Redesigning Data Centers for Renewable Energy

Anup Agarwal[†*], Jinghan Sun[§*], Shadi Noghabi[‡], Srinivasan Iyengar[‡], Anirudh Badam[‡], Ranveer Chandra[‡], Srinivasan Seshan[†], Shivkumar Kalyanaraman[‡*]

[†]Carnegie Mellon University, [§]University of Illinois at Urbana-Champaign, [‡]Microsoft

ABSTRACT

Renewable energy is becoming an important power source for data centers, especially with the zero-carbon waste pledges made by big cloud providers. However, one of the main challenges of renewable energy sources is the high variability of power produced. Traditional approaches such as batteries or transmitting to the grid fall short on scale, overhead, or "green-ness". We propose Virtual Battery: instead of adapting the availability of power to match the computation demand we shift computational demand to meet the availability of power. Virtual batteries shift demand by requiring applications to either be flexible and delay-tolerant or proactively migrating to where power is (going to be) available. We show that using multiple virtual battery sites in combination can meet the needs of modern applications. Moreover, we show how an intelligent network and power aware co-scheduler can not only provide availability despite variability but also help mitigate migration related network overhead by over 30% in total and 4.2× at peak.

ACM Reference Format:
Anup Agarwal[†*], Jinghan Sun[§*], Shadi Noghabi[‡], Srinivasan Iyengar[‡], Anirudh Badam[‡], Ranveer Chandra[‡], Srinivasan Seshan[†], Shivkumar Kalyanaraman[‡]. 2021. Redesigning Data Centers for Renewable Energy. In *The Twentieth ACM Workshop on Hot Topics in Networks (HotNets '21), November 10–12, 2021, Virtual Event, United Kingdom.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3484266.3487394

1 INTRODUCTION

Cloud computing has become a substantial source of carbon emission, even surpassing the aviation industry [49]. As a result, all major cloud providers have pledged to become "carbon neutral or negative" in the near-term future [30, 45, 60]. Renewable energy sources are an attractive choice for reducing carbon especially since technological advancements have reduced their production costs to values lower than traditional energy sources [36]. However, one challenge in making effective use of renewables is that their energy production varies significantly over time. For example, solar produces no power at night and daytime power output varies with clouds and across seasons. Similarly, wind production changes frequently based on the direction and speed of wind.

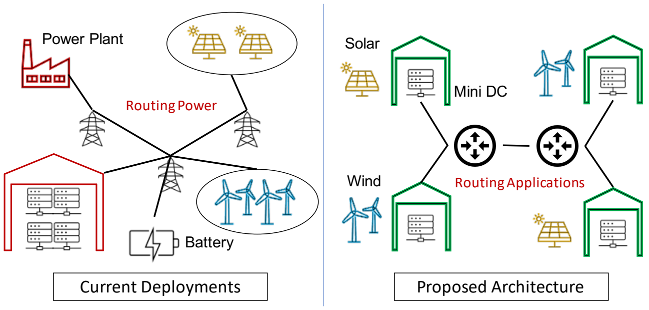


Figure 1: The Virtual Battery (VB) based cloud architecture vs the current deployments.

Existing systems either move energy across space (via transmission lines) or time (using batteries), e.g., using energy from batteries or traditional power generation sources during short-falls, or transmitting to/from the grid during periods of excess/low power generation. However, these solutions have significant shortcomings: 1) traditional non-renewable energy sources have high carbon footprints; 2) transmission lines incur significant overheads (≈ half of the cost is due to transmission and distribution [27]) and takes decades for installation; and 3) penetration of grid-scale Li-ion and other chemical batteries are minuscule in scale, e.g., in the US battery capacity is ≈ 0.4% of the overall solar and wind capacity [12, 26]. We need an alternate, more efficient approach to bridge the gap between the properties of renewable energy sources and requirements of cloud computing.

In this work, we propose a paradigm shift: *Instead of using techniques that adapt the availability of power to match the computation demand, we shift computational demand to meet the availability of power.* We call this *Virtual Battery (VB)* since it relies on shifting computation much like a battery shifts energy across time, and avoids transmission/distribution costs.

[*] Equal contribution.

HotNets '21, November 10–12, 2021, Virtual Event, United Kingdom
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-9087-3/21/11.
https://doi.org/10.1145/3484266.3487394

As shown in Figure 1, VB design enables a different approach to datacenter deployment. Currently, power is sent from various energy sources through distribution lines to large datacenters, as needed to meet current demand. Power is lost in transmission or battery storage, and non-renewable energy sources are used to meet demand. In the proposed architecture, each renewable energy source is coupled with a datacenter, sized to match local energy production. Computation at each site opportunistically scales up and down to make use of locally generated power. This comes with several economical incentives as discussed in §2.1. In practice, we expect a combination of both existing large centralized datacenters complemented by smaller VB edge deployments.

The key challenge associated with the VB approach is that when local power generation dips, servers *must* be shut down. As a result, apps can lose their resources drastically and frequently. Cloud providers can deal with this resource variation in two ways. First, they can require apps to be "degradable", i.e., apps can quickly adjust to fewer or more resources (high elasticity) and hibernate if no resources are available. This is similar to running on preemptable resources such as Spot instances, and Harvest VMs [5–8, 44, 53]. Unfortunately, from the cloud provider's perspective, preemptable apps are not the highest value options, e.g., spot instances are 60-90% cheaper than stable VMs [1]. Also, many apps cannot use degradable resources and require high availability. Thus, we need to maximize the number of resources that assure availability levels similar to the cloud, and only use the spare, variable capacity towards degradable apps.

Second, we observe that *we can reduce resource variability by relying on more than one virtual battery site (multi-VB).* Despite the variability in a single renewable site, across different (nearby) locations, times of the day, and sources (solar, wind, etc), renewable sources often exhibit uncorrelated and complementary patterns of energy production and can reduce overall variability by 3.7× (§2.3).

There are some trade-offs associated with the multi-VB approach. Running applications across multiple VB sites necessitates the use of replication, hot/cold stand-by, and migration, all of which introduce additional network overheads in an often bursty pattern. Based on our results on a 3 month solar and wind traces in Europe, migration spikes can be as high as tens of terabytes, which which can consume 40% of shared WAN capacity (§3). In addition, this can add latency overheads if the multi-VB sites are far apart. This factor limits the number of sites in a multi-VB group as networking overhead increases with the number of sites. We model these trade-offs in a linear optimization formulation and show how a proof-of-concept scheduler can provide significant networking overhead reduction.

Contributions and Roadmap. Our main contributions are:

- Introducing the concept of Virtual Battery (VB) cloud edge data centers that absorb renewable energy variability using a computation shifting paradigm (§1).
- Boosting availability of computation scheduled on VBs by leveraging multiple sites (multi-VB) (§2.3).
- Characterizing the network overhead of multi-VB apps and developing a power & network aware scheduler placing apps across multiple VB sites (§3, §3.1).

2 VIRTUAL BATTERY

In this section, we analyze the feasibility of a Virtual Battery (VB) by answering three main questions: (1) Do VBs make sense from an economic standpoint? (2) How variable are renewable sources? (3) Can we absorb this variability while maintaining availability for VMs?

2.1 Economic Motivation

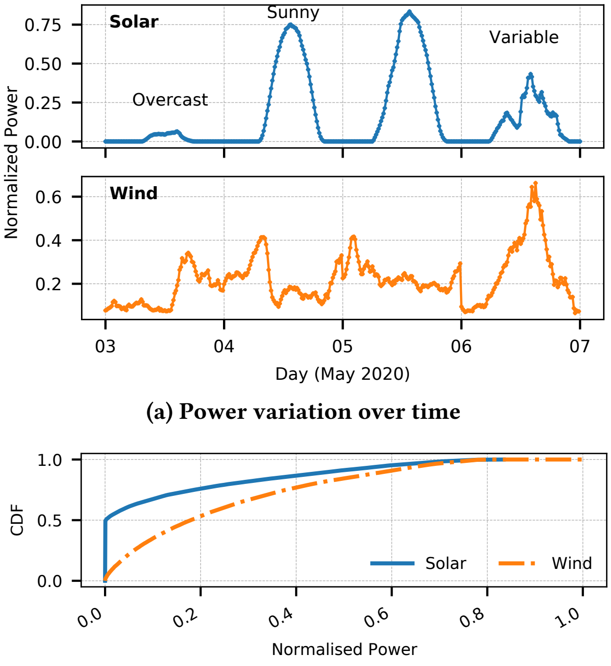
VBs place data-centers *right alongside the* energy farms creating a distributed cluster of edge data-centers (Figure 1). This design brings-up a main question: ***Why does VB make economic sense?***

We see four main reasons. First, VBs will be powered (almost) entirely by renewable energy, rather than an opaque source when using the grid, an important goal for cloud providers [30, 45, 60]. Second, the total datacenter cost can be reduced by ≈10% (= 20% * 50%). 20% of data center operating cost is due to power [31], and 50% of power expense is due to transmission [31]. Co-locating data centers obviates this transmission expense. Third, the growing penetration of renewables has increasingly caused grid operators to *force* renewable farms to *reduce* their output to maintain supply-demand balance [48]. This is done by either i. capping the renewable generation (as high as 6% of the overall renewable generation and increasing each year [17, 23]) or ii. dropping the marginal wholesale price of power to very low levels including negative prices [4], a phenomenon exacerbating with increased renewable penetration [39]. VBs can not only use this energy, but also generate high value from it. Finally, bringing edge data-centers to any remote location has become increasingly simpler and more common, e.g., with modular data centers that are connected anywhere [43].

2.2 Variability of Renewables

Renewables are volatile energy sources–their production depends on several external and uncontrollable factors. We quantify the variability of two main energy sources, solar and wind, by analyzing two datasets: the EMHIRES dataset [28, 35] with traces from >500 sites in Europe, and the ELIA dataset [25], a fine-granularity dataset (15 min) from 25 sites in Belgium which also includes per-site energy forecasts.

We see large amount of variability across space (different locations), time (times of the day, seasons, etc.), and different resources (solar, wind, etc). Figure 2a shows a 4 day sample of solar production (ELIA dataset), normalized to the max production capacity. As shown, solar energy follows a periodic diurnal pattern, but days overcast with heavy clouds can significantly reduce the peak production (3.5% vs 77% the following day), and days with variable cloud patterns



(a) Power variation over time

(b) CDF of power generation over a year

Figure 2: Quantifying Variability for solar and wind.

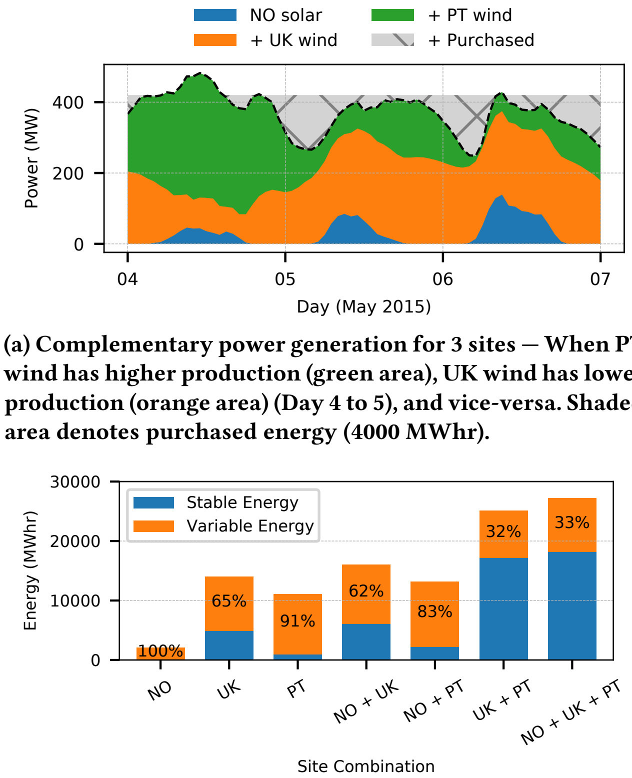
cause spiky energy production. Change of seasons similarly cause variability, e.g., peak production in winter is ≈75% less than summer. Different energy sources exhibit different patterns. As shown in Figure 2a, wind energy production for the same duration exhibits sharp peaks and valleys (depending on weather conditions), but rarely go down to zero.

Figure 2b, shows the CDF of power generation over a year for wind and solar. Both sources exhibit large variability, with median values reaching at most 20% the peak capacity for wind and over 50% zero values for solar energy due to night times. The tail is also high, with 99th divided by 75th percentile ratios of 4× for solar and 2× for wind.

2.3 Availability Despite Variability

We target two class of applications. First, degradable applications that can adaptively be down-graded (similar to when using Harvest VM or Spot instances). Many scenarios can potentially use such degradable VMs such as batch or ML training jobs. The other category are stable VMs that require high levels of availability (similar to on-demand VMs in the cloud). This category provides higher value for the cloud providers and is the commonly used approach for most applications. Our goal is to maximize the number of stable VMs that this infrastructure can support. In this section, we explore a promising approach to reach this goal.

Despite the large variability in a single renewable energy production site, we observed that these variability patterns are often independent and/or complimentary when looking at different sites or times of the day. This complimentary trend can be because of: a) using different energy sources (e.g., wind vs. solar at night time); b) effect of micro-climates or weather (e.g., same solar source but in two different locations, one of which is covered by clouds); or c) time of the day (e.g., day in one location and dusk in another).



(a) Complementary power generation for 3 sites — When PT wind has higher production (green area), UK wind has lower production (orange area) (Day 4 to 5), and vice-versa. Shaded area denotes purchased energy (4000 MWhr).

(b) Break-down of stable & variable energy.

Figure 3: Reducing variability in energy production by aggregating multiple VB sites.

We leverage this insight and question, *can we mask the variability in renewable energy sources by leveraging variability patterns across more than one site?*

We searched for complimentary groups of sites, all in close proximity of each other (<50ms ping latency), over 3 day intervals (in EMHIRES). Since our dataset only includes normalized power, we assume all sites have similar peak capacity of 400 MW–the median peak-capacity across large farms in the world [18][1]. We use the coefficient of variation, *cov* (standard deviation divided by mean) as our comparison metric. Based on our result, even when combining just two sites, > 52% of possible 2-site combinations improved *cov* by > 50%. Figure 3a demonstrates such an example as a stacked area graph, i.e., upper areas are added to all areas below them. As shown, the solar pattern in Norway (NO) when complemented with just one additional wind site (UK wind) reduces *cov* by 3.7×. Adding power from another wind

[1] We could not find any public datasets with large number of sites spread geographically that have raw power values. We also note that the max power capacity ranges for both solar and wind are very similar [61, 65]

source in Portugal (PT wind), further improves variability and reduces *cov* by an additional 2.3×.

Does aggregation increase the stable capacity? We quantify the amount of *stable* energy generated over a time window as: the minimum power level in the window multiplied by the size of a window. Since this energy is guaranteed to be available in that time window, it can reliably be used for stable VMs, and all remaining energy (called as variable energy) for degradable VMs (e.g., spot, harvest VMs). As shown in Figure 3b, when we combine VB sites, larger portion of energy is stable, e.g., in "NO+UK+PT" 67% of energy is stable while for "NO+UK" this value is 38% stable.

Would using a small reliable energy source alongside help? Traditional ways, back-up generators, purchasing from the grid, and using batteries, all come with their individual trade-offs. However, using these techniques in small scales, just enough to cope with minor variability, can be a beneficial option economically. For example, we can try to fill in some of the worst gaps in the "NO+UK+PT" combination, shown as gray shaded areas in Figure 3a. In this case, by purchasing an additional 4,000 MWhr energy from the grid, we can stabilize 8,000 MWhr of variable energy and achieve a total additional 12,000 MWhr of stable energy.

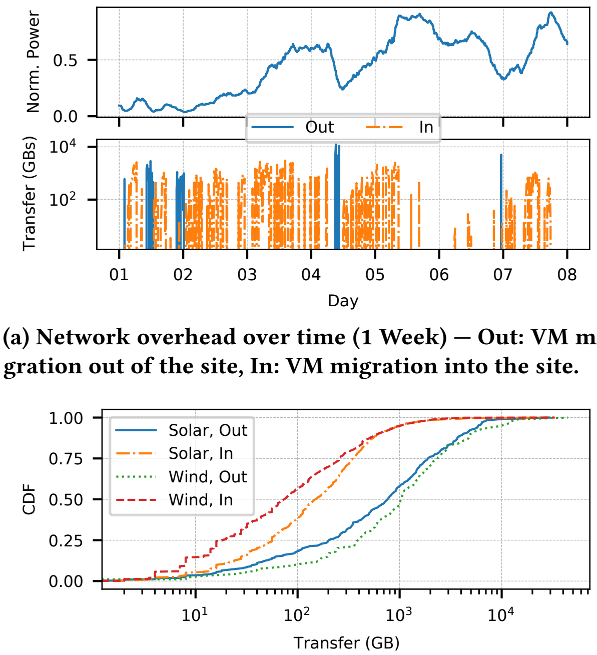
This multi-VB design introduces new networking challenges (§3) and power/economical challenges (§5).

3 A NEW NETWORKING CHALLENGE

The multi-VB design flattens the variability in energy, however, it introduces a new challenge–*Applications that could regularly run in a single data-center with high levels of availability now* ***must*** *run across multiple sites to achieve similar availability.* Such applications must rely on either hot/cold standbys using continuous replication or migration. This introduces continuous or bursty network overheads on the wide area links connecting sites. Such links may be more constrained given the remote locations of typical renewable energy farms. While replication and migration is a well-known overhead in traditional cloud settings, *the scale is vastly different.* In the cloud, many applications are not geo-replicated. For the ones that are, the migration events are due to failures or load balancing, which are rare and oftenly handled within the same data-center (e.g., zone replication).

We performed trace-driven simulations to quantify the networking overhead of multi-VBs as the volume of incoming and outgoing traffic per site due to VM migrations [2].

Setup and Workloads. We instantiate a site with ≈ 700 servers each with 40 cores and 512 GB memory. We use an Azure production VM arrival trace and Azure's VM allocation policy to assign VMs to servers. We scale the ELIA [25] dataset such that the cluster is fully powered at the max power capacity of the farm. When power decreases, we first power down unallocated cores, then if needed, we migrate out VMs from servers (in a round-robin order). We use an admission control policy that rejects VMs to maintain 70% utilization (similar to the VM trace). When power increases, we launch previously rejected VMs and consider these as VMs migrated into the site. We use the memory allocated to a VM for estimating migration traffic due to the absence of disk or memory utilization data.



(a) Network overhead over time (1 Week) — Out: VM migration out of the site, In: VM migration into the site.

(b) CDF of network overhead over 3 months of simulation. CDF only includes non-zero overhead values.

Figure 4: Network overhead of migration in a multi-VB setting with wind power.

Observations. As shown in Figure 4a (top graph), > 80% of the power changes don't incur migrations. Since the cluster is running at 70% utilization, minor variations in power are absorbed by simply powering down un-allocated cores. In the remainder 20% of the cases, migrations can cause movement of multiple TBs of data which is significant. For instance, if the migration is to complete within 5 minutes, then a 10 terabyte spike requires ≈ 200 Gbps network capacity for a single site. This is roughly 40% of the share of WAN capacity per site, assuming ≈ 100 sites (each with 1000 servers) share an aggregate WAN link with 50 terabits/sec capacity [67].

Figure 4b shows the CDF of migration overhead omitting the zero values. Migration overhead varies significantly with 99th divided by 50th percentile values as high as 18-30× for in migration and 12.5-16× for out migration. In migration, compared to out migration, has smaller spikes but spread further over time, e.g., 7× smaller 99th percentile for wind. This is because, minor power reductions are absorbed by under-utilized machines, while minor power gains cause migrations into the site.

[2] As future work, we plan to incorporate migration latency and impact to application's execution time similar to [2].

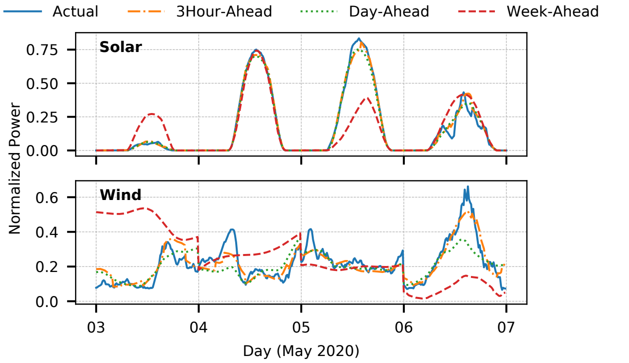


Figure 5: Energy prediction of solar and wind in near (3 hour, day ahead) and far-away future (week ahead).

3.1 Solving the network bottleneck

Despite networking challenges, VBs have a unique characteristic: ***migrations are spiky, but also predictable.*** The main source of variability is weather conditions, which can be predicted quite accurately. Figure 5, shows the power forecasts provided in the ELIA dataset [25] (based on weather forecasts). As shown, the prediction model is able to predict the near future very accurately, with a mean absolute percentage error (MAPE) for next 3 hour predictions of 8.5-9% and for day ahead predictions of 18-25%. Also, for far-away future (week-ahead) the predictions capture the general trend (MAPE of 44% and 75% for solar and wind). Bulk of migrations occur when there are sharp changes in power (§3), which can be predicted with at least a day of notice. Since these are points with sharp drops/increases, they are resilient to small errors in predictions. This predictable characteristic introduces a complex and yet unique scheduling problem:

How to schedule applications across highly-variable but predictable capacity locations in a way that i) ensures high level of availability, ii) introduces low & non-bursty network overheads, and iii) minimizes energy usage?

There is an inherent trade-off between these goals. To reach higher availability, applications would want to spread their compute across as many uncorrelated and complimentary locations as possible, e.g., different VB sites, racks, and servers. But, this would mean higher network overhead by introducing more migrations and replications and less energy efficiency by powering up more racks, top of rack switches, and servers even though they might be underutilized.

For each application, with a number of requested VMs, the scheduler needs to find a group of VB sites that a) are in close proximity to reduce latency overheads of splitting applications, b) have complimentary energy patterns to guarantee availability of resources & reduce migrations, c) have a good mix of stable and degradable VMs (based on their individual energy patterns) to cope with local power variations.

To solve this scheduling, we model the cluster of VB sites as a graph (Figure 6). Each node represents a VB site and has the energy prediction alongside total and allocated computation capacity. Two nodes are connected via an edge if their latency is below a fixed threshold, 50*ms* in our case. We use this model and break the scheduling problem into 4 steps: 1) subgraph identification, 2) subgraph selection, 3) site selection, 4) VM placement. We optimize the placement of incoming applications. As the environment changes, e.g., weather predictions update or applications complete and resources free up, we need to rerun the optimization.

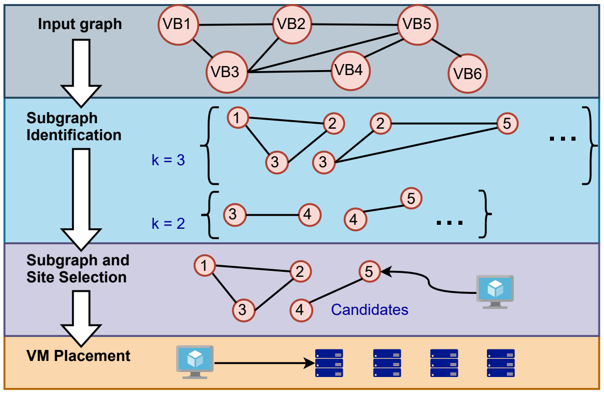


Figure 6: Overview of our scheduling approach

1) Subgraph identification. we identify subgraphs of the VB site graph which have *low latency and complimentary variability*. We formulate this as a k-clique identification and prioritization problem. First, we find all cliques (fully connected sub-graphs) of a given size k (k=2 to 5). These are subgraphs with a small latency between any two VB pairs, thus, an application split within such a subgraph would not experience high latency overheads. Then, for each k, we sort all subgraphs based on the total coefficient of variability. Identifying dense subgraphs has been a well-studied problem in literature with tractable approximate solutions [11].

(2, 3) Subgraph and Site selection. In these two steps we identify the exact site(s) the application should be placed on. Subgraph identification can quickly generate large number of subgraphs. Thus, first we select a smaller set of subgraphs candidates and for each candidate subgraph find the optimal site placement schedule. For both the steps, candidates/sites are selected to optimize for two things: (1) choosing options that are predicted to maintain good power levels to prevent migrations, (2) balancing load between subgraphs/sites. We formulate subgraph and site selection as a Mixed-Integer Programs (MIP) with the following two objectives:

$$\text{O1: Total} \quad \text{Min} \sum_{i \in apps} migration\_overhead(i),$$

$$\text{O2: Peak} \quad \text{Min} \max_{i \in apps} migration\_overhead(i),$$

The first order goal is to minimize the total migration overhead (in bytes) while looking ahead in the future. The second order goal reduces the peak migration overhead (in bytes). Optimizing for O1 and O2 implicitly maintains a balance of degradable applications in each locations allowing handling of abrupt power variations.

Policy	Total	99%ile	Peak	Std
Greedy	306,966	7,093	16,022	1,507
MIP-24h	236,217	3,711	80,942	4,081
MIP	**209,961**	9,379	62,753	2,697
MIP-peak	212,247	**1,684**	**1,941**	**562**

Table 1: Comparison of migration overhead (in GBs) between different scheduling policies.

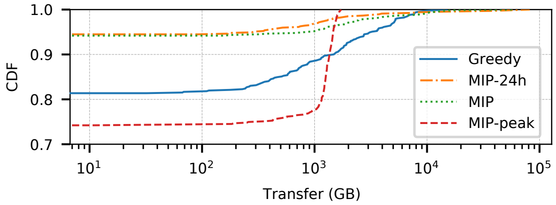


Figure 7: CDF of migration overhead across four different scheduling policies.

We evaluate our scheduler over a 7 day period with four different policies: (i) *MIP*: that optimizes only the total migration overhead (only O1) but for the entire period, (ii) *MIP-peak*: variant of MIP that optimizes both total overhead (O1) and peak overhead (O2), (iii) *MIP-24h*: a variant of MIP (only O1) that iteratively optimizes total migration for the next day, and (iv) *Greedy*: a baseline greedy policy that always assigns VMs to the site with the most available power;

As shown in Table 1, MIP improves the total overhead by > 30% compared to greedy. Other MIP variations have similar migration overhead (1-12.5% worse), but significantly better peak overhead. Compared to greedy, MIP-peak shows > 4.2× improvement at 99th percentile while also being less bursty with a 2.7× lower standard deviation. As shown in Figure 7, MIP-peak achieves this by performing more number of migrations (74% zero values vs. 81% in greedy and 94% in MIP), but each at a lower volume.

Sources of benefits. We observe three key insights that allow MIP and its variants to reduce total and peak migration overheads. First, all MIP variants place VMs on sites which are predicted to have stable power in the future. This reduces the need for migrations. Second, MIP variants also maintain a good mix of degradable VMs in line with predicted power availability. Thus, when power reduces, degradable VMs take most of the hit without needing to migrate stable VMs. Lastly, MIP-peak migrates VMs preemptively, spreading out migrations over time and reducing burstiness.

4) VM placement. We place VMs onto servers in the assigned VB site to minimize total power usage by consolidating as much as possible. Any state-of-the-art approaches as described in related work can be used for this step.

4 RELATED WORK

Migration, replication and check-pointing. A number of prior works have investigated use of migration [51, 54, 56], replication [19, 40] and check-pointing [46, 47] to manage faults and outages. A few other prior works [16, 22, 33, 50, 54, 55, 58], have looked at using migration/replication to ensure availability using spot markets. While several of them can be used by our work, the scale of variations in VBs are much higher (e.g., they can go all the way down to zero) such that these techniques don't directly translate to our context.

Power aware scheduling. We employ features like frequency scaling, powering down cores/caches/memory units to control power distributed to servers [21, 37]. For the VM placement, we rely on a number of prior works [10, 13, 20, 32, 38, 57, 62–64, 66, 68]. These minimize energy consumption by considering resource utilization, power consumption, and dynamic voltage & frequency scaling to perform packing, scheduling, and over-subscription of VMs and workloads within a data center. Energy efficiency can be further improved by tuning workloads [52] and opportunistically turning off unused servers/networking infrastructure [24, 34].

Energy based elasticity/migration. Some works [9, 41] have looked at adaptively adjusting computation based on the cost of energy (from the grid). Others [14, 42] have looked at reducing their peak demand to help stabilize the energy grid. Some approaches [29] migrate workloads based on energy availability. While these have inspired of our work, they do not address the many challenges of variability for compute or they do not target stable VMs with high availability.

5 DISCUSSION AND CONCLUSION

Other challenges of multi-VB. Realizing the multi-VB design comes with a few additional challenges. First, the additional network overhead could require powering on network infrastructure that would otherwise be idle/off. However, we observe that migration occurs only 2-4% of the time assuming 200Gbps WAN link per VB site (§3). Hence, the migration energy added by VB is negligible compared to upto 50% energy loss in power transmission [59]. Second, shutting down compute is the lost economic opportunity, where fully set-up hardware is not being utilized. This can be circumvented by using deprecated and refurbished hardware or purchasing power from the grid. Finally, VB requires managing more number of (edge) data centers compared to a centralized design, thus, requiring additional one-time infrastructure (e.g., WAN connections, buildings, etc.) and maintenance. The community is already building edge data centers [3, 15]. VB adds to these efforts by providing high availability on-top.

Conclusion. In this work, we investigate methods to build data centers with renewable energy. We propose Virtual Battery as an approach that adapts computation to accommodate power variations. In our approach, we envision data centers co-located with power generation to curb transmission costs. We observe that leveraging an ensemble of multiple sites significantly reduces variability at the cost of considerable network overhead. We finally propose a network and power-aware scheduler that reduces network overheads.

REFERENCES

[1] 2021. Pricing - Windows Virtual Machines | Microsoft Azure. https://azure.microsoft.com/en-us/pricing/details/virtual-machines/windows. (2021). [Online; accessed 23. Jun. 2021].

[2] Sherif Akoush, Ripduman Sohan, Andrew Rice, Andrew W Moore, and Andy Hopper. 2010. Predicting the performance of virtual machine migration. In *2010 IEEE international symposium on modeling, analysis and simulation of computer and telecommunication systems*. IEEE, 37–46.

[3] Sherif Akoush, Ripduman Sohan, Andrew Rice, Andrew W. Moore, and Andy Hopper. 2011. Free Lunch: Exploiting Renewable Energy for Computing. In *13th Workshop on Hot Topics in Operating Systems (HotOS XIII)*. USENIX Association, Napa, CA. https://www.usenix.org/conference/hotosxiii/free-lunch-exploiting-renewable-energy-computing

[4] Aleasoft. 2020. Negative prices in some European markets during the first weekend of July due to the wind energy. https://rb.gy/krt5jr. (2020).

[5] Amazon. 2020. Spot Instances. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-spot-instances.html. (2020).

[6] Pradeep Ambati, Íñigo Goiri, Felipe Frujeri, Alper Gun, Ke Wang, Brian Dolan, Brian Corell, Sekhar Pasupuleti, Thomas Moscibroda, Sameh Elnikety, et al. 2020. Providing SLOs for Resource-Harvesting VMs in Cloud Platforms. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI '20)*. 735–751.

[7] Amazon AWS. 2020. Burstable performance instances. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/burstable-performance-instances.html. (2020).

[8] Microsoft Azure. 2017. Introducing B-Series, our new burstable VM size. https://azure.microsoft.com/en-us/blog/introducing-b-series-our-new-burstable-vm-size/. (2017).

[9] Shahab Bahrami, Vincent WS Wong, and Jianwei Huang. 2018. Data center demand response in deregulated electricity markets. *IEEE Transactions on Smart Grid* 10, 3 (2018), 2820–2832.

[10] Josep Ll Berral, Íñigo Goiri, Ramón Nou, Ferran Julià, Jordi Guitart, Ricard Gavaldà, and Jordi Torres. 2010. Towards energy-aware scheduling in data centers using machine learning. In *Proceedings of the 1st International Conference on energy-Efficient Computing and Networking*. 215–224.

[11] Aditya Bhaskara, Moses Charikar, Eden Chlamtac, Uriel Feige, and Aravindan Vijayaraghavan. 2010. Detecting high log-densities: an O (n 1/4) approximation for densest k-subgraph. In *Proceedings of the forty-second ACM symposium on Theory of computing*. 201–210.

[12] University of Michigan Center for Sustainable Systems. 2020. US Grid Energy Storage Factsheet. http://css.umich.edu/factsheets/us-grid-energy-storage-factsheet. (2020).

[13] Angelos Chatzipapas, Dimosthenis Pediaditakis, Charalampos Rotsos, Vincenzo Mancuso, Jon Crowcroft, and Andrew Moore. 2015. Challenge: Resolving data center power bill disputes: The energy-performance trade-offs of consolidation. In *Proceedings of the 2015 ACM Sixth International Conference on Future Energy Systems*. 89–94.

[14] Hao Chen, Michael C Caramanis, and Ayse K Coskun. 2014. The data center as a grid load stabilizer. In *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 105–112.

[15] Andrew A. Chien, Richard Wolski, and Fan Yang. 2015. The Zero-Carbon Cloud: High-Value, Dispatchable Demand for Renewable Power Generators. *The Electricity Journal* 28, 8 (2015), 110–118. https://doi.org/10.1016/j.tej.2015.09.010

[16] Asaf Cidon, Robert Escriva, Sachin Katti, Mendel Rosenblum, and Emin Gun Sirer. 2015. Tiered replication: A cost-effective alternative to full cluster geo-replication. In *2015 USENIX Annual Technical Conference (ATC'15)*. 31–43.

[17] CleanEnergyWire. 2021. More renewables curbed to stabilise German power grid – report. https://www.cleanenergywire.org/news/more-renewables-curbed-stabilise-german-power-grid-report. (2021).

[18] Contributors to Wikimedia projects. 2021. List of onshore wind farms - Wikipedia. https://en.wikipedia.org/wiki/List_of_onshore_wind_farms. (2021). [Online; accessed 23. Jun. 2021].

[19] Brendan Cully, Geoffrey Lefebvre, Dutch Meyer, Mike Feeley, Norm Hutchinson, and Andrew Warfield. 2008. Remus: High availability via asynchronous virtual machine replication. In *Proceedings of the 5th USENIX symposium on networked systems design and implementation*. San Francisco, 161–174.

[20] Howard David, Chris Fallin, Eugene Gorbatov, Ulf R Hanebutte, and Onur Mutlu. 2011. Memory power management via dynamic voltage/frequency scaling. In *Proceedings of the 8th ACM international conference on Autonomic computing*. 31–40.

[21] Howard David, Eugene Gorbatov, Ulf R Hanebutte, Rahul Khanna, and Christian Le. 2010. RAPL: Memory power estimation and capping. In *2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*. IEEE, 189–194.

[22] Diego Didona, Kristina Spirovska, and Willy Zwaenepoel. 2017. Okapi: Causally consistent geo-replication made faster, cheaper and more available. *arXiv preprint arXiv:1702.04263* (2017).

[23] DownToEarth. 2020. Renewable energy: Curtailment is a bane. https://www.downtoearth.org.in/blog/energy/renewable-energy-curtailment-is-a-bane-68857. (2020).

[24] Truong Vinh Truong Duy, Yukinori Sato, and Yasushi Inoguchi. 2010. Performance evaluation of a Green Scheduling Algorithm for energy savings in Cloud computing. In *2010 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW)*. 1–8. https://doi.org/10.1109/IPDPSW.2010.5470908

[25] ELIA. 2021. Solar and Wind Generation. https://www.elia.be/en/grid-data/power-generation. (2021).

[26] US Energy Information Administration. 2020. Electricity explained: Electricity generation, capacity, and sales in the United States. https://www.eia.gov/energyexplained/electricity/electricity-in-the-us-generation-capacity-and-sales.php. (2020).

[27] Robert L Fares and Carey W King. 2017. Trends in transmission, distribution, and administration costs for US investor-owned electric utilities. *Energy Policy* 105 (2017), 354–362.

[28] Iratxe Gonzalez Aparicio, Thomas Huld, Francesco Careri, Fabio Monforti, and Andreas Zucker. 2017. EMHIRES dataset Part II: Solar power generation. *European Meteorological derived High resolution RES generation time series for present and future scenarios. Part II: PV generation using the PVGIS model* (2017).

[29] Google. 2020. Our data centers now work harder when the sun shines and wind blows. https://blog.google/inside-google/infrastructure/data-centers-work-harder-sun-shines-wind-blows. (2020).

[30] Google. 2021. Cloud sustainability. https://cloud.google.com/sustainability. (2021).

[31] Green Mountain Data Center. 2020. Data Center Power Costs. https://greenmountain.no/2020/07/01/data-center-power-costs/. (2020).

[32] Ori Hadary, Luke Marshall, Ishai Menache, Abhisek Pan, Esaias E Greeff, David Dion, Star Dorminey, Shailesh Joshi, Yang Chen, Mark Russinovich, et al. 2020. Protean:{VM} Allocation Service at Scale. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI '20)*. 845–861.

[33] Aaron Harlap, Andrew Chung, Alexey Tumanov, Gregory R. Ganger, and Phillip B. Gibbons. 2018. Tributary: spot-dancing for elastic services with latency SLOs. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*. USENIX Association, Boston, MA, 1–14. https://www.usenix.org/conference/atc18/presentation/harlap

[34] Brandon Heller, Srini Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Sujata Banerjee, and Nick McKeown. 2010. ElasticTree: Saving Energy in Data Center Networks. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI'10)*. USENIX Association, USA, 17.

[35] Gonzalez Aparicio Iratxe, Andreas Zucker, Francesco Careri, Fabio Monforti, Thomas Huld, and Jake Badger. 2016. EMHIRES dataset Part I: Wind power generation. (2016).

[36] JustEnergy. 2021. Renewable Energy vs. Traditional Energy: The Real Cost. https://justenergy.com/blog/renewable-energy-vs-traditional-energy-the-real-cost/. (2021).

[37] Vasileios Kontorinis, Liuyi Eric Zhang, Baris Aksanli, Jack Sampson, Houman Homayoun, Eddie Pettis, Dean M Tullsen, and Tajana Simunic Rosing. 2012. Managing distributed ups energy for effective power capping in data centers. In *2012 39th Annual International Symposium on Computer Architecture (ISCA)*. IEEE, 488–499.

[38] Alok Kumbhare, Reza Azimi, Ioannis Manousakis, Anand Bonde, Felipe Frujeri, Nithish Mahalingam, Pulkit Misra, Seyyed Ahmad Javadi, Bianca Schroeder, Marcus Fontoura, et al. 2020. Prediction-Based Power Oversubscription in Cloud Platforms. *arXiv preprint arXiv:2010.15388* (2020).

[39] Evangelos Kyritsis, Jonas Andersson, and Apostolos Serletis. 2017. Electricity prices, large-scale renewable integration, and policy implications. *Energy Policy* 101 (2017), 550–560.

[40] Horacio Andrés Lagar-Cavilla, Joseph Andrew Whitney, Adin Matthew Scannell, Philip Patchin, Stephen M Rumble, Eyal De Lara, Michael Brudno, and Mahadev Satyanarayanan. 2009. Snowflock: rapid virtual machine cloning for cloud computing. In *Proceedings of the 4th ACM European conference on Computer systems*. 1–12.

[41] Zhenhua Liu, Iris Liu, Steven Low, and Adam Wierman. 2014. Pricing data center demand response. *ACM SIGMETRICS Performance Evaluation Review* 42, 1 (2014), 111–123.

[42] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, and Niangjun Chen. 2013. Data center demand response: Avoiding the coincident peak via workload shifting and local generation. *Performance Evaluation* 70, 10 (2013), 770–791.

[43] Microsoft. 2020. More renewables curbed to stabilise German power grid – report. https://azure.microsoft.com/en-us/blog/introducing-the-microsoft-azure-modular-datacenter/. (2020).

[44] Microsoft. 2020. Use Azure Spot Virtual Machines. https://docs.microsoft.com/en-us/azure/virtual-machines/spot-vms. (2020).

[45] Microsoft. 2021. Azure sustainability. https://azure.microsoft.com/en-in/global-infrastructure/sustainability/. (2021).

[46] Jayashree Mohan, Amar Phanishayee, and Vijay Chidambaram. 2021. CheckFreq: Frequent, Fine-Grained DNN Checkpointing. In *19th USENIX Conference on File and Storage Technologies (FAST '21)*. 203–216.

[47] Adam Moody, Greg Bronevetsky, Kathryn Mohror, and Bronis R De Supinski. 2010. Design, modeling, and evaluation of a scalable multi-level checkpointing system. In *SC'10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 1–11.

[48] Eric O'Shaughnessy, Jesse R Cruce, and Kaifeng Xu. 2020. Too much of a good thing? Global trends in the curtailment of solar PV. *Solar Energy* 208 (2020), 1068–1077.

[49] The Shift project. 2019. *Lean ICT – TOWARDS DIGITAL SOBRIETY*. Technical Report.

[50] Asfandyar Qureshi, Rick Weber, Hari Balakrishnan, John Guttag, and Bruce Maggs. 2009. Cutting the electric bill for internet-scale systems. In *Proceedings of the ACM SIGCOMM 2009 conference on Data communication*. 123–134.

[51] Adam Ruprecht, Danny Jones, Dmitry Shiraev, Greg Harmon, Maya Spivak, Michael Krebs, Miche Baker-Harvey, and Tyler Sanderson. 2018. Vm live migration at scale. *ACM SIGPLAN Notices* 53, 3 (2018), 45–56.

[52] Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020. Green AI. *Commun. ACM* 63, 12 (Nov. 2020), 54–63. https://doi.org/10.1145/3381831

[53] Prateek Sharma, Ahmed Ali-Eldin, and Prashant Shenoy. 2019. Resource deflation: A new approach for transient resource reclamation. In *Proceedings of the Fourteenth EuroSys Conference 2019*. 1–17.

[54] Prateek Sharma, Stephen Lee, Tian Guo, David Irwin, and Prashant Shenoy. 2015. Spotcheck: Designing a derivative iaas cloud on the spot market. In *Proceedings of the Tenth European Conference on Computer Systems*. 1–15.

[55] Supreeth Shastri and David Irwin. 2017. HotSpot: automated server hopping in cloud spot markets. In *Proceedings of the 2017 Symposium on Cloud Computing*. 493–505.

[56] Rahul Singh, David Irwin, Prashant Shenoy, and Kadangode K Ramakrishnan. 2013. Yank: Enabling green data centers to pull the plug. In *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*. 143–155.

[57] Chandrasekar Subramanian, Arunchandar Vasan, and Anand Sivasubramaniam. 2010. Reducing data center power with server consolidation: Approximation and evaluation. In *2010 International Conference on High Performance Computing*. IEEE, 1–10.

[58] Supreeth Subramanya, Tian Guo, Prateek Sharma, David Irwin, and Prashant Shenoy. 2015. Spoton: a batch computing service for the spot market. In *Proceedings of the sixth ACM symposium on cloud computing*. 329–341.

[59] Kavita Surana and Sarah Marie Jordaan. 2021. We calculated emissions due to electricity loss on the power grid – globally, it's a lot. (Oct 2021). https://rb.gy/fhqhma [Online; accessed 11. Oct. 2021].

[60] The Climate Pledge. 2021. See which companies and organizations have committed to net zero carbon by 2040. https://www.theclimatepledge.com/us/en/Signatories. (2021).

[61] The Wind Power. 2021. Wind farms databases. https://www.thewindpower.net/store_windfarms_view_all_en.php. (2021).

[62] Abhishek Verma, Luis Pedrosa, Madhukar R. Korupolu, David Oppenheimer, Eric Tune, and John Wilkes. 2015. Large-scale cluster management at Google with Borg. In *Proceedings of the European Conference on Computer Systems (EuroSys)*. Bordeaux, France.

[63] Gregor Von Laszewski, Lizhe Wang, Andrew J Younge, and Xi He. 2009. Power-aware scheduling of virtual machines in dvfs-enabled clusters. In *2009 IEEE International Conference on Cluster Computing and Workshops*. IEEE, 1–10.

[64] Cheng Wang, Bhuvan Urgaonkar, Qian Wang, George Kesidis, and Anand Sivasubramaniam. 2013. Data center power cost optimization via workload modulation. In *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. IEEE, 260–263.

[65] Wiki Solar. 2021. Deployment of utility-scale solar power by continent. https://wiki-solar.org/region/continents/. (2021).

[66] Chia-Ming Wu, Ruay-Shiung Chang, and Hsin-Yu Chan. 2014. A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters. *Future Generation Computer Systems* 37 (2014), 141–147.

[67] Chi yao Hong, Subhasree Mandal, Mohammad A. Alfares, Min Zhu, Rich Alimi, Kondapa Naidu Bollineni, Chandan Bhagat, Sourabh Jain, Jay Kaimal, Jeffrey Liang, Kirill Mendelev, Steve Padgett, Faro Thomas Rabe, Saikat Ray, Malveeka Tewari, Matt Tierney, Monika Zahn, Jon Zolla, Joon Ong, and Amin Vahdat. 2018. B4 and After: Managing Hierarchy, Partitioning, and Asymmetry for Availability and Scale in Google's Software-Defined WAN. In *SIGCOMM'18*. https://conferences.sigcomm.org/sigcomm/2018/program_tuesday.html

[68] Chaojie Zhang, Alok Kumbhare, Ioannis Manousakis, Deli Zhang, Pulkit Misra, Rod Assis, Kyle Woolcock, Nithish Mahalingam, Brijesh Warrier, David Gauthier, Lalu Kunnath, Steve Solomon, Osvaldo Morales, Marcus Fontoura, and Ricardo Bianchini. 2021. Flex: High-Availability Datacenters With Zero Reserved Power. In *Proceedings of the International Symposium on Computer Architecture (ISCA'21)*.