Evolutionary Multiobjective Optimization for Green Clouds

Dung H. Phan
Dept. of Computer Science
University of Massachusetts, Boston
phdung@cs.umb.edu

Junichi Suzuki
Dept. of Computer Science
University of Massachusetts, Boston
jxs@cs.umb.edu

Raymond Carroll
TSSG, Waterford Institute of Technology, Ireland
rcarroll@tssg.org

Sasitharan Balasubramaniam
TSSG, Waterford Institute of Technology, Ireland
sasib@tssg.org

William Donnelly
TSSG, Waterford Institute of Technology, Ireland
wdonnelly@tssg.org

Dmitri Botvich
TSSG, Waterford Institute of Technology, Ireland
dbotvich@tssg.org

ABSTRACT

As Internet data centers (IDCs) have been increasing in scale and complexity, they are currently a significant source of energy consumption and CO_2 emission. This paper proposes and evaluates a new framework to operate a federation of IDCs in a "green" way. The proposed framework, called Green Monster, dynamically moves services (i.e., workload) across IDCs for increasing renewable energy consumption while maintaining their performance. It makes decisions of service migration and placement with an evolutionary multiobjective optimization algorithm (EMOA) that evolves a set of solution candidates through global and local search processes. The proposed EMOA seeks the Pareto-optimal solutions by balancing the trade-offs among conflicting optimization objectives such as renewable energy consumption, cooling energy consumption and response time performance.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; I.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*

General Terms

Algorithms, Management

Keywords

Evolutionary multiobjective optimization, Cloud computing, Internet data centers, sustainability, renewable energy

1. INTRODUCTION

Internet data centers (IDCs) have become an integral component to operate Internet services and scientific computation. Since they have been increasing in scale and complexity, they consume a growing and visible portion of energy supply [11, 24]. Energy-intensive IDCs are a major source of CO_2 emission. As the majority of computational processing and data storage have been moving to IDCs, with the client devices running simpler interfaces, IDCs continue to be a significant source of energy consumption and CO_2 emission in the near future. This trend has prompted increased scrutiny from regulators and non-governmental organizations (NGOs) [7, 9, 24].

In order to replace conventional fuels and reduce CO_2 emission, many countries actively pursue more renewable sources of energy through their own capital infrastructure projects or through grid feed-in tariff incentive schemes. As a result, the capacity of renewable energy has increased exponentially in the past decade [18].

Given the aforementioned issues and trends, this paper proposes a framework to operate a federation (i.e., cloud) of geographically-dispersed IDCs in a "green" way. The proposed framework, called Green Monster, dynamically moves services (i.e., workload) to IDCs with more desirable energy profiles while maintaining performance (e.g., response time). It makes decisions of service migration and placement with an evolutionary multiobjective optimization algorithm (EMOA) that evolves a set of solution candidates through both global and local search processes. Each solution candidate (or *individual*) represents a particular placement configuration of individual services. The proposed EMOA considers conflicting optimization objectives (e.g., renewable energy consumption and performance[1]) and seeks the Pareto-optimal solutions by balancing the trade-offs among those objectives subject to a given capacity constraint in each IDC.

Since there exits no single optimal solution under conflicting objectives but rather a set of alternative solutions of equivalent quality, the proposed EMOA is designed to search Pareto-optimal solutions that are equally distributed in the objective space. Therefore, it can produce both extreme service placement configurations (e.g., the one yielding high renewable energy consumption with high response time) and balanced service placement configurations (e.g., the one yielding intermediate renewable energy consumption with intermediate response time) at the same time. Given a set of heuristically-approximated Pareto-optimal solutions, an IDC operator can examine the trade-offs among them

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.
Copyright 2012 ACM 978-1-4503-1178-6/12/07 ...$10.00.

[1] Increasing renewable energy consumption can degrade performance. On the contrary, improving performance can decrease renewable energy consumption.

and make a well-informed decision to choose one of them, as the best service placement, according to his/her preferences and priorities. For example, an IDC operator can examine how he/she can trade (or sacrifice) response time for renewable energy consumption and determine a particular service placement configuration that yields a desirable/comfortable balance of response time and renewable energy consumption.

Simulation results show that Green Monster allows IDCs to place services for reducing their carbon footprint while maintaining their performance. The proposed EMOA outperforms conventional capacity-based service placement algorithms with respect to renewable energy consumption, cooling energy consumption and response time performance.

2. BACKGROUND AND RELATED WORK

IDCs reportedly consumed 203.4 to 271.8 TWh worldwide in 2010, which accounted for 1.1% to 1.5% of the total electricity usage [11]. The estimates are 2.9 to 3.8 times greater than the IDC energy consumption in 2000 (70.8 TWh), which accounted for 0.54% of the total electricity usage. In the U.S., it is estimated that IDCs consumed 67.1 to 85.6 TWh in 2010, which accounted for 1.7% to 2.2% of the total electricity usage [11]. The estimates are 2.4 to 3.0 times greater than the IDC energy consumption in 2000 (28.2 TWh), which accounted for 0.82% of the total electricity usage. The U.S. Environmental Protection Agency reports that IDCs in the U.S. consumed approximately 4.5 billion dollars of electricity (61 TWh) in 2006 and the energy consumption exceeded the electricity collectively consumed by all color television sets in the U.S. [24].

In 2007, Gartner estimated that the information and communications technology (ICT) industry produced 2% of global CO_2 emission, which is on par with the aviation industry[2]. IDCs were responsible for 23% of the ICT's emission

The Renewable Energy Policy Network for the 21st Century (REN21) reported that renewable energy provided 312 GW worldwide in 2010, which accounted for 3% of global electricity generation [18][3]. Wind power is growing at the rate of 30% annually. It provided two thirds of the total renewable power capacity in 2010 (198 GW). Solar photovoltaic capacity increased more than three times from 2007 to 2010.

There exists a considerable volume of research efforts that address energy efficiency issues in IDCs. Many of them focus on consolidating workload (i.e., services) on a limited number of servers in order to allow idle servers to be switched off/sleep and save power consumption [2, 3, 6, 13, 19, 23]. Instead of service consolidation, Green Monster focuses on sustainability-driven dynamic service placement.

CPU voltage/frequency scaling is another power saving strategy [10]. It intends to dynamically reduce the power consumption of each server's CPU via adaptive voltage/frequency adjustment. In contrast, Green Monster does not consider dynamic CPU voltage/frequency scaling but approaches energy efficiency through dynamic service placement.

Another line of relevant research is to reduce the load on cooling systems by scheduling workload within and among IDCs [1, 14, 16, 21, 25]. These work are similar to Green Monster in that it also considers cooling energy saving as one of its optimization objectives. However, unlike those relevant work, Green Monster considers multiple conflicting objectives including cooling energy saving and seeks the optimal trade-off solutions among them.

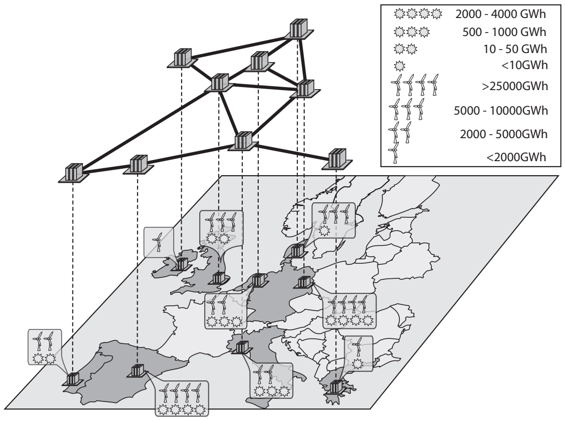


Figure 1: An Example Federation of IDCs

Qureshi et al. move network traffic among IDCs based on the current electricity costs [17]. They do not consider sustainability but energy costs only. Thus, their algorithm can sacrifice the sustainability of IDCs in favor of cost savings.

Garg et al. and Saurabh et al. take similar approaches to Green Monster's in that they consider sustainability (CO_2 emission) to operate applications in IDCs [8, 20]. Unlike them, Green Monster addresses sustainability not only by maximizing renewable energy consumption but also by minimizing cooling energy consumption according to changing indoor and outdoor temperature.

Zeratul and Maolin study a genetic algorithm to seek the optimal service placement with respect to the service execution time that includes the processing time in compute servers and the data access time between compute and storage servers [26]. In contrast, Green Monster considers sustainability (renewable energy consumption and cooling energy saving) as well as service execution time, which is represented by the user-to-service distance objective.

Tantar et al. leverage an EMOA to balance different types of loads (e.g., memory load and processing load) among servers in an IDC [23]. Rather than intra-IDC optimization for load balancing, Green Monster focuses on inter-IDC optimization for sustainable operation of federated IDCs.

3. GREEN MONSTER

This section describes the proposed framework: Green Monster. It assumes federated IDCs: $\{D_1, D_2, ..., D_j, ..., D_M\}$ where M denotes the total number of IDCs. IDCs host services, $\{S_1, S_2, ..., S_i, ..., S_K\}$, each of which implements a particular service type (e.g., data, voice or video service). Figure 1 shows an example federation of IDCs that are geographically distributed to EU countries ($M = 9$).

Figure 2 shows an architectural overview of the interactions between Green Monster and IDCs. Green Monster periodically collects each IDC's operational information such as service request rate and performs optimization for dynamic service migration and placement. It is designed pluggable for various types of optimization engines including EMOAs. Once an optimization engine determines a service placement configuration, Green Monster disseminates it to individual IDCs in order to trigger service migration.

[2] `http://www.gartner.com/it/page.jsp?id=530912`

[3] 19% if hydroelectricity is included

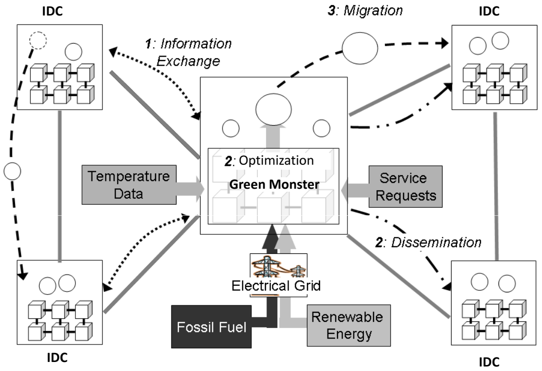


Figure 2: Interactions between Green Monster and IDCs

3.1 Optimization Objectives

Green Monster considers three optimization objectives: renewable energy consumption (RE), cooling energy consumption (CE) and user-to-service distance (USD). RE is to be maximized, CE is to be minimized, and USD is to be minimized. Minimizing CE implies minimizing power usage effectiveness (PUE) [12]. USD implies the response time of services to users.

The first objective, renewable energy consumption (RE), is computed as follows:

$$f_{RE} = \sum_{j=1}^{M} \sum_{i=1}^{K} (I_{ij} \times l_i \times R_j) \tag{1}$$

$I_{ij} = 1$ if the i-th service (S_i) is placed in the j-th IDC (D_j); otherwise, $I_{ij} = 0$. l_i is the daily workload given to S_i:

$$l_i = a_i \times u_i \tag{2}$$

a_i denotes the number of service requests per day given to S_i, and u_i denotes the per-request CPU utilization of S_i. R_j denotes the renewable energy ratio in D_j. It is the ratio of renewable energy production to the total energy production in the country where D_j is located.

The second objective, cooling energy consumption (CE), is computed as follows:

$$f_{CE} = \sum_{j=1}^{M} E_j \tag{3}$$

E_j denotes the cooling energy consumption of D_j:

$$E_j = \frac{H_j}{Q_j} \tag{4}$$

H_j indicates the energy consumed by computing equipment, which is then converted to heat:

$$H_j = \sum_{i=1}^{K} (I_{ij} \times l_i \times n(P_{max} - P_{idle})) + nP_{idle} \tag{5}$$

P_{max} denotes the maximum power that a single server consumes at the peak load, and P_{idle} denotes the power that a single server consumes when it is idle. n denotes the number of servers in an IDC.

Q_j denotes the coefficient of performance (COP) in D_j. A higher COP means that thermodynamic process is more efficient in the cooling system (e.g., air conditioner) in D_j. Under the principles of thermodynamics, COP highly depends on the indoor and outdoor temperature (T_i and T_o) [15]:

$$COP = \frac{1}{\frac{T_o}{T_i} - 1} \tag{6}$$

Given a fixed indoor temperature, the lower the outdoor temperature is, the more efficient a cooling system is. When the outdoor temperature is higher than the indoor temperature, Green Monster uses the DOE/ORNL Heat Pump Design Model[4] to compute COP values.

In addition to conventional air conditioning, Green Monster assumes a free-cooling system when the outdoor temperature is below the indoor temperature[5]. Free cooling allows IDCs to directly utilize the outdoor air to partially, or even fully, condition the indoor temperature. Free-cooling COP (COP_f) is obtained by adjusting conventional (i.e. non-free-cooling) COP:

$$COP_f = \alpha \times COP \tag{7}$$

The third objective, user-to-service distance (USD), is computed as follows.

$$f_{USD} = \sum_{i=1}^{K} (b_i \times d_i) \tag{8}$$

b_i indicates S_i's bandwidth consumption per day between IDCs:

$$b_i = w_i \times a_i \tag{9}$$

w_i denotes the per-request volume of data transmission for S_i. d_j is the shortest logical distance (i.e., hop count) between the initial IDC of S_i (i.e., the IDC that S_i was initially placed on) and the current IDC of S_i (i.e., the IDC that S_i currently resides on).

3.2 Capacity Constraint

Green Monster considers the following capacity constraint for each IDC (D_j).

$$\sum_{i=1}^{K} (I_{ij} \times l_i) < C_j \tag{10}$$

C_j denotes the capacity of D_j (i.e., the maximum daily workload that D_j can accept for K services).

The capacity violation of an individual (i.e., solution candidate in the proposed EMOA) is computed as follows:

$$v = \sum_{j=1}^{M} (I_j \times ((\sum_{i=1}^{K} I_{ij} \times l_i) - C_j)) \tag{11}$$

$I_j = 1$ if D_j violates its capacity constraint (C_j); otherwise, $I_j = 0$.

3.3 Individual Representation

In Green Monster, each individual represents a placement configuration of all K services in M IDCs. Figure 3 shows an example individual, which places S_1 in D_3 and S_2 in D_4 ($K = 5$ and $M = 4$).

3.4 The Proposed EMOA

Algorithm 1 describes the EMOA in Green Monster. Its algorithmic structure is designed based on NSGA-II, a well-known existing EMOA [4].

[4] `http://www.ornl.gov/~wlj/hpdm/MarkVII.shtml`

[5] Cooling (heat pumping) is still required in IDCs even when the outdoor temperature is below the indoor temperature because heat load is generated by the indoor computing equipment rather than the outdoor environment.

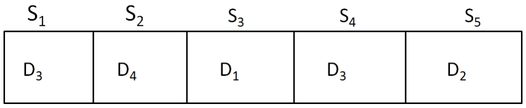


Figure 3: An Example Individual

Algorithm 1 Optimization Process in the Proposed EMOA

```
 1: $g = 0$
 2: $\mathcal{P}_g$ = initialPopulationGeneration($N$, $\mathcal{D}$, $\mathcal{C}$, $\mathcal{S}$, $\mathcal{L}$)
 3: while $g < g_{max}$ do
 4:     $\mathcal{O}_g = \emptyset$;
 5:     while $|\mathcal{O}_g| < N$ do
 6:         $p_1$ = tournament($\mathcal{P}_g$)
 7:         $p_2$ = tournament($\mathcal{P}_g$)
 8:         if random() $\leq P_c$ then
 9:             $\{o_1, o_2\}$ = crossover($p_1$, $p_2$)
10:         else
11:             $\{o_1, o_2\} = \{p_1, p_2\}$.
12:         end if
13:         $o_1$ = mutation($o_1$, $P_m$)
14:         $o_2$ = mutation($o_2$, $P_m$)
15:         $o_1$ = localSearch($o_1$, $P_l$, $\mathcal{D}$)
16:         $o_2$ = localSearch($o_2$, $P_l$, $\mathcal{D}$)
17:         $\mathcal{O}_g = \mathcal{O}_g \cup \{o_1, o_2\}$
18:     end while
19:     $\mathcal{R}_g = \mathcal{P}_g \cup \mathcal{O}_g$
20:     $\mathcal{F}$ = sortByDominationRanking($\mathcal{R}_g$)
21:     $\mathcal{P}_{g+1} = \emptyset$
22:     $i = 1$
23:     while $|\mathcal{P}_{g+1}| + |\mathcal{F}_i| \leq N$ do
24:         $\mathcal{P}_{g+1} = \mathcal{P}_{g+1} \cup \mathcal{F}_i$
25:         $i = i + 1$
26:     end while
27:     sortByCrowdingDistance($\mathcal{F}_i$)
28:     $\mathcal{P}_{g+1} = \mathcal{P}_{g+1} \cup \mathcal{F}_i[1 : (N - |\mathcal{P}_{g+1}|)]$
29:     $g = g + 1$
30: end while
```

At the *0*-th generation, N individuals are generated as the initial population $\mathcal{P}_0$ (Line 2). Algorithm 2 shows the process of generating the initial population. The proposed EMOA performs capacity-aware random service placement. It assigns each service S_i to a randomly-chosen IDC D_j as far as the service's workload l_i does not violate the IDC's capacity constraint C_j (Lines 9 to 11 in Algorithm 2). In case of a capacity violation, the service is assigned to the IDC that has the largest remaining capacity (Line 14 and 15 in Algorithm 2).

At each generation (g), two parent individuals (p_1 and p_2) are selected from the current population $\mathcal{P}_g$ with binary tournaments (Lines 6 and 7 in Algorithm 1). A binary tournament randomly takes two individuals from $\mathcal{P}_g$, compares them based on the notion of *constrained dominance*, and chooses a superior one as a parent.

The notion of constrained dominance is defined as follows. An individual i is said to *constrained-dominate* an individual j (denoted by $i \succ_C j$), if any of the following three conditions is hold:

- i is *feasible* ($v(i) = 0$; Equation 11), and j is not ($v(j) > 0$; Equation 11)[6].
- i and j are both feasible, and i *dominates* j.
- Both i and j are *infeasible*, but the total constraint violation of i is less than j's ($v(i) < v(j)$; Equation 11).

Given the notion of *dominance* [22], an individual i is said to dominate another individual j (denoted by $i \succ j$) if the both of the following conditions are hold.

- i's objective values are superior than, or equal to, j's in all objectives.
- i's objective values are superior than j's in at least one objectives.

With the crossover rate P_c, two parents reproduce two offspring with a single-point crossover operator (Lines 8 and 9). Then, mutation occurs on the two offspring (Lines 13 to 14). It assigns a new randomly-chosen IDC to each service in the offspring with the mutation rate P_m.

Algorithm 2 initialPopulationGeneration()

```
Require: $N$: The number of individuals in the population
Require: $\mathcal{D} = \{D_1, D_2, ..., D_M\}$: IDCs
Require: $\mathcal{C} = \{C_1, C_2, ..., C_M\}$: Each IDC's capacity constraint
Require: $\mathcal{S} = \{S_1, S_2, ..., S_K\}$: Services
Require: $\mathcal{L} = \{l_1, l_2, ..., l_K\}$: Each service's workload
 1: $\mathcal{P} = \emptyset$
 2: for each $C_j$ in $\mathcal{C}$ do
 3:     $c_j = C_j$
 4: end for
 5: $k = 0$
 6: while $k < N$ do
 7:     Create an individual $p_k$
 8:     for each $S_i$ in $\mathcal{S}$ do
 9:         Choose an IDC $D_j$ at random.
10:         if $c_j - l_i \geq 0$ then
11:             Assign $S_i$ to $D_j$
12:             $c_j = c_j - l_i$
13:         else
14:             $D_m = argmax_{D_n \in \mathcal{D}} c_n$
15:             Assign $S_i$ to $D_m$
16:             $c_n = c_n - l_i$
17:         end if
18:     end for
19:     $\mathcal{P} = \mathcal{P} \cup \{p_k\}$
20:     $k = k + 1$
21: end while
22: return $\mathcal{P}$
```

In Lines 15 and 16, local search is executed on the mutated offspring in order to improve their quality. Algorithm 3 shows this improvement process. It attempts to move each service to a new IDC with the local search probability P_l. It favors the improvement in renewable energy consumption (RE) than the other objectives. A service migration is performed if a service can yield better f_{RE} while it does not degrade the other objective values and constraint violation.

The binary tournament, crossover, mutation and local search operators are executed repeatedly on $\mathcal{P}_g$ to reproduce N offspring. The offspring ($\mathcal{O}_g$) are combined with the parent population $\mathcal{P}_g$ to form $\mathcal{R}_g$ (Line 19). This way, the proposed EMOA performs $(N + N)$ elitism.

The environmental selection process follows the reproduction process. N individuals are selected from $2N$ individuals in $\mathcal{R}_g$ as the next generation's population ($\mathcal{P}_{g+1}$). First, the individuals in $\mathcal{R}_g$ are ranked based on their constrained-dominance relationships. Non-dominated individuals are on the first rank. The i-th rank consists of the individuals dominated only by the individuals on the $(i-1)$-th rank. Ranked

[6] A feasible individual is an individual that violates none of given constraints (Equation 10). An infeasible individual is an individual that violates at least one of given constraints.

individuals are stored in $\mathcal{F}$ (Line 20). $\mathcal{F}_i$ contains the i-th rank individuals.

Then, the individuals in $\mathcal{F}$ move to $\mathcal{P}_{g+1}$ on a rank by rank basis, starting with $\mathcal{F}_1$ (Lines 23 to 26). If the number of individuals in $\mathcal{P}_{g+1} \cup \mathcal{F}_i$ is less than N, $\mathcal{F}_i$ moves to $\mathcal{P}_{g+1}$. Otherwise, a subset of $\mathcal{F}_i$ moves to $\mathcal{P}_{g+1}$. The subset is selected based on the crowding distance metric, which measures the distribution (or diversity) of individuals in the objective space [5] (Lines 27 and 28). The metric computes the distance between two closest neighbors of an individual in each objective and sums up the distances associated with all objectives. A higher crowding distance means that an individual in question is more distant from its neighboring individuals in the objective space. In Line 27, the individuals in $\mathcal{F}_i$ are sorted from the one with the highest crowding distance to the one with the lowest crowding distance. The individuals with higher crowding distance measures have higher chances to be selected to $\mathcal{P}_{g+1}$ (Line 28).

Algorithm 3 localSearch()

```
Require: $o$: An individual to be applied for a local search
Require: $P_l$: Local search probability
Require: $\mathcal{D} = \{D_1, D_2, ..., D_M\}$: IDCs
for each service $S_i$ in $o$ do
    $f_{RE} = f_{RE}(o)$
    $f_{CE} = f_{CE}(o)$
    $f_{USD} = f_{USD}(o)$
    $v = v(o)$
    if random() $\leq P_l$ then
        for each $D_j$ in $\mathcal{D}$ do
            Assign $S_i$ to $D_j$.
            $f'_{RE} = f_{RE}(o)$
            $f'_{CE} = f_{CE}(o)$
            $f'_{USD} = f_{USD}(o)$
            $v' = v(o)$
            if $f'_{RE} > f_{RE}$ and $f'_{CE} \leq f_{CE}$ and $f'_{USD} \leq f_{USD}$
            and $v' \leq v$ then
                $f_{RE} = f'_{RE}$
                $f_{RE} = f'_{CE}$
                $f_{USD} = f'_{USD}$
            else
                Cancel the assignment of $S_i$ to $D_j$.
            end if
        end for
    end if
end for
return $o$
```

4. SIMULATION EVALUATION

This section describes a series of simulation results to evaluate Green Monster.

4.1 IDC Configurations

This evaluation study simulates IDCs located in nine major European countries: Denmark, Germany, Greece, Ireland, Italy, Netherlands, Spain, UK and Portugal (Figure 1). These countries are chosen in an attempt to give a significant variation in both climates and sources of renewable energy used. For the temperature variations in each IDC's host country, this evaluation study uses the data from the European Climate Assessment & Dataset project, which records real temperature data in Europe[7].

Figure 4 shows the total renewable energy production in each IDC host country from January 2007 to December 2009.

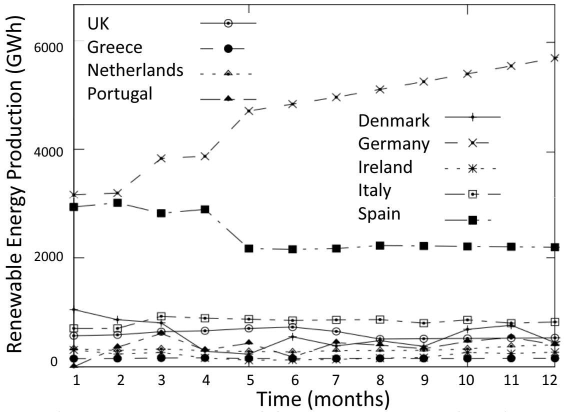


Figure 4: Renewable Energy Production

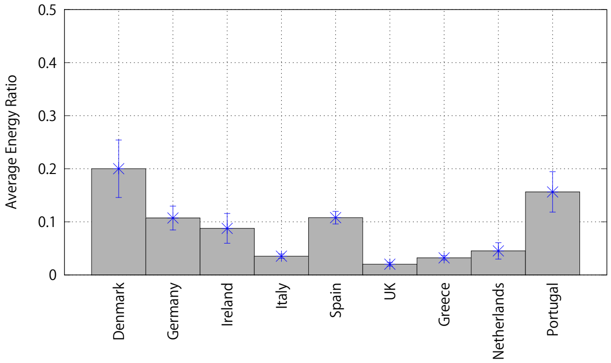


Figure 5: Renewable Energy Ratio

Figure 5 depicts the average renewable energy ratio in each IDC host country (R_j in Equation 1) during the same period. Figures 4 and 5 are produced with the data available from the International Energy Agency (IEA)[8].

Simulated IDCs are connected in a network topology in line with the European Optical Network (Figure 1). Three types of services are deployed on IDCs: data, voice and video services (Table 1). Each IDC operates a varying number of servers (8–200) and services (16–400), proportionate to the population of its host country. All IDCs operate a standardized model of servers: $P_{max} = 400$ W and $P_{idle} = 150$ W (Table 1).

Figure 6 shows the daily service request rates placed on different IDCs. The average total rate is two million requests per day. The dynamic changes in the request rates are configured by adapting the traffic trace in Akamai's IDCs [17][9]. The rates are configured across IDCs in proportion to the populations of their host countries. In each IDC, requests are evenly distributed to all deployed services.

4.2 EMOA Configurations

The proposed EMOA is configured with a set of parameters shown in Table 2. It runs bi-weekly for 12 simulated months. After a bi-weekly run, one of non-dominated indi-

[7] http://eca.knmi.nl

[8] http://www.iea.org/stats/surveys/elec_archives.asp

[9] In order to represent long-term fluctuations in request rates, this evaluation study adds a number of randomly distributed surges and falls on Akamai's short-term trace data.

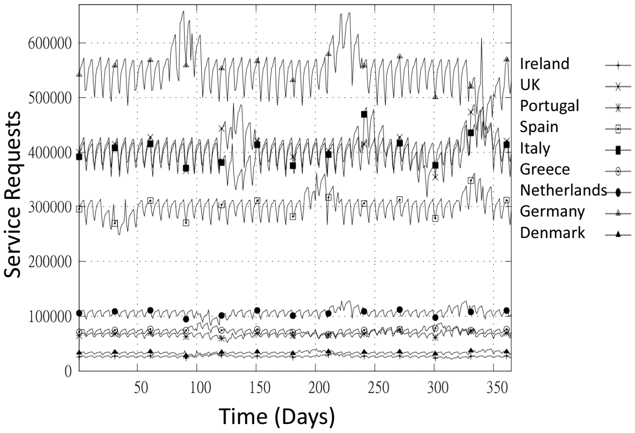


Figure 6: Daily Service Request Rates

Table 1: IDC Configurations

Parameter	Value
# of IDCs (M in Section 3.1)	9
Total # of servers in IDCs	878
# of service types	3
Total # of services (K in Section 3.1)	1756
P_{max} (Equation 5)	400 W
P_{idle} (Equation 5)	150 W
Per-request CPU utilization for data services (u_i in Equation 2 and 9)	[0.001, 0.01]
Per-request CPU utilization for voice services (u_i in Equation 2 and 9)	[0.011, 0.024]
Per-request CPU utilization for video services (u_i in Equation 2 and 9)	[0.025, 0.039]
Per-request data transmission volume for data services (w_i in Equation 9)	[0.01, 0.05]
Per-request data transmission volume for voice services (w_i in Equation 9)	[0.06, 0.15]
Per-request data transmission volume for video services (w_i in Equation 9)	[0.16, 0.25]
Free-cooling efficiency (α in Equation 7)	1.4

viduals is chosen as a simulated decision of an IDC operator in order to perform dynamic service migration and placement. The individual choice is based on the hypervolume metric [27]. (The individual with the highest hypervolume is chosen.) The metric measures the volume of a hypercube that an individual dominates in the objective space. It tends to favor *balanced* individuals that equally balance the trade-offs among all objectives rather than *extreme* individuals that yield superior performance only in a limited number of objectives.

In order to evaluate Green Monster, it is compared with the following two benchmark algorithms:

- Static placement: Randomly-selected two services are placed on each server at the beginning of a simulation. They do not dynamically migrate during a simulation.
- Random placement: Services dynamically migrate by executing Algorithm 2 bi-weekly.

4.3 Simulation Results

Figure 7 shows how individuals increase the union of the hypervolumes that they dominate in the objective space as the number of generations grows. The hypervolume measure rapidly increases in the first 30 generations and converges around the 70th generation. At the last generation, all individuals are non-dominated in the population. This indicates that the proposed EMOA allows individuals to efficiently evolve and collectively improve their quality (i.e., objective values) within 100 generations (g_{max} in Algorithm 1).

Table 2: EMOA Configurations

Parameter	Value
# of generations (g_{max} in Algorithm 1)	100
Population size (N in Algorithm 1)	100
Crossover rate (P_c in Algorithm 1)	0.9
Mutation rate (P_m in Algorithm 1)	0.1
Local search rate (P_l in Algorithm 3)	0.1
Interval between the proposed EMOA's runs	2 weeks

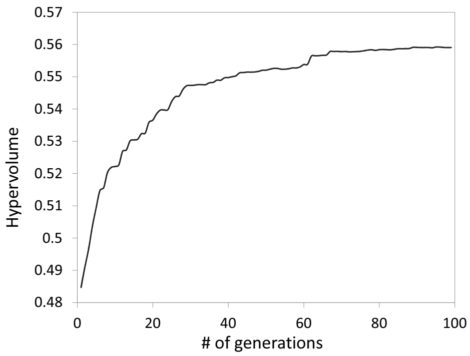
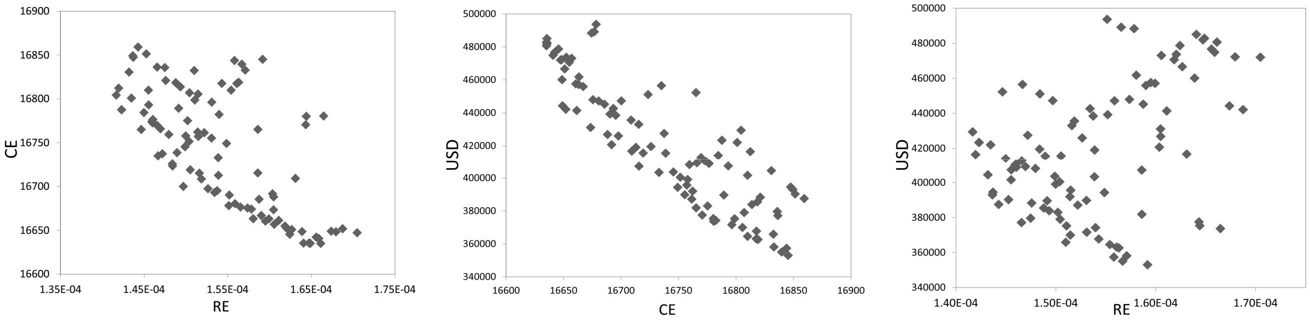


Figure 7: Hypervolume Convergence

Figures 8(a), 8(b) and 8(c) show the two dimensional objective spaces that plot non-dominated individuals obtained at the last generation (RE-CE, CE-USD and RE-USD, respectively). Note that the RE axis (X axis) indicates $1/f_{RE}$ in Figures 8(a) and 8(c). Figure 8 demonstrates that RE, CE and USD conflict with each other and the proposed EMOA successfully reveals the trade-offs among them. For example, in the RE-CE objective space (Figure 8(a)), the proposed EMOA finds the individuals around the left top corner (i.e., the ones yielding high f_{RE} values and high f_{CE} values), the individuals around the right bottom corner (i.e., the ones yielding low f_{RE} values and low f_{CE} values) and the individuals between the two corners (i.e., the ones yielding intermediate f_{RE} and f_{CE} values). As discussed in Section 1, the proposed EMOA allows IDC operators to make well-informed decisions for service migration and placement by providing them a diverse set of approximated Pareto-optimal solutions.

Figures 9(a), 9(b) and 9(b) show how Green Monster and two benchmark algorithms yield the RE, CE and USD values, respectively, throughout a simulated year. As depicted in Figure 9(a), Green Monster consumes a significantly higher amount of renewable energy than two benchmark algorithms. Table 3 shows the daily average of each objective value. On average, Green Monster yields 35.1% higher renewable energy consumption per day than the static placement algorithm. This difference accounts for 114 MWh per day. Similarly, Green Monsters yields 33.9% higher renewable energy consumption per day than the random placement algorithm. This difference accounts for 111 MWh per day. Figure 9(a) illustrates that Green Monster successfully migrates services to the IDCs with higher renewable energy ratios while considering the other two objectives.



(a) RE-CE Objective Space (b) CE-USD Objective Space (c) RE-USD Objective Space

Figure 8: Non-dominated Individuals in Two Dimensional Objective Spaces

Figure 9(b) shows that Green Monster saves more cooling energy than two benchmark algorithms in summer. This indicates that it successfully migrates services to the IDCs with lower outdoor temperatures and hence higher COP values. (Note that cooling energy consumption is proportionate to the COP and workload in an IDC. See Section 3.1.) On the contrary, Green Monster and two benchmark algorithms yield similar cooling energy consumption during winter because outdoor temperatures and COP values are similar across IDCs. As shown in Table 3, Green Monster saves 3.2 % more cooling energy than the random placement algorithm on a daily average basis.

Figure 9(c) depicts that Green Monster consistently outperforms the random placement algorithm in the user-to-service distance objective. On a daily average basis, Green Monster yields a 23.4% shorter USD than the random placement algorithm (Table 3). This implies that the response time of services is significantly lower in Green Monster. Note that USD is always zero in the static placement algorithm because it does not dynamically migrate services.

In summary, Figure 9 demonstrates that Green Monster successfully balances the trade-offs among objectives and yields superior performance than two benchmark algorithms.

Table 3: Daily-averaged Objective Values

	RE	CE	USD
Static	324.5	949.58	**0**
Random	327.3	947.34	487378
Green Monster	**438.5**	**919.02**	373172

5. CONCLUSIONS

The proposed framework, Green Monster, is designed to dynamically move services across IDCs for reducing their carbon footprint while maintaining their performance. Simulation results verify this and demonstrates that Green Monster outperforms conventional capacity-based service placement algorithms with respect to conflicting objectives.

6. ACKNOWLEDGMENT

This work is supported in part by Science Foundation Ireland via the “A Biologically inspired framework supporting network management for the Future Internet” starting investigator award (grant no. 09/SIRG/I1643).

References

[1] C. Bash and G. Forman. Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center. In *Proc. USENIX Annual Technical Conference*, 2007.

[2] D. J. Bradley, R. E. Harper, and S. W. Hunter. Workload-based power management for parallel computer systems. *IBM J. Res. Dev.*, 47(5-6), 2003.

[3] R. Das, J. O. Kephart, C. Lefurgy, G. Tesauro, D. W. Levine, and H. Chan. Autonomic multi-agent management of power and performance in data centers. In *Proc. Int'l Conf. of Autonomic Agents and Multiagent Syst.*, 2008.

[4] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proc. Int'l Conf. on Parallel Problem Solving from Nature*, 2000.

[5] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Proc. Int'l Conf. on Parallel Problem Solving from Nature*, 2001.

[6] T. V. T. Duy, Y. Sato, and Y. Inoguchi. Performance evaluation of a green scheduling algorithm for energy savings in cloud computing. In *Proc. Int'l Parallel and Distributed Processing Symposium*, 2010.

[7] European Commission. Code of conduct on data centres energy efficiency, version 1.0. 2008.

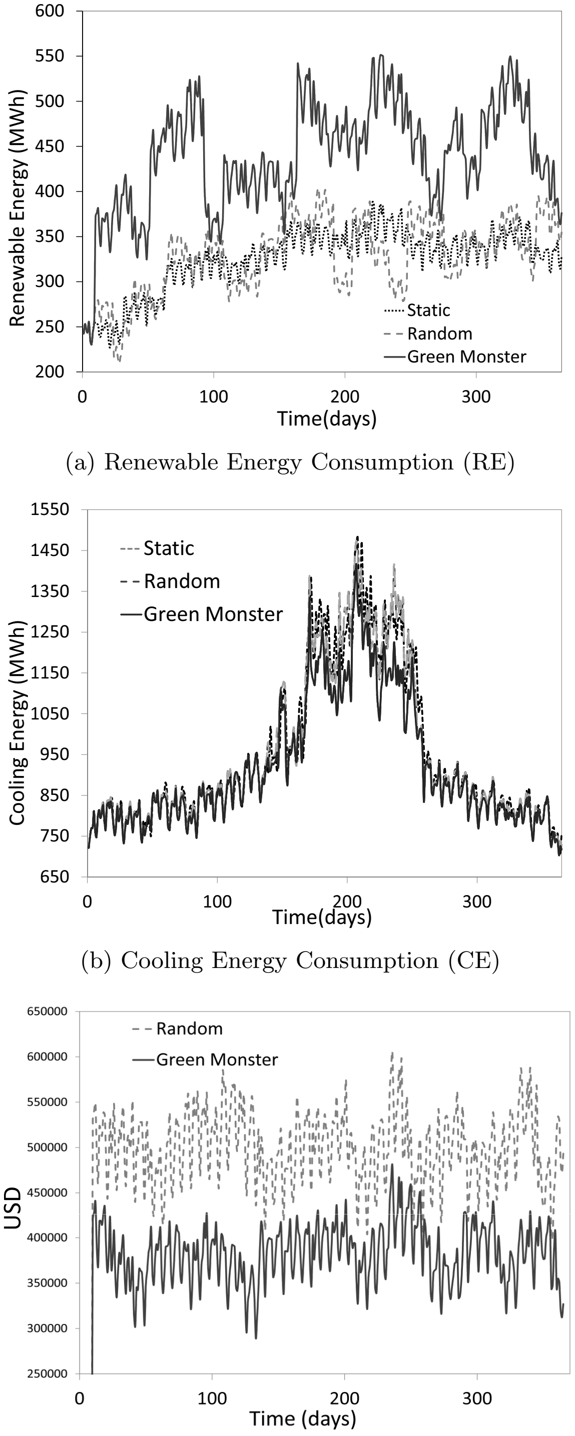
[8] S. K. Garg, C. S. Yeo, A. Anandasivam, and R. Buyya. Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers. *J. Parallel and Dist. Comput.*, 71(6), 2010.

[9] Global Action Plan. Green ICT handbook: A guide to green ICT. 2009.

[10] K. H. Kim, A. Beloglazov, and R. Buyya. Power-aware provisioning of virtual machines for real-time cloud services. *Concurrency Computat.: Pract. Exper.*, 23(13), 2011.

[11] J. Koomey. Growth in data center electricity use 2005 to 2010. *Analytics Press*, 2011.

[12] C. Malone and C. Belady. Metrics to characterize data center and it equipment energy use. In *Proc. Digital Power Forum*, 2006.



(a) Renewable Energy Consumption (RE)

(b) Cooling Energy Consumption (CE)

(c) User-to-Service Distance (USD)

Figure 9: Objective Values

[13] D. Meisner, B. T. Gold, and T. F. Wenisch. Powernap: Eliminating server idle power. In *Proc. Int'l Conf. on Architectural Support for Progr. Lang. and Operating Syst.*, 2009.

[14] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making scheduling "cool": temperature-aware workload placement in data centers. In *Proc. USENIX Annual Technical Conference*, 2005.

[15] M. J. Moran and H. N. Shapiro. Fundamentals of engineering thermodynamics. *Wiley*, 1995.

[16] C. Patel, R. Sharma, C. Bash, and S. Graupner. Energy aware grid: Global workload placement based on energy efficiency. In *Proc. Int'l Mechanical Engineering Congress and Exposition*, 2003.

[17] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs. Cutting the electric bill for internet-scale systems. In *Proc. ACM SIGCOMM Conference*, 2009.

[18] Renewable Energy Policy Network for the 21st Century (REN21). Renewables 2011: Global status report. 2011.

[19] C. Rusu, A. Ferreira, C. Scordino, and A. Watson. Energy-effcient real-time heterogeneous server clusters. In *Proc. IEEE Real-Time and Embedded Technology and Applications Symposium*, 2006.

[20] K. G. Saurabh, S. Y. Chee, and R. Buyya. Green cloud framework for improving carbon efficiency of clouds. In *Proc. Int'l Conf. on Parallel Proc.*, 2011.

[21] R. Sharma, C. Bash, C. Patel, R. Friedrich, and J. Chase. Balance of power: Dynamic thermal management for Internet data centers. *IEEE Internet Comput.*, 9(1), 2005.

[22] N. Srinivas and K. Deb. Multiobjective function optimization using nondominated sorting genetic algorithms. *Evol. Computat.*, 2(3), 1995.

[23] A.-A. Tantar, E. Tantar, and P. Bouvry. Load balancing for sustainable ICT. In *Proc. ACM Genetic and Evolutionary Computation Conference*, 2011.

[24] The. U.S. Environmental Protection Agency (EPA). Report to congress on server and data center energy efficiency public law 109-431. 2007.

[25] L. Wang, A. J. Younge, T. R. Furlani, G. von Laszewski, J. Dayal, and X. He. Towards thermal aware workload scheduling in a data center. In *Proc. Int'l S. on Pervasive Syst., Algorithms and Netw.*, 2009.

[26] Y. Zeratul and T. Maolin. Penalty-based genetic algorithm for the composite SaaS placement problem in the cloud. In *Proc. IEEE Congress on Evol. Computat.*, 2010.

[27] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms: A comparative study. In *Proc. Int'l Conf. on Parallel Problem Solving from Nature*, 1998.