Modeling Crowdsourcing Platforms to Enable Workforce Dimensioning

Christian Schwartz, Kathrin Borchert, Matthias Hirth, Phuoc Tran-Gia

University of Würzburg
Institute of Computer Science
Würzburg, Germany
{christian.schwartz|kathrin.borchert|matthias.hirth|trangia}@informatik.uni-wuerzburg.de

***Abstract*—Crowdsourcing platforms provide an easy and scalable access to human workforce that can, e.g., provide subjective judgements, tagging information, or even generate knowledge. In conjunction with machine clouds offering scalable access to computing resources, these human cloud provider offer numerous possibilities for creating new applications which would not have been possible a few years ago. However, in order to build sustainable services on top of this inter-cloud environment, scalability considerations have to be made. While cloud computing systems are already well studied in terms of dimensioning of the hardware resources, there still exists little work on the appropriate scaling of crowdsourcing platforms. This is especially challenging, as the complex interaction between all involved stakeholders, platform providers, workers and employers have to be considered.**

The contribution of this work is threefold. First we develop a model for common crowdsourcing platforms and implement the model using a simulative approach, which is validated with a comparison to an analytic $M^{[X]}/M/c - \infty$ system. In a second step we evaluate inter-arrival times as well as campaign size distributions based on a dataset of a large commercial crowdsourcing platform to derive realistic model parameters and illustrate the differences to the analytic approximation. Finally, we perform a parameter study using the simulation model to derive guidelines for dimensioning crowdsourcing platforms, while considering relevant parameters for the involved stakeholders, i.e., the delay before work on a task begins and the work load of the workers.

I. Introduction

Cloud computing has been seen as one of the main drivers fostering the development of numerous new services in the Internet in recent years. The relative low infrastructure costs enable even small start-ups building up complex systems involving a large human workforce with low investments during the ramp-up phase. Furthermore, the flexibility and scalability of this approach allows for a continuous and timely growth of the service to cope with an increasing number of users.

Following the cloud paradigm, crowdsourcing platforms can be seen as cloud providers for human workforce. Here, work is distributed to a large group of anonymous users, who complete it for small revenues or other incentives. Similar to cloud computing platform, the workforce can be flexibly scaled on a timely base and at relatively low cost. This in turn enables a wide range of new services based on, e.g., subjective judgements or requiring cognitive work, which is not yet possible to automate.

In order to make these human clouds easier accessible, most of the platform providers offer programming interfaces, which allow an automatic managing of the work on the platforms or even developing mixed human-machine cloud applications. However, in this context of commercial large-scale Internet applications, the question of service guarantees arises. Here, one of the most simple measures is the time until work on a task begins. This question is already actively researched in the machine-cloud environment, where the device infrastructure can be scaled quite freely by investing money. In crowdsourcing platforms, this is more complex, because new workforce cannot simply be acquired by monetary investments, but new users have to be willing to join the service. Therefore, the platforms have to be "attractive" to the users.

In this paper we address the topic of dimensioning crowdsourcing platforms by applying well known models from queuing theory [1] as well as simulative approaches, but also consider the special aspects of crowdsourcing related issues. We assume that users submitting work to the platform want to minimize the time it takes to complete the submitted work. As the time to actually complete the task depends on the complexity of the task and not the number of workers, we here focus on the task pre-processing delay. In parallel, the users completing tasks want to maximize their income per time, which can be estimated by the workers utilization. The contributions of this paper are threefold. First we present a model for evaluating completion times and worker utilization for commercial crowdsourcing platforms. This models is implemented as simulation and the simulation is validated by a comparison with an analytic $M^{[X]}/M/c-\infty$ model. Second we analyse a dataset of a large commercial crowdsourcing platform to derive realistic models for the arrival process of new work on the platform as well as the amount of submitted work. These results are used in the simulation to analyse the differences between the real world parameters and the analytic approximation by the $M^{[X]}/M/c - \infty$ model. In a third step we use the simulation model and the parameters derived from the platform to conduct a parameter study analysing the impact of different work arrival patterns on the required number of workers, while also considering an appropriate worker utilization.

The reminder of the paper is structured as follows. First, we give a brief overview of crowdsourcing in general and related work in Sec. II. Then, we describe our proposed model of a commercial crowdsourcing platform in Sec. III. Here, we also detail on the simulative implementation and the analytic validation of the implementation. The analysis of the real-world dataset is presented in Sec. IV and the results of the parameter study in Sec. V. Sec. VI finally concludes the paper.

II. Background and Related Work

This section first gives a short introduction in the general concept of crowdsourcing with a special focus on micro-task platforms. Thereafter, we discuss related work of modelling approaches for these platforms, as well as related work to our modeling approach in general.

A. Crowdsourcing

The term crowdsourcing is a neologism combining the terms 'crowd' and 'outsourcing'. It was first introduced by Jeff Howe in 2006 [2] and describes a new form of work organisation with a smaller granularity than traditional forms [3]. In contrast to traditional forms of work organization, work is divided in individual *tasks* that can be completed independent of each others. These tasks are not directly assigned to an employee but published on a *crowdsourcing platform* in form of an open call. Users publishing tasks on crowdsourcing platforms are referred to as *employer*, users accepting and accomplishing tasks as *workers*. Workers can freely choose which task to work on, other than in traditional forms of work organization. In commercial applications, workers are usually paid for successfully completed tasks and do not receive hourly wages. Crowdsourcing platforms act as mediators in this environment, i.e., provide infrastructure for posting tasks and submitting task results and negotiate in case of disagreements between workers and employers.

The crowdsourcing approach is used for a large variety of nonprofit, academic, and commercial applications, including information gathering during crisis [4], analysis of astronomic images [5], and by numerous large scale labour providers, e.g., Amazon Mechanical Turk (MTurk)[1], Microworkers[2], and Innocentive[3]. Depending on the specific use case, the features of the crowdsourcing platform, the workers, and employers differ. Therefore, we focus in this work on commercial micro-tasking platforms for the development and evaluation of our model.

Commercial micro-tasking platforms like MTurk or Microworkers are specialized labour provides for very fine granular tasks that can be easily completed a humans within a few second to a few minutes, but cannot be solved using automatic approaches. These tasks include, e.g., image tagging, text creation, or subjective ratings. As the tasks are highly

[1]http://www.mturk.com, Accessed Jul 2015
[2]http://www.microworkers.com, Accessed Jul 2015
[3]http://www.innocentive.com, Accessed Jul 2015

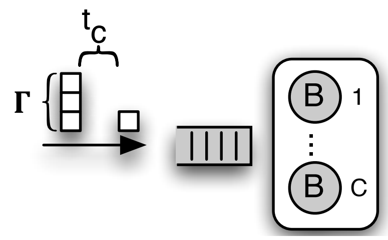


Fig. 1: Crowdsourcing Platform Model

repetitive, they are often submitted by the employers in from of *campaigns*, representing batches of similar tasks.

B. Related Work

Several efforts have already been made to describe certain aspects of micro-tasking platforms in analytic models. Faradani et al. [6] modelled the arrival process of workers using a non-homogeneous Poisson process in order to derive optimal pricing strategies for the employers. The model is based on a crawled dataset from MTurk including about 130,000 campaigns with in total over 4,000,000 tasks. Wang et al. [7] analysed the completion time of crowdsourcing campaigns using a survival analysis based on a crawled dataset from MTurk consisting of more than 160,000 campaigns and approximately 6,700,000 tasks over a period of 15 months. They were able to show the impact of time-independent factors, e.g., the payment or the type of the task, on the completion time. In order to optimize the costs and the completion times of single jobs, Bernstein et al. [8] use a $M/M/c$ queueing model to describe a crowdsourcing retainer approach. Here, workers are paid for staying online to wait for potentially upcoming tasks. The model was validated in a proof-of-concept experiment with 500 users on MTurk.

In contrast to existing work, we use a dataset directly provided from the commercial crowdsourcing platform Microworkers for our evaluation instead of a crawled subset of data. This allows us to gain a holistic view of all campaign submissions of the platform and the campaign sizes. We further extend the existing work of Bernstein et al. [8] by also considering the time until the processing of any task in the campaign begins, not only the time until the completion of the first one.

III. Model

In our model, schematically depicted in Fig. 1, we consider a crowdsourcing platform employing c workers. The time between two campaigns being submitted is given by the random variable t_c. Each campaign consists of a number of tasks, distributed according to the random variable Θ. We assume that each task is then completed by one of the c workers in order of arrival. The time required for completion is given by the random variable B.

From this model we study two metrics in order to evaluate the performance of the crowdsourcing platform. First, we consider the utilization ρ for all workers. This can be interpreted as a measure for workers about how much they can earn on the platform and should be maximized in order to keep current workers and attract new ones. Furthermore, we seek to obtain the mean task pre-processing delay $E[D]$, i.e., the time occurring before a worker begins to work on a task. This measure is relevant for the employer and should be minimized. We decided to use the $E[D]$ instead of the average completion time of the campaigns, as the completion time also depends on the task length, which is under control of the employer and not the platform operator.

In this section we first introduce an analytical model in Sec. III-A, which will be used to validate the simulation model developed in Sec. III-B. A comparative validation of analytical and simulative model is then performed in Sec. III-C.

A. Analytical Consideration

First, in order to provide exact results, we consider the crowdsourcing platform as a $M^{[X]}/M/c-\infty$ model. Here, both the campaign inter-arrival time t_c as well as the time to complete a task B are exponentially distributed with mean $E[t_c] = {}^1\!/\!_\lambda$ and $E[B] = {}^1\!/\!_\mu$, respectively. This seams feasible because of both the large number of employers submitting tasks and the large number of workers completing them. Furthermore, we model the number of tasks per campaign Θ using a geometric distribution with mean $E[\Theta] = {}^1\!/\!_p$.

State probabilities [1] can be obtained by solving the state equations, yielding

$$x(i+1) = x(i)\frac{\lambda + \mu \min(i,c)}{\mu \min(i+1,c)} - \sum_{k=0}^{i-1} x(k)\lambda\theta(i-k)$$

for $i \geq 0$ tasks unserviced in the platform. The state probability for $x(i+1)$ only depends on the state probabilities for $x(k)$ for $k \leq i$, lending itself to an iterative numerical computation. Note that $x(0)$ cannot be computed in such a manner and has to be obtained using the normalizing equation $1 = \sum_{i=0}^{\infty} x(i)$.

Thus, we first calculate $x(1)$ as a multiple of $x(0)$ that is $x(1) = \zeta(1)x(0)$. Then, we compute each $x(i)$ depending on the previously computed values for $x(j), j \leq i$ and thus obtain expressions $x(i) = \zeta(i)x(0)$. Once a sufficient number κ of state probabilities have been computed, the normalizing property is applied, and $x(0)$ is computed as

$$x(0) = \left(1 + \sum_{i=1}^{\kappa} \zeta(i)\right)^{-1}.$$

Finally, we can calculate the actual state probabilities as $x(i) = \zeta(i)x(0)$ for all $0 < i \leq \kappa$.

Based on these state probabilities, we obtain the mean worker utilization as

$$\rho = \sum_{i=0}^{\kappa} \min(i,c)x(i) = \frac{\lambda E[\Theta]}{c\mu}.$$

.

Next, we obtain the mean queue length Ω of the system as

$$\Omega = \sum_{i=c}^{\kappa} (i-c)x(i).$$

Now, we consider the mean task pre-processing delay and with Little's theorem applied to the systems queue, we get

$$\lambda E[\Theta]E[D] = \Omega.$$

We solve for task the pre-processing delay $E[D]$ and obtain

$$E[D] = \frac{\Omega}{\lambda E[\Theta]}.$$

B. Simulation

In order to allow for a larger variety of campaign inter-arrival time distributions, we implement a discrete event simulation using the OMNet++ simulation framework[4]. We augment the framework with support for bulk arrivals and support of empiric distributions taken from measurements described in Sec. IV. Similar, to the queueing model introduced in Sec. III-A, we consider campaign inter-arrivals according to a distribution t_c and a campaign size of Θ tasks. Task length is given by a distribution B and tasks are stored in an unbounded queue before being sent to service to the available workers c. During simulation, we record the mean worker utilization ρ as well as the mean task pre-processing delay $E[D]$.

C. Validation

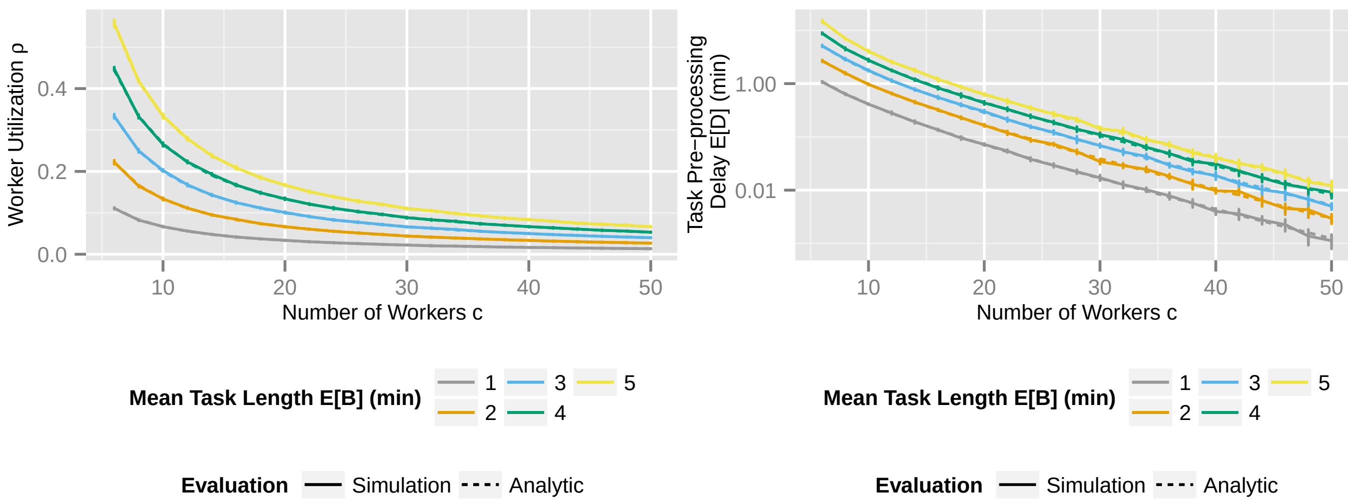
In this section, we validate the simulative model by comparing the metrics ρ and $E[D]$ for a representative parameter set with those obtained from the analytic model. We consider exponential campaign inter-arrival times with a rate of 4 campaigns per hour, and a campaign size Θ geometrically distributed with a mean of 100 tasks per campaign. For the task duration B we consider a set of possible values, to accommodate for different task types, between 1 and 5 minutes per task. In both simulation and analytical model, we consider between 5 and 50 workers.

Results are shown in Fig. 2. Simulative respective analytical results are shown by different by line types. However, due to the good fit of the analytic and simulative model, the line showing the simulative results completely covers the analytic results. For the simulation we give 95% confidence intervals based on 10 replications. In this, and all following figures, confidence intervals are given as error bars. For each simulation we consider a simulation duration of 1500 hours and accommodate for a transient phase of 150 hours. We observe that for both worker utilization and task pre-processing delays, the analytical results are well within the confidence intervals.

IV. Measurements

In this section we analyse a large dataset from a commercial crowdsourcing platform to derive realistic model parameters and compare the model based on these results with the analytic approximation.

[4] http://www.omnetpp.org/, Accessed Jul 2015



(a) Worker Utilization

(b) Task Pre-processing Delay

Fig. 2: Comparison of Simulation with Analytic Model

A. Deriving Realistic Model Parameters

Our analysis is based on a large dataset from the commercial micro-tasking platform Microworkers. The dataset contains information about more than 160.000 campaigns and more than 18.000.000 tasks submitted to the platform between May 2009 and Jan 2015, including the number of tasks per campaign as well as the time of the submission of the campaign.

First we study the inter-arrival times of the campaigns. During the observation period, the platform faced some downtime due to software updates or changes of the technical infrastructure. Here, no campaigns could be submitted resulting in relatively large campaign inter-arrival times. In our model we only consider the regular operation of the platform, therefore we removed all inter-arrival times larger then 97.5 % quantile of all observed values. We observe a mean campaign inter-arrival time of 14.46 min with a standard deviation 20.78 min. Fig. 3a shows the CDF of considered inter-arrival times, as well as the fitted distribution. For the fitting we considered several possible distributions but found the Gamma-distribution defined by shape α and rate β to be the most suitable. Using `fitdistrplus` for the R language we derive the distribution parameters by moment fitting and obtain $\alpha = 0.484071$ and $\beta = 2.009$.

Next, we consider the campaign sizes. The smallest possible campaign sizes on Microworkers is 30 tasks, however our dataset contained a few test campaigns with a small size. These test campaigns, as well as outliers larger than the 97.5 % quantile of the campaign size have been removed from the considered dataset. In total 3.7 % of the original dataset were filtered by these conditions, the remaining data resulted in a mean campaign size of 97.01 tasks and a standard deviation of 103.41 tasks. The CDF of the campaign sizes is depicted in Fig. 3b, together with the corresponding fitted distribution.

We observe that a very high share (35 %) of campaigns has only the minimum size of 30 tasks. Further, campaign sizes which are a multiple of ten or a multiple of 100 are quite frequent, due to rounding by employers.

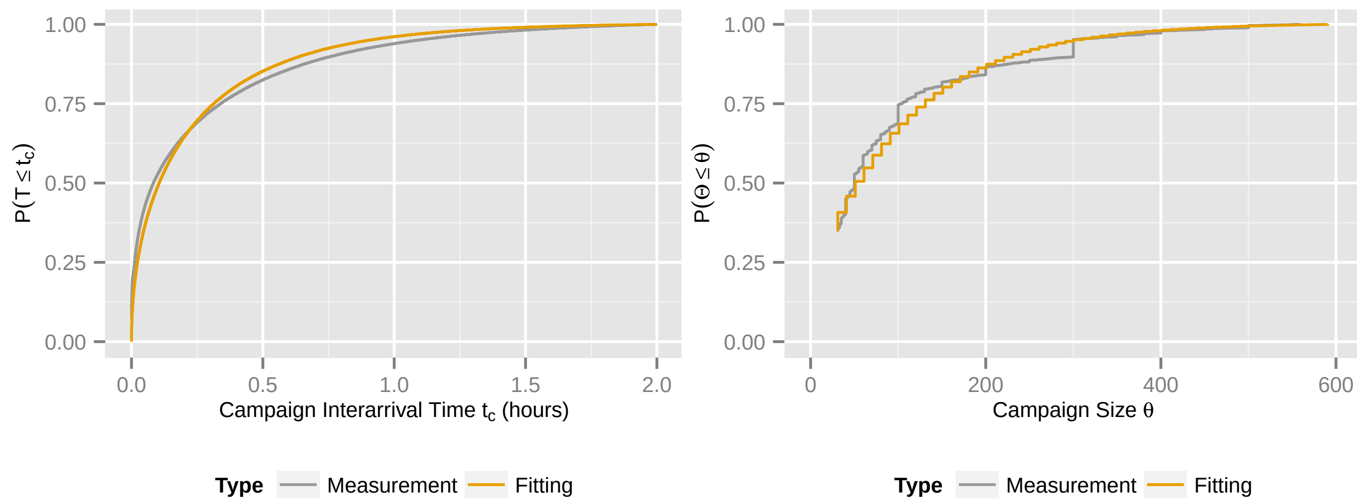
In order to obtain a suitable analytic distribution for the empiric values, we divided the observed values and use the following piecewise defined distribution.

$$P(\Theta = \theta) \sim \begin{cases} p_{\theta_{\min}} & \text{if } \theta = \theta_{\min} \\ GEOM(\theta) \cdot 10 + (\theta_{\min} + 1) & \text{else} \end{cases}$$

The minimum campaign size $\theta_{\min}$ is observed with a fixed probability $p_{\theta_{\min}}$, while all campaign sizes larger than $\theta_{\min}$ follow a shifted and scaled geometric distribution. Due to the relatively high frequencies of campaign sizes being multiples of 10 and 100, it is only possible to achieve a good fitting either for the lower or the higher region of the geometric part. As an overestimation of the campaign size will give us an upper bound of the platform work load, we decided to put a stronger emphasis on correct fitting of the larger campaign sizes. Again we used `fitdistrplus` to estimate $p = 0.086$ using quantile matching for the 90% quantile. $\theta_{\min} = 30$ and $p_{\theta_{\min}} = 0.350$ were taken from the empirical values.

Another relevant model parameter is the processing time of the tasks, i.e., the time a single worker needs to complete one task. Unfortunately, this information cannot be obtained from our dataset. Therefore, we assume that the processing times follow a negative-exponential distribution with rate μ. Even if the exact processing times are not available, each employer has to add an estimation about the time it takes to complete a task in the campaign description. In our dataset, 87.8 % of all tasks have an estimated completion time between 1 and 6 minutes. Therefore, we consider $\mu \in \{1/6, 1/5, \ldots, 1/2, 1\}$ for the following evaluations.

Finally, the last model parameter to estimate is the number of users on the crowdsourcing platform. At the time of this



(a) Observed Campaign Inter-arrival Times t_c and Corresponding Fit (b) Observed Campaign Sizes s and Corresponding Fit

Fig. 3: Campaign Inter-arrival Times and Size with their Corresponding Fit

analysis, Microworkers had over 650.000 registered user accounts. The proposed model does not consider vacation times, i.e., the workers would have to be available 24/7. In reality, many crowdsourcing workers also only work occasionally on the platforms or only for a few tasks. Taking this into account, the number of workers to be considered in our model has to be much small than the number of workers on the real world platform and consequently we decided to estimate meaningful values based on the model parameters instead of using the given number of workers from the dataset.

B. Comparison of Simulative and Analytic Model

An important question for the later analysis is whether the analytic model from Sec. III-A can be used as an approximation or if a simulative evaluation is necessary. To this end we compared the later considered metrics ρ and $E[D]$ for 1) a simulation using the empiric distributions for the task inter-arrival times and campaign sizes, 2) a simulation using the fitted distributions derived in Sec. IV-A, and 3) the analytic model from Sec. III-A. For the analytical model we used the campaign size distribution derived in Sec. IV-A and $\lambda = 4.14 \,{}^{1}\!/_{h}$. The results are shown in Fig. 4.

The worker utilization ρ is depicted in Fig. 4a, $E[D]$ in Fig. 4b. We observe that all models result in the same values of ρ, which is not surprising when considering $\rho = \frac{E[t_c]E[\Theta]}{c\mu}$ with the mean inter-arrival time $E[t_c]$. Here, all parameters are the same for the three compared models and therefore, no significant differences can be seen.

This is different for $E[D]$. Here, large discrepancies can be observed between the model based on the empiric distributions and the analytical model. These results show that the $M^{[X]}/M/c-\infty$ model can also not be used as a worst case estimation, as it underestimates $E[D]$. In contrast to this, the simulation model based on the Gamma distribution is an accurate fit compared to the model based on the empirical values. Therefore, we continue our evaluation with the simulation model based on the Gamma distributed inter-arrival times and the piecewise defined distribution for the campaign sizes.

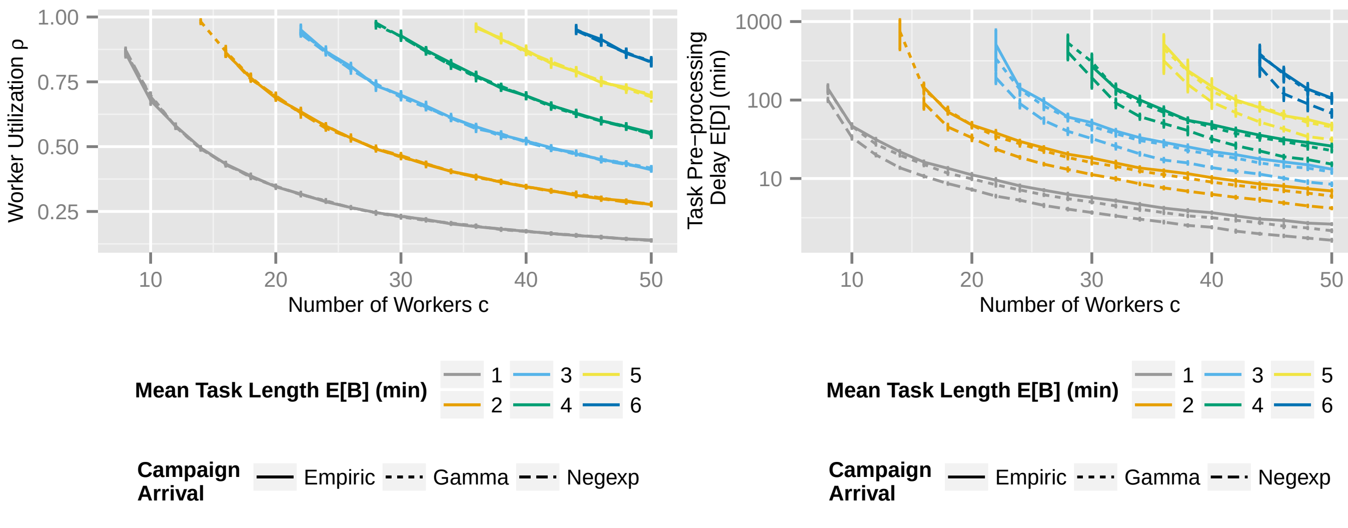
V. Numerical Evaluation

In this section we use the simulative model introduced in Sec. III-B and the measurements obtained from the Microworkers platform in order to analyse the impact of different parameters on the considered metrics. First, we study the impact of campaign inter-arrival times Sec. V-A. Then, in Sec. V-B we study trade-offs between metrics of interest for the different stakeholders. The results presented in this section can be used as guidelines for platform operators, in order to ensure that both stakeholders are sufficiently satisfied.

A. Impact of Campaign Interarrival Distributions

Campaign inter-arrival times influence both the work load of the individual workers as well as time required before a worker starts working on a task. From the perspective of an operator, understanding the influence of different inter-arrival processes is important. As shown in Sec. IV, the Gamma distribution can be used to approximate the campaign inter-arrival times as seen on the crowdsourcing platform Microworkers. In this section, we study the impact of such different processes by utilizing the parameter space afforded by the Gamma distribution and considering the impact on the metrics worker utilization and task pre-processing delay.

The characteristics of the Gamma distribution change depending on the parameters shape α and the rate β. While both shape and rate influence the mean $E[t_c] = \frac{\alpha}{\beta}$ and variance $\mathrm{Var}[t_c] = \frac{\alpha}{\beta^2}$ of the campaign inter-arrival times, only the shape influences the skewness $\mathrm{Skew}[t_c] = \frac{2}{\sqrt{\alpha}}$ of the distribution.



(a) Worker Utilization

(b) Task Pre-processing Delay

Fig. 4: Comparison of Campaign Arrival Distributions

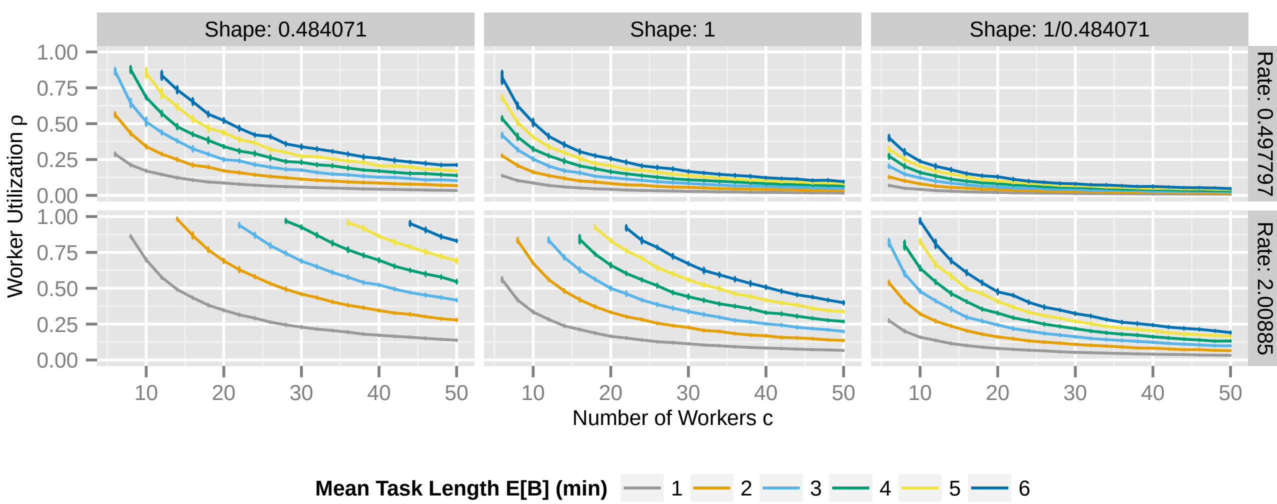


Fig. 5: Worker Utilization for Different Campaign Interarrival Times

A shape of 1 describes an exponential distribution. Increasing of the shape for the same rate changes the form of the distribution from an exponential type to a distribution which is similar to a normal distribution. By increasing the rate for the same shape the tightness of the distribution is modified. A rate less than 1 results in a distribution with a long tail. The increase of the rate decreases the broad of the distribution.

Transferred to the campaign inter-arrival process different shape and rate settings can be used to model different task types and varying the business of the platform. The range of the inter-arrival times is given by the rate and the shape defines the weighting of the different times. A lower shape means more campaigns arrive in a short time interval in combination with longer time periods without any campaign arrival.

Next, we use our simulation model introduced in Sec. III with the campaign size distribution and task completion times obtained in Sec. IV for different campaign inter-arrival times to study the impact on the worker utilization. Only stable systems, i.e., crowdsourcing platforms with an utilization $\rho < 1$ are considered in the following.

Independent of the campaign inter-arrival time distribution and the number of workers, we see in Fig. 5 that the introduction of more complex tasks in the platform by means of a higher mean task length $E[B]$ increases the worker utilization. The same number of workers now require more time to process the same number of tasks. Furthermore, for the same campaign inter-arrival times and mean task lengths, increasing the number of workers decreases the worker utilization, as a higher number of workers has to compete for the same number of tasks. For different shapes α of the campaign inter-arrival

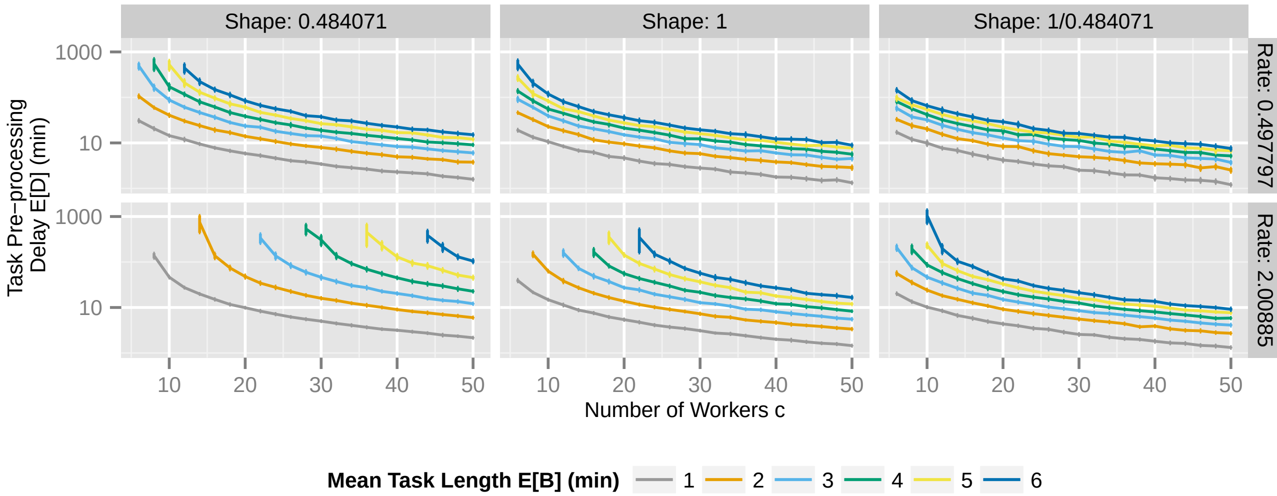


Fig. 6: Task Pre-processing Delay for Different Campaign Inter-arrival Times

times and the same rate β, with all other parameters fixed, we observe a decrease of the shape results in an increase in worker utilization. A decrease of the shape α directly decreases the mean campaign inter-arrival time $E[t_c] = \frac{\alpha}{\beta}$ and increases the rate $\frac{1}{E[t_c]}$ of incoming campaigns which increases the systems utilization. The same argument can be applied to the rate parameter of the campaign inter-arrival time distribution. An increase of the rate β again influences the mean $E[t_c]$ and the rate of the campaign inter-arrival time resulting in an increased worker utilization.

In Fig. 6 we consider the impact of different campain inter-arrival time characteristics on the task pre-processing delay $E[D]$. For a fixed number of workers and campaign inter-arrival distribution a larger mean task duration also increases the mean task pre-processing delay. As more tasks have to enter the queue, tasks which would not have been queued for lower task length now suffer queueing delay. For a fixed task length and campaign inter-arrival distribution, we see that increasing the number of workers results in a decreased task pre-processing delay. The waiting probability decreases due to the higher capacity of the platform, resulting in a lower waiting time per task. Next, we consider the shape of the campaign inter-arrival time for fixed other parameters. The curves show that increasing the shape decreases the mean task pre-processing delay. This is caused by an increasing mean $E[t_c]$ of the inter-arrival times which results in a decrease of the campaign arrival rate. Thus, the platform contains fewer for the same number of workers and fewer tasks have to wait for completion. The effect is more obvious for high rates.

Finally, we consider the effect of an increased rate while keeping all other parameters fixed. An increased campaign inter-arrival rate increases the task pre-processing time, as the number of campaigns arriving at the platform is increased. The increase of the rate β decreases the broad of the campaign inter-arrival times distribution. The greater β the lower is $E[t_c]$ and the higher the campaign inter-arrival rate. More tasks arriving at the platform and have to be completed with the same number of workers.

Based on this observations, we conclude that while both shape and rate influence the metrics worker utilization and mean task pre-processing delay, the rate parameter of the Gamma distribution has an higher influence on the considered metrics. In order to account for the higher influence of the rate on the considered metrics, we fix the shape parameter of the Gamma distribution to the value 0.484071 obtained in Sec. IV for the next section and focus on different rate parameters.

B. Trade-off Considerations for Platform Operators

A crowdsourcing platform operators business success depends on the satisfaction of the main stakeholders, i.e. the employers and workers. As discussed in Sec. III, workers are interested in a high worker utilization ρ because this correlates with their payment. Employers are interested in having their tasks completed as fast as possible, i.e. in a as small as possible task pre-processing delay $E[D]$. The interests of the stakeholders are opposing because lower task pre-processing delays can be achieved by hiring more workers, which in turn results in a lower worker utilization. Thus, the platform operator is forced to consider a trade-off between worker and employer satisfaction, which we consider in this section. The impact of different campaign inter-arrival rates on worker and employer satisfaction for the specific platform can be evaluated by following the colored lines in Fig. 7.

Given a fixed number of workers, decreases in the campaign inter-arrival rate results in lower worker utilization and task pre-processing delay. The effects on the worker utilization and the task waiting time decreases for a larger amount of workers. This means a platform with a larger number of workers is more robust against fluctuations in the rate of incoming campaigns than a system with a small number of workers.

Independent of the considered task duration, we observe that increasing the number of workers, for example by advertising

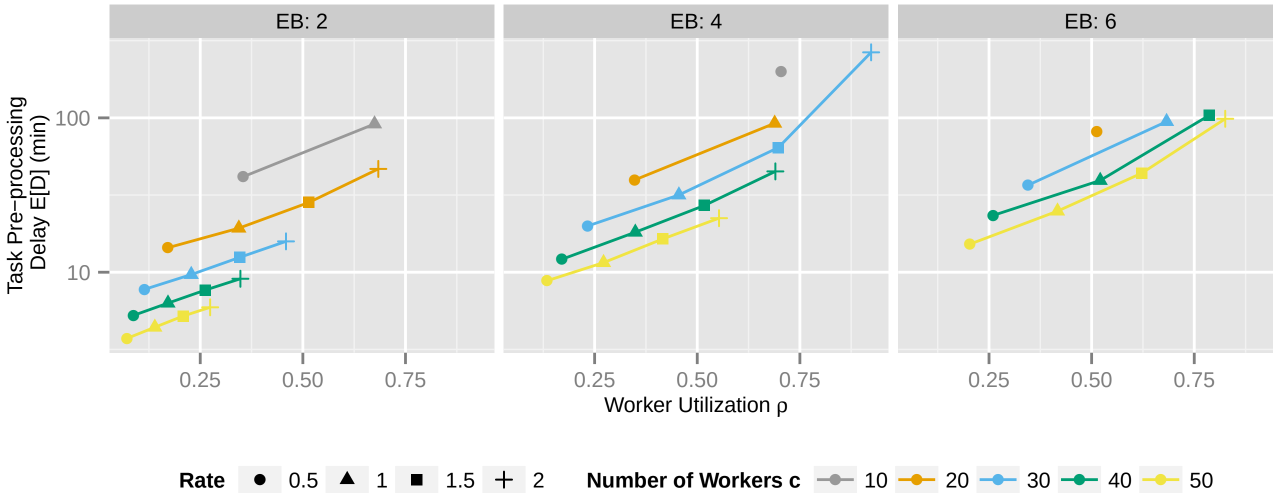


Fig. 7: Trade-off Analysis Between Worker Utilization and Task Pre-processing Delay

the platform, decreases both the task pre-processing delay as well as the worker utilization. However, this decrease is not linear. This means that a small increase of the number of workers reduces the worker utilization, which is generally not desired. However, this small degregation of the worker utilization results in an over proportional reduction of the task pre-processing delay. Thus, it is advisable to sighly overdimension the number of workers to optimize the trade-off between worker utilization and task pre-processing delay.

VI. Conclusion

In this paper we introduced an analytic and a simulative model for crowdsourcing platforms and derived the key performance metrics worker utilization and task pre-processing delay. We analysed a dataset from the commercial crowdsourcing platform Microworkers, obtaining campaign inter-arrival times, campaign sizes and provide fitted distributions. Finally, we performed an analysis of different campaign inter-arrival distributions and worker numbers, in order to present guidelines for platform operators, on how to find a sensible trade-off between the needs of workers and employers.

We have shown that crowdsourcing platforms can be dimensioned so that both workers and employers can be sufficiently satisfied. Our analysis indicates that crowdsourcing platforms are robust regarding different shapes of arrival distribution, but platform operators might need to take appropriate measures if the rate of campaign inter-arrivals changes. One example of such changes might be a major employer switching campaign submission from batch submission at midnight to a streaming submission over the whole day. Our results show that these changes have no impact on the performance metric of the platform and require no action from the platform operator.

The results further indicate that a relatively small number of workers is sufficient to run a crowdsourcing platform. However, this assumes that a worker is willing to work 24/7 and is able to complete all tasks on the platform, which does not hold in the real world. Thus, the given worker counts have to be scaled to realistic work-hours, which is beyond the scope of this paper. Nevertheless, it opens the opportunity to explicitly hire a dedicated workforce to overcome short time worker shortages, e.g., due to large scale internet outages.

Acknowledgement

The authors thank Microworkers for providing insights into the platform. This work is supported by the Deutsche Forschungsgemeinschaft (DFG) under Grants HO4770/2-1 and TR257/38-1. The authors alone are responsible for the content.

References

[1] Leonard Kleinrock. *Queuing Systems*. Wiley, 1975.

[2] Jeff Howe. “The Rise of Crowdsourcing”. In: *Wired Magazine* 14.6 (2006).

[3] Tobias Hoßfeld, Matthias Hirth, and Phuoc Tran-Gia. “Modeling of Crowdsourcing Platforms and Granularity of Work Organization in Future Internet”. In: *International Teletraffic Congress*. 2011.

[4] Nathan Morrow et al. “Independent Evaluation of the Ushahidi Haiti Project”. In: *Development Information Systems International* 8 (2011).

[5] M Jordan Raddick et al. “Galaxy Zoo: Exploring the Motivations of Citizen Science Volunteers”. In: *Astronomy Education Review* 9.1 (2010).

[6] Siamak Faradani, Björn Hartmann, and Panagiotis G Ipeirotis. “What’s the Right Price? Pricing Tasks for Finishing on Time”. In: *Human Computation*. 2011.

[7] Jing Wang, Siamak Faridani, and Panagiotis G Ipeirotis. “Estimating the Completion Time of Crowdsourced Tasks Using Survival Analysis Models”. In: *Crowdsourcing for Search and Data Mining*. 2011.

[8] Michael S Bernstein et al. “Analytic Methods for Optimizing Realtime Crowdsourcing”. In: *arXiv preprint arXiv:1204.2995* (2012).