2012

Energy-aware replica selection for data-intensive services in cloud

Bo Li

Energy-Aware Replica Selection for Data-Intensive Services in Cloud

APPROVED BY

SUPERVISING COMMITTEE:

Dr. Ivona Bezáková, Supervisor

Prof. Alan Kaminsky, Reader

Dr. Hans-Peter Bischof, Observer

Energy-Aware Replica Selection for Data-Intensive Services in Cloud

by

Bo Li, B.E.

THESIS

Presented to the Faculty of the Golisano College of Computer and Information Sciences
Rochester Institute of Technology
in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

Rochester Institute of Technology

May 2012

Acknowledgments

I would like to appreciate the work of my supervisor Dr. Ivona Bezáková. Her support to my master's thesis and research, her patience, and her encouragement help me going through the research of my master degree and completing the master's thesis.

I also want to thank Prof. Alan Kaminsky for his comments to my thesis. These professional suggestions have helped me to improve the work in my thesis.

My sincere gratitude goes to Prof. Kirk W. Cameron. The experiment platform on SystemG is one of the most important parts in my thesis work. Without his help and support, I cannot even accomplish the experiment work in my thesis.

Also, I want to express my appreciation to my colleague Shuaiwen Song. His help on SystemG and review to my thesis make me finish it efficiently.

At last, I want to thank to all the people who give me comments and encouragement on my research. Without these, the graduate life would be hard and tedious.

Abstract

Energy-Aware Replica Selection for Data-Intensive Services in Cloud

Bo Li, M.S.
Rochester Institute of Technology, 2012

Supervisor: Dr. Ivona Bezáková

With the increasing energy cost in data centers, an energy efficient approach to provide data intensive services in the cloud is highly in demand. This thesis solves the energy cost reduction problem of data centers by formulating an energy-aware replica selection problem in order to guide the distribution of workload among data centers. The current popular centralized replica selection approaches address such problems but they lack scalability and are vulnerable to a crash of the central coordinator. Also, they do not take total data center energy cost as the primary optimization target. We propose a simple decentralized replica selection system implemented with two distributed optimization algorithms (consensus-based distributed projected subgradient method and Lagrangian dual decomposition method) to work with clients as a decentralized coordinator. We also compare our energy-aware replica selection

approach with the replica selection where a round-robin algorithm is implemented. A prototype of the decentralized replica selection system is designed and developed to collect energy consumption information of data centers. The results show that in the best case scenario of our experiments, the total energy cost using the Lagrangian dual decomposition method is 17.8% less than a baseline round-robin method and 15.3% less than consensus-based distributed projected subgradient method. Also, the prototype is proved to be working efficiently with low computation and communication overhead. The proposed decentralized energy-aware replica selection system can also be easily adapted to the real world cloud environment.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the cloud, some services are replicated in geographically different data centers. If the clients want to use such services, they need to specify one or more data centers to connect with. The problem of how to choose from those data centers is called replica selection. The way in which replicas are selected for client requests is important to both clients and cloud service providers. It is because by choosing the right replicas, clients can achieve optimal experiences, such as minimal latency, the least packet loss, or maximal available bandwidth. Also, replica selection is valuable to the service providers because it can be utilized to balance the load between different replicas, as well as to minimize the operating cost.

According to the EPA report in 2007 [21], data centers consumed 61 billion kWh which is 1.5% of the total U.S. electricity in 2006. Since energy consumption has comprised a significant part of the costs in data centers, energy cost reduction becomes an essential way to reduce and minimize the operating cost of data centers. The electricity price is one of the factors affecting this cost in data center. It varies with different locations and different times in a day. For example, the electricity price in the U.S. may cost about

11.2 ¢/kWh, while that of another location in South Africa may cost 5.37 ¢/kWh. Also, the time zones result in the situation that one data center is at its load peak hours while another is at its off-peak hours, which also causes the electricity price to vary between such data centers. Therefore, an energy-aware replica selection system considering real time electricity price is highly in demand by cloud service providers to reduce the total energy cost.

In terms of system architecture, a replica selection system can be designed as either a centralized coordinator or several distributed agents working as a decentralized coordinator between clients and replicas. Centralized architecture, such as MapReduce [8], implements a centralized coordinator to handle and distribute all the tasks. Such architecture performs well when not many client requests need to be handled. However, the flaw of its scalability causes a bottleneck in handling a large amount of client requests. Also it is not reliable because once the centralized coordinator crashes, the replica selection system would stop working. Decentralized coordinator performs much better in terms of scalability and reliability. The burden from clients can be split into different agents in the replica selection system. However, previous works [23] on designing decentralized replica selection systems have not considered total system energy cost as the primary optimization objective. Therefore, a new decentralized replica selection system considering both energy cost and bandwidth capacity to achieve the most energy efficient load distribution solution for data centers needs to be designed. The prototype of such decentralized replica selection system can also be easily adapted to the real cloud environ-

ment.

In this work, we propose an energy-aware decentralized replica selection system for data-intensive applications in the cloud. It can reduce the total energy cost by distributing the data-intensive workload among all the data centers in the system. An energy consumption model for data centers is built to indicate the relationship between workload and energy consumption for data-intensive applications. The decentralized architecture requires each data center working as a distributed agent to cooperate with other data centers. In particular, we adapt the Lagrangian dual decomposition method (LDDM) to our replica selection system to solve this global optimization problem in parallel. The algorithm is developed and implemented into our runtime system. We compare both performance and total energy cost of our approach with those of consensus-based distributed projected subgradient method (CDPSM) [16] and baseline round-robin replica selection method. The results show that in the best case scenario of our experiments, our proposed replica selection system implemented with LDDM can reduce 17.8% total energy cost than that of round-robin method, while using CDPSM can only reduce 2.5%. The results also show that LDDM has better performance than CDPSM in terms of lower system complexity and faster convergence rate for solving global energy optimization problem.

The rest of this thesis is organized as follows: Section II is a review of the related work. Section III is the methodology of solving the energy-aware replica selection problem. It includes the energy consumption model in

data center, system architecture, the formulated optimization problem, and algorithms we adapt to solve the global energy cost reduction problem in parallel. Section IV evaluates the performance as well as the energy cost reduction of our decentralized replica selection system. Section V concludes and describes future work.

Chapter 2

Background

2.1 Energy Efficiency in a Data Center

Energy efficiency in the cloud has been studied as an important issue by others. The effort of reducing energy cost has been taken through hardware, software, as well as networking aspects [4]. For example, resource allocation [22] and scheduling algorithms considering QoS [5, 3] can improve energy efficiency of data center as well as guarantee the quality of services. Such work solves the problem of energy efficiency in the data center in the situation that resources can be allocated in a data center or among data centers. However, such work does not investigate the workload distribution of client requests among all the data centers, which can also affect the total energy cost even if the resources have already been optimally allocated. In particular, for data-intensive services such as online video sharing or distributed file systems, the distribution of workload to each data center can significantly affect the energy cost. Also, compared with latency in QoS criteria, bandwidth capacity of the data center is more important for such services. Zong et al. [25] apply a buffer-disk to schedule storage system tasks, so that energy consumption can be reduced by keeping a large number of idle data disks sleeping long enough. But they only consider the relationship between disk state and power

consumption regardless of different application types for the disk. Given the fact that energy efficiency serves for the goal of reducing cost of energy consumption in data centers, electricity price and bandwidth capacity are also core factors needed to be considered by the service providers. Rao et al. [17] involve multiple electricity markets into their model aiming at minimizing the total electricity cost. This work proves that our work in this thesis toward data-intensive services is necessary given that data centers have time and location diversity. However, they do not consider the bandwidth capacity which is the primary constraint for data-intensive applications. In order to minimize energy cost of data centers, Liu et al. [13] take workload and number of active servers in each data center into consideration. However, they assume that the single server energy consumption does not depend on the load, which is not practical for servers in the real world data centers.

The ultimate goal of energy management is to make energy consumption proportional to load [6]. For data-intensive applications in the cloud, the traffic load has been proved to be able to affect the energy consumption of the data center. In the work of Smith and Sommerville [20], the different types of applications can result in an energy consumption increment in different subcomponents within a server, such as memory, CPU, disk, etc. Furthermore, a linear relationship between data-intensive workload and energy consumption for hard disk in server systems is validated in [12]. Since data-intensive applications in the cloud are mainly disk intensive (e.g. online video streaming and sharing), we assume there is a linear relationship between the workload

and the energy consumption of a server in the cloud. However, we cannot make the assumption that a linear relationship exists between workload and energy consumption in the other components of data centers. For instance, the majority of network devices, comparing with a single server, are far from being energy proportional [14, 19].

In our work, we take the real time electricity price into consideration to calculate the total cost of energy consumption in the data centers. We propose an energy model with the assumption that the energy consumption comes from two major parts of the data center: servers and network devices. We use this model to minimize the total energy cost of the data centers.

2.2 Replica Selection in Cloud

In order to improve the service performance and quality, service providers in the cloud distribute data centers in geographically different locations. This can bring the users high speed access to the resources as well as improve the reliability of cloud services. However, redundant resources and infrastructure also lead to huge cost of the data centers [9]. Replica selection, which is described as the way of selecting data centers for a specific service, can help both users and service providers to maximize the utilization of data centers. It can improve the data center efficiency and lead to more benefits for the service providers.

The work of Ruiz-Alvarez and Humphrey [18] presents an approach for selecting storage services in the cloud. However, it relies on a storage system

description which is machine readable regarding its performance and other features. In fact, if the cloud service comes from different cloud providers, some credential information cannot be shared with others. So, this approach is hard to use in solving replica selection problems across multi-data centers.

Replica selection can be developed as a system or service in the cloud. The above approach can only be implemented as a centralized coordinator. Similarly, the work of Le et al. [11] optimizes both energy and cost of data centers but it can only be designed in a centralized architecture. The limitations of centralized architecture in scalability and reliability motivate us to choose the decentralized architecture solution as an alternative. Dabek et al. present a decentralized network coordinate system to predict the network latency in order to help users to select from hosts [7]. In the work of Wendell et al. [23], a decentralized replica selection system is proposed to direct clients' requests to the data centers aiming at minimizing network cost. In my thesis, we consider not only the bandwidth capacity but also the energy cost of data-intensive applications when formulating the replica selection problem.

In our work, replica selection problem is formulated as nonlinear convex optimization problem. The decentralized architecture requires us to solve the global optimization problem in parallel. Liu et al. present Gauss-Seidel method and gradient projection method to solve the global optimization problem in a distributed manner [13]. Wendell et al [23] propose a Lagrangian dual decomposition method in a decentralized system to solve the global optimization problem of network cost performance. A consensus-based distributed

projected subgradient method [16] can also be used to solve the global optimization problem by the collaboration of distributed agents. In my thesis, we integrate the Lagrangian dual decomposition method and distributed projected subgradient method into our replica selection system. We also compare the performance of these two methods working in the system.

Chapter 3

Methodology

In this section, we first present an energy consumption model for data centers. Based on this model, we then formulate the replica selection problem as a convex optimization problem which minimizes total energy cost of data centers subject to the bandwidth capacity of each data center. After that, we propose a simple decentralized architecture where the distributed nodes cooperate with each other to solve the global optimization problem in parallel. Finally, we adapt two algorithms for solving optimization problems in parallel into our problem to work under the decentralized replica system architecture. Some important notations in this section are summarized in Table 3.1. They are mapped to the system architecture in Fig. 3.1.

3.1 Energy Consumption Model for Data Center

In order to minimize the energy cost of data-intensive applications in the cloud, we need to build a correlation between energy consumption and the requests from clients. We consider the energy consumption in the data center coming from two major parts: the server nodes, and the network infrastructure. Since data-intensive applications in the cloud are mainly disk intensive

Table 3.1: Notations

C	Set of all clients
N	Set of replicas
p_{cn}	Traffic load mapped from client c to replica n
P_n	Constraint sets on replica n
B_n	Bandwidth capacity on replica n
R_c	Traffic load of the request from client c
u_n	Unit price (¢) of power in replica n
a_n	Weight value of replica n in consensus-based algorithm
α_n, β_n	Weight scalars for the energy consumption of computer devices and network devices in replica n

(e.g. video streaming and sharing), we make the assumption that there is a linear relationship between the workload and the energy consumption of a server in the cloud, which is also desired in a energy efficient data center [2] environment. The relationship between the energy consumption of the network infrastructure (such as routers and switches) and workload is determined by the technologies used in designing the hardware. For the equipment which use Dynamic Voltage Scaling (DVS) [10] as the energy reduction approach at integrated circuit level, the relationship between traffic load and energy consumption can be modeled as cubic. For instance, Ethernet interface cards applying DVS and DFS (Dynamic Frequency Scaling), have been proved to support this relationship [24]. Therefore, we can get a weighted combination of linear (for servers) and cubic (for network devices) relationship between energy consumption and network traffic load in our model. The total energy

consumption cost of all the replicas can be modeled as:

$$E^g = \sum_n u_n \cdot (\alpha_n \sum_c p_{cn} + \beta_n (\sum_c p_{cn})^3) \tag{3.1}$$

where α and β are weight scalars. The goal of our problem is to minimize E^g for the clients' requests to the data centers. The global optimization problem can be formulated as:

$$\begin{aligned} \underset{p_{cn}}{\text{minimize}} \quad & E^g = \sum_n E_n \\ \text{subject to} \quad & f_n(P) = \sum_c p_{cn} - B_n \leq 0, \forall n \in N \\ & h_c(P) = \sum_n p_{cn} - R_c = 0, \forall c \in C \end{aligned} \tag{3.2}$$

where $E_n = u_n \cdot (\alpha_n \sum_c p_{cn} + \beta_n (\sum_c p_{cn})^3)$ is the energy consumption cost in replica n, $f_n(P)$ is the bandwidth capacity constraint of replica n, $h_c(P)$ is the request constraint of client c. The problem turns out to be a cubic objective function with several linear equality and inequality constraints.

3.2 Decentralized Replica Selection System

The decentralized replica selection system is built on the infrastructure of data centers without any additional devices. The architecture is shown in Fig. 3.1. In the system, each replica keeps listening to the clients' requests. Once the requests to the replica selection system are received, these replicas will start to cooperate with each other to solve the global optimization problem.

The decentralized architecture of the replica selection system makes it capable of accepting more clients' requests than centralized architecture

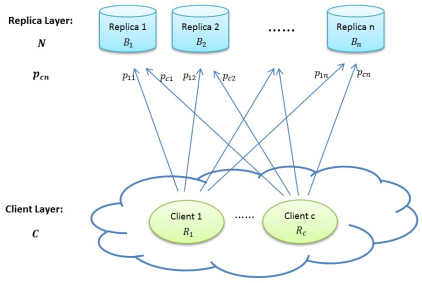

Figure 3.1: Distributed services in replicas

because each replica works individually to handle clients. Also the replica selection service is transparent to the clients, which means the client does not need to know which replica it is communicating with since each replica is doing exactly the same work in the system. Another advantage of decentralized architecture is that the system is more reliable. If the replica selection work is assigned to a single node, the crash of this node can cause the failure of the whole replica selection system. It is unlikely to happen in a decentralized replica selection system unless all the replicas malfunction. However, the decentralized solution is not perfect because the communication and computation overhead can decrease the efficiency of the replica selection system. In terms of energy consumption, the less efficient system consumes the more energy. Therefore, high performance of the selected distributed algorithms is highly desired for reducing the total energy cost.

3.3 Replica Selection System Design

The replica selection system involves the client side and the server side. The programs of both sides are designed as multiple thread programs using TCP/IP socket to communicate. The structure of the server side program is illustrated in Fig. 3.2. The ClientListener thread keeps listening to the

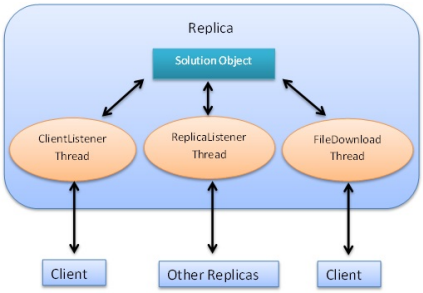

Figure 3.2: Server side components diagram

new requests from clients. The ReplicaListener thread keeps listening to the requests of solution information from other replicas. The FileDownload thread handles the sending of requested files to the clients. Once a new client request comes, it communicates with the ClientListener thread first and then waits for the solution of how to distribute its requested load. After the solution is achieved, client side will create new threads to communicate with all the replicas at the same time to download the computed amount of load.

The efficiency of the system can affect the energy cost of data centers. In order to achieve a highly efficient system, we implement such multi-threaded mechanism to handle as many client requests as possible. The decentralized replica selection system can accept more than one request at a time. After

getting the optimal request distribution, the FileDownload thread starts to send data to the clients while the decentralized replica selection system is ready to accepting new requests from the clients.

Reliability of the decentralized replica selection system can be guaranteed by the time-out mechanism in each replica. The ReplicaListener thread is used to communicate between replicas. Once a replica malfunctions, the other replicas can know and then remove this dead replica from their lists through the ReplicaListener thread.

3.4 Decentralized Replica Selection Algorithms

In order to design an efficient decentralized replica selection system, we are going to investigate the algorithms we select to solve the global optimization problem in parallel. Solving an optimization problem in a distributed environment is not as easy as on a single machine. Not much work has been done in solving this type of optimization problem with constraints [16]. The consensus-based distributed projected subgradient method, as a variant of gradient-based method, is a good candidate for solving such problem. Theoretically, it has been proven to be successful to solve the problem when the global objective function is a sum of some local objective functions. It provides a practical approach to solve the decentralized optimization problem. Another option for solving constrained optimization problems is the Lagrangian dual decomposition method. It can combine the objective function with the constraints by using the Lagrangian multiplier. Then a possible distributed

solution may be applicable to the dual problem. In my thesis, we implement both algorithms and then compare their system complexity and convergence speed.

3.4.1 Consensus-based Distributed Projected Subgradient Method (CDPSM)

This method is originally proposed to solve constrained optimization problems in multi-agents networks. In my thesis, we adapt this method to our decentralized replica selection system. The objective function E^g in our replica selection problem is the sum of functions which are local objective functions for replicas in the form of $E^g = \sum_n E_n$. Each replica works on solving its own local optimization problem E_n which is subject to the local constraints $p_{cn} \in P_n$, where P_n is a subset of the constraint sets that have local variables of replica n . The optimization problem in replica n can be formulated as:

$$\begin{aligned} \underset{p_{cn}}{\text{minimize}} \quad & E_n \\ \text{subject to} \quad & p^n_{cn} \in P_n \end{aligned}$$

The main idea of this algorithm is to use a consensus mechanism among distributed replicas to split the computation work. Each distributed replica keeps working on solving a subproblem of the global problem. The consensus mechanism can combine solution of subproblems to form the global optimization solution. Given p_{cn} is the solution to the global optimization problem, each replica n starts by estimating $\{p_{cn} \mid c \in C\, n \in N\}^n \in P_n$ and updating its solution p_{cn} iteratively by cooperating with other replicas. The consensus and

projection procedure for iteratively estimating can be denoted by the following equation:

$$p_{cn}^{n}(k+1) = Proj_{P_n}[\sum_{j=1}^{N} a_n^j \cdot p_{cn}^j(k) - d_k \cdot g_n(k)]^+ \tag{3.3}$$

where a_n^j are the weights of all the replicas, $d_k > 0$ is the step size, and $g_n(k)$ is the subgradient on its local objective function E_n. Since the objective function of our problem is twice differentiable, we could use gradient instead of subgradient as $g_n(k)$. The symbol $Proj_{P_n}[\cdot]^+$ denotes the operation of projection. We have:

$$Proj_{P_n}[p_{cn}^*]^+ = arg \min_{p_{cn} \in P_n} \|p_{cn}^* - p_{cn}\|$$

By projecting the solution p_{cn} back into its own local constraint set P_n, the algorithm guarantees that in each iteration the solution is feasible.

Based on this method, every replica in our system keeps running to handle client requests and the consensus mechanism. We can present the algorithm for each replica as follows:

Algorithm 1 Algorithm of CDPSM

1: **Initialization:** Set the unit price of replica i.
2: **repeat**
3: Collect the clients' requests from clients.
4: Collect the solution p_{cn} from other replicas.
5: Get the consensus solution $V_{cn} = \sum_n a_n p_{cn}^i$, where $\sum_n a_n = 1$
6: Update solution by $p_{cn} = V_{cn} - d \cdot g(V_{cn})$, where d is step size and $g(V_{cn})$ is gradient value of function E_n at V_{cn}.
7: Project p_{cn} to the constraint sets following the project rule $P_{X_n}[p_{cn}^*]^+$
8: **until** p_{cn} do not change.

As stated before, the efficiency of the replica selection system can af-

fect the energy consumption because the energy consumption for the solution computation period can be significant. Even though the reliability of the replica selection system implementing such algorithm is much better than a centralized architecture, the existence of both local and global constraints can increase the complexity of our system. The size of solution p_{cn} in each replica is $O(|C|\cdot|N|)$. The consensus mechanism requires distributed replicas to request the solutions from other replicas. So the communication complexity of each iteration is of size $O(|C|\cdot|N|\cdot|N-1|\cdot|N|)$ which is approximately $O(|C|\cdot|N|^3)$, where C is the number of clients and N is the number of replicas.

3.4.2 Lagrangian Dual Decomposition Method (LDDM)

Since there are dependencies in the global variables among replicas, we need to decouple them in order to solve the problem in parallel. LDDM provides us with a way to solve such problem. Given the original problem (3.2), we can formulate the Lagrangian dual problem from the global optimization problem as:

$$\begin{aligned} \underset{p_{cn}}{\text{minimize}} \quad & L(p_{cn}, \mu) = \sum_{n=1}^{N} E_n + \sum_{c=1}^{C} \mu_i \cdot h_c(P) \\ \text{subject to} \quad & f_n(P) = \sum_{c} p_{cn} - B_n \leq 0, \forall n \in N \end{aligned} \tag{3.4}$$

By using the Lagrangian multiplier μ, the equality constraints that have the global coupling variables of the original problem, are transformed into the objective function of its dual problem (3.4). So for the replicas in our system,

each of them just needs to solve the local optimization problem and update μ by the clients periodically. The local optimization subproblem is defined as (in replica n):

$$\begin{aligned} \underset{p_{cn}}{\text{minimize}} \quad & E_n + \sum_{c=1}^{C} \mu_i \cdot p_{cn} \\ \text{subject to} \quad & \sum_{c} p_{cn} - B_n \leq 0 \end{aligned} \tag{3.5}$$

where $\{p_{cn} \mid c \in C\}$ are the local variables in replica n. The task of updating μ is assigned to the clients since the equality constraints in the original problem (3.2) are associated with each client request. The updating of μ is done by solving the problem (3.6). Gradient method can be used to solve such linear programming problem. μ can be any real number.

$$\begin{aligned} \underset{\mu}{\text{minimize}} \quad & g(\mu) = \inf_{p_{cn}} L(p_{cn}, \mu) \\ \text{subject to} \quad & \mu \in R^C \end{aligned} \tag{3.6}$$

We implement the algorithm as below:

Algorithm 2 Algorithm of LDDM (Replica n)

1: **Initialization:** Set the unit price of replica i.
2: Collect the clients' requests from clients and their value of μ. Tell the other replicas such information.
3: **repeat**
4: Solve the local optimization problem (3.5).
5: Send solution p_{cn} to each client c.
6: Request the new μ_c from the client c.
7: Stops if $\{p_{cn} \mid c \in C\}$ do not change.
8: **until** p_{cn} do not change.

To achieve higher performance for parallel algorithms used for solving global optimization problem, both low system complexity and high algorithm

convergence rate are required. Comparing with CDPSM, the system with the LDDM implemented has lower complexity. In the system implemented with this method, the cooperation is between client and replica, so there is little communication among the replicas. The size of the solution of each replica is $O(|C|)$. The communication complexity of each iteration is $O(|C| \cdot |N|)$, which is lower than the complexity of using CDPSM shown in previous subsection. In theory, the LDDM also has higher convergence rate than CDPSM. Fig. 3.3 shows the comparison of simulated convergence rates of these two methods. We do the simulation work with three replicas using MatLab. For solving the same optimization problem, the CDPSM converges slower than the LDDM. So theoretically, for our energy-aware replica selection problem, the LDDM is expected to have higher performance.

3.4.3 Examples

I am going to use a simple example problem to illustrate how these two algorithms work for solving optimization problems. The problem is stated as below:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(X) = f_1(x_1) + f_2(x_2) \\ \text{subject to} \quad & h(x_1, x_2) \leq 0 \\ & x_1 \in C_1 \\ & x_2 \in C_2 \end{aligned}$$

In order to solve this problem using consensus based projected subgra-

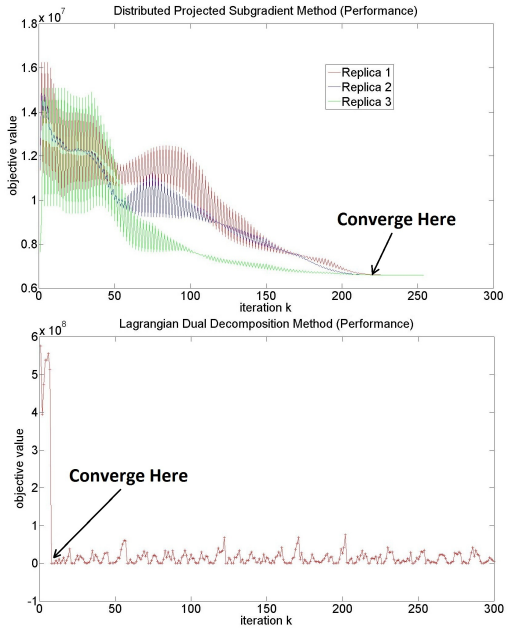

Figure 3.3: Simulation results

dient method, problem can be divided into two sub problems f_1 and f_2 sharing the same constraint set C. We express the solution of them as $X_1\{x_1, x_2\}$ and $X_2\{x_1, x_2\}$. k is the iteration number of this method.

So for the sub problem f_1, $V^{k+1} = a_1 \cdot X_1^k + a_2 \cdot X_2^k - \alpha_k \cdot g_k$ where $a_1 + a_2 = 1$, α_k is the step size which is usually set to be $1/k$, and g_k is the gradient of f_1 at the point $a_1 \cdot X_1^k + a_2 \cdot X_2^k$.

If V^{k+1} is feasible, it is the solution of iteration $k+1$.

$$X_1^{k+1} = V^{k+1}$$

If V^{k+1} is not feasible, the solution of iteration $k+1$ can be achieved by projecting V^{k+1} to the constraint set to find the its nearest solution in terms

of Euclidean distance. To sum these two conditions together, we can achieve the rule of projection as:

$$X_1^{k+1} = arg \min_{X \in C} \|V^{k+1} - X\|$$

Meanwhile, sub problem f_2 also does the same thing. After certain iterations, X_1 and X_2 can converge to the same solution through the consensus procedure.

The Lagrangian dual decomposition method can also be used to solve $f(X)$. We can formulate the Lagrangian function $L(X, \lambda)$ by using the Lagrangian multiplier λ:

$$L(X, \lambda) = f_1(x_1) + f_2(x_2) + \lambda \cdot (h_1(x_1) + h_2(x_2))$$

So the problem is constitute with two sub problems:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_1(x_1) + \lambda_1 \cdot h_1(x_1) \\ \text{subject to} \quad & x_1 \in C_1 \end{aligned}$$

and

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_2(x_2) + \lambda_2 \cdot h_2(x_2) \\ \text{subject to} \quad & x_2 \in C_2 \end{aligned}$$

The value of λ can be determined by solving

$$\inf_X (L(X, \lambda))$$

which is a simple linear programming problem.

Chapter 4

Results and Discussion

In this section, we first present a system which can mimic the behaviors of data centers in cloud in terms of energy consumption. Then, we use two types of data-intensive applications, video streaming and distributed file services, to test the prototype of our decentralized replica selection system. At last, we analyze the performance and energy cost of our system implemented with two distributed replica selection algorithms and demonstrate that LDDM outperforms CDPSM and baseline round-robin method in both performance and energy cost.

4.1 Assumptions and System Setup

4.1.1 Assumptions

In the following experiments, we are going to use a single cluster node to function like a real replica. We assume that, for data-intensive applications, energy consumption model of a single cluster node is very similar to that of a data center. It can be easily proved as below. Assuming we have workload p, we can know from the equation (3.1) that the energy consumption(E_s) of a

single cluster machine is:

$$E_s = \alpha p + \beta p^3 \tag{4.1}$$

If we are using a data center to handle p client requests, the task can be split into p_i where $\sum_{i=1}^{N} p_i = p$, N is the number of nodes involved with this task in this data center. So the energy consumption(E_d) of this data center for request p is:

$$E_d = \sum_{i=1}^{N} (\alpha p_i + \beta p_i^3) = \alpha \sum_{i=1}^{N} p_i + \beta \sum_{i=1}^{N} p_i^3 = \alpha p + \beta \sum_{i=1}^{N} p_i^3 \tag{4.2}$$

In reality, the energy consumption of network devices is much lower than that of servers in a data center. Therefore, we can assume that the value of β is much smaller than α in equation (3.1). So we can have $E_s \approx E_d$. Based on this equation, it is reasonable for us to use a cluster node to model the energy behaviors of a real replica in cloud.

4.1.2 System Setup

We use eight nodes of our SystemG cluster as replicas to conduct our experiments. They are strongly connected with each other in Ethernet. The SystemG cluster is a 22.8 TFLOPS supercomputer providing a research platform for development of high performance software and simulation tools. Each node is equipped with two quad-core 2.8 GHz Intel Xeon Processors, an 8 GB RAM, and a 6MB cache. SystemG is also equipped with both Ethernet and Infiniband adapters and switches. In this experiment, we use the Intelligent

Power Distribution Units(Dominion PX) to dynamically profile power consumption of controlled machines. The power sampling rate is 100 times/sec.

The model parameters used in the experiments in this section are defined in Table 4.1. We also set the value of scalars a_n,α_n, and β_n in Table I to be 1.

Table 4.1: Parameters setup in the model

Replica	1	2	3	4	5	6	7	8
Elec. Price (¢/kwh)	1	8	1	6	1	5	2	3
Band Cap (MB/s)	100	100	100	100	100	100	100	100

In our experiments, we use two types of data-intensive applications: the video streaming and the distributed file service. The size per request is different for these two applications. We set the size per request for the video streaming is approximately 100 MBytes and for the distributed file service it is approximately 10 MBytes. The experiments last for several minutes (depends on the request size and amount of requests). All the requests are sent from one client. The value of requests is shown in Fig. 4.1

The clients' requests arrive at the cloud one right after another. For example, the first request arrives at the cloud. Then data centers cooperate with each other to achieve the solution. After the calculation is done, the second request arrives at the cloud and the threads for downloading the first file start in each data center. In our experiment, we have totally 23 requests sent from the single client.

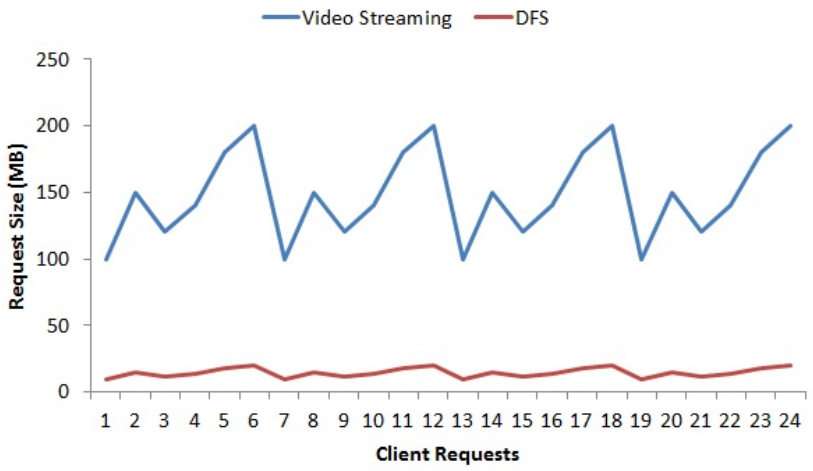

Figure 4.1: Client Requests

4.2 Performance and Power Analysis

In this subsection, we use two data intensive applications to study the power and performance characteristics of the proposed system implemented with CDPSM and LDDM.

The power profiles for using CDPSM and LDDM in our system running with distributed file service are shown in Fig. 4.2 and Fig. 4.3. The system power is consumed by both replica selection (including both local solution calculation and global information cooperation periods), and file transferring after the computation work. The "valleys" shown in these two figures represent the time when only replica selection process is running or system is listening to the new requests. The "peaks" represent the time when replicas are accepting new client requests and transferring files to the previous clients. The execution time of each replica shown in the graphs depends on both assigned workload

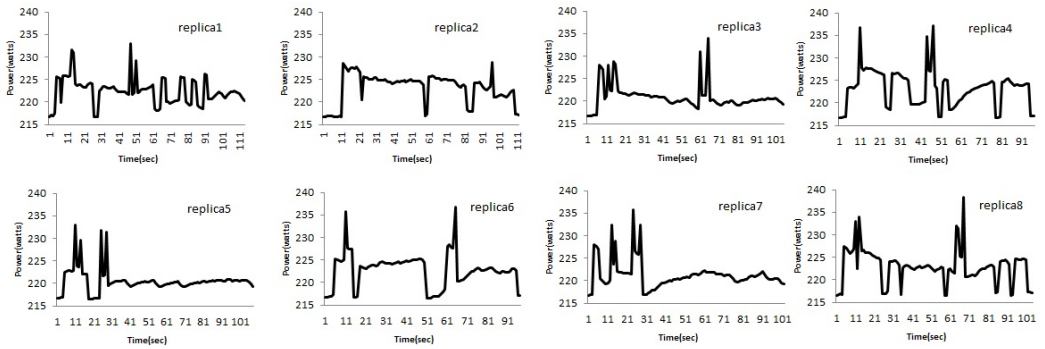

Figure 4.2: Runtime power profile for individual replica using CDPSM (distributed file service)

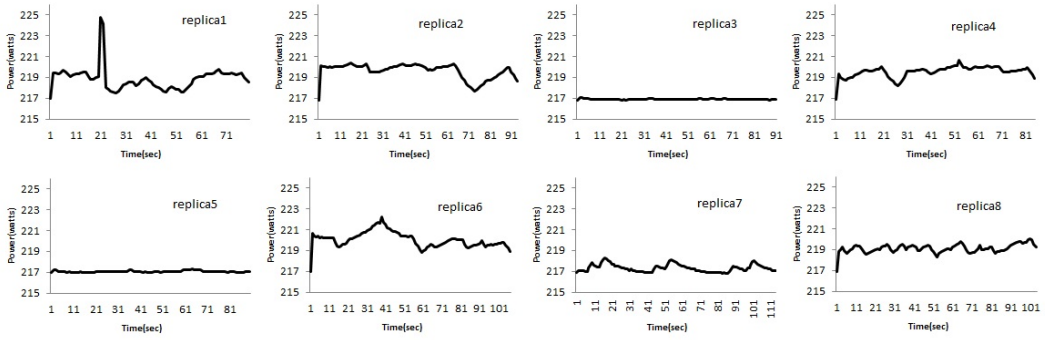

Figure 4.3: Runtime power profile for individual replica using LDDM (distributed file service)

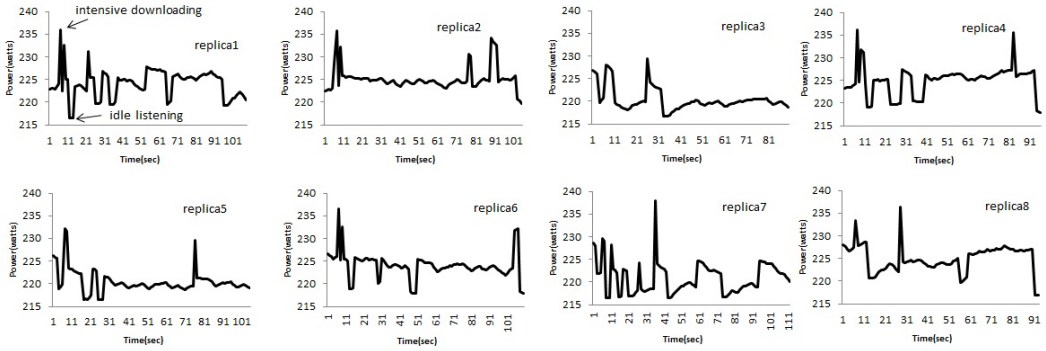

Figure 4.4: Runtime power profile for individual replica using CDPSM (video streaming)

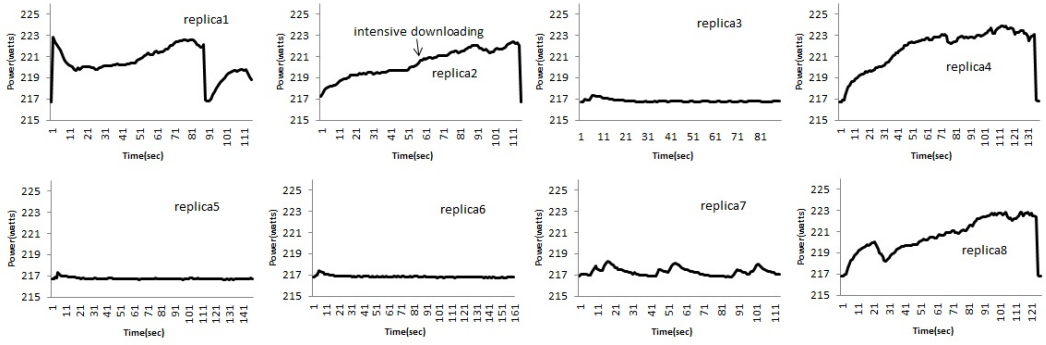

Figure 4.5: Runtime power profile for individual replica using LDDM (video streaming)

and solution calculation time. We can observe that handling the same number of client requests, our system implemented with LDDM runs faster than the system with CDPSM (for most of the individual cases, LDDM finishes earlier). It validates that LDDM has a lower communication complexity and better convergence rate than CDPSM. Also, the average power of using LDDM is lower than that of using CDPSM. It is because the system with CDPSM requires more work (compared to LDDM, CDPSM needs to collaborate with all other replicas and clients at every iteration in order to make scheduling decision) in each iteration than the system with LDDM. This also proves that CDPSM's system complexity is higher than LDDM. In Fig 4.3, we can observe that the power consumption of replica 3 and 5 remain constantly low during the execution. This is because these two replicas either have not been selected as downloading targets by replica selection calculation or have really low workload assigned.

Fig. 4.4 and 4.5 illustrate the runtime power profiles of running video streaming application using CDPSM and LDDM. From these figures, we could easily observe that the runtime load assignment is more balanced for CDPSM than LDDM after global solution calculation period. Replica 2, 4, and 7 share a power climbing trend during the execution. They start with less downloading requests and later on the bandwidth is saturated with more accumulated tasks, which cause the system power to climb. Replica 3, 5 and 6 in Fig. 4.5 are the ones that have little or no load assigned by the algorithm so that they appear to have constant low power consumption. From Fig. 4.4 and 4.5

alone, we can not determine which algorithm has better performance because the time proportion of the global solution calculation time through these two algorithms is much lower for this application due to larger request size and data transferring time. In the next subsection, we will compare their energy cost in order to decide which algorithm should be used in the proposed system.

4.3 Energy Cost Analysis

We evaluate the energy cost of our decentralized replica selection system with that of the system where round-robin algorithm is used to map client requests to the replicas.

We use both video streaming and distributed file service as the applications to evaluate the total energy cost of replicas. For video streaming, the energy cost of each replica and the total energy cost by using three different algorithms are shown in Fig. 4.6 and Fig. 4.7. For the application of distributed file service, the energy cost of each replica and the total energy cost by using three different methods are shown in Fig. 4.8 and Fig. 4.9. The actual numerical energy cost data of the two applications under three different algorithms is presented in Tab. 4.2

From Fig. 4.6 and Fig. 4.8 , we can observe that generally LDDM and CDPSM are better than round-robin method for individual replicas. And in terms of total energy cost shown in Fig. 4.7 and Fig. 4.9, the replica selection system implemented with LDDM has lower energy cost than that of CDPSM and round-robin methods. The reason why LDDM has lower energy cost than

Table 4.2: Actual energy cost of the two applciations under three different algorithms

App	Algo.	Nodes				Total
Video	LDDM	76051.2 192463.2	25560.2 34912.2	156104.8 127148.5	30818.1 56017.6	699292.5
	CPDSM	72588 138871.8	23633 24641	382632 123647.5	21594.9 41838.67	829446.3
	Round Robin	112233 158247	45317 32699.3	297655.2 104750.5	45324.7 54126.4	850353.1
DFS	LDDM	52521.6 117229.2	20641.8 23543.1	157913.6 123867	18660.7 45575.2	559952.2
	CPDSM	53972.7 142891.8	24999.7 21601.9	185511.2 112819.5	21657.1 46369.4	609823.3
	Round Robin	74391.6 115452.7	30289.2 22904.2	204839.3 95478.8	29315.8 38519.5	611191.1

CDPSM is because LDDM has better performance for solution calculation period of replica selection, which results in less energy consumption. Note that, even though CDPSM requires additional energy for computation and communication, it still performs better than round-robin method because CDPSM method does provide the global optimal solution for workload distribution.

4.4 Discussion

From the experiments above, we can get the obvious conclusion that both the replica selection systems using LDDM and CDPSM can reduce energy cost of data centers for data-intensive applications. Also, the efficiency of the system implementing LDDM is much higher than that of CDPSM. This has

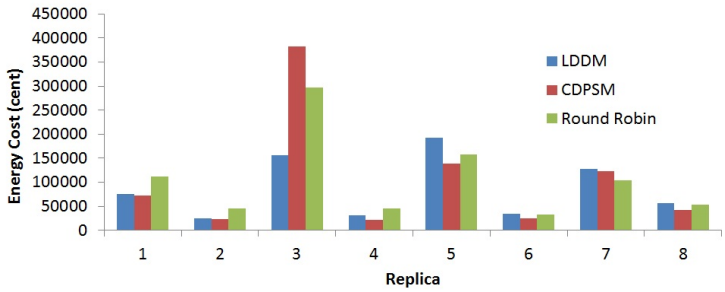

Figure 4.6: Energy cost of each replica for video streaming application

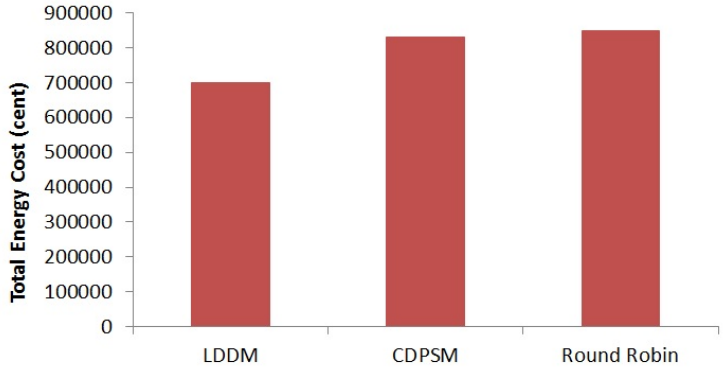

Figure 4.7: Total energy cost of all the replicas for video streaming application

already been proved theoretically as well as by the simulation results. The request data we use in our experiment is twenty continuous data-intensive requests within a few minutes. It simulates generally a worse situation than in the practical cloud, because real time requests in the cloud are less intensively spread in the time dimension.

However, the validation of the actual system performance for both LDDM and CDPSM requires more experiments other than a few minutes of highly data-intensive requests from one client. The details of such experimen-

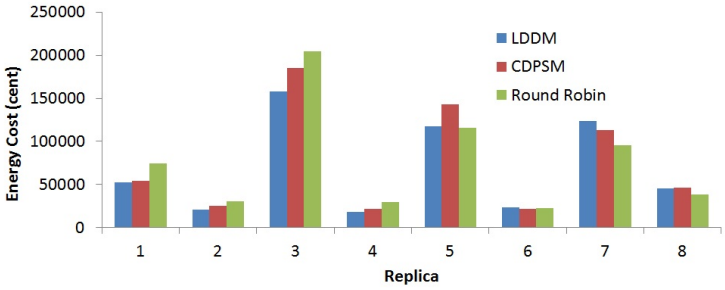

Figure 4.8: Energy cost of each replica for distributed file service

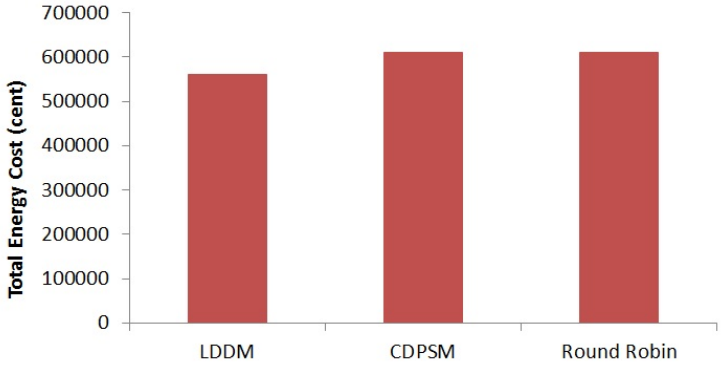

Figure 4.9: Total energy cost of all replicas for distributed file service

tal work is discussed in chapter 5.

4.5 Summary of the Contributions

In my thesis, I present a replica selection system which is used for handling data-intensive applications in cloud. Using such system, total energy cost in cloud can be greatly reduced.

I propose the energy consumption model of a data center based on the current research work. This model is used to formulate the energy cost

optimization problem in cloud.

I designed a simple decentralized architecture for the replica selection system. It can avoid the issue of reliability, scalability, and security. Comparing with the centralized architecture, it does not have the single point of failure problem.

Also, I adapted two methods, CDPSM and LDDM, into our system to solve the global optimization problem in a distributed manner. I designed the algorithms of CDPSM and LDDM for our decentralized replica selection system.

I developed the program of our replica selection system. It is available on the website: http://people.rit.edu/bxl4074/. The program implements CDPSM and LLDM. It can also work as API in the real cloud environment. In the program, we used Mosek [1] package for solving the optimization problem. It can be used to solve linear, quadratic, and cubic optimization problems.

I designed the experiment to validate the replica selection system. The experiment results show that our replica selection system can promisingly reduce the total energy cost of the data centers.

Chapter 5

Conclusion and Future Work

Our proposed system provides a decentralized architecture to solve the energy-aware replica selection problem for data-intensive applications in the cloud. It considers both total energy cost of the entire cloud and bandwidth capacity for each replica when forming the system-wide energy model. The performance of our prototype system proves that it is highly applicable to process different types of data-intensive applications. In the best case scenario of our experiments, the total energy cost using LDDM can be reduced by 17.8% comparing with a round-robin method and 15.3% comparing with CDPSM.

In future, we plan to further improve the LDDM algorithm used by our proposed system in order to achieve better performance and lower total energy cost. For example, the method of choosing the Lagrangian multiplier and the step size can affect greatly the efficiency our system. An appropriate configuration of these parameters can improve our system.

CDPSM has a lower efficiency in our system because it is designed to solve optimization problems which do not have public dependencies (global constraints). If we can adapt CDPSM to the problem with both local and global constraints, we can also improve the efficiency of our system. This

work is also indicated in [15] as part of the future work.

At last, we will try to use practical data in the cloud to test our system. The performance of our system varies if we use practical request data or scaling up the size of our replica selection system. In our experiment, we use a few minutes of intensive requests to test our system. It can be considered as the worst situation because in reality these requests will be spread over a longer period. Also, the size of our distributed replica selection system affects its efficiency. In the system with LDDM implemented, scaling up of system size can lead to a great burden to the clients because each distributed replica needs to communicate with clients during the approximation to the optimal solution. For CDPSM, increasing the number of distributed replicas means more consensus processes, which results in a greater communication overhead among the distributed replicas. It can reduce the efficiency of the replica selection system.

Bibliography

[1] Mosek ApS. http://www.mosek.com, 2012.

[2] L.A. Barroso and U. Holzle. The case for energy-proportional computing. *Computer*, 40(12):33 –37, dec. 2007.

[3] Anton Beloglazov and Rajkumar Buyya. Energy efficient resource management in virtualized cloud data centers. In *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, CCGRID '10, pages 826–831, Washington, DC, USA, 2010. IEEE Computer Society.

[4] A Berl, E Gelenbe, M Di Girolamo, G Giuliani, H De Meer, M Q Dang, and K Pentikousis. Energy-efficient cloud computing. *The Computer Journal*, 53(7):1045–1051, 2009.

[5] Rajkumar Buyya, Anton Beloglazov, and Jemal H. Abawajy. Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges. *CoRR*, abs/1006.0308, 2010.

[6] Jeffrey S. Chase and Ronald P. Doyle. Balance of power: Energy management for server clusters. In *In Proceedings of the 8th Workshop on Hot Topics in Operating Systems, HotOS*, 2001.

[7] Frank Dabek, Russ Cox, Frans Kaashoek, and Robert Morris. Vivaldi: a decentralized network coordinate system. In *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, SIGCOMM '04, pages 15–26, New York, NY, USA, 2004. ACM.

[8] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[9] Albert Greenberg, James Hamilton, David A. Maltz, and Parveen Patel. The cost of a cloud: research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73, December 2008.

[10] N.S. Kim, T. Austin, D. Baauw, T. Mudge, K. Flautner, J.S. Hu, M.J. Irwin, M. Kandemir, and V. Narayanan. Leakage current: Moore's law meets static power. *Computer*, 36(12):68 – 75, dec. 2003.

[11] K. Le, R. Bianchini, M. Martonosi, and T. Nguyen. Cost-and energy-aware load distribution across data centers. *Proceedings of HotPower*, 2009.

[12] Adam Lewis, Soumik Ghosh, and N.-F. Tzeng. Run-time energy consumption estimation based on workload in server systems. In *Proceedings of the 2008 conference on Power aware computing and systems*, HotPower'08, pages 4–4, Berkeley, CA, USA, 2008. USENIX Association.

[13] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H. Low, and Lachlan L.H. Andrew. Greening geographical load balancing. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*,
SIGMETRICS '11, pages 233–244, New York, NY, USA, 2011. ACM.

[14] P. Mahadevan, P. Sharma, S. Banerjee, and P. Ranganathan. Energy aware network operations. In *INFOCOM Workshops 2009, IEEE*, pages 1 –6, april 2009.

[15] R. Masiero and G. Neglia. Distributed subgradient methods for delay tolerant networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 261 –265, april 2011.

[16] A. Nedic, A. Ozdaglar, and P.A. Parrilo. Constrained consensus and optimization in multi-agent networks. *Automatic Control, IEEE Transactions on*, 55(4):922 –938, april 2010.

[17] Lei Rao, Xue Liu, Le Xie, and Wenyu Liu. Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment. In *INFOCOM, 2010 Proceedings IEEE*, pages 1 –9, march 2010.

[18] Arkaitz Ruiz-Alvarez and Marty Humphrey. An automated approach to cloud storage service selection. In *Proceedings of the 2nd international workshop on Scientific cloud computing*, ScienceCloud '11, pages 39–48, New York, NY, USA, 2011. ACM.

[19] Srini Seetharaman. Energy conservation in multi-tenant networks through power virtualization. In *Proceedings of the 2010 international conference on Power aware computing and systems*, HotPower'10, pages 1–8, Berkeley, CA, USA, 2010. USENIX Association.

[20] James W. Smith and Ian Sommerville. Workload classification & software energy measurement for efficient scheduling on private cloud platforms. *CoRR*, abs/1105.2584, 2011.

[21] the U.S. EPA ENERGY STAR Program. http://www.energystar.gov, 2007.

[22] R. Urgaonkar, U.C. Kozat, K. Igarashi, and M.J. Neely. Dynamic resource allocation and power management in virtualized data centers. In *Network Operations and Management Symposium (NOMS), 2010 IEEE*, pages 479 –486, april 2010.

[23] Patrick Wendell, Joe Wenjie Jiang, Michael J. Freedman, and Jennifer Rexford. Donar: decentralized server selection for cloud services. *SIGCOMM Comput. Commun. Rev.*, 41:231–242, August 2010.

[24] Bo Zhai, David Blaauw, Dennis Sylvester, and Krisztian Flautner. Theoretical and practical limits of dynamic voltage scaling. In *Proceedings of the 41st annual Design Automation Conference*, DAC '04, pages 868–873, New York, NY, USA, 2004. ACM.

[25] Z. Zong, M. Briggs, N. O'Connor, and X. Qin. An energy-efficient framework for large-scale parallel storage systems. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, pages 1 –7, march 2007.