CHAPTER 5

SHRINK: GREENING CDN DATACENTERS

Our growing consumption of digital content is increasing the energy use of datacenters used for content delivery, and datacenters today consume more than 1.3% of world's electricity consumption [102]. This growth is a result of increased availability of several forms of content such as video [33, 85], and social media content, as well new platforms for content consumption, such as smartphones and tablets [33]. Due to our growing dependence on digital content, mechanisms for reducing energy use of content delivery are an important societal need. Further, such mechanisms have potential to be an important cost-cutting tool for content delivery networks (CDNs). Energy is a major factor of operational costs [24] of a datacenter; reducing energy cost of their datacenters would help CDNs stay competitive in an commoditized marketplace.

Recent research has shown that techniques that consolidate demand in a datacenter on a subset of servers can enable turning off up to 50% servers and bring significant energy savings [82, 60, 76, 80]. However, the impact of this energy minimization on user-perceived performance in a CDN datacenter has not been evaluated before. These studies do not evaluate how cache hit rates in a CDN datacenter would be impacted due to server shutdown policies. Note that reduced hit rates adversely affect user-perceived performance. Further, their analyses are based only in terms of system-level metrics, e.g., aggregate load at a datacenter, and not based on user-perceived metrics such as file download times.

The network of a datacenter, which consumes about 10-20% of datacenter energy, can be made energy efficient by traffic engineering techniques as shown by prior work [113, 63, 123, 32, 16]. The energy savings that a scheme achieves depends on traffic patterns in the datacenter, traffic patterns that traffic engineering schemes assume to be fixed. This assumption isn't true for content delivery data centers, where traffic patterns could be influenced by load balancing decisions and server shutdown policies. We hypothesize that there exists potential for more network energy savings in datacenters than shown previously, provided techniques for saving server energy work in coordination with those for saving network energy.

Our goal is to quantify how much energy savings are achievable in a CDN datacenter with minimal or no impact on user-perceived performance. To this end, we seek to design server and network shutdown policies that coordinate with each other and increase datacenter network energy savings, and design server shutdown policies and load balancing algorithms that minimize the reduction in cache

hit rates. To conduct a realistic evaluation of proposed techniques, we plan to collect datasets of content access traces of various types from a CDN, and evaluate proposed strategies in terms of user-perceived metrics using a combination of trace-driven experiments and testbed experiments with a prototype.

5.1 Related work

To our knowledge, this would be first effort to evaluate energy minimizing strategies in a CDN datacenter in terms of user-perceived performance, and to explore coordinated server and network shutdown policies. Prior work has explored energy-minimizing strategies for datacenter and ISP networks, and has evaluated potential energy savings of datacenters, without evaluating user-perceived performance metrics.

Energy-minimizing routing: Several papers [113, 63, 123, 32, 16] have proposed network energy minimizing routing algorithms for ISP networks and data center networks. These approaches save energy by concentrating the traffic, which is input in the form of a traffic matrix, on a subset of links and switches, and shutting off remaining switches and links. In [113] and [63], authors demonstrate using Click and OpenFlow based prototypes respectively, the feasibility of implementing these routing protocols in today's switches. In comparison to approaches that optimize routing for a given traffic matrix, this work explores how server shutdown policies can coordinate with energy-minimizing routing strategies to increase energy savings in a datacenter.

Energy-proportional datacenters: Analysis of data center traces have shown that servers in datacenters are lightly loaded in many cases. Motivated by this observation, several papers [82, 60, 76, 80], based on trace-driven experiments, have explored how much energy savings can be obtained by using only a fraction of the servers at a given time, and by shutting-off remaining servers or switching them to a low power state. While these studies show that there is significant potential for energy savings, they implicitly assume the total load will be evenly distributed among the active set of servers. However, server shutdown policies could cause additional load imbalance in datacenters, thereby degrading user-perceived performance. Further, in case of a CDN datacenter, they do not evaluate the impact of server shutdown policies on cache hit rates, which are a key determinant of user-perceived performance. In comparison, a key goal of this work is to design load-balancing and server shutdown strategies that ensure cache hit rates are minimally affected, and load imbalance is small enough to not cause a degradation in user-perceived performance.

Other work: There are a number of recent efforts whose goals are complementary to this work. Work on device-level power management for switches [84] and servers [27] could complement our strategies. A CDN could use geographic load balancing to globally optimize energy use across datacenters while using our strategies in a single datacenter [96, 78, 58, 100]. Using recently developed techniques for designing highly scalable load-balancers such as ETTM [43] and Ananta

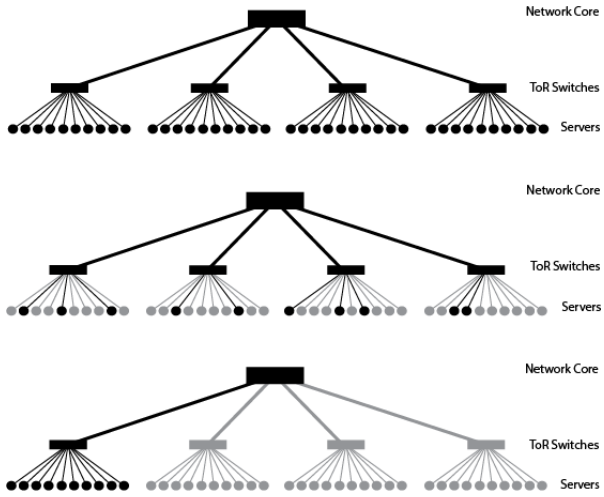

Figure 5.1. A datacenter topology. Black components are turned on and grey components are turned off. (Top) All servers and switches are on as is the current practice. (Middle) Demand is consolidated on randomly selected 10 servers and remaining servers are shutoff. All ToR switches must be kept on to provide connectivity to servers. (Bottom) Demand is consolidated on servers in one rack, which allows servers as well as ToR switches in other racks to be turned off.

[92], a CDN can extend the strategies we propose in this work to develop a scalable solution for use in datacenters with tens of thousands of servers.

5.2 Research outline

Our goal is to design a comprehensive solution for CDN datacenters which makes load balancing, server shutdown, and traffic engineering decisions to reduce energy use with a minimal impact on user perceived performance.

5.2.1 Algorithm design

There are three design questions which are key to achieving the performance and energy-efficiency goals of a CDN datacenter.

(1) How many servers to keep active? Reducing the number of active serves is necessary for energy-efficiency because servers consume more than 80% of power usage in datacenters [13], and today's servers consume 50-70% energy even in idle state [24]. We will decide the number of active servers to ensure the following:
Availability: The datacenter will have sufficient resources, e.g, compute, network, disk bandwidth, to handle all incoming requests. ensure both high availability and high performance.
Performance: Cache misses at datacenter require objects to be fetched from a remote datacenter and results in a noticeable increase in user-perceived performance. The

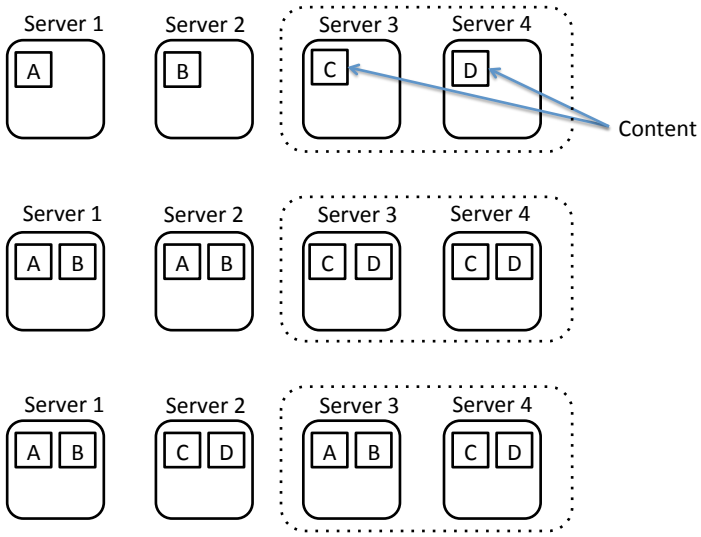

Figure 5.2. Replication strategy impacts content availability after server shutdown. All four servers are active during normal utilization periods, but the two servers on the right are turned off during low utilization periods. Squares with same letters represent replicas of the same content. (Top) One replica of each content is maintained as shown. When servers 3 and 4 are shutdown, two of the four content become unavailable. (Middle) Two replicas of each content are maintained, but still shutting down servers 3 and 4 makes two of the four content unavailable. (Bottom) Two replicas of each content are maintained, but servers 1 and 2 have one copy of all four content. In this case, all content is available despite shutting down servers 3 and 4.

aggregate storage available across servers will be sufficient to enough a cache hit rate that is comparable to the cache hit rates if all servers were active.

(2) Which servers to keep active? The set of active servers determine the potential energy savingsfrom network switches. As shown in Figure 5.2, randomly selecting the set of active servers requires entire datacenter network to be turned off. Therefore, we will select the active servers in a manner that enables more network switches to be turned off, e.g., selecting servers that are within a small sub-tree in a datacenter topology.

(3) Which servers to replicate each content? As Figure 5.2 shows, if insufficient number of replicas are maintained or the set of servers is not chosen carefully, then server shutdown could temporarily make content unavailable in the cluster. Such content unavailability decreases cache hits in datacenter, thereby hurting user perceived performance. To maximize cache hits, we will replicate content across servers so that despite ongoing server shutdown events, one copy of content is kept available in most cases.

5.2.2 System overview

Our algorithms would be implemented as a datacenter manager software, called **Shrink**, running at the front-end load balancer. All servers in the datacenter would be running a caching proxy software such as Squid [40]. All caching proxies would be configured as peers of one another, enabling them to request content from one another. Shrink would make its decisions based on content demand statistics from each server. To this end, Shrink would require support from daemon processes running at each server for reporting statistics. Periodically, Shrink would compute load balancing and traffic engineering decisions, and also decide which set of servers and switches to keep active in the next interval. The computed load balancing decisions would be updated locally. To implement routing decisions, Shrink would require OpenFlow [52] support at switches to communicate the computed routing. To turn servers on/off as necessary, Shrink would again depend on the daemon process running at each server.

5.2.3 Data collection

To evaluate our strategies for a realistic content workload, we have collected content access traces from a datacenter of a leading commercial CDN, Akamai. The traces include all requests received at a datacenter with 18 servers over a week in December 2013. Our anonymized traces traces include several major types of traffic observed in a CDN including video, social media, and other web traffic. Each anonymized log entry includes among other fields, the request timestamp, content URL, size of requested content, actual number of bytes sent, and IP address of the user. Overall, the traces contain more than 2 billion requests generating nearly 200 TB of network traffic.

5.2.4 Experimental evaluation

Our experimental evaluation would seek to find answers to following questions:

- How much energy savings do our strategies achieve compared to the current practice of leaving entire datacenters in "always on" state? How does it compare with respect to a lower bound on energy savings?
- How do the user-perceived performance metrics such as file download times impacted compared to an "always on" strategy? What, if any, is the increase in average and 99-percentile latency? Does a specific subset of content, such as that of a single content provider, see a performance degradation?
- How much additional energy saving does a coordinated server and network shutdown save over uncoordinated approaches for different datacenter network topologies?

A timeline of the proposed research for this project is given in Chapter 7.