Program Popularity and Viewer Behaviour in a Large TV-on-Demand System

Henrik Abrahamsson
Swedish Institute of Computer Science
Box 1263, Kista, Sweden
henrik@sics.se

Mattias Nordmark
TeliaSonera AB
Stockholm, Sweden
mattias.nordmark@teliasonera.com

ABSTRACT

Today increasingly large volumes of TV and video are distributed over IP-networks and over the Internet. It is therefore essential for traffic and cache management to understand TV program popularity and access patterns in real networks.

In this paper we study access patterns in a large TV-on-Demand system over four months. We study user behaviour and program popularity and its impact on caching.

The demand varies a lot in daily and weekly cycles. There are large peaks in demand, especially on Friday and Saturday evenings, that need to be handled.

We see that the cacheability, the share of requests that are not first-time requests, is very high. Furthermore, there is a small set of programs that account for a large fraction of the requests. We also find that the share of requests for the top most popular programs grows during prime time, and the change rate among them decreases. This is important for caching. The cache hit ratio increases during prime time when the demand is the highest, and caching makes the biggest difference when it matters most.

We also study the popularity (in terms of number of requests and rank) of individual programs and how that changes over time. Also, we see that the type of programs offered determines what the access pattern will look like.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of Systems; C.2.3 [**Computer Systems Organization**]: Computer-Communication Networks—*Network Operations*

General Terms

Measurement, Performance

Keywords

IPTV, TV-on-Demand, Program Popularity

IMC'12, November 14–16, 2012, Boston, Massachusetts, USA.

1. INTRODUCTION

Today increasingly large volumes of TV and video are distributed over IP-networks and over the Internet. Many telecom and broadband companies have become TV operators and distribute TV channels using IP multicast in their networks. The TV services also evolve, and are more and more changing towards TV-on-Demand and time-shifted viewing where the users can choose to watch the programs after its scheduled time. Distributing individual TV streams to each viewer requires a lot of bandwidth and server capacity. How to best use caching of popular content closer to the viewers is therefore an important issue to reduce network load.

In this paper we study access patterns in a large TV-on-Demand system over four months. We study user behaviour and program popularity and its impact on caching.

There are several studies of viewing behaviour in IPTV systems where traditional scheduled TV is distributed over IP networks [8], [17], [18]. This include studies of TV channel popularity and channel switching. Our work is different in that we look at TV-on-Demand where the viewers choose programs to watch outside of the TV schedule. The programs are not distributed using multicast but transferred with unicast streams to the viewers.

In this sense our work is closer to studies of content access patterns in traditional Video-on-Demand systems (VoD) [26]. But TV-on-Demand is different from traditional VoD in several ways. The TV-on-Demand service is more diverse. It is a mix of TV program libraries, time-shifted viewing, and rental video. Time-shifted viewing here means that the viewer can choose to watch ongoing scheduled TV-programs from the beginning. The TV schedule gives a large inflow of new programs each day. The programs available also come from a wide range of TV channels. There is a large variation in program types (news, drama, children's programs, movies, etc.) which each can have different access patterns. Many programs, like news and weather forecasts, also have a very short lifespan and are typically only interesting for a few hours.

The two main contributions of this paper are: (1) an investigation of program popularity and access patterns for TV and video on demand in a real network, (2) a trace-based study of caching. We characterize access patterns for different program categories, we show how program popularity changes over time and how this differs between different program types. We then use the request sequence in our data set for trace-driven simulation and study cache hit ratios for different cache sizes, cache replacement policies and population sizes.

Table 1: The data set in figures

	Requests	Clients	Programs
Total (over 125 days)	10294948	307347	89889
Daily median	80174	30232	7523
Daily max	121053	42451	8751
Daily min	56720	22194	6316

Our main results are:

- The popularity (ranking) of rental movies, news, and TV shows changes over time in very different ways. News programs are often only requested for a few hours, movies are popular for months and increase in rank during weekends, TV shows increase in rank when the next episode is shown, children's programs are top ranked in the mornings and early evenings. This means that programs jumps in and out of the top 100 list. It also means that the *type* of content offered is essential for what the access pattern will look like.
- The program popularity conforms with the Pareto principle, or 80-20 rule. There is a small set of programs that account for a large fraction of the requests: the 2% most popular programs get 48% of the requests, and the 20% most popular programs get 84% of the requests.
- The share of requests for the top 100 most popular programs increases during prime time and the change among the top 100 decreases during prime time and during weekends when the demand is the highest.
- The cacheability is very high. The hit ratio with LRU is above 50% when caching 5% of the average daily demand, and the hit ratio increases during prime time when it is needed most.

The rest of the paper is structured as follows: In Section 2 we describe the TV-on-Demand service and introduce the data set. In Section 3 we study access patterns and the daily and hourly change in user interest. In Section 4 we look at the program popularity in more detail, how the access patterns differs between different program categories, and how the popularity of individual programs changes over time. In Section 5 we show what impact the access patterns has on cacheability and cache hit ratios. Related work is in Section 6, we discuss future work in Section 7 and conclude the paper in Section 8.

2. THE DATA SET

We study logs from the TeliaSonera TV-on-Demand service. The program selection is a mix from a wide range of TV channels (news, drama, children's programs, movies, etc.). It is a mix of TV program libraries, time-shifted viewing, and rental video.

The TeliaSonera TV service also includes multicast distribution of traditional scheduled TV. Here we only study logs of on-demand requests but the TV schedule with many ongoing channels gives a constant inflow of new programs that become available for on-demand requests. In our data set, on average 8% of the programs each day have not been requested before.

The data set is a mysql database with logs from RTSP sessions where we for each session have:
<Timestamp, Length, ServerID, ClientID, AssetID>.

The timestamp shows when the session ended and by subtracting the length of the session we get the time when the request arrived. The AssetID identifies what TV program is requested. For each asset, we also have additional out-of-band information about providers and program descriptions that help us categorize the programs into genres.

The data set is summarized in Table 1. It contains TV-on-Demand requests over 125 days between May 12th and September 13th 2011. During this period almost 90000 different programs were requested. The data set includes more than 300000 clients making more than ten million requests.

Figure 1 shows the number of requests, viewers, and programs per day. There are distinct weekly cycles where the

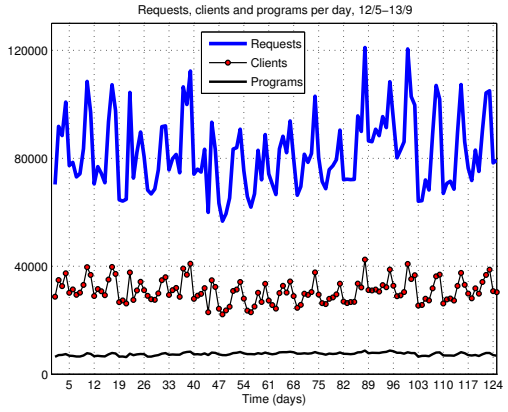


Figure 1: Number of requests, active clients, and distinct programs requested per day over 125 days. The grid shows weeks starting from Mondays.

number of active clients and the number of requests increase a lot during the weekends.

On average more than 30000 clients are active per day often increasing up to 40000 at the weekends. Some viewers are much more active than others and watch more TV programs. Viewers also subscribe to different TV packages and have access to different number of TV channels and program libraries. We can see that 5% of the viewers account for 41% of the requests and 20% of the viewers account for 75% of the requests. Figure 2 shows a log-log-plot of the number of requests per viewer. While many clients only watch a few on-demand programs per month, the most active viewer had more than 137 requests per day on average. Some of these sessions were 5-30 minutes long but many where short, jumping between different on-demand programs.

The clients in the data set are all in the same time zone and in the same geographical region. Later in Section 5, when looking at caching, we will also study smaller subsets of the population. We have one geographically close subset with 23304 clients in the same town. For the smaller populations in the study we randomly chose clients and include into sets of different size up to 10000. We will use the labels *region* (307347), *town* (23304), *rand10000*, *rand1000* etc. for the different populations.

On average 7523 different programs are requested per day. As expected, some programs are much more popular than

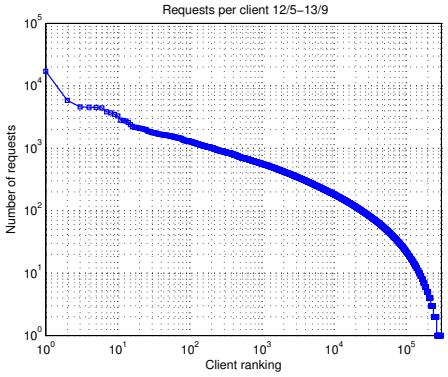


Figure 2: Log-log plot of requests per client over 125 days.

Table 2: The week 19-25/5 in figures

	Requests	Clients	Programs
Total (over 7 days)	585147	105698	14067
Daily median	76931	31542	6813
Hourly median	2626	1723	1190
Hourly max	16037	9019	2987
Hourly min	186	131	128

others. On average the top 10 programs each day get 11% of the requests, the top 100 get 35%, and the top 1000 account for 71% of the requests. We will look at the program popularity in more detail in Sections 3 and 4.

3. ACCESS PATTERNS

3.1 Access pattern over a week

Figure 3 shows the number of requests per hour during one week from Thursday to Wednesday.

We can see here in detail the typical daily and weekly variation in demand. There are large, predictable peaks in demand in the evenings. The number of requests are often four times higher or more during the peak hour compared to the average demand during daytime. As expected, the number of requests are the highest on Friday and Saturday evening. The demand during daytime also increases during weekends.

The number of distinct programs requested per hour follows a similar pattern to that of the demand but the peaks are not as pronounced. The number of different programs requested often doubles in the evenings compared to daytime. In this particular week the median number of programs requested per hour was 1190, the hourly maximum was 2987 programs (Saturday 21:00-22:00) and the minimum was 128 programs (Wednesday 04:00-05:00).

The bar plot in Figure 3 also shows the number of requests for the top 10 and top 100 most popular programs each hour. Figure 4 shows the share of requests per hour that the top 100 most popular programs account for. On an hourly basis the top 100 on average get 50% of the requests. The top 100 obviously have a large part of the traffic during night when not much more than 100 programs are requested. But more interestingly the top 100's share of requests also increase significantly during prime time. The number of different programs requested increases during the evenings

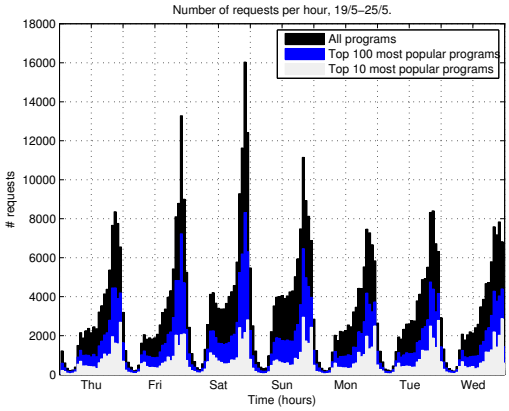


Figure 3: Number of requests per hour during a week. The grid points out the hours between 11:00-12:00 and 23:00-00:00 each day.

and so a hundred programs constitute a smaller share of the requested program volume. But even so the top 100's share of requests increase significantly.

3.2 Daily and hourly change in user interest

Which programs are most popular change over time. On average 6 of the top 10 programs are replaced each day. Figure 5a shows the daily change among the top 100 and top 1000 most requested programs. Here we also see the daily change among all requested programs. On average 73% of the requested programs are the same as yesterday. On average 56% of the top 100 and 42% of the top 1000 is different from the day before.

We notice in Figure 5a a weekly pattern with less change in top 100 during weekends (from Fridays to Saturdays and Saturdays to Sundays). This suggests that what the most

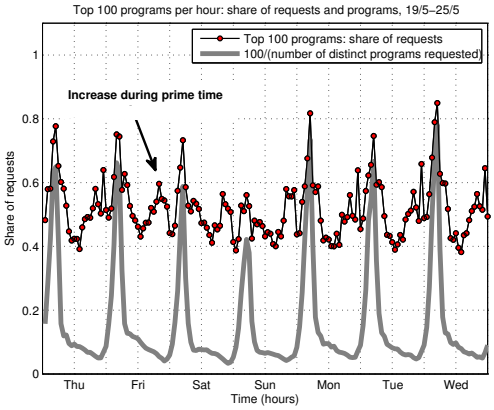
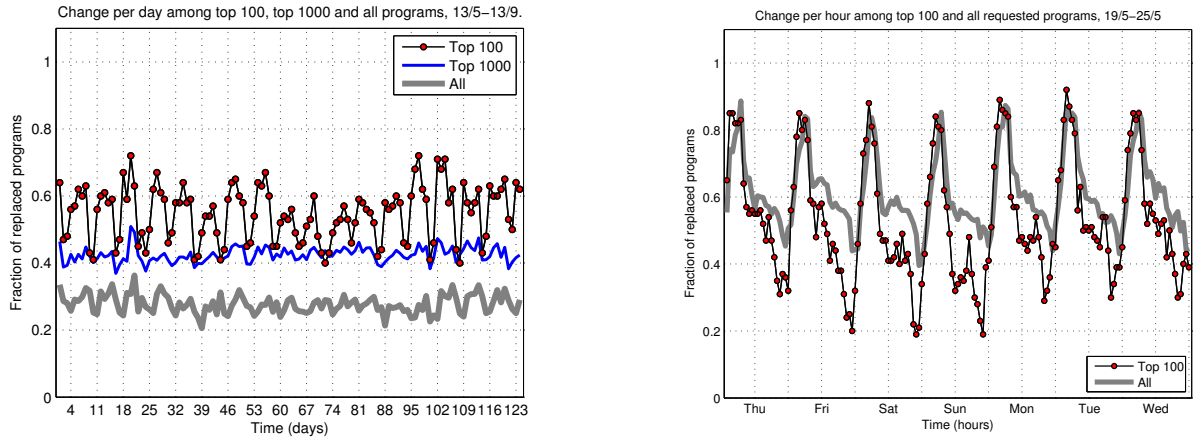


Figure 4: The figure shows the share of requests per hour that the top 100 most popular programs account for. It also shows the share of programs requested that a hundred programs comprises. The top 100's share of requests increases during night when few programs are requested but more interestingly it also increases during prime time when the demand is the highest. The grid points out the hours between 11:00-12:00 and 23:00-00:00 each day.



(a) (b)

Figure 5: (a) Daily change among the top 100 and top 1000 most requested programs. (b) Hourly change among the top 100 most requested programs. The figures also shows the fraction of all programs that was not requested the day and hour before.

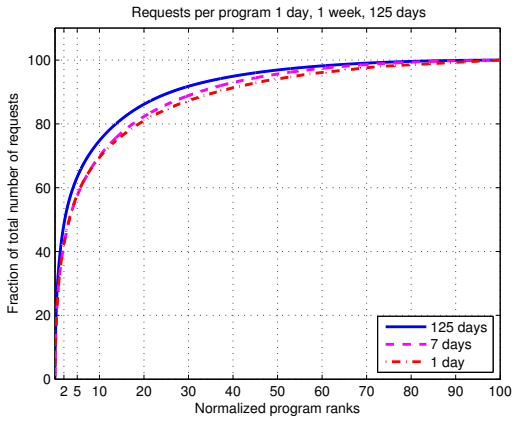


Figure 6: Cumulative distribution of requests to programs (1 day, 7 days and 125 days).

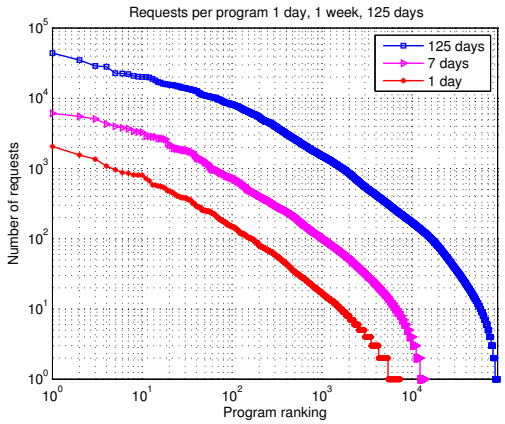


Figure 7: Log-log plot of requests per program (1 day, 7 days, 125 days).

popular items will be is more predictable during weekends when the demand also is the highest. This is even clearer on an hourly basis.

Figure 5b shows the hourly change among the top 100 most requested programs. On average 51 out of the 100 most requested programs are the same as the hour before. But the amount of change in the top 100 varies from hour to hour in a distinct daily cycle. During night up to 92% of the top 100 programs are changed from one hour to the next. While during prime time (19:00 to 23:00) the top 100-list becomes much more stable with down to 19% change among the top 100 programs.

4. PROGRAM POPULARITY

There is a small set of popular programs that account for a very large part of the requests. The Pareto principle, or the 80-20 rule, is often referred to when describing video popularity and the concentration of user interest towards a few popular programs [7], [26]. The users spread of requests across programs in the TV-on-Demand system conforms with this principle. The 20 % most popular programs account for more than 80% of the requests.

We calculated the number of requests for each program, sorted them in order of popularity, and plotted the cumulative distribution function shown in Figure 6. Here we can see the number of requests per program as a CDF-plot for 1 day, 1 week and for the entire 125-day period.

If we consider the entire 125-day period, then the 2% most popular programs account for 48% of the requests, the 10% most popular programs account for 74% of the requests, and the 20% most popular programs receive 84% of the requests. The figures are similar on daily and weekly basis as well.

This skewness in popularity for TV-on-Demand is somewhere in between what has been described in the literature for user-generated content and traditional Video-on-Demand systems. For Youtube traffic, investigated by Cha et al. [7], 10% of the videos accounted for 80% of the requests. In the chinese PowerInfo Video-on-Demand system described by Yu et al. [26], 10% of the videos accounted for 60% of the accesses. TV-on-Demand systems are more dy-

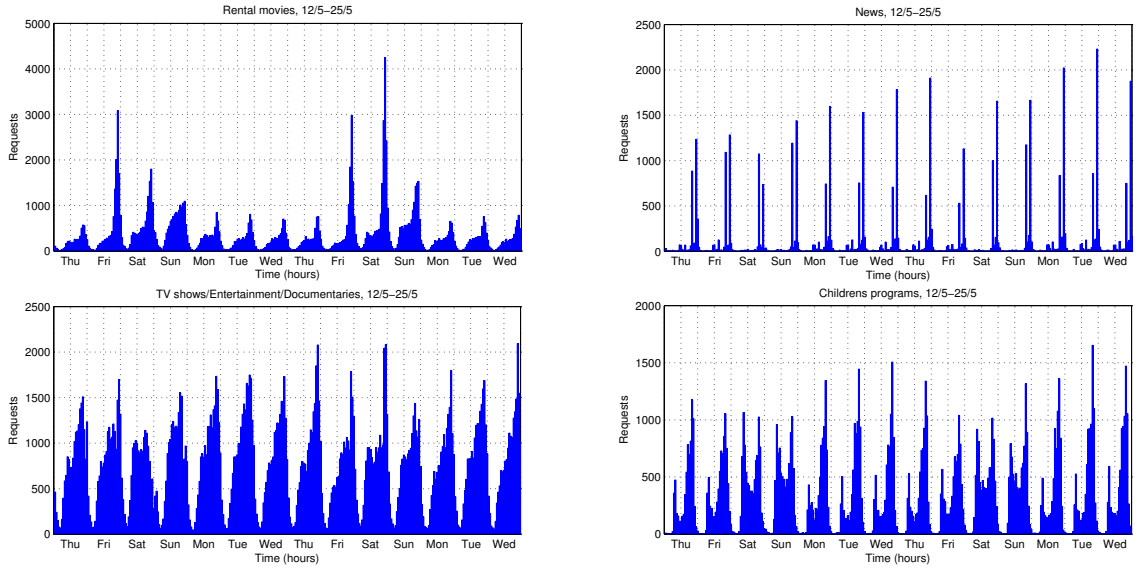


Figure 8: Access patterns per program category: rental movies, TV news, TV drama, and children's programs. Requests per hour over two weeks. The grid points out the hours between 11:00-12:00 and 23:00-00:00 each day.

namic than traditional VoD systems with a large daily inflow of new content. As we will see in Section 4.2, there are programs that are popular for several weeks and accumulate a lot of requests, but there are also many programs that have a very short life-span and are only requested for a few hours.

Figure 7 shows the number of requests per program as a log-log plot. It shows the number of times that a program has been accessed versus the ranking of the program in the data set.

There are a large number of research papers that deal with the popularity distribution of web pages and video. Much of the debate concerns whether the distribution of requests is Zipf-like or not [6, 7, 9, 13, 26]. Here we do not try to fit the curve to a specific probability distribution. However, we note that the curve does not follow a straight line on the log-log scale. This implies that the distribution of TV-on-Demand requests does not follow a Zipf-like distribution.

The 80-20 rule, and the concentration of requests to a small set of programs is important for caching. This is independent of what exact probability distribution best describes the access frequency. In Section 5 we perform trace-driven simulation, and directly use the sequence of requests to investigate the impact on caching.

4.1 Access patterns per program category

Different categories of programs have different access patterns. Figure 8 shows the number of requests per hour over two weeks for programs in four different categories: rental video, TV news, drama and children's programs. The figure demonstrates some clear and expected differences in access patterns.

The top left figure shows the access pattern for rental movies. These are movies that a viewer can pay to access for 24 hours. We can see that movie rentals are concentrated over weekends with large peaks in demand during Friday and Saturday evenings.

For TV news the traditional TV schedule determines to a large extent also when the program is requested on demand. The TV news is scheduled daily at 19:00 and 22:00. At the same time it becomes available for time-shifted viewing and we can see that most requests are close to these times.

For the other two categories we note that the TV reality and drama shows are watched during the daytime to a larger degree than other programs. We also see that the children's programs have peaks in demand in the mornings and early evenings. This is especially true for weekends.

4.2 Access patterns for individual programs: how program popularity changes over time

The popularity of a program changes with time and the demand pattern varies depending on the program type. Figure 9 shows the number of requests per day for 20 different programs over 125 days.

The top figures show the most requested rental movies and TV news programs in the data set. For each movie we can see a slow decline in popularity over time. The movies are requested many weeks after their premiere. There is also a clear weekly pattern with peaks in demand at the weekends. For TV news programs the access patterns are very different compared to movies. A news program is mostly requested the first evening and then quickly becomes outdated and loses its popularity when available on demand.

The figures at the bottom show the access pattern for five episodes of a daily TV reality show and five episodes of a weekly home improvement show. We can see that the request patterns for different episodes of the same show are surprisingly similar. For the reality show we can also see that after the initial peak in demand the program popularity quickly decline with time. The programs are requested daily also the following three months but there are often only a few requests per day.

4.2.1 The life of a rental movie

Figure 10a shows the number of requests per day and the rank for a comedy-drama rental movie. We can see the decline in popularity over 16 weeks from the premiere. The

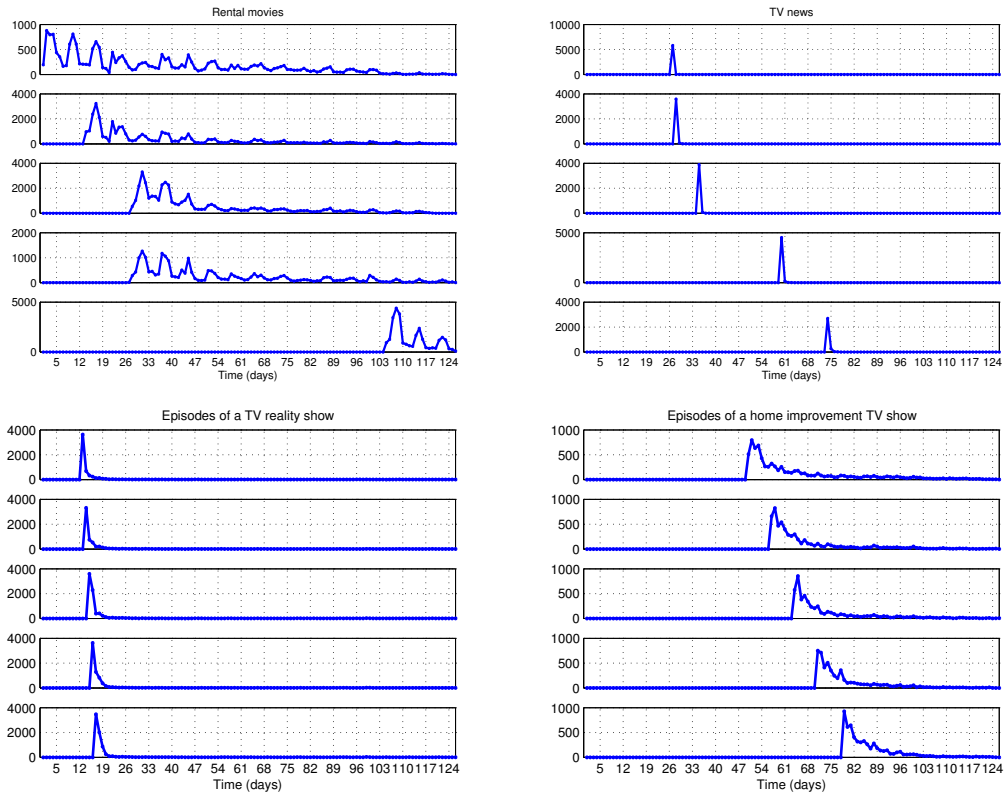


Figure 9: Requests per day for 5 movies, TV news programs and episodes of a reality show and a home improvement show.

figure also demonstrates the typical peaks in popularity for a rental movie during weekends where the number of requests increases and the program increase in rank.

Figure 10b shows the change in rank in more detail among the top 100 most popular programs each day. We can see that the movie jumps in and out of the top 10 and top 100 lists a number of times. This has implications for the choice of caching strategy. It is essential to have the right programs in the cache at Friday and Saturday evenings when the total demand is the highest. If the replacement policy acts on popularity over a short time window the program might be evicted when popularity temporarily goes down during weekdays and the program will not immediately be available in the cache when the demand increases again next weekend.

Figure 10b also compare the rank for our movie among all programs with the rank among only rental movies. The movie is the number one most popular movie for 14 days in a row and it stays in the top 10 for one month and in the top 20 for two months.

Figure 11 shows the rank and the number of requests per hour during the first week that the movie is available. The movie quickly climbs in rank and becomes one of the most popular programs. It is in the top ten during the evenings but the rank of the program sometimes drops during the daytime and during night. There are large, predictable daily variations in demand with peaks in the evenings. The number of requests increases significantly during Friday and Saturday.

4.2.2 *The life of a TV news program*

TV news programs have a very short lifespan compared to movies. Figure 12 shows the rank and number of requests per hour for a news program that was sent live at 19:00.

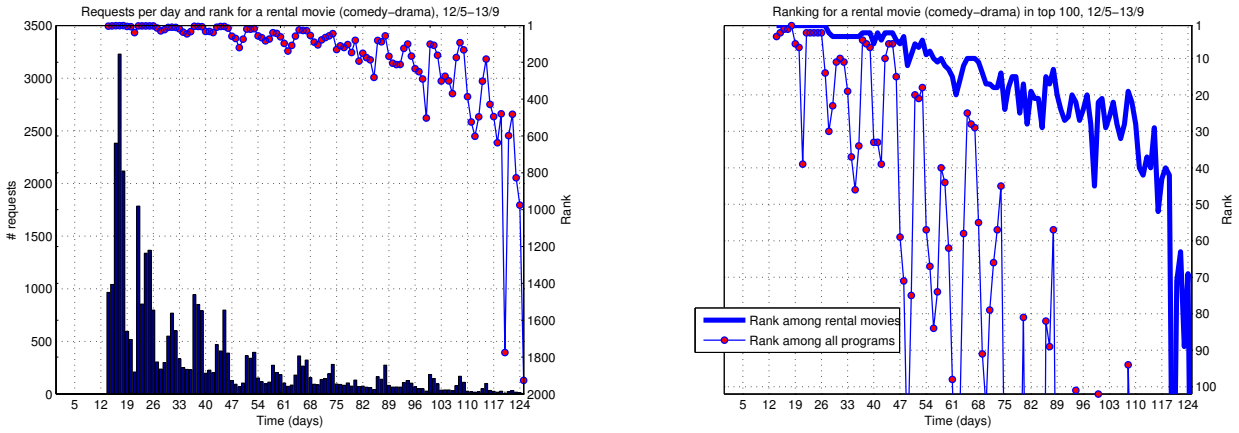
Most of the requests are the first hour when the program becomes available. The news program immediately becomes the most requested program that hour and number one on the ranking. The popularity then quickly declines. There are almost no requests at all for a news program after the first day. The access pattern is very different compared to what we see in Figure 11 for the simultaneously available movie.

4.2.3 *TV series and children's programs – periodic increase in popularity*

The interest in a TV program usually decreases with time. But more often than not the popularity of a program can also increase temporarily or periodically. We saw in the previous sections that the number of requests for a program varies during the day and the week. We also saw for rental movies that the ranking increased during weekends.

Many TV shows are part of a series of programs. When the next episode is sent there is often also renewed interest for old episodes available on-demand. Figure 13 shows an example with the rank and number of started sessions per day for an episode of a weekly home improvement TV show. We can see that the program increase in rank every Thursday when the series is shown on the traditional scheduled TV.

Figure 14 shows the number of requests per day and the



(a) (b)

Figure 10: (a) Requests per day and ranking for a rental movie (comedy-drama). The bar graph shows requests per day with the scale on the Y-axis shown to the left. The plotted line shows the ranking of the program among all other programs requested that day. The scale of the ranking is shown on the Y-axis to the right. The grid points out weeks starting on Mondays. (b) Detailed look at the rank among all programs (top 100) and among rental movies.

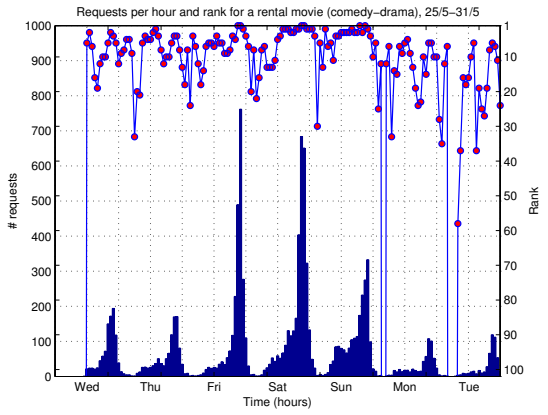


Figure 11: Requests per hour and ranking for a rental movie (comedy-drama). The grid points out the hours between 11:00-12:00 and 23:00-00:00.

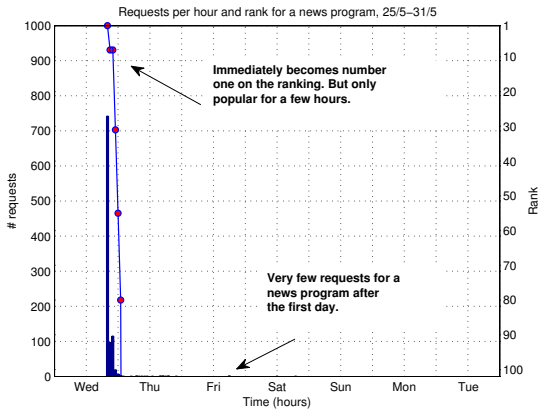


Figure 12: Requests per hour and ranking for a TV news program.

ranking for a cartoon. After the initial peak in interest the popularity decreases and remains at a steady level over the month when the program is available. This is different if we look at the ranking on an hourly basis. Figure 15 shows the ranking of the program per hour during the first week. The pattern is the same for the next three weeks as well. The program varies in popularity. It goes in and out of the top 100 list, often twice a day.

The number of requests for children's programs increases in the mornings and in the early evenings. This is a daily recurring pattern. Also, at these times of the day there is little demand for other TV-programs so few requests are needed to get into the top ranking.

5. IMPACT ON CACHING

In previous sections we have seen many aspects of the access patterns in a TV-on-Demand system. In this section we study the impact on caching. We examine the proportion of requests that are not first-time requests for a program and therefore potentially could be served from a cache. We look at this for different population sizes and time periods.

We then use trace-driven simulation to investigate the cache friendliness of the workload with a limited cache size and the classic LRU and LFU cache replacement policies. We run the sequence of requests in our data set through caches of different size and look at the resulting cache hit ratios.

5.1 Cacheability

For on-demand caching, the first request for a program needs to go to the central server. But if we imagine an unlimited cache size then all other requests could potentially be served from the local cache. It is therefore interesting to examine the proportion of requests that are not first-time requests. We here call that *cacheability*.

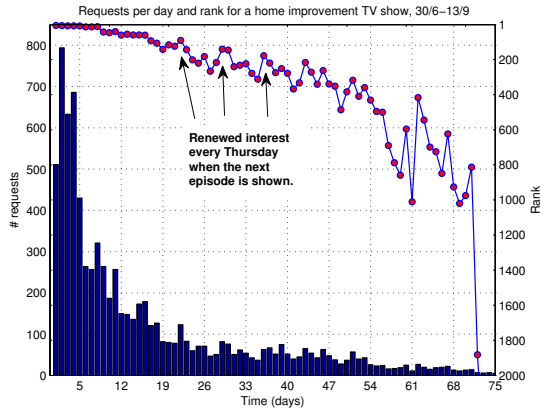


Figure 13: Requests per day and ranking for an episode of a weekly home improvement TV show. The grid points out weeks starting on Mondays.

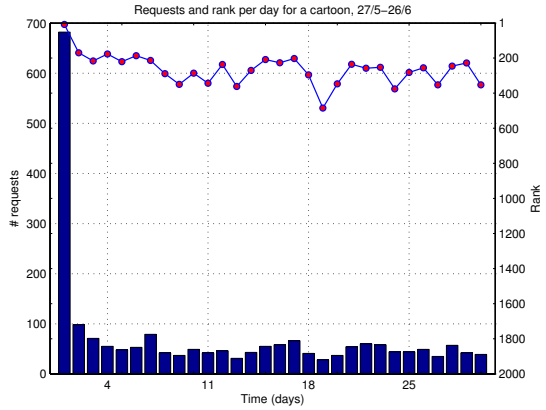


Figure 14: Requests per day and ranking for a cartoon.

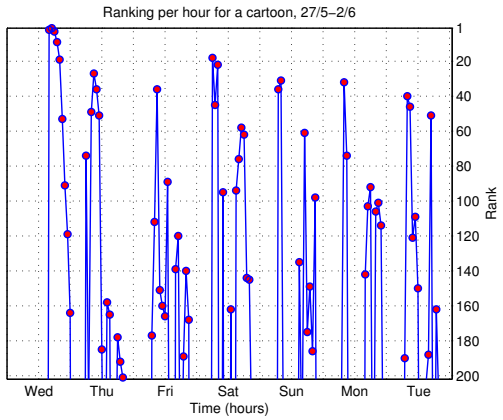


Figure 15: Ranking per hour for a cartoon. The grid points out the hours between 11:00-12:00 and 23:00-00:00 each day. The program jumps up and down in popularity. It always climbs to top 40 in the mornings and are often in top 100 in the early evenings.

We follow the definition of *cacheability* used by Ager et al. [2]. But our data set do not include information about program size so here we only consider requests. Cacheability is then the share of requests that are not first-time requests. If k_i is the total number of requests for a program k then the cacheability is $\sum_{i=1}^{n}(k_i - 1)/\sum_{i=1}^{n}(k_i)$, where n is the number of programs.

The share of first time requests is very low in the TV-on-Demand system if we consider all clients over a long period of time. The cacheability over 125 days is: 99.13%.

In Figure 16 we also look at the cacheability per day and per hour and for populations of different size. The calculation of cacheability starts from the beginning of each time interval. It is not considering what have been requested the hour or day before. For all clients in the region during the week in Figure 16, the median cacheability per hour is 59%. However, there are large daily variations. During night many programs are requested only once and the cacheability is low. During Friday and Saturday evenings the cacheability is above 80%.

Figure 17 shows examples of cacheability over 125 days for smaller populations. For very small populations the probability that a viewer will choose a program that nobody else in the group has requested before is high. So the share of first-time requests is high and the cacheability is low. But we see that already for groups of 1000 viewers the cacheability is above 60%. We calculated the cacheability for five different groups of 1000 viewers. The median was 63.9% and the group with lowest result had a cacheability of 61.7%.

5.2 Limited cache size

We saw in the previous section that the cacheabilty in the TV-on-Demand system is very high. But in practice there is a limited cache size. In order to investigate the cache friendliness of the TV-on-Demand workload we use trace-driven simulation. We run the sequence of requests in the data set through caches of different size and study the cache hit ratios for three classic caching policies:

Least Recently Used (LRU): with the LRU strategy we delete from the cache the program that has not been requested for the longest time.

Least Frequently Used (LFU): with LFU we discard the program that is requested least often. This is done by keeping track of the hit ratio for all programs currently in the cache (in cache LFU).

Clairvoyant: we also implement a clairvoyant strategy with the ability to look into the future and delete the program that will not be needed for the longest time. This is used for comparison to obtain an upper limit on the cache hit ratio. It is implemented by going through the traces twice. First, for each request of a program we look up and determine when the program will be requested next. This is then used in the simulation to determine what program should stay in the cache.

Figure 18 shows cache hit ratios for the LRU, LFU and Clairvoyant replacement policies for increasing cache sizes. The hit ratios are calculated over 3 days. The size of the programs are not taken into account. We calculate request (or program) hit rate and not the byte hit rate. The x-axis shows cache size in number of programs. The median number of distinct programs requested per day is 7523. To put the hit ratio and cache size in relation to the daily demand we therefore look specifically at cache sizes of 376

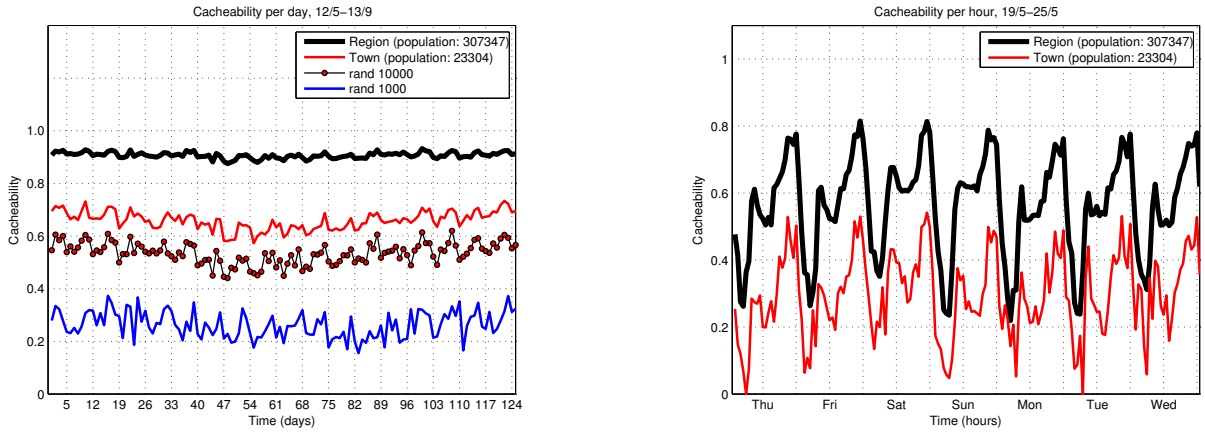


Figure 16: Cacheability per day and per hour. Comparison between different population sizes.

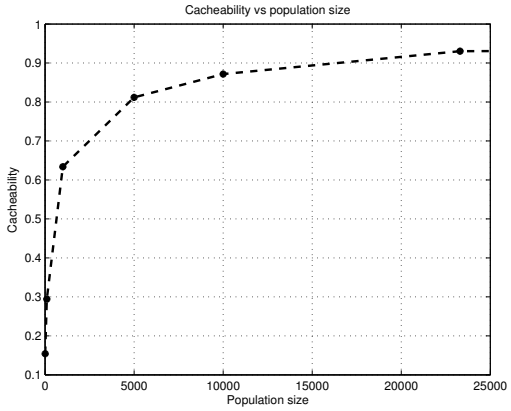


Figure 17: Example of cacheability versus population size.

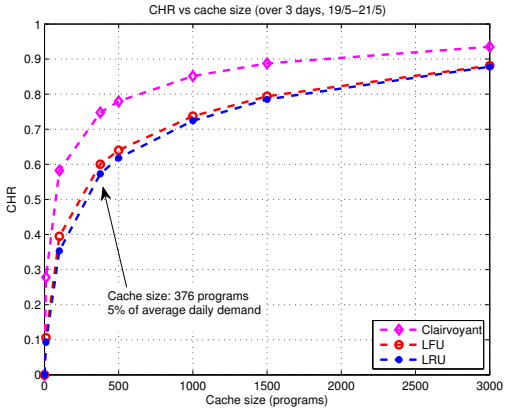


Figure 18: Cache hit ratio versus cache size, requests from all clients over 3 days. Comparison of the LRU, LFU and Clairvoyant replacement policies.

programs, which is 5% of average daily demand. We can see that caching 5% of the daily demand gives a hit ratio of 57% for LRU, 60% for LFU and 75% for the Clairvoyant replacement policy.

In Figure 18 we include the requests from all viewers. In Figure 19 we investigate the impact of population size. Here we use the LRU replacement policy and compare the cache hit ratios for populations of different size. For a cache size of 376 programs (5% of the daily demand) the town subset of 23304 clients get a hit ratio of 51%. This is close to the result for the full set of clients. For the small population with 1000 clients we get a hit ratio of 43%. The cacheability for this particular population of 1000 viewers was 0.64 over 125 days and we see the curve approaching that value at a cache size of 3000 programs.

The cache hit ratio also varies over time. Figure 20 shows hit ratio per hour for all viewers, the LRU replacement policy and a cache size of 376 programs. The hit ratio was calculated over 17 weeks and the figure shows the median (and max and min) value for each hour of the week. We can see that the cache hit ratio varies over the day and it increases when it is needed as most. During prime time, when there are the most requests, the hit ratio is over 60%.

From the results presented above we highlight three observations:

- The cacheability and the potential for caching is very high.
- The hit ratio with a simple LRU replacement policy is above 50% when caching 5% of the average daily demand.
- The hit ratio increases during prime time when it is needed most. This is consistent with the observations in Section 3 that the share of requests for the most popular programs increases during prime time.

We have here looked at the cache friendliness of the TV-on-demand workload in terms of cacheability and cache hit ratios for the basic LRU and LFU replacement policies. In Section 7 on future work we discuss how our observations about access patterns and program popularity can potentially be used to design a more informed caching strategy.

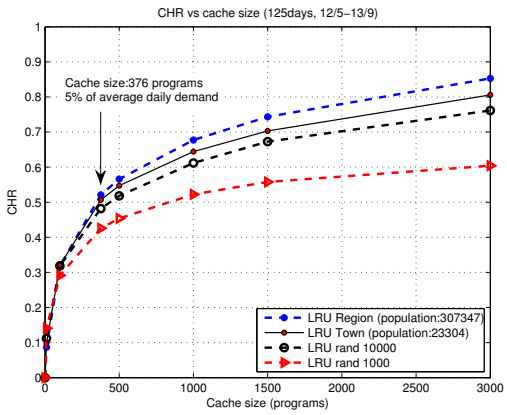


Figure 19: Cache hit ratio versus cache size. Comparison of cache hit ratios for populations of different size.

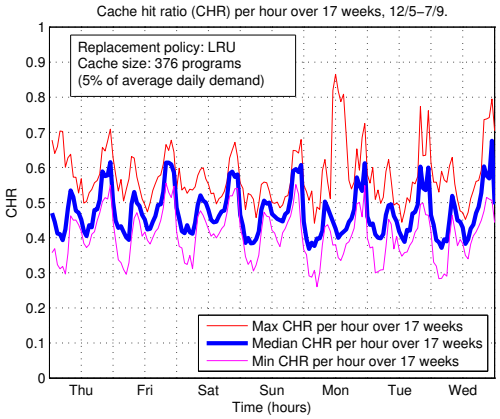


Figure 20: Cache hit ratio per hour over 17 weeks

6. RELATED WORK

There are several studies of viewing behaviour in IPTV systems where traditional scheduled TV is distributed over IP networks. Cha et al. [8] study viewing behaviour including channel popularity and channel switching in an operational IPTV network. Qiu et al. model TV channel popularity [18] and user activities [17] in a large IPTV system. Our work is different in that we look at TV-on-Demand where the viewers choose programs to watch outside of the TV schedule. In this sense our work is closer to studies of traditional VoD systems.

Yu et al. [26] present a large measurement study of the chinese PowerInfo Video-on-Demand system. This work is similar to ours in that they investigate many aspects of user behaviour and content access patterns. The PowerInfo system is a traditional VoD system. The videos in the library are old TV shows and movies and there are usually only a few new movies introduced to the system per day. This is different from the TV-on-Demand system that we study where there is a large inflow of new programs from the TV-schedule, time-shifted viewing, and programs with a very short life-span. Our work is also different in other aspects in that we investigate how the access pattern depend on genre, we study cacheability and use trace-based simulation to investigate what impact the access patterns have on caching.

There are many other interesting studies of VoD systems and video popularity. Griwodz et al. [12] model long-term popularity of videos on the time scale of days based on VHS rental statistics. Lou et al. [16] give examples of the popularity evolution of video files from a Chinese television station. Tang et al. [19] analyse and model many aspects of media server access. Avramova et al. [3] model the popularity evolution of TV-on-demand and video traces. Dan and Carlsson [9] measure and analyse BitTorrent content popularity. Guo et al. [13] study the probability distributions of Internet media workloads and analyse caching using a mathematical model. Yin et al. [25] study live VoD workloads from the 2008 Beijing Olympics. There are also many studies of Youtube and user generated videos [4, 7, 10, 14].

Gopalakrishnan et al. [11] study user behaviour in a large IPTV system. This is similar to our work but their focus is on modeling the interactive user behaviour in an IPTV environment, including how users fast-forward, pause and rewind to control their viewing.

In this paper we also investigate cacheability and we look at the potential for caching in a TV-on-Demand system. Caching has been widely studied for web content and video [6, 15, 22, 24]. More recently, Ager et al. [2] studied the cacheability for HTTP- and P2P-based applications. There are also several studies of caching strategies in IPTV on-demand systems [1, 5, 20, 21, 23], but these studies use analytical models and simulations whereas we present a trace-based study from a real TV-on-Demand system.

7. FUTURE WORK

In this paper we have studied many aspects of the access patterns in a TV-on-Demand system. We have looked at the cache friendliness of the workload in terms of cacheability and hit ratios for basic replacement policies. For future work we hope that our observations can be used as a basis for developing better caching strategies for TV-on-Demand systems.

When studying the cache friendliness of the request stream in Section 5 we used the basic LRU and LFU cache replacement policies. With these the last requested program is always cached and the choice of what to evict from the cache is between the least recently and the least frequently requested program. A more advanced system could use more knowledge about access patterns and program popularity to decide what program to put in the cache and what program to evict.

One such strategy could be to keep track of all programs in the system, also those that are not currently in the cache. One could monitor the popularity by counting requests, let the programs age over time and for each program keep a value that describes the probability that it will be requested. There are several observations in this paper that can be useful for such an informed caching strategy:

Give preference to new programs
With time-shifted TV ongoing scheduled programs immediately get a lot of requests. Some programs, like TV-news, also have a very short life-span. The value of a program should not have to be built up by requests over a long time.

Categorize programs by genre to predict change in popularity over time

We saw in Section 4 that the access pattern very much depends on the type of program. A news program that is top-ranked the first evening age quickly and have a very low probability for being requested the next evening. A rental movie however is popular for months and increase in rank during weekends. By categorizing programs by genre the probability for future requests can be predicted. The categorization of programs can also be more detailed. The request patterns for different episodes of the same show are often very similar as we saw in Figure 9, Section 4.2. For a new episode of a show it is a reasonable assumption that the popularity of the program will change over time in a way similar to that of the previous episodes.

Focus on prime time

The value of a program should reflect the probability that it will be requested during prime time. There are large peaks in demand in the evenings and at the weekends that need to be handled. If caching is used to limit the maximum link load then it is essential to have the right programs in the cache on Friday and Saturday evenings. There are program like cartoons that are top-ranked in the mornings and early evenings that probably should not be in the cache.

The observations and the predictions outlined above can be used to optimise the caching performance. However, the basic monitoring of request frequency is still needed as a basis, and to handle unexpected changes and sudden peaks in program demand for instance due to large news events.

8. CONCLUSIONS

We have analysed the access patterns in a large TV-on-Demand system and studied the potential for caching.

Our contribution in this paper is three-fold. As a first-order result, we provide reconfirmation of known observations with an independent dataset. We demonstrate that there is a small set of programs that account for a large part of the requests. The program popularity conforms with the Pareto principle, or 80-20 rule. The demand follows a diurnal and weekly pattern, and there are large peaks in demand on Friday and Saturday evenings that need to be handled.

Second, we provide systematic evidence of TV-on-Demand access pattern characteristics that are intuitive yet unconfirmed in the literature. We show that news programs have a very short lifespan and are often only requested for a few hours, children's programs are top ranked in the mornings and early evenings, and movie rentals are concentrated over weekends.

Finally, we also provide novel insights into access patterns that have not been reported previously to the best of our knowledge. We show how the popularity of TV-on-Demand programs changes over time. We see that the access pattern in a TV-on-Demand system very much depend on what type of content it offers. Furthermore, we find that the share of requests for the top most popular programs grows during prime time, and the change rate among them decreases. The cacheability is very high and the cache hit ratio increases during prime time when it is needed most.

We believe that these observations and findings can guide the design of future systems for TV-on-Demand infrastructures.

9. ACKNOWLEDGMENTS

This work has been performed within the SICS Center for Networked Systems funded by VINNOVA, KKS, SSF, ABB, Ericsson, Saab SDS, TeliaSonera, T2Data, Vendolocus and Peerialism.

10. REFERENCES

[1] H. Abrahamsson and M. Björkman. Simulation of IPTV caching strategies. In *Proceedings of SPECTS'10*, Ottawa, Canada, 2010.

[2] B. Ager, F. Schneider, J. Kim, and A. Feldmann. Revisiting Cacheability in Times of User Generated Content. In *Proceedings of 13th IEEE Global Internet Symposium*, San Diego, CA, USA, March 2010.

[3] Z. Avramova, S. Wittevrongel, H. Bruneel, and D. Vleeschauwer. Analysis and Modeling of Video Popularity Evolution in Various Online Video Content Systems: Power Law versus Exponential Decay. In *Proceedings of International Conference on Evolving Internet*, 2009.

[4] Y. Borghol, S. Mitra, S. Ardon, N. Carlsson, D. Eager, and A. Mahanti. Characterizing and Modeling Popularity of User-generated Videos. In *Proceedings of IFIP International Symposium on Computer Performance, Modeling, Measurements and Evaluation (PERFORMANCE)*, 2011.

[5] S. Borst, V. Gupta, and A. Walid. Distributed caching algorithms for content distribution networks. In *Proceedings of INFOCOM'10*, San Diego, USA, 2010.

[6] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker. Web Caching and Zipf-like Distributions: Evidence and Implications. In *Proceedings of IEEE INFOCOM*, 1999.

[7] M. Cha, H. Kwak, P. Rodriguez, Y. Ahn, and S. Moon. Analyzing the Video Popularity Characteristics of Large-Scale User Generated Content Systems. *IEEE Transactions on Networking*, 17:1357–1370, October 2009.

[8] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain. Watching Television Over an IP Network. In *Proceedings of Internet Measurement Conference (IMC'08)*, Greece, 2008.

[9] G. Dan and N. Carlsson. Power-law Revisited: A Large Scale Measurement Study of P2P Content Popularity. In *Proceedings of IPTPS'10*, San Jose, USA, April 2010.

[10] P. Gill, M. Arlitt, Z. Li, and A. Mahanti. YouTube Traffic Characterization: A view From the Edge. In *Proceedings of ACM SIGCOMM Internet Measurement Conference*, San Diego, USA, October 2007.

[11] V. Gopalakrishnan, R. Jana, K. Ramakrishnan, D. Swayne, and V. Vaishampayan. Understanding Couch Potatoes: Measurement and Modeling of Interactive Usage of IPTV at large scale. In *Proceedings of Internet Measurement Conference (IMC'11)*, 2011.

[12] C. Griwodz, M. Bär, and L. Wolf. Long-term Movie Popularity Models in Video-on-Demand Systems or The Life of an on-Demand Movie. In *Proceedings of ACM Multimedia'97*, Seattle, USA, 1997.

[13] L. Guo, E. Tan, S. Chen, Z. Xiao, and X. Zhang. The stretched exponential distribution of internet media access patterns. In *Proceedings of the twenty-seventh*

ACM symposium on Principles of distributed computing (PODC'08), New York, USA, 2008.

[14] X. Kang, H. Zhang, G. Jiang, H. Chen, X. Meng, and K. Yoshihira. Measurement, Modeling, and Analysis of Internet Video Sharing Site Workload: A Case Study. In *Proceedings of ICWS'08*, 2008.

[15] J. Liu and J. Xu. Proxy Caching for Media Streaming Over the Internet. *IEEE Communications Magazine*, 42:88–94, 2004.

[16] J. Lou, Y. Tang, M. Zhang, and S. Yang. Characterizing User Behavior Model to Evaluate Hard Cache in Peer-to-Peer Based Video-on-demand Service. In *Proceedings of MMM'07*, pages 125–134, 2007.

[17] T. Qiu, Z. Ge, S. Lee, J. Wang, J. Xu, and Q. Zhao. Modeling User Activities in a Large IPTV System. In *Proceedings of Internet Measurement Conference (IMC'09)*, USA, 2009.

[18] T. Qiu, Z. Ge, S. Lee, J. Wang, Q. Zhao, and J. Xu. Modeling Channel Popularity Dynamics in a Large IPTV System. In *Proceedings of SIGMETRICS*, pages 275–286, Seattle, USA, June 2009.

[19] W. Tang, Y. Fu, L. Cherkasova, and A. Vahdat. Modeling and Generating Realistic Streaming Media Server Workloads. *Computer Networks*, 51:336–356, 2007.

[20] D. D. Vleeschauwer, Z. Avramova, S. Wittevrongel, and H. Brueel. Transport Capacity for a Catch-up Television Service. In *Proceedings of EuroITV'09*, pages 161–170, Leuven, Belgium, June 2009.

[21] D. D. Vleeschauwer and K. Laevens. Performance of caching algorithms for IPTV on-demand services. *IEEE Transactions on broadcasting*, 55:491 – 501, 2009.

[22] J. Wang. A Survey of Web Caching Schemes for the Internet. *ACM SIGCOMM Computer Communication Review*, 29:36–46, 1999.

[23] T. Wauters, W. V. de Meerssche, F. D. Turck, B. Dhoedt, P. Demeester, T. V. Caenegem, and E. Six. Co-operative Proxy Caching Algorithms for Time-Shifted IPTV Services. In *Proceedings of 32nd EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA)*, pages 379–386, Dubrovnik, Croatia, September 2006.

[24] A. Wolman, G. M. Voelker, N. Sharma, N. Cardwell, A. Karlin, and H. M. Levy. On the scale and performance of cooperative Web proxy caching. In *Proceedings of the 17th ACM Symposium on Operating Systems Principles (SOSP '99)*, 1999.

[25] H. Yin, X. Liu, F. Qiu, N. Xia, C. Lin, H. Zhang, V. Sekar, and G. Min. Inside the Bird's Nest: Measurements of Large-Scale Live VoD from the 2008 Olympics. In *Proceedings of Internet Measurement Conference (IMC'09)*, USA, 2009.

[26] H. Yu, D. Zheng, B. Zhao, and W. Zheng. Understanding User Behavior in Large-Scale Video-on-Demand Systems. In *Proceedings of EuroSys2006*, pages 333–344, Leuven, Belgium, 2006.