

Processing Collections of Geo-Referenced Images for Natural Disasters

Procesamiento de Colecciones de Imágenes Georreferenciadas para Desastres Naturales

Fernando Loor¹, Veronica Gil-Costa¹, and Mauricio Marin²

¹CONICET, Universidad Nacional de San Luis, Argentina.
{floor,gvcosta}@unsl.edu.ar

²Universidad de Santiago de Chile, Chile.
mauricio.marin@usach.cl

Abstract

After disaster strikes, emergency response teams need to work fast. In this context, crowdsourcing has emerged as a powerful mechanism where volunteers can help to process different tasks such as processing complex images using labeling and classification techniques. In this work we propose to address the problem of how to efficiently process large volumes of georeferenced images using crowdsourcing in the context of high risk such as natural disasters. Research on citizen science and crowdsourcing indicates that volunteers should be able to contribute in a useful way with a limited time to a project, supported by the results of usability studies. We present the design of a platform for real-time processing of georeferenced images. In particular, we focus on the interaction between the crowdsourcing server and the volunteers connected to a P2P network.

Keywords: Geo-referenced images, support platforms for natural disaster, P2P network.

Resumen

En situaciones de desastres naturales, los equipos de respuesta a emergencia deben actuar con rapidez. En este contexto, el *crowdsourcing* emerge como un mecanismo poderoso donde voluntarios pueden ayudar a realizar diferentes tareas como procesar imágenes complejas usando técnicas de clasificación y etiquetado. En este trabajo proponemos abordar el problema de cómo procesar eficientemente grandes volúmenes de imágenes georreferenciadas usando *crowdsourcing* en un contexto de alto riesgo como desastres naturales. La investigación en ciencia ciudadana y *crowdsourcing* indica que los voluntarios deberían ser capaces de contribuir de manera eficiente en un tiempo limitado a un proyecto, con soporte de los resultados de estudio de usabilidad. Presentamos el diseño de una plataforma en tiempo real para el procesamiento de imágenes georreferenciadas. En particular, nos enfocamos en la interacción entre el

servidor de *crowdsourcing* y los voluntarios conectados a una red P2P.

Palabras claves: Imágenes georreferenciadas, plataformas de soporte para desastres naturales, redes P2P

1 Introduction

During the last years, Big Data has become a strong focus of global interest, attracting more and more attention from academia, industry, government and other organizations. The growing flow of data, which comes from different types of sensors, messaging systems and social networks, in addition to more traditional measurement and observation systems, has already invaded many aspects of our daily existence. Large data, including large geo-referenced data, have great potential to benefit many social applications such as climate change, health, surveillance, disaster response, critical infrastructure monitoring, transport, etc. Geo-referenced data describe elements in relation to the geographical space — location, often with location coordinates in a spatial reference system. The term commonly used for systems that use this type of data is known as GIS — Geographic Information System. The geo-tagged contents of the web, voluntary geographic information (VGI), satellite navigation, etc. are traditionally collected through sensors [1]. The authors in [2] claim that geo-referenced data from social networks is another form of VGI data.

The information that is posted on social networks can be very useful in case of a natural disaster. As direct witnesses of the situation, people share photos, messages and videos about events that get their attention. In an emergency operations center, these

Citation: F. Loor, V. Gil-Costa and M. Marin. *Processing Collections of Geo-Referenced Images for Natural Disasters*. Journal of Computer Science & Technology, vol. 18, no. 3, pp. 193-202, 2018.

DOI: 10.24215/16666038.18.e22

Received: June 11, 2018 **Accepted:** October 31, 2018.

Copyright: This article is distributed under the terms of the Creative Commons License CC-BY-NC.

data can be collected and integrated into the management process to improve the general understanding of the situation or rescue actions. This type of volunteer participation is known as crowdsourcing. Digital volunteers have helped to collect pertinent information much faster than officials or people in charge of coordination activities in natural disasters could do alone, with huge potential impacts on the responsibilities of officials in the management of the information. Emergency search and rescue teams already use pre-event remote sensing data when planning operations [3].

In general, the increasing volume and variable format of the large georeferenced data collected pose additional challenges in storage, management, processing, analysis, visualization and verification of data quality. The authors in [4] state that the size, variety and update rate of data sets exceed the capacity of spatial computing technologies and spatial databases commonly used to learn, manage and process data with a reasonable effort.

This paper describes the design of a distributed platform that combines algorithms, processing techniques and associated software tools to efficiently label large volumes of georeferenced images from different sources (satellite images, captured by unmanned aerial vehicles, originated in social networks), to optimize the work of digital volunteers in situations of natural disasters in order to build status maps in real time. In particular, we focus on the interaction between the crowdsourcing server receiving new images to be analyzed and the users inter-connected in a P2P network. The crowdsourcing server sends tasks to the peer-volunteers which tags the images or just vote for an option from a given option list. The crowdsourcing server gathers all the votes from the peers and evaluate their contributions. If there is an agreement in the voting (e.g. most of the volunteers vote for the same option), the task is finished, and the results are saved in a database for statistic purposes. Otherwise, the server selects volunteers with a high reputation and sends the task again.

The remaining of this paper is organized as follows. Section 2 presents previous works. In Section 3 we present the design of our proposed platform and the simulation model for the interaction between the crowdsourcing server and the volunteers. Section 4 presents experimental results and Section 5 concludes.

2 Previous Works

New forms of georeferenced data collection have emerged that have given rise to completely new data sources and data types of a geographical nature. The data acquired by the public -VGI-, and the data from the geo-sensor networks have led to a greater availability of spatial information. Whereas until recently, the authoritarian data sets dominated the topographic

domain, these new types of data expand and enrich geographic data in terms of thematic variation and the fact that they are more user-centered. The latter is especially true for VGI compiled by social media [5].

Some authors claim that "80% of the data is geographical in nature" [6]. Much of the data in the world can be georeferenced, which indicates the importance of georeferenced large data management. The georeferenced data describe objects and things in relation to the geographical space —location— often with coordinates of location in a spatial reference system. The term commonly used for systems that use this type of data is known as GIS —Geographic Information System.

The work in [7] presents a study of how to find out the current locations of users by tracking their mobile devices, such as smartphones. That study mentions services of geo-social networks such as Foursquare used to locate friends and to find nearby shops and restaurants, where many users register in several places and reveal their current location. In other words, it proposes to solve queries using information from different individuals through its mobile devices.

The work presented in [8], describes some examples of how the Department of Defense of the United States uses crowdsourcing to give answers to problems of natural disasters. The authors conclude that there is a great benefit in taking advantage of the power of the crowd, "a process that will continue to mature, evolve and define the way we help others today, tomorrow and in the future."

Barrington et al. [9] presents a review of the state of the art on the use of crowdsourcing and analysis of images, particularly high resolution aerial. This work describes the experiences obtained in the cases of the earthquake in Haiti and in 2008 in China. Lee and Kang [10] describe the impact of georeferenced data in different applications such as marketing and propose a three layers platform to index and analyze images. But this proposal does not consider the collaboration of volunteers.

Ofli et al. [11] propose a hybrid scheme based on automatic techniques and crowdsourcing for aerial image processing. In this case, manual annotations are used to train a supervised learning system. However, this work does not describe the platform used and does not consider the interaction with information collected by people who are in the place of the event.

There are some platforms such as Tomnod (<http://www.tomnod.com>), GeoTag-X [12], and some research works that address the problem of using volunteers for the processing of georeferenced images [9][13][14][15][16]. In particular, Tomnod of the company DigitalGlobe is using Artificial Intelligence (AI) driven by crowdsourcing to automatically identify the characteristics of interest in satellite and aerial images. Tomnod runs crowdsourcing

campaigns, where volunteers support data mapping by validating the results of an image mining algorithm. GeoTag-X is a research project aimed at researching and evaluating collaborative on-line environments and software tools for creative learning.

3 Proposed Methodology

In this section, we present our platform design for processing large number of georeferenced images. Our design focused on real-time applications devised for natural disasters which require a low latency and short response time.

Fig. 1 shows the general components of our proposed platform and how they communicate to each other. Upon a natural disaster, images are captured through different devices such as a drone and smartphones (Image acquisition). Then, these images are sent to the Object detection and scene labeling component—the black box at the top of the figure. For each incoming image we create a task which will be sent to the crowdsourcing server or to the Machine Learning server. The task contains metadata associated with the image like the GPS coordinates, the image identifier, and a list of options to vote or to label the image. If the machine learning server is not trained for the region where the images come from, the task is sent to the crowdsourcing server. This server will request to different volunteers to help to tag the images. It allows different users to classify images manually, in order to generate new classifiers in real time. Classifiers can detect emerging needs at the time of a disaster.

The results obtained from the crowdsourcing server are used to train the machine learning server. Once this last server is trained, it can start receiving incoming tasks. In this way, we can reduce the amount of work sent to volunteers.

The results achieved by the object detection and scene labeling component are sent to the high-resolution registration component. This last, establishes a correspondence between the image been analyzed and the geographical coordinates to create a situation map. The information inside the situation map is stored in a NoSQL database like MongoDB and can be later use for emergency management and help for victims.

At the bottom of the figure, we show the components deployed in our proposed platform for text messages crawled from Tweeter or others social networks. Relevant data is extracted from these texts The Geo-Tagging Platform uses the Gazetteer which is a geographic dictionary to parse the tweets to identify which coordinates they belong to.

3.1 Modelling the crowdsourcing server

Fig.2 shows the general scheme of our model for the crowdsourcing server and the P2P network. Our

model considers a group of users who execute the tasks delivered by the crowdsourcing server. Each task has information related to the incoming image like the time to live (TTL) of the task, the priority, the image identifier, and the list of options. If the priority feature is on, the priority task is high and should be processed immediately.

Peers build an overlay network managed by the Internet service providers (ISP). In particular, our model follows a P2P Distributed Hash Table —DHT [17]. Internet service providers (ISPs) are responsible for delivering Internet access to clients from a given geographic area. To communicate with other ISPs, it is necessary to access the Internet backbone. The backbone is a shared network which enables communication among ISPs of the world. To make use of the network, ISPs must respect a Service Level Agreement (SLA) contract in which they commit to regulate their traffic to not compromise another ISPs communication.

Each peer holds two task queues. The first queue stores tasks with the priority feature off. In this case, the priority is given by the TTL. The second task queue keeps tasks with the priority featured on. These tasks should be quickly process in order of arrival. In general, the sequence of steps executed by our model is as follows:

1. The crowdsourcing server receives incoming tasks from the platform.
2. A peer becomes visible to the server indicating that he wants to be a volunteer.
3. The server sends a list of tasks to the peer.
4. The peer agrees to start processing the tasks and request the first image.
5. The server sends the first image to the peer and set a time-to-live (TTL).
6. The peer executes the task and sends the result to the server.
7. If a total of H answers were received for a task, the server evaluates if there is an agreement in the voting.
 - (a) If so, the task is finished, and the data is sent to a data base server like MongoDB or Casandra.
 - (b) Otherwise, the server selects new peers with high reputation to send the task again.
8. If the peer wants to continue collaborating, return to step 4 for the next images.

To reduce the communication between the crowdsourcing server and the P2P network, each peer has an LRU cache memory with images received from the server. Thus, the next time a peer requests an image

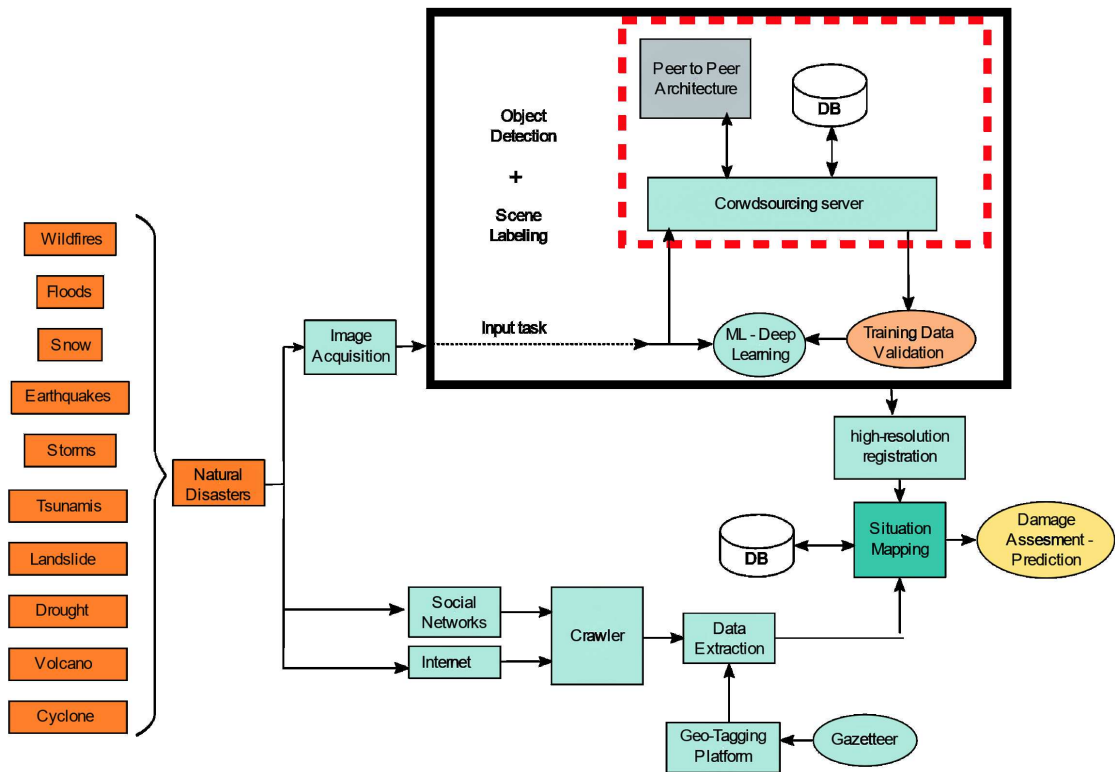


Figure 1: Proposed distributed platform

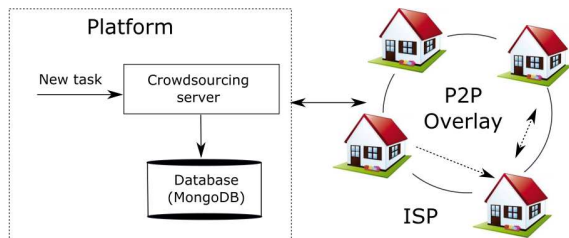


Figure 2: General scheme of our model for Crowdsourcing and P2P network

form the same task list, the peer will search for that image inside the P2P network using the DHT before sending a request to the server.

More precisely, as the whole space of images is divided among the peers, each peer is responsible for a particular set of images. As shown in Fig. 3, each peer is responsible for a different set of images (square, circle and the three types of stars). In this example, the peer 1 request a square image of color blue. Then it sends a message to peer 3 which is selected by the DHT. Due to peer 3 does not have the requested image, it sends the message to the next hop in the P2P network. In this case, the message reaches to peer 4, which is the peer responsible for this kind of images. As peer 4 does not have the blue square image,

it sends a request to the crowdsourcing server. The server sends the requested image to the peer 4, which is the responsible. Peer 4 insert this new image into its LRU cache memory. Finally, peer 4 sends the image to the requesting peer 1.

If the TTL of a given task expires, the server sends the results achieved at the moment to the database server. Additionally, some volunteers can offer to continue helping with the voting/tagging tasks. In this case, the value of H –number of expected answers per task- can be increased. This helps to improve the quality of the results and get a more adequate consensus of the tasks of voting/tagging.

The server applies two policies for the TTL of each task. (1) If the TTL of a given task expires and there are at least H results received from the peers, the server ranks the results. (2) Otherwise, if the TTL expires and there are less than H results, the server extends the TTL and sends additional request with high priority (the priority featured turned-on) to frequently active volunteers.

4 Experimental results

In this work, we evaluate the performance of the crowdsourcing server and its interaction with the network of volunteer forming a P2P network. We have built a simulator that implements a transport layer, a

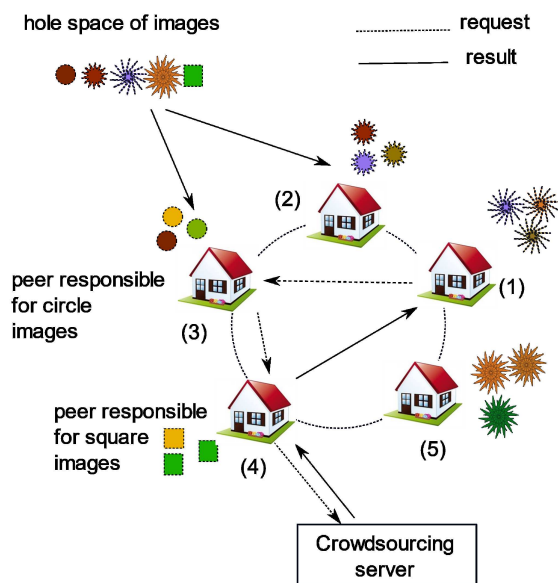


Figure 3: Image space division among peers in the P2P network

P2P overlay and our caching proposal. Pastry [17] has been used as the overlay network in our experimentation. We have also simulated the Web crowdsourcing server, a generator which creates the incoming tasks and a database server which receives data from the crowdsourcing server. The simulation is divided in simulation time windows of 100 units of time.

4.1 Simulation approach

The simulation model of this work uses a processes and resources approach. Processes represent the crowdsourcing server, and the peers in charge of transaction processing. Resources are artifacts such as the data of the incoming messages, global variables like the input queue of each process and also the CPU and the communication network. The simulation program is implemented using LibCppSim [18], where each process is implemented by a co-routine that can be blocked and unblocked at will during simulation.

The operations `hold()`, `passivate()` and `activate()` are used for this purpose. Thus, a coroutine C_i can be paused for a given amount of time Δt -which represents the duration a task. Once the simulation time Δt has expired, the coroutine C_i activates itself if a `hold()` operation was previously executed. Otherwise, the coroutine C_i is activated by another coroutine C_j using the `activate()` operation. This last case allows to represent the interaction among the different components of the simulated platform.

The simulated architecture assumes a classical DHT overlay composed by N physical nodes (or peers) and K object-keys (task with images ids)

mapped onto a ring. Objects (images) stored in a DHT such as Pastry [17], have a responsible peer in the network that is the peer with the closest ID to the key of the object. Thus, any given peer is responsible for a fraction of these images and (on request) it must contact the crowdsourcing server to get them.

4.2 Experiment Settings

First, we evaluate the communication cost between the crowdsourcing server and the peers. We set the parameters of our simulator as shown in Table 1.

Table 1: Parameters setting for the simulation.

Parameter	Value
Network size (num. peers)	500, 1000, 1500
Cache size in each peer	50, 100, 150
Total answers expected by the crowdsourcing server (H)	10, 15, 20, 25
Arrival rate (NR)	500, 1000, 2000

4.3 Gini Coefficient

To measure load balance among peers we use a metric based on Lorenz curves called the Gini coefficient, which is a metric commonly used on other fields like economics and ecology.

If all peers have the same load, the Lorenz curve is a straight diagonal line, called the line of equality or the perfect load balancing. If there is any imbalance, then the Lorenz curve falls below the line of uniformity. The total amount of load imbalance can be summarized by the Gini coefficient G , which is defined as the relative mean difference, i.e. the mean of the difference between every possible pair of peers, divided by their mean load. G values range from 1 to 0. The value 0 is achieved when all peers have the same load. The value 1 is achieved when one peer receives the whole system load while the remaining peers receive none. Therefore, when G approaches 0, global load imbalance is small, and when G approaches 1 the imbalance is large.

In Fig. 4, the y-axis shows the Gini coefficient for different configurations. The x-axis shows the values of peers available to collaborate with the crowdsourcing server. The experiment is performed with different H values. That is, the number of peers receiving a copy of the task to be solved. In Fig. 4 we vary the arrival rate (NR) from 500 to 2000 requests per seconds and the cache size from 5 to 150.

In most of the cases, the Gini coefficient is lower than 50%, meaning that the workload tends to be balance among the peers. Only an H value of 10 shows imbalance for numerous peers. Also, as we increase the arrival rate among the peers, the Gini coefficient

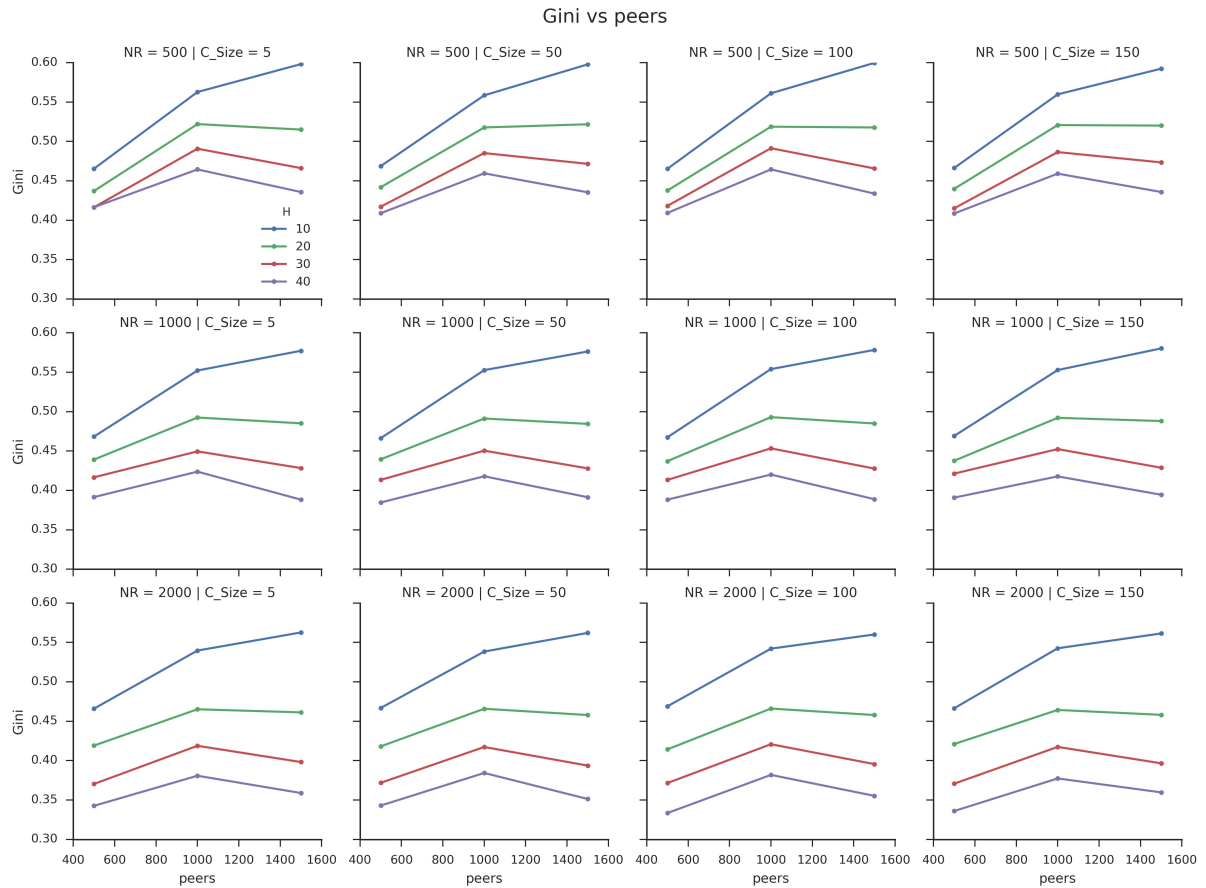


Figure 4: Gini coefficient for different cache size, number of peers, arrival rate and different values of H .

tends to be lower. That is, peers tends to be saturated with almost the same amount of workload.

On the other hand, the arrival rate (NR) and the cache size have very low impact on the Gini coefficient reported by the peers. However, if we increase the number of peers and keep H low, the workload tends to be less balanced. Mainly because, the tasks are not sufficiently well distributed among the peers tending to produce different queue size inside each peer.

4.4 Number of Hops

Fig. 5 shows the average number of hops that a message has to go through, inside the P2P network, before reaching the requested image. The x-axis shows the number of peers selected by the crowdsourcing server to send the tasks. We show the results obtained for a cache size ranging from 5 to 150 and an arrival rate of 500, 1000 and 2000 tasks per second. Again, we perform this experiment with different values of the H parameter.

As expected, the number of hops tends to increase with a larger number of peers in the network. On the other hand, the H value impacts on the number of hops. A larger H value tends to reduce the num-

ber of hops required to find the image inside the P2P network. That's because, as more peers are involved in the process of solving a task (e.g. voting or tagging), the images for those tasks are going to be used by more peers, and therefore a new peer requesting an image would probably find that image in a nearby peer.

4.5 Latency

In Fig. 6, we show the average latency reported by sending messages inside the P2P network. We show results for different arrival rates (NR) and cache sizes of 5, 50, 100 and 150. The x-axis shows different values of H . This experiment is performed with a P2P community formed with 500, 1000 and 1500 peers.

A larger arrival rate tends to increase the communication latency inside the P2P network, due to more messages are present in the network at the same time and those messages compete for the network resource. Notice that the cache size does not drastically impact on the communication latency among the peers.

As we increase the arrival rate, small P2P networks with 500 peers tends to report a larger communication latency, because the same small group of peers has to request the same tasks and images to process (figures

at right). These requests are messages traversing the P2P network.

5 Conclusions

In this work, we presented a new platform to support decision-making in cases of natural disasters, through the processing of information such as georeferenced images in real time. We presented the components of our proposal and how they communicate to each other.

We also modeled the interaction between the crowdsourcing server and the volunteers forming a P2P network. We simulated this model to analyze possible bottlenecks and the benefits of using cache memories in the peers to avoid accessing the crowdsourcing server for each new requested task.

Furthermore, we evaluated different metrics to analyze the effect of communication and the number of peers volunteers on the performance of the platform. Results show that the number of selected peers as volunteers affects the latency of the communication in the P2P network as the number of peers increases.

As future work we plan to evaluate the communication among the P2P network and the crowdsourcing server. Also, we plan to evaluate different approaches to select the experts peers which are going to re-evaluate a task when there is a tie in the voting operation executed by the crowdsourcing server.

Acknowledgements

This research was partially funded by PICT 2014 N 2014-01146 and PROICO-UNSL.

Competing interests

The authors have declared that no competing interests exist.

References

- [1] M. R. Evans, D. Oliver, X. Zhou, and S. Shekhar, "Spatial big data : Case studies on volume , velocity , and variety," in *MobiDE '12 Proceedings of the Eleventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, 2012.
- [2] S. Newsam, "Crowdsourcing what is where: Community-contributed photos as volunteered geographic information," *IEEE MultiMedia*, vol. 17, pp. 36–45, Oct 2010.
- [3] S. Thorvaldsdottir, E. Birgisson, and R. Sigbjornsson, "Interactive on-site and remote damage assessment for urban search and rescue," *Earthquake Spectra*, vol. 27, no. S1, pp. S239–S250, 2011.
- [4] S. Shekhar, V. Gunturi, M. R. Evans, and K. Yang, "Spatial big-data challenges intersecting mobility and cloud computing," in *Proceedings of the Eleventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, MobiDE '12, (New York, NY, USA), pp. 1–6, ACM, 2012.
- [5] M. Sester, J. Jokar Arsanjani, R. Klammer, D. Burghardt, and J.-H. Haurert, *Integrating and Generalising Volunteered Geographic Information*, pp. 119–155. Cham: Springer International Publishing, 2014.
- [6] "Where is the phrase "80% of data is geographic" from?." Available at: <http://www.gislounge.com/80-percent-data-is-geographic/>. Accessed on 2018-11-10.
- [7] M. Choy, J.-G. Lee, G. Gweon, and D. Kim, "Glaucus: Exploiting the wisdom of crowds for location-based queries in mobile environments," in *Proceedings of the 8th International Conference on Weblogs and Social Media, ICWSM 2014*, pp. 61–70, 2014.
- [8] D. Becker and S. Bendett, "Crowdsourcing solutions for disaster response: Examples and lessons for the us government," *Procedia Engineering*, vol. 107, pp. 27 – 33, 2015. Humanitarian Technology: Science, Systems and Global Impact 2015, HumTech2015.
- [9] L. Barrington, S. Ghosh, M. Greene, S. Har-Noy, J. Berger, S. Gill, A. Lin, and C. Huyck, "Crowdsourcing earthquake damage assessment using remote sensing imagery," *Annals of Geophysics*, vol. 54, no. 6, 2012.
- [10] J.-G. Lee and M. Kang, "Geospatial big data: Challenges and opportunities," *Big Data Research*, vol. 2, no. 2, pp. 74 – 81, 2015. Visions on Big Data.
- [11] F. Ofli, P. Meier, M. Imran, C. Castillo, D. Tuia, N. Rey, J. Briant, P. Millet, F. Reinhard, M. Parkan, and S. Joost, "Combining human computing and machine learning to make sense of big (aerial) data for disaster response," *Big Data*, vol. 4, no. 1, pp. 47–59, 2016. PMID: 27441584.
- [12] C. Smith, *A Case Study of Crowdsourcing Imagery Coding in Natural Disasters*, pp. 217–230. Cham: Springer International Publishing, 2017.
- [13] N. Witjes, P. Olbrich, and I. Rebasso, *Big Data from Outer Space: Opportunities and Challenges for Crisis Response*, pp. 215–225. Vienna: Springer Vienna, 2017.

- [14] C. Turk, “Cartographica incognita: ‘dijital jedis’, satellite salvation and the mysteries of the ‘missing maps’,” *The Cartographic Journal*, vol. 54, no. 1, pp. 14–23, 2017.
- [15] T. Onorati and P. Díaz, “Giving meaning to tweets in emergency situations: a semantic approach for filtering and visualizing social data,” in *SpringerPlus*, 2016.
- [16] P. Díaz, J. M. Carroll, and I. Aedo, “Coproduction as an approach to technology-mediated citizen participation in emergency management,” *Future Internet*, vol. 8, no. 3, 2016.
- [17] A. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems,” in *Middleware 2001* (R. Guerraoui, ed.), (Berlin, Heidelberg), pp. 329–350, Springer Berlin Heidelberg, 2001.
- [18] M. Marzolla, “Libc++sim: a simula-like, portable process-oriented simulation library in c++,” *ESM*, vol. -, no. 1, pp. 222—227, 2004.

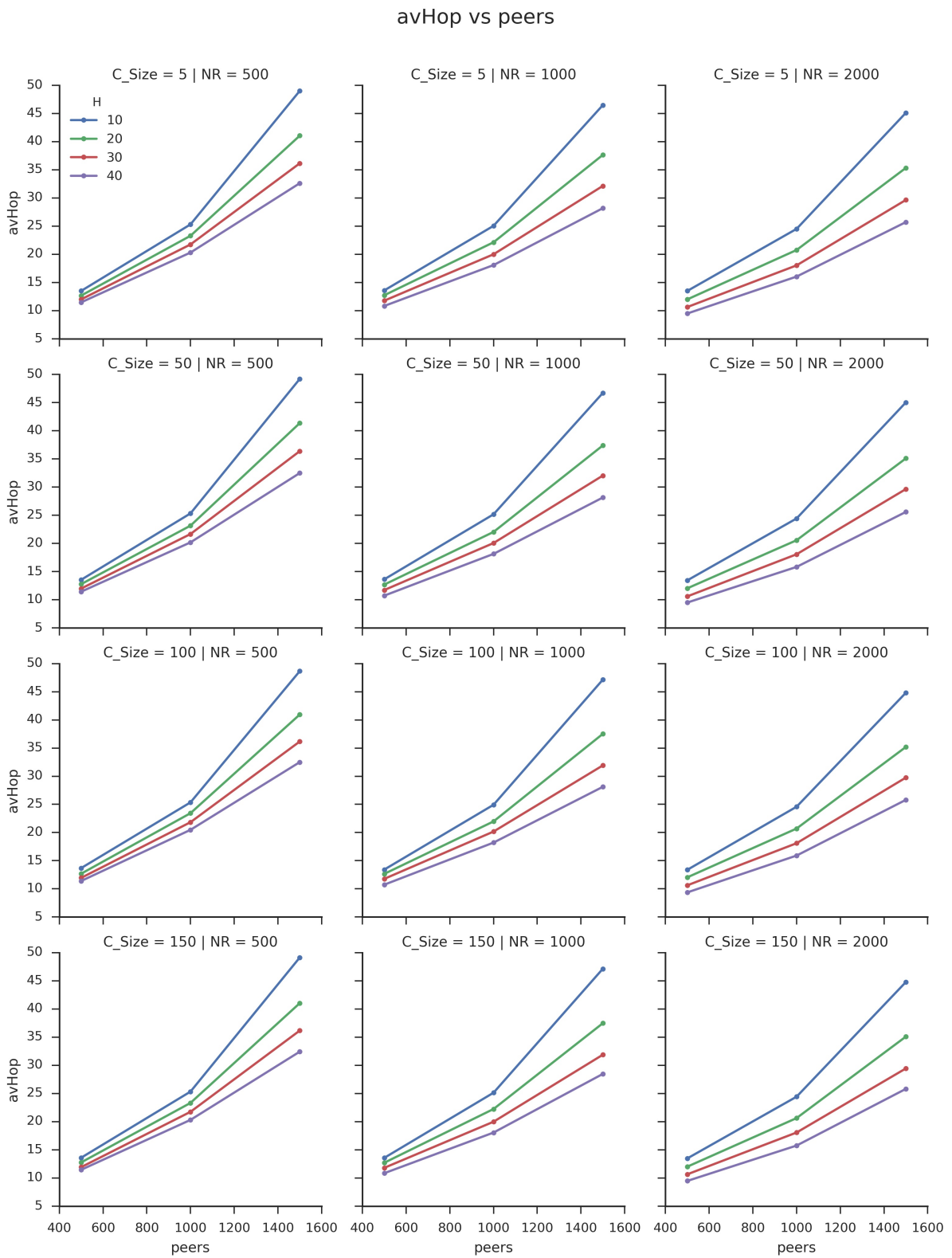


Figure 5: Average number of hops achieved by different parameters of our simulated platform.

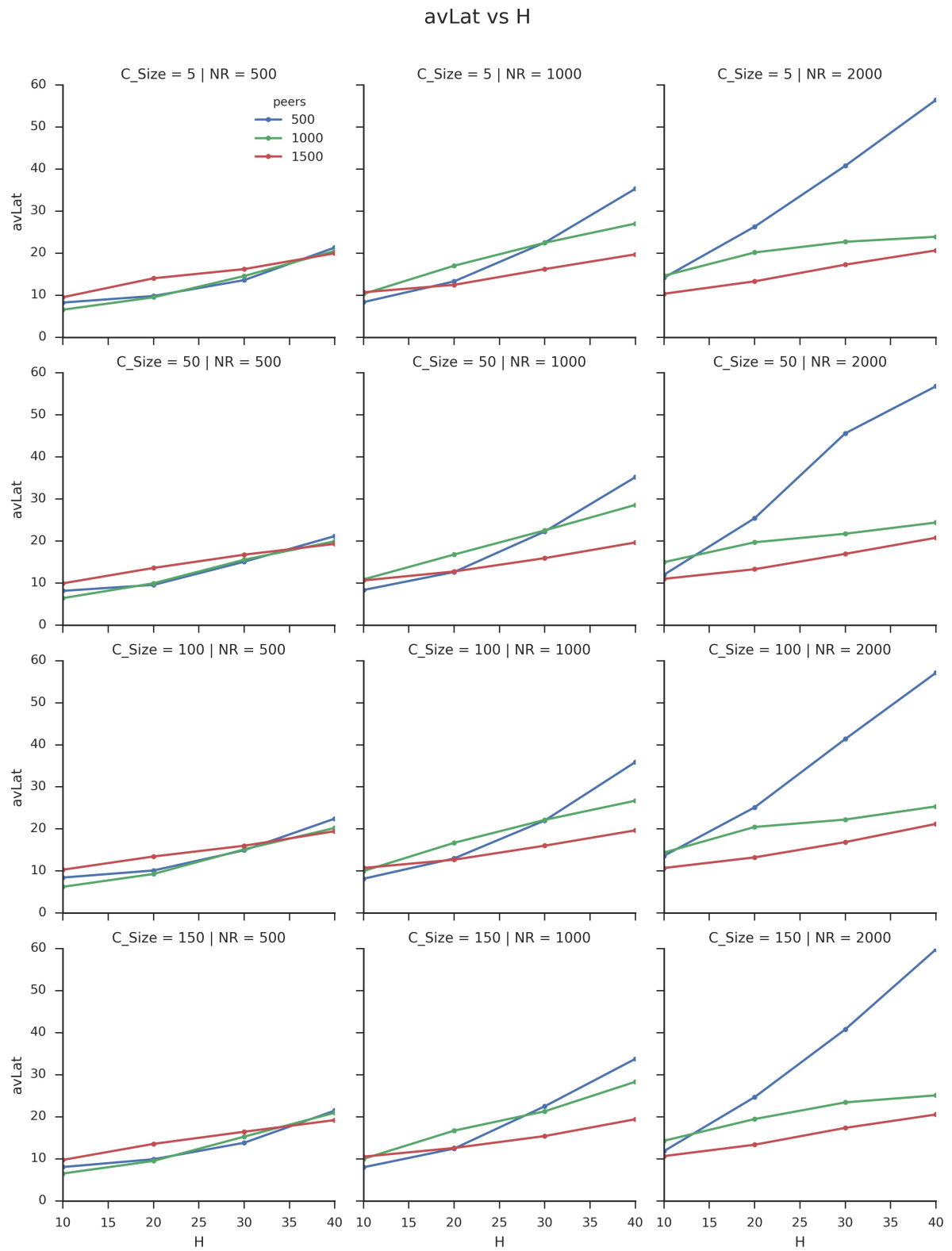


Figure 6: Latency reported by the P2P network for different arrival rates