

Minimizing Energy Dissipation in Content Distribution Networks using Dynamic Power Management

Tom Bostoen, Jeff Napper, and Sape Mullender

Bell Labs

Alcatel-Lucent

Antwerp, Belgium

{Tom.Bostoen, Jeff.Napper, Sape.Mullender}@alcatel-lucent.com

Yolande Berbers

Department of Computer Science

Katholieke Universiteit Leuven

Leuven, Belgium

Yolande.Berbers@cs.kuleuven.be

Abstract—The growing end-user demand for video services with superior quality on laptops, tablets, and smartphones spurs the deployment of telco content distribution networks (CDNs). Such CDNs provide scalable and bandwidth-efficient video delivery thanks to disk-packed cache servers deployed in the telco's data centers near the clients. However, a sustainable growth of these CDNs may be hindered by their lack of energy proportionality. In this paper we propose to apply dynamic power management (DPM) to the CDN's cache servers and their disks to increase the CDN's energy efficiency. We evaluate DPM using a CDN energy simulator driven by HTTP adaptive-streaming workload traces recorded by an operational CDN delivering IPTV to mobile devices. Even for a minimally-provisioned CDN, we observe a reduction of the energy dissipation by approximately 30 % thanks to large cyclic load fluctuations characteristic of IPTV delivery.

Keywords—IPTV; content distribution network; HTTP adaptive streaming; cache server; energy efficiency; disk drive; power reduction

I. INTRODUCTION

Growing end user demand for premium video content on mobile devices represents an opportunity for Internet service providers (ISPs) to extend their IPTV offering with multi-screen video services. Paying consumers of premium video content expect a high-quality viewing experience, which ISPs can provide by delivering IPTV over a content distribution network (CDN) deployed in their own regional network. Because such a CDN is typically owned by the ISP, it is qualified as a *telco* CDN. This type of CDN is composed of cache servers located in telco data centers in the proximity of the end users. The proliferation of such new cache servers will contribute to the continuing increase of data-center power consumption. These servers consume a lot of power in part because they are typically packed with hard disk drives, which require mechanical movement for their operation. Between 2005 and 2010, the power consumed worldwide in data centers increased by more than 50 % [1]. In 2010, the total amount of energy used in data centers represented between 1.1 and 1.5 % of the global electricity use. Because the cost of electricity dominates the operating cost of data centers, data-center operators are looking for

techniques to reduce the energy consumption.

In this paper, we target energy savings in content distribution networks by applying dynamic power management (DPM) to the CDN's cache servers and their disks. According to the exhaustive survey of power-reduction techniques for data-center storage systems that we made analyzing over a hundred high-quality papers in this domain, DPM is the basis for most of these techniques because of its high potential of saving energy [2]. However, DPM was, to our knowledge, never applied to a storage system distributed over multiple data centers such as a CDN. To save energy, DPM powers down system components or scales down their performance when they are idle or underutilized. We consider turning off cache servers and their disks under light load. However, in every data center we keep sufficient caches powered on such that clients can always be served from the closest data center to guarantee the superior quality of the viewing experience, which is considered more important than saving energy. Under this constraint, DPM doesn't alter the workload for any of the data centers; there is no impact on the geographical load balancing. Therefore, without loss of generality, we limit our analysis to one of the CDN's data centers. The number of active caches s in the data center under consideration combined with the number of active disks d_i for every active cache i determines what we call the CDN *power state*. Such a state p is fully identified by the tuple (d_1, d_2, \dots, d_s) . This identification implies that CDN power states with different caches and/or disks active are identical as long as the number of active caches and the number of active disks per active cache are the same.

We target a near-optimal offline algorithm that selects per time interval the CDN power state that minimizes the CDN's energy consumption subject to two performance constraints: (1) the selected power state has to allow the CDN to serve every client request and (2) the cache hit rate needs to be sufficiently high such that the data rate from the origin to the caches doesn't exceed a threshold agreed between CDN and content provider. The first constraint ensures that no requests are dropped while the second prevents trivial solutions with no active disks. An additional objective is to evaluate the

search algorithm based on a simulation of the CDN's energy consumption using real IPTV workload traces.

The main challenges lie in the potentially huge number of CDN power states and the large number of client requests recorded in the workload traces of an operational CDN. The number of CDN power states n_{cps} is of exponential order in the number of cache servers and polynomial order in the number of disks per cache, i.e., $n_{cps} = \sum_{s=1}^{s_{max}} (d_{max}+1)^s = \mathcal{O}(d_{max}^{s_{max}})$, where s_{max} represents the number of provisioned cache servers, d_{max} the number of disks provisioned in every cache server, and s the number of active cache servers. In addition, the large number of client requests recorded in the workload traces used to drive the simulation leads to a long simulation time. Therefore, we subsampled the traces by selecting clients randomly. In the subsampled traces used for the evaluation, we count on average still ~ 250 million requests per day.

In this paper, we make the following two research contributions. As our main contribution, we propose a CDN energy *optimizer*, which uses an offline heuristic algorithm that aims to find for each time interval the CDN power state minimizing the consumed energy while limiting the performance deterioration. We avoid the use of an exhaustive search algorithm because the time complexity of such an algorithm is linear in the number of CDN power states, which is exponential in the number of caches. The proposed algorithm first determines per time interval the minimum number of active cache servers required to guarantee sufficient bandwidth from the caches to the clients such that constraint (1) above is satisfied. Then, we use a greedy algorithm to find the minimum number of active disks required to keep the data rate from the origin to the caches below the agreed threshold such that we adhere to constraint (2) above. We indicate under which conditions our heuristic approach approximates the optimal solution. In practice, these conditions are most of the time fulfilled.

To evaluate the optimizer, we present a trace-drive CDN energy simulator [3]. The simulator is driven using real workload traces recorded by an operational telco CDN delivering IPTV to smartphones and tablets using HTTP adaptive streaming. This type of workload exhibits cyclic load fluctuations similar but not identical to related workloads such as traditional IPTV and web-based video sharing. The simulator models the energy consumption of a cache server (excluding disks) in a known way as a linear function of the server load [4]. The energy consumption of the disks is modeled separately as a known function of disk read and write accesses [2]. By turning off more or less caches and/or disks, the simulator trades off the CDN's energy consumption against its throughput and bandwidth efficiency. Because we consider a simulator to be the software for running a simulation, we use the terms simulator and simulation interchangeably.

The remainder of the paper has the following outline.

Section II describes the trace-driven CDN energy simulator. In Section III, we introduce the CDN energy optimizer. The optimizer is evaluated in Section IV. In Section V, we describe related work. Finally, Section VI is the conclusion of our paper.

II. TRACE-DRIVEN CDN ENERGY SIMULATOR

As input to our CDN energy simulator, we use workload traces generated by an operational telco CDN delivering IPTV for live television (linear) and video-on-demand (vod) to mobile devices using HTTP adaptive streaming. We present the workload characteristics most relevant to this paper for a subsampled (as explained in § I) 7-day workload trace recorded during the first quarter of 2013. Fig. 1 shows the file download bandwidth over a single week. We make a distinction between vod and linear video. We observe that, during the week under consideration and for this particular CDN, linear video accounts for the largest share of the download bandwidth. The figure reveals diurnal patterns of cyclic load fluctuations that can be expected based on end-user behavior: load peaks in the evening when most people are at home watching videos and dips during the night when most people are asleep. On-demand video shows a variation of the workload over time similar to linear video but vod exhibits no small load peak during the morning before people go to work like linear video does. During the week under consideration, the maximum load is roughly six times higher than the minimum load. Over longer time periods, the ratio between the maximum and minimum load can be expected to be even higher considering seasonal load fluctuations and gradual load changes due to a time-varying number of clients. Such load fluctuations represent an opportunity for saving energy because of the substantial overprovisioning of cache servers and disks most of the time. Cache servers and their disks can be turned off according to the time-varying workload. Because of the differences in load variation between linear and on-demand video, it is possible that the ratio between the optimal number of active disks and active cache servers changes over time.

The specific workload under consideration exhibits cyclic load fluctuations similar to other types of video-streaming workloads such as traditional IPTV [5], where the client is a TV screen, and web-based video sharing [6]. The traditional IPTV workload characterized in [5] has a less significant peak in the morning but shows a quite large peak around lunch time. The web-based video sharing workload analyzed in [6] peaks during the afternoon instead of during the evening.

The CDN energy simulator used in this paper models only those aspects of the CDN that significantly impact the energy consumed by the CDN's cache servers and their disks. The simulator models the CDN's distribution system and part of its request-routing system according to the state of the art in the design of telco CDNs. The request-routing system

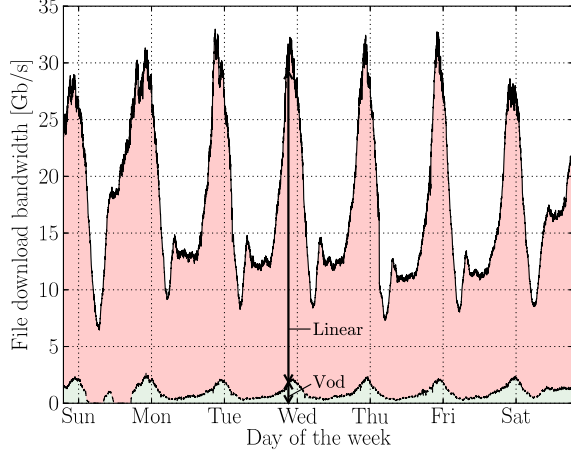


Figure 1: File download bandwidth split in linear video and vod over a single week.

typically redirects every client request from the origin to an available cache server in a data center nearest to the client. Because we consider only one of the CDN's data centers as justified in Section I, the simulator doesn't model the global geographical load balancing; only local (per data center) load balancing is simulated. A new client for the data center under consideration is redirected to the data center's cache that is least loaded. As long as that cache is not overloaded, every new request from the same client is redirected to it. When a client sends a request to an overloaded cache, that client is redirected to another cache in the same data center based on the load-balancing policy.

The CDN's distribution system is composed of caches that serve client requests by fetching the requested file in order of priority from (1) their memory, (2) one of their disks, (3) a cache in the same rack using the Internet Cache Protocol (ICP), or (4) the origin server. In the case of a cache hit, i.e., a replica of the file is available in the cache's memory or on one of its disks, the cache serves the requested file directly. Upon a cache miss, the cache first pulls the requested file from the origin or a neighboring cache in the same rack and caches it in memory before serving it to the client. For sake of simplicity, we assume in this paper that there is only one cache server per rack. If the cache server needs to read the file from disk, it moves the file from disk to memory. Also a file received from the origin is cached in memory. When a file needs to be written to memory but there is insufficient space available, the cache replacement policy is consulted to move files from memory to disk to free the required space. The disk to write the files to is selected based on a load-balancing policy. The disks apply the same cache replacement policy; we use LRU in our evaluation in Section IV. However, the disks do not cache linear video streams because the linear video traffic dominates the vod traffic and the memory is large enough to cache the linear

video for the short duration it is relevant to the clients. For simplicity we assume all cache servers to be of the same type. The same holds for the disks. In addition, we assume all cache servers to be provisioned with the same number of disks. These assumptions typically match reality for new deployments of telco CDNs.

The simulator is driven by a workload trace. The client requests logged in the trace are processed one after the other by the simulator based on the proposed CDN model. The simulator keeps track of the files cached in memory and on disk. The processing of the requests leads to energy dissipation in the cache servers which is recorded per time interval T_{res} of 5 min (default). Also the download rate (from caches to clients) and the upload rate (from origin to caches) are logged per time interval. The disks are modeled separately from the server itself. An active disk consumes at least the idle power P_{id} . While seeking or during data transfer the power consumption increases to the active power P_{act} . The energy consumed by a disk during time interval j is modeled as $E_j^{dsk} = T_{res}P_{id}^{dsk} + (T_j^{sk} + T_j^{tf})(P_{act}^{dsk} - P_{id}^{dsk})$. T_j^{sk} is the total seek time during the interval while T_j^{tf} stands for the total transfer time. We assume that reading, writing, and seeking requires the same power. The calculation of seek and transfer time per time interval depends on the exact sequence of client requests because of caching and therefore requires simulation.

The energy E_j^{srv-} consumed by an active server excluding disks during time interval j is modeled as a linear function of the server load $\lambda_j = D_j^{srv}/D_{max}^{srv}$ with D_j^{srv} the server download rate at interval j and D_{max}^{srv} the maximum server download rate, i.e., $E_j^{srv-} = T_{res}P_{id}^{srv-} + \lambda_j T_{res}(P_{max}^{srv-} - P_{id}^{srv-})$. P_{id}^{srv-} represents the server idle power and P_{max}^{srv-} the power consumed by the server under maximum load. We thus assume that the energy consumed by the server (excluding disks) for delivery of a file only depends on the file size and not on the location from where the file is fetched. The energy E_j^{srv} consumed by an active cache server including disks during time interval j is the sum of the energy consumption of the server and its disks during that interval. The CDN's energy consumption E_j^{cdn} is the sum of the energy consumed by each of its caches. We ignore the additional energy required for hosting these servers in a data center (such as energy for cooling and network access).

III. CDN ENERGY OPTIMIZER

In this section, we describe the CDN energy optimizer. The objective of the optimizer is to find offline for every time interval j the CDN power state $p_j = (d_{1,j}, \dots, d_{s,j,j})$ that minimizes the energy E_j^{cdn} consumed by the CDN while ensuring that (1) the maximum CDN download rate $D_{max,j}^{cdn}$ exceeds the rate $D_{req,j}^{cdn}$ required to serve all requests from the clients and (2) the CDN upload rate $U_{req,j}^{cdn}$ required to serve all requests from the caches does not exceed the

maximum rate U_{max}^{cdn} . The first constraint ensures client requests are not dropped, and the second constraint prevents trivial solutions with all disks turned off. This discrete optimization problem is formulated in (1). We call the first constraint in (1) the download constraint and the second one the upload constraint.

$$\begin{aligned} & \underset{p_j}{\text{minimize}} && E_j^{cdn}(p_j) \\ & \text{subject to} && D_{max,j}^{cdn}(p_j) \geq D_{req,j}^{cdn} \\ & && U_{req,j}^{cdn}(p_j) \leq U_{max}^{cdn} \end{aligned} \quad (1)$$

For every time interval, solving this optimization problem requires the exploration of different power states. Regardless of the time interval under consideration, each power state corresponds to one simulation. Indeed, for every simulation used to solve the optimization problem, the CDN remains in the same power state during the complete simulation time window; a simulation is completely identified by the power state of the simulated CDN. However, the solution to the optimization problem is a sequence of possibly different power states that, when applied to the CDN, leads to the minimum energy consumption over time. Thus, the minimum energy consumption that results from this sequence of power states is actually rather a lower bound for the CDN's energy consumption because we ignore the transitions between different power states. For sake of simplicity, we drop in the remainder of this section the subscript j .

To solve the energy minimization problem, we propose a CDN energy optimizer. In practice, this optimizer needs to be considered a heuristic method, which only approximates the energy-optimal solution to the constraints when the four conditions introduced below are fulfilled. In this section, we explain why these conditions are likely to hold most of the time. The evaluation (Section IV) shows that these conditions are indeed valid most of the time for the specific workload and CDN configuration used. In Section IV we also quantify our heuristics using an exhaustive algorithm guaranteed to find the minimal solution to the energy optimization problem under all conditions. In terms of power consumption, our heuristic algorithm only deviates by less than 0.01 % from the optimal.

The CDN energy optimizer approaches the energy minimization problem by considering in order the number of active cache servers (§ III-A) and the number of active disks per active cache (§ III-B). The optimizer first determines the minimum number of active caches that solves the download constraint. Minimizing the number of cache servers is also the first step in solving the upload constraint under the condition that, when all disks are activated, an additional cache server will not reduce the CDN upload rate. Moreover, this minimum number of active caches corresponds to the energy-optimal solution to the constraints under the condition that activating an additional cache increases the energy usage of the CDN.

Given this resulting number of active caches, the optimizer search for the CDN power state with the minimum total number of active disks that solves the upload rate constraint using a greedy algorithm. The greedy algorithm finds this CDN power state under the condition that a disk activation increases the hit rate less than the previous disk activation in the same cache server. Finally, this CDN power state approximates the energy-optimal solution to the constraints under the condition that, given a number of active caches, a CDN consumes more energy when more disks are active.

A. Number of Active Cache Servers

To find a solution to the energy minimization problem, we first consider the number of active caches. The maximum CDN download rate D_{max}^{cdn} depends only on the number of active cache servers s and not on the number of active disks per active server. This dependency is formulated as $D_{max}^{cdn}(p) = sD_{max}^{srv}$ where D_{max}^{srv} represents the maximum server download rate, which is a characteristic of the server and, therefore, remains constant over time. The disks never become the system bottleneck because, if they are overloaded, the cache fetches files directly from the origin. Therefore, the download constraint determines a minimum number of active cache servers as $s_{min} = \lceil D_{req}^{cdn} / D_{max}^{srv} \rceil$.

Selecting the minimum number of active caches s_{min} is the first step in solving the upload constraint under the condition that activating an additional cache server does not reduce the CDN's upload rate even when all disks are turned on. We expect this condition to hold most of the time as we explain next. The number of active cache servers determines the load distribution of the client requests over the caches. This load distribution has an impact on the performance of the individual caches. Therefore, changing the number of active caches affects the upload rate U_{req}^{cdn} required to serve all requests from the caches. Every additional active cache is expected to increase this required upload rate because increasing the number of active caches leads to fewer clients per active cache and therefore less file sharing. We thus expect in general that powering up an additional cache server (with all disks in all servers active) will not help satisfy the upload constraint but will on the contrary increase the required CDN upload rate due to diminished file sharing.

Thus, the minimum number of active cache servers s_{min} not only solves the download constraint but also represents the best option for addressing the upload constraint. In addition, selecting a minimal number of active cache servers minimizes the total energy consumed by the CDN under the condition that activating an additional server increases the energy usage of the CDN. This condition is likely to commonly hold because both the energy consumed by the servers (excluding disks) and the energy consumed by the disks are expected to increase when an additional cache server is activated. Every additional active cache simply

increases the energy consumption per time interval by the idle energy $P_{id}^{srv}T_{res}$ (derived from the model in Section II) in our model because the server energy depends on the server load, which is distributed over more servers but does not change in total. In addition, we also expect the disk energy to increase upon the activation of an additional cache because increasing the number of active caches tends to increase the required upload rate as explained above. To overcome such upload rate increase more data needs to be read from the disks to satisfy the upload constraint. Therefore, more disks may need to be activated and the disk energy increases.

B. Number of Active Disks per Active Cache Server

We consider the energy minimization problem given $|p_j| = s_{min}$ and only consider the upload constraint (assuming s_{min} provides a solution to the download constraint). This problem can be restated as how many disks to activate per active cache server so as to minimize energy usage while meeting the upload constraint. This problem can be formulated as a multiple-choice knapsack problem (MCKP) with caches as classes, numbers of active disks as items, energy savings as profit, and cache upload rate as weight.

We propose a greedy heuristic algorithm to find the number of active disks per active cache that is a simplified version of the MCKP-Greedy algorithm described in [7]. MCKP-Greedy leads to the so-called split solution, which is generally a good heuristic solution for the MCKP. Applied to our problem, MCKP-Greedy would take the ratio of upload rate decrease in activating a disk over the corresponding energy consumption increase as a metric to select the disk for activation. Our greedy algorithm uses the upload rate decrease alone, as we will discuss. Under certain conditions this simplification leads to a solution close to optimal for the MCKP under consideration.

Our greedy algorithm targets the minimum total number of active disks across the active caches required to satisfy the upload constraint. Because a total of 0 active disks is the minimum possible, the greedy algorithm starts from the CDN power state represented by the s_{min} -tuple $(0, \dots, 0)$. If this state satisfies the upload constraint, then this state is clearly the solution to the energy minimization problem because the consumed disk energy is minimal when all disks are turned off. As long as no CDN power state is found that satisfies the upload constraint, the greedy algorithm iteratively chooses in which server (of the s_{min} servers) to activate a disk. Activating a disk increases the cache's size, which may increase the cache hit ratio. A larger cache hit ratio will result in a lower required upload rate. At every step, the greedy algorithm explores s_{min} different CDN power states, each with an additional active disk in a different active cache. The algorithm then chooses the CDN power state with the greatest reduction in required CDN upload rate as the base for the next iteration. If exactly one of the explored CDN power states satisfies the upload

constraint, a solution is found. If multiple CDN power states (with the same total number of active disks) meet the upload constraint, the algorithm selects the one with the lowest energy consumption as solution.

The greedy algorithm satisfies the upload constraint by activating as few disks as possible under the condition that the activation of another disk in a cache server results in a smaller decrease of the required cache upload rate than the previous disk activation did in the same server. In terms of hit rate, the condition requires that activating another disk increases the hit rate less than the previous disk activation. This condition needs to hold for every cache server. We expect this condition to hold often because increasing the cache size (by activating more disks) follows the law of diminishing returns. However, this condition might not hold at all times in real systems where the cache replacement policy is suboptimal.

The CDN power state found by the greedy algorithm not only solves the download and upload constraint with the minimum total number of active disks but also approximates the minimum CDN energy consumption under the condition that given a number of active caches, disk energy consumption increases with the number of active disks. This condition is likely to hold because the idle energy consumed by the disks increases linearly with the total number of active disks regardless of the distribution of the active disks over the caches. For every additional active disk, the idle energy $P_{id}^{disk}T_{res}$ is added as can be derived from the model in Section II. The disk write energy does not depend on the number of active disks per active cache as long as for none of the caches the disks are overloaded and at least one disk is active. The disk read energy increases with the data volume directly served from the disk cache, which tends to increase with the total number of active disks.

The greedy algorithm, which determines the number of active disks per active cache, needs to explore in the worst case n'_{cps} CDN power states per time interval of linear order in the number of disks and of quadratic order in the number of servers, i.e., $n'_{cps} = 2 + (d_{max}s_{max} - 1)s_{max} = \mathcal{O}(d_{max}s_{max}^2)$, whereas an exhaustive search would require a number of power state visits per time interval exponential in the number of servers and polynomial in the number of disks.

IV. EVALUATION

We evaluate the CDN energy optimizer based on a 24-hour subsampled (as explained in §I) workload trace recorded by an operational CDN during the Monday of the week in the first quarter of 2013 for which workload characteristics are presented in Section II. Other days of the week show similar results. The simulated CDN differs from the operational CDN. The CDN energy simulator models one of the CDN's data centers and is configured with the minimum number of caches required to allow serving

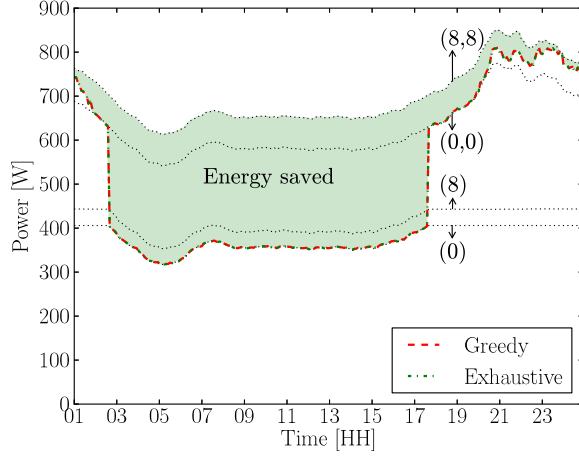


Figure 2: CDN's power consumption over a single day corresponding to the solution found by the greedy and exhaustive search algorithm, which almost perfectly coincide.

all client requests during any time interval (5 min). We don't expect an improvement of the bandwidth efficiency by adding caches as explained in Section III. Moreover, limiting the number of provisioned caches to the minimum leads to a conservative estimate of the potential energy savings. This minimum number of caches turns out to be two for the type of cache server used for the simulation.

The simulated cache is an HP proliant server equipped with a Dual Intel Xeon 5600 processor, 144 GiB of DDR3 RAM, and eight 146 GB hard disk drives. This type of server exhibits a maximum download rate of 18 Gb/s. When idle, this type of server excluding disks consumes 224.6 W. Its maximum power consumption excluding disks is 405.88 W. The 146 GB hard disk drive has a transfer rate of 141 MiB/s. On average, the disk's seek time is 2.98 ms and its rotational latency 2 ms. The disk idle power is 4.37 W and active power 7.25 W. These device specifications are mean values. We don't have access to the confidence intervals. Therefore, all simulation results presented in this section are mean values as well without confidence intervals. The maximum upload rate is set to 4.3 Gb/s. The caches use an LRU replacement policy and were filled before the time shown in the graphs.

We solve the energy minimization problem with the greedy algorithm proposed in Section III and, for validation, with an exhaustive search algorithm, which is guaranteed to find the global optimum. For both solutions, we present the CDN's power consumption (Fig. 2), download rate (Fig. 3), and upload rate (Fig. 4) over a single day. For comparison, we also show the result for the simulations (0), (8), (0,0), and (8,8).

Fig. 2 shows the CDN's power consumption over a single day. We observe that the solution found by the greedy algorithm approximates the optimal solution very well. The average absolute error is about 50 mW, which corresponds

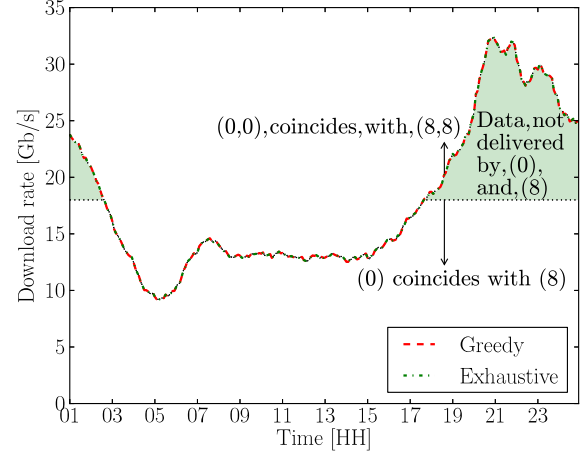


Figure 3: CDN's download rate over a single day corresponding to the power state found per time interval by the greedy and exhaustive search algorithm, which coincide.

to a relative error of 0.007 %. To serve all requests, only one active cache is required between $\sim 03:00$ and $\sim 18:00$; during the rest of the day two active caches are required. From $\sim 20:00$ till $\sim 02:00$ disks get activated to keep the upload rate below the configured limit; the rest of the day all disks are powered down. If all caches and their disks would remain turned on the whole day (simulation (8,8) in Fig. 2), the energy consumed by the CDN during this day would amount to 16.83 kWh. By applying DPM to the cache servers and their disks, the CDN's energy consumption is reduced by 4.88 kWh (the shaded area in Fig. 2), which corresponds to a relative reduction of 29 %. Turning off caches accounts for 79 % of the energy savings; the disk power management contributes 21 % to the savings.

The CDN's download rate is shown in Fig. 3. This figure clarifies why two active caches are required between $\sim 18:00$ and $\sim 03:00$ to satisfy the download constraint. The download rate of the type of cache server used for the simulation is limited to 18 Gb/s. If the download rate required to serve all client requests during a time interval exceeds 18 Gb/s, a second cache needs to be activated. The shaded area represents the data volume that cannot be delivered by a single cache.

Fig. 4 shows the CDN's upload rate over a single day. We observe that both the greedy and exhaustive search algorithm adhere to the upload constraint, which limits the upload rate to 4.3 Gb/s. The shaded areas above the upload rate limit represent data that doesn't need to be pulled from the origin but can be delivered directly from cache instead because disks were activated. In this case, energy is traded for bandwidth efficiency. The shaded area below the upload rate limit also represents data that doesn't require retrieval from the origin but can be distributed directly from cache because the second cache was turned off. In this case, the

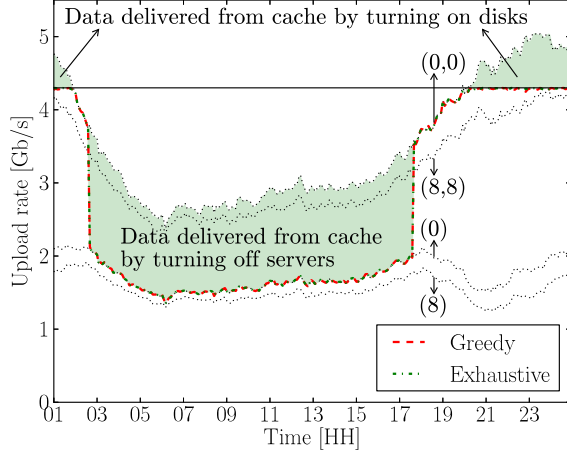


Figure 4: CDN's upload rate over a single day for the solution found by the greedy and exhaustive search algorithm, which almost perfectly coincide. The maximum upload rate equals 4.3 Gb/s.

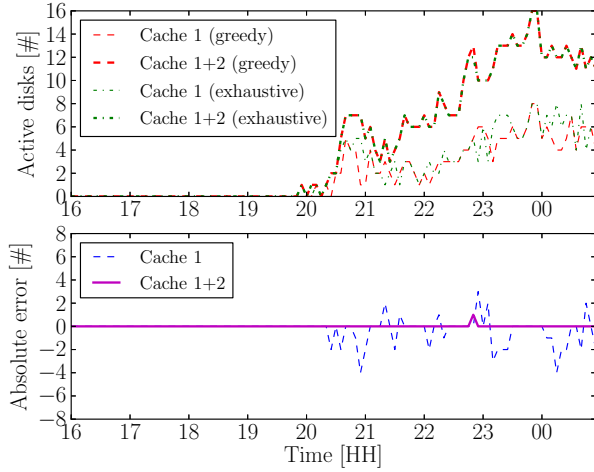


Figure 5: Top: Number of active disks found per time interval by the greedy and exhaustive search algorithm over the last nine hours of the day under consideration: for the first cache only (thin) and for the combination of the two caches (thick). Bottom: Difference in number of active disks between the greedy and exhaustive search algorithms given in the top graph.

upload rate reduction goes hand in hand with energy savings.

Fig. 5 shows the number of active disks per cache for the most relevant time window. The top of the figure reveals the total number of active disks per time interval as well as the distribution of the active disks over the two caches as determined by the greedy and exhaustive search algorithm. The bottom of the figure shows the difference between the greedy and exhaustive algorithm in terms of the total number of active disks and the number of active disks in the first cache. The greedy algorithm almost always finds the energy-optimal total number of active disks. Although the greedy and exhaustive algorithm do distribute the active

disks significantly differently over the active caches, the impact on the CDN's power consumption appears to be marginal. Moreover, the two caches seem to require more or less the same number of active disks. Therefore, it seems worthwhile to investigate whether the time complexity of the search through the state space could be further reduced without unacceptable impact on the accuracy of the solution by assuming an even distribution of the active disks over the active caches.

In addition, we verified the validity of the four conditions (introduced in Section III) under which our greedy algorithm approximates the optimal solution. It turns out all conditions hold during the complete simulation window except the condition that requires the activation of another disk in a cache to result in a smaller upload rate decrease than the previous disk activation in the same cache. The latter condition holds for most time intervals and caches but not for all.

V. RELATED WORK

Based on the analysis of over a hundred high-quality papers on power-reduction techniques for data-center storage systems, we prepared an exhaustive survey of this domain [2]. Power-reduction techniques were in succession proposed for individual disks, RAID (redundant array of independent disks), and clusters of storage servers. Although a CDN is composed of storage elements distributed over multiple data centers, we are not aware of any work in the specific domain of power-aware storage systems that targets CDNs. Most of the proposed techniques rely on DPM. However, a typical data-center workload characterized by short idle periods needs to be reshaped to enable DPM. One of the DPM-enabling techniques is DIV [8] (short for diverted accesses), which segregates original and redundant data on different storage devices such as different disks in a RAID [9] or different storage servers in a cluster [10]. Under light load, client requests are diverted from redundant to original devices such that the former can be turned off. Because of its inherent segregation of original and redundant data on different servers and its centralized request-routing system, it should be relatively straightforward to apply DIV to CDNs to enable DPM.

Although we are not aware of any work that applies power-reduction techniques for storage systems to CDNs, researchers have started working on energy-efficient CDNs. The main objective of this research is to come up with energy-aware cache-server and file-replica placement. [11] shows that a typical CDN architecture with caches placed near the clients is more energy-efficient than both peer-to-peer delivery and delivery from a central server. Other studies led to consistent results for the specific case of IPTV distribution [12]. More energy might even be saved by integrating the caches into the home gateways [13]. Content-centric networking (CCN) has more potential for saving

energy than traditional CDNs because CCN integrates the caches into relatively energy-efficient routers [14]. In this paper, we consider the cache-server placement characteristic of a telco CDN architecture as a given. Recent research in the domain of energy-efficient CDNs focuses on energy-aware load balancing [15], which is closest to our research focus. However, we consider DPM not only at the level of the cache servers but also their disks. Moreover, our evaluation is based on HTTP adaptive streaming workload traces recorded by an operational telco CDN delivering IPTV to mobile devices. Mathew et al., on the other hand, use traces from a traditional CDN. Finally, we are not aware of the existence of other CDN energy simulators although CDN performance simulators do exist [16].

VI. CONCLUSION

The proliferation of power-hungry, disk-packed cache servers in data centers of ISPs for the deployment of telco CDNs to deliver multiscreen IPTV services fuels the increase of data-center power consumption. In this paper, we target energy savings in telco CDNs by applying DPM to the cache servers and their disks. We propose a heuristic offline algorithm to find per time interval the minimum number of active caches and active disks per active cache required to serve all client requests and limit the data rate from the origin to the caches. This heuristic algorithm approximates the optimal under four conditions. We argue that these conditions can be expected to hold frequently. This expectation is confirmed for the specific workload and simulated CDN used for the evaluation. The deviation of our greedy algorithm from an exhaustive optimal one is insignificant (relative error less than 0.01 % on average). The greedy algorithm is only of quadratic order in the number of servers whereas the exhaustive one is of exponential order. We evaluate the algorithm using a CDN energy simulator driven by HTTP adaptive streaming workload traces recorded by an operational telco CDN delivering IPTV to mobile devices. Even for a minimally-provisioned CDN, the evaluation reveals potential DPM-based energy savings of approximately 30 % thanks to predictable cyclic load fluctuations. In the future, we plan to develop online algorithms, which can be implemented in real CDNs to realize the energy savings in an unpredictable environment.

ACKNOWLEDGMENT

The authors would like to thank their colleagues of Velocix, an Alcatel-Lucent company, and Koen Laevens for their support in getting access to CDN workload traces. In addition, this work is supported by the Flanders Agency for Innovation by Science and Technology (IWT), grant IWT 100690.

REFERENCES

- [1] J. Koomey, "Growth in data center electricity use 2005 to 2010," Analytics Press, Oakland, CA, USA, Tech. Rep., August 2011, <http://www.analyticspress.com/datacenters.html>.
- [2] T. Bostoen, S. Mullender, and Y. Berbers, "Power-reduction techniques for data-center storage systems," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 33:1–33:38, Jun. 2013.
- [3] T. Bostoen, J. Napper, S. Mullender, and Y. Berbers, "A simulator to assess energy-saving techniques in content distribution networks," in *Proceedings of the 2nd international workshop on energy-efficient data centres*, ser. E²DC '13. Berlin, Germany: Springer-Verlag, 2013, in press.
- [4] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *Computer*, vol. 40, pp. 33–37, December 2007.
- [5] M. Cha, P. Rodriguez, J. Crowcroft, S. Moon, and X. Amatriain, "Watching television over an ip network," in *Proceedings of the 8th ACM SIGCOMM conference on Internet measurement*, ser. IMC '08. New York, NY, USA: ACM, 2008, pp. 71–84.
- [6] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: a view from the edge," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 15–28.
- [7] H. Kellerer, U. Pferschy, and D. Pisinger, *Knapsack Problems*. Springer, 2004.
- [8] E. Pinheiro, R. Bianchini, and C. Dubnicki, "Exploiting redundancy to conserve energy in storage systems," *SIGMETRICS Perform. Eval. Rev.*, vol. 34, pp. 15–26, June 2006.
- [9] J. Wang, H. Zhu, and D. Li, "eraid: Conserving energy in conventional disk-based raid system," *IEEE Trans. Comput.*, vol. 57, pp. 359–374, March 2008.
- [10] E. Thereska, A. Donnelly, and D. Narayanan, "Sierra: practical power-proportionality for data center storage," in *Proceedings of the 6th conference on Computer systems*, ser. EuroSys '11. New York, NY, USA: ACM, 2011, pp. 169–182.
- [11] A. Feldmann, A. Gladisch, M. Kind, C. Lange, G. Smaragdakis, and F.-J. Westphal, "Energy trade-offs among content delivery architectures," in *Proceedings of the 9th conference on Telecommunications, Media and Internet Techno-Economics*, ser. CTTE '10. Los Alamitos, CA, USA: IEEE, 2010, pp. 1–6.
- [12] J. Baliga, R. Ayre, K. Hinton, and R. Tucker, "Architectures for energy-efficient iptv networks," in *Proceeding of the 2009 Conference on Optical Fiber Communication*, ser. OFC '09. Washington, DC, USA: OSA, 2009, pp. 1–3.
- [13] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the internet with nano data centers," in *Proceedings of the 5th international conference on Emerging networking experiments and technologies*, ser. CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 37–48.
- [14] U. Lee, I. Rimac, D. Kilper, and V. Hilt, "Toward energy-efficient content dissemination," *IEEE Network*, vol. 25, no. 2, pp. 14–19, Mar. 2011.
- [15] V. Mathew, R. Sitaraman, and P. Shenoy, "Energy-aware load balancing in content delivery networks," in *Proceedings of the 31st annual IEEE international conference on computer communications*, ser. INFOCOM '12. Los Alamitos, CA, USA: IEEE, 2012, pp. 954–962.
- [16] K. Stamos, G. Pallis, A. Vakali, D. Katsaros, A. Sidiropoulos, and Y. Manolopoulos, "Cdnsim: A simulation tool for content distribution networks," *ACM Trans. Model. Comput. Simul.*, vol. 20, no. 2, pp. 10:1–10:40, May 2010.