Analysing and Modelling Traffic of Systems with Highly Dynamic User Generated Content

Christian Wallenta* chrw@comlab.ox.ac.uk
Mohamed Ahmed† m.ahmed@cs.ucl.ac.uk
Ian Brown** ian.brown@oii.ox.ac.uk
Stephen Hailes† s.hailes@cs.ucl.ac.uk
Felipe Huici†‡ f.huici@cs.ucl.ac.uk
** Oxford Internet Institute, United Kingdom
* University of Oxford Computing Laboratory, United Kingdom
† Department of Computer Science, University College London, United Kingdom
‡ NEC Europe Ltd., Germany

ABSTRACT

As the web continues to evolve, its users have gone from only consuming content to actually producing it, resulting in systems with highly dynamic, user-generated content that cannot be easily modelled with existing tools. In this paper we investigate two such systems, `digg` and `reddit`, derive a general model for them, and show how this model can be used to improve their efficiency as well as that of other systems with similar characteristics.

In order to achieve this, we have collected data on hundreds of thousands of posts and member profiles from both sites. `digg` and `reddit` are social news sites that allow users to post links to other websites as well as to vote for them. We analyse the data to get an understanding of how content is generated and how the popularity of a post evolves over time. We use the results of this analysis coupled with user-location information to derive a general model that describes the user posting behaviour across different time zones.

We further demonstrate how this model can be used to do efficient replication and caching, improving these systems' performance. More importantly though, the periodic trends inherent in the model are not only applicable to these news sites, but also to applications as varied as chatting and online gaming servers, peer-to-peer content distribution and energy-efficient load balancing. We end by showing how the derived model can be used to improve some of these systems.

Keywords

Social Networks, User Generated Content, Simulation Models

1. INTRODUCTION

The manner in which users utilise the web has continued to evolve with technology. While once the web was mostly a read-only resource, nowadays end-users not only consume but also produce content. Some sites go as far as allowing users to vote on posts submitted by other users, letting the vote count determine a post's prominence on the site. As a result, users not only generate content but also act as filters, since their votes directly affect the likelihood that a post will be read by a large audience.

In this work we take a close look at the behaviour of users on two such *content-aggregation* sites, `digg` [1] and `reddit` [2]; we selected these since they are two of the largest in this area. These sites follow the simple structure of enabling members to post links to material published elsewhere on the Internet and then allowing other members to vote the content up or down. The product of this mass collaborative filtering exercise is the creation of a ranking order of the posted links. The ordered content is then presented on web pages that all visitors to the site can view.

Content-aggregation websites differ from sites such as `YouTube` [3] in that the primary focus is not the diffusion of member-generated content, but rather the popularity-based filtering of the content. The result of this service is that members essentially become both the initiators and the editors of the data, a different model from that of traditional web services.

These sites also display a very different set of dynamics compared to those observed in more traditional, user-generated content sites: users not only vote on content (determining its prominence) but also frequently post new content. These factors result in high variability in the relative rank of links as well as a severe limitation in the amount of time that posts stay popular. This is in stark contrast to the findings of recent research on video-orientated sites which shows that the popularity of content evolves and stays popular for months [4].

It is quite clear then that these highly dynamic, user-generated content systems present a distinct set of characteristics, and so we would like to see if it is possible to investigate their behaviour in order to improve their performance.

In order to do so, section 2 and section 3 provide an in-depth analysis of data gathered from `reddit` and `digg`, deriving, in section 4, a geo-temporal model that describes the posting behaviour of users within time zones. In section 5 we demonstrate how this new model can be applied to improve the efficiency of these highly dynamic systems. More importantly, we show that the trends in the model are applicable to a much wider range of applications such as peer-to-peer content distribution and energy-efficient load balancing, and discuss how the model can be used to improve these systems as well. Finally, section 6 discusses related work and section 7 concludes.

2. DATA COLLECTION

We collected data from two of the largest social news sites: digg and reddit. From digg we collected three different datasets by using their API [5] and crawling web pages in order to analyse the following three characteristics of the users:

- **Posting Behaviour** (Set 1 in Table 1) This dataset contains information on new posts coming into the system between May and November 2007. In total, we collected information on 1.5 million posts including their submission time, their authors and the number of votes they received.
- **Voting Behaviour** (Set 2 in Table 1) This dataset contains information on user votes for 87, 000 posts submitted between November 21^{st}, 2007 and December 1^{st}, 2007. We recorded the time a vote was placed and by whom, resulting in information on 1.6 million votes.
- **User Location** (Set 3 in Table 1) This dataset was derived from the user profiles of the authors of the 1.5 million posts in the first dataset. We collected location information on 144, 000 distinct users who stated a location in their profile (60% of all authors).

In contrast to digg, reddit does not provide an API with which to collect data. Since the site does not offer useful information on when posts are submitted, we were forced to retrieve data in real-time in order to timestamp them accurately. Moreover, reddit users do not provide any location information in their profiles, so we have less data for reddit than for digg. Nonetheless, we crawled *Reddit*'s web pages in order to collect two datasets, as shown below.

- the **Posting Behaviour** (Set 4 in Table 1) This dataset contains information on new posts coming into the system between November 2007 and February 2008. In total, we collected information on 183, 496 posts, including their authors and the number of votes they received. This dataset does not include accurate timestamps and was only used to analyse the user contribution and the distribution of votes.
- the **Voting Behaviour** (Set 5 in Table 1) This dataset contains information on new posts coming into the system as well as information on the votes they received. We periodically parsed the new section of reddit in order to monitor new posts coming into the system and revisited them periodically in order to record their current vote count; this set contains information on 13, 368 posts.

	Period	**Description**	**Quantity**
1	21/05 - 21/11 07	digg new posts	1,499,962
2	21/11 - 01/12 07	digg votes for new posts	1,619,696
3	21/05 - 21/11 07	digg user profiles	241,462
4	Nov 07 - Feb 08	reddit new posts	183,496
5	23/11 - 30/11 07	reddit new posts/votes	13,368

Table 1: Datasets collected for digg and reddit. New posts means information on all new posts coming into the system and votes means information on the votes for all these posts, such as user name and voting time.

3. ANALYSIS OF DIGG AND REDDIT

In this section we present a two-part analysis of the datasets discussed thus far. The first part of this section is interested in the properties of the data generation process and, more specifically, we analyse: (i) the manner in which new data enters the system; (ii) the distribution of users responsible for this content; (iii) the type of content; and (iv) what content becomes popular.

The information gathered here is used to help unravel the underlying dynamics that drive precisely how content comes into these systems. More specifically, this information can be used to build generalised models of similar systems for experimentation and to compare our results with the dynamics of other systems based around user content.

The second part of this section focuses on the popularity of the data submitted, highlighting: (i) the evolution of the popularity of content over time, (ii) the statistical features of content that becomes popular and (iii) the time that participants remain actively engaged with the content. This analysis will be used primarily to identify the behaviour of content once it enters the systems (i.e. what users do with new content).

3.1 Content Generation

3.1.1 *Incoming Data Frequency*

We first looked at the overall volume of data submitted to digg during the 26 week investigation period. Plotting the weekly average number of submitted posts reveals a clear linear increase between May and

November 2007 (Figure 1). We also observe an almost 30% increase in the average number of new posts per week, from $50,000$ to $65,000$.

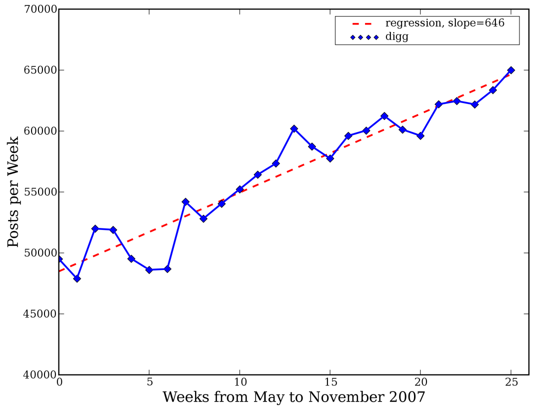


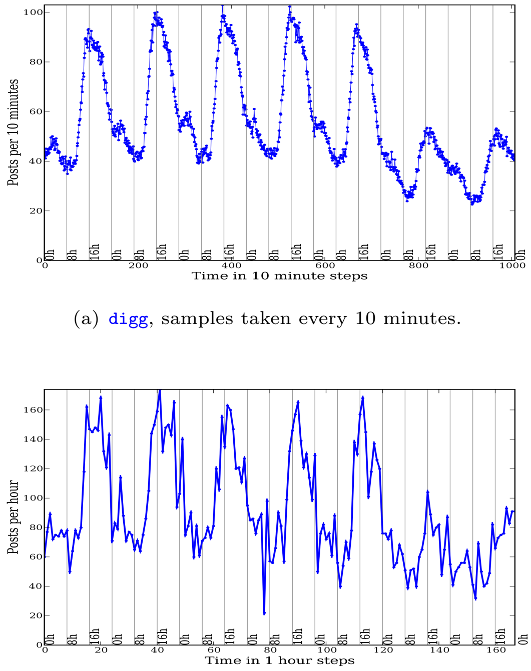
Figure 1: Number of posts submitted per week from May to November 2007 for digg. The system shows a clear upwards trend from 50,000 in May to about 65,000 in November.

We then looked at the average change in the volume of data submitted over one week. We used 10 minute sample intervals for digg and 60 minute intervals for reddit, since the latter receives about an order of magnitude fewer posts. Looking at Figures 2(a) and 2(b), we can see a clear periodic trend throughout the week in both systems. The average number of submissions varies throughout the day, peaking at approximately 16:00 GMT for both sites. There is a second peak in each weekday at approximately 02:00-06:00hrs GMT. As we shall see in section 4, this peak may be correlated with the culmination of the night-time behaviour of the GMT -5 to GMT -8 regions and the day-time behaviour of the smaller (in terms of active participant numbers) GMT +8 region.

As mentioned, the prominent daily peak in the upload of new content occurs in the hours around 16:00 GMT corresponding to the working hours in the East and West Coast of the United States; this is again corroborated by our analysis of the geographic location of users in section 4. Both sites accumulate on average twice the volume of new content in weekdays as opposed to weekends. Hence, we believe that the use of these sites is part of the normal work-day pattern.

3.1.2 User Contribution

Having examined the global behaviour of the systems, we now turn our attention to the behaviour of the sites' participants, focusing on how much each user contributes to the system. Many systems with user-generated content tend to follow the Pareto Principle, also called the 80-20 rule [6]. In the case of author contribution, this would mean that 20% of all authors contribute 80% of all posts. Were the system to follow this principle, it could be modelled with a Pareto distribution. To see if this is the case for our digg and reddit datasets, we plot the fraction of all authors against the fraction of all posts and show the results in Figure 3.



(a) digg, samples taken every 10 minutes.

(b) reddit, samples taken every 60 minutes.

Figure 2: Number of news submitted to digg and reddit from Monday to Sunday.

The 80-20 rule describes the digg (dashed line) behaviour almost perfectly: 80% of the posts were submitted by 21.1% of the users. In absolute numbers, the $1,200,000$ posts were submitted by $51,000$ users. reddit (solid line) is even more extreme, with 80% of all posts submitted by only 10.7% of the users, equal to $2,500$ users contributing $147,000$ posts. For both systems, $1.2-3.4\%$ of the users generate 50% of the content and under 0.6% of the users generate 25%: content generation is extremely dependent on just a few users. Since the data closely matches the 80-20 rule, we now aim to describe both systems with a Pareto distribution.

A common method of fitting this type of data is to look at a loglog plot and then attempt to fit it with either a Zipfian, Pareto or power law distribution. Because all three distributions are interchangeable, we choose a Pareto distribution. This distribution states that the probability of some author having more than x posts is inversely dependent on the power of x: $Pr(X \geq x) \sim x^{-k}$ [7].

Table 2 and Figure 4 show the results of fitting the

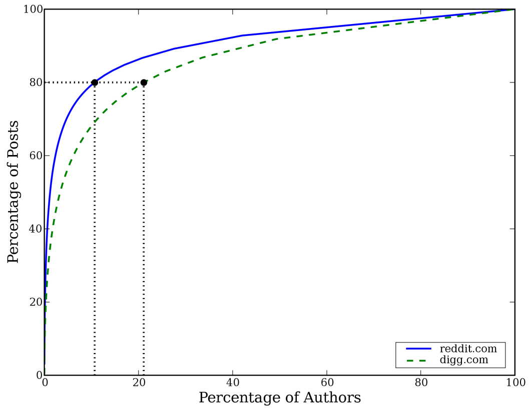


Figure 3: Author contribution for digg and reddit.

Pareto distribution using the Maximum Likelihood Estimator described in [8]. As can be seen, the distribution holds well for the head of the reddit dataset and the tail of the digg set, with an $x_{\min}$ of 2 and 247, respectively[1]. The Kolmogorov-Smirnov statistic[2] is used to test the goodness of the Pareto fit yielding a KS statistic (D) of 0.0101 and 0.0166 respectively for the sites. Since both values are close to 0, we may interpret these results to indicate a good fit in the captured areas.

What do these results tell us? For reddit, the line fits the head part of the graph better than the tail, meaning that the top authors produce less data than the fitted model would predict. For digg the model accounts for the users that contribute many posts better than for the ones that submitted only a few. We believe this is partly due to the fact that we have a larger dataset for digg than for reddit. We suspect that many users register but then only submit a single post; indeed, 50% of the digg users in this dataset submitted only one post. We would expect this part of the distribution to become heavier in a bigger data set.

These models allow us to describe similar systems by simply tuning the parameter of the distribution. Moreover, the parameters for digg and reddit can be compared to the measurements of other systems with user-generated content.

	Pareto k	$x_{\min}$	D (KS-Test)
reddit	1.84	2	0.0101
digg	1.88	247	0.0166

Table 2: Results of fitting a Pareto distribution to the data using a Maximum Likelihood Estimator. k is the Pareto parameter, $x_{\min}$ is the minimum x for which the Pareto distribution holds and D is the result of the Kolmogorov-Smirnov Test.

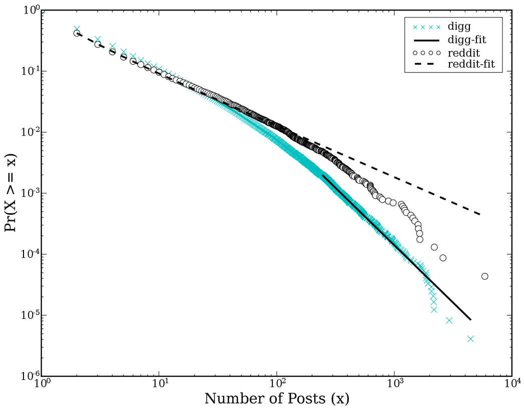


Figure 4: Loglog plot of the number of posts versus the probability that a user contributed more than x posts. This plot is used to fit Pareto distributions to both measurements. The best-fitted models are shown as straight lines.

3.1.3 Link Analysis

Having examined the behaviour of active participants in both systems, we next look at the content of the posts. Since a semantic content analysis is beyond the scope of this paper, we look at the links in the posts instead in order to see what domains are submitted and, more importantly, which posts become popular. The aim here is to understand whether the filtering accomplishes its aim and how. Having noticed that certain domains and authors tend to dominate the sites, we now study whether this is also reflected in the popularity of the domains, i.e. does the filtering work?

Table 3 lists the domains submitted most to digg and reddit. For reddit, 9 of the top 10 are popular news pages and papers. digg has four news pages and three video platforms in the top 10. The popular video platform YouTube is in the top 2 for both sites. The reader may also note the web portals asssociatedcontent[9], squidoo[10] and helium[11] in the digg table: all three sites allow users to publish content and, more interestingly, offer them a share of the ad revenue generated with user content. Hence, it is worthwhile for a content creator to publish links on a social news site such as reddit or digg since a popular post in any of these systems significantly increases the traffic and possibly the ad revenue.

Though from Table 3 we can see the volume of content linked to certain domains, we also looked at the popularity of domains in order to get an idea of what both communities prefer to see. The number of votes

[1] $x_{\min}$ is the minimum x for which the distribution holds.

[2] $D = \max_{x \geq x_{\min}} |S(x) - P(x)|$, $S(x)$ is the CDF of the data and $P(x)$ the CDF for the model.

a post obtains greatly affects its placement on the site which, in turn, determines how many users read the post and follow the link. Hence, by voting posts up and down the ranking, the registered users act as a filter for all the users that only read posts.

Table 4 shows the domains with the highest percentage of popular posts among their submissions. This is the number of popular posts divided by the number of total posts for any domain with more than 100 submissions. We used *Digg*'s API to retrieve information about a post's promotion. For reddit, we considered any posts with more than 300 votes as popular. 1.62% of all posts submitted to digg and 2.78% of those submitted to reddit became popular.

We notice that none of the domains from Table 3 appear in Table 4. In fact, most of the domains in Table 4 have around $2 - 4\%$ of their posts becoming popular. More interestingly, neither asssociatedcontent, squidoo nor helium have a single popular post, despite being among the most submitted domains, meaning that the users filter this kind of content.

To some extent, Table 4 is also of interest for predicting the popularity of posts. The domain name in a post can be an additional parameter that helps to predict whether a post becomes popular or not; looking at the history of a domain with enough past submissions can be an additional indicator.

	digg **All Posts**		reddit **All Posts**	
	Domain	Posts	Domain	Posts
1	youtube.com	46,646	nytimes.com	4,835
2	associatedcontent.com	11,348	youtube.com	4,609
3	nytimes.com	10,825	news.bbc.co.uk	3,568
4	news.bbc.co.uk	9,484	news.yahoo.com	3,094
5	squidoo.com	9,307	reddit.com	2,460
6	news.yahoo.com	8,425	reuters.com	1,706
7	reuters.com	7,344	huffingtonpost.com	1,443
8	metacafe.com	61,64	rawstory.com	1,403
9	dailymotion.com	5,814	cnn.com	1,397
10	helium.com	5,552	washingtonpost.com	1,388

Table 3: **Domains submitted most to diggand reddit.**

	digg **Popular Posts**		reddit **Popular Posts**	
	Domain	% Pop	Domain	% Pop
1	torrentfreak.com	54.1	newsandpolicy.com	22.7
2	5min.com	37.7	xkcd.com	20.8
3	doubleviking.com	36.1	seattlepi.nwsource.com	14.4
4	last100.com	33.1	rollingstone.com	12.1
5	colourlovers.com	32.8	pizdaus.com	12.1
6	jalopnik.com	31.4	afterdowningstreet.org	11.5
7	zenhabits.net	31.1	presscue.com	9.2
8	howtoforge.com	26.0	dailykos.com	9.1
9	cracked.com	25.7	self.politics	8.7
10	phoronix.com	25.4	flickr.com	8.1

Table 4: **Most popular domains with regards to the number of submissions for digg and reddit.**

3.2 Popularity Analysis

The second part of our analysis concentrates on the popularity of posts. Unfortunately, neither digg nor reddit publish information about how often a post has been clicked on, so we cannot measure popularity directly. We believe, however, that the vote count can be used as a good enough popularity metric since in both systems a post's vote count directly determines its prominence on the site.

We will first look at the vote distribution among all posts. Then we will analyse the popularity evolution of posts over time to see how dynamic both systems are. Finally, we look at the lifetime of a post in the system, which we expect to be short since the content is mostly news stories.

3.2.1 Vote Distribution

The vote distribution reveals what percentage of votes goes to what percentage of posts. We would expect this distribution to follow to 80-20 rule as well, which means that most of the votes go to a small number of posts. In order to see if this is the case, we plotted the fraction of all posts against the fraction of all votes; the results for both systems are shown in Figure 5. For reddit 80% of the votes are for 16.3% of the posts. In absolute numbers, around 7.2 million votes went to $30,000$ posts. digg is more extreme, with 80% of the votes going to 2.3% of the posts (26 million votes go to only $35,000$ posts). Moreover, the great majority of the posts received only very few votes: 76% of all digg and 40% of all reddit posts received fewer than 5 votes. Hence, if we are right in the assumption that posts with few votes are not seen by many people due to their placement on the sites, this means that the great majority of the content submitted to both systems is not used at all.

This is an important result because if a distribution mechanism can identify the popular data early enough, it can pro-actively distribute and replicate them in order to increase system performance. This would clearly not be limited to text-based postings, and greater benefit is likely to be obtained for systems hosting larger files such as pictures or videos.

3.2.2 Popularity Evolution

The previous section showed how data popularity can be modelled statically, with most of the votes going to the top posts while the majority of posts receive only a few votes. In fact, this static model is used quite often when data requests are simulated. While this makes sense in systems where the data does not change much over time, it is not accurate in highly dynamic systems for which new data becomes popular and pushes old data away at a high rate. If we are able to understand such a dynamic process, we could derive a general, more accurate popularity model that could be used to evaluate other systems. Further, the information gathered here could be used to identify the key parameters to use for predicting the popularity of content.

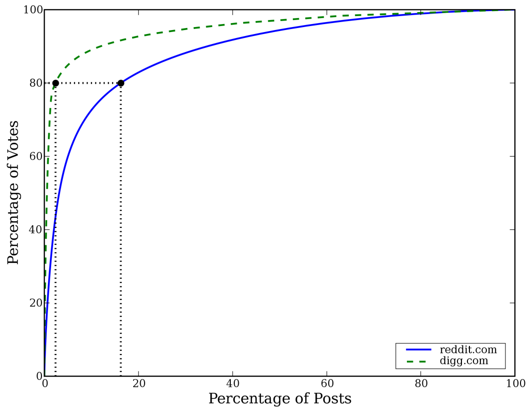


Figure 5: Vote distribution for reddit and digg. reddit closely follows the 80-20 rule while for digg even fewer posts receive most of the votes.

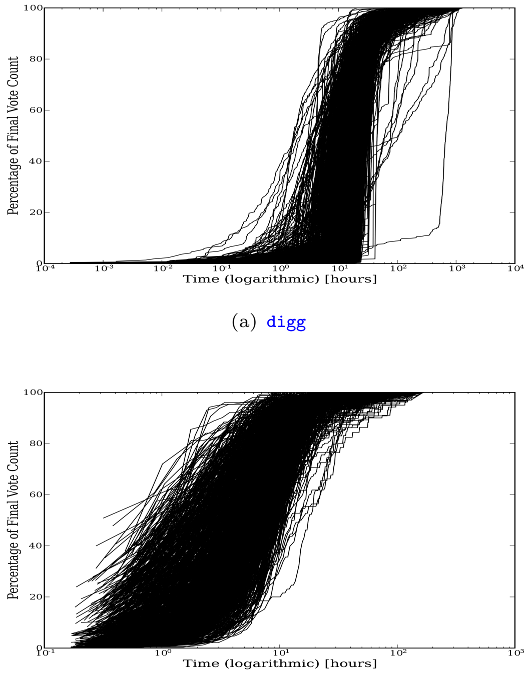
In order to analyse these dynamics, we monitored each post from the time it was submitted to the time it ceased to receive votes. Figure 6(a) shows the popularity evolution of digg posts with more than 200 votes, while Figure 6(b) shows the reddit case for posts with more than 50 votes (we selected 50 instead of 200 because the dataset was much smaller and this gave us more posts). Note that the time is logarithmic and the vote count is normalised for better readability.

Both graphs show that posts receive the majority of votes in the first few hours after being submitted: most of the posts receive around 80% of their votes in the first 10 to 20 hours. Some of the posts still receive votes for a longer time period, but this is only a small fraction of the total votes they receive. One explanation is that some very popular posts will appear in the daily, weekly or monthly top posts. Hence, they are displayed over a longer time period and receive votes over a longer time period.

We also examined the time between a post having more than 10% and less than 90% of its final vote count and found this time to be very short. For 50% of the posts, this period lasts less than one hour in digg and less than 11 hours in reddit; for 90% of the posts, it is 7 and 15 hours, respectively. This shows how dynamic the evolution of post popularity is in both systems.

We already mentioned that replication and caching of top posts can increase system performance. In highly dynamic systems, this has to be done fairly early, so it is important to identify popular data as soon as possible. Further, replication could be stopped at a certain cut-off point, since posts quickly stop receiving votes because they are supplanted by new content.

3.2.3 Data Lifetime



(a) digg

(b) reddit

Figure 6: The evolution of the popularity of posts over time. Time is plotted using a log scale and the vote count has been normalised.

We now consider the lifetime of a post, which we define as the time between its submission and the time it received 95% of its votes (we chose this percentage in order to reduce the effect of long tails). We consider only posts with more than 50 votes for our analysis in order to avoid a distribution skewed towards very short lifetimes; this filtering criteria results in 1,105 posts for reddit and 2,458 posts for digg.

Based on the analysis of popularity evolution already presented, we expect the majority of the posts to have a very short lifetime. Some of them, those that are very popular, will receive votes for a longer period since they might appear in the daily, weekly or monthly top rankings. Figure 7 plots the time in hours (log scale) against the percentage of posts that are still "active" according to our definition. As expected, the great majority of the posts have a very short lifetime. In fact, after around 22 hours for reddit and 78 hours for digg, 80% of the posts no longer show much activity.

In summary, we found that the rate of new posts coming into the system shows clear daily and weekly patterns peaking at approximately 16:00 GMT which suggest that the use of these sites is part of the normal work-day pattern. User contribution can be described well with a Pareto distribution, since both systems fol-

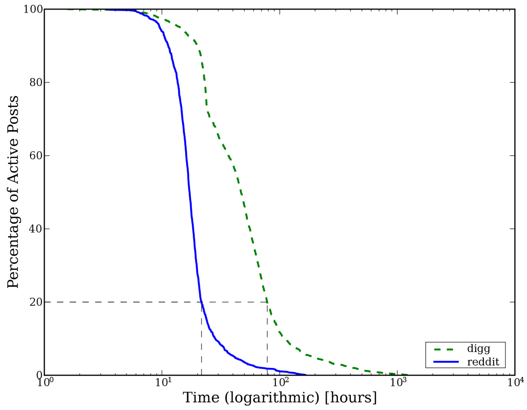


Figure 7: Time in hours (log scale) versus the percentage of active posts. Only posts with more than 50 votes are considered. The tail for reddit is shorter than that for digg since data was collected for longer for the latter.

low the 80-20 rule meaning that most of the content is produced by only a few users.

The analysis of the popularity of posts shows how highly dynamic these kind of news systems are compared to, for example, video platforms, in which content is of interest for a much longer period. Data with very short lifetimes and fast popularity evolution makes it necessary to identify popular data quickly if we wish to utilise caching and replication. If this happens too late, data may have already been supplanted by new content, rendering the caching mechanism ineffective. From the data life-time we can also see that replication and caching is only useful up to a certain time, after which data attracts fewer and fewer votes.

4. DATA GENERATION MODEL

The aim of this section is to get an understanding of the underlying processes that drive the generation and evolution of user generated content. In order to do so, we develop a linearly-weighted model that captures the geo-temporal nature of the data obtained from digg and reddit.

Since the general behavior of the data sets for both sites is similar (see Figure 2, Figure 3 and Figure 5), we choose to use the larger 6-month data set from digg to build our model and the smaller reddit data set to test its applicability and generality.

From the peaks and troughs of the time series given in Figure 2 we can deduce that the volume of data entering the system at a given sample interval is the sum of the relative contributions of the site users located around the world (we assume that the users at these locations behave independently of each other, i.e. the amount of content posted from one location does not affect how much other locations can post). The total volume of content received at any given sample period of time is the sum of weighted contributions from all the users locations for each sample interval in the period. Further, if we assume that the users of a site share roughly the same content uploading habits (e.g. everybody likes to post new content in the morning) regardless of their location, then the variance in the time series can be explained as the aggregate result of users behaving in the same way, but across different time zones and uploading different quantities of new posts.

To capture this behaviour, we develop a generalisable linear-weighted model that takes into account three parameters:

- v: The volume of content posted on the site at any given sample interval. Because the unprocessed time series (Figure 2(a)) is noisy, we filter it with a Fourier transformation to reveal its dominant frequencies and use only the combination of the k dominant frequencies to specify a smooth target (subsection 4.1).
- w: The relative contributions of the users in each of the 24 possible time zones (subsection 4.2).
- p: The expected behaviour of a user throughout a 24hr period. Knowing this distribution enables us to estimate the volume of content we ought to expect from any given time zone (subsection 4.3).

4.1 Identifying the k Dominant Frequencies

Plotting the graphs of the average number of weekly submissions (Figure 2(a)), we see that both of the datasets exhibit the same periodic behaviour throughout the week. Weekdays look almost identical with peaks around 16hrs GMT while weekends exhibit the same periodic trend but follow a different set of dominant frequencies with a smaller amplitude, indicating a fall in the activity of users over the weekend.

To understand the time series we apply a Discrete Fourier Transform (DFT) to the cumulative weekly averages for the digg data set presented in Figure 2(a); this data set is composed of 1,008 data points representing 10 minute slices across the 7 days of the week, averaged across 6 months.

A quick glance at the data reveals two trends with a significant difference in amplitude, representing weekdays and weekends. As a result, we separate the data into two clusters of 720 data points for the five weekdays and 288 data points for the weekend. We work with each cluster individually before combining them to reconstruct the original 7-day weekly trend.

From this result, the simplest information to interpret are: (i) the dominant frequencies for weekdays

Weekdays			Weekends		
index	magnitude	hrs	index	magnitude	hrs
5	8,931	24	2	1,503	24
10	3,489	12	4	702	12
1	1,260	120	1	320	48
20	1,245	6	6	114	8
15	819	8	3	99	16

Table 5: The 5 most dominant frequencies for the Discrete Fourier Transform of the incoming data graph of digg.

and weekends have indices of 5 and 2 respectively (Figure 8(b) and Table 5), representing the periodic 24hr behaviour of users, and interestingly (ii) the 1 frequency indices for both weekdays and weekends have peaks at midweek (Wednesday) for weekdays and mid-Saturday for the weekend (Figure 8(c)), accurately capturing the variance in the volume of data submitted across the week. Figures 8(d) and 8(e) result from applying the Inverse Discrete Fourier Transform (IDFT) to the filtered spectrum and Figure 8(f) gives the result of fitting the model constructed by summing only the 4 and 3 dominant frequencies for weekdays and weekends and re-merging the two models.

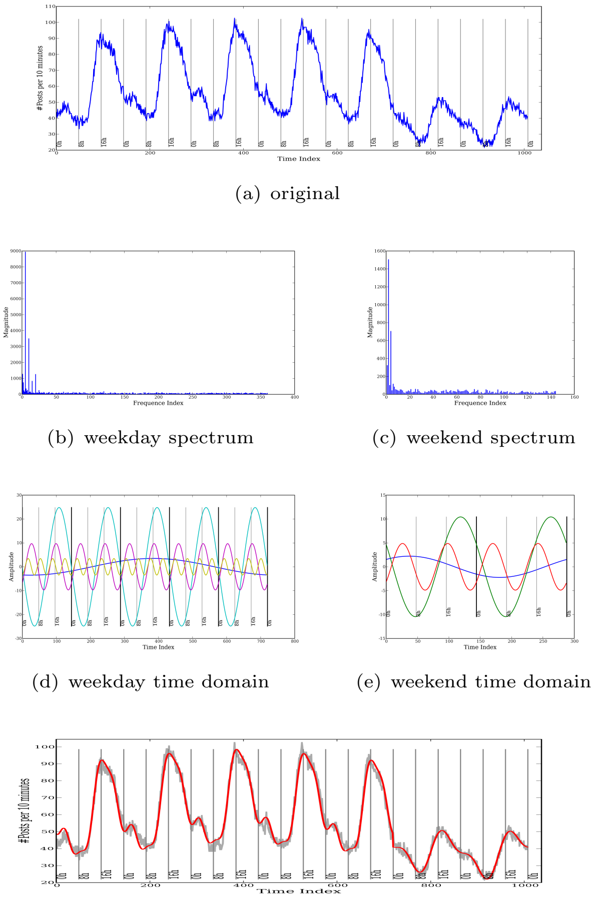
In specifying the number of frequencies (k) to use for modelling the times series, we must choose between accuracy and complexity. Since this model is intended to capture a more generic behaviour than the original dataset, we must take care about both over-fitting and under-fitting the data.

We have two methods available to us in evaluating this parameter. First we may simply look at the spectrum graph generated by the DFT, pick the frequencies associated with dominant indices, and using those directly, since adding the indices with smaller magnitudes simply capture the peculiarities of the data set. The second method is to design a cost function, penalising the gain in accuracy for each extra frequency by the total number of frequencies already incorporated. For our model we settled on $k = 4$ for weekdays and $k = 3$ for weekends.

As can be seen in Figure 8(f), the model using $k = 4, 3$ fits the data well and has a mean error of 2.67 posts per time interval of 10 minutes; in absolute terms, we achieve an error rate under 4% which is concentrated around the outliers per interval. In effect this informs us that the fit, though not perfect, is good and the errors that occur are concentrated around the troughs of the curve where the relative amplitude is small and hence have little influence on the overall trend.

4.2 Identifying the Weighted Time Zone Distribution (w)

Though the Fourier analysis affords us useful information on the dominant frequencies in the time series, it does not give us direct information regarding the time zones from which new content is submitted.



(a) original

(b) weekday spectrum

(c) weekend spectrum

(d) weekday time domain

(e) weekend time domain

(f) 7 day model

Figure 8: Using Fourier Analysis to model a digg week. The top graph shows the original incoming data frequency graph. The next one shows the frequency spectrum of this graph. For the next graph we cut all but the strongest frequencies which are plotted as sine waves. The last graph shows the model which was obtained by applying the inverse Fourier transformation using the most dominant frequencies and the DC component.

To construct this geographical information, we collected the user profiles of all 241,464 unique user names responsible for generating the content of our 6 month data set and noted the stated locations in user profiles. Unfortunately, the information given varied, since some users specified countries while others cities, etc. To minimise ambiguities, we cropped the data set and used only 60% of the user profiles where users stated an identifiable country as their location and used this information to create the geo-tagged data set depicted in Table 6.

To test whether the geo-tagged data remained representative of the original data, the Fourier analysis was reapplied and the resulting frequency domain gave the same dominant frequencies only with smaller amplitudes. This indicated that at least the likelihood of users revealing their country of location is uniformly distributed, and that the sample is representative of the original data set. As shown in Table 6, the geo-tagged user base is scattered around the word, favouring English speaking nations, and, in particular, the US.

Country	Time Zone (GMT)	Weight
United States	+10, -4 to -11	0.588
United Kingdom	0	0.075
Canada	-3 to -8	0.050
India	+5:30	0.036
Australia	+8 to +10:30	0.027
Germany	+1	0.013
France	+1	0.011
Italy	+1	0.010
China	+8	0.009
Brazil	-2 to -5	0.008

Table 6: The relative contribution of the 10 dominant countries.

The *Weights* column in Table 6 gives the relative contribution of the 10 prominent countries represented in the geo-tagged data set, and is used as input to the model to account for the relative contribution of each of the time zones to the volume of content we collect per sample interval. However, before it can be applied it must be transformed to account for countries that span multiple time zones.

To do so, we augment the information we have about larger countries with the population densities across their time zones. This way, we can create a weights vector ($w = [w_{-11}, \ldots, w_{12}]$) to contain the relative contribution of each of the 24 time zones in the geo-tagged data set. For example, as can be seen from Table 7, 47.3% of the US population lives in the East Coast (the -5hrs GMT time zone); with no further information, we can assume that 47.3% of US content is published from -5hrs GMT.

Though these assumptions are difficult to justify formally without more data, anecdotal evidence supporting our assumptions may be found in [12], where geo-located graphs of a recent 24hr snapshot of digg traffic redirected is given, and by inspection, the results mirror our assumptions.

Time Zone	% Population
-8	16.1 %
-7	6.1 %
-6	28.4 %
-5	47.3 %

Table 7: Population distribution for the four main time zones of the United States. Note that the figures do not add up to 100% because the other remaining time zones are not shown.

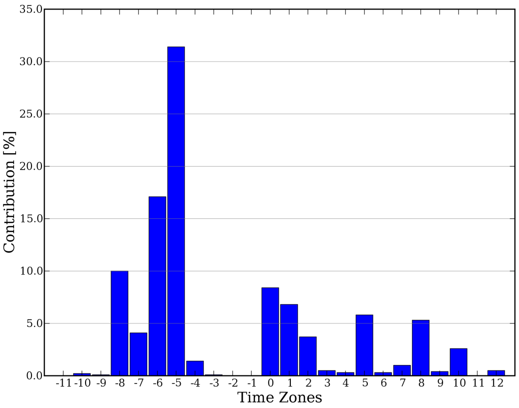


Figure 9: The histogram shows the time zone distribution for the digg users in our data set.

4.3 Identifying the generalised behaviour distribution (p)

With the time zone information, the total volume of content received in a 24hr period can simply be viewed as a linear shifted combination of the content received from each of the 24 time zones. More formally, given the set of time zones $Z = [-11, 12]$, for each sample interval i, the volume of content entering the system at that interval $v(i)$ is given by the sum of the contribution from each of the time zones at the interval, expressed as:

$$v(i) = \sum_{z \in Z} v_z(i) \tag{1}$$

The unknown quantity, $v_z(i)$, may be expressed as a function of: (1) the relative representation of each time zone in the data set (the weight of the time zone) ($w_z \in w$) and (2) some unknown distribution (p) governing the expected number of new posts per interval ($p(i)$), shifted to take account for the zone offset (o_z). More formally, given N intervals of equal length, the contribution of each time zone at some interval i may now be expressed:

$$v_z(i) = w_z \times p((i - o_z) \mod N) \tag{2}$$

where the offset (o_z) is given by $z \times \frac{N}{24}$. Substituting Equation 2 back into Equation 1, gives us:

$$v(i) = \sum_{z \in Z} w_z \times p((i - o_z) \mod N) \tag{3}$$

Note that expressed this way, the distribution p is independent of the time zone and depends only on the shifted sample interval. In other words, it assumes that when it comes to posting new content, users across the time zones behave in exactly the same way throughout the day; the only difference is the aggregate content posted within the given time zones, as expressed by the weights vector w.

To evaluate the value of p, Equation 3 can now be cast as:

$$\mathcal{W} \times p = v \tag{4}$$

Where $\mathcal{W}$ is an $N \times N$ sparse matrix, and each row of $\mathcal{W}$ holds the weights set ($w = [w_{-11}, \ldots, w_{12}]$) with the column index of weight w_z in row r given by $((r - o_z) \bmod N)$. All other elements in the row are zero.

Solving Equation 4 with respect to p yields the vector $p = \mathcal{W}^{-1} \times v$. The discrete distribution p can now be used to template the generalised 24 hour behaviour of a system with N sample intervals.

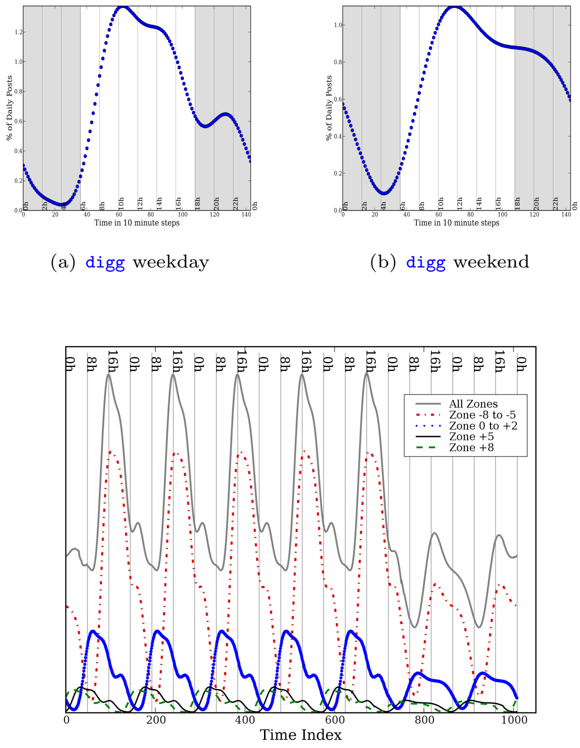
Figures 10(a) and 10(b) show the p distribution for the weekdays and weekends respectively and we gain a new perspective on their differences. Comparing the p distributions, we notice that the weekend has a flatter distribution over the 24 period (Figure 10(b)), characterised by the smaller magnitude of content at peak hour(s) (10:00 - 13:00hrs); and the relatively more gradual slopes before and after peak-time activity. On the other hand, the weekday behaviour shows a p distribution with much sharper slopes before and after the peak hours (09:00 - 11:00hrs) and incurs more content.

Figure 10 presents the 7 day week trend reconstructed by our model based on a linear combination of the expected contribution of the dominant (with regard to the digg data) time zones in America, Europe, Asia and Australia.

Figure 10 behaves as expected: the sum of the linear combinations from the model does recreate the k dominant frequencies it is built from, and more importantly, the shifts and peaks of the time zones are clearly displayed.

The geo-temporal model developed has been tested with the unprocessed 7 day data set gathered from reddit (see Figure 2(b)), to ascertain whether it provided a general model that can be applied to other content aggregation systems. Since it is only a 7 day trace of user content posting behaviour at a sample rate of 60 minutes (a total of 168 data points), the reddit trace is extremely sparse and provides a challenging example. Figure 11(a) plots the fit resulting from applying our model to this data.

The resulting fit captures the overall trend of the data. However, since it is a generalisation it misses outliers. More importantly, the generated trend is shifted to the left of the real data trace. Referring to Equation 4, this result implies that either: (a) the users of reddit have a heavy usage period starting later in the day compared to digg users, or, (b) the reddit user base comes form a different time zone distribution to the digg user base, thereby accounting for the time shift.



(a) digg weekday

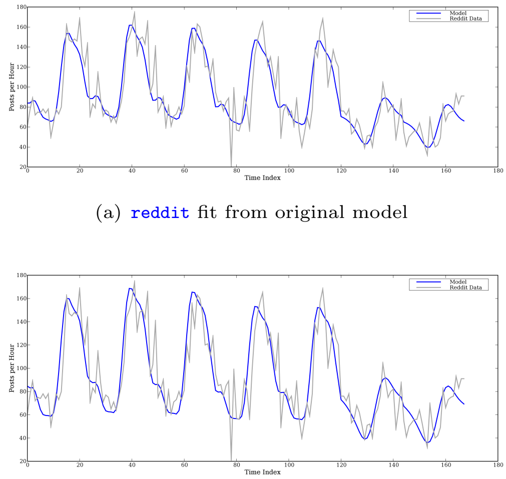
(b) digg weekend

(c) digg dominant time zones

Figure 10: The top graphs show the distribution that describes the percentage of the daily posts for weekdays and weekends in digg. The bottom graph shows these models applied to the dominant time zones in the dataset.

Given the orders of magnitude difference in size between the two datasets, a reasonable assumption to make here is that the digg data set is more representative of typical user behaviour. Therefore, the difference is due to a change in the distribution of the different time zones (the distribution of the w vector) in the reddit data set. Figure 11(b) is the product of optimising the w vector based on this assumption (see subsection 4.4 for a discussion on the optimisation). The resulting optimisation now has a mean squared error of 18.6137939226, compared to 17.732752241 from the original fit (Figure 11(a)). As can be seen, it primarily does a better job of fitting the peaks by removing the left shift observed in Figure 11(a). It is noted that the time zone optimised fit of Figure 11(b) appears to under-fit at the troughs, but we believe this to result from a combination of the smaller data size and the possibility of multiple local-minima around the calculated solution. However, we don't consider a better fit of the peaks more important since they have a higher influence of the expected values.

4.4 Generalisation of the Model



(a) reddit fit from original model

(b) reddit fit optimised

Figure 11: The top graph shows the fit of the digg model applied to reddit without any changes. The bottom graph shows the model with a stronger emphasis on the American time zones.

So far, we have discussed the modelling of the geo-temporal behaviour of users. In this section we look at how the model we have developed can be extended and specialised by either changing the assumptions or incorporating more information. To do so, we look at how to: (i) specialise the user location, (ii) the user contribution and (iii) the behaviour within a timezone.

User location: In creating the geo-temporal model, we inferred our user distribution from a data set and saw the significant effect of the time zone weighting - the proportional skewness of content towards preferred time zone. However, though our distributions carry an empirical justification, examiners can still change the weight distribution to model either more or less evenly spread systems.

In fact, changing the user distribution is a legitimate optimisation exercise in order to fit the model to a given data set. This is accomplished by rearranging the problem in Equation 4 to solve the following optimisation problem:

$$\begin{array}{rc} \textbf{minimise} & \mathcal{W} \times p - v = 0 \\ \textbf{subject to} & \sum_{\mathbb{W}_i} = 1 \end{array}$$

However, given the size and sparseness of state space defined by $\mathcal{W}$, there is a lot of opportunity for local minima unless users constrain not only the sum of columns of $\mathcal{W}$ but also the values ranges. For example when applying this optimisation to the reddit data set, we specified minimum values for the expected contribution of the US and Europe. The result yielded a weight distribution that favoured the -8hrs to +1hr GMT time zones and resulted in a more right leaning shift.

User contribution: So far in this section, we have not discussed the behaviour of individual users and have instead concentrated on aggregate behaviour. This can be extended by combining these aggregate statistics with the Pareto information in section 3, to develop finer grained models that micro model the behaviour of individual, rather than the macro viewpoint we have so far taken, i.e. multi agent based systems.

Inter and intra-time zone behaviour: We have built our models by making strong assumptions about the self similarity of users within and across time zones; however, these are not rigid. Examiners may change the intra-time zone behaviour of users e.g. their activity throughout a 24hr period. This assumption may be violated by placing constraints on the p vector. For example, given a data set composed of M intervals, we can place arbitrary constraints on the values that can be held in each of the individual M slots. Therefore, an examiner can explore the extremes of different user habits throughout an interval period.

5. DESIGN IMPLICATIONS

The previous sections have analysed data and derived a model for services with highly dynamic, user-generated content. In this section we turn our attention to some of the design implications that arise from these results.

5.1 Applying Geo-Temporal Information

Given enough data, the geo-temporal models developed in section 4 yield a reasonable estimation of the location and activity of users. This information can be used to deploy a distributed system according to where its users are located, or even to implement adaptive schemes that react to changes and deviations from the expected target.

Even though our models have been based on the posting behaviour of members, the periodic trends we have showed apply to other, quite different systems. In [13], for example, the connection behaviour of Microsoft's Live Messenger users resembles this trend, as does the behaviour of online gamers described in [14]. These results encourage us to treat this periodic behaviour as a generic phenomenon that can be applied to a variety of scenarios, including:

- **Energy-efficient load balancing.** Our model can be used to predict the behaviour of users both throughout a given day and across timezones. Chen et al. [13] have developed energy-saving server provisioning and load dispatching algorithms for the

MSN Live Messenger based on temporal data with much the same properties as our data set. Our models could be used to extend their algorithms in order to predict not only the expected load throughout the day, but also the timezone distribution. In this way, the energy-saving algorithms can be based not only on the number of servers, but also on *which* servers should be used at any point in time.

- **Peer-to-Peer churn.** Systems such as Chord [15], CAN [16] or Pastry [17] take advantage of structured graphs in order to achieve good routing performance. It is essential to keep these routing tables intact and up-to-date in order to route messages and locate content efficiently. Users of these systems tend to arrive and depart frequently, and so there is a high cost associated with routing table maintenance.

  Our results can be used to extend the work of Rhea et al [18] by improving the notion of proximity-based neighbour selection (currently measured in terms of latency) to include the location of users and the intra-timezone behaviour of populations. Such an extension would result in nodes having more stable neighbours (i.e. neighbours that stay in the system for longer periods of time), thus reducing routing table churn.

- **Peer-to-Peer content distribution.** The performance of content distribution systems such as BitTorrent [19] can be greatly enhanced by replicating and caching popular content. However, this mechanism does not come for free, since copying content consumes both bandwidth as it is transferred and hard drive space at the receiver. To alleviate this, our temporal model can help so that replicas are only sent to nodes that are more likely to stay in the system for long periods. In addition, the model's ability to describe peak usage hours for each time zone would help the distributed system to decide ahead of time where replication and caching is most needed at any point in time.

5.2 Popularity Prediction and Caching

In subsection 3.2 we showed that both `digg` and `reddit` follow the Pareto Principle, meaning that most of the votes are for a few popular posts and that the great majority of the content is essentially discarded. Any system that shows this kind of behaviour has the potential to decrease costs by replicating and caching popular data [20][21].

However, the dynamic nature of systems like `reddit` and `digg` makes the design of caching and replication strategies more challenging compared to that of systems with longer-lived content. The high rate of submissions makes it difficult not only to predict which posts will become popular, but also how long they will remain so, both important factors when considering caching. Contrast this to a scenario where, for instance, a web server is hosting resources that are not changed very often: popular resources are identifiable much more easily and will most likely stay popular for a long time.

As a result of these characteristics, we need a predictive model in order to minimise the costs of replication and improve the effectiveness of caching. Preliminary analysis of the popularity evolution of data over time (see Figure 6) has shown that it is possible to predict the popularity of data at a given time t based only on its popularity between 0 and $t-1$. While a more detailed investigation is needed, these early results lead us to believe that, combined with time zone information, we can use a predictive model to decide how much to cache and where these caches should be located.

6. RELATED WORK

In [13], the authors aim to reduce the energy costs of maintaining services for users of the Windows Live Messenger service. To do so, they measure the activity of users (in terms of login rates) and use this information to develop novel load balancing and dispatching algorithms that take advantage of daily and weekly patterns in user activity. The aggregate patterns of activity studied in this work closely resemble those shown in this paper, albeit with a different daily workload distribution. Daily and weekly usage patterns similar to our findings are also reported by Chambers et al. [14] when analysing the workload of several online games. Our work can be applied to complement these works by suggesting not only how to distribute finite resources, but also in which locations to place them.

Golder et al.[22] analysed several million messages exchanged on the social networking site Facebook, showing that the usage patterns of this system display strong periodic trends (weekdays to weekends).

Cha et al. [4] give a detailed data-driven analysis of the popular video platform `YouTube`. They found that 80% of the videos views are for 10% of the videos, similar to our findings about the vote distributions. Furthermore, they looked at the popularity life-cycle of videos, reporting that 80% of the video requests over the course of a day are for videos that are older than 1 month. Moreover, it can happen that old videos suddenly become popular due to long tail effects and recommendation from other sites, in sharp contrast to the behaviour in our dataset.

Mickens and Noble [23] look at node availability in distributed systems. They develop predictors to estimate the future uptime of nodes and apply these techniques in order to identify highly available nodes when placing replicas. The analysis we have presented offers the potential to extend this approach, making use of richer context information such as geographical location and content popularity in order to support replica selection.

Other research focuses on the structure of social networks and examines how users can be clustered according to their commenting behaviour on each others' posts [24]. Mislove et al. [25] report small-world properties and a high level of local clustering in their analysis of the networks Flickr, YouTube, LiveJournal and Orkut. They propose to take advantage of these structures when designing information dissemination algorithms by, for example, using hierarchical structures. Singla et al. [26] show that there is a relationship between people's shared interest and how often and how long they chat on messenger systems.

7. CONCLUSIONS AND FUTURE WORK

We have presented a detailed analysis of digg and reddit, two of the most popular social news sites and have looked at (i) the content generation process and (ii) the popularity distribution and evolution of content in these systems.

With regard to content generation, we found that both systems show clear, periodic trend throughout the week, while the volume of data coming into the system not only varies between weekdays and weekends but also within a 24 hour period. With regard to the popularity of content, we found that the systems have a few extremely popular posts, while most of their content remains largely unused. Further, the content that becomes popular does so very fast but loses its popularity just as quickly due to new content entering the system.

Accounting for user location information has enabled us to derive a geo-temporal model that describes the 24 hour behaviour of digg users within a given time zone - for both weekdays and weekends. The model shows that digg follows a work-day pattern peaking at 10am in the morning. From the location information we learnt that although users are scattered around the world, both systems are dominated by English speaking nations, in particular, the US.

Our results indicate that potential benefits to content availability and system performance can be gained by caching and replicating popular content. However, in the systems we have studied, most of the content displays rapid popularity evolution, coupled with a short lifetime. Hence, in order to utilise caching and replication best, it is essential to identify popular content quickly.

As discussed, recent literature describes periodic trend-based behaviour similar to our observations exhibited by the users of quite different systems. We assert that the geo-temporal model that we have developed provides the basis for simple and parametrisable tools to study the macro-scale behaviour of a range of user participation orientated systems i.e. instant messaging and online gaming. The temporal and geographic details of our model can be used to address a range of issues including energy-efficiency, load balancing, peer-to-peer node churn, and peer-to-peer content distribution.

In fact, we have taken some steps towards these goals in two directions; first, preliminary work shows that we can accurately predict the future popularity of content based on its previous popularity, enabling us to contemplate proactive rather than reactive content replication. Second, understanding the inter-time zone interaction with content affords us to identify not only *what* content needs replication but also *where* it is needed next and *when*.

8. REFERENCES

[1] http://digg.com/.

[2] http://reddit.com/.

[3] http://www.youtube.com/.

[4] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I Tube, You Tube, Everybody Tubes: Analyzing the World's Largest User Generated Content Video System," in *ACM Internet Measurement Conference*, October 2007.

[5] http://apidoc.digg.com/.

[6] Pareto principle - Wikipedia, the free encyclopedia. http://en.wikipedia.org/wiki/Paretoprinciple.

[7] L. A. Adamic, "Zipf, power-laws, and pareto - a ranking tutorial." http://www.hpl.hp.com/research/idl/papers/ranking/ranking.html.

[8] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," 2007.

[9] http://associatedcontent.com.

[10] http://squidoo.com/.

[11] http://helium.com/.

[12] S. Hogan, "Map of digg traffic." http://www.shawnhogan.com/2006/05/map-of-digg-traffic.html, May 2006.

[13] G. Chen, W. He, J. Liu, S. Natha, L. Rigas, L. Xiao, and F. Zhao, "Energy-aware server provisioning and load dispatching for connection-intensive internet services," in *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pp. 337–350, 2008.

[14] C. Chambers, W. chang Feng, S. Sahu, and D. Saha, "Measurement-based characterization of a collection of on-line games," in *IMC '05: Proceedings of the 5th ACM SIGCOMM conference on Internet measurement*, (New York, NY, USA), pp. 1–14, ACM, 2005.

[15] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” in *Proceedings of the ACM SIGCOMM ’01 Conference*, (San Diego, California), August 2001.

[16] *A scalable content-addressable network*, vol. 31, ACM Press, October 2001.

[17] A. Rowstron and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems,” in *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, (Heidelberg, Germany), November 2001.

[18] S. Rhea, D. Geels, T. Roscoe, and J. Kubiatowicz, “Handling churn in a dht,” in *Proceedings of the 2004 USENIX Annual Technical Conference (USENIX ’04)*, (Boston, Massachusetts), June 2004.

[19] B. Cohen, “Incentives build robustness in bittorrent,” 2003.

[20] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, “Web caching and zipf-like distributions: Evidence and implications,” in *NFOCOM*, pp. 26–134126–134, 1999.

[21] C. Cunha, A. Bestavros, and M. Crovella, “Characteristics of World Wide Web Client-based Traces,” Tech. Rep. BUCS-TR-1995-010, Boston University, CS Dept, Boston, MA 02215, April 1995.

[22] S. A. Golder, D. Wilkinson, and B. A. Huberman, “Rhythms of Social Interaction: Messaging within a Massive Online Network,” in *3rd International Conference on Communities and Technologies (CT2007). East Lansing, MI*, June 2007.

[23] J. W. Mickens and B. D. Noble, “Exploiting availability prediction in distributed systems,” in *NSDI’06: Proceedings of the 3rd conference on 3rd Symposium on Networked Systems Design & Implementation*, (Berkeley, CA, USA), pp. 6–6, USENIX Association, 2006.

[24] V. Gómez, A. Kaltenbrunner, and V. López, “Statistical analysis of the social network and discussion threads in slashdot,” in *WWW ’08: Proceedings of the 17th international conference on World Wide Web*, (New York, NY, USA), ACM, April 2008.

[25] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, “Measurement and analysis of online social networks,” in *IMC ’07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, (New York, NY, USA), pp. 29–42, ACM, 2007.

[26] P. Singla and M. Richardson, “Yes, there is a correlation - from social networks to personal behavior on the web,” in *WWW ’08: Proceedings of the 17th international conference on World Wide Web*, 2008.