SpinThrift: Saving Energy in Viral Workloads

Nishanth Sastry and Jon Crowcroft
Computer Laboratory, University of Cambridge
first.lastname@cl.cam.ac.uk

ABSTRACT

This paper looks at optimising the energy costs for storing user-generated content when accesses are highly skewed towards a few "popular" items, but the popularity ranks vary dynamically. Using traces from a video-sharing website and a social news website, it is shown that the non-popular content, which constitute the majority by numbers, tend to have accesses which spread locally in the social network, in a viral fashion. Based on the proportion of viral accesses, popular data is separated onto a few disks on storage. The popular disks receive the majority of accesses, allowing other disks to be spun down when there are no requests, saving energy.

Our technique, SpinThrift, improves upon Popular Data Concentration (PDC), which, in contrast with our binary separation between popular and unpopular items, directs the majority of accesses to a few disks by arranging data according to popularity rank. Disregarding the energy required for data reorganisation, SpinThrift and PDC display similar energy savings. However, because of the dyamically changing popularity ranks, SpinThrift requires less than half the number of data reorderings compared to PDC.

Categories and Subject Descriptors

C.5.5 [**Computer System Implementation**]: Servers; D.4.2 [**Operating Systems**]: Storage Management—*Secondary storage*

General Terms

Measurement, Performance

Keywords

energy conservation, social networks, viral workloads

Green Networking 2010, August 30, 2010, New Delhi, India.

1. INTRODUCTION

The mushrooming of HD quality user-generated content[1] is increasing both the amount of storage required, as well as the energy costs of maintaining the stored content on the Web. To illustrate, consider the following statistic recently announced by YouTube[2]: every minute, over 24 hours of videos are being uploaded. At a conservative (for HD) bitrate estimate of 3-5Mbps, this translates to 44-74 TB of data per day. Using conventional 512 GB SATA disks with a 12 W operating power, merely storing one day's uploads *for an hour* costs 1.056-1.776 kWh, more than the quarterly consumption of a household in Scotland[3]. Note that YouTube's current limits of 2GB file uploads and 10 minute long videos allows for a full HDV encoding of 27.3 Mbps. Thus, the required storage capacity and energy could be up to 9 times larger.

This paper presents an approach to conserve energy by exploiting the access patterns of user-generated content. We use data drawn from Vimeo, a video-sharing website with an active social aspect, and digg, a social news website, as examples of such access patterns. On both sites, we observe the following:

- Popularity distribution is highly skewed—a few stories are accessed many more times than others.
- The set of items accessed, as well as their popularity ranks, vary over successive time windows.
- Accesses to popular content tend to occur over large time windows in comparison with non-popular content, which experience brief periods of interest.

Based on the above findings, we design *SpinThrift*, a strategy for saving energy in the storage subsystem. The core idea, similar to Popular Data Concentration (PDC) [12], is to exploit the skewed access pattern and intelligently arrange the data by periodically migrating popular content to a subset of disks. This allows the other disks to be spun down to a low power mode, when not accessed.

Maintaining a strict popularity-based data ordering as in PDC would require a large number of migrations because of the changes in popularity ranks and the set of items accessed. Instead, we make a binary classification separating

[1] e.g. See `http://youtube-global.blogspot.com/2009/11/1080p-hd-comes-to-youtube.html`
[2] `http://youtube-global.blogspot.com/2010/03/oops-pow-surprise24-hours-of-video-all.html`
[3] `http://www.sesg.strath.ac.uk/Reasure/Info_pack/RE_info/hec.htm`

the popular content from the non-popular. Because popular items see accesses over large time windows, the number of accesses seen during any given migration period is not a good indicator of long-term popularity, and might increase the number of data migrations. Similarly, because the popularity distribution has a high variance, the cumulative accesses seen until a given time point is not truly representative of eventual popularity rank.

SpinThrift addresses the above problems by employing information from the social network to identify popular items. If an item is accessed by a user after a direct friend on the social network has accessed it, we term the access as *viral*. In contrast, if no direct friend has accessed the item, then the access is termed as *non-viral*. Relative proportions of viral and non-viral accesses are used to infer popularity.

Contrary to expectation, our traces show that items that predominantly propagate virally, from friend to friend, *do not* lead to popularisation of the item. Rather, stories that become popular are typically independently (non-virally) introduced at different points in the social network, by different people. Intuitively, an item which is globally popular becomes interesting to people in unrelated parts of the social network, whereas virally propagating items tend to be localised to parts of the network because of effects like homophily.

SpinThrift exploits this phenomenon and uses the relative proportion of non-viral to viral accesses to separate popular content from unpopular content. Simulations show that SpinThrift achieves a similar energy consumption as Popular Data Concentration while requiring less than half the number of migrations.

The rest of this paper proceeds as follows: Section 2 discusses access pattern trends seen in our traces. Section 3 applies the findings about access patterns and proposes the SpinThrift scheme to save energy. Section 4 discusses related work. Section 5 concludes.

2. TRENDS IN CONTENT ACCESS

This section focuses on identifying trends in access patterns, using data from a social news site and a video sharing sites. First, we describe the data sets. Following this, we discuss the popularity distribution. Then we discuss the social network aspects of popularity.

2.1 Datasets

Our first trace is based on one week of data from Aug 29, 2008-Sep6, 2008 from Digg.[4]. Digg is a social news website: Users submit links and stories. Digg users collectively determine the popularity of a story by voting for stories, and commenting on it. Voting for a story is termed as "digging", and the votes are called "diggs". Collectively, "diggs" and comments are termed item activities. Some stories are identified as "popular" by digg, using an undisclosed algorithm. We use these for ground truth about popularity values[5].

Digg also has a social networking aspect. Users can follow the activities of other users by "friending" them. Note that these are one way links, and are frequently not reciprocated. The Digg website highlights friends' activities and stories "dugg" by friends.

[4] `http://www.digg.com`

[5] Our results are qualitatively similar if we equivalently use the method described below for vimeo on the digg data.

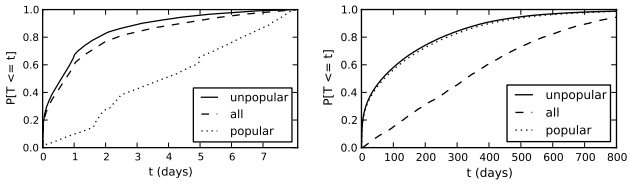
Our second trace is sampled from videos and user ids found in all the groups and channels of the video-sharing website Vimeo[6]. Similar to digg, users can "like" videos; these are counted as story activities. A story whose "likes" number more than one standard deviation in excess of mean is counted as "popular". Users also have contacts, similar to digg friends. The vimeo website highlights the activities of contacts by allowing users to follow their friends' channels, subscriptions, likes etc. The details of both data sets are summarised in Table 1.

	digg	vimeo
Item statistics		
Number of items	163,582	443,653
"Popular" items	17,577	7,984
item activities	2,692,880	2,427,802
Graph statistics		
Number of users	155,988	207,468
directional links	1,516,815	718,457

Table 1: Trace details

2.2 Popularity trends

In this section, we examine the popularity distribution of data activities to determine how best to optimise storage.



(a) digg (b) vimeo

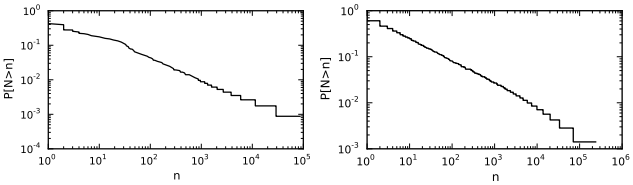
Figure 1: **Popular items have sustained period of interest**: CDF of time window of interest (time of last access minus time of first access) shows that unpopular items have much briefer windows of interest than popular items.

Fig 1 shows that popular stories get accessed over a much longer time window than unpopular stories. For instance, nearly 80% of unpopular stories in digg have a time window of less than 2 days, whereas the time window for popular stories is spread nearly uniformly over the entire duration of the trace.

This implies that for most popular stories, there is a *sustained* period of interest rather than a brief surge. This in turn suggests that a disk management strategy which treats popular stories differently from unpopular stories. In the next section, we develop a simple scheme to predict which stories will be popular and have a different window of interest than other stories.

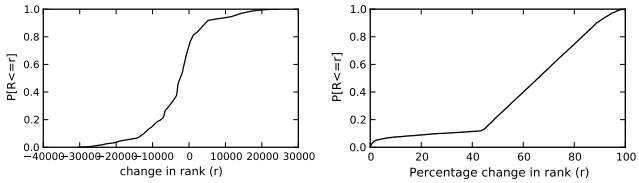
It must be noted that there is a boundary effect at play. Some of the digg news articles (similarly videos) could have a longer interest span than the one week we consider. It is equally possible that interest in an old article could have

[6] `http://www.vimeo.com`



(a) digg (b) vimeo

Figure 2: CCDF of number of accesses suggests a power-law.



(a) Signed change in rank (b) Percentage change in rank

Figure 3: **Dynamic popularity**: CDF of difference in popularity ranks between two successive days shows that ranks can vary dramatically in digg (left) and vimeo (right)

just ended one or two days into our study. Regardless, it is easy to see that there is a significant difference in the distributions of the time windows of interest between popular vs. unpopular stories.

Next, we look at the number of accesses received by a story through the entire duration of our trace. Fig 2 depicts the complementary cumulative distribution of the number of the accesses, showing a distinct power law-like distribution—a select few stories receive well over 10,000 accesses whereas a large number of stories receive very few (< 100) accesses.

Fig 2 by itself would suggest placing the popular stories on "hot" disks, and storing the unpopular stories on other disks that can be switched off or put into a low-power mode when not being accessed. However, this ignores the dynamics of popularity rankings. We examine this further by looking at the evolution of popularity between two successive one-day windows (chosen randomly) in the traces. Stories are first ranked by the number of accesses. If a story is accessed in only one of the days, it is assigned a maximum rank equal to the number of stories seen across both time periods.

Fig 3 shows the distribution of the change of popularity ranks. Fig. 3 (a) shows the signed change in rank for digg. Fig. 3 (b) provides a different perspective and shows the absolute rank change for vimeo as a percentage. Both views depict rapidly changing popularity ranks. For instance, in vimeo, nearly half the videos have a rank change of more than 60%. This dramatic change in ranks is driven by the large numbers of new items that are added each day. This implies that a strict popularity ordering of items across disks would require excessive shuffling of items.

2.3 Social network aspects of popularity

Crucially, both the vimeo and digg data sets have information about the social network of the users as well as their access patterns. We next examine how the social network impacts on the popularity. Inspired by theories of viral propagation of awareness about stories, products, etc. that have become prominent recently (e.g. [8]), we tested the hypothesis that the stories which become popular are the ones that spread successfully within the social network. Fig 4 presents a set of results that collectively suggest (contrary to expectation) that stories are extremely unlikely to become popular when viral accesses predominate over non-viral accesses.

Fig. 4 (a) bins videos in vimeo by the ratio of viral accesses to non-viral accesses (rounded to one decimal place). It then measures the number of popular stories, counted as the number of stories with more than the average number of likes. It can be seen that as the ratio of viral to non-viral accesses increases, the number of popular stories falls drastically. A qualitatively similar result can be obtained for digg.

Note that digg, vimeo and many other websites hosting user-generated content typically highlight a few items on the front page. Such items can be expected to be popular in terms of number of accesses. Furthermore, many of the accesses are also likely to be from people unrelated to each other (non-viral). To discount this effect, Fig. 4 (b) examines stories on digg *before* they are highlighted on the front page as "popular" stories. Digg denotes a small subset of stories as "popular" using an undisclosed algorithm that reportedly [6]

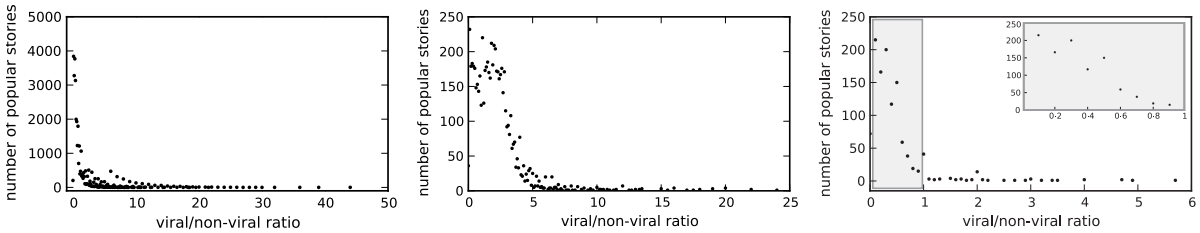
> "takes several factors into consideration, including (but not limited to) the number and diversity of diggs, buries, the time the story was submitted and the topic."

Fig. 4 (b) shows that even in this subset of "yet to be marked popular" stories, a predominance of viral accesses greatly decreases the possibility that a story is popular.

We conjecture that while there may be individual successes of "viral marketing" strategies, in general, a story which spreads only by viral propagation remains stuck in one part of the social network. In contrast, an inherently popular story is independently seeded at several places in the social network and therefore becomes popular. Note that even the local success of viral strategies could be partly attributed to homophily, the tendency of friends to like the same things, rather than users actively influencing their friends to watch a video or news story [1].

Finally, Fig. 4 (c) counts the ratio of viral/non-viral accesses in stories *deemed by digg* to be popular. Interestingly, this graph has a knee around a viral to non-viral ratio of 1. There are hardly any popular stories with this ratio greater than 1, i.e., when a story has more viral than non-viral accesses, the probability that the story is popular is almost negligible. When the viral to non-viral access ratio is less than one, the probability that the story is popular is proportional to the ratio[7]. This is clearly seen from the inset graph which zooms in on the gray region (between 0 and 1 on the x-axis) shown on the main graph. In other words, it appears that digg's well-tuned algorithm for marking a story as popular can be closely approximated by a simple algorithm that marks a story as popular with a probability proportional to the ratio of viral to non-viral accesses. In the next section, we will adopt a similar strategy to mark a story as popular or unpopular in SpinThrift.

[7]The actual counts shown in the figure can be re-scaled by the total number of popular stories to obtain the probability.



(a) vimeo (popularity from #likes) (b) digg (popularity from #diggs obtained before being deemed popular) (c) stories deemed popular by digg (Inset focuses on $\frac{\text{viral}}{\text{non-viral}} < 1$)

Figure 4: **Non-viral accesses predominate in popular stories**: x-axis shows ratio of number of viral accesses to non-viral accesses. Y axis shows the number of popular stories that have the corresponding ratio. To clearly show zero valued points, the origin (0,0) is slightly offset from the corners of each graph.

3. DATA ARRANGEMENT SCHEMES

In this section, we investigate the use of intelligent data arrangement to save energy in the storage sub-system. The basic idea is to arrange data so as to skew the majority of access requests towards a small subset of disks. Other disks can then be put in an idle mode at times when there are no access requests for content on those disks.

The highly skewed nature of the number of accesses (Fig 2) suggests the basic data arrangement strategy of organising data according to their popularity. Within this space, we explore two alternatives: The first, Popular Data Concentration, uses the Multi Queue algorithm to maintain a popularity ordering over all files. The second, SpinThrift, uses the relative proportions of viral and non-viral accesses to distinuish popular and unpopular data. Within each class, data is ordered so as to minimise the number of migrations.

Our simplified simulations indicate that both Popular Data Concentration and SpinThrift end up with similar energy consumptions, ignoring energy involved in periodically migrating data. SpinThrift results in significantly fewer data migrations, thereby requiring lesser energy overall, as compared to Popular Data Concentration.

3.1 Popular Data Concentration

We use Popular Data Concentration (PDC) as specified by Pinheiro and Bianchini, with minor changes[8]. PDC works using the Multi Queue (MQ) cache algorithm to maintain popularity order. The MQ cache works as follows [14]:

There are multiple LRU queues numbered in Q_0, Q_1,..., Q_{m-1}. Following Pinheiro and Bianchini, we set $m = 12$. Within a given queue, files are ranked by recency of access, according to LRU.

Each queue has a maximum access count. If a block in queue Q_i is accessed more than2^i times, this block is then promoted to Q_{i+1}. Files which are not promoted to the next queue are allowed to stay in their current LRU queue for a given lifetime (The file-level lifetime is unspecified by Pinheiro and Bianchini. We use a half hour lifetime). If a file has not been referenced within its lifetime, it is demoted from Q_i to Q_{i-1}. Deviating from PDC, we do not reset the access count to zero after demoting a file. Instead, we halve the access count, giving a file a more realistic chance of being promoted out of its current queue during our short half-hour lifetime period. In our method, all files within a queue Q_i obey the invariant that their access count is between 2^{i-1} and 2^i.

Periodically (every half hour, in PDC), files are dynamically migrated according to the MQ order established as above. Thus, the first few disks will end up with the most frequently accessed data and will remain spinning most of the time. Disks with less frequently accessed data can be spun down to an idle power mode, typically after a threshold period of no accesses (17.9 seconds, as used by Pinheiro and Bianchini).

3.2 SpinThrift

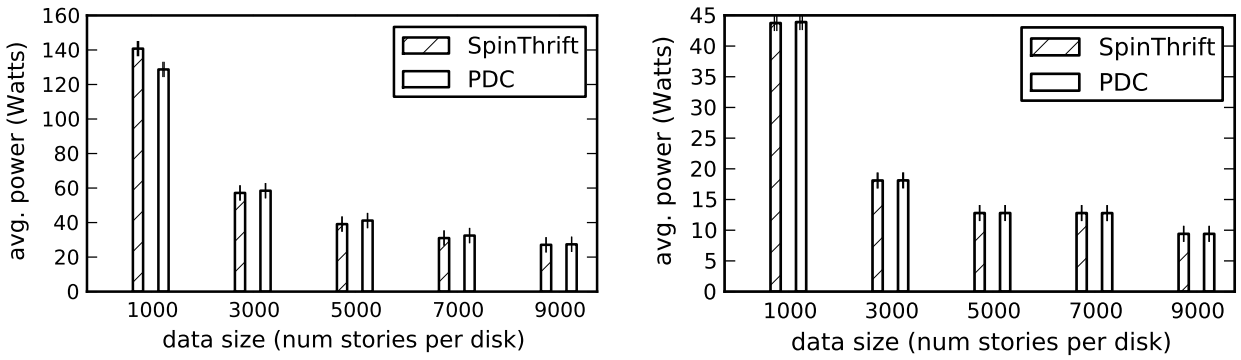
The periodic migration of PDC, coupled with the changing nature of popularity ranks (see Fig. 3) can lead to a large number of files being moved across disks at each migration interval. To alleviate this, we propose SpinThrift, which separates the popular data from the unpopular, without imposing a complete ordering.

We use results from Section 2.3 to find popular data: SpinThrift uses the social network graph of the users, and keeps track of whether an access to a story is viral or non-viral. An access by a user is *viral* if one of the friends of the user has accessed the same story before. In contrast, if the user who is accesssing a story has no friends amongst previous users of the data, we deem that access as *non-viral*.

SpinThrift implements a policy of labeling a story as popular when the number of non-viral accesses exceeds the number of viral accesses. Popular stories are much fewer in number than unpopular ones. Therefore unpopular data is subclassified based on the median of the median hour of access of previous users. Since the window of interest for unpopular stories is much smaller than for popular stories (see Fig 1), this organisation could reap additional locality benefits from time of day effects in user accesses.

The above classification is used to construct a ranking scheme for data, which is then used to periodically migrate data to different disks. The ranking works as follows: At the top level, popular stories are ranked above unpopular ones. Unpopular data are further ranked based on the median hour of previous accesses as above. Data within each sub

[8]PDC is specified in detail for handling block-level requests. At the file level, it is only specified that the operation is analogous to the block-level case. Details which are left unspecified for the file case are filled in by us.



(a) digg (b) vimeo

Figure 5: **Relative energy profiles**: SpinThrift consumes roughly same energy as Popular Data Concentration (PDC), without taking migration energy or number of migrations into account.

group maintain the same relative ranking to each other as before, thereby reducing the number of migrations required.

Just as with PDC, a migration task runs every half hour, rearranging stories according to the above order. Similarly, disks which do not receive any access requests are spun down to an idle power mode after a threshold period of 17.9 seconds. The migration interval and idle threshold values were chosen to be the same as PDC for ease of comparison below.

3.3 Evaluation

Operating power	12.8 Watts
Idle power	7.2 Watts
Idle threshold	17.9 Secs
Transition power	13.2 Watts
Transition time	6.0 Secs

Table 2: Power parameters

In this section, we investigate the relative merits of PDC and SpinThrift using a simplified simulation scheme, similar to that used by Ganesh et al. [7]. We assume that there is an array of disks, sufficient to hold all data. A disk moves into operating mode when there is an access request for data held in that disk. After an idleness threshold of 17.9 seconds, the disk is moved back into an idle mode. Disks consume 12.8 Watts when operating at full power, as compared to 7.2 Watts in idle mode. The transition between modes consumes 13.2 * 6 = 79.2 Joules. These details are summarized in Table 2.

Our simulations are driven by access requests from the data sets described before. Note that our data set does not indicate the sizes of the data as stored on the disk. We account for this using disks of different capacities, by counting the number of data items that fit on each disk. We use conservative numbers of between 1000 and 10,000 data items per disk[9].

[9]These are quite realistic: Extending the example from the introduction, only about 1400 10 minute long clips can be stored on a 512 GB disk if the bitrate is 5 Mbps.

Our first simulation uses requests drawn from a random 3 day interval from the digg trace, and measures the average power consumption on the final day after an initial warm up period. For the vimeo trace, a random 30-day interval is used, and the average power consumption on the final day is measured. Fig 5 shows the relative power profiles for disk arrays using disks of different sizes. Observe that both PDC and SpinThrift require similar power, with PDC doing slightly better because it tracks the popularity more accurately.

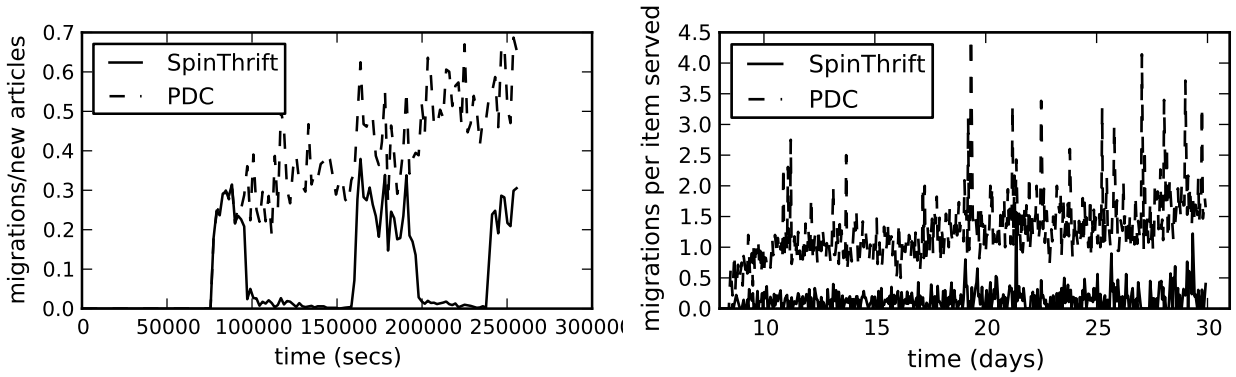
The above experiment does not take into account the energy involved in data migration. The next simulation, in Fig. 6, plots the number of files needing to be migrated for a disk which can accommodate 5000 data items. The number of files migrated at every half-hourly migration point is shown as a fraction of the total number of new articles that arrived during the interval.

Observe that after an initial warm up period, the number of files requiring migration under PDC keeps growing continuously. This is a result of the changing popularity of content items – as new articles are introduced into the system and become popular, they move to the head of the list, requiring all articles before them to be moved lower on the ranking list. In contrast, SpinThrift requires many fewer migrations.

In summary, the simple evaluation above suggests that SpinThrift is able achieve a similar power profile as PDC, with many fewer migrations. We emphasise that our simulations are highly simplified. In particular, they do not consider the effect of repeated data migration or power cycling on disk reliability. We also assume that disks are not bandwidth constrained, i.e., that the most popular disks can support all the data requests without increasing request latency. We also do not model the costs of increased latency for accesses directed at disks in idle/low-power mode.

4. RELATED WORK

While it is hard to make concrete claims about the representativeness of our data set for access patterns involving other kinds of user-generated content, various trends such as skewed access distributions and dynamically changing popu-



(a) digg

(b) vimeo

Figure 6: SpinThrift has significantly fewer migrations than Popular Data Concentration (PDC).

larity have been found to hold for other user-generated content as well [2].

Our scheme improves upon Popular Data Concentration, which was designed explicitly for highly skewed, zipf-like work loads [12]. Popular Data Concentration, in turn improves upon an older scheme, Massive Array of Idle Disks (MAID), which attempts to reduce disk energy costs by using temporal locality of access requests [5]. The above family of disk energy conservation techniques look at read-heavy work loads, and can be viewed as orthogonal to techniques which look at write work loads (e.g. [11] [7]).

Our work looks at the trade-off between saving energy in the storage subsystem using intelligent data arrangement policies, and the number of file migrations that the policy requires. Various other trade-offs exist in the space of disk energy savings and could be applied in addition to our technique, depending on the work load. For example, Gurumurthi et al. [9] looks at the interplay between performance and energy savings. Zheng et al. [4] examine the trade-off between dependability, access diversity and low cost.

To the best of our knowledge, our work is one of the first to look at work loads caused partly by viral propagation of information. In a related piece of work, we have previously looked at replica placement for similar work loads [13].

An important contribution of this paper is the study of information propagation on digg and vimeo. Others have looked at information propagation in Flickr [3], Amazon [10] etc. In both, as in our study, there is evidence that purely viral propagation of information is largely ineffective.

5. CONCLUSION

This work presented SpinThrift, a technique designed to mitigate the energy costs of operating hard drives containing user-generated content. Such strategies would be useful both on server systems specialising in rich-media user-generated content, as well as on proxies, content-delivery networks, and other intermediary caches that store the content closer to users.

SpinThrift exploits the observation that accesses are skewed towards a subset of popular items and periodically re-arranges data so as to direct the majority of access requests to a small subset of disks. Disks containing unpopular items can then be put in an idle mode, at times when there are no access requests for content on those disks.

SpinThrift re-arranges data by using a novel social-network based predictor for identifying popular data items and separating them from the rest. Our evaluations show that this binary separation of items achieves a power consumption comparable to a scheme that computes an optimal ordering of items based on strict popularity ranking. Furthermore, the binary separation significantly decreases the number of data items that need rearrangement as compared to a strict popularity ranking-based scheme.

6. ACKNOWLEDGEMENTS

We are deeply grateful to Anthony Hylick for early discussions and the original suggestion that started us on this work. This work was supported in part by the EPSRC INTelligent Energy awaRe NETworks (INTERNET) Project. Nishanth Sastry is funded by a St. John's Benefactor's Scholarship.

7. REFERENCES

[1] Aral, S., Muchnik, L., and Sundararajan, A. Distinguishing influence-based contagion from homophily-driven diffusion in dynamic networks. *Proceedings of the National Academy of Sciences 106*, 51 (2009), 21544–21549.

[2] Cha, M., Kwak, H., Rodriguez, P., Ahn, Y.-Y., and Moon, S. I tube, you tube, everybody tubes: analyzing the world's largest user generated content video system. In *IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement* (2007).

[3] Cha, M., Mislove, A., and Gummadi, K. P. A measurement-driven analysis of information propagation in the flickr social network. In *WWW '09: Proceedings of the 18th international conference on World wide web* (New York, NY, USA, 2009), ACM, pp. 721–730.

[4] Chen, M., Stein, L., and Zhang, Z. Dependability, access diversity, low cost: pick two. In *HotDep'07:*

Proceedings of the 3rd workshop on on Hot Topics in System Dependability (2007).

[5] Colarelli, D., and Grunwald, D. Massive arrays of idle disks for storage archives. In *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing* (2002), pp. 1–11.

[6] Digg. Digg faq. `digg.com/faq`.

[7] Ganesh, L., Weatherspoon, H., Balakrishnan, M., and Birman, K. Optimizing power consumption in large scale storage systems. In *11th USENIX Workshop on Hot Topics in Operating Systems* (May 2007).

[8] Gladwell, M. *Tipping Point*. Back Bay Books, 2002.

[9] Gurumurthi, S., Zhang, J., Sivasubramaniam, A., Kandemir, M., Franke, H., Vijaykrishnan, N., and Irwin, M. J. Interplay of energy and performance for disk arrays running transaction processing workloads. In *ISPASS '03: Proceedings of the 2003 IEEE International Symposium on Performance Analysis of Systems and Software* (2003).

[10] Leskovec, J., Adamic, L. A., and Huberman, B. A. The dynamics of viral marketing. *ACM Trans. Web 1*, 1 (2007), 5.

[11] Narayanan, D., Donnelly, A., and Rowstron, A. Write off-loading: Practical power management for enterprise storage. *Trans. Storage 4*, 3 (2008).

[12] Pinheiro, E., and Bianchini, R. Energy conservation techniques for disk array-based servers. In *ICS '04: Proceedings of the 18th annual international conference on Supercomputing* (2004), pp. 68–78.

[13] Sastry, N., Yoneki, E., and Crowcroft, J. Buzztraq: predicting geographical access patterns of social cascades using social networks. In *SNS '09: Proceedings of the Second ACM EuroSys Workshop on Social Network Systems* (2009).

[14] Zhou, Y., Philbin, J., and Li, K. The multi-queue replacement algorithm for second level buffer caches. In *Proceedings of the General Track: 2002 USENIX Annual Technical Conference* (2001).