add on, the distance between the recent position and pbest, the distance between the current position and gbest are also considered.

13. Partial Critical Path (pcp)

The brief description of the assignment of PCP (partial critical path) is detailed in algorithm one vertices and an edge connects variables that appear in the scope of same constraint function. (IaaS Cloud Partial Critical Path) algorithm schedules the workflow applications in a public cloud environment. It implements the workflows. Bellow shows the fig13.1 partial critical path graph.

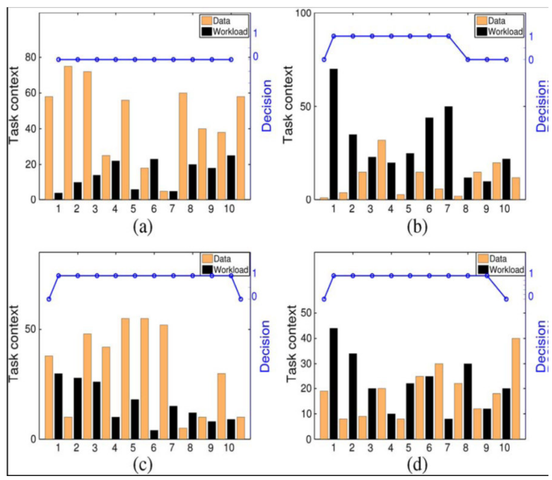

Fig. 13.1 Partial Critical Path (PCP)

This algorithm works by assigning the tasks of the workflow to the Partial Critical Path (PCP). The PCP (Partial Critical Path) will initially hold the parent of one of the exit task which was not already scheduled. If it was already scheduled, the task with the latest finish time will be chosen and added to the Partial Critical Path (PCP). the mobile device under the stochastic wireless channel, compared to the local execution and the remote execution. Deadline depends on execution. Optimized solution and that has to be implemented within the specified deadline depends widely on genetic algorithms. Problems are the existing workflow scheduling algorithms on deadline constraints have been tabulated reduced into simpler ones to find out the good optimized path through the graphs.

14. Dynamic Programming Optimization

As constraints for the abstraction model give a mapping to the space of feeder trees, existing dynamic programming algorithm DPOP [8] cannot be used. The algorithm we propose is based on DPOP, but changes its bottom up until propagation phase to deal with feeder trees. The primal graph of a constraint network is an undirected graph with variables as vertices and an edge connects variables that appear in the scope of same constraint function. Our algorithm works on the DFS traversal of the primal graph for the given constraint network. DFS trees have already been investigated as a means to boost search [5, 4]. Due to the relative independence of nodes lying in different branches of the DFS tree, it is possible to perform search in parallel on independent branches, and then combine the results. In a DFS tree, P (Xi) refers to the parent of the node Xi, C (Xi) to the children, PP (Xi) to the pseudo parents Of Xi and Semi to the separator of Xi. For DFS related definitions [9]. As with DPOP, our algorithm has three phases. We will describe in detail only the UTIL phase, which is significantly different from DPOP. Phase 1 - a DFS traversal of the graph is generated using a distributed algorithm [10]. The outcome of this protocol is that all nodes consistently label each other as parent/ child or pseudo parent or pseudo child, and edges are identified as tree/back edges. The DFS tree thus obtained serves as a communication structure for the other 2 phases of the algorithm. Phase 2 - UTIL propagation.

This is a bottom up pass on the DFS arrangement in which the utility information is aggregated and propagated from the leaves towards the root (in the form of UTIL messages), from each node to its parent through tree edges but not back edges. A UTIL message sent by a node Xi to its parent P (Xi) is a multidimensional matrix which informs P (Xi) how much utility the subtree rooted at Xi provides for different assignments of values to the variables that define the separator Sepi for the subtree. In our case, the utility for an assignment u is the feeder tree and its cost i.e. util (u) = (_, −c_) (costs can be treatedas negative utilities). In more detail, the agents perform

Cite this article as: S Ganapathi Ammal, M Suresh, S Mani, S Senthil kumar, M Senkottaiyan. "An Efficient Power Aware Mobile Job Execution in Public Mobile Cloud". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 267-273. Print.

following steps: 1. Wait for UTIL messages from all their children, and store them. 2. Perform an aggregation: join messages from children, and also the relations they have with their parents and pseudo parents. 3. Perform an optimization: project them out of the resulting join by picking their optimal values for each combination of values of the other variables in the join.4. Send the result to parent as a new UTIL message. Aggregations apply the JOIN operator and optimization applies the PROJECT operator as described below. Let UTILj I denote the message sent by Xi to its parent Xj and dim (UTILj I) denote its dimensions i.e. the set of variable defining this message. Definition 1 (JOIN). The _ operator: UTILj UTILjk is the join of two UTIL matrices.

The dimensions asdim (UTILj I) dim (UTILj k). The value of each cell in the join is obtained by applying the P operator to the corresponding cells in the source matrices. The above join operator easily extends to feeder trees as we have defined the P operator which combines feeder trees. Definition 2 (PROJECT). The Operator: if Xi 2 dim (UTILj I), then UTILj I? Xi is the projection through optimization of the Utile I matrix along the Xi axis: for each instantiation u of the variables in {dim (UTILj I) − Xi}, all the corresponding values from Utile I (one for each value of Xi) are tried, and the one which gives maximal utility (or\ minimal cost) is chosen. That is, the utility until? (u) For the instantiation u in the matrix Utile I? Xi is given by until? (u) = maxi utile I (u, x).we can apply the above projection operator only in the case of complete feeder trees (which have all leaves as sinks). This optimization is a case of domination based pruning as described in define a new operator in highlight node.

15. Conclusion and Feature Work

Collaborative task execution as a constrained stochastic shortest path problem over an acyclic graph, witha constraint of a hard time deadline or a probabilistic time deadline. Optimal one climbs policy and proposed an enumeration algorithm, followed by a set of necessary conditions for optimal task scheduling. Proposed an adapted LARAC algorithm to obtain the energy efficient scheduling policy for collaborative task execution. Workflows have been gaining the main focus scheduling by moving it from Grid computing to Cloud computing. Related work on the existing workflow scheduling algorithms on deadline constraints have been tabulated based on their domain and type of algorithm which aims to satisfy the Quality of Service (Quos) of the user. a new algorithm called LARAC for the well known Delay Constrained Least Cost problem, which is the most common QoS routing problem when delay sensitive traffic should be considered. The improved LARAC algorithm running time is only slightly greater than that of CBF, and due to its stability in all kind of networks, it can be a reasonable algorithm to solve the DCLC problem.

In this proposed constraint experiment to prove that Mobile cloud computing can save Energy as well as time of the smart phone by offloading its tasks to the cloud using high speed stable internet connections. We also conclude that in mobile cloud computing As performing a task on the device itself consumes the main memory, Processor or CPU as well as the energy of the smart phone, Computation Offloading is a Better alternative to it. Framework Energy and Time is directly proportional to the speed of the Internet connection and its stability. In addition, we will investigate power saving mode and power adaptation on mobile devices to further reduce energy consumption while bringing little application performance penalty.

References

1. Weiwen Zhang, Yonggang Wen, Senior Member, IEEE, and Dapeng Oliver Wu, Fellow, IEEE.Co S. M. Huan Xu, "Probabilistic goal Markov decision processes," in Proc elaborative Task Execution in Mobile Cloud Computing Under a Stochastic Wireless Channel. ieee transactions on wireless communications, vol. 14, no. 1, 2015
2. S. Abrishami, M. Naghibzadeh, and D. H. Epema, "Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds," Future Gener. Comput. Syst., vol. 29, no. 1, pp. 158–169, Jan. 2013.
3. D. P. Bertsekas, Dynamic Programming and Optimal Control, vol. 2. Belmont, MA, USA: Athena Scientific, 2001.
4. S. M. Huan Xu, "Probabilistic goal Markov decision processes," in Proc .Int. Joint Conf. Artif. Intell, 2011, pp. 2046–2052.
5. A. Juttner, B. Szviatovski, I. Mécs, and Z. Rajkó, "Lagrange relaxation based method for the QoS routing problem," in Proc. IEEE INFOCOM, 2001, vol. 2, pp. 859–868.
6. A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in Proc. 2nd USENIX Conf. Hot Topics Cloud Computer., 2010, p. 4.
7. L.Johnston and V.Krishnamurthy, "Opportunistic file transfer over a fad in channel: A POMDP search theory formulation with optimal threshold policies," IEEE Trans. Wireless Common., vol. 5, no. 2, pp. 394–405, Feb. 2006.
8. Z. Wang and J. Crow croft, "Quality-of-service routing for supporting multimedia applications," IEEE J. Sel. Areas Common., vol. 14, no. 7,pp. 1228–1234, Sep. 1996
9. A. Wald, "On cumulative sums of random variables," Ann. Math. Stat., vol. 15, no. 3, pp. 283–296, Sep. 1944. A. Wald, "On cumulative sums of random variables," Ann. Math. Stat., vol. 15, no. 3, pp. 283–296, Sep. 1944.
10. M. Zafer and E. Modiano, "Minimum energy transmission over a wireless fading channel with packet deadlines," in Proc. IEEE Conf. Decision Control, 2007, pp. 1148–1155.

Cite this article as: S Ganapathi Ammal, M Suresh, S Mani, S Senthil kumar, M Senkottaiyan. "An Efficient Power Aware Mobile Job Execution in Public Mobile Cloud". *International Conference on Systems, Science, Control, Communication, Engineering and Technology 2016*: 267-273. Print.