Scheduling in Vehicular Cloud Using Mixed Integer Linear Programming

Puya Ghazizadeh
Department of Computer Science
Millersville University
Millersville, Pennsylvania
Puya.Ghazizadeh@millersville.edu

Ravi Mukkamala
Department of Computer Science
Old Dominion University
Norfolk, Virginia
mukka@cs.odu.edu

Samy El-Tawab
Department of Integrated Science and Technology
James Madison University
Harrisonburg, Virginia
eltawass@jmu.edu

ABSTRACT

Statistics show that most vehicles spend many hours per day in a parking garage, parking lot, or driveway. At the moment, the computing resources of these vehicles are untapped. Inspired by the success of conventional cloud services, a group of researchers have recently introduced the concept of a Vehicular Cloud. In this model each vehicle is a computation node. The main difference between traditional cloud computing and vehicular cloud computing is in availability of nodes. In vehicular cloud as opposed to traditional cloud nodes are not available all the time. Random arrival and departure of vehicles create a dynamic environment in terms of resources availability. We present a scheduling model for vehicular cloud based on mixed integer linear programming. This model uses migration in order to prevent interruptions that may be caused by random departure of vehicles.

Categories and Subject Descriptors

H.4 [**Vehicular Clouds**]: Vehicular Networks, Cloud Computing, Linear Programming

Keywords

cloud computing; vehicular clouds; scheduling; linear programming; resource management

1. INTRODUCTION

Nowadays, vehicles are beyond transportation machines. In fact they are the machines with high computing powers. People can get different services (navigation, weather forecast, entertainment, etc) while they are in the vehicle as a driver or passenger. In the other hand advances of cloud computing have encouraged companies and users to move their computations, IT services or digital entertainments to the cloud. The main factor of cloud computing is computation power. Vehicles in parking lots could be potential resources for computation power [1–3].

Olariu et al through series of researches introduced a new concept Vehicular Cloud Computing [4–6]. Vehicular Cloud (VC) is a network of vehicles in parking lot that can provide computation services to users. In this model each vehicle is a computation node. Some of the applications of VC are: datacenter at the airport, data cloud in parking lot and datacenter at the mall.

Traditional cloud includes large numbers of servers. These servers are established in cloud infrastructure and are available to provide service anytime unless there is a failure. Failure can be from software or hardware. In other words traditional cloud has static environment in term of computation power unless there is a failure. In contrast VC has dynamic computation nodes. Vehicles which are the computation nodes can arrive or depart any period of time which makes VC a dynamic environment in terms of computation power. Random arrival and departure of vehicles may interrupt the jobs that are running. In this paper we present a model for scheduling in VC that prevents interruption in job execution. This model is based on migration.

The remainder of this paper is organized as follows. In Section 2, we describe migration and different scheduling approaches. In Section 3, we present our approach with an example. Section 4 gives the details of implementation and experiment result. Conclusions is given in section 5. Future work is presented in section 6.

MSCC'14, August 11, 2014, Philadelphia, PA, USA.

http://dx.doi.org/10.1145/2633675.2633681.

2. RELATED WORK

2.1 Virtualization

Virtualization, in computing, is a simulating techniques that provide a layer between hardware and operating system. This layer can provide a virtual resources (e.g., CPU, storage, network, etc). In other words virtualization provides a logical view of resources rather than physical view. The main idea of virtualization is to utilize the resources. Virtual machine (VM) is the main factor of cloud computing. In other words cloud computing has been developed based on VM technology.

IT administrators use virtual machine migration in data centers and clusters to manage load balance, handle failure and lower maintenance cost [7]. Virtualization is a solu-

tion for migrating computation between different physical servers. There are two types of virtual machine migration: Live Migration and Cold Migration. In live migration operating system and application can be transferred to different physical machines while they are running. In cold migration operating system and application are suspended before transferring to physical machines. In our approach, we use live migration in order to transfer VMs to different physical machines. In our model each vehicle is physical machine or computation node.

2.2 Scheduling

Scheduling problem has been a topic for computer science operations research from mid-twenty century. Job scheduling problems are defined as allocating limited resources to jobs in order to optimize the objective function. In terms of computational complexity, most of the job scheduling problems are classified as NP-hard problems. NP-hard problem can be solved in polynomial time using a non-deterministic Turing machine. Different solutions have been presented for scheduling: artificial intelligence, priority rules, heuristic algorithms (e.g., GA, PSO, etc) and operational research which includes simplex method, cutting plane, branch and bound.

2.2.1 Operational Research

In operational research method, problem is separated from solving algorithm. Problem should be defined based on mathematical model and solving algorithm (e.g., simplex method) is used to solve the problem.

2.2.2 Priority Rules

There are different types of rule-based algorithms like First Come First Service (FCFS). Shortest Processing Time (SPT). Longest Processing Time (LPT) etc. Some of these algorithms have been used in CPU scheduling. Priority rules are efficient in terms of execution complexity but in some of the large and complex environment optimal solution can not be found by using priority rules [8]. Priority rules are extracted based on analyzing different experiments. Therefore efficiency of the rules depends on the experiments. priority rules cannot be adapted to all environments [9].

2.2.3 Heuristic Algorithms

Heuristic algorithms have been developed based on the biology principles. Based on these principle solutions should evolve during each generation. Evolution of the solution leads to optimal or near optimal solution. Genetic algorithm (GA), taboo search algorithm and simulated annealing algorithm are examples of heuristic algorithms.

2.2.4 Artificial Intelligence

Neural Network and constraint programming are examples of artificial intelligence scheduling method [10]. One of the advantages of artificial intelligence method is considering the changes in environment and responding to these changes.

There are combined methods which apply two different scheduling algorithms like priority rules and heuristic algorithms in order to create an efficient scheduling algorithm. In our approach we use linear programming model that belongs to operational research class.

2.3 Linear Programming

Linear programming has been introduced in early mid-20th century [11] and is highly structured mathematical model. In this model objective function and constraints are linear. Each constraint can be a linear equality or inequality. Objective function is a mathematical function of decision variables that measures the performance of the solution. Based on the decision variables, model can be classified into linear programming, integer linear programming or mixed integer linear programming. In linear programming [11] decision variable are non-integers. Integer programming is another version of linear programming that decision variables are integer values. The model that some of the decision variables are integer and some of them are non-integer is called mixed integer programming. Both integer programming and mixed integer programming mathematical models are linear programming model. In linear programming solving algorithm and problem model are separate form each other. In other words by changing the problem model, there is no need to change the solving algorithm. This feature makes linear programming a flexible approach. Different version of problem can be defined by adding or modifying the equations without changing the solving algorithm.

2.4 Migration

In a typical vehicular cloud (e.g. airport parking lot) vehicles arrive and depart in different period of time. As we mentioned vehicles are the computation resources in vehicular cloud. Therefore computation resources are dynamic in terms of availability. In order to handle the interruption which is caused by vehicle's departure, we use migration. In VC migration is a process of transferring a job from one vehicle to another vehicle or server. We present an approach for job scheduling in dynamic environment in terms of computing resources. Our model is based on mixed integer linear programming.

3. OUR APPROACH

3.1 Problem Definition

Our scheduling problem can be defined as a parallel machine scheduling. In this type of problem, jobs are assigned to several parallel identical machines. This type of scheduling includes combination and permutation. Permutation is assigning jobs to machines and combination is for defining different sequence of jobs on each machine. There are m identical parallel machines and n jobs with known processing time. This work investigates a job scheduling problem involving non-preemptive tasks with known processing time where job migration is allowed. Assigning a job to resources is valid if job has been executed fully and continuously (no interruption). A job can not be executed in parallel. In this work resources are dynamic in terms of availability.

In our approach, the determination of an optimal job schedule can be formulated as maximizing the utilization of VC and minimizing the number of job migrations. Utilization can be calculated as a time period that vehicles have been used as computation resources.

3.2 0-1 Model Description

In this model we define a decision variable X_{ijk}. i is the index for jobs. j is the index for hour slots. Each parking spot has an ID and k is the index for parking spots in parking lot. X_{ijk} value is one if job i has been assigned to a vehicle in parking spot k at time slot j. Otherwise value for X_{ijk} is zero. In this model parking lot has constant number of parking spots. The first part of the objective function for this model is the following:

$$\sum_{ijk} X_{ijk} * (PMAX)_{jk} \tag{1}$$

PMAX is a matrix that shows the parking occupancy. Each element of this matrix can have value either 1 or -1. Value 1 means that a vehicle will be available as a computation node in parking spot j and hour slot k. Value -1 means in parking spot k and hour slot j vehicle is not available. Constraint 2 is for defining that each parking spot in a specific hour slot can not have more than one job.

$$\forall j,k \qquad \sum_{i} X_{ijk} \leq 1 \tag{2}$$

D_i is the duration of job i. Constraint 3 shows that a job will be assigned to resources if it is executed fully. Otherwise it will not be assigned.

$$\forall i \qquad \sum_{j,k} X_{ijk} - D_i * y_i = 0 \tag{3}$$

y_i is for implementing OR operator and can have value 0 or 1. Constraint 4 shows that a job can be executed only on one parking spot in specific hour slot (No parallelization).

$$\forall i,j \qquad \sum_{k} X_{ijk} \leq 1 \tag{4}$$

Constraint 5 determines the starting time of job i.

$$\forall i,j,k \qquad X_{ijk} * S_i \leq j * X_{ijk} \tag{5}$$

S_i is starting time for job i.
starting time of jobs are used in constraint 12 to enforce continues execution of jobs. Constraint 5 is not linear constraint. We replace a set of linear constraints that enforce constraint 5. Followings are constraints that are used to replace constraint 5. We define a new variable W_{ijk} that replaces the product of two variables X_{ijk} and S_i.

$$W_{ijk} = X_{ijk} * S_i \tag{6}$$

$$W_{ijk} - X_{ijk} * j \leq 0 \tag{7}$$

$$W_{ijk} - X_{ijk} * NHS \leq 0 \tag{8}$$

$$W_{ijk} - S_i \leq 0 \tag{9}$$

$$S_i - NHS + NHS * X_{ijk} - W_{ijk} \leq 0 \tag{10}$$

$$W_{ijk} \geq 0 \tag{11}$$

NHS is number of hour slots. following constraint is for executing the units of job continuously.

$$\forall i \qquad \sum_{j,k} X_{ijk} * j - S_i D_i = 0.5(D_i)^2 - 0.5D_i \tag{12}$$

Equation 13 is second part of objective function. In our model we minimize equation 13 in order to minimize the number migrations.

$$\forall i \qquad \sum_{i} (Max_i - Min_i) \tag{13}$$

Max_i is the variable that defines the parking spot with the highest ID number which has been assigned to job i. Min_i is the variable that defines the parking spot with the lowest ID number which has been assigned to job i. In second part of objective function we try to minimize the distance between Max_i variable and Min_i variable. By minimizing the second part of this objective function, number of the parking spots that have been assigned to job i will decrease. This will decrease the number of migrations. In other words a job tends to be executed on minimum number of parking spots. Here we define the Min_i and Max_i:

$$Min_i * X_{ijk} \leq X_{ijk} * k \tag{14}$$

$$Max_i * X_{ijk} \geq X_{ijk} * k \tag{15}$$

Constraint 14 will enforce the definition of lowest parking spot ID to Min_i. Constraint 15 will enforce the definition of highest parking spot ID to Max_i. Since constraint 14 and 15 are not linear we define a set of constraints to convert constraint 14 and 15 to linear constraints. We define a new variable Q_{ijk} that replaces the product of two variables X_{ijk} and Min_i. Following are the constraints for converting constraint 14 to linear constraint:

$$Q_{ijk} = X_{ijk} * Min_i \tag{16}$$

$$Q_{ijk} - X_{ijk} * k \leq 0 \tag{17}$$

$$Q_{ijk} - X_{ijk} * NM \leq 0 \tag{18}$$

$$Q_{ijk} - Min_i \leq 0 \tag{19}$$

$$Min_i - NM + NM * X_{ijk} - Q_{ijk} \leq 0 \tag{20}$$

$$Q_{ijk} \geq 0 \tag{21}$$

NM is number of machines. We define a new variable R_{ijk} that replaces the product of two variables X_{ijk} and Max_i. Following are the constraints for converting constraint 15 to linear constraint:

$$R_{ijk} = X_{ijk} * Max_i \tag{22}$$

$$-R_{ijk} + X_{ijk} * k \leq 0 \tag{23}$$

$$R_{ijk} - X_{ijk} * NM \leq 0 \tag{24}$$

$$R_{ijk} - Max_i \leq 0 \tag{25}$$

$$Max_i - NM + NM * X_{ijk} - R_{ijk} \leq 0 \tag{26}$$

$$R_{ijk} \geq 0 \tag{27}$$

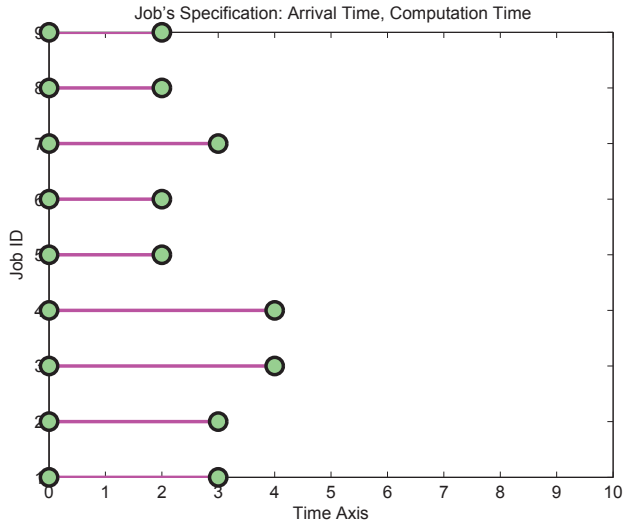

Figure 1: Scheduling 0-1 Model (Jobs' Specification)

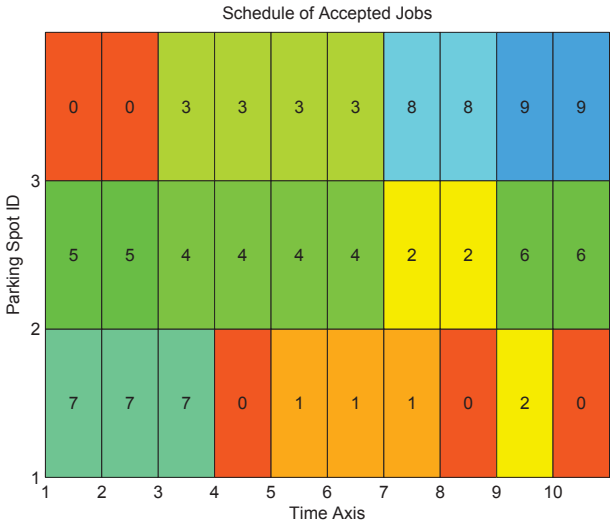

Figure 3: Scheduling 0-1 Model (Result)

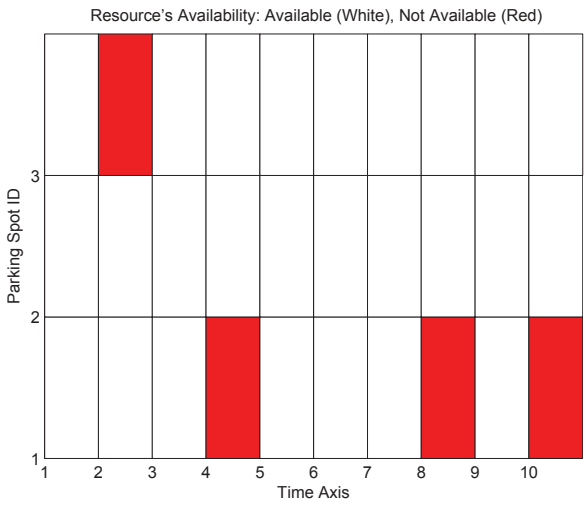

Figure 2: Parking Occupancy Model

4. IMPLEMENTATION AND EXPERIMENT RESULTS

We implemented the proposed MILP model in Matlab and utilized the Gurobi optimization [12] through its Matlab interface to solve the scheduling problem in VC. Our model prevents interruptions that may occur due to departure of vehicles by using live migration. Figure 1-3 shows a scheduling problem for nine jobs. Figure 1 is job specifications that shows duration of each job. Figure 2 shows the occupancy of three parking spots during ten hours. Color red defines that there is not vehicle in that parking spot for that specific hour. Figure 3 is the output of the scheduling model for jobs considering three parking spots during ten hour slots. Jobs are distinguished by their ID that are shown in Figure 3. Zero means there is no job assigned to that parking spot in that specific hour slot. Result shows that there is a migration for the job with ID two at the end of hour slot eight from parking spot two to parking spot one.

5. CONCLUSION

Vehicular clouds are motivated by the abundant computational resources in present-day vehicles and the fact that most of these vehicles are parked every day, for hours on end, while their owner is working, shopping, travelling, etc. Given the huge number of vehicles on our roads and city streets, vehicular clouds are expected to have a huge societal impact. One of the main differences between vehicular cloud and traditional cloud is in resource availability. In this paper we presented a model for scheduling based on mixed integer linear programming that uses migration to handle interruption in dynamic environment. Simulations have shown that optimum and near optimum solutions are found.

6. FUTURE WORK

We plan to investigate future work as follows: 1) considering stochastic arrival and departure of vehicles 2) decreasing the freedom degree of scheduling problem by defining arrival time and deadline for jobs.

7. REFERENCES

[1] M. Fontaine, *Traffic monitoring: in Vehicular Networks: From Theory to Practice*, S. Olariu and M. C. Weigle, Eds. Taylor and Francis, 2009.

[2] S. Olariu and M. C. Weigle, *Vehicular Networks: From Theory to Practice*. CRC Press, Boca Raton, 2009.

[3] X. Wu, P. Michalopoulos, and H. X. Liu, "Stochasticity of freeway operational capacity and chance-constrained ramp metering," *Transportation Research Part C*, vol. 18, pp. 741–756, 2010.

[4] M. Abuelela and S. Olariu, "Taking vanet to the clouds," in *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, ser. MoMM '10. New York, NY, USA:

ACM, 2010, pp. 6–13. [Online]. Available: http://doi.acm.org/10.1145/1971519.1971522

[5] S. Arif, S. Olariu, J. Wang, G. Yan, W. Yang, and I. Khalil, "Datacenter at the airport: Reasoning about time-dependent parking lot occupancy," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 11, pp. 2067–2080, 2012.

[6] S. Olariu, I. Khalil, and M. Abuelela, "Taking vanet to the clouds," *International Journal of Pervasive Computing and Communications*, vol. 7, no. 1, pp. 7–21, 2011.

[7] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield, "Live migration of virtual machines," in *Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation - Volume 2*, ser. NSDI'05. Berkeley, CA, USA: USENIX Association, 2005, pp. 273–286. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251203.1251223

[8] S.-C. Lin, E. D. Goodman, and W. F. Punch, "A genetic algorithm approach to dynamic job shop scheduling problems," in *Seventh International Conference on Genetic Algorithms*. Morgan Kaufmann, 1997, pp. 139–148.

[9] H. Zhou, Y. Feng, and L. Han, "The hybrid heuristic genetic algorithm for job shop scheduling," *Computers and Industrial Engineering*, vol. 40, no. 3, pp. 191 – 200, 2001. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360835201000171

[10] Y. Chen, Z. Guan, and X. Shao, "A comparative analysis of job scheduling algorithm," in *Management Science and Industrial Engineering (MSIE), 2011 International Conference on*, 2011, pp. 1091–1095.

[11] F. Hillier and G. Lieberman, *Introduction to Operations Research*, ser. Introduction to Operations Research. McGraw-Hill Higher Education, 2010. [Online]. Available: http://books.google.com/books?id=NvE5PgAACAAJ

[12] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2013. [Online]. Available: http://www.gurobi.com