

Bandwidth Modeling in Large Distributed Systems for Big Data Applications

Bahman Javadi

School of Computing, Engineering and Mathematics
University of Western Sydney, Australia
Email: b.javadi@uws.edu.au

Boyu Zhang, Michela Taufer

Dept. of Computer and Information Sciences
University of Delaware, USA
Email: {bzhang, taufer}@udel.edu

Abstract—The emergence of Big Data applications provides new challenges in data management such as processing and movement of masses of data. Volunteer computing has proven itself as a distributed paradigm that can fully support Big Data generation. This paradigm uses a large number of heterogeneous and unreliable Internet-connected hosts to provide Peta-scale computing power for scientific projects. With the increase in data size and number of devices that can potentially join a volunteer computing project, the host bandwidth can become a main hindrance to the analysis of the data generated by these projects, especially if the analysis is a concurrent approach based on either in-situ or in-transit processing. In this paper, we propose a bandwidth model for volunteer computing projects based on the real trace data taken from the Docking@home project with more than 280,000 hosts over a 5-year period. We validate the proposed statistical model using model-based and simulation-based techniques. Our modeling provides us with valuable insights on the concurrent integration of data generation with in-situ and in-transit analysis in the volunteer computing paradigm.

Keywords—Volunteer Computing, Big Data, Internet Bandwidth, Statistical Modeling.

I. INTRODUCTION

Volunteer computing (VC) is a large-scale distributed paradigm that harnesses the computing power and storage capacity of thousands or millions of hosts owned by the public for scientific applications. Volunteer computing has proven itself as a paradigm that can fully support Big Data generation [7], [13]. For example, to date, the Docking@Home project has generated 1.8 TBytes of data.

While important aspects of this paradigm, such as computing power and storage capability, have been critically analyzed in the past [2], [9], other aspects, such as the host bandwidth has not been addressed yet. The modeling and analysis of the bandwidth is becoming particularly relevant for volunteer computing projects as several middleware packages for in-situ and in-transit analysis are emerging [4], [8], and the number of computing devices that can potentially join the projects is increasing. Ideally, as the volunteers' hosts generate larger and larger amounts of scientific data, these middleware packages can concurrently process the data and extract their scientific meaning either locally by using a reduced part of the volunteered resources on the hosts (in-situ analysis) or in the transient stage between the hosts and the server (in-transit analysis). Both in-situ and in-transit analysis can

potentially improve the science that volunteers' hosts perform. In both cases, the data ultimately received by the project server is a reduced, processed amount that summarizes the scientific directions of the simulations. When receiving this new knowledge rather than the raw data, the server is provided with run-time insights in the simulation's convergence toward meaningful results, allowing the scientists to stop or steer those simulations whose patterns explore solutions that violate scientific laws. At the same time, the increasing amount of computing devices available on the market (e.g., smart phones and tablets) can potentially join volunteer computing projects; this large number of hosts can eventually result in a large number of data communication between the hosts and their server to the point that the server becomes the bottleneck.

In this paper, we claim the urgent need for studying data communication between the hosts and a server in volunteer computing projects. We address this need by proposing the modeling and analysis of the network bandwidth between the hosts and the server for an established volunteer computing project, the Docking@Home project. Specifically, to answer the question whether the network is able to support the requirements for in-situ and in-transit analysis as new devices join the project, we define a statistical model to predict the metric of interest (i.e., the host bandwidth). Once validated, our model can provide us with insightful information to study different simulation scenarios for the volunteer computing project. While the dataset used for the modeling, validation, and analysis are specific for the Docking@Home project, the overall methodology can be easily applied to other distributed computing systems. The key contributions of this paper are:

- We propose a general methodology to model the network bandwidth of a volunteer computing project.
- We apply the methodology to model and predict the bandwidth between the hosts and the server of a volunteer computing project with more than 280,000 hosts over a 5-year period.
- We validate the accuracy of the proposed model and its predictions by using different methods under realistic working conditions.

The rest of this paper is organized as follows: Section II defines the data collection and modeling methodology; Section III describes the bandwidth modeling for Dock-

ing@Home; Section IV discusses the model validation; and Section V concludes the paper by summarizing the results and introducing future work.

II. MODELING METHODOLOGY

Our proposed statistical model predicts the network bandwidth for a specific time and takes into account both download and upload bandwidth. In the following, we discuss the real dataset of the traces and the methodology used for the modeling.

A. Trace Characterization

To extract the trace used for our modeling, we sample the real bandwidth from the Docking@Home project. The Docking@Home project simulates the behavior of ligands when docking into the active site of a protein. This project utilizes the BOINC software [1] to harness the computing power and storage capacity of hosts owned by the general public.

The Docking@Home trace covers the data of more than 280,000 hosts connected to the Internet between September 11, 2006 and May 5, 2014 [6]. This trace is publicly available in the Failure Trace Archive (<http://fta.scem.uws.edu.au/>) [10]. New entry in the trace is added every time a host contacts the server (i.e., the BOINC server); data includes host resource measurements such as processor speed, memory size, disk size, as well as download and upload bandwidth. In order to clean the data from possible recording noise and transmission error, we exclude hosts with download/upload bandwidth less than zero or more than 10^3 Gbps. Therefore, we removed about 16% of total hosts from the dataset.

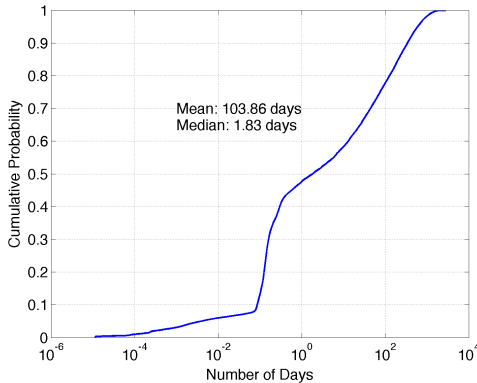


Fig. 1. Distribution of host lifetime of Docking@Home over an interval of time ranging from September 11, 2006 to April 1, 2014.

To make sure that the trace is sufficiently long for the modeling, we exported the host lifetime. We define the lifetime of a host as the time between the first and last connection to the server. The distribution of the host lifetime in Docking@Home is plotted in Fig. 1. The average lifetime is about 103 days that is slightly less than other volunteer computing projects but still sufficient for our purpose of modeling the host bandwidth [9]. Note that for this figure we exclude the hosts which connected

to the server after April 1, 2014 to avoid the impact of short lifetime.

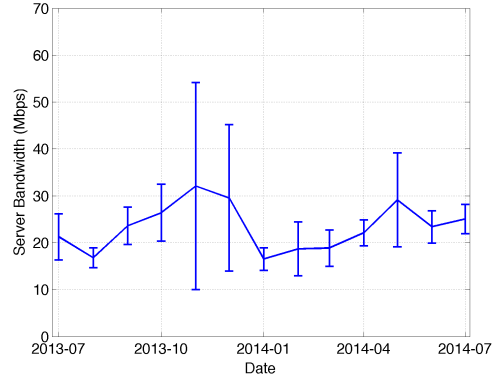


Fig. 2. Bandwidth usage of the Docking@Home server at the University of Delaware.

Since we are modeling the host bandwidth, we must make sure that the project server is not a bottleneck. As a matter of fact, it has been shown that server bandwidth could be a system bottleneck when server cannot handle the traffic generated/requested by volunteer computing hosts [3]. Thus we analyze the bandwidth usage of the server at the University of Delaware for the duration of the project. Since the beginning of 2009, the server has been connected to an 1Gbps network connection; Fig. 2 shows that only 10% of this bandwidth is used. Under these conditions, we can be confident that the project server is not saturated and the host bandwidth can reflect the real bandwidth of volunteer computing hosts.

B. Host Analysis

The host bandwidth is a relevant metric of an active host. An *active host* at time T is a host that had connected to the server before time T and whose last connection to the same server takes place after time T [9]. It is worth noting that the host activity, meant as the time a host is active, is different from the host availability; the latter was investigated in detail in previous research and refers to the time a host is exclusively dedicated to the execution of the scientific simulations [11].

Fig. 3 shows the number of active hosts in Docking@Home for the duration of four years (i.e., from January 1, 2009 to December 31, 2012). As it can be seen in the figure, there are on average more than 10,000 active hosts in this period; thus this period is sufficiently populated by active hosts to be used for the modeling. The project did not have lots of active hosts before 2009 as it was in an early stage of recruitment. Note that we use data collected during the Year 2013 for the model validation, which therefore is not included in the interval of time used for building the model.

In order to identify the trend of host bandwidth over time, we plot the average download and upload rates of active hosts from 2009 to 2012 in Fig. 4. The mean download bandwidth is about 30 times more than the mean upload bandwidth for each year. This observation indicates the demand for a higher

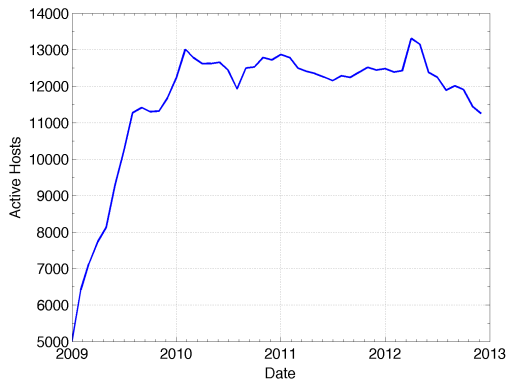


Fig. 3. Distribution of active hosts in Docking@Home from 2009 to 2012.

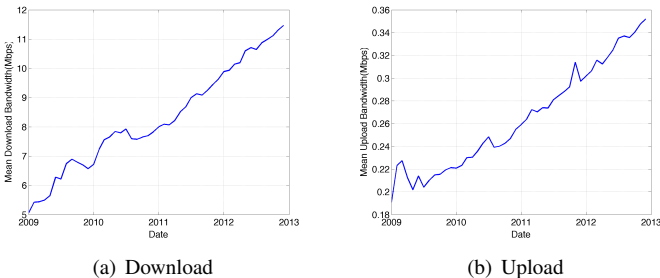


Fig. 4. Average download and upload bandwidth for active hosts in Docking@Home.

download rate by the Internet users. As plotted in Fig. 4, the average download bandwidth increases about 20% per year while the upload bandwidth increases about 15% per year. This increase in the mean values is less than we expected based on Nielsen’s Law; Nielsen stated that the network connection speed for high-end home users increases 50% per year. One possible answer for this behavior is the shorter lifetime or the lack of contribution from hosts with a better bandwidth in the volunteer projects.

To complete the bandwidth analysis over time, we also plot the boxplot of the average bandwidth for active hosts in Fig. 5 in which the three 25%, 50% and 75% quartiles are illustrated. The figure outlines how the interquartile range (IQR) for the upload bandwidth is smaller than the download bandwidth. This reveals that the spread of the distribution for download is larger than the upload and should be considered to candidate the statistical models. Moreover, we can observe an increasing of the variance for the hosts bandwidth over time for both download and upload rate. Note that the upload bandwidth has more outliers in the data and this may result in a hindrance to find a good model.

III. BANDWIDTH MODELING

A. Host Bandwidth Correlations

The first step towards discovering the best model for the host bandwidth is finding possible correlations (e.g., between upload and download bandwidth, and between host bandwidth

TABLE I
P-VALUE RESULTS FROM GOF TESTS FOR THE 2012 DATASET (AD, KS).

Model	Download	Upload
Exponential	0.026 0.003	0.403 0.255
Gamma	0.179 0.077	0.492 0.378
Log-normal	0.548 0.391	0.608 0.477
Weibull	0.323 0.188	0.442 0.311

and the host time zone). We examined the correlation between upload and download bandwidth using the Pearson correlation coefficient and we found out that there is no clear correlation. This can be justified by the fact that most Internet users tend to subscribe to Internet service providers based on the download rate as they do not care much about the upload rate.

We also checked the possible correlation between the host bandwidth and the host time zone. The results of this analysis is plotted in Fig. 6 where the download and upload bandwidth for the six largest timezones in the project are illustrated. The x-axis in these figures shows the offset from the GMT. As one can see, there is no strong correlation between the timezone and the host bandwidth. However, we can observe that hosts in Eastern North America (-4,-5) have slightly higher download and upload bandwidth, which might be because of the proximity to the project server.

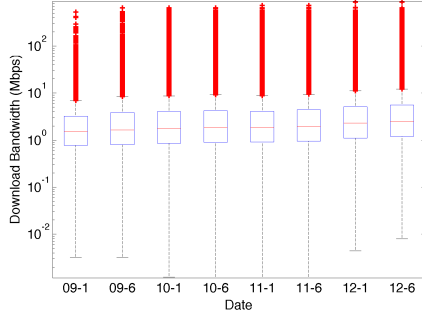
The absence of obvious correlations in the host bandwidth drives our modeling approach towards the design of an *independent* statistical model that predicts the download and upload rate for a given host and at a specific time.

B. Statistical Modeling

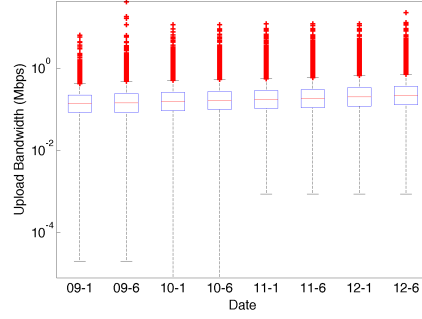
To design the statistical model, we first examine the distributions of host bandwidth for different time periods. Fig. 7 shows the mass-count disparity of the host bandwidth in 2010 and 2012. From these figures we can clearly observe that about 10% of total bandwidth is created by 90% of low bandwidth hosts. Thus, the 10% of high bandwidth hosts contributed for the rest of 90% of total bandwidth. This information reveals the *long-tail* behavior of the host bandwidth and suggests to us the need to focus on modeling of higher bandwidths by using a statistical model that exhibits this behavior.

In our search for the suitable model, we considered various statistical models (i.e., Weibull, Log-normal, Gamma and Exponential) which can be a good fit for this type of behavior. For all these distributions, we conducted parameter fitting using maximum likelihood estimation. We measured the goodness of fit (GoF) of the resulting distributions using probability-probability plots and two quantitative methods, i.e., KS-test and AD-test [11]. Due to space limitation, we do not show the full results of these tests and only present the GoF results in form of p-value for the 2012 dataset in Table I. A model with a higher p-value is the better fit for the dataset. The results show that we can model the hosts bandwidth (both download and upload) using the Log-normal distribution for each year with a high level of accuracy.

Since we are looking for a model to predict the host bandwidth over time, we need to integrate the time parameter

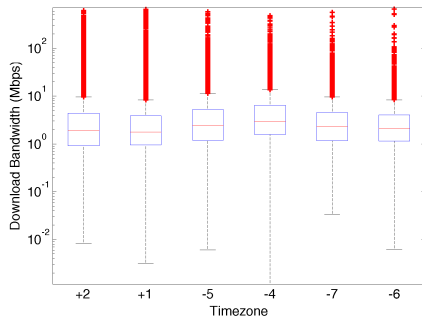


(a) Download

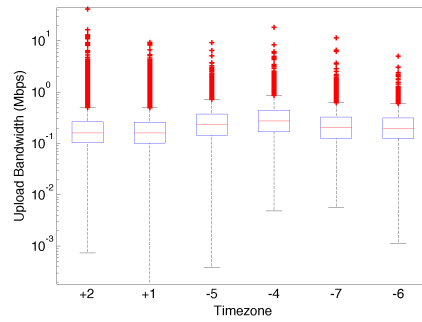


(b) Upload

Fig. 5. Download and upload bandwidth for active hosts from 2009 to 2012.

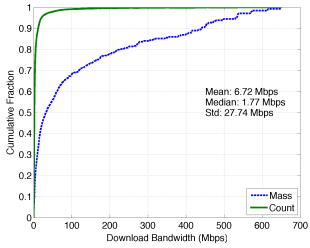


(a) Download

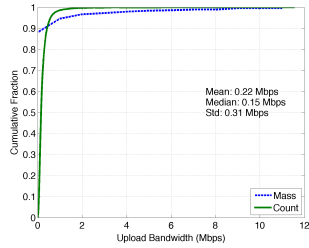


(b) Upload

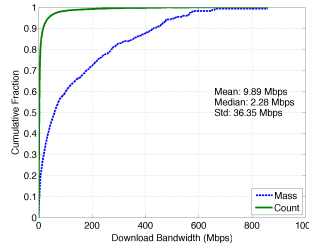
Fig. 6. Download and upload bandwidth for active hosts in different timezones from 2009 to 2012.



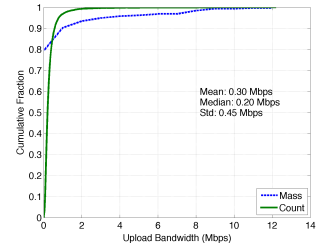
(a) Download 2010



(b) Upload 2010



(c) Download 2012



(d) Upload 2012

Fig. 7. Mass-count of download and upload bandwidth for active hosts in 2010 and 2012.

as well. To this end, we need to determine the required parameters by Log-normal distribution as the best fit model. Log-normal has two parameters: the mean μ and standard deviation σ of the variable whose logarithm is *normally* distributed. Thus, we need to find the mean and standard deviation of the logarithm of the host bandwidth over time. We extract the mean and variance (σ^2) of the logarithm download and upload rates of active hosts for each month from 2009 to 2012. We fit the data with different exponential and polynomial functions and find the following exponential function as a closet fit for these values:

$$f(t) = ae^{bt} \quad (1)$$

where a and b are the parameters to be fitted for each metric and t is the time variable. We define the time as $t = date - Rdate$, where $date$ is the given date per year and month and $Rdate$ is the reference date, which is the start date of the modeling trace (i.e., January 2009). We fit this function separately for the mean and variance of host download and upload bandwidth. The results of fittings are depicted in Fig. 8 and Fig. 9 for the download and upload bandwidth, respectively. As one can see in these figures, the proposed exponential function is a good model to predict the mean and variance of the host bandwidth, except for the variance of the upload bandwidth. The parameters for the

TABLE II
PARAMETERS FOR THE FITTED EXPONENTIAL MODEL.

Model	a	b	R^2
Mean download	5.698	$0.493e^{-3}$	0.9748
Variance download	590.7	$0.729e^{-3}$	0.9227
Mean upload	0.1955	$0.404e^{-3}$	0.9735
Variance upload	0.129	$0.745e^{-3}$	0.0376

fitted exponential model are given in Table II where R^2 is the coefficient of determination to show the goodness of fit. As we expected, this exponential model is not a best fit for the variance of upload bandwidth, as it can be seen in Fig. 9(b). However, it is the closest fit among the extensive parameter and the model fitting studies that we performed; thus, we keep this simple model as the most effective and study its accuracy in next section.

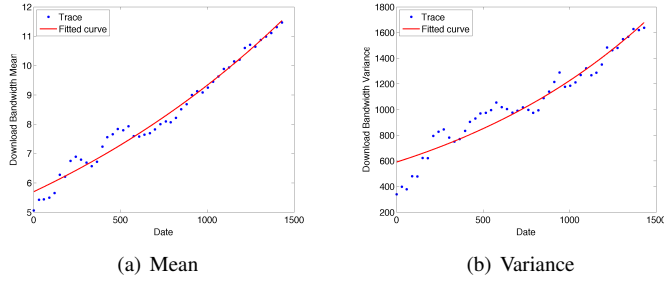


Fig. 8. Fitting of the mean and variance for download bandwidth.

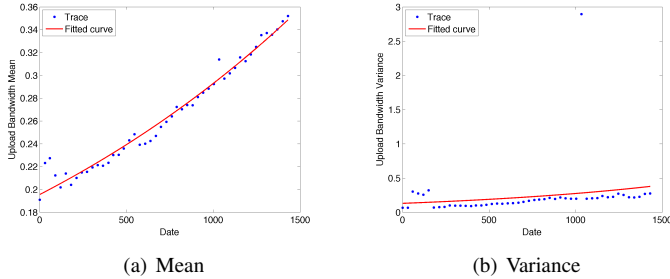


Fig. 9. Fitting of the mean and variance for upload bandwidth.

IV. MODEL VALIDATION

In the following, we present the validation of the proposed statistical models. We use two methods to validate the models: a model-based and a simulation-based validation.

A. Model-based Validation

For the validation, we utilize the exponential model given in Eq. 1 and the values in Table II to find the mean and variance of the host bandwidth. Then we generate a number of hosts with the download and upload bandwidth using the Log-normal distribution, given the calculated mean and variance. We predict the host bandwidth for May 2013 and plot the cumulative distribution function (CDF) of download

TABLE III
INPUT PARAMETERS FOR THE SIMULATION.

Parameter	Distribution
Download size	Uniform (12MB,18MB)
Upload size	Uniform (0.5MB,1.2MB)
Job runtime	Normal ($\mu = 1.2hrs$, $\sigma = 0.9hrs$)

and upload rates against the real data in Fig. 10. As previously mentioned, the data for Year 2013 is not included in the modeling, so these figures reveal that the proposed model can predict the host bandwidth with a good degree of accuracy.

As one can see in Fig. 10(a), the model for download bandwidth has higher accuracy than the model for upload bandwidth. The average relative error for the download and upload bandwidth are about 3% and 15%, respectively. The main reason for this discrepancy in the model accuracy is the complex behavior of the upload variance and the low accuracy of the exponential model as listed in Table II. This observation is consistent for other data and dates. The accuracy of the proposed model for the upload bandwidth could be improved at the cost of using a more complex model (e.g., two exponential components model). This approach is part of our future work.

B. Simulation-based Validation

Since most of the Big Data applications focus on utilizing bandwidth to transfer data for processing using Internet-based computing systems, we design a simulation to evaluate a similar scenario. As we mentioned earlier, we have a client-server system where the server dispatches the jobs to hosts and then waits for the results. Hosts must download the input files and upload the results after job completion. In this case, the host service time is defined as the summation of the time for download, data processing, and upload the results. So if we consider jobs as the incoming requests, then each host provides a service with a *variable* service time. This behavior can be simulated in a queuing system where we have two types of service time generated by model or real traces. Hence, we can use $M/Model/1$ and $M/Trace/1$ queuing systems for model evaluation. This technique has been used for the similar model validation [11].

In order to simulate the two queuing systems, we implemented a discrete-event simulator using the Objective Modular Network Testbed in C++ (OMNeT++) [12]. The simulation parameters are given in Table III, which have been extracted from a real application [5]. We used the generated download and upload bandwidths for a set of hosts in May 2013 from Section IV-A to calculate the service time of $M/Model/1$ queue. The same date is also used to generate the service time of $M/Trace/1$ from the real dataset. We consider the exponential distribution for the inter-arrival time of input jobs to focus only on the target performance metric, which is the response time of the queue.

The simulation results for the response time of the queues while using the proposed model or the real traces as the service time are depicted in Fig. 11. As one can see, the

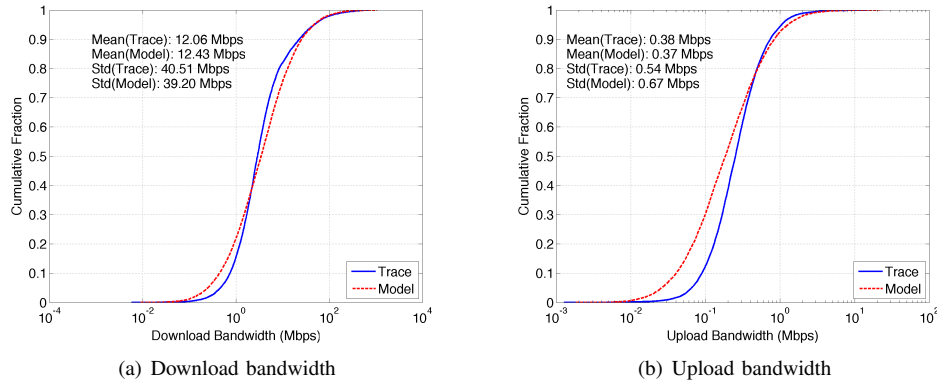


Fig. 10. The CDF of the predicted and real download and upload bandwidth for May 2013.

proposed model shows a close fit to the real data. We also observe the similar results when we used different dates. The simulation-based validation confirms the high accuracy of the proposed model and shows that this model is a good candidate in analyzing the bandwidth requirements for the Big Data applications.

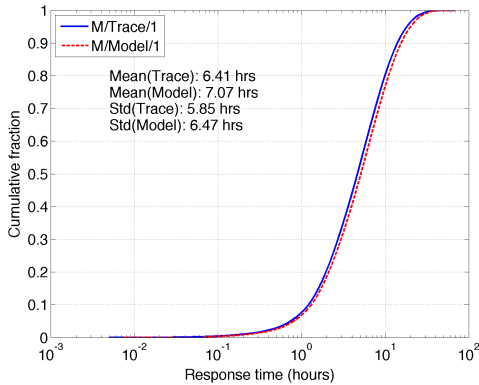


Fig. 11. Simulation-based validation.

V. CONCLUSIONS

In this paper, we present a general method to analyze and model the host bandwidths of volunteer computing projects; we use real traces of the Docking@Home project from 280,000 hosts over the Internet in a 5-year period. In our analysis we do not observe any obvious correlation in the host bandwidth. We successfully mimic the hosts bandwidth (download and upload) using the Log-normal distribution in combination of an exponential model to predict the mean and variance. Validation results in which we first compare the predicted download and upload bandwidths and then predict the host service times versus real traces confirm the high accuracy of our model. Thus the model is a good candidate to analyze the bandwidth requirements for the Big Data applications. Work in progress includes the study of scenarios in which our model is used for the prediction of in-situ and

in-transit analysis of data generated in Docking@Home and other volunteer computing projects.

REFERENCES

- [1] D. Anderson. Boinc: A system for public-resource computing and storage. In *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, Pittsburgh, USA, 2004.
- [2] D. Anderson and G. Fedak. The Computational and Storage Potential of Volunteer Computing. In *Proceedings of the IEEE International Symposium on Cluster Computing and the Grid (CCGRID'06)*, 2006.
- [3] D. P. Anderson, E. Korpela, and R. Walton. High-performance task distribution for volunteer computing. In *e-Science*, pages 196–203, 2005.
- [4] J. C. Bennett, H. Abbasi, P.-T. Bremer, R. Grout, A. Gyulassy, T. Jin, S. Klasky, H. Kolla, M. Parashar, V. Pascucci, P. Pebay, D. Thompson, H. Yu, F. Zhang, and J. Chen. Combining in-situ and in-transit processing to enable extreme-scale scientific analysis. In *Proceedings of the 2012 IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, November 2012.
- [5] T. Estrada, K. Reed, and M. Taufer. Modeling job lifespan delays in volunteer computing projects. In *Proceedings of the 9th IEEE International Symposium on Cluster Computing and Grid (CCGrid)*, 2009.
- [6] T. Estrada and M. Taufer. On the effectiveness of application-aware self-management for scientific discovery in volunteer computing systems. In *Proceedings of the IEEE/ACM International Conference for High Performance Computing, Networking, Storage, and Analysis (SC12)*, November 2012.
- [7] T. Estrada, B. Zhang, P. Cicotti, R. Armen, and M. Taufer. A scalable and accurate method for classifying protein-ligand binding geometries using a mapreduce approach. *Comp. in Bio. and Med.*, 42(7):758–771, 2012.
- [8] H. Hasan Abbasi, M. Wolf, G. Eisenhauer, S. Klasky, K. Schwan, and F. Zheng. Datastager: scalable data staging services for petascale applications. *Journal of Cluster Computing*, 13(3):277–290, 2010.
- [9] E. M. Heien, D. Kondo, and D. P. Anderson. Correlated resource models of internet end hosts. In *ICDCS*, pages 278–287, 2011.
- [10] B. Javadi, D. Kondo, A. Iosup, and D. Epema. The Failure Trace Archive: Enabling the comparison of failure measurements and models of distributed systems. *Journal of Parallel and Distributed Computing*, 73(8):1208 – 1223, 2013.
- [11] B. Javadi, K. Matawie, and D. P. Anderson. Modeling and analysis of resources availability in volunteer computing systems. In *IPCCC*, pages 1–9, 2013.
- [12] A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. In *International Conference on Simulation Tools and Techniques for Communications, Networks and Systems*, Marseille, France, 2008.
- [13] B. Zhang, T. Estrada, P. Cicotti, and M. Taufer. Enabling in-situ data analysis for large protein folding trajectory datasets. In *Proceedings of the 28th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2014.