Towards a Workload Model for Online Social Applications

[ICPE 2013 Work-in-Progress Paper]

Alexandru-Corneliu Olteanu
University Politehnica of Bucharest, Romania
alexandru.olteanu@cs.pub.ro

Alexandru Iosup
Delft University of Technology, the Netherlands
a.iosup@tudelft.nl

Nicolae Ţãpuş
University Politehnica of Bucharest, Romania
nicolae.tapus@cs.pub.ro

ABSTRACT

Popular online social applications hosted by social platforms serve, each, millions of interconnected users. Understanding the workloads of these applications is key in improving the management of their performance and costs. In this work, we analyse traces gathered over a period of thirty-one months for hundreds of Facebook applications. We characterize the popularity of applications, which describes how applications attract users, and the evolution pattern, which describes how the number of users changes over the lifetime of an application. We further model both application popularity and evolution, and validate our model statistically, by fitting five probability distributions to empirical data for each of the model variables. Among the results, we find that most applications reach their maximum number of users within a third of their lifetime, and that the lognormal distribution provides the best fit for the popularity distribution.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modeling**]: Model Development; K.8 [**Computing Milieux**]: Personal Computing—*Games*

Keywords

workload model, workload characterization, online social applications, social gaming, statistical modeling

1. INTRODUCTION

Online social applications hosted by social platforms, such as Facebook, gather an increasing number of interconnected users. With a total of more than 1 billion Facebook users, there are applications that reach tens of millions of daily active users. Moreover, the hosting platform itself is a heterogeneous system, where many third-party developers bring into the platform tens of thousands of applications—for gaming, multimedia streaming, travel, communication, etc.—and their own infrastructure to host them. Given the diversity in both infrastructure and user coverage, understanding usage patterns and modeling the workload is crucial in providing means to optimize the performance and the cost of hosting such applications.

We analyse traces gathered over a period of thirty-one months for 630 of the most popular Facebook applications, with the purpose of modeling their workload. To identify usage patterns and to understand the workload of social applications, we investigate two research questions: *What is the number of users per application?* and *How does the number of active users evolve over time for online social applications?*.

Although tens of recent studies have focused on the workloads of web applications [1] and gaming workloads [8], relatively few studies focus on online social applications. Existing models for online social applications focus on social interactions and use a limited number of applications over a limited period of time [6]. Our effort is conducted on large datasets with the objective of proposing a workload model. We will use this model to generate test workloads and we will further investigate the possibility of insightful predictions based on it. In our future efforts, we will use the model to provide resource offloading and to fine-tune provisioning and allocation for online social applications. Our main contribution is three-fold:

1. We collect data describing the usage of a number of top applications over a multi-year period (Section 2);
2. We propose a model with two components: the popularity distribution, and the evolution of the number of users during the lifetime of an application (Section 3);
3. We use a data-driven empirical method to provide a characterization and validate our model statistically, for each of the two components (Section 4).

2. DATASETS

For this study, we have collected several large-scale datasets. We describe in the following the steps of the data collection process: crawling, parsing, storing, and sanitizing.

Our analysis is based on data collected from Facebook and from various third-party websites, which report data about Facebook applications. Facebook reports daily, through its APIs, application usage data, such as daily active users (DAU) and monthly active users (MAU). In our analysis we use DAU and MAU to investigate both popularity and the evolution of the number of users. However, we are also interested in other data that might provide insights, such as application developer, category and subcategory. We use *app* as an abbreviation for application.

To gather information, we have been *crawling* two websites, namely *appdata.com* and *developeranalytics.com*. We have chosen these websites because they report data about

ICPE'13, April 21–24, 2013, Prague, Czech Republic.

Table 1: Volume of crawled, parsed, and stored information.

Source	Files	Apps	Samples
appdata.com	47,056	16,664	1,864,812
developeranalytics.com	49,177	630	133,594
graph.facebook.com	16,901	16,901	16,901

a large number of Facebook applications, in a paged-tabular form. From such data, it is easy to obtain information about the applications that are top ranking at the time of the crawling, but also to follow several individual applications for a time. When choosing our data sources, we have considered *threats to validity*, as described in several works focused on workloads [2][10]. We have cross-checked the values reported by our two sources. Moreover, we have not used data such as ranks, which could be skewed by the sampling that the websites are using.

In this paper, we present data collected between January 1st, 2010 and July 31st, 2012. From *appdata.com* we started to extract MAU and rank for the top 6,000 apps, but, due to a change in the query parameters in March 2010, our crawlers retrieved only information regarding top 40 apps. From *developeranalytics.com* we extracted DAU, MAU, rank, and developer for apps while they were in top 100 during the thirty-one months. We have also followed a sample of 50 apps, that were in the top in our first day of crawling, during the whole period. Moreover, *developeranalytics.com* displays charts with historical data, going back to the beginning of 2008, which gave an increased coverage for around 50 apps. Overall, we have gathered information about around 150 apps for more than 12 consecutive months.

At the end of the crawling period, we developed Python parsers that we used to *parse* and *store* useful data in an unified SQLite database. We completed the datasets with information regarding developer, category and subcategory by crawling *graph.facebook.com* for all the apps obtained from the other two sources. A quantification of the volume of crawled, parsed, and stored data can be found in Table 1.

To *sanitize* the data, we tried to identify and remove any reporting and reading errors. For example, we consider a reporting error when several applications appear to have no users at a given date, although they have users both before and after that date.

3. MODEL OVERVIEW

We propose a model with two components: the popularity distribution, which describes how users are spread throughout the population of applications, and the evolution pattern, which describes how the number of users varies during the lifetime of an application. We detail our model in the remainder of this section.

3.1 Popularity

To quantify application popularity, we investigate the relation between maximum number of DAU and the rank of the application. We expect this relation to conform with a Pareto-like principle, with top ranking applications gathering many more users even than the ones ranked immediatly below them.

We investigate how the popularity distribution varies over time. Given the increasing number of Facebook users, it is interesting to analyse how new users distribute along popular apps. We also investigate whether the popularity distribution is affected or not by seasonality.

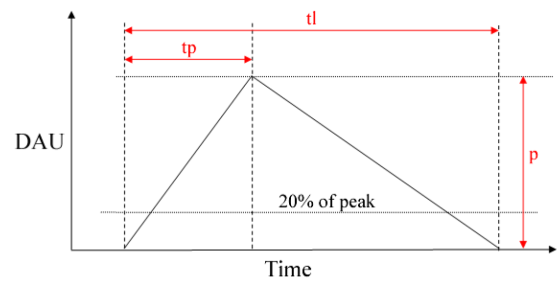

Figure 1: A simplified model to approximate the growth and decay of the number of users for social apps. Labels: p, peak value; t_p, time to peak; t_l, total length of life.

To model popularity, we conduct curve fitting on the empirical distribution for various reference distributions, namely Exponential, Weibull, Pareto, Log-normal, and Gamma. We use maximum likelihood estimation method (MLE) to estimate the distribution parameters, and the Kolmogorov-Smirnov (KS) test to evaluate goodness of fit (GOF). In this paper, we only report the parameters that provide the best fit for each of the reference distributions. For results in terms of the p-values and D-values, we refer to our technical report [7].

3.2 Evolution

In previous studies [9], the number of users over time seems to first grow quickly, then reach a peak, then decrease. Such an evolution can span over months, and even years for popular applications, but people eventually quit using them as new versions or competing applications appear. Thus, we propose in this section a model that captures the growth and decay processes observed for online social applications.

Specifically, we propose a linear model consisting of two line segments. The first segment is on a line with a positive slope, depicting the growth of the number of users, and the second segment is on a line with a negative slope, depicting the decay of the number of users (Figure 1).

We parameterize this model with: peak value (p), growth rate (m), and decay rate (n). Peak value is measured as the maximum DAU throughout the lifespan of the app. Growth rate is given by $m = \frac{p}{t_p}$ and decay rate is given by $n = \frac{-p}{t_l - t_p}$.

We further investigate the underlying relation between growth rate and decay rate by studying the maturity of the app at the peak (r), expressed as the ratio between the time it took the app to reach the peak (t_p) and its total lifetime (t_l). Dividing the equations for growth rate and decay rate, we obtain: $m = (1 - \frac{1}{r})n$

For each of the four parameters described in this section, we plot the cumulative distribution function (CDF), and we conduct curve fitting following the same procedure that we use for fitting the popularity distribution.

4. EXPERIMENTAL RESULTS

In this section we present experimental results related to the two aspects that constitute the focus of our study, app popularity and evolution of the number of app users. For each, we first empirically characterize the aspect using the datasets introduced in Section 2, then validate empirically the models proposed in Section 3. We only present selected results; for more details, we refer to our technical report [7].

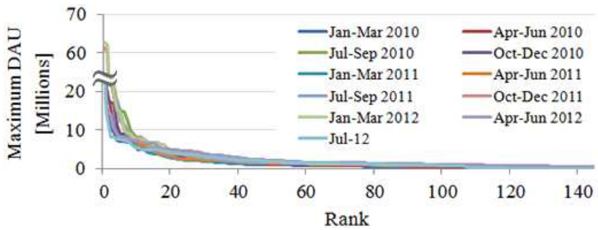

Figure 2: Popularity distribution, depicted as the distribution of maximum DAU over rank. One curve per trimester.

Table 2: Parameters for the reference distributions (Exponential, Weibull, Log-normal, and Gamma) that provide best fit for the popularity distribution. Values for μ_e, λ and θ are expressed in multiples of 10^6. One trimester per year.

Series	Exp(μ_e)	Wbl(k_w, λ)	LogN(μ_l, σ)	Gam(k_g, θ)
Jan-Mar'10	1.76	0.82 1.53	13.69 1.03	0.84 2.09
Jan-Mar'11	2.31	0.84 2.03	14.02 0.90	0.92 2.52
Jan-Mar'12	3.43	0.80 2.84	14.31 0.99	0.81 4.26

4.1 Popularity

We analyse the maximum observed DAU and MAU as metrics of app popularity. To account for seasonal transitions yet still be able to follow the results, we analyse the information we have for these metrics per trimester. We characterize and model, in the remainder of this section, the trimestrial datasets resulting from our 31 months of data.

Characterization: For each trimester in our datasets, we select the maximum DAU for each of the apps, then sort the maxima in descending order to rank the apps. Figure 2 depicts the maximum app DAU over app rank. The shape of the curves for different trimesters are very similar, which indicates that seasonal effects do not influence the relative order of maxima; a more detailed investigation resulting in the same conclusion is included in our technical report [7]. Each trimestrial curve exhibits the same phenomenon: **the best-ranked 5-10% apps** (apps rank 1 through 11-21) **attract many more users than all the remaining apps taken together**. Specifically, the first 20 apps have each over 5,000,000 DAUs; many of them also have over 20,000,000 MAUs (not depicted here). This phenomenon and the observed values indicate that there may be two types of app operators: operators of the first 20 apps, who due to their app sizes could largely own the physical infrastructure on which their apps operate, and the other operators, who could lease infrastructure from a cloud, only when needed.

Modeling through Curve Fitting: We conduct curve fitting for all the eleven datasets. The p-values and D-values obtained through curve fitting, whose exact values are included in our technical report [7], show that **the lognormal distribution provides the best fit with the data we analysed, from the five distributions we have tried**. The parameters obtained for each distribution are summarized in Table 2.

4.2 Evolution

We analyse in this section the evolution of the values observed for app DAU and MAU. Similarly to the previous section, we present characterization and modeling results.

Characterization: To compare apps while accounting for the high variability among them—both lifespan, and evolution of DAU and MAU—, we first normalize both the lifespan and the number of users of each app. In the characterization plots presented in this section, we express for each app the number of users at a moment in time as a percentage of the app's peak number of users, and the moment in time as a percentage of the app's total lifetime. We further present results only for a selection of apps with at least 5,000,000 DAUs or 20,000,000 MAUs, which are thresholds inspired by the results obtained while characterizing app popularity (Section 4.1); these apps account together for more than 80% of the total DAUs and MAUs observed in our datasets.

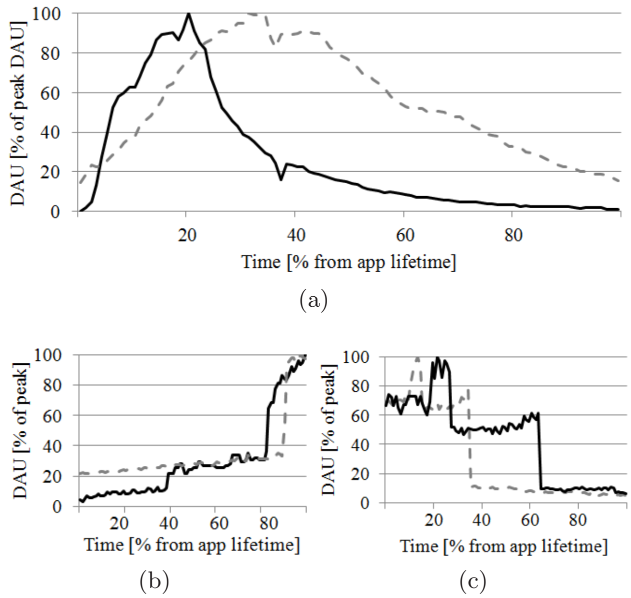

Figure 3: Examples of DAU evolution for several applications: (a) typical single-peaked pattern, (b) suddenly ascending, (c) suddenly descending.

We divide the selected apps into three classes. Class-1 apps (the **common pattern**) grow steadily to a peak, then decay steadily (Figure 3a). The growth and decay can occur with various velocities, but the decay occurs slower than the growth. The growth and decay periods may be interrupted by limited periods (maximum 2 weeks) of opposite effect. Apps in this class are favorable for predictive provisioning of resources for the infrastructure on which they operate. However, the periods of opposite effect may lead to inefficient resource provisioning which, especially when these periods occur close to the peak, may lead to significant unnecessary costs.

Class-2 apps (the **bursty pattern**) exhibit bursts, that is, sharply ascending or descending periods (Figure 3b,c). These apps represent a difficult case for resource provisioning, requiring not only adequate predictions of when the burst will occur, but also finding a provider that can provide the large amount of needed resources.

For Class-3 apps (the **inconclusive pattern**), the evolution cannot be characterized, as for these apps the metrics may be continuously ascending or descending (not depicted here); these apps take a long time to reach their peak or we track only a part of their lifetime in our datasets. These apps are less difficult to provision for than class-1 apps, but require constant managerial attention to avoid missing the peak and thus cause unnecessary costs, and may require a special commercial agreement between the app operator and the resource provider.

Table 3: Percentage of apps in each type of evolution pattern, for two samples of apps: with peak DAU larger than 5,000,000 (column "DAU") and with peak MAU larger than 20,000,000 ("MAU").

	Parameter	
Trend of covered lifespan	DAU	MAU
Single peak	61.0	64.4
Suddenly ascending	7.3	6.7
Suddenly descending	9.9	8.9
Continuously ascending	17.0	13.3
Continuously descending	4.8	6.7
All	100	100

Table 4: Parameters for the reference distributions that provide best fit for each of the parameters of the evolution model (see Section 3.2).

Series	Exp(μ_e)	Wbl(k_w, λ)	LogN(μ_l, σ)	Gam(k_g, θ)
p	4.36×10^6	0.87 4.04×10^6	14.58 1.33	0.83 5.26×10^6
m	50,792	0.66 35,256	9.68 1.62	0.54 93,242
n	8,409	0.68 6,382	7.96 1.66	0.58 14,619
r	19	1.37 21	2.64 0.85	1.76 10

We summarize the percentage of each type of pattern among our selected apps in Table 3. **The common pattern occurs in over 60% of all cases, which leaves a significant percentage of apps exhibiting un-common patterns** and, thus, are difficult cases for provisioning.

In further analysis, with results presented only in our technical report [7], we have found that **over 50% of the apps reach their peak DAU within 15% of their lifetime, and that 85% of the apps reach their peak within their first third of their lifetime**.

Modeling through Linear Fitting: We validate the linear model proposed in Section 3.2 empirically, using our datasets. However, we can only use data coming from apps for which we have a full lifespan coverage, which would greatly reduce the size of our sample. Instead, we find a compromise in using, from our datasets, all the apps having their first and their last DAU reading below 20% of the peak DAU value. With this compromise, we can use 53 apps for modeling, from the 630 apps with at least one DAU reading.

We report from our modeling results the p-values and D-values in our technical report [7], and the distribution parameters in Table 4. **The Weibull and Gamma distributions provide best-fits for both peak value and maturity at the peak. The growth rate can be best approximated with the Log-normal distribution and for the decay rate Weibull, Log-normal, and Gamma are close approximations**.

5. RELATED WORK

Our work is closely related to studies of Internet workloads, including web applications [1, 3, 4], peer-to-peer file sharing [9], and online gaming [8]. In contrast, our study focuses on online social applications, which represent a new class of applications.

Closest to this work, several research efforts focus on online social applications. Nazir et al. [6] study the workload of three Facebook applications, with a focus on social interactions. Kirman et al. [5] study two Facebook games in comparison with a stand-alone game, and find that the social networks show a sharp cut-off, in comparison with the scale-free nature of the social network of the stand-alone game. In contrast to this body of work, ours is conducted on a much larger data set and over a much longer period of time, and the focus of our investigation provides new characterization and modeling insights.

6. CONCLUSION

Understanding the workload of online social applications is an important step in optimizing the performance and reducing the operational costs of these applications, with impact on millions of customers. In this work, we have collected data for hundreds of popular Facebook applications over a period of thirty-one months. Then, we have conducted a data-driven study of the popularity and evolution of online social applications, for which we have provided characterization and modeling results.

We are currently extending our model with a micro-model of user sessions arrival and in-session behavior, which will describe the workload with a finer granularity. We plan to use the extended model to investigate prediction-based resource provisioning and allocation.

Acknowledgments

We thank Boxun Zhang (TUD) for his help. This work is funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Romanian Ministry of Labour through POSDRU/107/1.5/S/76909 and by the STW/NWO Veni grant 11881.

7. REFERENCES

[1] M. Arlitt and C. Williamson. Web server workload characterization: The search for invariants. In *ACM SIGMETRICS*, volume 24, pages 126–137, 1996.

[2] Y. Borghol, S. Mitra, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti. Characterizing and modelling popularity of user-generated videos. *Performance Evaluation*, 68(11):1037–1055, 2011.

[3] L. Cherkasova, Y. Fu, W. Tang, and A. Vahdat. Measuring and characterizing end-to-end internet service performance. *ACM TOIT*, 3(4):347–391, 2003.

[4] A. Iyengar, M. Squillante, and L. Zhang. Analysis and characterization of large-scale web server access patterns and performance. *WWW*, 2(1):85–100, 1999.

[5] B. Kirman, S. Lawson, and C. Linehan. Gaming on and off the social graph: The social structure of facebook games. In *CSE*, pages 627–632, 2009.

[6] A. Nazir, S. Raza, D. Gupta, C. Chuah, and B. Krishnamurthy. Network level footprints of facebook applications. In *IMC*, pages 63–75, 2009.

[7] A. C. Olteanu, A. Iosup, and N. Ţăpuş. Towards a workload model for online social applications: Extended report. Tech.Rep. PDS-2013-003, TU Delft, January 2013. `http://www.pds.ewi.tudelft.nl/fileadmin/pds/reports/2013/PDS-2013-003.pdf`.

[8] M. Suznjevic and M. Matijasevic. Player behavior and traffic characterization for mmorpgs: a survey. *Multimedia Systems*, pages 1–22, 2012.

[9] B. Zhang, A. Iosup, J. Pouwelse, and D. Epema. Identifying, analyzing, and modeling flashcrowds in bittorrent. In *P2P*, pages 240–249, 2011.

[10] B. Zhang, A. Iosup, J. Pouwelse, D. Epema, and H. Sips. Sampling bias in bittorrent measurements. *Euro-Par*, pages 484–496, 2010.