

Relaxing CDN energy

Mohamad Dikshie Fauzie* Achmad Husni Thamrin* Rodney Van Meter† Jun Murai†

*Graduate School of Media and Governance

†Faculty of Environment and Information Studies

Keio University, 252-0882 Kanagawa, Japan

dikshie@sfc.wide.ad.jp

husni@ai3.net

rdv@sfc.wide.ad.jp

jun@wide.ad.jp

Abstract—Many CDN companies utilize peer-to-peer to scaling the services and saving bandwidth. However, we still have little knowledge about the energy consumption in peer-assisted CDN. This paper presents an analysis of a energy consumption in peer-assisted CDN system, where an ISP manages its own CDN and its users participate in a P2P network to assist content delivery. To evaluate energy consumption in peer-assisted CDN, we take two cases: (1) live stream and (2) online storage. We developed a simple energy consumption model for live streaming service and online storage service and evaluate this model using a empirical existing live streaming model and online storage model. We find that in live streaming service, total energy system increase with growing number of peers. However, if we concern only on CDN service part, delegate part of workload to peers can save CDN server up to 11% compare to pure CDN architecture, while total energy system difference with pure CDN only less than 1%. In peer assisted online storage, we study three server bandwidth allocation strategies: (1) theoretical lower bound, (2) request driven, and (3) water level. total energy system depends on strategies of server bandwidth allocation. Compare to pure CDN architecture peer-assisted, peer assisted online storage total system energy saving less than 1%.

I. INTRODUCTION

Streaming content, especially video, represents a significant fraction of the traffic volume on the Internet, and it has become a standard practice to deliver this type of content using Content Delivery Networks (CDNs) such as Akamai and Limelight for better scaling and quality of experience for the end users. For example, Youtube uses Google cache and MTV uses Akamai in their operations.

With the spread of broadband Internet access at a reasonable flat monthly rate, users are connected to the Internet 24 hours a day and they can download and share multimedia content. P2P (peer to peer) applications are also widely deployed. In China, P2P is very popular; we see many P2P applications from China such as PPLive, PPStream, UUSe, Xunlei, etc. Some news broadcasters also rely on P2P technology to deliver popular live events. For example, CNN uses the Octoshape solution that enables their broadcast to scale and offer good video quality as the number of users increases.

From the Internet provider point of view, the presence of so many always-on users suggests that it is possible to delegate a portion of computing, storage and networking tasks to the users, thus creating P2P networks where users can share files and multimedia content. Starting from file sharing protocols, P2P architectures have evolved toward video on demand and support for live events.

Broadband network access helps P2P applications to perform better. xDSL networks are deployed worldwide, and in some countries, such as Japan, even higher bandwidth fiber to the home (FTTH) already exceeds DSL in market penetration. In the coming years, FTTH will be massively deployed by network operators throughout the world. As access bandwidth increases, P2P systems may become more efficient since a peer can contribute much more.

In Peer assisted CDN¹, users can download content from CDN nodes for from other users or peers the content. A user may cache the content after download to serve requests from other users. Due to complexity of behavior of peers, the process should be done in home gateway user where ISP can have control on it.

On the other hand, data center where CDN server placed faces costs for powering the data center. The Uptime Institute, a global data center authority, surveyed 1100 data center owners and operators on 2012, reported that 55% organizations must increase budget 10% than 2011 [1]. 30% of organizations will run out of data center capacity (power, cooling, space, and network) in the end of 2012 [1]. More than 50% organizations surveyed reported that saving energy² is major priority [1]. Even in the data centers using the state of art cooling technologies heat dissipation accounts for at least 20% until 50% of the total power consumption [2]. The increases in energy cost and the demand due to growth of traffic urges the data center operators and owners to look for ways to reduce energy usage in the years to come. Although reducing energy consumption can effectively reduce overall cost, this will limit the capacity growth and scalability of the service provisioning. For example: routers and servers spend most of their energy on the baseline activities such as running the fans, spinning harddisks, powering backplane, and powering the memory. Even in an idle state, modern systems can be consuming anything from 50% until 80% of the power consumed under maximum load [3], [4]. Alternatively, the data center can be revamped by relocating some services to end-host computers or peers. Peers contribute their communication, storage, and computation resources to exchange data and provide services while the data center performs central administration and authentication as well as backend processing. P2P network, formed by peers offer flexibility and scalability in service

¹in this paper, we may interchange Peer Assisted CDN and CDN-P2P.

²in this paper, we may interchange energy and power.

delivery. It is not our aim to advocate one system architecture over another. Many issue such as manageability, reliability, and ease of deployment must be taken into account when making high level architectural decisions.

In this paper, we study the energy consumption of hybrid CDN-P2P. It has been known that CDN energy consumption is better than P2P architecture [5], [6], unfortunately that model only think small part of problem energy consumption (devices/hardwares). To be fair, we also need to see how combination of some architectures can help to reduce energy footprint, for example: combination of CDN and P2P.

The rest of this paper is organized as follows. Section 2 present motivation of this paper. Section 3 provides an overview of system model and energy model. In section 4, we will present result and analysis. Section 5 will conclude this paper.

II. MOTIVATION

CDN architectures are host-oriented: content is delivered to end users through host servers that are centrally managed in a few data centers. The growing of Internet traffic dominated by video, the energy consumption of a host oriented architecture becomes problematic due to over provisioning factor. The idea of utilizing the user's computation power to support ISP operation is not new. The figaro project proposed residential gateway as an integrator of different networks and services, becoming an Internet wide distributed content management for a proposed future Internet architecture. The important key in this architecture is both user home gateway and access bandwidth controlled and managed by ISP. In this case, ISP can offload part of workload on their CDN to user's home gateway. By offloading workload, their CDN can relax energy demand thus relaxing data center power budget or relaxing capacity planning for power. This architecture depends largely on the number of devices which can be found online at a point in time. While some users keep their home gateway online at all times, there are small number of users might turn off or their home gateway can not be reached. Based on study from Dischinger et al., [7] and Valancius et al., [8], the average availability of the residential gateways in 12 major ISPs over one month is up 85% with 5% until 8% uptime variations thought out a day thus residential gateway has potential to use as peers.

III. SYSTEM DESCRIPTION

A. Live Streaming

Figure 1 shows example model of CDN with peer assisted for live streaming adapted from [9]. CDN server deliver video contents from content providers to end users. The CDNs are organized into several tiers according to the scale demand, meanwhile satisfying the QoS and reliability requirement. Edge CDNs are directly responsible for serving end users and the remaining part are core CDN. The goal of the server side overlay is efficient data distribution with some measures to guard against some node failures and network delay. Since CDNs are more stable, the server side distribution overlay is

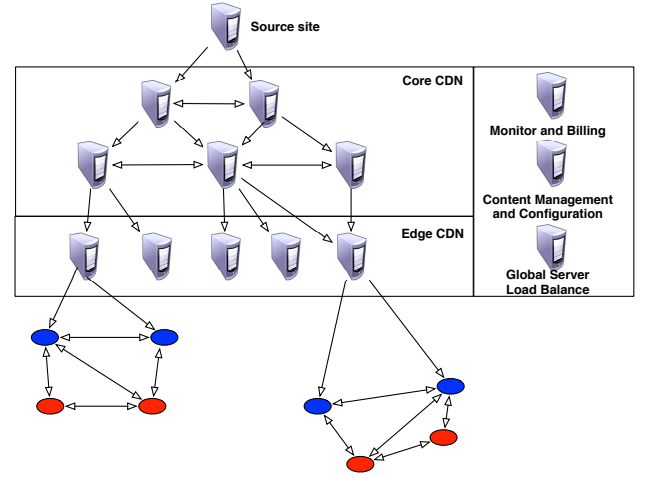


Fig. 1: Example model of live stream content delivery architecture with peer assisted.

largely tree-based. However, to provide greater reliability in the presence of node or network failures, CDN is allowed to retrieve the content either from CDN higher up in the hierarchy or from peer CDN in the same level. Edge CDNs are responsible for serving end users, they are typically heavily loaded. In this system, a peer that is served by edge CDN named as seeders while a peer that is served by seeders named as leechers.

Before accessing the video stream, a peer first obtain the URL from content source. The global load server balance takes into account the peer location, edge CDN location, and edge CDN load to find a suitable edge CDN for this peer. The peer is then redirected to this edge CDN. The edge CDNs are preconfigured with some decision logic that decides if a new peer should be served directly by edge CDN or it should be redirected to P2P overlay.

In P2P transmission overlay, the stream is separated into several substreams according the stream id. For example, if the video is divided into six substreams, substream-0 consists of frames 0, 6, 12, 18, ... , substream-1 consists of frames 1, 7, 13, 19, .. and so on. Peers are organized in a tree-based overlay on a per-substream basis. Each peer has a parent node for each substream and obtain the stream frames belonging to this substream from the parent node. Additionally, in order to be robust to network or node failures, peers also construct a mesh-style overlay to retrieve missing frames for continuous playback. The working environment for peer-assisted CDN live streaming system is defined by several parameters such as: (1) media playback bitrate, (2) the total number of end users, (3) the bandwidth capacity of the edge CDNs, and (5) Characteristics of end users such as their upload bandwidth capacity and churn rates. Given the working environment, the control parameter is the number of end users that are served directly by the edge CDNs. Maximum number of seeders is bounded by maximum CDN's capacity, while maximum

number of leechers are bounded by number of seeders can support the bitrate. Denote number of seeders is n_s , number of leechers is n_l , ρ is maximum bitrate that supplied by seeders to leechers, and $r = 1$ is video bitrate, therefore we have number of leechers that can be supported by seeders is:

$$\lfloor n_l \rfloor = n_s \cdot \rho \quad (1)$$

Number of seeders that support or upload content to leecher is:

$$n_s^u = n_l \cdot \frac{r}{\rho} \quad (2)$$

The illustration as follows, suppose we have video bitrate $r = 1$, seeder upload rate $\rho = 0.25$, and maximum CDN capacity is $643Mbps$. Maximum number of seeders supported by CDN is $n_s = 643$. Maximum number leechers supported by seeders is $n_l = 160$. Number of seeders that upload content to leechers is $n_s^u = 640$. Therefore we have three seeders that do not need to upload content to leecher.

B. Peer Assisted Online Storage

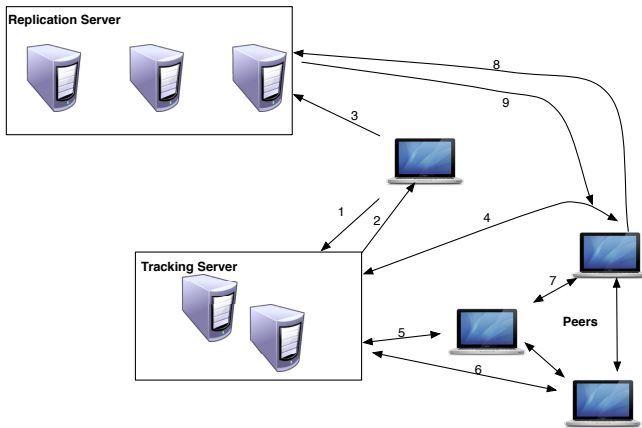


Fig. 2: Example model of peer-assisted online storage architecture.

Figure 2 illustrates the architecture of peer-assisted online storage for file hosting system (one-click hosting services with peer-assistance) [10]. Each file provided by users is treated as a swarm, since it is distributed live on an ongoing basis to other users. Each end user is treated as a peer, inheriting the terminology of pure P2P systems. In fig.2 arrows 1,2,and 3 represent the interaction between a participating peer and dedicated tracking and replication servers for uploading new content. Arrows 4,5, and 6 represent the interaction between peers and the tracking server to maintain the peer topology. Arrow 7 represents the sharing of file blocks and exchange of availability among peers. Arrows 8 and 9 represent peer requests and server response when the requests cannot be satisfied by other peers alone.

Tracking server serves the purpose of maintaining swarm information and bootstrapping peers. Specifically, it maintains a unique swarm ID and a secure hash value for each file provided by a peer and keeps track of the group of users that are participating in the distribution of each swarm. Replication servers serve as dedicated content servers to maintain availability of swarms when they are not actively served by peers alone. All peers involved in the system, i.e., either downloading the file in the system of holding a replica of the file, are organized into a topology, so that block availability information can be exchanged and blocks can be shared among neighboring peers in the topology. When a downloading peer join the swarm, it contacts the tracking server and obtains a list of initial partners (determined by a system) that are randomly selected from peers associated with the same channel. This partnership list is periodically updated and new partners can be added. Partners can be active or inactive, which is determined by whether there are actual connections and data block transfer between the peer and its partners.

Peers need to keep a reasonably number of active and inactive partners in order to maintain a sustainable level of downloading efficiency and to be resilient to network dynamics. Each peer can have up to N partners in its pool of inactive partners, which is related to the number of peers simultaneously being online in a swarm. In case the size of the inactive partner pool increases to over threshold N , a peer will discard aged partners or partners who it has failed to establish a connection with. On the other side, with respect to active partners, the maximum number per swarm that peer can have is bounded by a system parameter K_{max} . Connections to active partners can be broken from time to time due to various reasons, such as slow downloading rates and idling for a long time. Each peer periodically monitors the number of its current partner. If the number of active partners falls below a threshold K_{min} , it exchanges active partner lists with its current active partners through gossip and attempts to establish new connections with a random subset of inactive partners. If successful, the status of these inactive partners will be promoted to active.

To maintain accurate list of peers in each swarm, peers report their status to the tracking server periodically which contains vital peer information such as unique peer identifier, its IP address, and information about swarms that it has joined. Since each peer can be potentially involved in a large number of swarms, to keep overhead low, the reported status only includes information of a bounded number of top C swarms, including swarm identifier and download ratios, defined as the amount of file that has been downloaded so far. The top C swarms represent files that have been downloaded by this peer, ranked by a value computed by the combination of the file size, download ratio, and the time when the peer joined the swarm. Upon receiving status reports from peers, the tracking server periodically updates the corresponding list of peers associated with each swarm. Such a periodic refresh of peer lists in each swarm (associated to each file) assists peers to gain access to active partners that are most helpful, with a reasonable level

of overhead. Consequently, the downloading performance can be improved and the load on servers can be alleviated.

In content distribution strategies, each file is divided into fixed size blocks. A block map (BM) is introduced to specify the availability of blocks at each peer. The periodic exchange of BMs among peers enable them to locate the needed blocks. Each peer can retrieve a number of distinct blocks from multiple active partners simultaneously up to the number of its current active partners. If multiple partners hold a desired block, a peer will randomly choose one of its active partners to request that block. Since server hold important role in this system, we present simple mathematical model of server bandwidth allocation strategies [11], [12] as base for energy calculation later in Sec.IV-C as follows:

type-1 represents less popular files and type-2 represents popular files.

S_{t1} represents server bandwidth allocated to a file type-1 and S_{t2} represents server bandwidth allocated to a file type-2.

S is total server bandwidth.

S_{max1} is the maximum amount of server bandwidth that can be assigned to a file of type-1 and S_{max2} is the maximum amount of server bandwidth that can be assigned to a file of type-2.

M_{t1} is number of type-1 files and M_{t2} is number of type-2 files.

λ_{t1} is the arrival rate of new peers in type-1 file and λ_{t2} is the arrival rate of new peers in type-2 file.

$\lambda = M_{t1} \cdot \lambda_{t1} + M_{t2} \cdot \lambda_{t2}$

$M = M_{t1} + M_{t2}$.

η is the file sharing effectiveness. It is the fraction of the upload capacity of peers that is being utilized.

There are three server bandwidth allocation strategies: (1) lower bound of the average downloading time; (2) request driven strategy; (3) water leveling strategy. In lower bound of the average downloading time, we shall first assign the server bandwidth to type-1 files until S_{t1} reaches its maximum value then the residual server bandwidth can be assigned to type-2 files. This strategy to achieve the lower bound of the system wide average downloading time is also applicable to the general case of multiple types of files: the server should always satisfy the requests from the peers who are downloading the least popular files.

In request driven strategy, the server serves every request from peers. The server bandwidth is equally divided among all the peers. We assume that the number of requests for a file to the server is proportional to the peer arrival rate of the file. We also assume that when the amount of server bandwidth assigned to one of the two types of files has reached its maximum value, the residual server bandwidth will be assigned to the other type of files.

In water leveling strategy, this strategy equalizes the server bandwidth across all the files by taking file popularity into consideration. The server serves the requests from peers according to a certain probability which is inversely proportional to the peer arrival rate of the file. We assume that the number

of requests for a file to the server is proportional to the peer arrival rate of the file, the server will serve the same number of requests for different files and therefore the server bandwidth is equally allocated across all the files.

Average downloading time can be calculated as [12]:

$$T_d = \frac{1}{M_{t1} \cdot \lambda_{t1} + M_{t2} \cdot \lambda_{t2}} \left(\frac{M_{t1} \cdot f_{t1} \cdot \lambda_{t1} \cdot \eta_{t2} + M_{t2} \cdot f_{t2} \cdot \lambda_{t2} \cdot \eta_{t1}}{\mu \cdot \eta_{t1} \cdot \eta_{t2}} - \frac{S_{t1}(M_{t1} \cdot \eta_{t2} - M_{t2} \cdot \eta_{t1}) + S \cdot M_{t2} \cdot \eta_{t1}}{\mu \cdot \eta_{t1} \cdot \eta_{t2}} \right) \quad (3)$$

next, we can calculate number of peers as [12]:

$$\sum x_i = T_d \cdot \sum \lambda_i \quad (4)$$

C. Energy Model

In this paper, our goal is to provide a general view and a fair comparison of the energy consume by a CDN and hybrid CDN-P2P architecture. To do so, we designed a series of model and perform an analysis. Our energy model is similar to the models used in [13]. The difference with [13], firstly, our baseline energy is not function of bitrate flow. Our baseline energy is based on minimum energy required to turn-on the device without any traffic low trough the device. Secondly, our overhead number based on COP value.

For each component, we consider two energy measurement: baseline energy consumption and energy consumed per work unit δ . The baseline energy is the energy consumed to keep the device on. Even when there is no traffic, the device still consume energy. The energy consumption for a single request in CDN server as:

$$E_d = E_s \quad (5)$$

while the energy consumption for a single request in network as:

$$E_r = d_s \cdot E_p \quad (6)$$

and energy of peer or client is E_p .

where d_s is number of hops or the path length. We define δ_s and δ_r work-induced energy consumed per additional bit transferred by a server and router. We can express these per-bit work induced consumptions as follows:

$$\delta_s = \frac{(S_{max} - S_{base})}{M_s} \quad (7)$$

S_{base} is a server's baseline power consumption. S_{max} is a server's power when operating at maximum capacity. M_s is the maximum capacity in bit per second for a server. Same formulation also applied for network component.

$$\delta_r = \frac{(R_{max} - R_{base})}{M_R} \quad (8)$$

R_{max} is a router power when operating at maximum capacity. R_{base} is a router's baseline power consumption. M_R is maximum capacity in bit per second for a router.

Substituting eq.7, and eq.8 to eq.5 and eq.6, we can rewrite eq.5 and eq.6 as follows:

$$\begin{aligned} E_d &= \delta_s.B + E_{s-baseline} \\ E_r &= d_s.\delta_r.B + E_{r-baseline} \\ E_p &= \delta.B + E_{p-baseline} \end{aligned} \quad (9)$$

Finally total energy is:

$$\begin{aligned} E_t &= E_d + E_r + E_p \\ E_t &= \delta_s.B + E_{s-baseline} + d_s.\delta_r.B \\ &\quad + E_{r-baseline} + \delta_p.B + E_{p-baseline} \end{aligned} \quad (10)$$

where B is size of file to be transferred.

We have shown energy model for server, router, and peer. Now, we introduce the overhead for server and router. Overhead that we will consider here is cooling power. Since servers and routers are placed in data center, data center seek to provision the cooling adequately to extract the heat produce by servers, switches, routers, and other devices. Practically, cooling provisioning are done at facilities level. Data center operator operate cooling facilities based on power ratings of servers, switches, routers, and other devices often with some additional headroom for risk tolerance.

The cooling cycle of a typical data center operates in the following way: Cooling units operate by extracting heat from the data center and pumping cold air into the room, usually through a pressurized floor plenum. The pressure forces the cold air upward through vented tiles, entering the room in front the hardware Fans draw the cold air inward and through the server; hot air exits through the rear of the server. The hot air rises sometimes with the aid of fans and a ceiling plenum, and is sucked back to the cooling units. The cooling units force the hot air past pipes containing cold air or water. The heat from the returning air transfer through the pipes to the cold substance. The heated substance leaves the room and goes to a chiller and cooling unit fans force the cold air back into the floor plenum.

The efficiency of this cycle depends on several factors, including the conductive substance and the air flow velocity, but it is quantified by coefficient of performance (COP). The COP is the ratio of heat removed (Q) to the amount of work necessary (W) to remove that heat [14]:

$$COP = \frac{Q}{W} \quad (11)$$

Therefore, the work necessary to remove heat is inversely proportional to the COP. A higher COP indicates a more efficient process, requiring less work to remove a constant amount of heat.

However, the COP for a cooling cycling is not constant, increasing with the temperature of the air the cooling unit pushes into the plenum. COP value empirically can be computed using [14]:

$$COP(T) = 0.0068.T^2 + 0.0008.T + 0.458 \quad (12)$$

Where T is temperature supply by cooling unit.

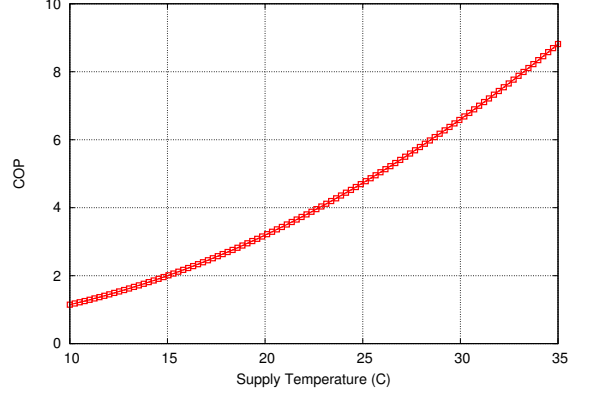


Fig. 3: COP curve for the chilled water cooling units from HP Lab utility data center. As the target temperature of the air the cooling unit pumps into the floor plenum increases, the COP increases [14].

Cooling cost can be calculated as [14] :

$$C = \frac{Q}{COP(T)} \quad (13)$$

Where Q is amount of power the servers and hardwares consume. Currently we assume a uniform T from each cooling units due to the complications introduced by non-uniform cold air supply.

Considering cooling energy overhead, we can substitute eq.13 to eq.10, therefore total energy consumption:

$$\hat{E}_t = E_d \cdot \left(1 + \frac{1}{COP(T)}\right) + E_r \cdot \left(1 + \frac{1}{COP(T)}\right) + E_p \quad (14)$$

We do not include cooling overhead in peer energy consumption because most of peers in home do not need cooling. The air conditioning in home mainly for human not for devices.

IV. RESULT AND DISCUSSION

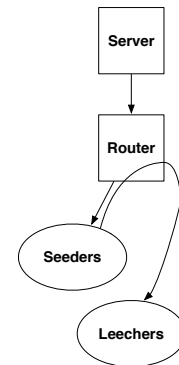


Fig. 4: Physical representation of the network.

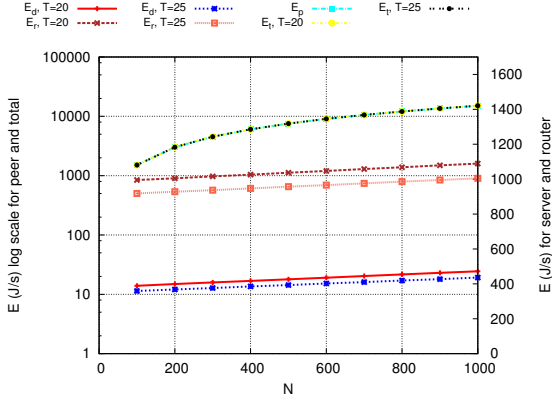


Fig. 5: Power consumption in CDN architecture.

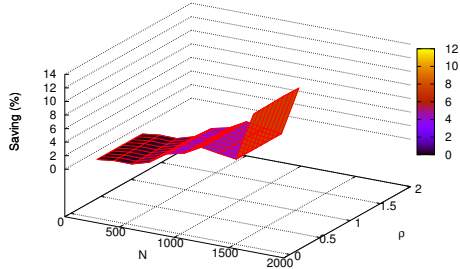


Fig. 7: Different power consumption between CDN and CDN-P2P for CDN server component with $\rho = 0.25, 0.5, 0.75, 1$.

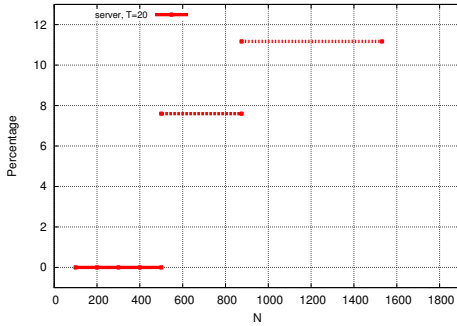


Fig. 9: Different power consumption between CDN and CDN-P2P for CDN server component with $\rho = 0.75$.

A. Numerical Parameters

The parameters used in this numerical simulation were adapted from [13], [8], [4] and [12]. Parameters values is shown in table I. The parameters've description as follow:

Server Parameter - S_{base} and δ_s : To measure the per bit energy for servers, we use a study from [13]. The author found that an idle power draw of 290 watt for the server, that are the baseline power consumptions $E_{s-baseline}$. To measure

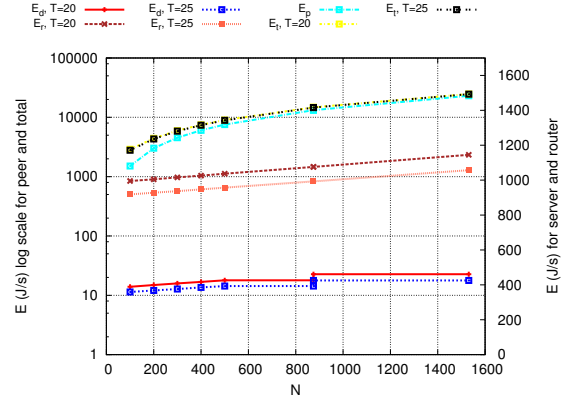


Fig. 6: Power consumption in CDN-P2P architecture with $\rho = 0.75$.

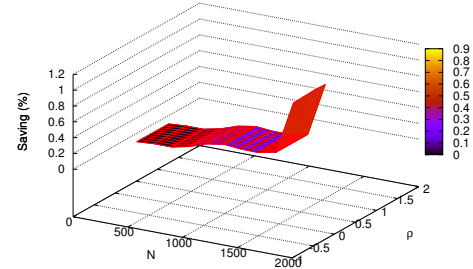


Fig. 8: Different power consumption between CDN and CDN-P2P for total system with $\rho = 0.25, 0.5, 0.75, 1$.

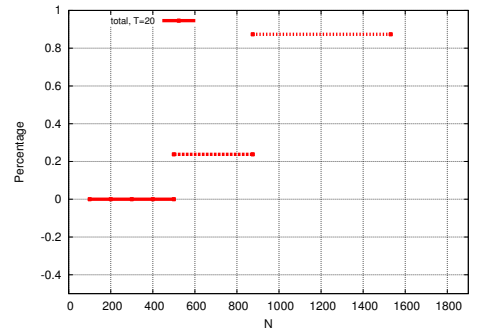


Fig. 10: Different power consumption between CDN and CDN-P2P for total system with $\rho = 0.75$.

work induced power consumption the author repeatedly downloaded files from two machines using a number of clients and httpperf benchmarking tool. The author measured the average power draw, network throughput and CPU utilization at maximum capacity. These measurement allows the author to calculate δ_s using eq.7.

Peer Parameter - P_{base} and δ_p : we use a study from [8]. The author measured power consumption through a specialized energy measurement device by varied throughput

TABLE I: Numerical Simulation Parameters from [13], [8],[4], and [12].

Symbol	Value
δ_s	$5.2 \cdot 10^{-8}$ (J/b)
δ_r	$8.0 \cdot 10^{-9}$ (J/b)
δ_p	$16 \cdot 10^{-9}$ (J/b)
$E_{r-baseline}$	750 watt
$E_{s-baseline}$	290 watt
$E_{p-baseline}$	13.5 watt
d_s	1
N_s^u	[0.25, 0.5, 0.75, 1] Mbps
N	[100, ..., 1000]
λ_{t1}	0.1
λ_{t2}	1
η_{t1}	0.5
η_{t2}	1
M_{t1}	10
M_{t2}	1
$f_{t1} = f_{t2}$	100 MB
μ	0.5 Mbps
c	1Mbps
T	[20, 25] correspond to COP value [3.194, 4.728]

the downloaded video streams.

From this measurement we can get $E_{p-baseline}$ and δ_p . **Router Parameter** - R_{base} and δ_r : we use a study from [4] to derive router parameter. A recent measurement from [4] reports idle power draw of Cisco GSR router $E_{r-baseline} = 750$ watt. When routing at 2.5 Gbps, the study reports an increase of at most 20 watt for a per-bit energy increase $8 \cdot 10^{-9}$ J/bit.

N_s^u is upload rate that we will use in live streaming case.

N is number of peers that we will use in live streaming case. d is number of hops for live streaming case and online storage case.

λ_{t1} is peer arrival rate to less popular files in online storage case.

λ_{t2} is peer arrival rate to popular files in online storage case. η_{t1} and η_{t2} is file sharing effectiveness. It is the fraction of the upload capacity of peers that is being utilized in online storage case.

μ is upload rate of peers in online storage case. We assume it is uniform.

c is downloading rate of peers in online storage case. We assume it is uniform. M_{t1} is number of files in type-1 files or less popular files and M_{t2} is number of files in type-2 files or popular files in online storage case.

$f_{t1} = f_{t2}$ is file size in online storage case.

T is air temperature supply from cooling in data center. $T = 20$ corresponds to $COP = 3.194$ and $T = 25$ corresponds to $COP = 4.728$.

B. Live Streaming Case

In live streaming case, video stream flows from CDN to network in this case router then arrive in seeders. If seeders need to upload data to leechers then, it will flows through router then arrive in leechers. We showed the logical network architecture of peer-assisted CDN for live streaming in fig.1.

Figure.4 shows simplified the physical representation of the network. While in logical network the peer can communicate with another peer, in real physical world the communication between peer always through router inside data center.

Figure 5 shows energy usage for CDN server, router, total energy consumption for CDN scenario (without peer assisted). We plot energy consumption for CDN server, router, clients, and total energy for every T or COP coefficient value. In CDN only scenario, the energy consumption is increase as number of client increase. This is natural because increasing of clients imply to increasing of traffic from CDN server through router to clients. The effect of different T to total energy is small.

Figure 6 shows energy consumption for CDN, router, and total energy consumption for CDN-P2P scenario. We show peer upload rate $N_s^u = 0.75$ in fig.6. We observe that peer power consumption is increasing as number of peers increase. Router power consumption is also increasing as additional number of peers generate more traffic. CDN server power consumption on the other hand relatively flat because some of seeders can supply video stream to some leechers.

Figure 7 shows percentage difference of energy consumption between CDN and CDN-P2P for CDN server component with $\rho = [0.25, 0.5, 0.75, 1]$. The power saving that CDN server get is from 3.8% to 11%. More clear figure is shown in fig.9 that we sliced for $\rho = 0.75$ from fig.7. CDN server can save energy because the utilization of CDN server is same when the system has number of peers increasing from 500 to 875. Same situation is also happen when number of peers increasing from 875 to 1531. The saving is up to 11%

Figure 8 shows percentage difference of energy consumption between CDN and CDN-P2P for total system with $\rho = [0.25, 0.5, 0.75, 1]$. The power saving that total system get is from 0.2% to 0.9%. Figure 10 reports percentage difference of energy consumption between CDN and CDN-P2P for total system with $\rho = 0.75$. While CDN server part can save more energy 9, due to peers high energy consumption then total energy saving in the system that we can get is small, from 0.2% to 0.9%. For fig.7, 8, 9, and 10, we only show graph for $T = 20$ since the difference with $T = 25$ is very small as shown in fig.6.

C. Online Storage Case

In peer-assisted online storage case, numerical simulation parameters that we used in this section same as [12]. Its parameters explanation and values are shown in sect.IV-A.

Figures 11, 12, and 13 show power consumption for every bandwidth strategy. From fig.11, 12, and 13 we found that increasing number of peers can decrease CDN server power consumption because CDN server can reduce bandwidth or work due to increasing of peers that exchange blocks. We also found that router power consumption relatively flat. This is because compensation of server bandwidth reduction by increasing number of peers. Figure 14 shows comparison energy consumption between request driven to lower bound strategy and water-leveling strategy to lower bound strategy.

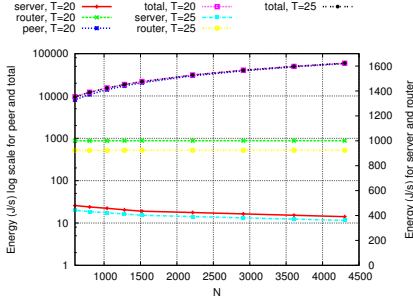


Fig. 11: Power consumption for lower bound strategy. Left axes in log scale for peers and total power consumption. Right axes in linear scale for router and CDN server.

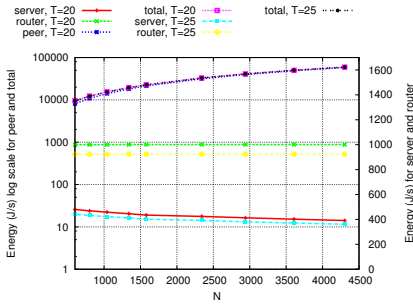


Fig. 13: Power consumption for water-leveling strategy. Left axes in log scale for peers and total power consumption. Right axes in linear scale for router and CDN server.

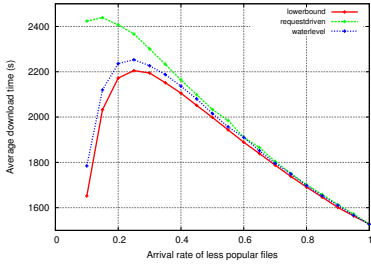


Fig. 15: Comparison of average downloading time under different server bandwidth allocation strategies when the peer arrival rate of less popular files varies.

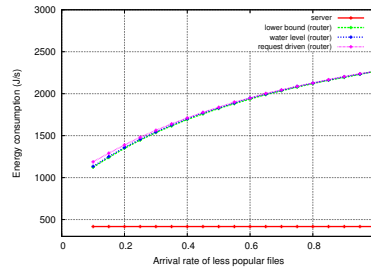


Fig. 16: Power consumption for server and router under different server bandwidth allocation strategies when the peer arrival rate of less popular files varies. We use server bandwidth $S = 50\text{MBps}$.

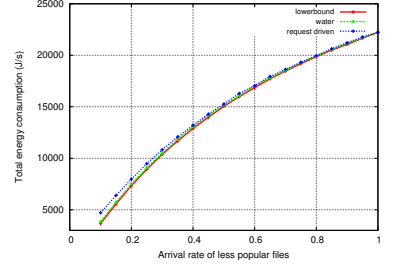


Fig. 17: Total power consumption under different server bandwidth allocation strategies when the peer arrival rate of less popular files varies. We use server bandwidth $S = 50\text{MBps}$.

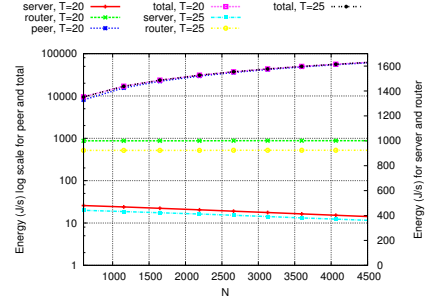


Fig. 12: Power consumption for request driven strategy. Left axes in log scale for peers and total power consumption. Right axes in linear scale for router and CDN server.

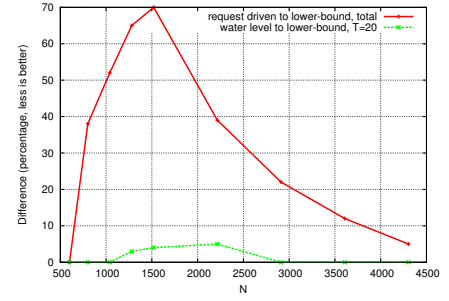


Fig. 14: Power consumption difference between request driven and water-leveling strategy to lower-bound strategy.

Compare to water-leveling strategy, request driven strategy required more energy than water-leveling strategy because request driven strategy equalizes the server bandwidth across all the peers. Water-leveling strategy equalizes server bandwidth across all the files by taking file popularity into consideration thus minimize downloading time. This minimization result near to lower-bound strategy.

File popularity has strong correlation with downloading performance. We examine popularity by varying the peer arrival rate of less popular files. Specifically we linearly increase file

popularity from 0.1 to 1 and we choose server bandwidth $S = 50\text{MBps}$ which is quite constraint and similar to FS2You case. As shown in fig.15 when file popularity increases, downloading performance decrease first and increases again as files become more popular. When file popularity is low there are fewer peers downloading the file and hence each of them can obtain relatively more sever bandwidth. When file popularity is high peers can enjoy high file sharing effectiveness and hence high average downloading rates. Semi popular files get the worst downloading performance in the system as difficulty in

both file sharing among peers and requesting bandwidth from server. In request driven strategy, we do not find such as U-shaped curve due to server bandwidth strategy that is equally divided across all peers and not affected by file popularity.

Figure 17 shows total energy of system as function from file popularity. More popular files imply having more peers in the system thus increases total energy. While CDN server energy consumption is constant, router energy consumption is increases fig.16.

V. RELATED WORK

Content Distribution Networks with peer assist have been successfully deployed on the Internet, such as Akamai [15] and LiveSky [9]. The authors of [15] conclude from two real world traces that hybrid CDN-P2P can significantly reduce the cost of content distribution and can scale to cope with the exponential growth of Internet video content. Yin et al. [9] described LiveSky as commercial operation of a peer-assisted CDN in China. LiveSky solved several challenges in the system design, such as dynamic resource scaling of P2P, low startup latency, ease of P2P integration with the existing CDN infrastructure, and network friendliness and upload fairness in the P2P operation. Measurement from LiveSky showed that LiveSky can save bandwidth around 40% [9]. The author in [16] and [17] proposed mechanisms to minimize CDN server bandwidth to make the content distribution cheap. They designed different peer prefetching policies of video on demand system in surplus mode while ensuring user quality of experience. A similar analysis has been done in [18] for live video streaming system where the authors proposed different limited peer contribution policies to reduce CDN bandwidth requirement and eventually off the distribution process from CDN to P2P system. An ISP friendly rate allocation schemes for a hybrid CDN-P2P video on demand system in [19]. These technique try to minimize CDN server bandwidth while reducing ISP unfriendly traffic and maximizing peer prefetching. Load on CDN server has been shown to be reduced using this approach while reducing cross ISP traffic. Above studies were performed for video on demand or live video streaming. While video is the most popular content, the systems can be also for other type contents. Moreover while content based services are growing, energy consumption of a content distribution system has not been analyzed.

Related to CDN and energy usage, in a seminal work [20], the authors show that if costs are based on electricity usage and if the power prices vary in real-time, global load balancing decision can be made such that users are routed to locations with the cheapest power without significantly impacting user performance or bandwidth cost. In [21], the author proposed utilizing batteries for CDN for reducing total supplied power and total power costs. The authors in [21] also proposed battery provisioning algorithms based on workload of CDN server. The result shows that batteries can provide up to 14% power savings.

The idea that utilize ISP controlled home gateway to provide computing and storage services and adopts managed peer-

to-peer model is presented in [8]. Valancius et al. [8] show that distributing computing platform in NaDa (Nano Data Center) save at least 20-30% energy compare to traditional data centers. The saving in NaDa comes from underutilizing home gateway, avoidance of cooling costs, and the reduction of network energy consumption as a result of demand and service co-localization.

The comparison between CDN architecture and peer-to-peer architecture are discussed in [6] and [5]. Both authors in [6] and [5] agree that CDN architecture is more energy saving compare to peer-to-peer architecture. Another interesting study of energy efficient in content delivery architectures is presented by Guan et al. [22]. by Guan et al. [22] comparing energy efficient of CDN architecture and CCN architecture. The authors in [22] conclude that CCN is more energy efficient in delivering popular content while the approach with optical bypass is more energy efficient in delivering infrequent accessed content.

To the best of our knowledge, the study of energy in that considering peer-to-peer in CDN architecture is presented in [23]. Mandal et al. [23] mentioned that hybrid CDN-P2P systems can reduce a significant amount energy in the optical core network around 20-40% less energy. The authors only considered energy consumption of hardware especially optical devices and does not include overhead inside data center, CDN server energy consumption, and consumed power by peers. Our work is quite different, we take number of peers and add overhead of data center which is power of cooling cause by temperature of hardware for different purpose: live streaming and online storage.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have introduced the comparison of energy consumption between the peer-assisted CDN architecture and CDN architecture for live streaming and video on demand service. Integrating peer to peer capability to assist the existing CDN has a potential to save energy consumption. In this study, we show that event without explicitly considering energy consumption while assigning content, the peer assisted CDN can save energy consumption. Although the energy savings depend on number of request (number of clients), number of router and its configuration, for total system energy saving is around 0.5 to 1.2. If we break per component in live streaming case, the CDN server is the part that can be pushed to save energy up to 11% and can be more if new generation of power proportional server is used [24]. Unfortunately, in peer-assisted online storage, we do not see such energy saving. In peer-assisted storage, all of strategies that we discussed before focus how to efficiently allocated bandwidth from CDN server for every peers. CDN server is fully used to serve peers, thus we do not see energy saving in CDN server. We agree with [4], Router component in the other side is quite difficult to optimize for energy saving, because different chassis size, different network interface type slot, and different configuration can lead to different energy consumption. Different COP values, in our case $T = 20$ and $T = 25$, only give

7% difference power consumption. Obviously, with $T = 25$ overhead is minimal compare to $T = 20$. Once again, It is not our aim to advocate air cooling supply temperature value over another since there is reliability issue related to hardware. We only show that increasing temperature can minimize overhead. Several areas that we have been identified for future work are: more correlation analysis of time period to peer energy usage pattern in live streaming, continued characterization of different peer energy usage based on flash memory storage, and comparing energy model with different file popularity models.

ACKNOWLEDGEMENTS

We thank A,B,C, and D

REFERENCES

- [1] U. Institute, "the 2012 uptime institute data center industry survey," <http://uptimeinstitute.com/2012-survey-results>.
- [2] Google, "Google data center power usage," <http://www.google.com/corporate/datacenters/measuring.html>.
- [3] L. Barroso and U. Holzle, "The case for energy-proportional computing," *Computer*, vol. 40, no. 12, pp. 33–37, 2007.
- [4] J. Chabarek, J. Sommers, P. Barford, C. Egan, D. Tsang, and S. Wright, "Power awareness in network design and routing," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, 2008, pp. 457–465.
- [5] A. Feldmann, A. Gladisch, M. Kind, C. Lange, G. Smaragdakis, and F.-J. Westphal, "Energy trade-offs among content delivery architectures," in *Telecommunications Internet and Media Techno Economics (CTTE), 2010 9th Conference on*. IEEE, 2010, pp. 1–6.
- [6] J. Baliga, K. Hinton, and R. S. Tucker, "Energy consumption of the internet," in *Optical Internet, 2007 and the 2007 32nd Australian Conference on Optical Fibre Technology. COIN-AOFT 2007. Joint International Conference on*. IEEE, 2007, pp. 1–3.
- [7] M. Dischinger, A. Haeberlen, K. P. Gummadi, and S. Saroiu, "Characterizing residential broadband networks," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, ser. IMC '07. New York, NY, USA: ACM, 2007, pp. 43–56. [Online]. Available: <http://doi.acm.org/10.1145/1298306.1298313>
- [8] 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*. ACM, 2009, pp. 37–48.
- [9] H. Yin, X. Liu, T. Zhan, V. Sekar, F. Qiu, C. Lin, H. Zhang, and B. Li, "Livesky: Enhancing cdn with p2p," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 6, no. 3, pp. 16:1–16:19, Aug. 2010. [Online]. Available: <http://doi.acm.org/10.1145/1823746.1823750>
- [10] Y. Sun, F. Liu, B. Li, B. Li, and X. Zhang, "Fs2you: Peer-assisted semi-persistent online storage at a large scale," in *INFOCOM 2009, IEEE*, 2009, pp. 873–881.
- [11] S. Das, S. Tewari, and L. Kleinrock, "The case for servers in a peer-to-peer world," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 1, 2006, pp. 331–336.
- [12] Y. Sun, F. Liu, B. Li, and B. Li, "Peer-assisted online storage and distribution: modeling and server strategies," in *Proceedings of the 18th international workshop on Network and operating systems support for digital audio and video*, ser. NOSSDAV '09. New York, NY, USA: ACM, 2009, pp. 13–18. [Online]. Available: <http://doi.acm.org/10.1145/1542245.1542249>
- [13] S. Nedevski, S. Ratnasamy, and J. Padhye, "Hot data centers vs. cool peers," in *Proceedings of the 2008 conference on Power aware computing and systems*, ser. HotPower'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 8–8. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1855610.1855618>
- [14] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling cool": temperature-aware workload placement in data centers," in *Proceedings of the annual conference on USENIX Annual Technical Conference*, 2005, pp. 5–5.
- [15] C. Huang, A. Wang, J. Li, and K. W. Ross, "Understanding hybrid cdn-p2p: why limelight needs its own red swoosh," in *Proceedings of the 18th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, ser. NOSSDAV '08. New York, NY, USA: ACM, 2008, pp. 75–80. [Online]. Available: <http://doi.acm.org/10.1145/1496046.1496064>
- [16] C. Huang, J. Li, and K. W. Ross, "Can internet video-on-demand be profitable?" *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 133–144, Aug. 2007. [Online]. Available: <http://doi.acm.org/10.1145/1282427.1282396>
- [17] C. Huang, J. Li, and K. Ross, "Peer-assisted vod: Making internet video distribution cheap," in *Proc. of IPTPS*, vol. 7, 2007.
- [18] D. Xu, S. S. Kulkarni, C. Rosenberg, and H.-K. Chai, "Analysis of a cdn-p2p hybrid architecture for cost-effective streaming media distribution," *Multimedia Systems*, vol. 11, no. 4, pp. 383–399, 2006.
- [19] J. Wang, C. Huang, and J. Li, "On isp-friendly rate allocation for peer-assisted vod," in *Proceedings of the 16th ACM international conference on Multimedia*, ser. MM '08. New York, NY, USA: ACM, 2008, pp. 279–288. [Online]. Available: <http://doi.acm.org/10.1145/1459359.1459397>
- [20] A. Qureshi, R. Weber, H. Balakrishnan, J. Gutttag, and B. Maggs, "Cutting the electric bill for internet-scale systems," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 123–134, 2009.
- [21] D. S. Palasamudram, R. K. Sitaraman, B. Ugaonkar, and R. Ugaonkar, "Using batteries to reduce the power costs of internet-scale distributed networks," in *Proceedings of the Third ACM Symposium on Cloud Computing*, ser. SoCC '12. New York, NY, USA: ACM, 2012, pp. 11:1–11:14. [Online]. Available: <http://doi.acm.org/10.1145/2391229.2391240>
- [22] K. Guan, G. Atkinson, D. Kilper, and E. Gulsen, "On the energy efficiency of content delivery architectures," in *Communications Workshops (ICC), 2011 IEEE International Conference on*, 2011, pp. 1–6.
- [23] U. Mandal, C. Lange, A. Gladisch, and B. Mukherjee, "Should isps adopt hybrid cdn-p2p in ip-over-wdm networks: An energy-efficiency perspective?" in *Advanced Networks and Telecommunications Systems (ANTS), 2012 IEEE International Conference on*, 2012, pp. 6–11.
- [24] A. Krioukov, P. Mohan, S. Alspaugh, L. Keys, D. Culler, and R. Katz, "Napsac: design and implementation of a power-proportional web cluster," *SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 1, pp. 102–108, Jan. 2011. [Online]. Available: <http://doi.acm.org/10.1145/1925861.1925878>