Autonomic Computing Capacity Management Approach with Cloud for Building e-Business

Yuan Wencheng and Zhu Yian

***Abstract*—Web Cluster is able to combine a set of backend capacities to function as a powerful Web Server for handling massive workloads towards successful e-business. However traditional Web Cluster has shortcomings with pre-deployed and pre-configured capacities upon traditional rigid infrastructure, which will lead more cost into redundancy and wasteness because of inappropriate up-front capacity plan. Moreover the more energy will be consumed accordingly, as capacities were over-reserved for the rare peak. Thus a novel autonomic computing capacity management approach for overcoming such issues is therefore proposed in this paper. Firstly we present the novel approach with Cloud insight to provision virtual infrastructure for hosting Web Cluster instead of hosting Web Cluster upon a rigid infrastructure. Afterward we propose an autonomic provisioning mechanism based on Advance Reservation with Lend-Borrow policy and the low-high threshold detection algorithm, in order to enable capacity balancing and repurposing from Web Cluster adaptively with low demand to the other Web Clusters with high demand, so as to optimize the utilization of capacity efficiently on behalf of drawdown cost and energy saving. Finally we verify that, proposals in this paper are capable of provisioning elastic capacities in cost-effective and energy-saving way against capacity redundancy and wasteness, with a couple of simulations.**

***Index Terms*—Cloud Computing, Web Cluster, e-Business, Provisioning Mechanism, Advance Reservation.**

I. Introduction

The Web Cluster comprised a set of Web Server backend nodes as a super Web Server, has been exposed as a prevalent means to server large amount of workloads, through balancing heavy workload to cooperated Web Servers [1], [2].So it's widely used for hosting e-business for enterprise gaining maximum profits from internet. However typical Web Cluster requires enterprise to estimate the scale of business and relevant required peak previously, in order to reserve a set of capacities might involved, before pre-deployment and pre-configuration. However it's not so easy to achieve as the requirement of capacities may vary drastically from one time to another. Hence there must be times in which, enterprise need more extra cost for scaling up Web Cluster towards satisfying peak demand which is higher than reserved capacities, otherwise the Web Cluster has to face such situations e.g., crashing, data losing, etc. In another hand, the over-reserved capacities are probable to become wasteness in the future as demand decreased. Moreover in the case of employing multiple Web Clusters for business, redundancy and wasteness must be there as physical resources need to be pre-deployed and pre-configured respectively into traditional Web Cluster statically. Furthermore the redundant and wasted capacities will aggravate more energy consumption and cost increased. To overcome these issues, traditional researches as Web Cluster [1],[2] with the aim of the load-balance and fault-tolerance, are not helpful due to the fixed and rigid infrastructure is relied on.

Cloud Computing is emerging as a novel paradigm to deliver virtual capacities by the means of virtualization technologies. Particularly it is capable of leveraging infrastructure as a service (IaaS) to deliver virtualized infrastructure, which encapsulates hardware, software and networking in a VM instance, instead of using traditional physical capacities under rigid infrastructure directly [3], [4]. Hence we propose a novel approach to provision Web Clusters for hosting e-business based on Cloud Computing, similar to other literatures presented in section II.

This paper is organized as following. A couple of researches related to this paper would be surveyed in order to further explain our difference to them in next section. Afterward in section III, we present a novel approach for autonomic Web Cluster provisioning based on Advance Reservation with Lend-Borrow policy in Cloud, on behalf of energy-saving and cost-effective capacity's consumption to enterprise. And in section IV, we evaluate our proposed in simulation, and then prove that, capacities for hosting Web Clusters can be provisioning elastically on demand in cost-effective and energy-saving way. Finally we conclude the paper in the rest.

II. Related work

Numerous researches surrounding Cloud Computing have been explored and employed widely around the world. Commercially EC2 [5] offers a public IaaS cloud utility for enabling customers to gain capacities in pay-as-you-go manner initially, and then in 2009 they move forward to provide Amazon CloudWatch [6] service with auto-scaling facility in extending and shrinking EC2 instances on demand for helping customer drawdown cost. Meanwhile many open-source toolkits (e.g., OpenNebula [7], Encalyptus [8], etc.) were implemented and being functionality to establish Cloud so as to further provision virtual infrastructure instead

Manuscript received March 15, 2011. This work was supported by Northwestern Polytechnical University under Grant Z200852

Yuan Wencheng is a PhD candidate of computer school, Northwestern Polytechnical University, Xi'an, Shaanxi, PRC. (e-mail: whimtn@gmail.com).

Zhu Yian, is a professor and PhD supervisor of computer school, and is Vice dean of the School of Software and Electronics, Northwestern Polytechnical University, Xi'an, Shaanxi, PRC. (e-mail: zhuya@nwpu.edu.cn).

of experiencing physical cluster upon rigid infrastructure.

Cloud's IaaS is primary playing as an innovative means to IT infrastructure management in consideration of its inexpensiveness and convenience in IT infrastructure's outfit and maintenance. For instance, [9] given a novel approach to automate the delivery of IT Service Continuity Management (ITSCM) only in the rare event of rehearsal and actual disaster based on Cloud methodology. It replicates configuration state and business state on dedicated Virtual Appliance, which is hosted and controlled by the Cloud provider, on behalf of cost-efficiency to ITSCM. Furthermore IaaS can also be leveraged for reducing energy consumption against IT infrastructure utilization as [10].

Besides, a large amount of literatures were proposed in High Performance Computing (HPC) area, with the aim of establishing elastic cluster for handling job submissions by leveraging IaaS and associated scaling strategies. Typically [11] presented a novel approach using wall-time and estimated waste-time with three policies to represent a dynamical scalable and responsive HPC cluster, which can extend and shrink the scale of cluster dynamically based on the workload of job submission. Particularly [12] and [13] emphasized that it's also important to maximize capacity's utilization efficiently, apart from the capability to scale up and down HPC Cluster dynamically on demand.

Analogously the capable of delivering virtualized capacities based on Cloud's IaaS is also applied to optimize Web Application performance and consolidation, with respective mechanism for auto-scaling Web Cluster elastically on demand regarding to time-varying workload. As we know, Web Cluster's workload is different from HPC Cluster owns, because it is generated from long-lived scalable services and is required to response the request immediately in a very short time. So we need to plan more enough computing capacities to host web cluster for e-business, but the problem is that the over planned capacities would be wasteness. Hence inappropriate up-front capacities reserved to Web Cluster would lead it into interruption by the burst overload in request growth either, or wasted under ineffective utilization.

Recently researches around Web Cluster leveraging IaaS were proposed with some provisioning approaches leveraging specific performance index detection to achieve auto-scale Web Cluster. Typically [14] presented a new elastic Web proxy cache cluster with the use of hybrid resources include local resources and cloud resources, it introduced a performance index named R and a threshold associated with R for provision a encapsulated node to Cluster when the load higher than the threshold, and release the unnecessary node when the lower than the threshold. Similarly [15] presented an approach for scaling Web Cluster dynamically according to the amount of active session. From another view on Web Cluster, [16] highlights that, Web Application can be divided into multi-tiers and each of them is probable to affect the performance when it becomes to bottleneck, scaling of Web Cluster is therefore simplified to extern or shrink the bottleneck-tier. Moreover [16] introduced SLA to guarantee a maximum average response time for satisfying Web Cluster's QoS requirement. Unfortunately all mentions around Web Cluster seem to acquire virtual capacities on demand in pay_per_use mode without capacity reservation, so the Web Cluster has low capacities initially and is just scaled up when reaches the saturation points, which risks Web Cluster in interruption and crashing as sudden burst overload. Moreover they did not take into account how to maximize profit in consumption of capacity as consideration in both [12] and [13], which is the most importance to make enterprise benefits from Cloud involved. Significantly [17] presented a business-driven model to provision IT infrastructure for hosting Web Applications, by the means of capacity plan for making reservation and consuming capacities on demand. However so much effort has to contribute on making such capacity plan and the action for resizing capacity plan will result in congestion in capacity's competition and heavy overhead applied.

For line of business in where that, enterprise is probable to employ a large number of capacities for hosting multiple Web Clusters cooperated, but it's actually a high cost way to do a e-business, so they have to expect a cost-effective and energy-saving way to utilize entire available capacities efficiently towards easing off capacity redundancy and wasted, or to hire more computing capacities from outside in a lower-cost way when there are not enough capacities. Although many researches employing IaaS especially in Web Cluster as above, were proposed and taken advantage in somehow, but there is still no one can address such purposes very well. Thus in order to achieve these goals, contribution in this paper is to propose an innovative computing capacity management approach for provisioning capacities among multiple Web Clusters elastically on demand, based on Cloud Computing.

III. Problem Statement and Methodology

Cloud makes possible to deliver flexible infrastructure as discussed, by leveraging external hypervisor (Xen [18], VMWare [19], KVM [20]) technologies to instantiate capacities as an encapsulated Web Cluster backend node as required. It is therefore desirable for addressing shortages against using capacities throughout traditional rigid infrastructure for hosting Web Cluster. With this capability achieved from Cloud, we are capable of overcoming pre-configuration and pre-deployment issues by enforcing pre-configuring and pre-deploying in a VM instantiation on the fly. In such way, Web Server belongs to specific Web Cluster is replaced with a VM instance, which hires a variety of capacities with flexible personalities and then acquires repurpose-ability, that is useful for scaling Web Cluster elastically in convenience and non-redundancy. Moreover each VM can serve independent workload it owns and is hosted together with other VMs on the same host in energy-saving way instead of hosting web servers on multiple physical machines.

Typically capacity consumption in Cloud is under a lease or contract, which promises a set of capacities with QoS requirement to consumer. With such contract, consumer can hire capacities in two ways consist of reservations in a scope

of Advance Reservation and pay_per_use on demand, under a duration signed with IaaS provider. Advance Reservation is a critical part in this paper as it plays as a commitment to guarantee that, capacities can be offered to consumer on time as assurance signed with Cloud provider in contract [21]. It is composed of several constraints for facilitating the use of capacity following consumer's QoS requirement. However existed successful reservations will probably bring the next one reservation request into rejection since the new request conflicts with the available capacities in rest. [22] was therefore presented against such issue via introduction of negotiation. But such approaches applied to Advance Reservation still has shortcoming in somewhere, consumer still want to hire more capacities even if confliction exists and expect provider can offer them more capacities they can offer in somehow. Moreover capacities in reservation are delivered to consumer whatever they are exploited or not at the beginning of lease or contract, and they are unable to be hired again by another consumer, hence these capacities will potentially lead into a wasteness. Furthermore it's not so easy to make up-front capacity plan for capacity's reservation against the time-varying drastically demand, quite a few capacities are therefore wasted in reservation made for the rare peak, meanwhile more cost will be also charged for over-using capacities from pay_per_used part. Thus in order to accomplish such shortcomings above, a autonomic approach to provision capacities for hosting Web Clusters on demand elastically as [23], will be proposed relied on above achievement. Afterward an Advance Reservation with Lend-Borrow policy refers to [24] will further be considered and improved towards the capability to share and repurpose reserved capacities across multiple Web Clusters for the purpose of cost-effective and energy-saving capacities provisioning.

A. *Cloud-enable Capacity Management Model*

A Cloud-enable system model for hosting application or service is illustrated as Fig.1, it's divided into 4 parts from top to down. User's applications or services are hosted on the top of whole system. It can obtain computing capacities from Provisioner, which is the key component for provisioning capacities automatically to users, and we will go down details in B. As above mentioned, computing capacities are instantiated by Virtualization Toolkit component via interaction with Hypervisors.

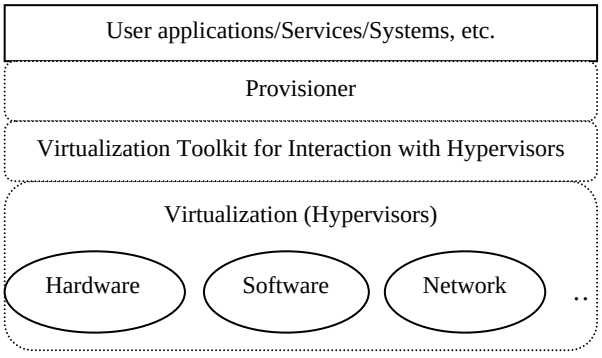

Fig.1 Cloud-enable system model

B. *Autonomic Capacity Management Structure*

Our approach for provisioning Web Cluster automatically focus on the Provisioner part, and are constructed with five key components, they are able to interoperate for repurposing capacities on demand dynamically in Cloud environment, as illustration in Fig.2.

As discussed, Web Cluster consumer need make a lease or contract with reservation at first, in order to guarantee the minimum capacity's requirement for serving general purpose. Afterward capacities consumption and health information will under monitoring so as to gather information for repurposing Web Cluster's capacity on demand. Combined such information with Web Cluster's workload sampling, we can make decision to balance and repurpose capacities from one Web Cluster to another according to the pre-defined threshold for scaling. All such events are completed by interoperation in following components.

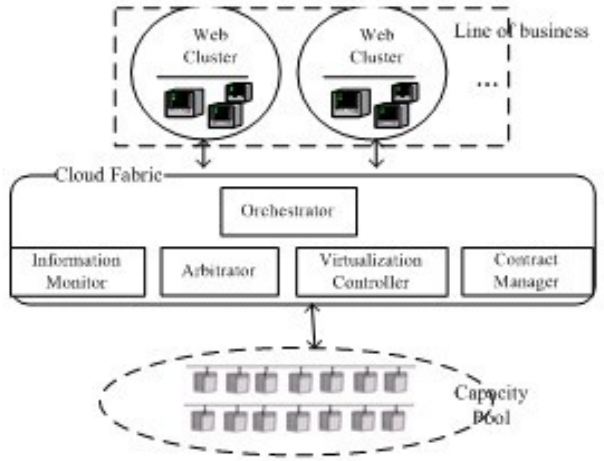

Fig.2 Provisioning Architecture

Contract Manager: It is capable of maintaining lifecycles of contract. When a new Contract is applied, Contract Manager will verify if there are enough capacities available or not, through interaction with Arbitrator.

Arbitrator: It is responsible for scheduling and allocating capacities to contract if reservation is placed in success, according to statistics on capacity's reservation. Moreover Arbitrator is able to repurpose capacities from one reservation to another according to consumption and restrictions in reservation.

Information Monitor: It dedicates to acquire information with respect to capacity consumption, health, hypervisor's information, contract's information, Web Cluster's workload, etc., among entire Cloud at runtime.

Virtualization Controller: It is used for VM lifecycle management, with facilities to instantiate, start, suspend, resume, stop and remove VM instance, through interaction with hypervisor.

Orchestrator: It is able to cooperate with Arbitrator for repurposing capacities, according to combined information from Information Monitor and Contract Manager. And it is capable of provisioning capacities for hosting Web Cluster by adoption of Advance Reservation with Lend-Borrow policy presented as following.

C. *Advance Reservation with Lend-Borrow Policy*

This section describes a higher flexible Advance Reservation with Lend-Borrow policy which allows capacities in reservation can be repurposed from one to

another under a set of restrictions instead of owned by one. It not only allows reservation to acquire more capacities from partner in case of quota exceeding, but also allows capacities in reservation can be repurposed from low-utilized one to other who is imperative, so as to prevent reservation from conflict and wasteness.

Novel expansion to this achievement is inspired by the vision of human behaviors, which is to lend unnecessary resources to other people needed, and to borrow more owned by another people who has no requirement and willing to share. We named it as Lend-Borrow policy, which existed widely in our live, e.g., business to business, person to person, etc. Accordingly Fig.3 illustrates the division of reserved capacities into two portions upon the vision preliminarily, one is ***owned*** which indicates the least capacities consumer need to hold even if there is non-consumption or low-utilization, and the rest is ***flexed*** which indicates that, there are quite a few capacities are supposed to cooperate for diversity consumer entities in cloud ecosystem for the event of overload or waste. Similarly we can treat the Cloud Ecosystem as enterprise which has a number of Web Cluster need to be hosted for line of business, it can therefore hire capacities to deal with time-varying workload across multiple Web Clusters throughout entire enterprise against redundancy and wasteness.

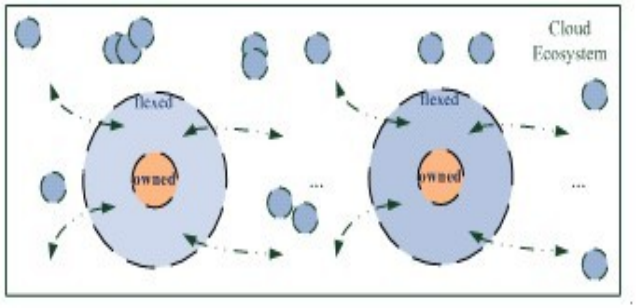

Fig.3 Consuming Activities living in Cloud ecosystem

The Lend-Borrow policy further defines that, reserved capacities are divided into shared and non-shared capacity for each portion, then Lend-Borrow policy functions at shared portion. Beside, Lend-Borrow policy prescribes the shared part with such restrictions, how many capacities are willing to lend or borrow, and who has permission to borrow and who can lend from, and the capable of reclaiming capacities which were lent out when need. We define the price of capacity with alpha P, so P_o means the cost on ***owned***, P_f means the cost on ***flexed***, and P_p means the capacity cost on the ***pay-per-used*** way. And the number of capacity consumption is defined with alpha N, so N_o dedicates the usage of ***owned*** part, the N_f is for ***flexed*** and the N_p is for ***pay-per-used***. And energy consumption with alpha E, available capacities with alpha C, E_o and C_o for ***owned*** part, E_f and C_f for ***flexed***, E_p and C_p for ***pay-per-used***. Moreover C_{ob} and C_{ol} are for capacities borrowed and lent from ***owned*** part, C_{fb} and C_{fl} are for ***flexed*** part. So benefiting from Lend-Borrow policy enabling, the cost for each consumer to obtain capacities for hosting Web Cluster and the actual capacities utilized as below.

$$\begin{aligned} Cost &= \sum P_o + \sum P_f + N_p \times P_p \\ Energy &= N_o \times E_o + N_f \times E_f + N_p \times E_p \end{aligned} \quad (1)$$

$$\begin{aligned} C_{\max} &= C_o + C_f + C_{ob} + C_{fb} \\ C_{\min} &= C_o - C_{ol} + C_f - C_{fl} \end{aligned} \quad (2)$$

Typically we assume that for pricing, cost for ***owned*** and ***flexed*** is fixed to bill, but the cost for ***pay_per_used*** is dynamical on demand. According to (1), consumer can reduce monetary cost and energy usage by reducing capacity consumption from ***pay_per_used*** part through (2) which shows consumer is able to exploit borrowed capacities in overloading. Moreover, on the provider side, there is not concern about capacities wasted in inappropriate reservation anymore and it's outstanding to lower down the energy consumption by reduction of ***pay_per_used*** capacity, as a result of repurposing capacities with the facility to balance multi-consumer with Lend-Borrow policy.

According to above, the novel Advance Reservation is proposed to extend with following constraints defined except general capacity's specification, which includes the type of capacity and the number of capacity, duration, etc.:

1) ***Component Definition:*** It defines reservation with a couple of components for multiple Web Clusters, one component per Web Cluster, so as to simplify reservation's maintaining. Moreover it can significantly function at balancing capacities across multiple Web Clusters inside one consumer entity, against preventing economic complexity from share-usage of capacities.
2) ***Owned Capacity:*** It defines how many reserved capacities are belong to ***owned***, and uses an element <lend/> with properties "to" and "slots" to limit who has permission to borrow and how many is granted to. All reserved in ***Owned*** part are utilized exclusively by default, if no <lend/> specified.
3) ***Flexed Capacity:*** It defines how many reserved capacities are belong to ***Flexed***, and uses element <lend/> which is resemble to the definition on Owned Capacity. All reserved in ***Flexed*** part are utilized exclusively by default, if no <lend/> specified.
4) ***Borrow tag:*** It's used for marking reservation as borrowable, and uses an element <borrow/> with properties "from" and "slots" to describe who is lender and how many capacities are expected. Acquired capacities are most probable less than expectance, however, when lender's grant is lower and available grant is not enough. For this aspect, the borrower will acquire all available and suitable capacities in grant. It's unable to borrow capacities by default, if no <borrow/> specified.
5) ***Pay_per_used tag:*** it's used for marking reservation with capability to satisfy more expectation temporarily over the amount of reservation and borrowed. And it's able to limit the cost for hiring capacities with a specified number in properties field. It's unable to use ***pay_per_used*** capacities by default, if no <pay_per_used/> specified.
6) Finally it's limited that, specifications 2 – 5 are only embodied in specification 1 for each component, which should define them respectively towards capacity cooperation for purpose.

D. Autonomic Provisioning mechanism

Above Advance Reservation with Lend-Borrow policy

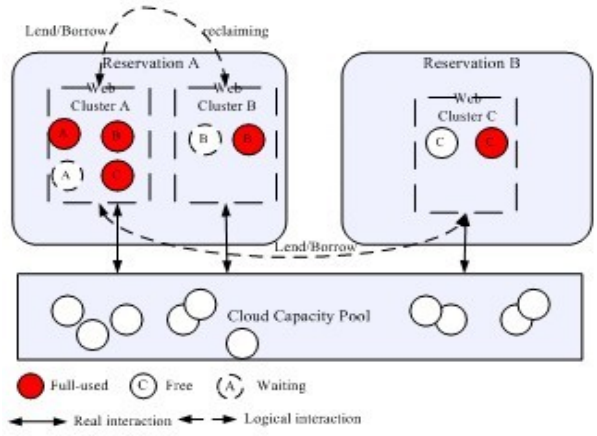

Fig.4 Interaction for repurposing Web Clusters

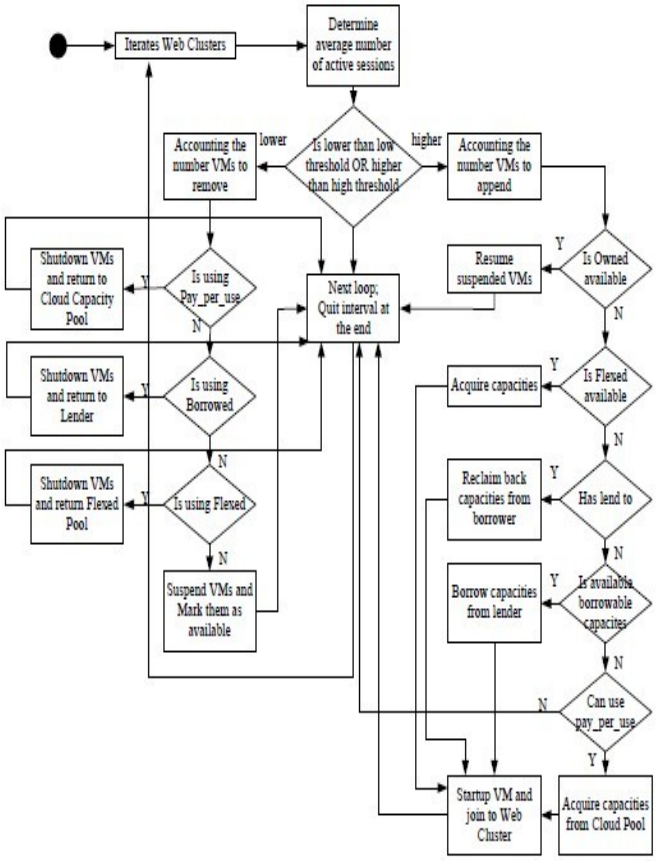

Fig.5 Autonomic Provisioning Mechanism

```
Sampling all Web Cluster's info
For an instance in Ncluster
  If (Nsessions < Tlow) then
    Calculate Nremove
      If (Nremove <= Npay_per_used) then
        Shutdown Nremove VMs, setValue and skip
      else
        shutdown Nremove-Npay_per_used VMs and setValue
      If (Nremove <= Nflexed_borrow_used) then
        shutdown Nremove VMs, setValue and skip
      else
        shutdown Nremove-Nflexed_borrow_used VMs and
setValue
      If (Nremove <= Nflexed_used) then
        shutdown Nremove VMs, setValue and skip
      else
        shutdown Nremove-Nflexed_used VMs, setValue
      If (Nremove <= Nowned_used) then
        suspend Nremove VMs, setValue and skip
  If (Nsessions >= Thigh) then
      Calculate Nexceed
      If(Nexceed <= Nowned_avail) then
        resume instances=Nexceed, setValue and skip
      else
        resume instances=Nexceed-Nowned_avail, setValue
      If (Nexceed <= Nflexed_avail) then
        Instantiate instances=Nexceed, setValue and skip
      else
        instantiate instances=Nexceed-Nflexed_avail, setValue
      If (Nexceed <= Nflexed_lent) then
          find Cborrower and reclaim Nexceed, setValue and skip
      else
        find Cborrower reclaim Nexceed-Nflex_lent and set Value
      If (Nexceed <= Nflexed_borrow_avail) then
        find Clender and borrow Nexceed, setValue and skip
      else
        find Clender and borrow Nexceed-Nflex_borrow_avail,
setValue
      If (Nexceed != 0 and it's able to use pay-per-used capacties)
        Instantiate instances=Nexceed
```

where

setValue means re-set related value
skip means skip to next cluster
Tlow denotes low threshold
Thigh denotes high threshold
Nsessions means average active sessions belongs to this cluster
Nexceed means the number of VM instance need append to this cluster
Nowned_avail=**Nowned**-**Nowned_used**
Nflexed_avail=**Nflexed**-**Nflexed_used**
Nflexed_borrow_avail=**Nflexed**-**Nflexed_borrow_used**

Fig.6 Capacity Scheduling Algorithm for Autonomic Provisioning

enabling, makes great effort on the flexible capacities reservation in which, reserved capacities are able to be repurposed elastically on demand. An associated provisioning mechanism is required accordingly, however, so as to employ such reservation to balance and repurpose capacities for hosting Web Clusters cost-effectively in an energy-saving way.

Firstly, provisioning mechanism refers to the approach from [18] and [15], for repurposing capacities based on low-high threshold within reservation, and employs the average number of active sessions as the performance tuning index. Moreover capacities are divided into variety of slots, which is a unit composed with CPU and Memory and identified as 1 slot = (1 cpu, 256 memory) in here. And one Web Cluster's backend node instance is presented as an instantiation of encapsulated VM appliance, which is with specification of concrete slots and pre-deployment Web application. Finally we can scale up and down Web Cluster elastically on demand through balancing and adjusting the amount of instances, and further repurpose capacities, which are reserved in demonstrated Advance Reservation with Lend-Borrow policy, from one Web Cluster with low demand to the desired one. Fig.4 illustrates the interaction.

Initially capacities in Owned definition will be totally instantiated as VM instances and be started, then joined into the Web Cluster as backend nodes. Whereas capacities in Flexed definition will not be instantiated, they are just able to be instantiated for the purpose of satisfying more demand over the capability of Owned capacities. Then after during each time interval, Orchestrator will sample active VMs for

obtaining the number of active sessions belong to respective Web Cluster from Information Monitor, such that it is capable of provisioning capacities adaptively according to the sampling statistics and the pending expectations. Fig.5 describes the autonomic provisioning mechanism for adjusting capacities among Web Clusters.

Fig.6 demonstrates the capacity scheduling algorithm applied for adjusting capacities allocation with autonomic provisioning mechanism. It shows that, if the Web cluster's workload is under the scope of [low_threshold, high_threshold], capacities exploited by this cluster has no change unless the other clusters has repurposing event associated with him. Capacity releasing event will be trigger and related VM instances will be shutdown or suspended if the number of average active sessions is lower than the low threshold, and the release order is that, capacities hired from ***pay_per_used*** will be released at first, then is the borrowed capacities, capacities exploits from ***flexed*** are the next, and the final portion to release is from ***owned*** and capacities from ***owned*** will be released by suspending VM instance instead of shutting down VM. For the cluster extending event occurs in case of the number of average active sessions is higher than high threshold, capacity pick order is that, to extending capacities in ***owned*** by resuming VM instances at first if there are available capacities in ***owned*** part, otherwise to try available capacities in ***flexed*** part, to reclaim back the capacities lent to other clusters if ***flexed*** capacities are not enough, then the next is to try to borrow more capacities from other clusters permitted to borrowed, finally hire more enough capacities from ***pay_per_used*** with permission to use.

IV. Experiment and Results

As above demonstration, our provisioning mechanism is innovative in drawdown cost and saving energy consumption from balanced and repurposed capacities in reservation by means of Lend-Borrow policy and threshold definition. It's encouraged to reduce consumption from ***pay_per_used*** through balancing to repurposing unused reserved capacities in sharable portion of ***Owned*** and ***Flexed***, in order to decrease extra cost over reservation and wasted energy consumed by idle capacities. Hence simulation in this paper just concern about validation in such that, whether presented structure and such provisioning mechanism can repurpose unused reservation across multiple Web Clusters, and they can whether make great deal with capacities in cost-effective and energy-saving way, apart from the view on performance.

Simulation environment exploits 3 Intel i5 3.2 Ghz machines as hypervisor, and each of them has 4 cores and 4G memory, denotes that there are 48 slots totally. And two hypervisors are both Xen-enabled redHat-server-5.4-x86_64 interoperated via 100M network, and are capable of delivering distinct VM instance from encapsulated VM appliance through Network File System (NFS). We pre-deploy and pre-configure Web Application into two VM appliances with using of virtualized capacities and virtualized network preliminarily, one requires 256M memory for hosting JBoss [25] Cluster and another requires 512M memory for hosting WebLogic [26] Cluster. Moreover we instantiate VM instance in LVM (Logical Volume Management) enabled way on behalf of speedy VM distribution and deployment benefited from snapshot in LVM. Finally we employ OpenNebula to interact with Xen hypervisor, and associate with Haizea which is an opensource VM-based lease management architecture, together for building Cloud fundamental environment.

We plan to verify our achievement with three Web Clusters consists of 2 JBoss Clusters and one WebLogic Cluster, and assume capacities are reserved respectively as following:

1) ***For JBoss Cluster A:*** 2 ***Owned*** instances reserved, 1 instance is lendable to WebLogic Cluster C. 2 ***Flexed*** instances reserved, 2 instances are lendable to WebLogic Cluster C, and it is also borrowable from WebLogic C with no restriction which means can borrow as many as possible. Moreover it's also able to use ***pay_per_used*** instances with no restriction.
2) ***For JBoss Cluster B:*** 2 ***Owned*** instances and 2 ***Flexed*** instances reserved, without using of our provisioning mechanism, but it's able to use ***pay_per_used*** instances with no restriction.
3) ***For WebLogic Cluster C:*** 2 ***Owned*** instances reserved, 1 instance is lendable to JBoss Cluster A. 2 ***Flexed*** instances reserved, 1 instance is lendable to JBoss Cluster A, and it is also borrowable from JBoss Cluster A with no restriction which means can borrow as many as possible. Moreover it's also able to use ***pay_per_used*** instances with no restriction.

Finally we define the low threshold at 30 and high threshold at 60 for the simplified performance index - average number of active sessions, afterward we use a test toolkit for Web Cluster's workload generation, named Mercury LoadRunner [28], to generate or remove access sessions periodically on three prepared Web Clusters respectively, so as to survey our proposed provisioning mechanism.

Illustration in Fig.7 shows that, the Owned capacities will be tuned to suspend status for saving energy if there is no consumption requirement from itself or borrower, and the lendable capacities will be repurposed to desired borrower on behalf of guard against using ***pay_per_used*** capacities so as to drawdown extra cost over reservation. Be note that, As we don't repurpose in-used capacities towards preventing vibration from over-repurposing in short time, if there are ***pay_per_used*** capacities being in used and workload is between the scale of [low threshold, high threshold], the consumption in ***pay_per_used*** will be kept even if available capacities are given back by borrower or has borrowable capacities again. Another remind is that, per instance for WebLogic Cluster C requires 512M memory that is double capacities to JBoss Cluster's instance and denotes WebLogic Cluster C can only borrow capacities from JBoss Cluster A in case of 2 unused slots available. Finally Figure 5 resulted that, there are 9 times energy-saving repurposing accomplished and 8 times cost drawdown in ***pay_per_used*** repurposing achieved.

The same workload generation script was applied for

testing Web Clusters cooperated without our approach for provisioning capacities. Comparing to Fig.7, Fig.8 illustrated that, JBoss Cluster B still consumed energy in lower workload, and capacity's consumption in pay_per_use were 5 times more than WebLogic Cluster which leveraged our mechanism for provisioning capacities. Moreover JBoss Cluster A had to use 5 capacities in pay_per_use more than first simulation.

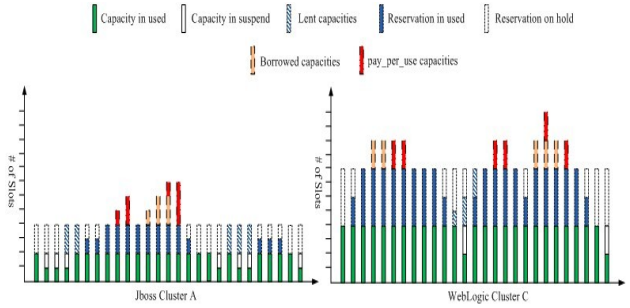

Fig.7 JBoss Cluster A cooperate with WebLogic Cluster C

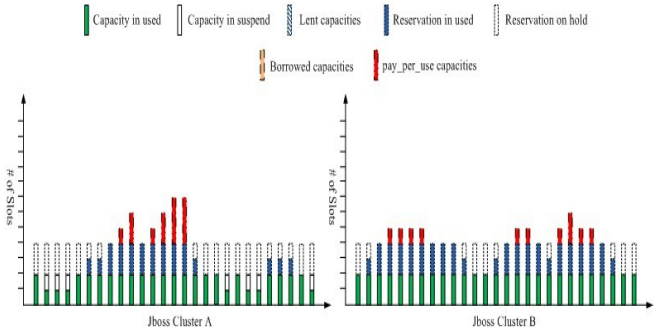

Fig.8 JBoss Cluster A cooperate with JBoss Cluster B

V. Conclusion

Paper is innovatively contributed on provisioning capacities elastically for hosting Web Cluster for doing e-Business towards overcoming the insufficiency in making up-front capacity plan for the time-varying workload, which will cause cost increased and energy consumption aggravated in capacity's redundancy and wasted, or lead into insufficient capacities for heavy workload. Particularly the proposed structure and provision mechanism, leveraging Advance Reservation with Lend-Borrow policy, are capable of balancing and repurposing capacities efficiently across multiple Web Clusters which reserve a number of capacities in sharing. And simulation results demonstrated that, with this achievement, Web Cluster is able to scale up or down elastically on demand, moreover the idle and unused capacities in reservation are able to repurpose to another desired Web Cluster dynamically, and Web Cluster is also capable of borrowing capacities from others which has lendable capacities in available. Furthermore relied on the facility from our creativeness, we can reduce over-reserve capacities for the rare peak which will result in another reservation failed and more energy consumption, with the help of capability to borrow capacities from others instead of hiring from pay_per_use in peak situation, so as to drawdown extra cost and save energy among the whole Cloud enabled environment.

References

[1] T. Schroeder, S. Goddard, B. Ramamurthy, "Scalable Web server clustering technologies" IEEE Network, May-June 2000, pp. 38-45.

[2] V. Cardellini, M. Colajanni, P. Yu, "Dynamic Load Balancing on Web-Server Systems" IEEE Internet Computing, Vol. 3, No. 3, May-June 1999, pp. 28-39.

[3] M. Armhurst et al., "Above the Clouds: A Berkeley View of Cloud Computing" Comm. ACM, vol. 53, no. 4, Apr. 2010, pp. 50-58.

[4] R. Buyya, Y. S. Chee, V. Srikumar, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities" Department of Computer Science and Software Engineering, University of Melbourne, Australia, July 2008, pp. 9.

[5] Amazon Web Services. Amazon.com, Inc. [Online]. Retreived October 7, 2010, from: http://aws.amazon.com/ec2/

[6] Amazon CloudWatch. Amazon, Inc. [Online]. Retreived October 7, 2010, from: http://aws.amazon.com/cloudwatch/

[7] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds" IEEE Internet Computing, vol. 13, no. 5, pp. 14–22, 2009.

[8] D. Nurmi, et al., "Eucalyptus: an open-source cloud computing infrastructure", Journal of Physics: Conference Series, vol. 180, 2009.

[9] m. klems, S. Tai, L. Shwartz, G. Grabarnik, "Automating the Delivery of IT Service Continuity Management through Cloud Service Orchestration" Network Operations and Management Symposium (NOMS), April 2010, pp. 65-72.

[10] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "Enacloud:An energy-saving application live placement approach for cloud computing environments," IEEE International Conference on Cloud Computing. IEEE Computer Society, 2009, pp. 17–24.

[11] P. Marshall, K. Keahey, T. Freeman, "Elastic Site:Using Clouds to Elastically Extend Site Resources" Proc. IEEE/ACM International Conference. Cluster, Cloud and Grid Computing(CCGrid 2010), IEEE Press, May, 2010, pp. 43-52, doi: 10.1109/CCGRID.2010.80.

[12] L. Li, "An Optimistic Differentiated Service Job Scheduling System for Cloud Computing An Optimistic Differentiated Service Job Scheduling System for Cloud Computing" Third International Conference on Multimedia and Ubiquitous Engineering, June 2009, pp. 295-299.

[13] Y. Chen, T. Wu, J. Li, "An Efficient Resource Management System for On-line Virtual Cluster Provision" IEEE International Conference on Cloud Computing, 2009, pp. 72-29.

[14] Z. Duan, Zh. Gu, "EWPCC:An Elastic Web Proxy Cache Cluster Basing on Cloud Computing" IEEE International Conference on Computer Science and Information Technology, July 2010, pp. 85-88.

[15] T. C. Chieu, A. Mohindra, A. A. Karve and A. Segal, "Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment", Proceedings of the IEEE International Conference on e-Business Engineering (ICEBE 2009), Macau, China, Oct. 2009, pp. 281-286.

[16] W. Iqbal, M. N. Dailey, D. Carrera, "SLA-Driven Dynamic Resource Management for Multi-tier Web Applications in a Cloud" Proc. IEEE/ACM International Conference. Cluster, Cloud and Grid Computing(CCGrid 2010), IEEE Press, May, 2010, pp. 832-837, doi: 10.1109/CCGRID.2010.59.

[17] R. Lopes, F. Brasileiro, P.D. Maciel, "Business-Driven Capacity Planning of a Cloud-based IT Infrastructure for the Execution of Web Applications" Proc. IEEE Symp. Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW 2010), IEEE Press, April, 2010, pp. 1-8, doi: 10.1109/IPDPSW.2010.5470726.

[18] P. Barham et al., "Xen and the Art of Virtualization" Proc. 19th ACM Symp. Operating Systems Principles, ACM Press, 2003, pp. 164-177.

[19] VMware. Inc. [Online]. Retreived October 7, 2010, from: http://www.vmware.com/

[20] KVM. Inc. [Online]. Retreived October 7, 2010, from: http://www.linux-kvm.org/

[21] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, and A. Roy, "A Distributed Resource Management Architecture That Supports Advance Reservations and CO-Allocation," in International Workshop on Quality of Service, June 1999, pp. 27-36.

[22] S. Venugopal, X. Chu, and R. Buyya, "A negotiation mechanism for advance resource reservation using the alternate offers protocol" Proc. 16th International Workshop. Quality of Service(IWQoS 2008), June, 2008, pages 40--49, doi: 10.1109/IWQOS.2008.10.

[23] Yuan Wencheng, Zhu Yian. "Novel Measures for More Efficiently Utilizing Web Cluster" Journal of northwestern polytechnical university, China, 2010, vol. 28, no. 1, pp. 72~76.

[24] Yuan Wencheng, Zhu Yian. Exploring Virtualized Resource Management Mechanism for Cloud Computing. Journal of northwestern polytechnical university, China. 2010, vol. 28, no. 5, pp. 704~708.
[25] JBoss. Inc. [Online]. Retreived October 7, 2010, from: http://www.jboss.com
[26] BEA Weblogic. Inc. [Online]. Retreived October 7, 2010, from: http://www.oracle.com/us/products/middleware/application-server/index.html
[27] Haizea. [Online]. Retreived October 7, 2010, from: http://haizea.cs.uchicago.edu/
[28] HP LoadRunner. Inc. [Online]. Retreived October 7, 2010, from: https://h10078.www1.hp.com/cda/hpms/display/main/hpms_content.jsp?zn=bto&cp=1-11-126-17%5E8_4000_100

Yuan Wencheng received BSc(2003) and MSc(2007) from the department of computer science and engineering, Northwestern Polytechnical University (NPU), Xi'an, Shaanxi, PRC. And now He is a PhD candidate of NPU in computer science and engineering with a specialization in Distribution Computing, particularly in Grid Computing and Cloud Computing. He has experience on Grid and Cloud product development in Platform Computing Corp. from 2005 to 2010, focus on resource management includes architecture and allocating and scheduling policy design. Contact him via whimtn@gmail.com

Zhu Yian is professor and PhD supervisor of computer school, and is Vice dean of the school of software and electronics, Northwestern Polytechnical University, Xi'an, Shaanxi, PRC. His research interests include Parallel Computing, Embedded Computing, Software Engineering, Mobile Computing, Grid Computing and Cloud Computing. He received the scientific and technical progress award for "Study of Parallel Processing Technology Oriented Special Applications" is sued by Committee of National Education of China, and the scientific and technical progress award for "EP--860 Parallel Computer System" issued by Aviation Industry Corporation of China, and "The man with outstanding achievement" awarded by Aviation Industry Corporation of China, Contact him via zhuya@nwpu.edu.cn