Delft University of Technology
Parallel and Distributed Systems Report Series

Towards a workload model for online social applications: Extended report

Alexandru-Corneliu Olteanu, Alexandru Iosup, and Nicolae Ţăpuş
alexandru.olteanu@cs.pub.ro, A.Iosup@tudelft.nl

Completed Jan 2013.

report number PDS-2013-003

ISSN 1387-2109

Published and produced by:
Parallel and Distributed Systems Group
Department of Software and Computer Technology
Faculty Electrical Engineering, Mathematics, and Computer Science
Delft University of Technology
Mekelweg 4
2628 CD Delft
The Netherlands

Information about Parallel and Distributed Systems Report Series:
reports@pds.ewi.tudelft.nl

Information about Parallel and Distributed Systems Section:
http://pds.ewi.tudelft.nl/

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.

Contents

List of Figures

List of Tables

1 Introduction

Online social applications are applications dedicated to socially interconnected users, developed for various purposes, such as gaming, multimedia streaming, travel, communication, etc. They are usually hosted by social platforms, such as Facebook, MySpace, and Orkut. The hosting environments favor a dynamic user activity. The virality of online social applications has been noted as the property to undergo an exponential growth in the user base [1], due to the quick spread in information and due to the social relations between the 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 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; these constitute the focus of this work.

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?*. These questions help us investigate the possibility of making insightful predictions, based on correlations between the workload and several characteristics, such as the app's category, developer, and milestones. The model that answers these questions can be used to generate test workloads for a system that analyses mechanisms to optimize performance and cost. Such a system can help to investigate resource offloading, and fine-tune allocation and provisioning.

Although tens of recent studies have focused on the workloads of web applications [2] and gaming workloads [3], 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 [1],[4]. 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).

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

2 Datasets

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

Our analysis is based on data collected from Facebook and from various third-party websites. Facebook reports daily, through its APIs, usage data concerning applications dedicated to its users. A number of third-party websites collect this data and report it, over a period of time, in ranks or individually. We use in this paper the names given by Facebook to their usage indicators: 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 extract information, we *crawled* two websites, namely *appdata.com* and *developeranalytics.com*, daily for thirty-one months, between January 1st, 2010 and July 31st, 2012. We chose these websites because they report extensive data about a large number of Facebook applications, in a paged-tabular form, covering 40 and 50 apps on each page, respectively. From such data it is easy to obtain an index of applications, sorted by the number of users. Moreover, *developeranalytics.com* displays charts with historical data, going back to the beginning of 2008, which were used to obtain an increased coverage for 50 apps.

As an alternative to our approach, crawling dirrectly app pages on *facebook.com* is impractical due to the changing login mechanism, and the varying page structure and layout. Another alternative is to crawl dirrectly *graph.facebook.com*, which retrieves a lot of useful information in easy-to-parse JSON format, but requires the Facebook id of the app to be known in advance.

From *appdata.com* we started to extract MAU and rank for the top apps fitting on 150 pages, for a total of 6,000 apps. However, in March 2010, an unforseeen change in the query parameters lead our crawlers to retrieve in each subsequent session the first page for 150 times. So, in the end, we have information about top 6,000 apps for three months and top 40 apps for thirty-one months. From *developeranalytics.com* we extract DAU, MAU, rank and developer for the apps in top 100, over the same thirty-one months. Also, for a sample of 50 apps, that were in the top during our first day of crawling, we gathered information for the whole period.

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, some applications were reported to have an exact same number of users for a few days in a row, or to suddenly have no users for sporadic dates. While this can happen in real-life, due to special events such as failures in the infrastructure, we consider that such features are most probably reporting errors, especially when they affect the whole population of apps. Plotting the evolution of DAU for a few apps shows that the number of users can display a bursty behavior, as it will be shown in Section 4.2.1. We cannot presume that this behavior is caused by reporting errors and, thus, we investigate this situation.

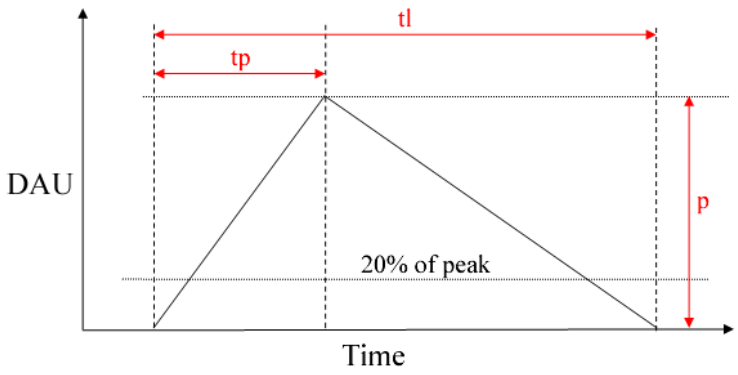

Figure 1: A simplified model to approximate the growth and decay of the number of users for social apps. Labels: p, peak value; t_p, time to peak; t_l, total length of life.

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.

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). The results are reported in terms of the p-values and D-values, as well as the parameters that provide the best fit for each of the reference distributions. The p-value shown is the average of 1000 values, each of which was computed by selecting 30 samples randomly from each dataset.

3.2 Evolution

In previous studies [5], the number of users over time seems to first grow quickly, then reach a peak, then quickly decrease. 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.

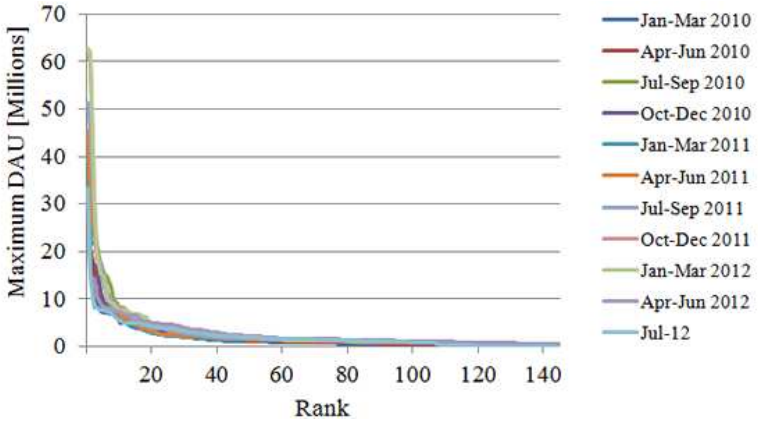

Figure 2: Popularity distribution, depicted as the distribution of maximum DAU over rank. One curve per trimester.

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 present a characterization of online social and empirical results validating the two components of our proposed model.

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 eleven trimestrial datasets resulting from our 31 months of data.

4.1.1 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. The shape of the curves for different trimesters are very similar, which indicates that seasonal effects do not influence the relative order of maxima (Figure 2).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.

4.1.2 Modeling through Curve Fitting

We conduct curve fitting for all the eleven datasets. The p-values in Table 2 and the D-values in Table 3 show that **the Log-normal distribution provides the best fit with the data we analysed, from the five**

Table 2: p-values resulting from KS tests conducted for the popularity distribution over five reference distributions: Exponential, Weibull, Pareto, Log-normal and Gamma. A gray box denotes a p-value above the significance level of 0.05.

series	Exp	Wbl	Prt	LogN	Gam
Jan-Mar 2010	0.073	0.064	0.000	0.238	0.079
Apr-Jun 2010	0.083	0.092	0.000	0.284	0.099
Jul-Sep 2010	0.056	0.063	0.000	0.282	0.066
Oct-Dec 2010	0.065	0.044	0.000	0.202	0.066
Jan-Mar 2011	0.038	0.027	0.000	0.218	0.039
Apr-Jun 2011	0.049	0.027	0.000	0.228	0.048
Jul-Sep 2011	0.050	0.031	0.000	0.223	0.049
Oct-Dec 2011	0.055	0.045	0.000	0.240	0.063
Jan-Mar 2012	0.057	0.051	0.000	0.233	0.065
Apr-Jun 2012	0.132	0.168	0.000	0.320	0.196
Jul 2012	0.205	0.125	0.000	0.044	0.155

Table 3: D-values resulting from KS tests conducted for the popularity distribution over five reference distributions: Exponential, Weibull, Pareto, Log-normal and Gamma.

series	Exp	Wbl	Prt	LogN	Gam
Jan-Mar 2010	0.23	0.22	0.65	0.13	0.21
Apr-Jun 2010	0.23	0.19	0.67	0.13	0.21
Jul-Sep 2010	0.24	0.20	0.66	0.12	0.22
Oct-Dec 2010	0.23	0.20	0.69	0.15	0.23
Jan-Mar 2011	0.27	0.24	0.70	0.16	0.25
Apr-Jun 2011	0.24	0.25	0.66	0.14	0.22
Jul-Sep 2011	0.26	0.24	0.69	0.14	0.23
Oct-Dec 2011	0.27	0.22	0.65	0.15	0.22
Jan-Mar 2012	0.27	0.22	0.68	0.17	0.23
Apr-Jun 2012	0.19	0.16	0.69	0.13	0.18
Jul 2012	0.15	0.20	0.59	0.28	0.20

distributions we have tried, except for the last timeframe, which actually consists of a single month. The parameters obtained for each distribution are summarized in Table 4.

There seems to be no seasonality in the way users are spread among top applications. The increasing trend in the total number of Facebook users lead, at the end of 2011, to an accentuation of the differences among the top apps in terms of maximum DAU (Figure 3).

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 first present characterization, and then modeling results.

4.2.1 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

Table 4: 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

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
Apr-Jun'10	2.03	0.84 1.79	13.86 1.01	0.87 2.34
Jul-Sep'10	2.64	0.83 2.29	14.10 0.99	0.86 3.09
Oct-Dec'10	2.13	0.89 1.96	13.99 0.90	1.00 2.14
Jan-Mar'11	2.31	0.84 2.03	14.02 0.90	0.92 2.52
Apr-Jun'11	2.55	0.82 2.18	14.08 0.93	0.87 2.93
Jul-Sep'11	2.94	0.82 2.50	14.20 0.95	0.85 3.47
Oct-Dec'11	3.22	0.77 2.58	14.20 1.02	0.76 4.24
Jan-Mar'12	3.43	0.80 2.84	14.31 0.99	0.81 4.26
Apr-Jun'12	2.73	1.06 2.82	14.41 0.83	1.36 2.01
Jul'12	2.17	0.63 1.65	13.27 2.67	0.48 4.51

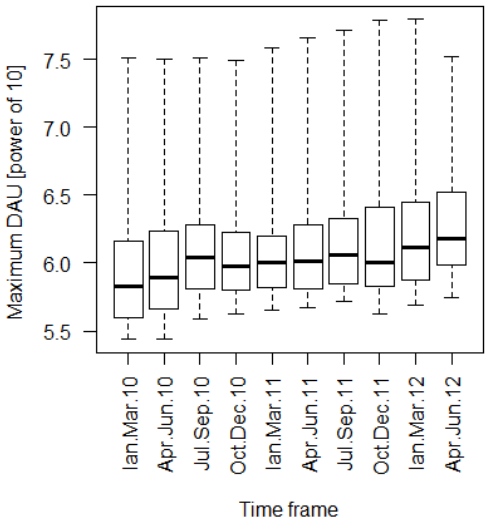

Figure 3: Evolution of maximum DAU for top apps, on a trimestrial basis.

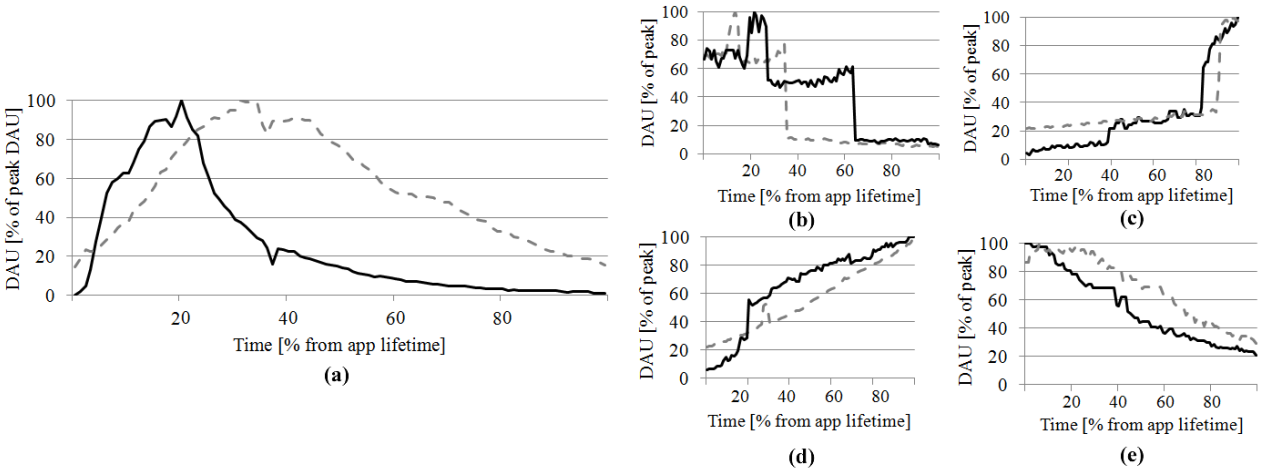

Figure 4: Examples of DAU evolution for several applications: (a) typical single-peaked pattern, (b) suddenly ascending, (c) suddenly descending., (d) continuously ascending, (e) continuously descending.

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.

The evolution of the workload of an app can be obtained by simply plotting either DAU or MAU over time. Given the high degree of variability among apps in terms of both lifespan and workload, a visualization of evolution patterns can be difficult. Thus, it is necessary to normalize both number of users and lifespan before plotting. In our plots, we express the number of users as percent of the peak number of users and the time as percent of the app's lifetime.

In our datasests, we have at least one DAU reading for 630 apps. To make better use of our data, for evolution characterization we select a sample of apps. First, we disregarded apps with very few readings, i.e., less then 100, for which normalization would actually reduce resolution. We further selected only those apps that reach a peak of at least 5,000,000 DAU and we end up with a sample of 46 apps. Figure 4 depicts the DAU evolution for several apps, with normalized values.

For comparison, we also select a sample of apps based on their MAU. We select apps that reach a peak of more than 20,000,000 MAU and with at least 100 readings and we obtain a sample of 55 apps. The two samples (the one based on DAU and the one based on MAU) do not completely match. We find that 40 apps are in both samples, 6 are only in the DAU-based sample and 15 are only in the MAU-based sample.

We divide the selected apps into three classes. Class-1 apps (the **common pattern**) grow steadily to a peak, then decay steadily (Figure 4a). 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 4b,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.

Table 5: 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

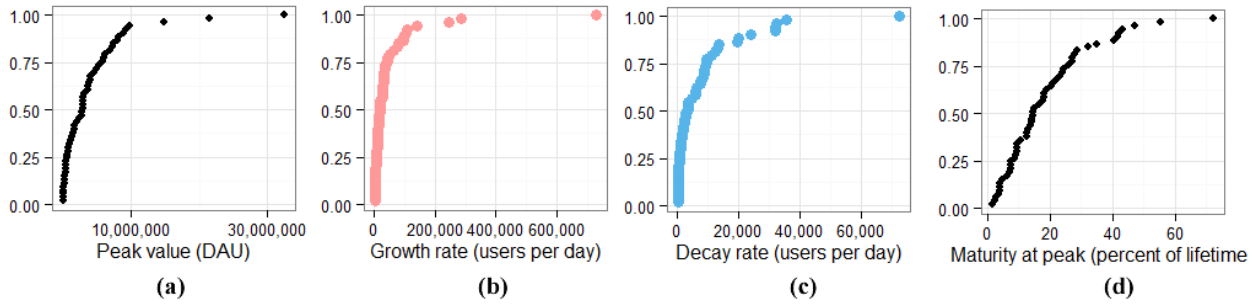

Figure 5: CDFs for the parameters of the evolution model: (a) peak value, (b) growth rate, (c) maturity at the peak and (d) decay rate.

Table 6: Pearson correlation coefficients between three of the evolution model parameters.

decay rate		
0.41	growth rate	
0.57	0.30	peak value

For Class-3 apps (the **inconclusive pattern**), the evolution cannot be characterized, as for these apps the metrics may be continuously ascending or descending; 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.

We summarize the percentage of each type of pattern among our selected apps in Table 5. **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.

4.2.2 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.

Using data from the selected apps, we plot a cumulative distribution function (CDF) for each of the parameters of the evolution model, as depicted in Figure 5. We find that the values of growth rate and decay rate

Table 7: p-values resulting from KS tests conducted for the parameters of the evolution model over five reference distributions: exponential, Weibull, Pareto, lognormal and gamma. A gray box denotes a p-value above the significance level of 0.05.

Series	Exp	Wbl	Prt	LogN	Gam
peak value	0.334	0.426	0.000	0.299	0.406
growth rate	0.063	0.354	0.000	0.359	0.282
decay rate	0.116	0.399	0.000	0.355	0.372
maturity at peak	0.261	0.445	0.000	0.399	0.445

Table 8: D-values resulting from KS tests conducted for the three parameters of the evolution model over five reference distributions: exponential, Weibull, Pareto, lognormal and gamma.

Series	Exp	Wbl	Prt	LogN	Gam
peak value	0.10	0.05	0.58	0.11	0.06
growth rate	0.24	0.10	0.55	0.07	0.15
decay rate	0.22	0.07	0.54	0.08	0.09
maturity at peak	0.12	0.07	0.71	0.07	0.05

Table 9: 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	$4.36{\cdot}10^6$	0.87 $4.04{\cdot}10^6$	14.58 1.33	0.83 $5.26{\cdot}10^6$
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

differ with one order of magnitude (Figure 5b,c). Thus we could presume that usually the peak comes rather early in the life of an app: after a quick growth, follows a slow decay. This fits with some of the apps included in the characterization (Figure 4a), but not with all of them. To better understand this relation, we compute Pearson correlation coefficients for peak value, growth rate, and decay rate, summarized in Table 6, and we find that the values are significant.

A CDF of maturity at the peak is given in Figure 5d: 52% of the apps have a peak within 15% of their lifetime and 85% have a peak within their first third of their lifetime.

To model the distributions for each of the parameters, we conduct curve fitting against five reference distributions (exponential, Weibull, Pareto, log-normal and gamma) and we conduct Kolmogorov-Smirnov (KS) tests to see which is the best fit.

We report from our modeling results the p-values and D-values (in Tables 7 and 8), and the distribution parameters (in Table 9). **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 Discussion

In this section, we examine limitations and threats to validity that our proposed model has in this incipient stage. We also provide some ideas about overcoming them.

One of the model's limitations is its granularity. Currently, our system uses daily active users as a main indicator of user activity. However, given the dynamics of the underlying user connectivity, performance and cost challenges can appear much more rapidly than at one-day intervals. Thus, we need to study finer-grained user activity indicators. Using additional data, we plan to complete our workload model with a component that models user sessions.

The evolution of the number of users, as proposed here, describes only the trend component. We currently use time series analysis to gather more information, which will help us describe the cyclical and irregular components as well. Moreover, the linear approximation we proposed might prove to be too coarse in describing realistically the evolution of the number of users. We will also investigate other shapes for the evolution model.

One of the greatest threats to the validity of our proposed model is that it was validated with a rather small number of apps. Gathering data for the top-most applications accounts for the most users, as described by the popoularity distribution. Observing these apps only while they are in the top means that we do not have data for their whole lifetime. We have a limited number of apps followed throughout our whole data collection period. Even making the assumption that the lifetime above 20% of the peak DAU is a good approximation to the complete lifetime, we can validate with only 53 apps out of 630. In order to overcome this threat, we need to work with more data. We are currently investigating solutions such as computing DAU from MAU, the latter being more abundant in our datasets, as well as searching for new sources of data to crawl.

The bursty behaviour identified in Section 4.2 can have a significant impact on system performance. We rule out some particular features, such as apps not having any users for sporadic dates, as being reporting errors. However, other types of bursts cannot be explained by reporting errors. For example, some bursts compose positive spikes, synchronised for several, but not all the apps. The investigation of irregular components, as described in the previous paragraph, should bring more information on these interesting features.

6 Related Work

Our work is closely related to studies of Internet workloads, including web applications [2, 6, 7, 8], peer-to-peer file sharing [5], and online gaming [3]. The research in [7] and [8] brings a focus on access patterns, based on a spectral analysis of data collected during the 1996 Olympic Games. 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. [1, 4] study the workload of three Facebook applications, with a focus on social interactions. Kirman et al. [9] 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.

We have also found very useful the efforts to model usage related features in online systems, such as failures in large scale distributed systems [10],[11], flashcrowds in bittorrent systems [5] and workloads in online gaming systems [12],[13].

7 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.

Our main findings are:

1. The best-ranked 5-10% apps (apps rank 1 through 10-20) attract many more users than all the remaining apps taken together;

2. The Log-normal distribution provides the best fit with the data we analysed, from the five distributions we have tried. We also show the distribution does not vary significantly during the thirty-one months of data collection;

3. We divide the selected apps into three classes: Class-1 apps (the common pattern) grow steadily to a peak, then decay steadily; Class-2 apps (the bursty pattern) exhibit bursts, that is, sharply ascending or descending periods; Class-3 apps (the inconclusive pattern) are either continuously ascending or continuously descending, as they either take a long time to reach their peak or we track only a part of their lifetime in our datasets;

4. 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;

5. In the sample of 53 apps, for which we have complete lifetime data, 85% of the applications have a peak within the first third of their lifetime.

6. 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.

We are currently extending our model with a micro-model of user sessions arrival and in-session behavior. We are also looking at correlations between model parameters and more in-depth information that has been collected for this study, such as app genre, developer, and time of launch. We plan to use the extended model to investigate prediction-based resource provisioning and allocation, and also to generate test workloads.

Acknowledgments

This work was also supported by the STW/NWO Veni grant @larGe (11881). The authors would also like to thank Boxun Zhang (TUD) for his help.

References

[1] A. Nazir, S. Raza, and C. Chuah, "Unveiling facebook: a measurement study of social network based applications," in *IMC*, 2008, pp. 43–56. 4, 14

[2] M. Arlitt and C. Williamson, "Web server workload characterization: The search for invariants," in *ACM SIGMETRICS*, vol. 24, no. 1. ACM, 1996, pp. 126–137. 4, 14

[3] M. Suznjevic and M. Matijasevic, "Player behavior and traffic characterization for mmorpgs: a survey," *Multimedia Systems*, pp. 1–22, 2012. 4, 14

[4] A. Nazir, S. Raza, D. Gupta, C. Chuah, and B. Krishnamurthy, "Network level footprints of facebook applications," in *IMC*. ACM, 2009, pp. 63–75. 4, 14

[5] B. Zhang, A. Iosup, J. Pouwelse, and D. Epema, "Identifying, analyzing, and modeling flashcrowds in bittorrent," in *P2P*. IEEE, 2011, pp. 240–249. 6, 14

[6] L. Cherkasova, Y. Fu, W. Tang, and A. Vahdat, "Measuring and characterizing end-to-end internet service performance," *ACM TOIT*, vol. 3, no. 4, pp. 347–391, 2003. 14

[7] A. Iyengar, M. Squillante, and L. Zhang, "Analysis and characterization of large-scale web server access patterns and performance," *WWW*, vol. 2, no. 1, pp. 85–100, 1999. 14

[8] A. Iyengar, E. MacNair, M. Squillante, and L. Zhang, "A general methodology for characterizing access patterns and analyzing web server performance," in *MASCOTS*. IEEE, 1998, pp. 167–174. 14

[9] B. Kirman, S. Lawson, and C. Linehan, "Gaming on and off the social graph: The social structure of facebook games," in *CSE*, vol. 4. IEEE, 2009, pp. 627–632. 14

[10] D. Kondo, B. Javadi, A. Iosup, and D. Epema, "The failure trace archive: Enabling comparative analysis of failures in diverse distributed systems," in *CCGrid*. IEEE, 2010, pp. 398–407. 14

[11] N. Yigitbasi, M. Gallet, D. Kondo, A. Iosup, and D. Epema, "Analysis and modeling of time-correlated failures in large-scale distributed systems," in *GRID*. IEEE, 2010, pp. 65–72. 14

[12] Y. Guo and A. Iosup, "The game trace archive," in *NetGames*. IEEE, 2012, pp. 1–6. 14

[13] Y. Guo, S. Shen, O. Visser, and A. Iosup, "An analysis of online match-based games," in *HAVE*. IEEE, 2012, pp. 134–139. 14