Enabling Green Content Distribution Network by Cloud Orchestration

Yahav Biran*
Microsoft
Redmond, WA, USA
*ybiran@microsoft.com

Sudeep Pasricha†
Department of Electrical and Computer Engineering†
Colorado State University
Fort Collins, CO, USA
†sudeep@colostate.edu

George Collins‡
Department of Systems Engineering†
Colorado State University
Fort Collins, CO, USA
‡gcollins@colostate.edu

Joel Dubow§
Fulcrum Co.
Centreville VA
§jdubow@fulcrumit.com

***Abstract*—Cloud providers seek to maximize their market share. Traditionally, they deploy datacenters with sufficient capacity to accommodate their entire compute demand while maintaining geographical affinity to its customers. Achieving these goals by a single cloud provider is increasingly unrealistic from a cost of ownership perspective. Moreover, the carbon emissions from underutilized datacenters place an increasing demand on electricity, and is an increasing factor in the cost of cloud provider datacenters. However, the recognition that on-demand video streaming now constitutes the bulk portion of traffic to Internet consumers provides a path to mitigate rising energy demand. On-demand video is usually served through Content Delivery Networks (CDN), often scheduled in backend and edge datacenters. This publication describes a CDN deployment solution that utilizes green energy to supply on-demand streaming workload. A cross-cloud provider collaboration will allow cloud providers to both operate near their customers and reduce operational costs, primarily by lowering the datacenter deployments per provider ratio. Our approach optimizes cross-datacenters deployment. Specifically, we model an optimized CDN-edge instance allocation system that maximizes, under a set of realistic constraints, green energy utilization. The architecture of this cross-cloud coordinator service is based on Ubernetes, an open source container cluster manager that is a federation of Kubernetes clusters. It is shown how, under reasonable constraints, it can reduce the projected datacenters carbon emissions growth by 22% from the currently reported consumption.**

***Index Terms*—Data center networks; Computer systems organizations; Cloud computing; Dynamic adaptation; Scheduling; energy efficiency; energy consumption, Content Delivery Networks**

I. Introduction

Over the past decade, cloud-based systems have been required to serve an increasing demand from users work flows and data. Cloud-based systems may be classified into two categories: *serving* systems and *analytical* systems. The former provides low-latency read/write access to data. For example, a user requests a web page to load online or requests video or audio streaming. The latter provides batch-like compute tasks that process the data offline that are later sourced to the serving systems. The service level objectives (SLO) for serving jobs are on the order of fractions of a second, while the SLO for analytical jobs are on the order of hours, sometimes days.

Today, public cloud service providers (CSP) attempt to process both of these workloads with a rich platform that guarantees cost and SLO to their clients. Cloud computing is an emerging infrastructure with limited regulation and compliance requirements [1]. Recently the Office of Management and Budget issued a Federal Data Center Optimization Initiative that promotes increasing use of Green Energy and increased utilization efficiency for all US Federal datacenters [3]. Specific target numbers are set for the end of fiscal year 2018. This publication addresses how those federal requirements may be attained and how federated cloud computing is a key enabler for attaining those performance targets.

Beginning in 2013, the US government initiated a carbon-tax on IT organizations to encourage major CSPs to pursue green energy opportunities for their datacenters operations [2]. US datacenters are projected to consume approximately 73 billion kWh by 2020 [2] with a corresponding increase greenhouse gases. Green energy generation growth is expected to triple by 2040 [4]. However, there is no cohesive system existing to coordinate the rising datacenter energy demand with rising green energy supply. This work suggests a multi-cloud resource coordination system that matches computational resource demands with available energy supply to maximize the utilization of green energy for processing cloud-workloads.

Most CSPs seek more market share in competition with other CSPs. One outcome of such competition is an ever-growing infrastructure in the form of new datacenters across the globe with no countervailing forces to meet user demand more efficiently and satisfy societal environmental and energy requirements. This sub optimum use of infrastructure increases the carbon footprint attributable to cloud computing services and also drives up costs to CSP's.

On-demand streaming constitutes up to 85% of Internet traffic consumption [7]. On-demand streaming content is managed and distributed by content service providers. It then cached and distributed by Content Delivery Networks(CDN) located at the edges of the Internet network close to the consumers. Because streaming constitutes such a large fraction of Internet resource consumption, this paper will, of necessity, focus on methods to employ green energy to better operate CDN instances of on-demand streaming jobs, which include both video and audio content.

Meeting the Federally mandated approach of maximizing the utilization of green energy to operate CDN instances (for government with recommendations for private sector use as well) requires an energy source-demand coupling scheme that insures SLO levels of power availability but is structurally biased towards green energy sources over hydrocarbon fueled energy sources. A system to accomplish this will have to

978-1-5090-4476-4/16/$31.00 ©2016 IEEE

provide seamless failover in the case of sudden interruption of green energy to grid-energy sources or vice versa i.e., fallback from grid-energy to green energy when surplus green energy is available.

User expectations in on-demand streaming requires different service level requirements than other serving systems workloads. Serving systems workloads are comprised of interactive sessions that pivots on minimum latency. However, low latency is less critical in analytical on-demand streaming since application clients use buffering techniques to mitigate long latency effects. Therefore, on-demand streaming workloads fits, more closely than interactive workloads, with the observed intermittent and varying green energy availability characteristics.

Green energy supply is unpredictable and requires a complex, adaptable, resource allocation system to provide CDN services with steady energy supplies while concurrently seeking minimal carbon footprint. This dynamic availability of green energy resources in a smart grid requires real time communication of both short term and predictive energy needs from cloud service providers to green energy providers. The green energy providers need to disclose availability dynamically to CSPs, who, in turn, disclose their changing energy demands for near term computing. SPs can then better maximize the use of green energy for on-demand streaming processing.

This is a classical resource management and coordination problem [15]. Our approach builds upon prior work that was done in this area [6], [13], [14], specifically that done on alleviating the sudden lack of green energy to meet low-latency workloads. Our approach employs an application-buffering scheme that better allows for opportunistic, green, on-demand streaming processing. It requires an extended, cohesive, federated system that aggregates supply and demand across multiple geographic locations employing the smart grid command and control infrastructure to achieve an optimal dynamic matching of green energy sources and computing loads.

This paper proposes an implementation that utilizes a control component in a federated cloud that coordinates and optimizes the resource allocation among the participant CDN providers. It treats the volatile nature of green energy resources as a resource allocation problem, the solution of which is a resource orchestration system that is optimized with the goal to operate increasingly near to the limit of supply by green energy sources constrained by SLO reliability requirements. This system will be demonstrated by modeling a prototype that simulates resource allocation in a micro federated cloud eco-system to achieve an energy supply-computation demand match optimized within seconds

Since the focus of this work is on green energy utilization in a federated cloud, the scheduling algorithms and resource management issues, while important, are discussed only to the extent necessary to help the reader understand the required architecture for heterogeneous energy compute clustering. This work is meant both as a case study in energy utilization, and a presentation of a novel method of coordinating high-velocity data streams, and extended to a unified orchestration system, to optimize the performance of federated cloud systems.

The paper starts, in Section II, with the on-demand streaming economics, increased green energy utilization and anticipated smart grid progressions as applied to on-demand streaming. In Section III, the the green energy utilization problem is analyzed. Finally, we present a cloud coordinator prototype that is built on Kubernetes, an open source cluster manger [29] (Section IV), and extend that prototype to discusses the need for and requirements of a unified system that orchestrates cluster compute resources in a federated cloud (Section V and VI).

II. On-Demand Streaming and CDN

Over the last decade, video and audio traffic became the dominant segment of consumer internet traffic.. Cloud service-providers such as Netflix, Amazon Instant and YouTube disrupted the prior linear TV data distribution model. Also, video streams delivered by mobile terminals grew as mobile connectivity improved [10]. Video streaming is expected to constitute up to 85% of Internet consumers traffic [7] within a few years. The US portion of video streaming is 14% [11] and the number of US Unique IPv4 connected addresses is 17% [10]. The streaming workload is comprised of live streaming and on-demand streaming, with the relative fractions of 6% and 94% respectively [7]. Other predictions support similar ratios, 12% live-streaming and 88% for on-demand [11].

A key driver for the rapid expansion of streaming video was the shift from specialized streaming protocols and infrastructures such as RTSP, and RTMP [20] to a simple HTTP progressive download protocol. This led to a shift from proprietary streaming appliances to commodity servers. In turn, this change removed a barrier for CDN's to process on-demand workloads. Most present day, CDN service providers support a seamless integration with cloud-based object storage that pipelines the digital content from the organization site to the CDN instance that runs at the Internet provider edge [21]. Furthermore, the HTTP chunk-based streaming protocol support in a CDN allows the client application sufficient time to detect the optimal CDN instance to handle user workload. The optimal CDN instance assignment is done by the cloud control plane resource manager. The prototype described in Section IV will demonstrate such optimal resource allocation.

We used server utilization and power metrics from [2], [10], [11] to design the prototype. Most of these sources we considered have limited utilization rates and server utilization distribution. Also, utilization and power consumption do not scale linearly [23]. However, for clarity, in the interest of maintaining focus on the larger goal of the paper, CDN resource management systems for green energy utilization, we assumed linear relationships and accepted the risk of loss of accuracy in our estimates. High fidelity simulation accuracy is not critical for the goal of this paper.

A. Energy Saving Potential in Operating a Distributed CDN Resource Management System

The approach is to aggregate the required traffic for on-demand workload processing, and use standard compute device specification to assess the electrical energy and carbon footprint that will be required by that workload [22].

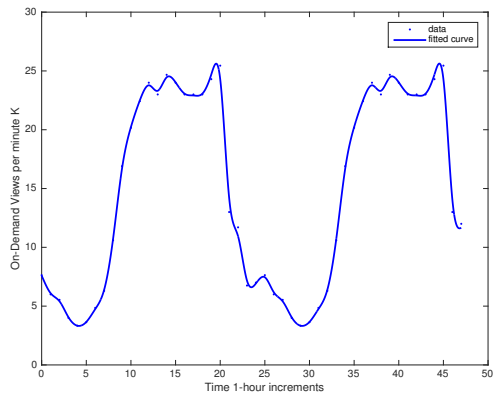

Fig. 1: **On-Demand Video views observed throughout 48 hours with 1-hour increments. Data was fitted with smoothingspline for curve and surface fitting [7]**

The estimated data rate for streaming is given as $S_{total} = 63000PB/mo$ [11] (where PB is Petabytes). Figure 1 shows users workload pattern of 9 busy hours in which the workload spans throughout 13 hours a day which yield $126PB/sec/mo$ [11]. The on-demand streaming portion is estimated as 78% across four main US regions denoted by $k = 4$, the number of region used, for the purposes of this paper although k can be varied depending upon the degree of granularity desired in the simulations. $S_{on-demand}$ denote the on-demand portion.

$$S_{on-demand} = S_{total} \cdot 78\% = 49140PB/mo$$

$n_{hours/day}$ denotes the number of effective hours in a day for streaming.

$$n_{hours/day} = n_{busy} + k = 13$$

$$D_{rate} = \frac{S_{on-demand}}{n_{hours/day}} = 126PB/sec/mo$$

N_{max} denotes the estimated number of required servers in maximum CPU capacity. N_{max} is bounded by the maximum network throughput a single server can ingest. Standard commodity servers can handle up to 8.5Gbps i.e. 1.026GB/sec.

$$N_{max} = \frac{D_{rate}}{T_{max}} = \frac{126 \cdot 10^6 GB/sec}{1.026GB/sec} = 118588235.3$$

u_{opt} denotes the CPU utilization factor so servers has sufficient capacity to handle management tasks. We estimate 60% utilization factor $u_{opt} = 100/60$.

$$N_{opt} = N_{max} \cdot u_{opt} = 118588235.3 \cdot \frac{100}{60} = 197647058.8$$

E_s denotes the midrange server energy consumption for various server types s. We consider three types of servers:(1) compute server(5kWh/server), (2) digital storage server(1.7kWh/server) and (3) network server(1kWh/server). Storage server acts the digital storage controller. The network server acts as the router and switch. The compute server is the server that processes the on-demand streaming.

$$E_y = N_{opt} \cdot \sum\nolimits_{s \in S} E_s \quad (1)$$
$$= N_{opt} \cdot (5kWh + 1.7kWh + 1kWh)$$
$$= 1.521 \cdot 10^6 kWh/mo = 18.26GWh/y$$

The saving potential from running on-demand video streaming using green energy resources is 18.26GWh a year based on current on-demand consumption and expected to grow 89% by 2019 [8]. i.e. 34.5GWh per year for on-demand streaming. The next sections will explain the challenges in utilizing green energy followed by a method that that addresses some of these challenges and thereby maximizes the utilization of green energy.

III. Why is Green Energy Utilization Hard?

The following section describes why utilizing green energy for compute purposes, while a justifiable goal, is limited by SLO reliability. It will present a scenario where balancing time-varying energy generation patterns with changing dynamic energy demands of cloud computing sometimes conflict. The green energy time varying generation patterns considered by us focuses on wind and solar generation. Figure 5 show historical data on dynamic nature of green energy sources and Figure 6 shows the dynamic cloud energy demand.

A. Frequency Stability in Wind-Power Generation

The daily wind power variation characteristics will be employed as a metric that illustrates the duration and level for a given amount if wind energy availability. The electricity generation process from wind is comprised of a wind turbine extracting a kinetic energy from the air flow. The wind is rarely steady; it is influenced by the weather system and the ground surface conditions, which are often turbulent [17]. Also, the generation process must happen at the same instant it is consumed [18] unless it is stored in grid level battery banks. Unfortunately, grid level energy storage technology is not keeping up with grid level energy generation technology.

Sample wind and power generation data were obtained from NREL [16]. We used datasets from 2006-2012 across different regions in the US and aggregated more than 600 observations. Finally, the data were fit using smoothing splines [27]. The usable power generated from a wind turbine is generally described by a Rayleigh distribution [17]. It defines three main points in the wind power generation process: (1) the cut-in is the minimum viable wind speed for electricity generation from a wind turbine. (2) the rated level, describes the point where the power reached its local maximal capacity without adverse effects on the turbine life by too strong a wind. (3) Cut-Off, is the term for the local minimum for the generation

cycle. Beneath that speed, there is not enough power for viable electricity generation. Thus, if the wind velocity is too low, the data-server gets no wind energy. Figure 2 shows wind generation variations that crudely fit a Rayleigh distribution with $b = 300$ assuming the form of the Rayleigh Probability Distribution Function is:

$$f(x|b) = \frac{x}{b^2} e^{\frac{-x^2}{2b^2}}$$

The measured generation cycles range between 140 and 180 minutes per cycle. Equation 2 expresses the generated power by a wind turbine, given a wind velocity. The function g describes a viable electricity generation given a wind power. The wind power availability indications will be generated by a wind turbine and fed into the coordinator database as a potential power source to datacenters in a region. Our prototype will assume wind power availability indications as the wind tuple {*region,cut-in,rated,cut-off*} indications.

$$P_{output}(wind_v) = \begin{cases} 0 & : \text{ if } wind_v \leq rated \\ g(wind_v) & : \text{ if } rated < wind_v \leq c_{out} \\ 0 & : \text{ if } wind_v \leq c_{out} \end{cases} \quad (2)$$

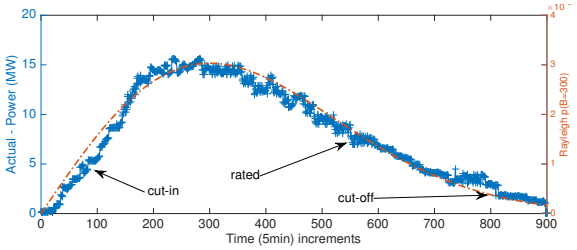

Fig. 2: **Aggregated wind power measurements between 2007-2012 that fits Rayleigh Distribution with b=300 [16]**

B. Efficiency and Daily/Hourly Availability in Solar-Power Generation

Photovoltaic solar (PV) energy availability is defined by the solar power intensity denoted by s(*Watts/m2*), which varies with local daylight hours and the clear or cloudy sky conditions [18]. Moreover, the PV cells are most effective at lower temperatures [19]. The PV cells electrical power generation, defined by Equation 3, is a function of the solar intensity denoted by η_{solar}. Solar power generation also depends on the PV power efficiency denoted by s. It encapsulates both the predicated temperature,the sky conditions, the solar cell efficiency, and the DC to AC inverter efficiency. The solar cell area denoted by $a(m^2)$.

$$P_{output}(s) = \eta_{solar} \cdot s \cdot a \quad (3)$$

Our prototype will assume solar power availability indications as the solar-tuple {*region,power-efficiency*}. Based on the solar generation pattern presented in figure 3, the generation prediction utilizes the the local time in a region and the given power-efficiency

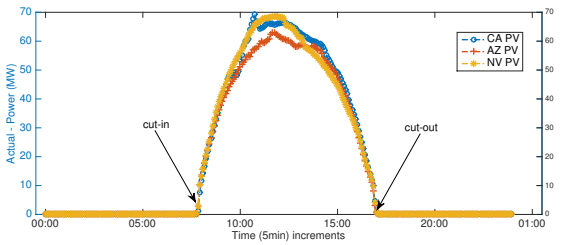

Fig. 3: **Aggregated solar power generation between 2008-2011 taken in Palm Springs CA, Prescott Airport CPV, AZ and Nevada Solar One, NV, indicating on a stable and fixed solar-based power [16]**

C. Optimum Green Energy Utilization is a Resource Management Problem!

This study suggests that efficient; green energy utilization for on-demand video streaming workloads has three main requirements: (1) efficient compute resource discovery, (2) efficient load balancing among the provisioned compute resources and (3) smart failover mechanism that mask failover events from *green* CDN-edge instance to *grid* CDN-edge instance while end-users stream on-demand video [24]. These are discussed below.

Compute Resources Discovery. This assessment comprises both internal and external discovery. Internal discovery refers to CDN-edge instances that run in compute pods that must be able to be easily discovered and connected to control-plane endpoints consistently regardless of which cloud-service-provider is hosting the CDN-edge. External discovery refers to the ability of end-users discovering CDN-edge instances through DNS services for HTTP(S) on-demand video streaming.

Optimal Load Balancing is the seeking of the "best" CDN-edge, based on optimization criteria, for any given the workload processing. After initial discover and connection, clients should be served by the optimal instances based on proximity from the end-users, current load factor, and the availability of green energy resources. e.g., session requests originated from New Jersey should be served by US East as oppose to US West to avoid latency and signal loss.

Efficient Failover is a main component for on-demand video streaming based on green energy. If the endpoint becomes unavailable, in this case due to a sudden lack of green energy, the system must failover the client to another available endpoint that manages the streamed content. Also, failover must be completely automatic i.e. the clients end of the connection remains intact, and the end-user oblivious to the failover event, which means that the end-user's client software requires no support handling failover events. Finally, multiple CDN-edge instances co-located in a region should be accessible by end-users through Domain Naming Service (DNS), as most clients-streaming (browsers) software supports DNS resolutions for finding available CDN-edges.

The green energy utilization model for processing on-

demand video streaming is different than the classic scheduling problem where classical optimal resource allocation techniques are applied [25], [26]. We argue that based on the green energies' volatile nature and the on-demand video streaming workload characteristics, the optimal resource allocation approach should be opportunistic. It requires an effective resource management system for processing on-demand video streaming workloads. Our prototype will employ a Kubernetes flavor "Ubernetes" that implements the three main requirements above [29].

IV. Evaluation

In the following section we evaluate a compute load coordination system component that harmonizes *on-demand streaming* job demands with available compute resources, with prioritiy given to those powered by green energy sources. Such resources will be published to the coordination system through a resource availability tuple {*region,cut-in,rated,cut-off,power-efficiency*}, where *region* indicates the geographic availability region. *power-efficiency* indicates solar or wind based energy power efficiency. *cut-in, rated* and *cut-off* the values appropriate to those energies.

On-demand streaming job demand includes the specific *region, total-job workload, load-factor*, as well as *contract deadline* SLA. The *load-factor* indicates the required number of CPU cores per the *total-job-workload*. The geographic *region* indication will be used to optimize the match between the supply and demand. Also, the *total-job-workload* and the *deadline* will be compared against the *cut-in,rated, cutoff* time for wind or *power-efficiency* for solar, based on the published *load-factor*.

We suggest a hybrid datacenter that does not deviate from the common datacenter architecture. The core difference lies on an automatic transfer switch (ATS) that switches between different available power sources: generator, grid or green energy when available. In both cases the datacenter design does not change and requires incremental changes only by adding green energy power sources to the datacenter's ATS's (Figure 4).

We suggest two types of compute clusters, green-clusters powered by green energy and grid-clusters powered by the electrical grid. Figure 4 shows a simplified datacenter power distribution that supports green energy sources. In such datacenter, both *serving* and *analytical* systems deployed in grid clusters. Further, for incoming *analytical* workloads, few clusters use green resources when there are a viable green energy and mostly standby. As a mitigation strategy, a compute live migration procedure will be available in case of unpredicted lack of renewable resources during a workload processing which presents a risk for SLO violation.

A. Experiment Planning

Below is simulation of a cross-regional platform that is comprised of control-plane, workload-plane and coordinating components. This will be embodied in a resource allocation system (Kubernetes). This system will: (1) provision resources

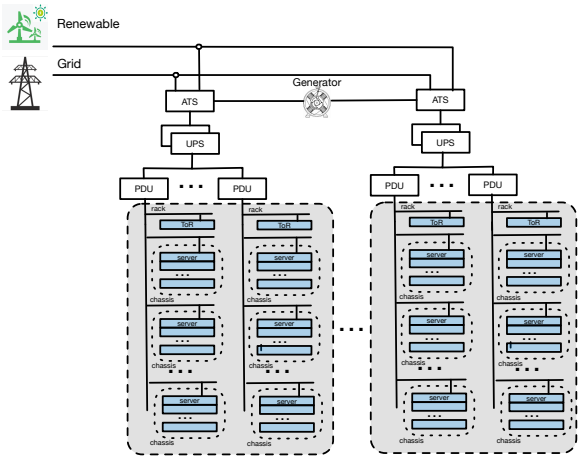

Fig. 4: **Hybrid CDN Edge Data Center Power Supply - The green clusters runs analytical jobs only when green energy is availablele while grid clusters runs on-demand streaming systems. In case of unexpected drop in green energy and SLO requires, a failover will occur so the orchestration system will exclude the green instance from the available compute pool.**

to be neared users; (2) optimize utilization by prioritizing the use of underutilized resources; and (3) seamlessly remove malfunctioning hardware from the system. The control-plane will enable an effective compute resource provisioning system that spans across different public cloud providers and regions. The coordinating components will accept user-workload demands as well as green energy availability from various regions and opportunistically seek to process streaming workloads using compute resources provisioned by green energy resources. The workload-plane will be comprised of edge streaming servers that process the end-user on-demand video streaming. It will built of standard Apache HTTP [28] servers that runs on the edge location.

The control-plane software infrastructure is based on Kubernetes [29], it facilitates internal discovery between CDN instances so instances can connect across different cloud boundaries and regions. Further, end-users can discover the optimal CDN-edges that are (1) nearby, (2) less loaded and (3) healthy. Finally, the Kubernetes automation framework allows the failover mechanism with no dependency upon the end-user client. In particular, we will exploit the *livenessProbe* option that automatically removes green-compute pods, a set of CDN-edge instances, in case of a sudden lack of green energy.

The coordinator component accepts incoming supply and demand traffic, calculates a potential match, within minutes, and notifies back the CSP and the SP for transaction completion. We use Redis [30] as the in-memory data store as the database that stores the system supply and demand calls originated by the end-user workload. The workload is generated by Jmeter instances [31]. The workload generated based on the on-demand video views observed are [10] depicted in Figure 1. Green energy availability simulated based on the known regional patterns depicted by Figure 2 and Figure 3.

We count the number of matches i.e. on-demand video streaming processed by CDN-edge instances operating on green energy. Also, we measure the false-positive cases where a match was suggested but did not met the SLO's deadline due to a violation that caused by a sudden lack of green energy resources. We use the data to extrapolate the possible energy (kWh) that could be generated by the using green CDN-edge instances depicts in Equation 1.

B. Execution

1) The Preparation: The prototype experiment included the setup of three virtual datacenters deployed in different regions: (1) Central US, (2) West US and (3) East US. The clusters were sized based on US population distribution [9] by regions i.e. 20% for West US, 40% for East US and 40% Central US. The cluster sizes for West US, Central US, and East US are 3, 7 and 7 machines respectively. Each machine is standard 2-CPU cores with 7.5GB of memory. Also, the user demand simulation will rely on the US population distribution. Finally, the green energy supply simulation will be based on wind or solar availability observed in the various regions.

The control-plane is comprised of Kubernetes API server and controller-manager. The controller coordinator component will need to allocate resources across several geographic regions different cloud providers. The API server will run a new federation namespace dedicated for the experiment in a manner such that resources are provisioned under a single system. Since the single system may expose external IPs it needs to be protected by an appropriate level of asynchronous encryption [32]. For simplicity, we use a single cloud provider, Google Container Engine, as it provides a multi-zone production-grade compute orchestration system. The compute instances that process the user workloads are deployed as Docker containers that run Ubuntu 15 loaded with Apache HTTP server. For simplicity, we avoid content distribution by embedding the video content to be streamed in the Docker image [32]. We run 52 Docker containers that span across the three regions and act as CDN-edges. Green CDN-edge instances differ from grid CDN-edge instances by Kubernetes labeling. The simulation of the hybrid datacenter is depicted in Figure 4.

A coordination database system that aggregates green energy, solar or wind, availability, was built in software. When energy sources manifest the cut-in patterns depicted by Equation 2 and Equation 3, the coordination system starts green CDN edges in the availability regions. Also, when green energy availability reaches cut-off rates, the coordination system turns off green CDN edge instances.

2) Baseline and Variability of Workloads: The baseline execution included data populations of both green energy availability and user demand for for video streaming. The data population was achieved by the Kubernetes Jmeter batch jobs. The loader jobs goal is to populate the coordinator database with green energy supply based on using a Weibull distribution, which is a generalization of the Rayleigh distribution described above for wind and a normal distribution for solar. Also, the user demand was populated according to the observed empirical patterns depicted by Figure 1.

We simulated the availability and unavailability of green energy using Jmeter-based [31] workload plan against the coordination system [32]. Our implementation starts green CDN-edge instances opportunistically upon green energy availability. Once a CDN-edge instance declares its availability it processes live workloads.

We use the Kubernetes *livenessProbe* for communication between CDN-edge instance pool and its load-balancer that divert traffic to its pool members. Finally, another workload Jmeter-based simulator generates on-demand streaming calls. This workload simulates end-user demand. It includes HTTP progressive download calls to pre-deployed video media in the CDN-edges.

3) Main Execution: In each of the three regional CDN-edge clusters the Kubernetes Jmeter batch jobs that generated green availability traffic to the coordination component were executed. The simulation is comprised of availability indication that are based on Figure 2 and Figure 3. We randomized solar production by using a factor of $\alpha = 0.2$ based on collected data between 2008-2011 in Palm Springs CA, Prescott Airport CPV, AZ and Nevada Solar One, NV [2]. Also, we randomized the wind production by a factor of $\beta = 0.4$ based on collected data between the years 2007-2012 [16]. The demand simulations included a set of calls to the coordinator component spread across 48 hours. The calls originated from three different timezones. The supply simulations consist of wind and solar-based energy time and power windows.

The experiment executions generated two main data traces that we used for the resulting generation computation. The first trace is the simulators logs. The simulator logs includes the demand and supply records. Demand records stored in the Redis key-value store under the key "DemandEvents" followed by timestamp, region and the required compute capacity. The supply calls were stored in the Redis key-value store under the key "SupplyEvents" follows by timestamp, region and supply phase i.e. cut_{in}, cut_{out} or $rated$. For query simplicity the loader ingested three types of records for each supply and demand the by the keys: (1) supply or demand (2) timestamp, and (3) by region. This approach optimized the coordinator queries by timestamp and regions for green CDN-edge instances allocation.

The second trace is the actual allocation logs. It is generated by the coordination system that invokes the Kubernetes command for green CDN-edge instances initialization and disposal. This was used to determine the green energy utility translated into energy (kWh) that did not use grid energy sources.

4) Limitations: Every supply and demand was recorded three times to ease the query process. This approach was used since Redis provided limited query abilities by different keys. This approach might suffer data inconsistency issues where a supply metric was successfully committed to one key recode but missing on other key. Production systems should add extra

safety gates when ingesting data. We used Redis because of its popularity in the Kubentese community. However, our approach is not limited to Redis or other database systems for that matter.

When measuring the green energy overall utility, we used the container initialization and disposal as indication that green energy utility was used. Specifically, we used the *'kubectl logs POD'* command based on the assumption that the coordination system invocation commands are tightly coupled with green energy availability. It is likely that collecting the actual video streaming traces through the various Apache access logs of the CDN-edges will be more accurate.

In the case of a sudden lack of green energy while streaming video a failover occurs. Such failover event relies on domain naming services (DNS), the impact of DNS caching was not included since that might cause streaming delay on the user side. Also, when the coordinator algorithm determines there is enough green energy available it will take grid pods down and activate green pods up in a controlled fashion e.g., one at a time so that no requests are lost during the transition phase. For simplicity, the algorithm avoid that.

V. Analysis

The green energy supply simulation plotted in Figure 5 shows the energy generation in MW for both wind and solar sources. The simulated amounts were adjusted to the amount observed in the traces between 2007-2012 [16].

The user workload simulation plotted in Figure 5 follows the observed user patterns depicts in Figure 1. Also, it shows the aggregated green energy availability for each region. The cloud-coordinator uses these data sets to determine if there is enough green energy available before provisioning green-pods and possibly taking grid-pods down. The utility of the green

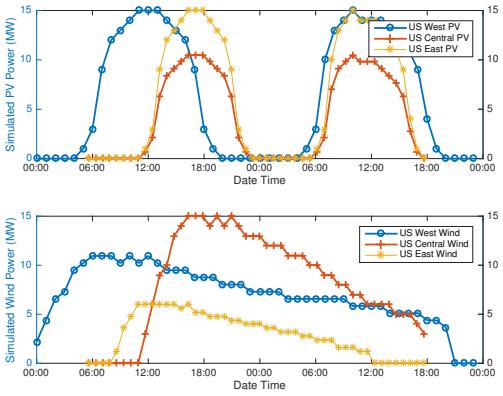

Fig. 5: **Green energy availability simulated in MW across three regions. Amounts are adjusted to NERL measurements i.e., wind generation in West US moderate, Central US outstanding, East US fair. For solar generation in West US strong, Central US moderate-high, and East US low.**

energy was calculated based on the cases where sufficient green energy was available to run the green-CDN pods within the same region. Otherwise cross-regional latencies might degrade the on-demand video experience. The measurements in Figure 6 were adjusted to the estimates of required energy (kWh) for operating the green compute pods. The case where there was negative green energy available it was considered as a miss in the overall utility reckoning.

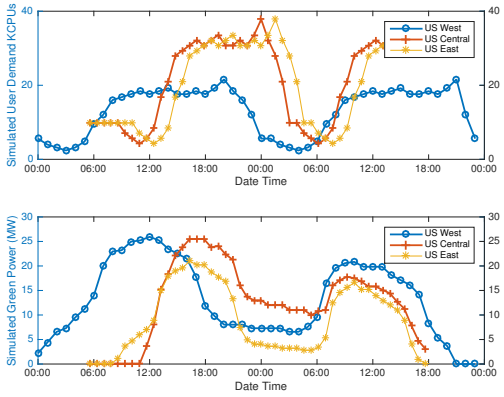

Fig. 6: **Video on-demand user workload per region adjusted with user population opposed to aggregated green energy availability, solar and wind energy.**

VI. Discussion

The scenario described above simulated the usage of green 801.3 kWh out of total 3642 kWh to process the video on-demand streaming workload. i.e. 22% by opportunistic matching. When counting the utility per region West US used 42% of the green energy. Central US used 28% of the simulated green energy. East US utilized only 18% as the initial ratio between user demand and green energy availability was relatively low. Although West US reached 40% utilization it contributed nationally only to the 20% portion it contains from the entire experiment test set. By way of comparison, Jeff Barr of AWS noted that their data centers utilize a 28% cleaner power mix [33]. Extrapolating the simulation results to the initial assessment in Equation 1 yields to a saving of:

$$(18.26(GWh) \cdot 1.89) \cdot 0.1(\$/kWh) \cdot 22\% = \$759,250.8/year$$

VII. Conclusions

The future growth of cloud computing will increase its energy consumption as a fraction of grid power and will cause a significant addition to the ever growing carbon emission since 70% of US power is generated by hydrocarbon fired power plants. Using rapidly emerging green energy for processing cloud computing workloads can limit the anticipated carbon emission growth. However, balancing time varying green energy utilization with time varying energy demands of cloud computing is a complex task that requires sophisticated command and control prediction algorithms beyond the scope of this paper but are emerging in the form of a smart grid

system of systems [34]. Our study shows that green energy utilization for on-demand streaming workload is best described as a resource management problem. The solution presented demonstrates real time balance of green resource supply and cloud computing workload demand and utilizes Ubernetes an open source container cluster manager. The results approximate within 21% those observed in a single cloud instance in the field.

REFERENCES

[1] NIST Cloud Computing Program retrieved from http://www.nist.gov/itl/cloud/

[2] A. Smith, S.J. Horner, N., Azevedo, I., Brown, R., Koomey, J., Masanet, E., Sartor, D., Herrlin, M., Lintner, W. 2016. "United States Data Center Energy Usage Report". Lawrence Berkeley National Laboratory, Berkeley, California. LBNL-1005775

[3] T. Scott Data Center Optimization Initiative , M-16-19, Memorandum for Heads of Executive Departments and Agencies , August 1, 2016

[4] US Renewable Generation in all Sectors by Energy, 2013-2040; http://www.eia.gov/forecasts/aeo/executive_summary.cfm

[5] T. Raftery. 2014. Cloud computing companies ranked by their use of renewable energy. Retrieved from http://greenmonk.net/2014/04/02/cloud-computing-companies-ranked-by-their-use-of-renewable-energy/

[6] Y. Biran, G. Collins and J. Liberatore, "Coordinating Green Clouds as Data-Intensive Computing," 2016 IEEE Green Technologies Conference (GreenTech), Kansas City, MO, 2016, pp. 130-135. doi: 10.1109/GreenTech.2016.31

[7] S. Krishnan, R. K. Sitaraman, "Understanding the Effectiveness of Video Ads: A Measurement Study" ACM Internet Measurement Conference (IMC), 2013, Barcelona, Spain, pp. 23-25.

[8] H. Joshi, Digital Media: "Rise of On-demand Content" retrieved from https://www2.deloitte.com/content/dam/Deloitte/in/Documents/technology-media-telecommunications/in-tmt-rise-of-on-demand-content.pdf

[9] US Population Distribution retrieved from https://www.census.gov/popclock/data_tables.php?component=growth

[10] D. Belson, "State of the Internet Report", Akamai Technologies. 2016, https://www.akamai.com/us/en/our-thinking/state-of-the-internet-report

[11] Internet Statistics retrieved from https://www.statista.com/chart/2647/global-internet-usage-by-the-numbers/

[12] Youtube statistics Retrieved from https://www.youtube.com/yt/press/en-GB/statistics.html

[13] Z. Liu, M. Lin, A. Wierman, S.H. Low, and L.H. Andrew. "Geographical load balancing with renewables". SIGMETRICS Perform. Eval. Rev. 39, 3 (December 2011), 62-66.

[14] N. Lim, S. Majumdar, and P. Ashwood-Smith. "Engineering resource management middleware for optimizing the performance of clouds processing mapreduce jobs with deadlines". 2014. ICPE '14

[15] L. Xi et al. "A case for a coordinated internet video control plane." Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication. ACM, 2012.

[16] NREL, 2015, Western Wind and Solar Integration Research. Retrieved from nrel.gov/electricity/transmission/western-wind-1

[17] M. Patel, Wind and Solar Power Systems, CRC Press, 1999

[18] National Renewable Energy Laboratory. 2015 http://dx.doi.org/10.5439/1052221

[19] T. Arvind, M. S. Sodha. "Performance evaluation of solar PV/T system: an experimental validation." Solar Energy 80.7 (2006): 751-759.

[20] M. Gregor, et al. "On dominant characteristics of residential broadband internet traffic." Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference. ACM, 2009.

[21] Cloud Front reference retrieved from https://aws.amazon.com/cloudfront/

[22] OpenCompute Project, Servers Specification guide retrieved from http://www.opencompute.org/wiki/Server

[23] B. Luiz Andr, J. Clidaras, and U. Hlzle. "The datacenter as a computer: An introduction to the design of warehouse-scale machines." Synthesis lectures on computer architecture 8.3 (2013): 1-154.

[24] M. Abdelbaky, J. Diaz-Montes, M. Parashar, M. Unuvar and M. Steinder, "Docker Containers across Multiple Clouds and Data Centers," 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), Limassol, 2015, pp. 368-371.

[25] B. Anton and J. Abawajy. "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing." Future generation computer systems 28.5 (2012): 755-768.

[26] Z. Qi, Q. Zhu, and R. Boutaba. "Dynamic resource allocation for spot markets in cloud computing environments." Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on. IEEE, 2011.

[27] Matlab Smoothing Splines retrieved from https://www.mathworks.com/help/curvefit/smoothing-splines.html?requestedDomain=www.mathworks.com

[28] Apache Web Server reference retrieved from https://httpd.apache.org

[29] Kubernetes reference retrieved from http://kubernetes.io

[30] Redis refernce retrieved from http://redis.io

[31] Jmeter Loader reference retrieved from http://jmeter.apache.org

[32] Simulation code and data retrieved from https://github.com/yahavb/green-content-delivery-network

[33] Amazon Web Services Sustainability reference retrieved from https://aws.amazon.com/about-aws/sustainability/

[34] Y. Biran, J. Dubow, G. Collins, S. Azam, "Federated Cloud Computing as System of Systems" 2017 Workshop on Computing, Networking and Communications (CNC)