MeteCloud: Meteorological Cloud Computing Platform for Mobile Weather Forecasts based on Energy-aware Scheduling

Wei Fang[1,2], Victor S. Sheng[3], XueZhi Wen[1]

1 School of Computer & Software, Nanjing University of Information Science & Technology, China
2 State Key Lab. for Novel Software Technology, Nanjing University, China
3 Computer Science Department, University of Central Arkansas, USA
fangwei@nuist.edu.cn, ssheng@uca .edu, ww_pub@163.com

Abstract

Nowadays, more and more large-scale data intensive applications such as meteorological big data executed in data centers require a huge amount of electrical energy and energy costs. Therefore, minimizing the energy consumption and reducing the environmental impact is our goal of Green Cloud Computing. In this paper, a new meteorological cloud computing platform (MeteCloud) for Mobile Weather Forecasts based on energy-aware scheduling for improving the energy efficiency is proposed. This approach is different from the existing researches, which wants to emphasize the importance of energy consumption in the study of constructing cloud computing platform for meteorological applications. And, a novel MeteCloud architecture and a hybrid scheduling algorithm are given to testify the availability of meteorological cloud computing platform. Finally, the experimental results demonstrate that MeteCloud has better performance and efficiency.

Keywords: MapReduce, MeteCloud, Energy-aware scheduling, Meteorological cloud computing

1 Introduction

To the best of our knowledge, it is a higher requirement for the information technology in the atmospheric science. A large number of scientific computing demands in the meteorological science are increasing every day, in order to obtain accurate weather forecast information for complex weather forecasting systems, such as typhoon track forecast [1]. In the past, the meteorological organization used the traditional "chimney" mode to deploy the application system. This deployment model from basic (hardware) to the top (software) looks like a chimney. So, this model will inevitably lead to wasting all kinds of resources. Their energy consumption and space occupation are also very large. Moreover, this traditional model also makes it very difficult to backup and update.

Facing above challenge, cloud computing is the best solution to the problem. The private meteorological cloud will be able to integrate these isolated servers into one whole part, as forming a large resource pool. So, it not only can greatly improve the utilization rate of resources but also reduces energy consumption. A significant advantage of cloud computing is that it reduces the computing cost and improving time efficiency of applications [3]. Under the support of cloud computing platforms, users of meteorological operations can demand access to various services (such as precision computing, information storage, network services, software usages, data sharing, etc.). They do not need to know what cloud computing is or how it works since cloud computing integrates a large number of resources together as a virtual resource pool [4].

Meteorological cloud platform (MeteCloud) is the application of cloud computing technology in meteorological service, which provides these capabilities including resource sharing, rapid expansion, customizable, low-cost of meteorological data processing and service etc. Meteorological data are the most crucial and most valuable resource for our life. With the rapid development of satellite, radar, automatic weather stations, etc., the large amount of data grows up in 100TB magnitude annually. Furthermore, the amount of data observed is also updated frequently. Towards such a large scale meteorological data problem, high-efficient computing power (more than a trillion times) is urgently required. Therefore, establishing a cloud computing weather information processing system is very important and significant. An elastic Meteorological Cloud Computing platform based on energy-aware scheduling is proposed in this paper.

The rest of the paper is organized as follows. Section 2 discusses some related work. The architecture and the higher level design of the elastic meteorological cloud platform in Section 3 are presented. Section 4 provides the details on specific

*Corresponding Author: Wei Fang; E-mail:fangwei@nuist.edu.cn
DOI: 10.3966/160792642018051903031

elastic scheduling used in the elastic meteorological cloud platform. Section 5 implements this system, conducts experiments and analyzes the experimental results. Finally, Section 6 concludes the paper with a summary.

2 Related Work

As cloud computing better business prospects, some major companies such as IBM, Google, Microsoft, Amazon, VMware and Alibaba all have launched their own cloud computing platform or related products [5], for examples, IBM Blue Cloud, Google Google App Engine, Microsoft Azure Service Platform, Amazon EC2 and VMware vSphere, etc. However, for some important fields such as military, meteorological, etc., are not suitable to use this public cloud according to security and privacy issues.

In the meteorological field, there is a little research about cloud computing application. Andrew L. Moltha, et al. focus on weather forecasts and applications within cloud computing environments. They use the Amazon Elastic Compute Cloud (EC2) resources to apply meteorological cloud applications. S. Venkataraman, et al. propose a cloud system named CloudCast, an application that provides short-term weather forecasts depending on users' current location. They only apply the cloud computing technology to the meteorological applications [2]. But they do not consider the energy consumption issue in their cloud applications.

One of the first works, Pinheiro et al proposed a power management method at the data center level of cloud computing platform [6]. There exists an extensive research on resource allocation and scheduling in data centers and cloud computing construction that does not consider the energy efficiency issue (e.g., [7-8]). Anton Beloglazov, et al. designed an architectural framework and principles for energy-efficient Cloud computing [9]. Calheiros et al. have investigated the energy problem of mapping VMs on physical nodes optimizing network communication between VMs [10]. [11] presented a system named KMN for analytics frameworks that leverage key property choices to perform data-aware scheduling. It is, therefore, imperative that schedulers for cloud computing exploit the available choices to improve performance. X.M Zhu etc. thinks energy conservation is a major concern in cloud computing systems [12]. Then, they propose a novel rolling-horizon scheduling architecture and energy consumption model for real-time task scheduling in virtualized clouds. G.V Laszewski et al. [13] suggests scheduling VM in a computing cluster to reduce power consumption through Dynamic Voltage Frequency Scaling (DVFS), implementation of energy efficient algorithm to allocate VMs. B.Li et al. proposed an Energy-aware heuristic algorithm on the base of distributes workload in VM with a minimum number of VMs or nodes workload [14]. Z.J. Fu, et al. designed a tree-based index structure which supports parallel search to take advantage of the powerful computing capacity and resources of the cloud server. And, the search efficiency is well improved [15]. [16] introduces Job Consolidation Algorithm (JCA) that efficiently utilize the resource in the cloud considering machine heterogeneity, and implement DVFS technique which remains efficiently to produce liable replacement among jobs guaranteeing a reduction in energy consumption. It should be noted that these methods do not consider resource virtualization, the most important feature of clouds, thus they do not efficiently improve the resource utilization in real cloud computing applications, especially in the meteorological field.

To the best of our knowledge, the work presented in this paper is the first to look into the feasibility of meteorological and research clouds for real-time application of weather forecasting using the energy-aware scheduling. So, the exploration of our novel approach is timely and crucial considering the proliferation of cloud computing environments for large scale of meteorological data applications.

3 MeteCloud Architecture

3.1 Components of MeteCloud

The overall basic architecture of the MeteCloud based on Xen is shown in Figure 1, which includes three components, i.e., meteorological users, a MeteCloud management control center, and a virtualization cluster. Meteorological users can use PCs, iPad and Mobile phone to connect to the MeteCloud management control center, to gain a lease of using virtual machines, including performing a virtual machine startup, shutdown, and other operations.

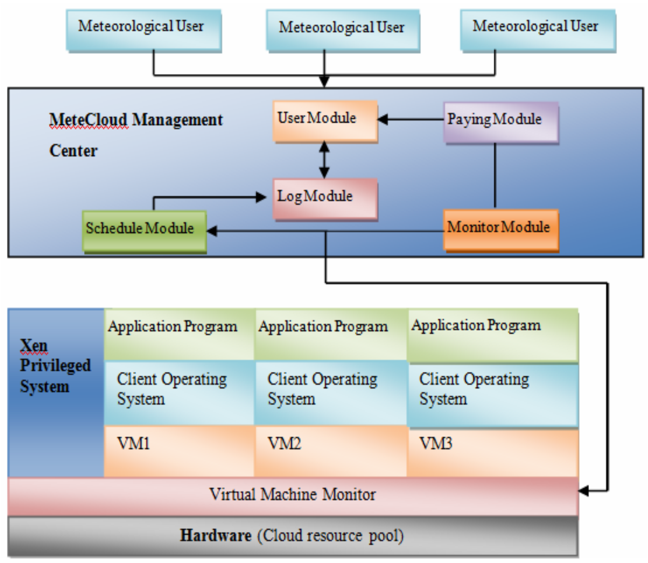


Figure 1. The architecture of MeteCloud

The management control center is the core of the whole elastic meteorological cloud platform. It contains a number of modules to perform all administrative tasks, including a monitoring module, a scheduling module, a logging module, a paying module, and a user module. It dynamically schedules virtual machines for users. Thus, it ensures that the elastic meteorological cloud platform performs flexible scheduling.

MeteCloud will organize a wide variety of physical computing resources in a large resource pool. The MeteCloud creates a virtual pool above the physical computer resource pool and turns the virtual pool into a new meteorological data processing center. Meteorological users only need to communicate with MeteCloud by different equipment such as Mobile phone, iPad, Laptop computer, and remote Server.

3.2 Design of the Virtual Cluster Architecture

The overall architecture of the MeteCloud is based on Xen. A physical machine is a member of a cluster. Nodes are the basic cluster infrastructure, and they can fault tolerant in order to achieve high availability for instance. A virtual machine can run on a cluster. In Figure 2, VM1, VM2, and VM3 are virtual machines, which are used to construct the virtual cluster. Each virtual machine is a virtual cluster computing node running Xen virtualization software. The virtual machines perform a variety of complex scientific computing, in response to various requests submitted by users. The number of jobs allocated to each VM is dynamically changed by elastic scheduling module operations. If current virtual machine node doesn't process so many submitted jobs timely and accurately, then a flexible scheduling system will adjust its load by assigning some jobs to another virtual machine node, according to the scheduling policy of the virtual machines in the cluster. Under this situation, the number of running virtual machines is increased. When the amount of work is reduced, some virtual machines in the cluster will be idle. Then the flexible scheduling system recycles these idle virtual machines and releases the resources to improve the utilization of CPU and memory [17-18].

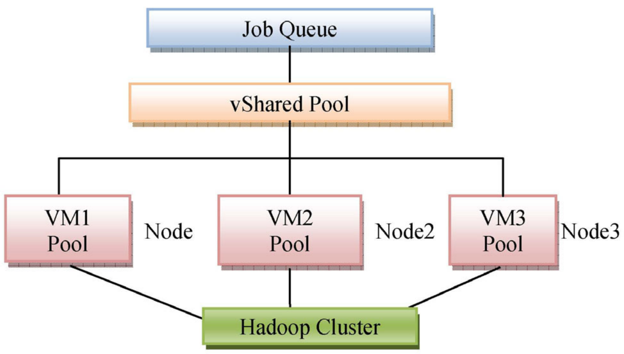


Figure 2. Virtual cluster architecture

The cluster work management system is an important component of the whole cluster system, which used widely in the PBS, CONDOR, LSF, TORQUE etc. And, PBS [19] (Portable Batch System) is an open source, easy using, and a complete API, which can apply the new scheduling strategy, be supported by multiple operating modes.

3.3 Modules of the Management Control Center

Monitoring module. The monitoring module consists of two parts. One part is located on the physical host server-side, and another is the client of the part on the server-side. The part of the server side is responsible for monitoring the physical hosts and virtual machine resource usage. The part as the client issues the requests of aggregating data to the server-side on a regular basis. This module is constructed by using the Libvirt library. With Libvirt, the monitoring module of a virtual machine can provide a series of operations, such as start, pause, stop, restoration, preservation, and migration. It can also get the running status and the resource usage information of physical hosts and virtual machines.

The monitoring method using Libvirt is shown in Figure 3. The libvirtd program is the server side daemon component of the libvirt virtualization management system. Physical hosts on Libvirtd are registered in the virtual machine cluster. A domain (refer to Dom0 shown in Figure 3) is running in a virtual machine, whose main responsibility is to collect the usages of physical hosts, virtual machine CPUs and memory, and network resources. Through the API and the Libvirt, the monitor module defines common protocols for remote communications of transferring monitoring data. Then, it defines data structures to obtain the usages of physical hosts and virtual machine resources. The data structure is shown as follows.

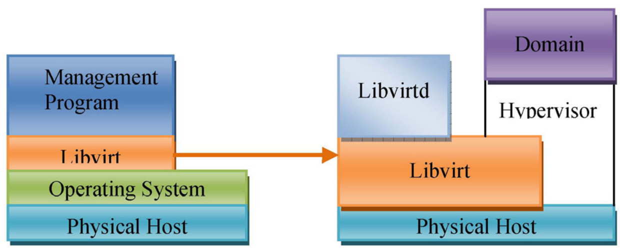


Figure 3. Monitor modular using Libvirt

Struct DomainInfo
{ Unsigned char state; //Current status
Unsigned long maxMem; //Maximum memory
Unsigned long memory; //Usage memory
Unsigned short nrVirtCpu; //Virtual CPU number
Unsigned long cpuTime; // Virtual CPU Running time
Unsigned long network; //Network brand
}

Scheduling module. The Scheduling problem is the map between tasks and resources. There are values of N independent tasks that have same QoS requirement and values of M heterogeneous computing resources that meet the computing requirement in the cloud. The scheduling objective is to minimize Makespan and total energy consumption between tasks and resources.

Power consumption by computing in cloud platform is mostly determined by the CPU, memory, disk storage, and network interfaces. And the CPU consumes the major part of entire energy [19]. Then, this paper focuses on CPU power consumption to monitor the CPU utilization. The Scheduling module can obtain the running data including CPU, memory, storage, network, etc. When a user presets a threshold and a scheduling strategy, the scheduling module chooses virtual machines as well as objects to migrate within the virtualization cluster. And, the number of applications running on each virtual machine of MeteCloud is different. The loading issue of each physical host and each virtual machine can be solved using a loading balancing and scheduling policy. The procedure of loading balancing and scheduling policy is as follows. The virtualized cloud is represented by infinite set $H=\{h_1,h_2,\ldots, h_m\}$ of physical computing hosts. For a given host h_i, it is characterized by its CPU performance, the amount of Memory, storage and network bandwidth, then, $h_i=\{c_i, m_i, s_i, n_i\}$, where c_i, m_i, s_i, n_i denote the CPU capability, Memory, storage and network bandwidth of the i-th host, respectively.

We define $C=\{R,T\}$ as resource-task model, where resource $R=\{r_1,r_2, \ldots, r_m\}$ denotes a set of resources, r_i is the i-th resource. Task $T=\{t_1,t_2, \ldots, t_n\}$represents a set of independent tasks, where t_j is the j-th task.

In this work, we use the power model defined in (1) to obtain the CPU monitoring values.

$$P(u) = k \cdot P_{max}+(1-k) \cdot P_{max} \cdot u \quad \textbf{(1)}$$

Where $P_{\max}$ is the maximum power consumed when the server is fully utilized; k is the fraction of the power consumed by the idle server (i.e.70%), and u is the CPU utilization. In our experiments, P_{max} is set to 300W. The CPU utilization is a function of time and is represented as u(t). Then, the basic energy consumption E_{basic} within a period of time $[t_0,t_1]$ by a physical node H_i is defined in (2).

$$E_{basic}= \int_{t_0}^{t_1} P(u_i(t))dt \quad \textbf{(2)}$$

When loading exceeds the threshold on a virtual machine, the scheduling module will observe the next N monitoring values every ten minutes. Within the N monitoring values, if there are more than K values exceeding the threshold (i.e. 0.95), then the scheduling module is triggered to conduct the resource re-allocation or dynamic migration. Note that the number N is decided by the elastic meteorological cloud platform. K is determined based on the indicated ratio K/N set up in advance by a meteorological user. That is, meteorological users can choose the scheduling policy automatically by setting up the ratio K/N. If a user selects the ratio value close to 0, it means that the users adopt a more aggressive scheduling strategy in the peak to virtual machine scheduling. If the user selects the ratio value close to 100%, then the user adopts a more conservative scheduling policy to a certain peak to schedule virtual machines. Current studies seldom consider the energy consumption of migration. This work considers this issue. Suppose there is a period of time $[t_0,t_1]$, a VM running in the cloud server H_i migrates to cloud server H_k, let t_s be the start time of VM migration, and let t_c be the complete time of VM migration, where $t_0 \le t_s \le t_c \le t_1$. Then the energy consumption of migration $E_{migrate}$ is denoted in (3):

$$E_{migrate} = \int_{t_s}^{t_c} P(\propto \cdot u_i(t))dt + \int_{t_s}^{t_c} P(\beta \cdot u_k(t))dt \quad \textbf{(3)}$$

Where, α and β represent the CPU usage of host H_i and H_k, respectively. Consequently, the total energy consumption to execute all the allocated tasks in a cloud server can be denoted as (4):

$$E_{total} = E_{basic}+E_{migrate} \quad \textbf{(4)}$$

From the above analysis of energy consumption, the less running costs, the less migration, and the less energy consumption can be obtained. To make virtual machine scheduling decisions, the scheduling module first determines the usage of resources in the virtual machine, and further determines whether the physical host of the virtual machine needs to re-allocate appropriate resources. Thus, it can reduce the network overload via virtual machine migrations.

Paying module. The paying module calculates the cost of different users according to the statistics of different users. Then it sends the cost of the subscriber module. The cost is corresponding to the scheduling policy in accordance with the thresholds set up by users, and their physical resource usages. Besides, the standards and methods of payments are different for different storage usages, memory usages, numerical calculations, CPU usages, and network usages. It also decides the demand for computing resources and priority service allocations effectively.

And, let $P(R_k)= (P_c^k, P_m^k, P_s^k, P_n^k)$ as the CPU capability, Memory, storage and network bandwidth of k-th host price, respectively. The cost only considers the energy consumption and VM migration. Given an assignment A_0, performance profit function is $B(A)$, the cost of energy consumption is $P(A)$, the cost of migration is Mig, then as to every assignment A, we want to find an A_j(Where $x_{i,j}$ represents the assigned resources to application V_i in Host H_j), So the maximum profit is in (5):

$$\max \sum_{i=1}^{N} \sum_{j=1}^{m} b(x_{i,j}) - \sum_{j=1}^{M} P(A_I) - Mig(A_O, A_I) \quad \textbf{(5)}$$

Log module. After the information is obtained from the monitoring module, the log module records the state of the physical hosts, virtual machines running within the MeteCloud, resource usages, users, and the virtual machine operating conditions, which are used by users. The log module is conducive to the user who has rented the virtual machine management. The feedbacks of the virtual machine operating and the control center, which are recorded in the log module, will facilitate the user at any time to view. The log module is also convenient for managing the elastic meteorological cloud platform. The administrative center of virtual machines is responsible for virtual machine crashes when the user rents it. The cause of the crash and other information will be written to the log module, so it is convenient for inspection technicians to find the cause of the crash, and to help us restore the system. These historical data of the resource usage and log can be used to improve resource allocation decisions.

User module. The user module includes both servers and clients. A user can send a request by a client to its server-side, which can create, start, pause, stop, restoration and preservation of the operation of a virtual machine, and can also set their own scheduling threshold and strategy of the virtual machine. The user can also get the running state of the virtual machine leased, such as the status of its memory, CPU, and network, and the fees to be paid. The corresponding server receives the request from the user module and then passes the request to the monitoring module to process. The user's settings, such as the threshold and the scheduling policy, are passed to the dispatch center, while the state of the virtual machine hired and the cost is returned back to the user.

4 Energy-Aware Scheduling Methods of MeteCloud

4.1 Dynamic Resource Allocation

When the resources and services that current physical servers provide does not meet the operational requirements of the virtual machine, the virtual machine can also be migrated to a higher level physical server to get better service resources.

We use the minimization of migration policy in our MeteCloud system. The Minimization of Migrations (MM) policy selects the minimum number of VMs needed to migrate from a host to lower the CPU utilization below the upper utilization threshold if the upper threshold is violated. Let V_j be a set of VMs currently allocated to the host j. Then $P(V_j)$ represents the power set of V_j. The MM policy gets a set $R \in P(V_j)$ defined in equation (6) [9].

$$R \begin{cases} \left\{ S \mid S \in P(V_j), u_j - \sum_{\upsilon \in S} u_a(\upsilon) < T_u, |S| \mapsto \min \right\}, & \text{if } u_j > T_u; \\ V_j, & \text{if } u_j > T_l; \\ \phi, & \textit{otherwise} \end{cases} \quad \textbf{(6)}$$

Where T_u refers the upper utilization threshold; T_l represents the lower utilization threshold; u_j is the current CPU utilization of the host j, and $u_a(v)$ is the fraction of the CPU utilization allocated to the VM v.

4.2 Hybrid Energy-aware Scheduling Policy

The computing resources of the dynamic virtual machine with the elastic scheduling module can solve this issue to meet the needs of the user. The elastic scheduling module dynamically adds new computing resources to the virtual cluster [20-21]. The workflow of the elastic scheduling module is shown in Figure 4.

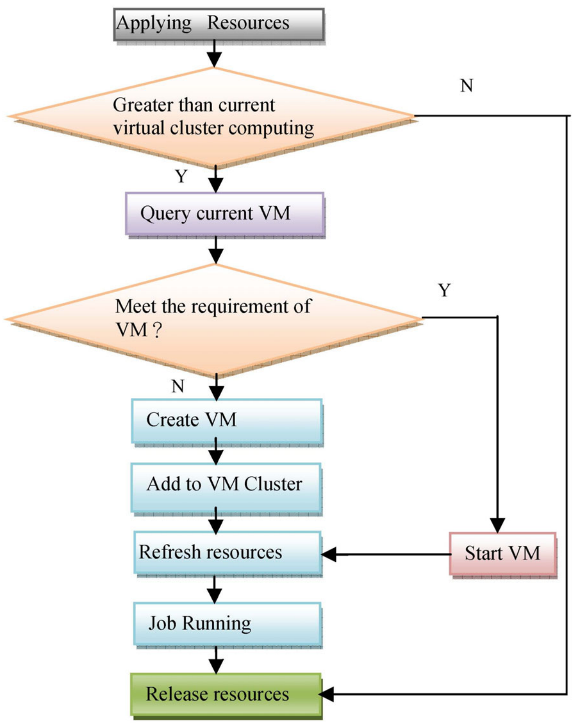


Figure 4. Work flow of the elastic scheduling module

According to Figure 4, the main function of the elasticity scheduling module is shown as following:

(1) Create a virtual machine (VM)
(2) Start a VM and log in the VM
(3) Refresh the current computing resources
(4) Release the idle computing resources

Our hybrid Energy-aware schedulig policy can obtain a good trade-off between the guarantee ratio and energy saving by dynamically starting hosts, closing hosts, creating VMs, placing VMs and migrating VMs according to the system workload. The pseudo code of Minimization of Migrations algorithm and Energy-aware Best Fit Decreasing algorithm are as follows.

Algorithm 1: Minimization of Migrations (MM)

```
Input: hostList, VMList,
        threshold_Up Tu, threshold_Low Tp
Output:Migration VMList
for each h in hostList do
      VMList←h.getVMList()
   VMList.sortDecreasingbyCPUUtilization
      hostUtil←host.Util()
bestFitUtil←MAX
While hostUtil> Tu  do
   For each vm in VMList do
     if VM.getUtil()>hostUtil- Tu then
       tmp←VM.getUtil()-hostUtil+ Tu
       if  tmp<bestFitUtil then
          bestFitUtil←tmp
bestFitVM←vm
end if FitVM
else  if  bestFitUtil=MAX then
        best←vm
           break
          end if
end for
     hostUtil←hostUtil-bestFitVM.getUtil()
         migrationList.add(bestFitVM)
         VMList.remove(bestFitVM)
      End While
 If hostUtil< Tp   then
        migrationList.add(host.getVMList())
VMList.remove(host.getVMList())
  End if
End for
Return migrationList
```

We sort all migration VMs in decreasing order of their current CPU utilizations and allocate each VM to a host that provides the least increase energy consumption of all object hosts. The complexity of the allocation algorithm is $n \bullet m$, where n is the number of VMs that have to be allocated and m is the number of hosts. The pseudo code of Energy-aware Best Fit Decreasing (EBFD) is Algorithm 2.

Algorithm 2: Energy-aware Best Fit Decreasing (EBFD)

```
Input: hostList, VMList
Output: allocation of VMs
 VMList.sortDecreasingbyCPUUtilization()
For each vm in VMList do
    minPower←MAX
   allocatedHost←NULL
   For each host in hostList do
      If host has enough resource for vm then
         power←estimatePower(host,vm)
         if power<minPower then
            allocatedHost←host
            minPower←power
        end if
     end if
end for
if allocatedHost ≠NULL then
   allocate vm to allocatedHost
 end if
return allocation
```

5 Experiments and Result Analysis

In this section, we conduct experiments to compare the effectiveness between the physical host and the elastic MeteCloud. And, we also evaluate our design with existing algorithms: Power-aware scheduling algorithm and Energy-aware heuristic algorithm.

Power-Aware scheduling algorithm [13]. It calculates cluster to minimize power consumption via Dynamic Voltage frequency scaling (DVFS). And it can make significant energy savings without increasing execution time by varying scheduling granularity.

Energy-Aware heuristic scheduling algorithm [14]. It can estimate the number of future requests to predict the future state of the system and perform necessary reallocations. If it is low then energy consumption reduces, it schedules VM workload in such a way that it requires less number of VMs. The algorithm contains these processes of workload migration, virtual machine migration and resizes of workload.

Note that the same amount of meteorological queries from several users completely runs on the physical host and on the elastic cloud platform respectively. The amount of time used on the physical host and on the elastic cloud platform is shown in Table 1 respectively, where the operating time of the physical host takes its running time and waiting time into consideration. Time spending by the elastic cloud platform includes run time and the time of creating a virtual machine.

Table 1. Running time

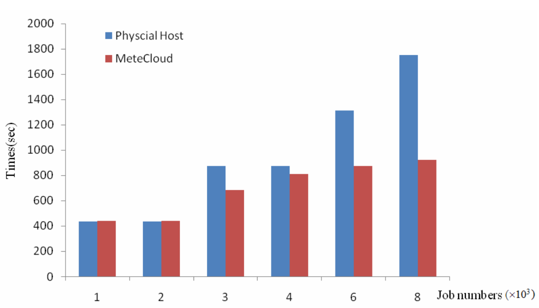
Number of Jobs ($\times 10^3$)	Running Time on a physical host /waiting time (S)	Jobs Running Time on the elastic cloud platform/ creating VM time (S)
1	436.45/0.00	440.55/0.00
2	436.56/0.00	441.45/0.00
3	874.82/436.89	681.01/232.64
4	871.61/434.72	808.03/348.24
6	1310.41/878.05	871.54/372.16
8	1749.34/1311.90	920.40/398.28

As shown in Table 1, the running time for both the physical host and the elastic MeteCloud platform is increasing with the increment of the number of meteorological job submissions. However, the increment of the running time of the elastic cloud platform is much smaller than that of the physical host. The more the number of meteorological job submissions, the more significant the advantage of the elastic MeteCloud is.

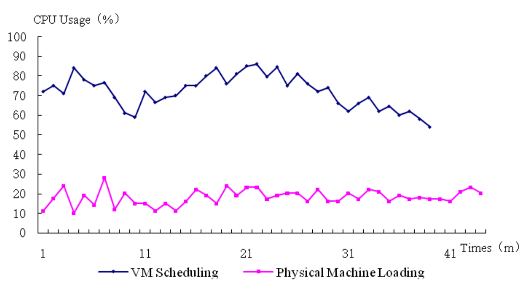
5.1 Comparison of Jobs effectiveness between physical host and MeteCloud

As shown in Figure 5(a), along with the job submitted by the increase in the amount of elastic weather cloud platform through elastic scheduling technique and the virtualization technology to create a new virtual machine computing node and added to the virtual cluster to meet the demand for the computing resources of the job, reduce the waiting execution time. All jobs run less than the time it takes from submission to completion of the run time spent on physical hosts. This shows that the elasticity of the Metecloud platform scheduling jobs caused by the lack of computing resources and technology can effectively solve queued for too long or do not run.

One of the main criteria for evaluating the performance of the elastic meteorological cloud platform is the scheduling performance. The scheduling performance is measured in term of CPU usages. The experiments are conducted to evaluate the scheduling performance of the virtual scheduling, compared with the physical host under different levels of loads. The experimental results are shown in Figure 5(b). Our elastic scheduling module increases its physical CPU resources in a timely manner. Thus, our elastic scheduling approach ensures the high availability of the CPU. However, when a virtual machine enters its idle state, the elastic scheduling module releases its physical CPU allocation, so that these recycled CPU resources can be reallocated to other virtual machines.



(a) Jobs running times



(b) Scheduling performance under Different Loads

Figure 5. Running performance of MeteCloud

5.2 Scheduling Performance

In our simulations, a data center comprising 100 physical nodes. Each node has 4GB RAM, 1 TB of storage and one CPU core and with the CPU performance equivalent to 1,500 MIPS, 2,000 MIPS, or 2,500 MIPS. The energy consumption rate of the three different kinds of hosts is 200, 250, or 400 W. The start-up time of a host is 85 s and the creation time of a VM is 15s. And, the simulations generate 1000 VMs and test several Terasort workloads from the HiBench benchmark set [14]. The task is set to 2500*24*60*60MI.

The Minimization of Migrations (MM) policy and Energy-aware Best Fit Decreasing (EBFD) policy to the benchmark experiment compared with existing Power-aware scheduling algorithm (PS) [13] and Energy-aware heuristic algorithm (EH) [22] is performed. First, the experiment runs 30 times respectively and take the average result of experiments as the data of figures. The results are shown in Figure 6.

Figure 6. Energy consumption of the three energy-saving algorithms under the different number of cloud servers

It can be seen that our scheduling policy has a significant improvement in saving energy consumption compared to other algorithms. When the CPU utilization load is certain, with the increase in the number of cloud servers, PS algorithm only consider the idle energy consumption and dynamic adjustment of the operating frequency and voltage of CPU (DVFS) will be as much as possible the cloud server for computing, resulting in high energy consumption of the cloud servers. And EH algorithm through some heuristics for dynamic adaption of VM allocation at runtime in terms of energy saving than the PS algorithm has been greatly improved. If less number of machines is used then power is conserved. Our hybrid algorithm proposed has better energy saving effect than the other two algorithms, which is in the way to choose the best migration VM and the deployment of the VM.

6 Conclusion

This paper proposes an elastic cloud platform

(MeteCloud) based on Energy-aware Scheduling for querying and processing meteorological information. MeteCloud integrated the resource management, flexible computing, disaster recovery, and other policy configurations together. In addition, the cloud platform facilitates and expands traditional meteorological industries. Many other real application systems can be developed under the same framework of MeteCloud. With the rise of cloud computing, the data center will become increasingly large scale. Energy issues will become increasingly prominent. However, how to apply the existing virtualization to a larger scale computing environment efficiently and reduce the virtualization overhead has become new challenges. For the future, the scheduling algorithm and visualization clustering will be improved again. We look forward to a new energy-saving technology, which can be realized in various levels of virtualization computing architecture.

Acknowledgements

This work was supported in part by Soft Science Program of China Meteorological Administration (2017 [29]), the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

[1] J. T. Overpeck, G. A. Meehl, S. Bony, D. R. Easterling, Climate Data Challenges in the 21st Century, *Science*, Vol. 331, No. 6018, pp. 700-702, February, 2011.

[2] D. K. Krishnappa, D. Irwin, E. Lyons, M. Zink, CloudCast: Cloud Computing for Short-term Mobile Weather Forecasts, *Proc. of 2012 IEEE 31st International Performance Computing and Communications Conference*, Austin, TX, 2012, pp. 61-70.

[3] J. Dean, S. Ghemawat, MapReduce: Simplified Data Processing on Large Clusters, *Proc. of OSDI'04: 6th Symposium on Operating Systems Design and Implementation*, San Francisco, CA, 2004, pp. 137-150.

[4] A. L. Molthan, J. L. Case, J. Venner, R. Schroeder, M. R. Checchi, B. T. Zavodsky, A. Limaye, R. G. O'Brien, *Clouds in the Cloud: Weather Forecasts and Applications within Cloud Computing Environments, American Meteorological Society*, Vol. 96, No. 8, pp. 1369-1379, August, 2015.

[5] C. Deed, P. Cragg, *Business Impacts of Cloud Computing, Cloud Computing Service and Deployment Models: Layers and Management*, IGI Global, 2013.

[6] E. Pinheiro, R. Bianchini, E. V. Carrera, T. Heath, Load Balancing and Unbalancing for Power and Performance in Cluster-based Systems, *Proc. of the Workshop on Compilers and Operating Systems for Low Power*, Barcelona, Spain, 2001, pp. 182-195.

[7] L. Mashayekhy, M. M. Nejad, D. Grosu, A Truthful Approximation Mechanism for Autonomic Virtual Machine Provisioning and Allocation in Clouds, *Proc. of the ACM Cloud and Autonomic Computing Conference*, Miami, Florida, 2013, pp. 1-10.

[8] Q. Liu, W. Cai, J. Shen, Z. Fu, X. Liu, N. Linge, *A Speculative Approach to the Spatial‐ Temporal Efficiency with Multi‐ objective Optimization in a Heterogeneous Cloud Environment, Security and Communication Networks*, Vol. 9, No. 17, pp. 4002-4012, November, 2016.

[9] A. Beloglazov, J. Abawajy, R. Buyya, *Energy-aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing, Future Generation Computer Systems*, Vol. 28, No. 5, pp. 755-768, May, 2012.

[10] R. N. Calheiros, R. Buyya, C. A. F. D. Rose, A Heuristic for Mapping Virtual Machines and Links in Emulation Testbeds, *Proc. of the 38th International Conference on Parallel Processing*, Vienna, Austria, 2009, pp. 518-525.

[11] S. Venkataraman, A. Panda, G. Ananthanarayanan, M. J. Franklin, I. Stoica, The Power of Choice in Data-Aware Cluster Scheduling, *Proc. of 11th USENIX Symposium on Operating Systems Design and Implementation*, Broomfield, Co, 2014, pp. 301-316.

[12] X. Zhu, L. T. Yang, H. Chen, J. Wang, S. Yin, X. Liu, Real-time Tasks Oriented Energy-aware Scheduling in Virtualized Clouds, *IEEE Transactions on Cloud Computing*, Vol. 2, No. 2, pp. 168-180, April-June, 2014.

[13] G. V. Laszewski, L. Z. Wang, A. J. Younge, X. He, Power-Aware Scheduling of Virtual Machines in DVFS-enabled Clusters, *Proc. of IEEE International Conference on Cluster Computing and Workshops*, New Orleans, LA, 2009, pp. 1-10.

[14] B. Li, J. X Li, J. P. Huai, T. Y. Wo, Q. Li, L. Zhong, EnaCloud: An Energy-saving Application Live Placement Approach for Cloud Computing Environments, *Proc. of IEEE International Conference on Cloud Computing*, Bangalore, India, 2009, pp. 17-24.

[15] Z. J. Fu, X. M. Sun, Q. Liu, L. Zhou, J. G. Shu, Achieving Efficient Cloud Search Services: Multi-keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing, *IEICE Transactions on Communications*, Vol. E98-B, No. 1, pp. 190-200, 2015.

[16] P. Sanjeevi, P. Viswanathan, Towards Energy-aware Job Consolidation Scheduling in Cloud, *IEEE International Conference on Inventive Computation Technologies*, Coimbatore, India, 2016, pp. 1-6.

[17] H. A. Lagar-Cavilla, J. A. Whitney, A. M. Scannell, P. Patchin, S. M. Rumble, E. de Lara, M. Brudno, M. Satyanarayanan, SnowFlock: Rapid Virtual Machine Cloning for Cloud Computing, *Proc. of the 4th ACM European Conference on Computer Systems*, Nuremberg, Germany, 2009, pp. 1-12.

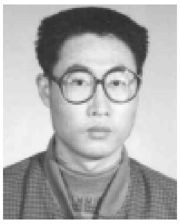
[18] N. Bhatia, J. S. Vetter, *Virtual Cluster Management with Xen*, in: L. Bougé, M. Forsell, J. L. Träff, A. Streit, W. Ziegler, M. Alexander, S. Childs (Eds.), *Euro-Par 2007 Workshops: Parallel Processing, Euro-Par 2007, Lecture Notes in Computer Science*, Vol. 4854, Springer, 2008, pp. 185-194.

[19] Y. Kong, M. Zhang, D. Ye, A Belief Propagation-based Method for Task Allocation in Open and Dynamic Cloud

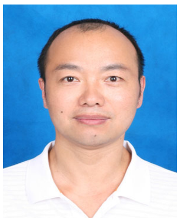
Environments, *Knowledge-based Systems*, Vol. 115, pp. 123-132, January, 2017.
[20] W. Fang, W. B. Pan, Z. M. Cui, View of MapReduce: Programming Model, Methods and its Applications, *IETE Technical Review*, Vol. 29, No. 5, pp. 380-387, September, 2012.
[21] L. Mashayekhy, M. M. Nejad, D. Grosu, D. Lu, W. Shi, Energy-aware Scheduling of MapReduce Jobs, *2014 IEEE International Congress on Big Data*, Anchorage, AK, 2014, pp. 32-39.
[22] S. Huang, J. Huang, J. Dai, T. Xie, B. Huang, The Hibench Benchmark Suite: Characterization of the MapReduce-based Data Analysis, *Proc. of the IEEE 26th Conf. on Data Engineering Workshops*, Long Beach, CA, 2010, pp. 41-51.

Biographies

Wei Fang is an associate professor in the Jiangsu Engineering Center of Network Monitoring at the NanJing University of Information Science & Technology in China. His research interests are in the areas of Big Data Mining and Cloud Computing. He is a member of ACM.



Victor S. Sheng received his Ph.D. Degree in Computer Science from the University of Western Ontario in 2007. He is an assistant professor in the Department of Computer Science at UCA, and the founding director of Data Analytics Lab (DAL). His research interests are data mining and machine learning, and related applications. He is a member of ACM and IEEE.



XueZhi Wen is an associate professor with the School of Computer and Software, Nanjing University of Information Science and Technology, China. He is a member of ACM. His research interests include Pattern Recognition, Image Processing and Cloud computing.