2019 IEEE 12th International Conference on Cloud Computing (CLOUD)

Towards Approximating Expected Job Completion Time in Dynamic Vehicular Clouds

Aida Ghazizadeh*, Puya Ghazizadeh†, Ravi Mukkamala* and Stephan Olariu*
*Department of Computer Science, Old Dominion University, Norfolk, VA 23529, {aghaziza, mukka, olariu}@cs.odu.edu
†Department of Mathematics and Computer Science, Seton Hall University, South Orange, NJ 07079, puya.ghazizadeh@shu.edu

***Abstract*—Motivated by the success of cloud computing, vehicular clouds were introduced as a group of vehicles whose corporate computing, sensing, communication, and physical resources can be coordinated and dynamically allocated to authorized users. Our main contribution is to offer an easy-to-compute approximation of job completion time in a dynamic vehicular cloud model involving vehicles on a highway. We assume estimates of the first moment of the time it takes the job to execute without any overhead attributable to the working of the vehicular cloud. Our simulations have shown that our approximation is very accurate. To the best of our knowledge, this is the first paper dealing with estimating job completion time in dynamic vehicular clouds.**

***Keywords*-Vehicular clouds, job completion time.**

I. Introduction and Motivation

Inspired by the success of Cloud Computing, a number of papers have introduced the concept of a *Vehicular Cloud*, (VC). In their seminal paper, Ghazizadeh *et al.* [1], [2] have argued that in the near future the vehicles on our roads and city streets will self-organize into VCs utilizing their corporate resources on-demand and largely in real-time for resolving critical problems that may occur unexpectedly.

VCs are characterized by resource volatility [3]: as vehicles enter the cloud new compute resources become available; when vehicles depart, often unexpectedly, they take their resources with them creating a volatile environment where reasoning about job completion time is very challenging [4]–[7]. Predicting job completion time requires full knowledge of the probability distributions of the intervening random variables. In practice, the datacenter manager does not know these distributions. Instead, using accumulated empirical data, she can estimate the first moments of these random variables.

Our main contribution is an easy-to-compute approximation of job completion time in a VC model involving vehicles on a highway. We assume estimates of the first moment of the time it takes the job to execute in the absence of any overhead attributable to the working of the VC. Our extensive simulations have shown the accuracy of our approximations.

II. The VC and workload model

We envision a *dynamic* VC that is harnessing the compute power of vehicles moving on a highway. In order to implement this idea, the VC controller is connected by optical fiber to pre-installed *access points* (APs, for short) deployed along the highway. Referring to Figure 1, the access points are placed d meters apart along the highway and are numbered consecutively as $AP_0, AP_1, \cdots, AP_n$. As illustrated in Figure 1, each AP has a radio coverage area of c meters. A vehicle can communicate with an AP only while under its coverage area. Each AP continuously sends out frames of size F bits containing a fixed length payload of b bits. In each frame, vehicles that are under the coverage area and wish to communicate with the AP compete to secure a communication slot in that frame. From the standpoint of a given vehicle, a frame is *successful* if the vehicle has secured a slot in that frame. In each frame, the payload of b bits is partitioned evenly among the successful vehicles in that frame. We let p_{k+1} denote the conditional probability of successful slot acquisition given that the vehicle competes with k other vehicles in a given frame. We assume a communication bandwidth of $B = 27$MBps which is the maximum data transmission rate in DSRC.

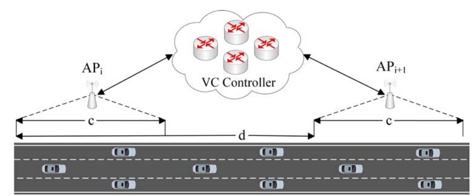


Fig. 1. *Illustrating two consecutive APs and their coverage areas.*

A vehicle that just entered the highway informs the first AP it encounters, say AP_i, of the last access point AP_j before it exits. Given this information, and the average speed on the highway between AP_i and AP_j, the VC controller can estimate the amount of time the vehicle will spend on the highway and the workload that can be allocated to the vehicle. Jobs are encapsulated as container images. The vehicle will begin by downloading the corresponding container image, will execute the job and, upon termination, will upload the results to the first available AP. In case the vehicle leaves the highway before completing job execution, the corresponding container will have to be migrated to another vehicle, using one of several migration strategies. In this work we assume workloads that do not require migration.

The speed v_{k+1} of a vehicle is a function of traffic density ρ_{k+1} corresponding to the presence of $k+1$ vehicles in the coverage area of an AP [8].

The workloads for the VC discussed in this paper are delay-tolerant, with small input size, short processing time, and no inter-processor communication. A typical example is machine translation of short documents such as memoranda, personal and business letters, job applications, newsletters, resumes, and the like. Given these workloads, we assume that (a) the size of the downloaded container is (roughly) equal to the size of the uploaded result; (b) the container can be downloaded/uploaded under the coverage of one AP; (c) the job can be completed by one vehicle with no need for container migration.

2159-6190/19/$31.00 ©2019 IEEE
DOI 10.1109/CLOUD.2019.00084

Let W be the size of the container image (in bits) and let the random variable r_{k+1} denote the number of successful frames necessary to download (or upload) the corresponding job. Clearly, $r_{k+1} = \frac{W}{\frac{b}{(k+1)p_{k+1}}} = \frac{(k+1)Wp_{k+1}}{b}$. For all k, $(k \geq 0)$, let the random variables D_{k+1} (resp. U_{k+1}) represent the *total number of frames* in which the vehicle has to compete, in order to complete the download (resp. upload) of a job, given that in each frame k other vehicles are also competing to for a slot. D_{k+1} and U_{k+1} have a negative binomial distribution with parameters r_{k+1} and p_{k+1} and so $E[D_{k+1}] = E[U_{k+1}] = \frac{r_{k+1}}{p_{k+1}} = \frac{W(k+1)}{b}$. It is interesting to note that $E[D_{k+1}]$ is independent of the success probability p_{k+1} and only depends on the size W of the container image to download, the payload b, and the traffic density $k+1$.

III. Approximating job completion time

The notation used in the paper is summarized below.

Symbol and Description	Value
l (number of lanes)	3
B (available bandwidth)	27×10^6 bps
W (size of the job)	8×10^6 bits
b (payload in one frame)	53792 bits
F (frame length in bits)	56624 bits
F_s (frame length in seconds)	0.002 s
M (number of available slots for competing)	20
c (access point coverage range)	100 m
d (distance between two consecutive APs)	2000 m
a (parameter used for intervals of job processing time)	600 s
v_{k+1} (vehicle's speed when k other vehicles are in the area)	(10, 30) m/s
v_b (average travel speed at stop and go condition)	9 kph
v_f (free flow speed)	107.44 kph
k_t (turning point for the speed-density curve)	17.53
θ_1 (scale parameter for speed-density function)	1.8768
θ_2 (parameter which controls the lopsidedness of the curve)	0.0871

Let T denote the execution time of the user job in the absence of any overhead attributable to the VC. We do not assume knowledge of the probability distribution of T. Instead, we only assume $E[T]$ known. Similarly, let N be the subscript of the AP at which the results of job execution finish uploading. We distinguish between the three cases below.

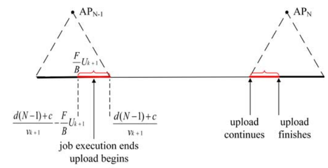
Case 1: *Job execution ends under the coverage of some AP; the results finish uploading under the coverage of the next AP.* Referring to Figure 2(a), job execution ends under the coverage of AP_{N-1} and the upload of the results begins immediately. The upload is interrupted when the vehicles leaves the coverage area of AP_{N-1} and resumes when the vehicle enters the coverage area of AP_N. Thus, job completion time J_1 reads

$$J_1 = \frac{F}{B}(D_{k+1} + U_{k+1}) + T + \frac{d-c}{v_{k+1}}, \tag{1}$$

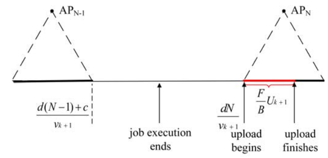
where $\frac{F}{B}D_{k+1} + T$ is the time to download and to execute the job, $\frac{d-c}{v_{k+1}}$ is the time to move between the coverage areas of AP_{N-1} and AP_N, and $\frac{F}{B}U_{k+1}$ is the combined time to upload the result. It follows that

$$E[J_1] = E[T] + \frac{2F}{B}E[D_{k+1}] + \frac{d-c}{v_{k+1}}. \tag{2}$$

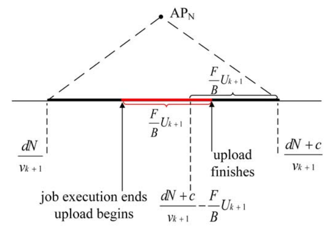
Case 2: *Job execution terminates between two coverage areas.* Referring to Figure 2(b), job execution ends between the coverage areas of AP_{N-1} and AP_N. Upon entering the coverage



(a) *Illustrating Case 1.*



(b) *Illustrating Case 2.*



(c) *Illustrating Case 3.*

Fig. 2. *Illustrating Cases 1 through 3.*

area of AP_N the upload begins. We define the job completion time J_2 as:

$$J_2 = \frac{dN}{v_{k+1}} + \frac{F}{B}U_{k+1}. \tag{3}$$

Notice that the vehicle must first physically reach the coverage area of AP_N before it can start uploading the results. The former time is $\frac{dN}{v_{k+1}}$, while the latter is $\frac{F}{B}U_{k+1}$. Here, N is the *unique* natural number for which job download and execution terminates strictly between the coverage areas of AP_{N-1} and that of AP_N, and so

$$\frac{(N-1)d+c}{v_{k+1}} < \frac{F}{B}D_{k+1} + T \leq \frac{Nd}{v_{k+1}}. \tag{4}$$

By applying the expectation operator to (3), we obtain

$$E[J_2] = \frac{dE[N]}{v_{k+1}} + \frac{F}{B}E[U_{k+1}]. \tag{5}$$

In order to obtain an expression for $E[J_2]$ we proceed as follows: From (4) by simple algebra we obtain, in stages,

$$\frac{F}{B}D_{k+1} + T < \frac{dN}{v_{k+1}} \leq \frac{F}{B}D_{k+1} + T + \frac{d-c}{v_{k+1}}. \tag{6}$$

Applying the expectation operator to (6) yields

$$\frac{F}{B}E[D_{k+1}] + E[T] \leq \frac{dE[N]}{v_{k+1}} \leq \frac{F}{B}E[D_{k+1}] + E[T] + \frac{d-c}{v_{k+1}}.$$

After adding $\frac{F}{B}E[U_{k+1}]$ throughout, we obtain

$$E[T] + \frac{2F}{B}E[D_{k+1}] \leq E[J_2] \leq E[T] + \frac{2F}{B}E[D_{k+1}] + \frac{d-c}{v_{k+1}},$$

which yields the following approximation for $E[J_2]$

$$E[J_2] \approx E[T] + \frac{2F}{B}E[D_{k+1}] + \frac{d-c}{2v_{k+1}}. \tag{7}$$

Case 3: *Job execution and result upload complete under the coverage of the same AP.*
Referring to Figure 2(c), job execution ends under the coverage area of AP_N and the results finish uploading in the same coverage area. In this case, we define job completion time J_3 as:

$$J_3 = \frac{F}{B}\left[D_{k+1} + U_{k+1}\right] + T \tag{8}$$

and so:

$$E[J_3] = E[T] + \frac{2F}{B}E[D_{k+1}]. \tag{9}$$

We now combine the three cases discussed above into an approximation of job completion time, J. Let π_1, π_2 and π_3 be, respectively, the *limiting* probabilities of Case 1, Case 2 and Case 3. Thus, the expectation $E[J]$ of J becomes

$$E[J] = \pi_1 E[J_1] + \pi_2 E[J_2] + \pi_3 E[J_3]. \tag{10}$$

To evaluate the limiting probabilities π_1, π_2, π_3, consider the time interval I of length $\frac{d}{v_{k+1}}$ delimited by $\frac{d(N-1)+c}{v_{k+1}} - \frac{F}{B}E[U_{k+1}]$ and $\frac{dN+c}{v_{k+1}} - \frac{F}{B}E[U_{k+1}]$. Since the probability distribution of T is not known, to a first approximation, we assume that job execution ends uniformly at random in the time interval I. This implies that π_1, π_2, π_3 are given by the expressions $\pi_1 = \frac{FE[U_{k+1}]v_{k+1}}{Bd}$, $\pi_2 = \frac{d-c}{d}$, and $\pi_3 = \frac{c}{d} - \pi_1$.
Upon replacing the expressions of π_1, π_2, π_3 above into (10), we obtain our approximation of job completion time:

$$E[J] \approx E[T] + \frac{(3d-c)FE[D_{k+1}]}{Bd} + \frac{(d-c)^2}{2dv_{k+1}}. \tag{11}$$

IV. Simulation Model and Results

We have simulated a three-lane highway with APs placed every 2000 meters. Each AP has a coverage area of 100 meters. The APs continuously send out frames of length $F = 56624$ bits with a payload of $b = 53792$ bits. The size of containers was $W = 1$ MB. To determine vehicular speed as a function of density, we have used a five-parameter logistic speed-density function of [8]. The average processing time $E[T]$ was between 20 and 30 minutes. Upon receiving a job, the vehicle starts downloading the container image. Job execution starts immediately after download and once job execution is completed, the vehicle immediately attempts to upload the results. The process of uploading the results is similar to download, in terms of competing for transmission slots. If a vehicle is not under the coverage area at the time, it attempts to upload the results at the next AP. We record the job completion time from the moment that the job is assigned to a vehicle until the results are uploaded. We have assumed that the residency time in the VC exceeds job completion time.

Due to space limitations, in Figure 3 we only show the simulation results for $k = 7$ and 8, against the predicted values in (11). The maximum relative error is less than 0.24%, with an average of 0.05% for uniform distributed job duration, and less than 1.96%, with an average of 0.1% for exponential distribution.

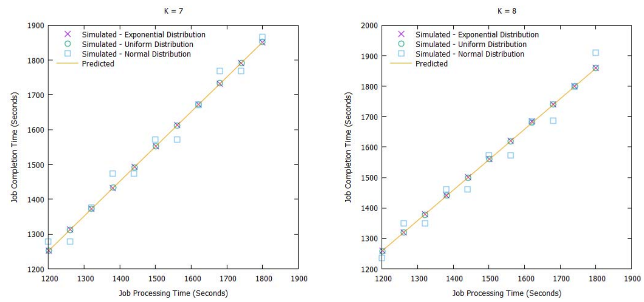


Fig. 3. *Simulated and predicted expected job completion times for $k = 7$ and $k = 8$.*

V. Directions for future work

In future work, we will look at workloads where the underlying VM cannot be downloaded under the coverage area of a single AP and needs several APs to complete this operation. Of great interest are scenarios involving short vehicular residency times, where VMs (or containers) need to be migrated to other vehicles as well as benefits of reinforcement-learning and other similar techniques [9] for resource provisioning in dynamic VCs [10]. Finally, security and privacy are important and will be addressed in future work.

References

[1] A. Ghazizadeh and P. Ghazizadeh, “Design and analysis of a communication protocol for dynamic vehicular clouds in smart cities,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 1–6.

[2] A. Ghazizadeh, P. Ghazizadeh, and S. Olariu, “Reasoning about a communication protocol for vehicular cloud computing systems,” in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2019.

[3] P. Ghazizadeh and R. Fathi, “A propagation based model for communication in vehicular cloud,” in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, March 2018, pp. 295–299.

[4] P. Ghazizadeh, S. Olariu, A. Ghazi Zadeh, and S. El-Tawab, “Towards fault-tolerant job assignment in vehicular cloud,” *Proceecings IEEE SCC*, pp. 17–24, June 2015.

[5] A. Boukerche and R. E. De Grande, “Vehicular cloud computing: Architectures, applications, and mobility,” *Computer Networks*, vol. 135, pp. 171–189, 2018.

[6] A. Ghazizadeh and S. Olariu, “Vehicular clouds: A survey and future directions,” in *Cloud Computing for Optimization: Foundations, Applications, and Challenges*, B. Mishra, H. Das, S. Dehuri, and A. Jagadev, Eds. Springer, 2018, ch. 17, pp. 435–463.

[7] R. Florin, A. Ghazizadeh, P. Ghazizadeh, S. Olariu, and D. Marinescu, “Enhancing reliability and availability through redundancy in vehicular clouds,” *IEEE Transactions on Cloud Computing*, 2019.

[8] H. Wang, Q.-Y. Chen, and D. Ni, “Logistic modeling of the equilibrium speed-density relationship,” *Transportation Research Part A: Policy and Practice*, vol. 45, no. 6, pp. 554–566, 2011.

[9] R. Florin, P. Ghazizadeh, A. Ghazi Zadeh, R. Mukkamala, and S. Olariu, “A tight estimate of job completion time in vehicular clouds,” *IEEE Transactions on Cloud Computing*, pp. 1–1, 2018.

[10] P. Ghazizadeh, “Resource allocation in vehicular cloud computing,” PhD Thesis, Old Dominion University, July 2014.