Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

A SIMULATION MODEL FOR VOLUNTEER COMPUTING MICRO-BLOGGING SERVICES

Christopher Bayliss
Javier Panadero
Laura Calvet
Joan Manuel Marquès

IN3 – Computer Science Dept.
Universitat Oberta de Catalunya
Av. Carl Friedrich Gauss 5
Castelldefels, 08860, SPAIN

ABSTRACT

Micro-blogging services (MBSs) have become increasingly popular during recent decades but most have a poor reputation in terms of privacy of the user data. Volunteer computing enables the implementation of decentralized systems based on heterogeneous resources that are donated by volunteers. However, volunteer computing is characterized by the unreliability of the resources: users are under no obligation to remain online. In this context, we define the problem of designing a directory service policy for a distributed volunteer computing MBS (DVCMBS). This service relies on repositories donated by volunteers. It is managed by a centralized directory service, which stores replicas of the blogs of other users to ensure their online availability and allocates blog replicas to online repositories in order to maximize the availability of all blogs. Likewise, efficiency is essential. We describe a simulation model of a DVCMBS, which includes a parameterized directory service policy. Preliminary numerical results are discussed.

1 INTRODUCTION

Nowadays, micro-blogging online social networks such as Twitter and Tumblr are extremely popular (Statista 2019a; Statista 2019b). In these micro-blogging systems users may post short messages, share interests, and communicate with each other. Most popular micro-blogging systems have a poor reputation regarding protecting the privacy of the user data (Oukemeni et al. 2018). For this reason, decentralized social networks are recommended (Seong et al. 2010). This work proposes a hybrid pere-to-peer (P2P) (Darlagiannis 2005) micro-blogging service (MBS) which combines P2P and client server models, referred to as a distributed volunteer computing MBS (DVCMBS). A centralized directory service node coordinates the storage of blog replicas among the donated decentralized storage capacity. Data privacy is retained since data is ciphered and transmitted directly between nodes, and not via the centralized directory service node. Furthermore each user and computing node is identified by the directory service using a random ID.

Volunteer computing (VC) (Anderson and Fedak 2006) is a decentralized paradigm for offering Cloud-like services over the Internet. It proposes the use of idle or surplus computing capacities of underutilized personal computers connected to the Internet to deploy the infrastructure. The main characteristics of VC are: a) no central authority is responsible for providing the required computational resources; b) heterogeneous and low capacity computer resources spread across the Internet; and c) the computational infrastructure belongs to the user. Regarding reliability and Quality of Service (QoS), these computing systems have to guarantee to the user: a) availability (the user can access data anytime from anywhere); b) freshness (the user gets up-to-date data); and c) immediateness (the user obtains the data in a time

978-1-7281-9499-8/20/$31.00 ©2020 IEEE

that is felt as immediate). The fact that users' resources have an uncertain behavior regarding connection, disconnection, capacity and failure is the main challenge in this field.

This work tackles the problem of developing blog replica decision policies for the directory service of a distributed VC micro-blogging platform. The goal is to maximize the system's reliability, in terms of content availability, while minimizing the total number of blog replicas generated to make an efficient use of the resources. The contributions of our work are: 1) a prototype model of a VC micro-blogging service; 2) an event-based simulation model of the VC micro-blogging platform to model both connection and disconnection of nodes and blogger messaging behavior; 3) a proposed sort and select directory service replication decision mechanism; and 4) description of the insights provided by preliminary numerical results regarding reliable and efficient VC micro-blogging services.

The remainder of this paper is structured as follows. Section 2 reviews related research. Afterwards, Section 3 describes the problem studied and Section 4 proposes a model for an example of DVCMBS. Then, Sections 5 and 6 explain a simulation model and define the decision policies proposed for the directory service, respectively. Section 7 illustrates our approach through preliminary computational experiments. Finally, Section 8 draws a few conclusions.

2 RELATED WORK

This section describes works on the selection of resources in distributed large-scale systems based on heterogeneous and non-dedicated components. For an extended literature review of resource allocation in VC, the reader is referred to Mengistu and Che (2019).

2.1 Resource Allocation in VC Systems

Panadero et al. (2018) proposes the multi criteria biased randomized method, a multi-objective selection method for large-scale systems. The method is assessed in a micro-blogging social network formed by a large number of microservices hosted by volunteer nodes. Panadero et al. (2017) develops a simheuristic (Juan et al. 2015) for the resource allocation problem considering stochastic node quality. The method relies on a VNS metaheuristic to generate allocation mappings that achieve a minimum level of reliability. Similarly, Alsenani et al. (2018) presents SaRa, a system to estimate the reliability of untrusted edge resources in volunteer cloud computing. Selimi et al. (2019) develops BASP, a service placement algorithm to place micro-cloud services in community networks. It relies on k-means for clustering and a light weight bandwidth estimation heuristic. Guler et al. (2015) puts forward various heuristics to distribute jobs, while maximizing throughput, without violating money budget constraints. Ghafarian and Javadi (2015) proposes a method to schedule scientific and data intensive workflows, enhancing the use of VC systems. Their method increases the percentage of workflows that meet the deadline, satisfying the QoS constraints in terms of the deadline, minimum CPU speed, and minimum RAM or hard disk requirements.

2.2 Online Social Networks

Duong-Ba et al. (2014) develops three heuristics, which are based on user communication patterns, to address the client-server assignment problem in online social network applications, taking into account load balance. Zhang and Tang (2014) presents three heuristics to assign clients to servers in continuous distributed interactive applications (DIAs) (Diot and Gautier 1999). They reduce the network latency to maximize the interactivity under consistency requirements. The minimum achievable interaction time for DIAs to preserve consistency is analyzed. Zheng and Tang (2016) puts forward two efficient server placement methods for hosting continuous DIAs, which optimize the interactivity performance and consider the interaction between clients. Nishida and Nguyen (2011) proposes a heuristic based on relaxed convex optimization which returns an approximately optimal client-server assignment for a given communication pattern among the clients. The heuristic may be used in distributed applications such as instant messaging systems.

3 PROBLEM DESCRIPTION

The problem addressed in this work is the development of blog replica decision policies for the directory service of a distributed volunteer computing micro-blogging platform. A distributed volunteer computing micro-blogging service (DVCMBS) provides a platform for a community of users where messages can be transmitted for all other users. There is no central database for recording all previous messages; instead, some users donate a portion of their computer's resources to the blogging service and they are used to create a parallel distributed database. The users of such a service are volunteers, they are under no obligation to leave their computers online to ensure that the full record of previous messages remains available. The challenge is to develop a directory service that undertakes the role of deciding on which memory resources to store replicas of user blogs. The goal is to maximize the availability of all messages over time, with replica efficiency as a secondary goal.

Here we describe a directory service that considers replication decisions whenever a computer arrives online. Upon arrival, the new online computer may be a blogger signing in, in which case replicas of the new arrivals set-of-previous-messages (*blog*) can be stored within the denoted memory space (*repository*) of other online computers. Additionally, the new online computer may donate a repository to the DVCMBS, in which case blog replicas can be stored in it. This work focuses only on replication decision motivated by arrival events, and does not consider replication decisions motivated by computers going offline. This approach allows for a clear analysis of replication policies initiated by arrival events.

Directory service replication decision policies are required to ensure that the volunteer computing micro-blogging service functions reliably. In this work reliability is defined as the average rate at which all messages are available within the repositories of online users.

4 A MODEL FOR A VOLUNTEER COMPUTING MICRO-BLOGGING SERVICE

A DVCMBS is composed of computing nodes. They include a node that hosts the directory service, which is assumed to be the only node that is always available. The remaining nodes are computers whose online presence is uncertain: computers can leave and enter the system freely and as frequently as they choose. A computer can correspond to: 1) a blogger who donates memory (storage not RAM) resources; or 2) a donated resource with no associated blogger. The directory service is in charge of assigning blog replicas to the donated distributed storage capacity. Furthermore, it keeps a record of the blog replicas stored in each repository. For simplicity, we consider the case where the set of potential computing nodes is fixed. Each computing node has a set of parameters governing its behavior: arrival rate (if offline), distribution of online duration, associated blogger's messaging rate, distribution of the sizes of the blogger's messages and amount of memory donated. If a node does not have an associated blogger then the parameters relating to messaging behavior are set to zero. Similarly, if a node does not contribute any memory the donated memory size is set to zero. We assume that both online duration and blogger message size follow log-normal distributions. Nodes are assumed to arrive online according to a Poisson distribution. While these assumptions allow a concise theoretical presentation of our approach, the proposed methods extend to other distributions including empirical ones.

It is assumed that each blogger keeps a separate copy of its own blog. This means that if all replicas of a blog are lost whilst the associated blogger is offline, complete replicas can be generated when the blogger arrives back online. Furthermore, this means that the online availability of a replica of a blog can be treated as a binary issue.

5 SIMULATION

This section describes a simulation model of the VC micro-blogging platform. The model is both time discretized and event based. In each time interval, node arrival and exit events are simulated along with blogger messaging behavior. Figure 1 shows a flowchart of the model.

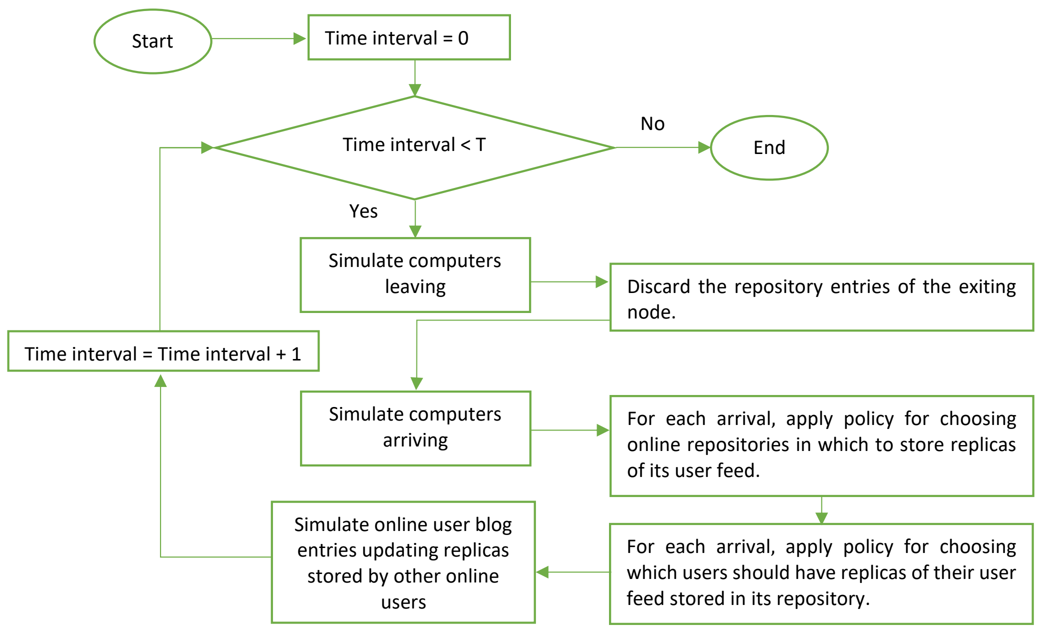

Figure 1: Flowchart of the simulation model for the DVCMBS.

Within each simulated interval of time node exit events are simulated first. This ordering ensures that replicas are not assigned to these nodes in a time interval. The DVCMBS can operate in two ways, periodic checking for exit, arrival and message events, or in a pure event based manner. As the time interval size is decreased the proposed models tends to a pure event based model. An exit event occurs when a randomly sampled online duration has been completed. In this work we assume that the contents of the repositories of exit nodes are discarded. The alternative to this assumption is to try to recycle the contents of arrival nodes, when nodes arrive back online after exiting.

Following the consideration of the exit events, user arrival events are simulated. Two types of arriving node related replication decisions can be considered by the directory service. Firstly, replicas of the blog associated with the arrival node's blogger can be stored in the repositories of other online computers, this step simulates a blogger signing in and being assigned to repositories. Secondly, blog replicas stored within other online repositories can be stored within the repository of the new arrival. In this case the directory service determines which blogs are in greatest need of additional replicas. A maximum limit is placed on the number of replicas assigned to the arrival's repository, this helps to avoided over utilization of donated resources and provides a mechanism for controlling efficiency. In both cases blog replicas can only be stored in repositories with sufficient remaining capacity. Section 6 outlines the criteria used to determine which and how many replicas are generated in both cases. Following this step, the messaging behavior of all online bloggers is simulated. Every time a blogger posts a new message, the directory service updates the replicas of that blogger's blog. If a blog replica becomes too large for its repository it is deleted. The simulation continues until a specified number of time intervals have been completed. The number of repeat simulations used to test candidate directory service policies was set to 300 because a very tight level of convergence of average availability and efficiency was achieved for a wide range of random seeds. The simulation model was implemented as a single threaded java application on a 2.8GHz core i5 desktop.

6 DECISION POLICIES FOR THE DIRECTORY SERVICE

This section describes the directory service decision policy considered, namely a sort and select based policy. Such a policy is applied whenever we need to choose a repository in which to store a given blog replica or to choose a blog to replicate and store within a given repository. Sort and select policies sort the candidate repositories (or blogs) according to a measure of reliability of each replication decision. Replication decisions are then made sequentially in decreasing reliability order until a termination criteria is reached. The sorting criteria determines *which* replicas are made (Subsection 6.2), while the termination criteria determines *how many* (Subsection 6.3) replicas are made in each decision epoch.

6.1 Sorting Policies for Different Replication Decision Epochs

The directory service, considered in this work, can be faced with two different types of blog replication decision epochs, which are listed as follows:

- *Replication decision epoch 1:* A computer arrives online and replicas of the blog associated with a blogger using that computer can be stored within the repositories of other online computers. In this work candidate repositories are sorted in *decreasing order of expected exit time*. That is, we want to maximize the amount of time that replicas are available. This criteria reflects this works focus on ensuring the availability of replicas of blogs whilst the associated bloggers are offline, since the directory service can allocate new replicas once the blogger arrives back online, if there is found to be an insufficient amount.
- *Replication decision epoch 2:* A computer arrives online and donates repository space. The decision in this case is which, if any, additional blog replicas should be stored within the new available repository space. Candidate blogs, for replication and storage on the repository of the new online computer, are sorted in *increasing order of the current number of replicas of the candidate blog.* This criteria provides a mechanism for generating additional replicas of blogs which have few blogs, which may be because repositories hosting replicas of their blogs have gone offline.

6.2 Which Replica?

Having sorted the candidate repositories for storing a given blog replica (or blogs to replicate and store in a given repository), single candidates are selected sequentially until some pre-specified termination criteria are satisfied. In this work, candidates are selected from the tops of the sorted candidate lists.

6.3 How Many Replicas?

Each time additional replicas of a blog are made, the criteria outlined in Subsection 6.1 are used to sort the available host repositories. The number of replicas generated is determined by the target number of replicas per blogger (N). The termination criteria can be expressed in pseudocode as follows.

Algorithm 1 Termination criteria when generating replicas of blog b

```
1: Inputs: C sorted candidate list of available host repositories, R^b current set of available replicas of blog b
2: while (|R^b| < N) ∩ (|C| > 0) do
3:    Select a host repository (c) for an additional replica (r) of a given blog, using a greedy selection policy
4:    R^b ← {R^b ∪ r}
5:    C ← {C \ c}
6: end while
7: Output: R^b
```

In general, the number of replicas generated, before these termination criteria are satisfied, increases with N. The choice of the value for N influences the availability rate and the number of replicas generated by the directory service. It is important to note that whilst generating more replicas increases blog replica availability rates, it also increases the number of replicas that have to be updated each time a blogger posts a message. This point reflects the secondary objective of a directory service replication policy, that of efficient use of resources.

For the case of blog replication decision 2) in Subsection 6.1 the maximum number of blog replicas that can be stored within the repository of an arrival node is an input parameter denoted M, which is designed to prevent the overloading of the computational resources donated by the arrival node. This parameter also has an influence upon the availability rate (reliability) and the number of replicas generated (inefficiency).

7 NUMERICAL RESULTS

This section analyzes the trade-off between reliability and efficiency of DVCMBS directory services operating using a range of replication decision parameters (N,M) as well as for a variety of blogger population sizes ($|B|$) and arrival rates (λ). We firstly describe the details of the test instance and the measures of reliability (average blog replica availability rate) and efficiency (average number of replicas generated per blog).

7.1 Description of Instances

The experiment results presented in this section are based on a test instance defined by the following:

- 100 time intervals, with each time interval representing a duration of one unit of time.
- 50 bloggers/computers.
- Repository sizes of 1 Gigabyte (effectively infinite since we are not considering the memory constrained case).
- Computer/blogger arrival rates uniformly distributed in the range $[0.01, 0.4]$ per time interval (capturing infrequent and frequent arrivals).
- Log-normally distributed computer/blogger online sessions with mean duration uniformly distributed in the range $[1, 10]$ per time interval and standard deviations uniformly distributed in the range $[1, 10]$ (capturing relatively short and long online sessions).
- Bloggers with blog entry rates uniformly distributed in the range $[0.1, 0.9]$ per time interval (capturing infrequent and frequent messaging behavior).
- Log-normally distributed blog entry sizes with mean sizes uniformly distributed in the range $[1, 100]$ bytes and standard deviations uniformly distributed in the range $[1, 20]$ bytes (capturing short and long message behaviors each with low and high message size variance).

The rationale behind these choices of user behavior parameters is to create an instance that is fully heterogeneous with respect to all of the behavioral parameters with a wide range of parameters for each.

7.2 Analysis of the Results

Replica availability and replica efficiency are the main performance measures with which alternative blog replication policies are to be assessed. The overall blog replica availability rate is calculated using the equation $\frac{n^a}{n^r}$, where n^a is the total number of times that replicas of blogs are available when they are required and n^r is the total number of times that replicas of blogs are required. A replica of a blog (b) is required in each time interval (t) for which a blog has a size greater than zero ($m_{bt} > 0$).

As a measure of blog replica inefficiency we calculate the average number of blog replicas generated per blog using the equation $\frac{n^g}{|B|S}$, where n^g is the total number of replicas generated in S repeat simulations and B is the set of bloggers.

$$n^r = \sum_{b \in B} \sum_{t=1}^{T} \begin{cases} 1 & if m_{bt} > 0 \\ 0 & otherwise \end{cases} \quad (1)$$

$$n^a = \sum_{b \in B} \sum_{t=1}^{T} \begin{cases} 1 & if |R_t^b| > 0 \\ 0 & otherwise \end{cases} \quad (2)$$

Our experimental results will be based on the simulation testing of a variety of blog replication policies. We have identified the necessary number of simulation repeats per experiment as at least 100. We have selected 300, since this number leads to a high level of convergence of the key performance measures, starting from different random seeds. It is also worth noting availability rates reach a reasonably steady state within each individual repeat simulation.

Hereafter, we adopt the convention of referring to *average blog replica availability rate* as ***availability*** and *average number of blog replicas generated per blog* as ***inefficiency***, which are the main performance measures of interest.

7.3 Results For Different Replication Policy Parameter Sets

This section presents results for simulation experiments, based on the test instance described in Subsection 7.1, in which the replication policy parameters N and M are varied. In particular, we test all combinations of $N \in \{0,1,2\dots8,9,10\}$ and $M \in \{0,1,2\dots8,9,10\}$. For all combination of N and M, Figure 2 (left) provides blog replica availability rates, whilst Figure 2 (right) provides inefficiency results. The x-axis, in both cases, is the value of M. Each data series represents the results for a different value of N. Figure 2 (left) shows that, in general, increasing both N and M has the effect of increasing availability rate. This is the expected result, since both parameters directly control the number of replicas of blogs that can be generated. Furthermore, it shows that the additional benefit of increasing the value of M diminishes as $N \to M$ and beyond. The explanation for this is that the value of N controls the demand for blog replication, while M controls the number of blog replicas which can be generated if there is additional demand for replicas. Figure 2 (right) confirms that, increasing both N and M increases the number of blog replicas generated and that the impact of increasing M diminishes as $M \to N$.

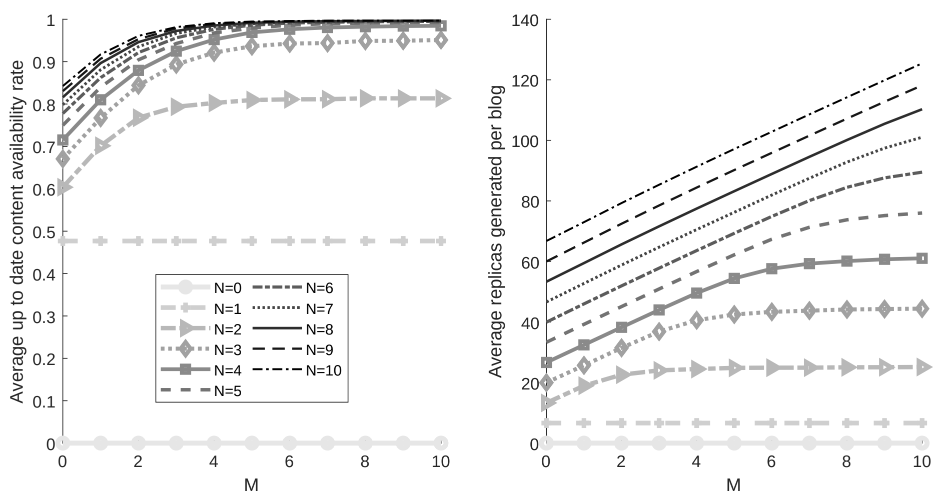

Figure 2: The effect of the replication policy parameters N and M on blog replica availability and the average numbers of blog replicas generated per blog.

Figure 3 provides a different perspective on the results of the same set of experiments. Figure 3 plots, with circular markers, non-dominated pairs of values for N and M, with respect to availability maximization and inefficiency minimization. Labels annotating the (N,M) values for four of the non-dominated replication policy parameters are given. Reading those from left to right, shows that the value of N is the most important parameter to vary for exploring the trade off between availability and inefficiency. Low values for N lead to low inefficiency and low availability, while high values of N lead to high inefficiency and high availability.

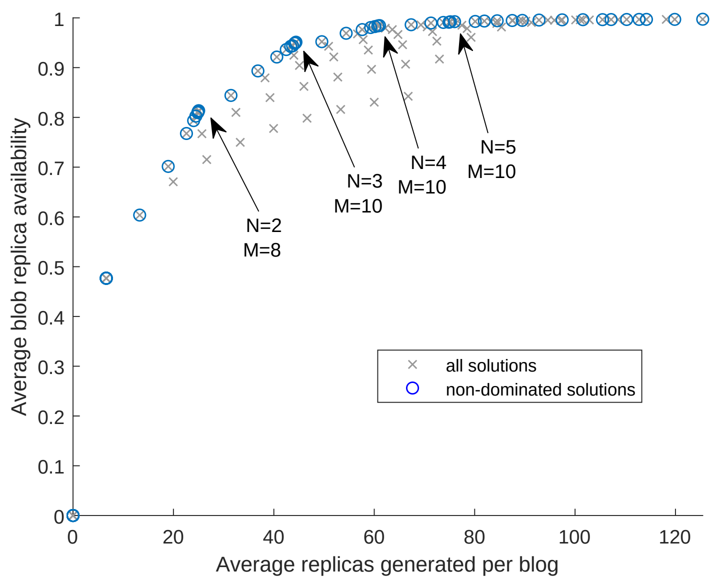

Figure 3: The non-dominated replication policy parameters with respect to maximizing blog replica availability and minimizing the total number of blog replicas generated.

7.4 Results for the Case of $N = 3$ and $M = 5$

In this section we present some additional experiment results, focusing on the specific case of $N = 3$ and $M = 5$, which was one of the non-dominated sets of replication policy parameters, and which attained a balanced trade off between availability and inefficiency. Firstly, Figure 4 provides a time interval trace, from one repeat simulation, of the numbers of: online bloggers; total replicas; blogs with replicas available; and blogs requiring replicas. Figure 4 shows that, in the initial phase of the simulation, the number of online bloggers increases, because initially there are none. At the same time, the number of blogs requiring replicas, the number of blogs with replicas and the total number of replicas all increase. By time interval 20, a reasonably steady state has been attained. A steady state occurs when the number of arriving and exiting nodes reach an equilibrium. Figure 4 shows that the proposed directory service policy is effective. Firstly, we can see that the number of blogs with replicas remains significantly higher than the number of online bloggers, which proves that the policy is ensuring the online availability of blogs, even when the associated bloggers are offline. Secondly, it can be seen that the total number of blogs with replicas remains very close to the number of blogs requiring replicas. Figure 4 also shows that the number of replicas, stored within the online bloggers' repositories, that are required for maintaining a high availability rate, is generally far in excess of the number of blogs which require replicas.

Whilst still considering the case of $N = 3$ and $M = 5$, we now consider the impact of the size of the population of bloggers $|B|$ and the arrival rate of bloggers. In particular, we consider all combinations of blogger populations sizes $|B| \in \{1,2,5,10,20,50,100,200,500,1000\}$ and blogger arrival rates $\lambda \in \{0.01,0.02,0.05,0.1,0.2,0.5,1\}$. The remaining test instance parameters are those defined in Subsection

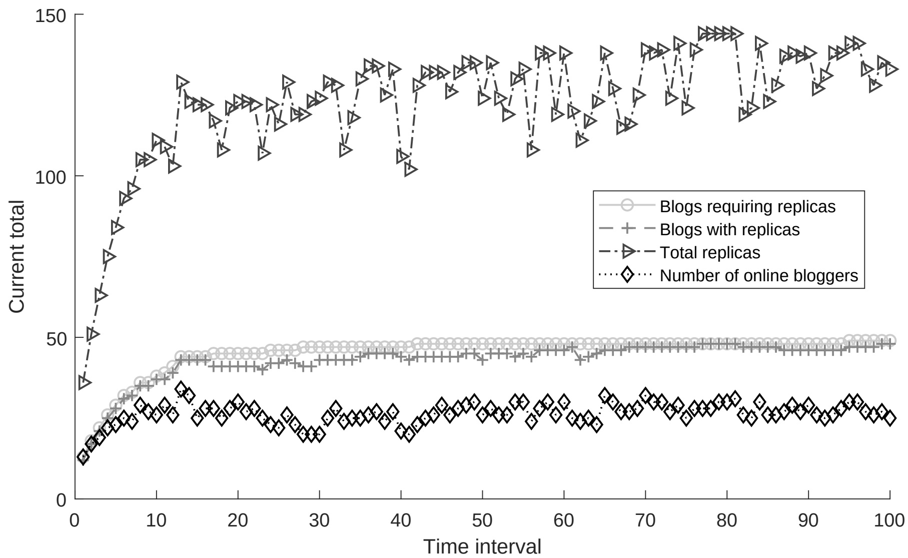

Figure 4: Simulation trajectories of the numbers of blogs requiring replicas, blogs with replicas, total replicas and online users.

7.1. Figure 5 displays availability (left) and inefficiency (right) values for each considered combination of blogger population size ($|B|$) and arrival rate (λ). Blogger population size values are given on the x-axis, while arrival rate values are displayed with a data series for each. Figure 5 (left) shows that, in general, availability rates increase with increasing population size and arrival rate. This makes sense when considering that larger numbers of bloggers and arrival rates both increase the availability repository space for storing blog replicas. Furthermore, high arrival rates increase the overall total amount of time that nodes are online and available for storing blog replicas, as well as decreasing the amount of time that bloggers are offline, which is the time when their blogs are in greatest need of available replicas. Figure 5 (right) displays inefficiency values, and shows that, in general, inefficiency increases with arrival rate and also blogger population size. Figure 5 (left and right) show that availability and efficiency are conflicting objectives. Beyond a population size of 100 ($log_{10}(100) = 2$), inefficiency per blog stabilizes, whilst availability continues to increase. The stabilization of inefficiency can be explained by the fact that small populations will demonstrate a lower observed variance of online duration times, and hence a lower number of nodes with very high online duration times. The use of such nodes, for replica storage, is beneficial for improving availability and reducing the number of occasions on which additional replicas are required, since such nodes remain online for longer. We can conclude that directory service performance, with respect to availability maximization and inefficiency minimization, benefits from high arrival rates and blogger populations sizes.

8 CONCLUSIONS

This work has considered the problem of developing blog replica decision policies for the directory service of a volunteer computing micro-blogging service (DVCMBS) with a focus on maximizing both content availability and efficiency in terms of the number of replicas generated. As a basis for this investigation, a discrete time and event based simulation model of a DVCMBS was proposed and the possible blog replication decisions were enumerated. This work proposed a sort and select approach for determining which repositories are selected to host replicas of given blogs and for selecting which blogs require additional replicas given newly available repository space. Candidate replication decision sorting criteria

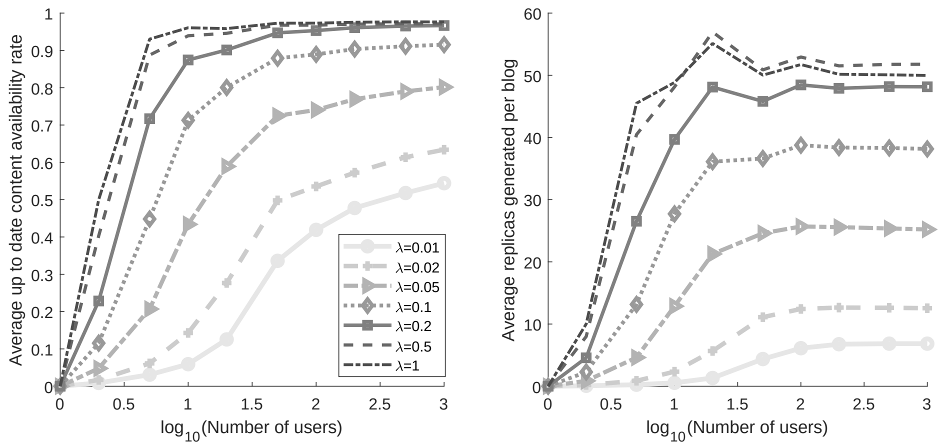

Figure 5: The effect of number of bloggers ($|B|$) and blogger arrival rate (λ) on blog replica availability and the average numbers of blog replicas generated per blog.

were proposed, which were based on expected online duration values and numbers of existing blog replicas. Termination criteria were proposed for determining how many replicas of each blog are necessary. The experiment results revealed that in order to achieve high availability rates large numbers of replicas of blogs were required. In regard to the parameters of the blog replication termination criteria, it was found that setting a minimum target number of replicas per blog was an effective parameter for improving blog replica availability but moderate choices of this parameter were required for achieving a good trade off between availability and efficiency. High blogger population sizes and arrival rates were both found to be beneficial for attaining high availability rates. Evidence was also found for the existence of a critical population size over which efficiency levels stabilize, due to the presence of a larger number of nodes with very high online duration times, which are highly effective storage nodes.

Several lines of future work stem from this work. For instance, the proposed directory service could be extended to consider replication decisions in response to node exit events, the recycling of the contents of repository entries of arriving nodes, and a randomized approach for selecting candidate replication decisions. In addition, it would be interesting to focus on the case where only very few computing nodes donate repository space. Latency and bandwidth can also be considered as selection criteria for replication and storage of blogs, these features introduce an energy efficiency dimension to efficiency when considering the trade off between availability and efficiency.

ACKNOWLEDGMENTS

This work has been partially supported by the Spanish Ministry of Science, Innovation and Universities (PGC2018-097599-B-100).

REFERENCES

Alsenani, Y., G. Crosby, and T. Velasco. 2018. "SaRa: A Stochastic Model to Estimate Reliability of Edge Resources in Volunteer Cloud". In *2018 International Conference on Edge Computing (EDGE)*, 121–124. June 25th-30th, Seattle, USA: Institute of Electrical and Electronics Engineers, Inc.

Anderson, D. P., and G. Fedak. 2006. "The computational and storage potential of volunteer computing". In *Sixth International Symposium on Cluster Computing and the Grid (CCGRID'06)*, Volume 1, 73–80. May 16th-19th, Singapore: Institute of Electrical and Electronics Engineers, Inc.

Darlagiannis, V. 2005. "Hybrid peer-to-peer systems". In *Peer-to-Peer Systems and Applications*, edited by R. Steinmetz and K. Wehrle, 353–366. Berlin Heidelberg: Springer.

Diot, C., and L. Gautier. 1999. "A distributed architecture for multiplayer interactive applications on the internet". *IEEE Network* 13(4):6–15.

Duong-Ba, T., T. Nguyen, B. Bose, and D. A. Tran. 2014. "Distributed client-server assignment for online social network applications". *IEEE Transactions on Emerging Topics in Computing* 2(4):422–435.

Ghafarian, T., and B. Javadi. 2015. "Cloud-aware data intensive workflow scheduling on volunteer computing systems". *Future Generation Computer Systems* 51:87 – 97.

Guler, H., B. B. Cambazoglu, and O. Ozkasap. 2015, Nov. "Task allocation in volunteer computing networks under monetary budget constraints". *Peer-to-Peer Networking and Applications* 8(6):938–951.

Juan, A. A., J. Faulin, S. E. Grasman, M. Rabe, and G. Figueira. 2015. "A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems". *Operations Research Perspectives* 2:62 – 72.

Mengistu, T. M., and D. Che. 2019. "Survey and taxonomy of volunteer computing". *ACM J. Comput. Surv*:1–35.

Nishida, H., and T. Nguyen. 2011. "Optimal client-server assignment for internet distributed systems". In *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN 2011)*, Volume 1, 1–6. July 31$^{\text{th}}$-August 4$^{\text{th}}$, Maui, Hawaii: Institute of Electrical and Electronics Engineers, Inc.

Oukemeni, S., H. Rifà, and J. M. Marquès. 2018. "Privacy in Microblogging Online Social Networks: Issues and Metrics". In *Reunión española sobre criptología y seguridad de la información (RECSI)*, 107–112.

Panadero, J., L. Calvet, J. M. Marquès, and A. A. Juan. 2017. "A simheuristic approach for resource allocation in volunteer computing". In *Proceedings of the 2017 Winter Simulation Conference.*, edited by W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, 1479–1490. Las Vegas, NV, USA: Institute of Electrical and Electronics Engineers, Inc.

Panadero, J., J. de Armas, X. Serra, and J. M. Marquès. 2018. "Multi criteria biased randomized method for resource allocation in distributed systems: Application in a volunteer computing system". *Future Generation Computer Systems* 82:29 – 40.

Selimi, M., L. Cerdà-Alabern, F. Freitag, L. Veiga, A. Sathiaseelan, and J. Crowcroft. 2019, Mar. "A Lightweight Service Placement Approach for Community Network Micro-Clouds". *Journal of Grid Computing* 17(1):169–189.

Seong, S.-W., J. Seo, M. Nasielski, D. Sengupta, S. Hangal, S. K. Teh, R. Chu, B. Dodson, and M. S. Lam. 2010. "PrPl: a decentralized social networking infrastructure". In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, edited by R. Han and L. Erranli, 8. New York: Association for Computing Machinery.

Statista 2019a. Number of monthly active Twitter users worldwide from 1st quarter 2010 to 1st quarter 2019. https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/, accessed 1st June.

Statista 2019b. Cumulative total of Tumblr blogs from May 2011 to April 2019. https://www.statista.com/statistics/256235/total-cumulative-number-of-tumblr-blogs/, accessed 1st June.

Zhang, L., and X. Tang. 2014. "The client assignment problem for continuous distributed interactive applications: Analysis, algorithms, and evaluation". *IEEE Transactions on Parallel and Distributed Systems* 25(3):785–795.

Zheng, H., and X. Tang. 2016. "The server provisioning problem for continuous distributed interactive applications". *IEEE Transactions on Parallel and Distributed Systems* 27(1):271–285.

AUTHOR BIOGRAPHIES

CHRISTOPHER BAYLISS is a post-doctoral researcher in the ICSO group at the IN3 – Universitat Oberta de Catalunya. His main research interests include metaheuristics, simulation optimization, revenue management, packing problems, airline scheduling, and logistics optimization. His email address is cbayliss@uoc.edu.

JAVIER PANADERO is an Assistant Professor of Simulation and High Performance Computing in the Computer Science Dept. at the Universitat Oberta de Catalunya. He is also a Lecturer at the Euncet Business School. He holds a Ph.D. and a M.S. in Computer Science. His major research areas are: high performance computing and simheuristics. He has co-authored more than 40 scientific articles. His website address is http://www.javierpanadero.com and his email address is jpanaderom@uoc.edu.

LAURA CALVET is an Assistant Professor of Statistics in the Computer Science Dept. at the Universitat Oberta de Catalunya and a Lecturer in Economics at the Universitat Internacional de València. Her main lines of research are: i) design of optimization algorithms relying on metaheuristics, machine learning and/or simulation applied to sustainable logistics & computing; ii) applied statistics & economics. Her email address is lcalvetl@uoc.edu.

JOAN MANUEL MARQUÈS is an Associate Professor at Computer Science, Multimedia & Telecommunication Studies at Universitat Oberta de Catalunya since 1997. He graduated as a Computer Science Engineer from the Facultat d'Informàtica de Barcelona (UPC) and a Ph.D. in Computer Science from UPC. His research interests include the design of scalable and cooperative Internet services and applications. He is member of the Association for Computing Machinery.