An energy aware cost recovery approach for virtual machine migration

Muhammad Zakarya and Lee Gillam

Department of Computer Science, University of Surrey, UK
{mohd.zakarya,l.gillam}@surrey.ac.uk

Abstract. Datacenters provide an IT backbone for today's business and economy, and are the principal electricity consumers for Cloud computing. Various studies suggest that approximately 30% of the running servers in US datacenters are idle and the others are under-utilized, making it possible to save energy and money by using Virtual Machine (VM) consolidation to reduce the number of hosts in use. However, consolidation involves migrations that can be expensive in terms of energy consumption, and sometimes it will be more energy efficient not to consolidate. This paper investigates how migration decisions can be made such that the energy costs involved with the migration are recovered, as only when costs of migration have been recovered will energy start to be saved. We demonstrate through a number of experiments, using the Google workload traces for 12,583 hosts and 1,083,309 tasks, how different VM allocation heuristics, combined with different approaches to migration, will impact on energy efficiency. We suggest, using reasonable assumptions for datacenter setup, that a combination of energy-aware fill-up VM allocation and energy-aware migration, and migration only for relatively long running VMs, provides for optimal energy efficiency.

Keywords: Datacenters, Resource management, Server consolidation

1 Introduction

Cloud service providers are looking for opportunities to make cost-effective use of energy [1]. Costs of operating large datacenters are substantial, largely due to energy costs, and are suggested to be in the billions of dollars per year for all datacenters in the United States [2]. There are also environmental reasons for decreasing the amount of energy used by datacenters, with predictions that the ICT industry will be accountable for an estimated 2-3% of the global CO_2 emissions by 2020 [3]. Both environmental and economic reasons motivate scholars and industrialists to explore effective methods for saving energy in datacenters. This is more profound for Cloud service providers who have large numbers of such datacenters. In Infrastructure Clouds, datacenters comprise large numbers of hosts that cloud customers can use in the amounts they require for as long as they are willing to pay. When a cloud customer makes a request for (part of) a host, a VM is launched on a host selected by the Cloud service provider. The user decides how long to run the VM for. The unpredictability of users in such on-demand environments can lead to a number of hosts either being idle or running a minimal VM loading – in principle, wasting energy as an idle host

may still consume 60% of its peak power usage [4]. When hosts are not needed because demand is low, it may be possible to switch hosts off or enable lower energy states. However, hosts would need to be powered back on, or up, quickly when demand spikes. Switching hosts off has the potential to offer operational cost savings with some resource management efforts, but a researcher from one Cloud provider [5] suggests it is unreasonable to switch hosts off due to demand variation. Power cycling a host also carries costs in energy and may cause performance degradation if the boot time is quite long. Similarly, if the workload demand for resources (CPU) is low, then runing a host in a lower energy state, for example using Dynamic Voltage and Frequency Scaling (DVFS), can reduce energy consumption but with non-trivial performance loss [6]. Those paying for Cloud services are unlikely to be keen on resources of diminished performance, unless costs are correlated with performance.

Virtualization allows several VMs to be run on a single host, making server consolidation possible [7], and virtualization is a key component of most Infrastructure Clouds . Taken over a number of hosts, server consolidation attempts to find a minimum number of hosts that would still be able to run all of the VMs in the datacenter, offering further potential to make energy savings. In [8] the authors show that in Google's cluster [9], hosts are not highly utilized and some significant power can be saved through consolidation techniques. Similarly, task runtime distributions show that the majority of tasks run only for a short duration − which could lead to unnecessary migrations that should be avoided. Server consolidation is similarly achieved through server (here, VM) migration. However, server migration also has a cost and may impact on Service Level Agreements (SLAs). Further, with unpredictable VM runtimes in an on-demand environment it is possible that the cost is never recovered through increased efficiency if the VM is terminated during, or even just after, migration.

In this paper, we investigate how migration decisions can be made such that the energy costs involved with the migration are recoverable, after which energy is saved. We explore the impact on energy efficiency of VM allocation heuristics such as Round Robin (RR), Random (R), Best Resource Selection (BRS) [10], Minimum Power Difference (MPD) [11], First Fit (FF) and Fill Up (FU) when combined with different approaches to migration. Key to this exploration is the recovery of costs incurred by a migration. This exploration is conducted through simulations that use the Google workload traces for 12,583 hosts and 1,083,309 tasks [9] in combination with CloudSim [12].

The rest of the paper is organized as follows. In section 2 we explain VM migration, its energy overhead, and how to measure (virtualized) host efficiency. In section 3, we discuss server consolidation as a binpacking problem, and propose as Consolidation with Migration Cost Recovery (CMCR) technique that avoids migrating VMs which would not recover the energy used in migration. We validate CMCR using real workload traces from Google cluster in section 4 and show that CMCR can reduce the migration energy overhead with reduced numbers of migrations, and that the majority of migrated VMs now recover their migration

cost and continue to save energy and therefore cost. We offer an overview of related work in section 5, and section 6 concludes the paper.

2 Background

The migration of a VM may happen for a number of reasons within a datacenter, including host maintenance or load balancing. During VM migration, the running VM is moved from one host to another. This means migrating memory pages and, depending on the underlying approach to storage, data on disk. This leads to two kinds of migration: (i) live/on-disk migration, where a VM image is run from shared storage, for example in Amazon's Elastic Block Storage (EBS), and only memory is copied; (ii) block live/over-Ethernet migrations, where a VM image is run from a local disk, for example Amazon's Instance Store, and both memory and disk are copied. Since the VM image may itself be large, this latter form of migration may take rather longer.
If we perform migrations for reasons of energy efficiency, there will be an energy cost in the additional VM running on the source host for the duration of migration. The cost will relate to the source host's power profile P. We assume P of each host is a linear function of its utilization level – the more the host is utilized, the more energy it will consume, according to the power model proposed in [13]. The relationship of CPU utilization to power consumption can be expressed as shown in equation (1).

$$P(u) = P_{idle} + (P_{max} - P_{idle}).u \tag{1}$$

where $P(u)$ is the estimated power consumption, P_{idle} is static power consumed when host is idle, P_{max} is the power consumed when the host is fully utilized, and u is the current CPU utilization. The portion $(P_{max} - P_{idle}).u$ is called dynamic power consumption, and is treated as a linear function of utilization. This simplified model predicts non-virtualized host power consumption with less than 5% error, but requires modification to account for virtualization. In the first part of this section, we extend this power model to address virtualized hosts; in the second part we discuss measuring the migration energy cost.

2.1 Comparing Hosts Efficiencies

Our work explores migration cost recovery, which is only possible if two conditions are both met: (i) a VM is migrated to a more efficient target host; (ii) the migrated VM then runs for a sufficient length of time on the target host. In this section we discuss measuring the efficiency of hosts in order to address these conditions.
In non-virtualised platforms, if one host consumes less power than another to execute a specific workload, it is more efficient. However, efficiency should be addressed across a range of workloads as there may be other workloads that run less efficiently. In virtualised environments, multiple VMs can be running different workloads on a single host, and so several factors must be considered in order to compare power efficiency; we consider, first, division of the host to

VMs and so the total power consumption of a virtualized host is characterised by:

$$P_{host} = P_{idle} + \sum_{i=1}^{n} P_i^{vm} \tag{2}$$

Where P is the total power consumed by the host, n is the number of active VMs on host, P_{idle} is the host static power consumption and P_i^{vm} is the dynamic power consumption of VM i which is calculated by the linear power model discussed in section 2:

$$P_{vm} = W_{vm}.P_{dynamic} \tag{3}$$

Where W_{vm} is the fraction of host total CPU allocated to the VM. This allows us to simplify concerns by considering each VM equivalent with respect to a host; in an Infrastructure Cloud, VM size may be equally divided by the number of allocated (hyperthreaded) cores out of m cores on the host, or by allocation of an amount of memory. For simplicity, we use the number of (hyperthreaded) cores.

$$W_{vm} = \frac{cores_{vm}}{m} \tag{4}$$

Including static power, the total power consumed by a single VM will be given by:

$$P_{vm} = \frac{P_{idle}}{n} + W_{vm}.(P_{max} - P_{idle}).u \tag{5}$$

Where n is the total number of VMs running on host, u is the utilization level of vm. Hence, efficiency of a host can be related to the number of VMs that are allocated to it and, if need be, to their individual efficiencies.

In this model, due to P_{idle}, the energy used in order to run a single VM is going to be at its highest, and the more VMs that are run on the host, the more power efficient each VM is. We also make use of the notion of VM density, used elsewhere both to address the number of VMs running on a host, and the maximum number that can be run whilst avoiding resource starvation; we combine these to understand VM density as the present fraction of the maximum for a host.

Suppose there are n VMs allocated to host H_1 and m VMs allocated to host H_2. Each VM is utilizing 100% of its proportional resources allocated. The per VM power consumption of each VM on H_1 and H_2 are $P^{H_1}_{vm_{i=1:n}}$ and $P^{H_2}_{vm_{j=1:m}}$ respectively according to above equation. The total power consumption of each host H_1 and H_2 is given by $P_{H_1} = \sum_{i=1}^{n} P_i^{vm}$ and $P_{H_2} = \sum_{j=1}^{m} P_j^{vm}$ respectively. For a VM vm_k selected for migration from H_1, with sufficient space to allocate on H_2, and provided that $P^{H_1}_{vm_k} > P^{H_2}_{vm_k}$, then H_2 is more power efficient than to H_1 with a factor of E_f given by:

$$E_f = \frac{P_{H_1}}{P_{H_2}} \tag{6}$$

2.2 The Migration Model

During a live VM migration [14], an extra VM is created on the target host and is progressively synchronized. Once synchronized, the VM is started on target host

and its copy is terminated on the source host. This means that a migration costs roughly double the resources for the duration of migration. If the VM terminates during the migration process, or before this resource cost is recovered, this effort is wasted. A number of studies [11] [7] [15] discuss consolidation but appear to ignore the cost that is due to the migration energy overhead, and with the notable exception of [16] this is rarely addressed. The migration cost is dictated by the cost of the most expensive VM (at source host) running for the duration of migration, plus any associated network cost during migration. The overhead also includes some marginal extra costs of migration C_m if this requires changes in power state of either or both hosts [17].
For homogeneous hosts, the time required for a migration can be given by:

$$t_{mig} = \frac{V_{mem} + V_{disk}}{B} \tag{7}$$

Where t_{mig} is dependent on VM memory size V_{mem}, VM ephemeral disk image V_{disk} (in case of block live migration) and the available network bandwidth B for data transfer. For live migration, V_{disk} is zero and V_{mem} is calculated using the VM memory size and the dirty pages that are continuously copied in multiple rounds n, during the migration process. If the VM is idle then the dirty pages are zero and hence the network traffic is only equal to V_{mem} measured in MB, otherwise:

$$V_{mem} = \sum_{i=0}^{n} V_i \tag{8}$$

$$V_i = D.T_{i-1} \tag{9}$$

Where i denotes the round, D is the rate at which the memory pages are being dirtied in MB/s, T is round duration in seconds and V represents the size of dirty pages in MB. The migration energy overhead $Cost_{mig}$ is given by:

$$Cost_{mig} = t_{mig}.(P_{source} + P_{net}) + C_m \tag{10}$$

Where C_m denotes the marginal cost needed to switch on/off hosts, P_{net} is the network power consumption and P_{source} is the cost of the most expensive VM running at source host. For the present paper, we simplify concerns by $C_m = 0$; subsequently we would need address this as part of the overall energy use. The amount of data transferred $data_t = V_{mem} + V_{disk}$ has a significant effect on t_{mig}. In [16], the authors have validated a model for measuring the energy consumption of a live migration with 0.993 R^2 value, which is proportional to $data_t$.

$$Cost_{mig} = 0.512 * data_t + 20.165 \tag{11}$$

Based on experimental results, the authors claim that migration is I/O intensive with energy mostly consumed in data transfer. Because of this simplicity and accuracy, we use this directly to compute migration cost. Another approach is proposed in [18], which offers a linear relationship between V_{mem} and B, hence the energy consumed is equal to $\alpha.V_{mem} + \beta.B + C$. This model does not take load into account, so only suits scenarios when the migrating VM is idle.

3 Problem Description

Server consolidation with migration can be considered as a multidimensional bin-packing problem that tries to minimize the number of hosts needed to accommodate a set of VMs [7]. Such NP-complete problems are typically solved using Linear Programming (LP) or heuristics. Dynamic consolidation is typically suggested to be an improvement on doing nothing, allowing: (a) to switch off the underutilized host if the accommodated VMs can be relocated to other hosts; (b) to withdraw hosts from an overloaded state if the sum of accommodated VMs becomes larger than its capacity [14]. Besides the trade-off involved between migrating VMs and decreasing the number of hosts to accommodate VMs, live VM migration can be completed without needing downtime, and ideally without impacting performance (and, specifically, SLAs).
If every VM can first recover the migration cost, and then continues to run on the energy efficient host, then the migration is effective in energy saving and hence in cost reduction. Dynamic consolidation can be considered as an optimization problem in minimizing the amount of energy consumed by avoiding migrations. We describe the problem as CMCR, further explained in section 3.1, and address it by exploring the effect of VM runtimes. In an on-demand environment, VM runtimes are unknown, so we can only consider the past runtime R_{past} in order to decide on migration.

3.1 CMCR

We consider migrations for the purpose of consolidating to fewer hosts to minimize the cost of energy consumption. The migration cost must be considered as part of the migration decision. If the target host is similar or less energy efficient than the source host, based on the total number of accommodated VMs, then the migration cost cannot be recovered. Otherwise the migration cost will eventually be earned. Using the efficiency factor of the source and target hosts, we can find a time point t_{off} on the target host at which the VM has earned back the cost of migration $Cost_{mig}$ and will now be saving energy if it continues to run.
Consider a VM vm_1 that runs on source host H_1. A migration decision is triggered to target host H_2 at time t. Assume that we know H_2 is more energy efficient than H_1 with a factor of E_f. If there are no VMs running on both hosts, then the host with less static power consumption is considered more energy efficient. If there are some VMs running on source and target hosts then the efficiency of each host depends on the number of running VMs (n VMs on source host and m VMs on target host). The E_f as explained in section 2.1, can be computed as:

$$E_f = \frac{P_{VM_{source}}}{P_{VM_{target}}} \tag{12}$$

If $E_f = 1$, it means that the power profile is identical and we cannot recover the migration cost. If $E_f < 1$, the target host is less efficient. The offset of migration cost and further savings can only be made if $E_f > 1$. $Cost_{mig}$ is measured

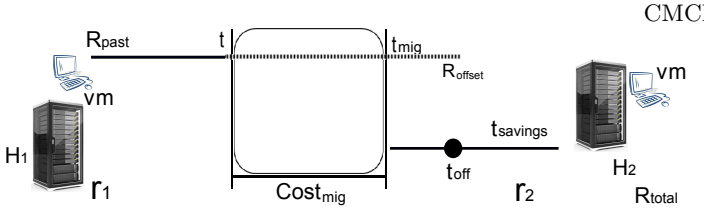

Fig. 1: CMCR technique description

in Watts per hour and is computed as explained in section 2.2. The difference between the power consumption values of both source and target hosts is:

$$\triangle x = P_{VM_{source}} - \frac{P_{VM_{source}}}{E_f} = P_{VM_{source}} - P_{VM_{target}} \tag{13}$$

And so t_{off} is given by:

$$t_{off} = \frac{t_{mig}.Cost_{mig}}{\triangle x} \tag{14}$$

For vm_1 with past runtime R_{past} on source host, and migration to target host started at time t, migration completes in time t_{mig} as shown in Fig. 1. vm_1 total runtime on the source host is $r_1 = R_{past} + t_{mig}$, and the remaining runtime on target host is $r_2 = R_{total} - (t - t_{mig}) = R_{total} - r_1$. If $r_2 > t_{off}$, then it means vm_1 has recovered $Cost_{mig}$ and subsequently runs more efficiently to save energy. The remaining runtime of vm_1 on the target host after the t_{off}, is given by:

$$t_s = r_2 - t_{off} \tag{15}$$

The savings $P_{savings}$ with an energy efficient migration are then only:

$$P_{savings} = t_s.\triangle x \tag{16}$$

Hence the minimum value for $r_1 + r_2$ (Fig 1) which is sufficient to offset $Cost_{mig}$ at time t is $R_{offset} = t_{mig} + t_{off}$. For any VM running for R_{past}, the R_{offset} is given by:

$$R_{offset} = R_{past} + t_{mig} + t_{off} \tag{17}$$

If $R_{offset} \geq t_{off}$, then the migration is energy efficient. If the vm_1 is terminated before t_{off}, migration cost is not recovered. If R_{offset} is not sufficient to recover $Cost_{mig}$ then t can be estimated to make a migration efficient, using $t = t - t_{off}$ and $R_{past} = R_{past} - t_{off}$.
In the above formulation R_{total} denotes the time for which the VM will run, which is unknown. To make the scenario realistic for on-demand systems, we only consider the past runtime R_{past} of VMs in order to determine if a VM is a suitable candidate for migration.

4 Performance Evaluation

Bin-packing problems are solved using various heuristics which may not ensure optimal results but are fast enough to deal with large problems [7]. It is possible to consider an analogous VM packing problem as moving from a given datacenter

state to an ideal state, which should be one using fewest hosts. We achieve a datacenter state by implementing scheduling heuristics (RR, R, BRS, MPD, FF, FU, as initially stated in section 1), with VM packing then needing to guarantee energy efficiency is assured (as explained in section 2.1) and migration cost can be recovered. To evaluate the effect of this, we consider (i) no migration; (ii) dynamic consolidation (all possible migrations); and (iii) CMCR (runtime-based migration).

4.1 Experimental Setting

We use real workload traces from Google to study the feasibility of our approach within the event driven simulator CloudSim [12]. The Google dataset comprises 12,583 hosts in one datacenter and 1,083,309 tasks and as explained in [9] a task runs in a Linux container (section 4.2), its CPU requirements are measured in core seconds per second, and the values are normalized to the maximum cores host available in the Google's cluster.

To address a Cloud context, each task is assigned a single, notional, VM that maps to Google instance types. We assume that hosts are comparable by a single measure which allows for performance ranking, for which we adopt CloudSim's use of Million of Instructions Per Second (MIPS) as a proxy; we would not endorse this as a good performance indicator for real systems for a number of CPU architecture and workload comparability reasons. One approach is to assign a VM as a single core for the maximum value 1, half a core (hyperthread) for 0.5, and assume that higher VM gearing leads to a quarter of a core for 0.25. But to address allocation more flexibly, along lines of certain Cloud providers, we map CPU frequency for the hosts given to Google Compute Engine Units (GCEUs) as: 2 GHz CPU, 1.25GB RAM, giving types A1 (0.5 GCEU), B1 (0.25 GCEU) and C1 (1 GCEU). The GCEU then maps MIPS for consistency with CloudSim, and we assume that every instance needs at most 1 GCEU. Memory requirements then also map to these types, as shown in Table 1.

CLASS	INSTANCE NAME	GCEUs	MEMORY (GB)
A1	a1.tiny	0.5	0.03
	a1.xtiny	0.5	0.06
	a1.micro	0.5	0.12
	a1.small	0.5	0.25
	a1.medium	0.5	0.5
	a1.large	0.5	0.75
	a1.xlarge	0.5	0.97
B1	b1.small	0.25	0.25
C1	c1.medium	1.0	0.5
	c1.large	1.0	1.0

Table 1: Instance types

When a task is submitted, the task scheduler finds the most suitable instance type and the allocation policy places it on a host: RR allocation policy places the VM on the next available host; R allocation policy selects a suitable host randomly; MPD [11] is a modified Best Fit Decreasing (BFD) off-line heuristic that

(at 1 second interval – to mimic on-line behavior) sorts all VMs in decreasing order of CPU utilization and allocates each VM to a host that increases energy consumption the least – selecting the most energy efficient host first, based on the linear power model (section 2 equation 1); BRS [10] places a VM on a host with the least free capacity to maximize resource utilization; FF and FU are both on-line heuristics and place the VM on the first available host, with FU selecting the most efficient host based on the model proposed in section 2.1. The host efficiency model and the on-line behavior of FU differentiate it from MPD. The task scheduler implements a First In First Out (FIFO) mechanism to dispatch submitted tasks for execution. A cluster of 12,583 heterogeneous hosts, which consists of three different architectures and characteristics as shown in Table 2, is available. The heterogeneous hosts available in datacenter are set up based on assumptions that Google had certain kinds of commonly available machines in their datacenters in May 2011, when the trace was captured [19].

HOST TYPE	HOST NAME	SPEED (GHz)	NO OF CORES	NO OF GCEUs	MEMORY (GB)	P_{IDLE} (Wh)	P_{MAX} (Wh)	AMOUNT
A	Intel Xeon E3110	3.0	2	3	4	75.2	117	4,195
B	Intel Xeon X3470	2.9	4	5.75	8	41.6	113	4,194
C	Intel Xeon E5540	2.5	8	10	8	67.0	218	4,194

Table 2: Host characteristics and number suggested to be in Google's cluster in May 2011 [19]

The power consumption values for these hosts are taken from SPEC power benchmarks [20]. The tasks are submitted according to arrivals in the Google dataset. When VMs terminate, slots are made available to the scheduler and are also available for migrations. The migration policy regularly (every 5 minutes) checks all host utilizations, and if a host utilization level goes below a predefined lower threshold value e.g. 20%, VMs can be migrated to other hosts to consolidate the current demand on fewer hosts to save energy. In principle, if host utilization exceeds a predefined upper threshold value i.e. 100%, some VMs are migrated from the overloaded host to less utilized hosts to avoid SLA violations. We assume, here, that sensible ways of addressing VM density will not lead to overloading. A migration decision is based on only the lower utilization threshold value, current state of the datacenter (consolidation opportunities) and other constraints as explained in section 3.1. If several VMs are selected for migration, the list is sorted in decreasing order of their past runtimes, and migrated in order until all VMs in the list are migrated. For the sake of simulation, migration duration is computed by dividing the VM memory size by network bandwidth (set at 1Gbps) as discussed in section 2.2. The migration energy overhead and host efficiency factor is calculated as discussed in section 2 equation 7.

4.2 Experimental Results

The simulated infrastructure is composed of 12,583 hosts with configuration shown in table 2. We first run the simulation with a single day of data from the Google trace. We assume that the VM workload is homogeneous and so it

does not change even when a VM is migrated from one host to another. The selected trace (day 2) comprises 1,083,309 tasks with average arrival rate of 12.54 tasks per second and terminations at 12.24 tasks per second. After each 5 minute interval, CMCR checks for consolidation opportunities, and selects VMs running for longer times from a list of migration possibilities. Each experiment was performed with five different values for past VM runtime given in hours [0, 0.5, 1, 2, and 4], where 0 means migrate all – dynamic consolidation – and 0.5 means migrate only those VMs which are running for 30 minutes or longer, 1 means running for 1 hour, and so on.

Metrics The metrics are the number of migrations, average number of hosts used to run the VMs and total datacenter energy consumed. An overall calculation of datacenter efficiency, D measures the efficiency of a scheduling approach on datacenter level. This accounts for the load proportion (% slots filled i.e. VM density – explained in section 2.1), the number of hosts switched on, the number of idle hosts that still consumes significant energy (idle power consumption), and factor of energy efficiency in respect to whether more or less efficient hosts were in use.

$$VM_{density} = \frac{VMs_{onHost}}{Host_{capacity}} \tag{18}$$

$$D = \frac{\sum_{hosts} VM_{density} * E_f}{Hosts_{used}} + \frac{\sum_{hosts} Hosts_{unUsed} * E_f}{Hosts_{unUsed}} \tag{19}$$

The host efficiency model presented in section 2.1, is used to calculate E_f for each host. Lower values for D represent efficient datacenter resource management with maximum VMs running on a minimum number of the most efficient hosts, and hence also offers potential for hosts to be powered off in the second term.

Discussion Fig 2 presents the results obtained from running the Google cluster's tasks submitted on day 2 using different scheduling heuristics. The results show that efficient scheduling techniques would be more economical than consolidation techniques. For example, without migration a 52.43% decrease in energy consumption was achieved using FU instead of RR. But using FU, only 3.04% decrease in energy consumption was achieved with dynamic consolidation. The metric D shows an average decrease of 16.10% in energy consumption for FU compared to R scheduler. Similarly for CMCR, FU is on average 0.49% more cost efficient as compared to FF scheduler. We also note that no migration can be more economical than the dynamic consolidation if an efficient scheduling approach is used. CMCR beats both techniques as it allocate VMs to the most efficient hosts first, minimizes the total number of migrations (runtime-based migration) and increases the probability that a VM recovers its migration cost. Table 3 shows the mean number of hosts in use, and datacenter utilization, measured in 5 minute intervals. For each allocation policy, CMCR have reduced the total number of migrations and migration energy. The D value (section 4.2) shows that in terms of scheduling approach, FU is effective in using a minimum number of most efficient hosts: FU did not allocate VMs to host type A which has larger idle power consumption and is less energy efficient compared to types B &

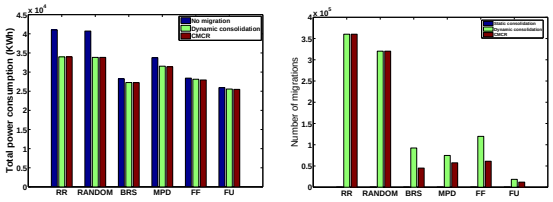

Fig. 2: Power consumption & no. migrations for Google trace day 2

C. As MPD is an off-line heuristic, higher D values confirm inability to address online problems. For each scheduling approach, cost savings are compared to a baseline no-migration policy. On average, CMCR reduces the number of hosts in use, with a reduced number of migrations. The runtime of VMs migrated depends on the scheduling heuristics. RR scheduler equally distributes VMs among the available hosts, keeping all the hosts active and less-utilized most of the time – making more consolidation opportunities. Similarly, R scheduler al-mostly selects a different host for VM placement randomly, which results in more energy consumption and increased number of migrations – as all hosts are switched on but less utilized. The optimal value for these two algorithms is always achieved with dynamic consolidation, i.e past runtime ≥ 0 minutes. BRS, MPD and FF were able to give minimum power consumption results by migrating VMs with past runtime ≥ 30 minutes. The most efficient heuristic, FU, produces optimal results by migrating VMs with past runtime ≥ 60 minutes.

Scheduling approach	Consolidation technique	Total hosts used A	B	C	Avg used hosts	Datacenter Util (%)	D (KWh)	Cost savings (%)
RR	No migration	4,195	4,194	3,351	3,157	24.81	1464.00	0
	Dynamic	4,195	4,194	2,750	2,228	47.40	1450.11	25.77
	CMCR	4,195	4,194	2,750	2,228	47.40	1450.11	25.77
RANDOM	No migration	4,195	4,194	3,835	3,005	25.18	1474.19	0
	Dynamic	4,195	4,194	3,713	2,148	48.64	1471.70	23.75
	CMCR	4,195	4,194	3,713	2,148	48.64	1471.70	23.75
BRS	No migration	2,664	2,662	2,667	1,157	50.31	1240.98	0
	Dynamic	2,504	2,888	2,658	1,095	69.61	1242.02	5.87
	CMCR	2,612	2,899	2,683	1,089	62.95	1242.97	6.04
MPD	No migration	4,195	4,194	1,908	2,412	28.21	1427.92	0
	Dynamic	4,195	4,194	2,227	1,965	33.09	1436.74	9.96
	CMCR	4,195	4,194	2,581	1,984	34.84	1445.93	10.66
FF	No migration	2,665	2,664	2,666	1,212	51.61	1241.03	0
	Dynamic	2,619	2,790	2,620	1,219	61.22	1243.03	1.69
	CMCR	2,636	2,790	2,635	1,190	61.64	1242.91	2.85
FU	No migration	0	4,194	2,700	1,241	48.61	1236.88	0
	Dynamic	0	4,194	2,700	1,167	67.21	1236.88	2.96
	CMCR	0	4,194	27,00	1,166	66.74	1236.88	3.04

Table 3: Experimental mean results for different approaches (5 min. interval)

The data and migration statistics produced in Table 4, show that combining CMCR and FU means only 1.1% of VMs are migratable and 99.5% of these were able to recover their migration cost. For FU with dynamic consolidation,

1.71% VMs were migrated with 98.98% of recovering migration cost. The migration statistics given in Table 4, also include multiple entries for VMs that were migrated multiple times during their lifetime. If we now assume a PUE [21] of 1.2 and energy cost of $0.08kWh, dynamic consolidation would save $1519 per hour for this cluster (a little over $36k per day) in comparison to a no-migration approach. Using above assumptions, CMCR would further save $42 per hour compared to dynamic consolidation. Hence, CMCR makes total savings of $1,561 per hour (almost $37.5k) as compared to a no-migration approach.
For 5,48,584 tasks submitted on the first day of Google cluster trace [9], with the same simulation, we see that no-migration technique would be more economical than dynamic consolidation if efficient VM scheduling heuristics are used. Our second finding is that migrating relatively long running VMs to more energy efficient hosts to recover their migration cost, are more economical and energy efficient.

Scheduling approach	RR		R		BRS		MPD		FF		FU	
	DC	CMCR	DC	CMCR	DC	CMCR	DC	CMCR	DC	CMCR	DC	CMCR
Migratable VMs (%)	33.2	33.2	29.6	29.6	8.5	4.2	6.9	5.3	11.0	5.6	1.71	1.1
VMs recovered $Cost_m$ (%)	98.9	98.9	98.6	98.6	98.5	99.0	98.5	99.5	98.5	99.4	99.0	99.5

Table 4: Cost recovery with Dynamic Consolidation (DC) & CMCR

5 Related Work

Researchers elsewhere have addressed various aspects of energy savings, mindful in some cases that idle hosts consume up to 60% of the power of the fully loaded host. Khanna et al. [14] perform migration to avoid overloading leading to SLA violations, and also to switch of underloaded hosts. The specification of a compute unit such as GCEU should help to avoid overcommitting resources and avoid overloading, except where resource contention exists. Our datacentre measure offers a means to measure the gain by switch off. Wood et al. offer Sandpiper [22], a system to monitor and detect hotspots, also remap and reconfigure VMs when required. The proposed system migrates VMs based on high memory, network and CPU loads; again, not overcommitting should help to avoid migrations in the first place. Bobroff et al. [23] investigate estimating demand based on historical data in order to address dynamic server consolidation. Revisiting this work with respect to the Google data could certainly offer interesting insights for pre-empting demand, and help to reduce costs incurred due to unnecessary power state changes, which is beyond the scope of the present paper. Beloglazov [24] discusses adaptive thresholds for VM consolidation, but this does not address the migration cost, and nor does Tiagos [7] work on minimizing the number of VM migrations by not migrating VMs with steady usage, which might be considered a counterpoint to our findings.
All of these techniques are focused on live migration, but power consumed in migration and its recovery through runtime is not addressed. When a significant proportion of tasks are relatively short-lived, as is the case in the Google

data, the inability to recover such costs would appear quite detrimental. The ReCon system [25] and pMapper [26] are notable exceptions in addressing migration costs, and this should motivate further appraisal of their techniques with respect both to short-lived tasks and also to block/live migration.

6 Conclusion and Future Work

Consolidation with migration is often claimed to increase the energy efficiency in datacenters. Analysis of Google workload data shows that most tasks run only for a short time, and allowing all possible migrations could create additional costs in energy. In this paper, we considered combinations of scheduling approaches and consolidation methods with knowledge of the past runtime of VMs to investigate energy saving potential. Under certain circumstances, we found that not migrating would be more energy-efficient than using dynamic consolidation, and the best approach overall limits migrations to 1.1% of VMs, of which 99.5% recover their migration cost.

The immediate priority is to investigate whether these findings would be consistent over time by evaluating using the whole Google trace (29 further days). We also need to be able to account for both heterogeneous hosts, which cause variability in runtimes for a workload, and the impact of the marginal energy costs involved due to power state changes in respect to unused hosts.

References

1. Saurabh Kumar Garg and Rajkumar Buyya. Green cloud computing and environmental sustainability. *Harnessing Green IT: Principles and Practices*, pages 315–340, 2012.
2. NRDC. America's Data Centers Are Wasting Huge Amounts of Energy: critical action needed to save billions of dollars and kilowatts. *IB:14-08-A*, pages 1–6, 2014.
3. Sherali Zeadally, Samee Ullah Khan, and Naveen Chilamkurti. Energy-efficient networking: past, present, and future. *The Journal of Supercomputing*, 62(3):1093–1118, 2012.
4. David Meisner, Brian T Gold, and Thomas F Wenisch. Powernap: eliminating server idle power. In *ACM Sigplan Notices*, volume 44, pages 205–216, 2009.
5. https://www.youtube.com/watch?v=7MwxA4Fj2l4. [Online; accessed 3-Oct-15].
6. Anton Beloglazov, Rajkumar Buyya, Young Choon Lee, Albert Zomaya, et al. A taxonomy and survey of energy-efficient data centers and cloud computing systems. *Advances in computers*, 82(2):47–111, 2011.
7. Tiago C. Ferreto, Marco A S Netto, Rodrigo N. Calheiros, and César A F De Rose. Server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems*, 27(8):1027–1034, 2011.
8. Charles Reiss, a Tumanov, and Gr Ganger. Towards understanding heterogeneous clouds at scale: Google trace analysis. *... Center for Cloud ...*, 2012.
9. Charles Reiss, John Wilkes, and Joseph L Hellerstein. Google cluster-usage traces: format+ schema. *Google Inc., Mountain View, CA, USA, Technical Report*, 2011.
10. Daniel Guimaraes do Lago, Edmundo R M Madeira, Luiz Fernando Bittencourt, and Daniel Guimaraes do Lago. Power-aware virtual machine scheduling on clouds using active cooling control and DVFS. *Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science*, pages 2:1———2:6, 2011.

11. Anton Beloglazov and Rajkumar Buyya. Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints. *IEEE Transactions on Parallel and Distributed Systems*, 24(7):1366–1379, 2013.
12. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A F De Rose, and Rajkumar Buyya. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software - Practice and Experience*, 41(1):23–50, 2011.
13. Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. Power provisioning for a warehouse-sized computer. In *ACM SIGARCH Computer Architecture News*, volume 35, pages 13–23. ACM, 2007.
14. G Khanna, K Beaty, G Kar, and a Kochut. Application Performance Management in Virtualized Server Environments. *2006 IEEEIFIP Network Operations and Management Symposium NOMS 2006*, 20(D):373–381, 2006.
15. Benjamin Speitkamp and Martin Bichler. A mathematical programming approach for server consolidation problems in virtualized data centers. *IEEE Transactions on Services Computing*, 3(4):266–278, 2010.
16. Haikun Liu, Hai Jin, Cheng-Zhong Xu, and Xiaofei Liao. Performance and energy modeling for live migration of virtual machines. *Cluster Computing*, pages 249–264, 2011.
17. Luiz André Barroso and Urs Hölzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007.
18. Anja Strunk and Waltenegus Dargie. Does Live Migration of Virtual Machines Cost Energy? *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, pages 514–521, 2013.
19. Google-cluster-data. https://groups.google.com/. [Online; accessed 7-May-16].
20. Klaus-Dieter Lange. Identifying shades of green: The specpower benchmarks. *IEEE Computer*, 42(3):95–97, 2009.
21. Christian Belady, Andy Rawson, John Pfleuger, and Tahir Cader. Green Grid Data Center Power Efficiency Metrics: PUE and DCIE. 2008.
22. Timothy Wood, Prashant Shenoy, Arun Venkataramani, and Mazin Yousif. Sandpiper: Black-box and gray-box resource management for virtual machines. *Computer Networks*, 53(17):2923–2938, 2009.
23. Norman Bobroff, Andrzej Kochut, and Kirk Beaty. Dynamic placement of virtual machines for managing SLA violations. In *10th IFIP/IEEE International Symposium on Integrated Network Management 2007, IM '07*, pages 119–128, 2007.
24. Anton Beloglazov and Rajkumar Buyya. Adaptive Threshold-Based Approach for Energy-Efficient Consolidation of Virtual Machines in Cloud Data Centers. *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*, (December 2010):6, 2011.
25. Sameep Mehta and Anindya Neogi. ReCon: A tool to recommend dynamic server consolidation in multi-cluster data centers. In *NOMS 2008 - IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubiquitous Networks and Services*, pages 363–370, 2008.
26. Akshat Verma, Puneet Ahuja, and Anindya Neogi. pMapper: Power and migration cost aware application placement in virtualized systems. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 5346 LNCS, pages 243–264, 2008.