**What is the goal?**

1. **Minimize the cost** of running a business using an **IaaS/cloud platform** by automatically allocating and **deallocating cloud server instances** to and from a cloud server network as the ratio of resource capacity vs resource demand fluctuates compared to a calculated/configurable threshold.

**Field**

The invention is generally directed toward cloud server network monitoring and management systems, and more *particularly, allocating* and deallocating cloud server instances to and from a cloud server network according to a cloud server balancing and consolidation algorithm which continuously monitors the loads on the cloud servers and deploys or ***releases*** cloud server instances according to the distribution of workload on server resources across a cloud server network.

**Background**

Conventional [load balancers](https://www.google.com/patents/WO1999032956A3) in computer systems “balance the load on a plurality of servers using a load balancing algorithm which continuously examines the loads on the plurality of servers and makes adjustments in the loads accordingly. Among the factors considered in the load balancing are the power of each server relative to other servers, the load on each server relative to the other servers, and a ‘credit’ for each server based on their power and load.”

Conventional load balancing, as defined in the preceding, is helpful when applied to an internal server network, but companies that host their server instances with a cloud service provider are introduced to a new load balancing problem that conventional load balancers cannot address.

“[Cloud service providers offer programs](https://www.google.com/patents/WO2011041101A1), such as infrastructure as a service (IaaS), which use different pricing schemes when charging for use of cloud resources.” While a conventional load balancer may help ensure a company’s cloud server instances operate efficiently in a cloud server network, especially during times of peak usage, a conventional load balancer cannot minimize the cost of using a cloud server network because it balances the load among the cloud servers rather than assessing whether a new cloud server instance should be allocated or an old one deallocated.

*Comparable to the way conventional load balancers allocate or deallocate resource demand to or from a server instance, this cloud server network management system allocates and deallocates cloud server instances from the cloud server network*. However, resource demand must be allocated differently in this scenario to accommodate efficient allocation and deallocation of cloud server instances.

Allocation and deallocation of resource demand, for example, could be completed using strategies similar to those found in the patents below:

**bin packing**

[https://www.google.com/patents/US20130235416](https://www.google.com/patents/US20130235416?dq=bin+packing&hl=en&sa=X&ei=IoHXUtz-D8HtoATOhoLIDA&sqi=2&pjf=1&ved=0CEAQ6AEwAQ)

“Cloud computing services, such as production services, offer customers many benefits including reduced capital expenditures, scalability, managed upgrades and access to new workflows. Often, Infrastructure-as-a-Service cloud systems rent the use of virtual machines by the hour. Production jobs typically have an associated deadline by which the production job needs to be completed. Accordingly, it is important to schedule production jobs efficiently in order to maximize the use of a virtual machine during a rental period and therefore minimize the production cost.”

**scheduler**

<https://www.google.com/patents/US8429659>

(this patent is also really good for teminology)

a method for prioritizing jobs within a cloud computing environment, comprising: determining a processing capacity of the cloud computing environment; determining an income, a processing need, a set of service level agreement (SLA) terms for each of a plurality of jobs; and prioritizing the plurality of jobs based on their respective income, processing need, and SLA terms, wherein the prioritizing ensures that the set of SLA terms are met while maximizing the income and without exceeding the processing capacity.

**NOTE: This patent app, written in Chinese, is very similar to this patent** <https://www.google.com/patents/CN102495759A>

[Because](http://www.google.com/patents/US20110264748) resource demand often changes, over or under utilization of these resources can be an issue. For example, as a cloud server instance process progresses, resource demand may decrease with the amount of tasks it has left to process. Conversely, resource demand may increase as new tasks are requested. Because conventional systems do not dynamically adapt to changing resource requirements, cumbersome manual intervention may be required to allocate new cloud server instances and increase capacity, or deallocate assigned cloud server instances and reclaim unused resources.

The pdf diagram shows a high level flowchart of how this idea will work, but processes will need to be described at the server level too.

**Schedule tasks/resource demand/workload**

I don’t think I need to describe this too thoroughly, but I will note that it is very important that tasks be assigned to cloud server instances in such a way that each cloud server instance has minimum surplus resources.

Although tasks can be assigned in such a way that each cloud server instance has minimum surplus resources, tasks will be completed at different rates and servers will inevitably be left with disparate amounts of surplus resources ranging from 0% to 100%.

**continuously calculate distribution of tasks/resource demand/workload**

Since the cloud server instances will inevitably have different resource demands on them as processes progress, it will be necessary to continuously monitor each server instance to make sure they are operating at maximum capacity.

**consolidate resource demand when possible**

If cloud server instances are found to not be fully utilized, then measures must be taken to maximize resource demand on cloud server instances. This will usually mean moving tasks from an under-utilized cloud server instance to a cloud server instance that has more tasks to process. However, by moving tasks away from the cloud server instance, the cloud server instance is even more underutilized. When this happens, the system must determine if it is cost-effective to deallocate the underutilized cloud server instance. If it is cost-effective, then the cloud server instance needs to be configured to refuse all future tasks from the cloud scheduler (this is the first time I’ve used this term, I should probably go back and give the scheduling step a name for a part of the system) so it can be deallocated without the risk of negatively impacting the processing of tasks in the cloud server network.

It is important to note that it may not always be feasible to deallocate a cloud server instance, even when it’s underutilized. For example, tracking historical peak demand for a company’s cloud network may reveal that resource demand will likely increase before the end of the hour. If this is the case, then it may be more cost-effective to extend the contract to run the cloud server instance instead of deallocating it because allocating another cloud server instance to meet the future increased resource demand will cost more money than would be saved by deallocating the underutilized server.

This system will take these scenarios into account as it manages the cloud server network. In the case described above, the system may need to take into account the pending increase in resource demand and stop consolidating resource demand on the cloud server network in order to more efficiently process jobs. That way the underutilized cloud server instance can still process tasks.

**When there are surplus computing resources in the cloud server network**

If there are surplus resources, then the system should calculate the cost of deallocating as cloud server instance. Depending on the resource cost and savings associated with deallocatng a cloud server instance, the system may or may not deallocate the cloud server instance. There are a lot of factors to take into account:

1. Time needed to deallocate a cloud server instance
2. Time needed to allocate a new cloud server instance when one is needed
3. Cost of deallocating a cloud server instance
   1. if a cloud server instance is bid to run for 8 hours and has only ran for 6 hours, how much money is lost?
   2. if a cloud server instance is deallocated, how many jobs will not be processed during the cloud server instance’s contract that otherwise could have
4. Cost of allocating a new cloud server instance
   1. Bidding for new cloud server instances costs money

If the underutilized cloud server instance is not deallocated, then the system will continue monitoring and consolidating resource demand as cost-effectively as possible.

How to achieve this?

1. Calculate resource demand on each cloud server instance
2. Keep track of resource demand history
3. Calculate

**Calculate Allocation/De-allocation of Cloud Server Instance**

Can I use this? <http://technet.microsoft.com/en-us/library/cc181325.aspx>

We will have functions called:  
double getServerLoadCapacity()  
double getServerLoad()  
double getServerLoad