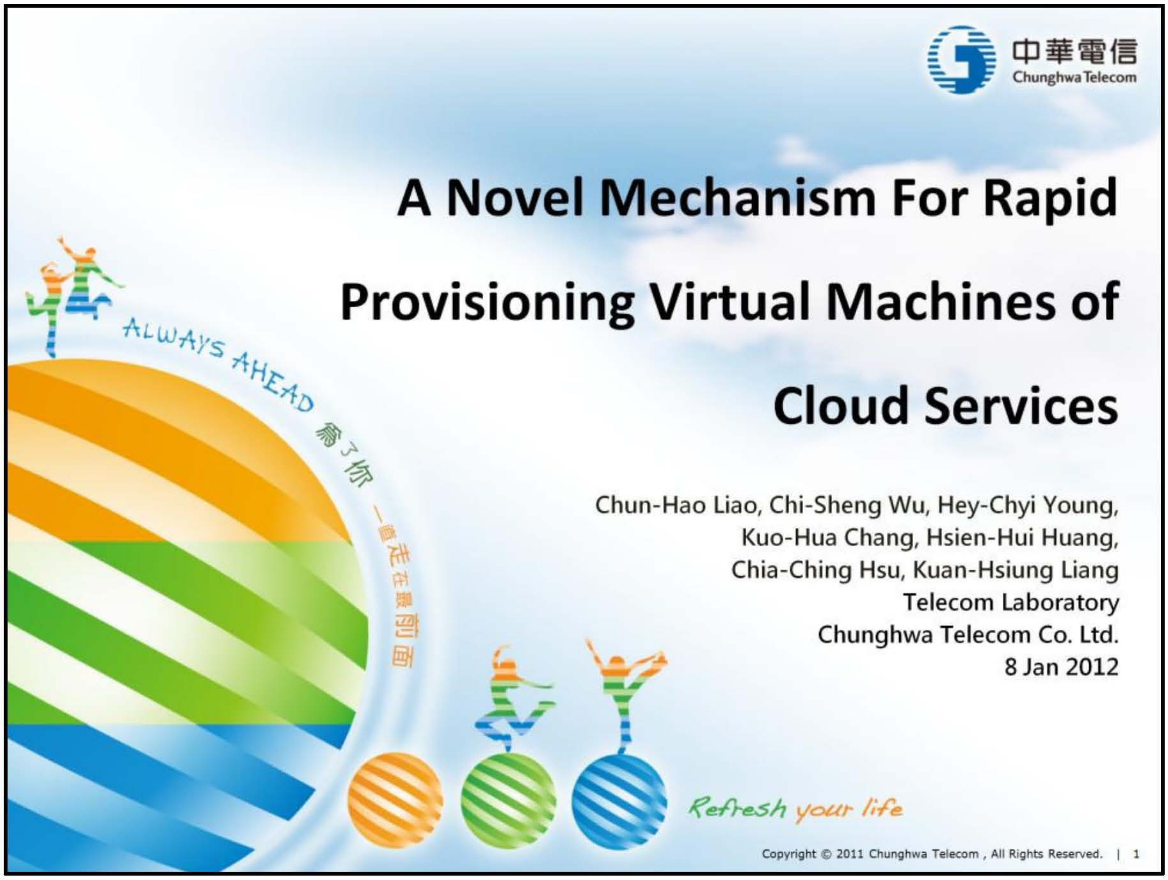


Abstract — Cost down and energy saving are concerned issues in recent years. As a result, cloud computing services for enterprises and persons have been increasing. As for cloud Infrastructure-as-a Service (IaaS), virtual machines (VM) in "cloud" are also starting to replace the original local hosts commonly. Meeting consumers' IaaS requirements with different computing powers, different operating systems, and different quantities in a fast and on-demand fashion has become an important issue for cloud service providers. For Chunghwa Telecom's IaaS service, branded "hicloud", we propose a novel mechanism for rapid provisioning a variety of virtual machines. We illustrate the end-to-end IaaS fulfillment process, kernel VM Pool Management and multi-threading VM activation mechanism for fast VM provisioning in this paper in detail.

Keywords-component; Cloud Computing, IaaS, Virtual Machine, Provision, VM Pool Management, hicloud CaaS

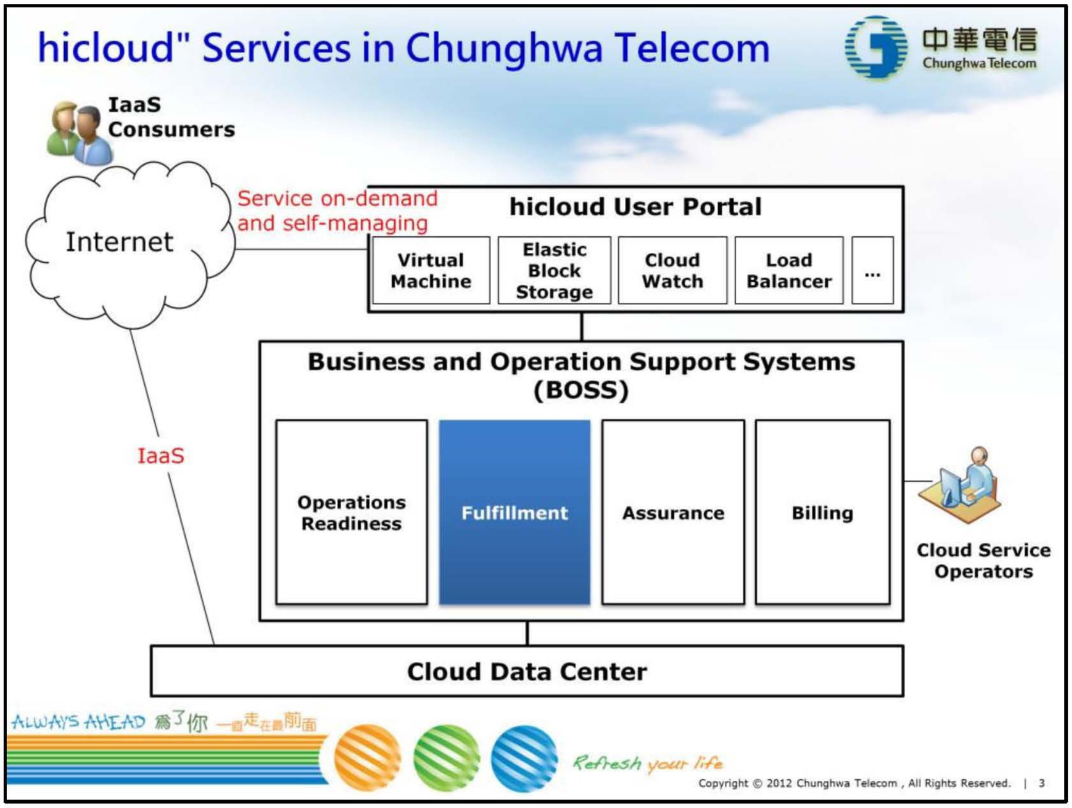
Agenda

- Introduction of hicloud Service in Chunghwa Telecom (CHT)
- Problems
- Rapid Provisioning System Architecture
- Experiment
- Evaluation
- Summary & Future Work
- References

ALWAYS AHEAD 爲了你 一直走在最前面

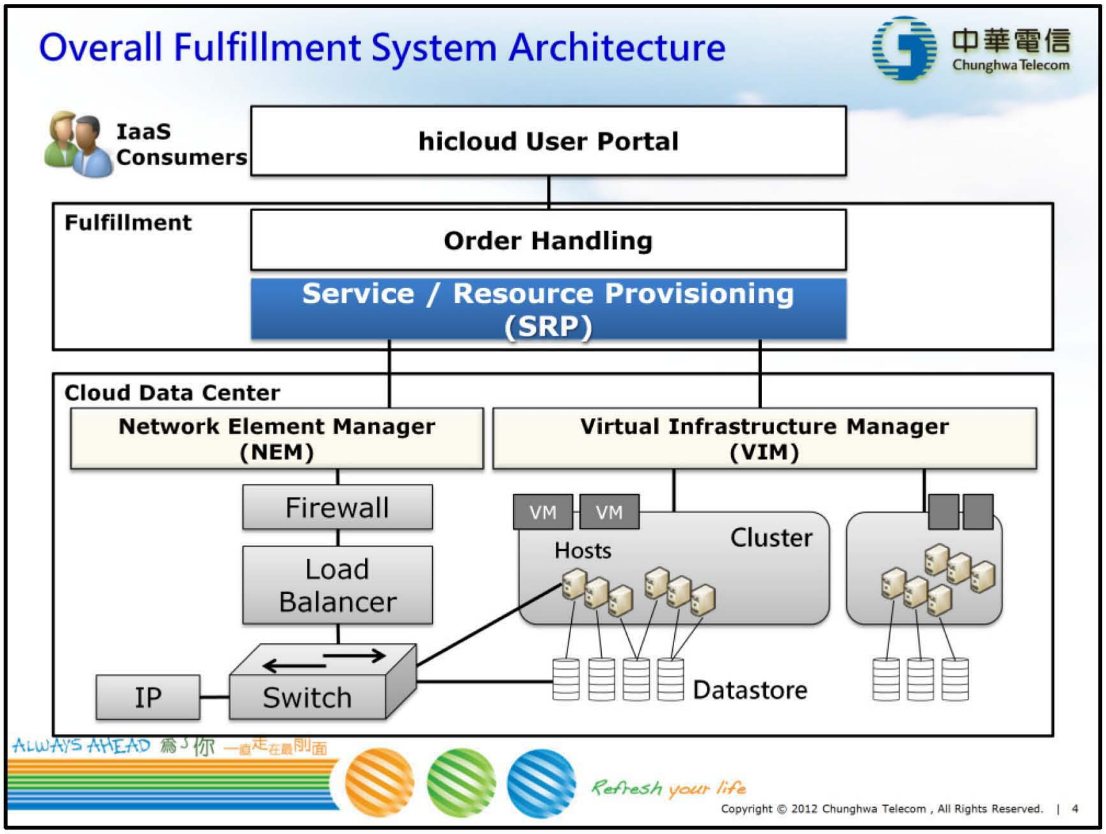
Refresh your life

 | 2



Chunghwa Telecom hicloud is a cloud service provider which has been serving the public and enterprises since 2009 [1] [2]. It consists of service portal, business and operation support systems and cloud data center. Pubic or enterprise hicloud consumers can send orders from the service portal to acquire services such as virtual machine, elastic block store, cloud watch, server load balancer [3] and other IaaS [4] cloud services. Business and operation support systems(BOOS)[5] literally takes advantage of Business and Operations Support System to provide cloud services that meet cloud essential characteristics. Following new generation operations systems and software (NGOSS) eTOM framework [6], Chunghwa's Cloud BOSS supports several important end-to-end business / operations processes including operations readiness, service fulfillment, assurance and billing (FAB). We will focus on the fulfillment of Chunghwa Telecom's hicloud computing as a service (CaaS) which is catalogued as IaaS.

In this page, the fulfillment functions of Cloud BOSS will then automatically provision the virtual machines from a shared pool of computing resources for the consumers. With the access information to the virtual machines sent by BOSS, the consumers can then use the virtual machines in the cloud for their computing purpose. And cloud data center is a kind of data center which has virtual computing, storage and network resource. In order to meet consumer's requirement, the fulfillment of BOSS obtains these resources which are defined by different specific services.



For more details, we has illustrated the end to end fulfillment process of CaaS service in this page. Cloud Service Portal is the web user interface to accept consumers' on demand service request. The consumer's service requests are delivered to the order handling (OH) module which is the first process of the Fulfillment. In order handling module, requests are transferred into orders, which contain information of the purchased services, will be forwarded to service and resource provisioning (SRP) function. SRP allocates and assigns a set of logical resources corresponding to networking and computing infrastructure to construct the CaaS services. After that, SRP starts to invoke the internal provision process under planned control.

There are two important components in cloud data center - network element manager (NEM) and Virtual Infrastructure Managers (VIM). NEM manages network elements like switches, firewalls, server load balancer, DHCP, VLAN and IP to establish the path and policies of the networking for the service. SRP asks NEMs for network elements and also acquires computing and storage resources through VIM if it is required by specific services such as virtual machine (VM). VIM takes charge for managing clusters and datastores. One cluster which provides computing resources is a group of physical hosts having identical modules and computing performance. The physical storage server is divided into datastores that are virtualized by VIM. Datastores can store VM, image template of VM and used for EBS.

Based on this architecture, we are able to provide cloud services to consumers. These services include VM, which is a software implementation of a machine (i.e. a computer) that can execute programs like a physical machine with an operation system. When consumers request orders of VM with selected computing specifications (e.g., 2 CPU core with 4GB RAM and 100G disk size) from hicloud user portal, orders are delivered to OH to keep the customer consumption records and valid the qualifications. If the request order has been approved, OH sends the request to SRP and decides which procedure to be launched and which virtualized resources are to be assigned according to the orders. SRP sets firewall, DHCP, VLAN configurations and IP Address on NEM to get virtual resources and activate real functions. Not only from NEM, but SRP also gets cluster and datastores from VMM. After completing all required processes, SRP sends OH a message that all tasks are done. Then OH notifies hicloud user portal to let consumers know that provisioning is finished. Finally, consumers are able to directly utilize virtual machine in cloud data center.

Details of VM Provisioning Process

中華電信 Chunghwa Telecom

Process	OS selection in VM		Spent Time (sec.)
	Windows	Linux or other OS	
Clone VM from image template	√	√	660
Initialize	System Prepare	Others	~120
Rename VM	√	√	2
Change cluster	√	√	10
Change datastore	√	√	720
Set CPU & memory	√	√	2
Set network configuration: SLB, Firewall, DHCP	√	√	20

***Environmental Setting**: (1.) Host: Cisco N20-B6625-1 Intel Xeon X5670, 6 * 2.93GHz dual core(12 cores) 48GB ram. (2.) Bandwidth: Service Network = 1GEService Network = 1GE, Storage Network = 10GE. (3.)Image templates: Windows 2008 R2 64bit with root disk 100GB, CentOS 5.5 64bit with root disk 100GB.

Refresh your life

However, in fact, the CaaS customers utilize VMs for different purposes. Many professors or laboratories may require CaaS for complicated mathematical calculation and analysis at some serious research steps. They sometimes need a great number of VMs with various operation systems (OSs) to be provided as soon as possible. But it takes so much time to provision more VMs. In order to provision VMs with various computing needs and value-added service specifications, a variety of OSs like Windows, Linux, etc. with different computing capabilities like fast CPU, large memory have to be supported. In addition, some value-added services like firewall, extra IP addresses should also be considered.

To provision VMs simultaneously, the issues we have are:

- Slowing down of committed provisioning speed

  Provisioning many VMs at a same time might cause serious delay of delivering VM. For service level agreement, service providers must make their commitment of VM delivery time.
- Negative performance influence on existed VM

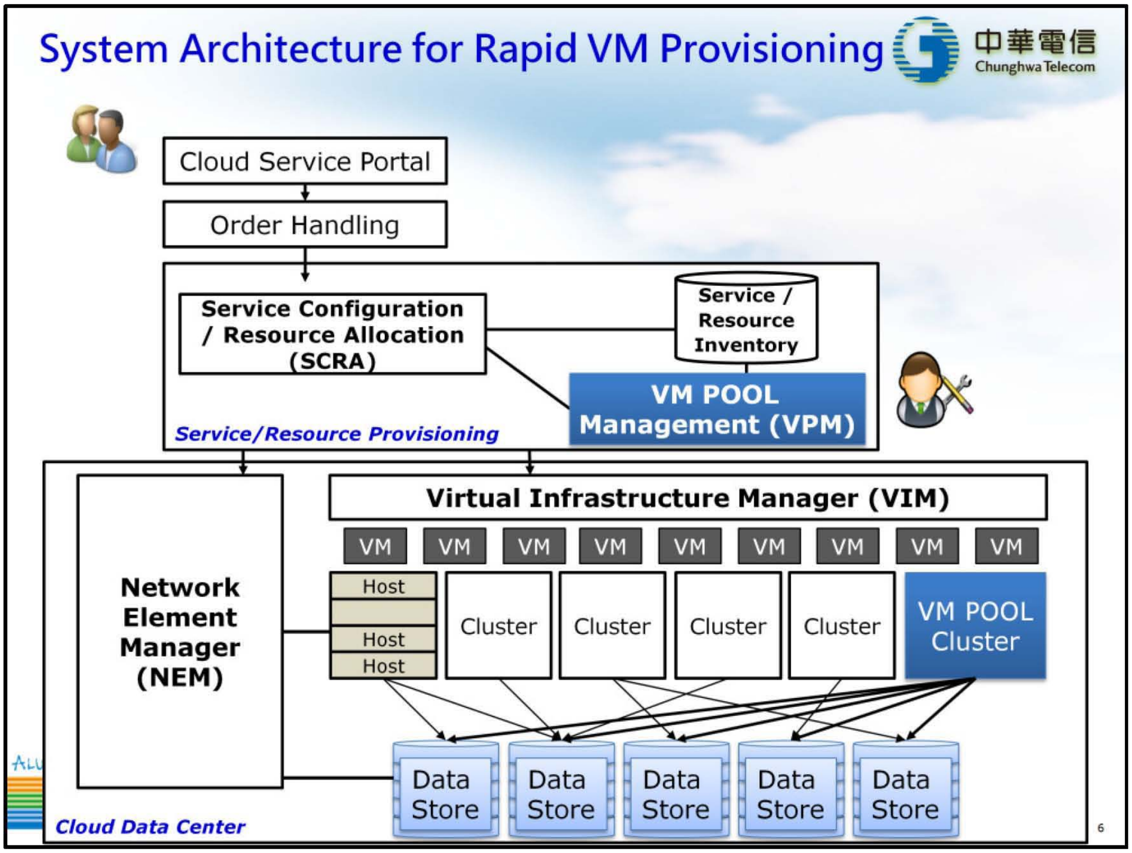
  Since VM needs computing and storage resources, if simultaneous multi-VM provisioning is not under proper control, the performance of already running VM may be degraded a lot due to resource competition.
- Failure of automatic provisioning

  For provisioning, the capacity of VMM may be limited. When running too many provisioning instructions and exceeding the VIM computing limits, automatic provisioning of VM may fail.

To solve these issues, we have to examine VM provisioning in detail. Generally speaking, the process of VM provisioning is showed in this page. The process has several required steps. The order of steps depends on different kinds of hypervisors in VIM. Cloning VM from template image means get another copy of primitive one called template image. Computers where customer can operate are virtualized as VMs in cloud service. For creating VMs, each template image, as data file of virtualized physical devices with OS, drivers and basic configuration, is a copy to create a generic VM in datastore. Initialization is an important step for VM provisioning. For example, system prepare is required by every Windows-based OS. To manage VMs, renaming VM is necessary to give each VM an id. If cloned VM is not in the specific cluster and datastore, it has to be moved to target ones. According to requests, CPU, memory and network configurations are set into cloned VM. Successful Provisioning VM requires whole steps without any problems.

After we do experiments on each step, we found some facts. Cloning VM from template, initialization and changing datastore steps cost over one minute but the rest of steps just cost few seconds. Compared with other steps, these three steps needs much more time to complete. Because cloning VM from Template and changing datastore involve file transmitting, they require considerable disk I/O and network bandwidth. Initialization sometimes does multiple times of retry which spend time on operating and waiting. Besides, considerable and simultaneous cloning causes exponentially increasing required time and raising the risk of cloning failure.

In fact, we, Chunghwa Telecom have been providing cloud services in Taiwan for a long while. These cloud services named hicloud have encountered several practical problems. To raise stability and quality of hicloud, we are trying to overcome these challenges. Some companies (e.g. IBM, Amazon) note these and have proposed a preliminary thought. Unfortunately, until now, we don't see any published paper about this problem or any practical methodologies. In the next chapter, we will propose our design based on our architecture.



Compared with the direct-cloning architecture [7], this page shows the architecture from overall perspective. There are many essential components playing different roles in the provisioning process. Initially, we interpret the VM on-demand service of a customer visiting web site on internet. The first one is cloud service portal where customers can purchase VM and use bunch of VM self-service functions (e.g., power on a VM, shutdown a VM and etc.). Second component is order handling module that is responsible for analyzing the customer orders with related service specifications (e.g., selected OS, quantities of VM and so on). OH takes care of the customer order lifecycle and reports the latest provisioning status to customers in real time on cloud service portal. Service configuration / resource allocation (SCRA) module checks service profile sent from order handling and start to allocate the appropriate and required resources (e.g., public IP address) from service and resource inventory (SRI). SRI is a place where information of all services and resources (e.g., IP address, firewall policy, etc. in cloud data center) are kept. When SCRA gets necessary information of resources from SRI by request orders, it will set up on NEM and VMM.

VM Pool management (VPM) is a new-designed technology for rapid provisioning VM. It is composed of VM pool configuration and management mechanism. VM pool is a logical inventory and controlled by specified management policy. VM pool stores cloned VMs without customization. These are called primary VMs. Before introducing VPM, we talk about the design of VPM. We think a good design of VPM must have 4 characteristics:

(1) Automated refilling and synchronization

VPM can refill primary VMs automatically when the current number is under lower bound. It can stop refilling if the quantity reaches upper bound. Besides, VPM needs to synchronize information from VM pool periodically to make sure that the correct information is always kept.

(2) Minimum overhead cost of system's performance

Refilling functionality should be allowed to reduce performance of system as less as possible. For this reason, we set up a specified cluster is only for VPM refilling and restrict the number of refilling instructions. That is also beneficial for operating and trouble shooting.

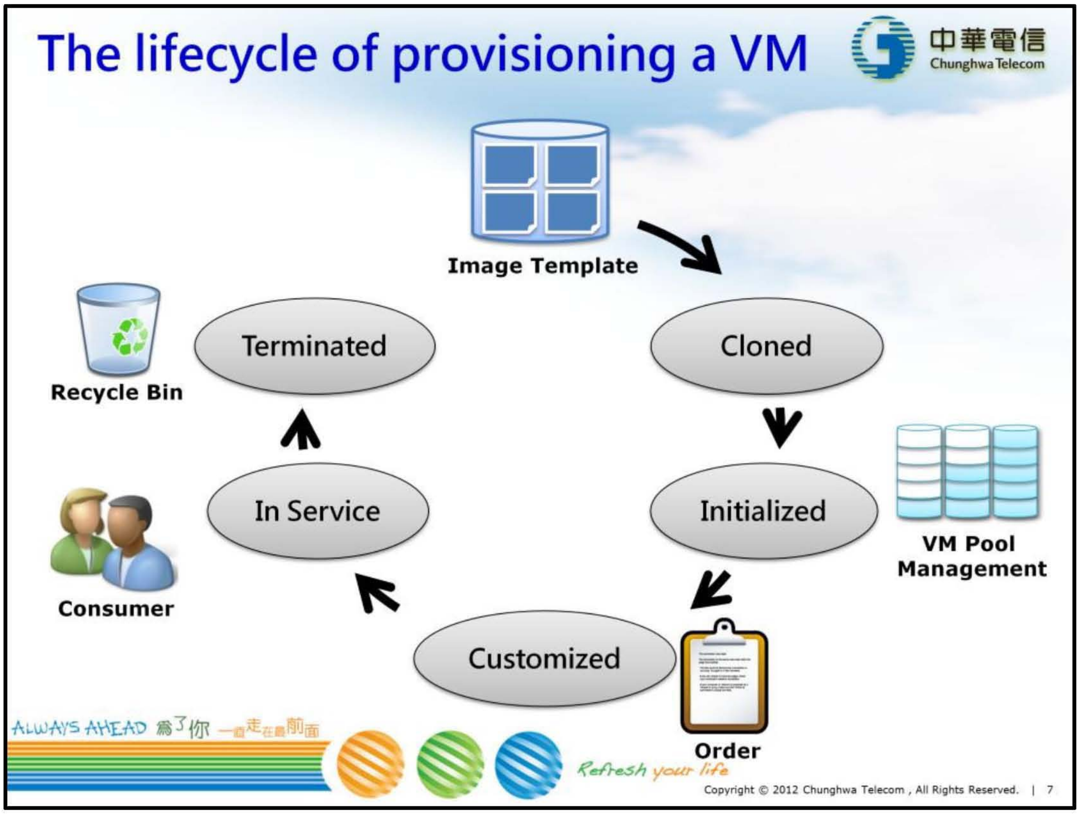
(3) Scheduling

Heavy loading of hypervisors makes low probability of successful cloning VM from template image step. Therefore, we need to schedule refilling tasks to make sure all primary VMs are able to be provisioned for high quality of VM service.

(4) Artificial manageability

VPM needs well-designed user interface to manage artificially. We provide a dynamical VM pool control for operators to configure the upper bound and lower bound reflecting business needs. Furthermore, supervising which OS's template image is refilling and which one is still waiting in schedule through control panel is an essential and attractive feature for operators.

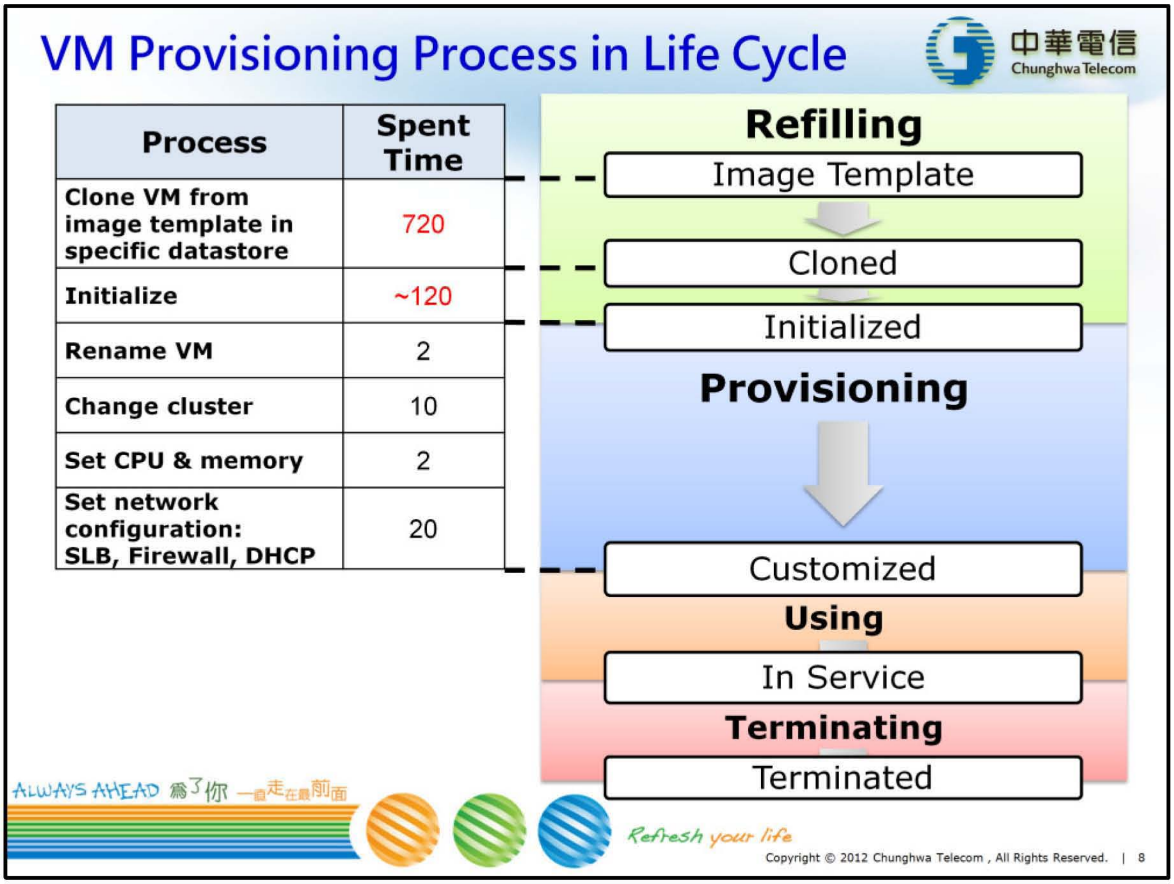
For these characteristics, we choose one cluster which takes charge for VM refilling. This cluster is linked with each datastore. That can refill primary VMs in every datastore to avoid file transmitting. Besides, consumers can utilize whole computing resources from other clusters rather than reduced performance by VM pool cluster. Refilling and synchronization are two fundamental functionalities in VPM. When refilling each unit of VM pool, VPM starts cloning VM from template, and initializes to generate primary VMs in the specified datastore which information are stored in SRI for VMs readiness. On synchronization process, VPM requests information for existed quantities for primary VMs in VM pool and that will always keep the correct number from underlying VIM to SRI. We can configure upper bound, lower bound and status of each unit of VPM. For example, we can limit maximum and minimum of primary VMs in each kind of OS and datastore. The status can be set to close VPM.



In the life cycle of provisioning a VM, there are five statuses for provisioning a VM. An image template is a resource containing an operation system with languages, bits, types, and disk size. It can be used as a virtual machine disk or a computer. This data can be then saved overwriting the original image. Image template status means that VMs haven't cloned and cannot be used by guests.

1. **Cloned** - VM in cloned status means VM is copied and instantiated from image templates and ready for initialization.
2. **Initialized** - Initialized status means VMs have been cloned from image template and initialized. As mentioned previously, we call these *primary VMs*. *Primary VMs* can be stored in VM pool and managed by VPM. VPM can start refilling mechanism which supplies more primary more initialized cloned VMs regularly.
3. **Customized** - In customized status, VMs have been configured by specifications (e.g., 4x core CPU, 4G RAM, Disk size 100GB) according to on-demand request by customers.
4. **In Service** - Consumers can utilize VMs by self-management through internet. The VM is in service status.
5. **Terminated** - Terminated status means VMs are going to be deleted.

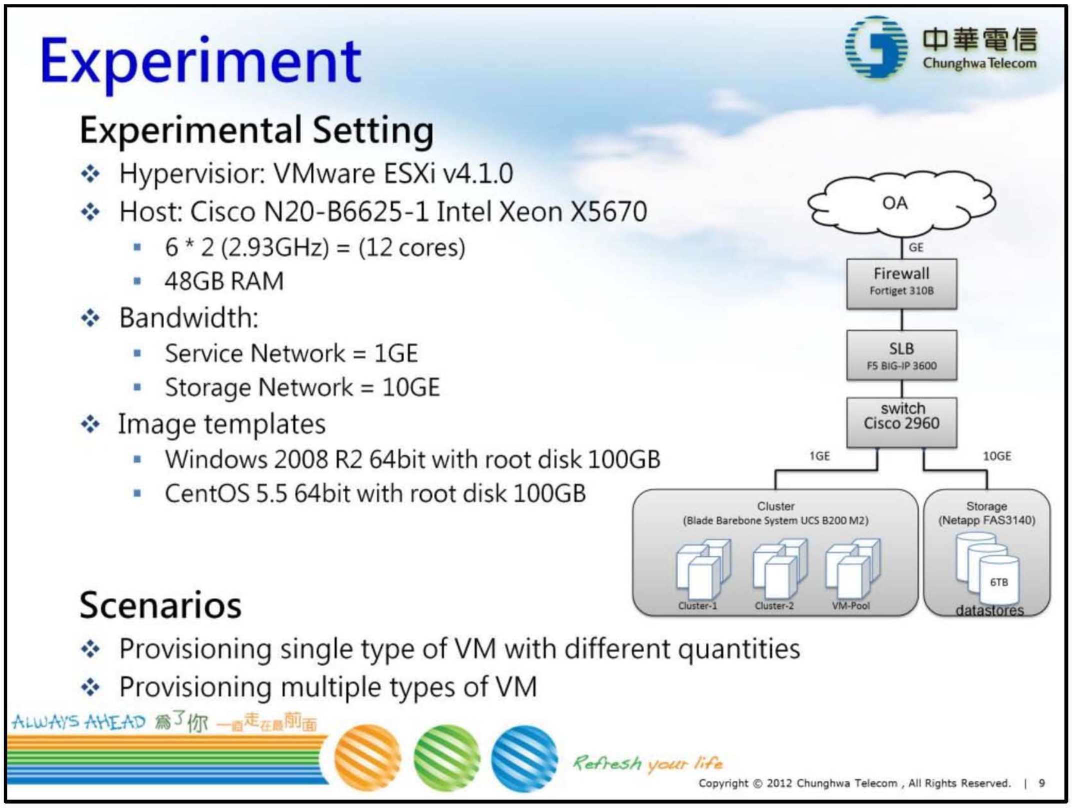
When does VM change status? There are four situations. The first one is refilling of VPM. VPM checks the number of primary VMs periodically. If one type of VMs in VM pool isn't enough, which means the number of primary VMs is under lower bound, VPM will trigger refilling mechanism until the number reach its upper bound. The second one is VM provisioning. When consumers order VMs, primary VMs which are from initialized status is customized by requirements of order requests. In VM provisioning, VM will change status from initialized to customized status. The third situation is when consumers start to use VMs, which will change from customized status to in service one. Then consumers are able to operate instructions in VMs. The last situation is terminating VMs. When consumers terminate VMs, VMs will be in terminated status and deleted. Besides, all related resources (e. g. IP address) are recycled to be re-used in the future.


VM Provisioning Process in Life Cycle

Process	Spent Time
Clone VM from image template in specific datastore	720
Initialize	~120
Rename VM	2
Change cluster	10
Set CPU & memory	2
Set network configuration: SLB, Firewall, DHCP	20

Corresponding to the process of VM provisioning, we show each status including required steps. For example, the customization of initialized VM needs VM renaming, change of cluster, CPU and memory and other network configuration. From this page, the most time-consuming steps, cloning VM from image template in specific datastore and initialization, are completed by VPM refilling. Therefore, it saves much time on VM provisioning. When implementing VM provisioning, we just need customization which can be done very quickly.

In conclusion, while consumers had to be waiting from image template status to use VM, with VPM, they just need to wait for customization, because VPM does the process from image template to initialization in advance. In our design, VM provisioning should be very fast.



Our purpose is to compare the VM provisioning efficiency between the direct-cloning VM and this new-designed architecture. In practice, there are many fundamental factors to decide how quick we should provision a VM such as VM specifications (e.g., virtual CPU cores, memory), disk I/O in physical datastores and service/functional network bandwidth. However in this paper, we aim at improvement of the novel mechanism based on the same environmental parameters. Each result value we provided in the paper may not apply in any physical setting or conditions.

In this experiment, we use VMware ESXi hypervisor with v4.1.0 version. We also uses two Cisco N20-B6625-1 Intel Xeon X5670 hosts which have twelve 2.93GHZ CPU and 48GB ram. There are two clusters. Each cluster has one host. One cluster is for VM pool and another one is for providing computing resource to consumers. The bandwidths for service and datastore are 1GE and 10GE respectively. Two kinds of image templates are for experimental scenarios. There are more details on the figure of system architecture.

In scenarios, we focus on provisioning VMs of specified OS image by simulating a customer’s need of considerable number of VMs with one type of OS image located in one datastore. Besides, in order to providing different observations in VM provisioning with various OS images, we provision VMs of two types of OS images located in different datastores.

Evaluation(1/3)

中華電信
Chunghwa Telecom

#	Average	Fastest	Slowest
1	0:14:48	0:14:48	0:14:48
10	0:15:19	0:14:57	0:29:12
30	0:36:07	0:14:41	0:59:07
50	0:54:40	0:14:46	1:45:43
100	1:41:25	0:14:59	3:12:14
50x2	0:57:31	0:15:01	1:48:36

Time of simultaneous VM provisioning
in the direct-cloning architecture (h:m:s)

#	Average	Fastest	Slowest
1	0:00:45	0:00:45	0:00:45
10	0:02:31	0:01:36	0:03:26
30	0:05:08	0:03:06	0:06:42
50	0:09:45	0:03:31	0:12:14
100	0:15:39	0:02:43	0:23:05
50x2	0:15:52	0:03:00	0:23:44

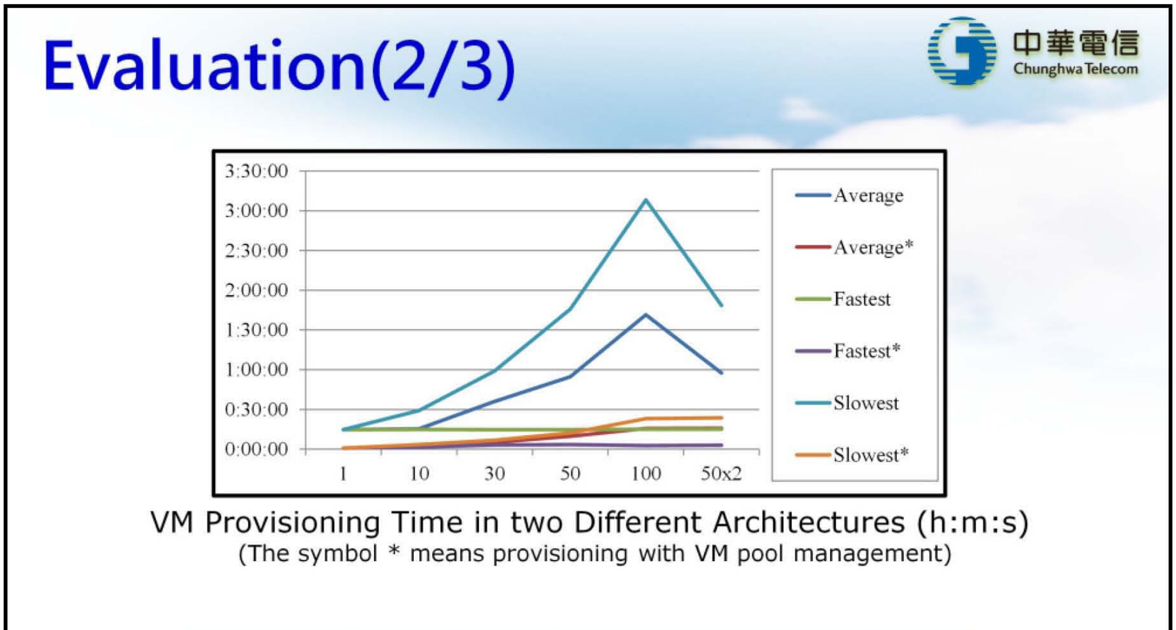
Time of simultaneous VM provisioning
in the architecture with VM Pool Management (h:m:s)

ALWAYS AHEAD 爲了你 一直走在最前面

Refresh your life

This page presents the experimental result of simultaneous VM provisioning with the directly cloning and VPM architectures. In this experiment, VPM prepares enough primary VMs for VM provisioning with different quantities. These tables show results of two scenarios. Each value in the first column means the number of provisioned VM. In particular, “50x2” means we choose 2 types of OS images and provision 50 VMs of each one. Each value on the rest of columns means the average, fastest and slowest provisioning time in different numbers of VM provisioning respectively. For example, on average, we take more than 45 minutes on provisioning 50 VMs.

Compared with these two tables, we have several discoveries. Firstly, in the direct-cloning, the time provisioning VM is proportional to the quantities of VMs. In this case, tested hypervisor is VMware ESXi, which takes VM provisioning in batch processing, especially in “cloning VM from template image” step. Secondly, obviously, provisioning time in new-designed architecture is much less than the direct-cloning. For example, provisioning 100 VMs just need about 16 minutes on average instead of one and half hours. Even in “multiple types of VM” scenario, provisioning time is considerably reduced to about 16 minutes. Lastly, even with VPM, provisioning 100 VMs still needs 16 minutes rather than a short time. That’s because this architecture can’t handle considerable requests simultaneously for now. Network configuration merely processes requests one by one. Therefore, network configuration has become the bottleneck of VM provisioning with VPM.

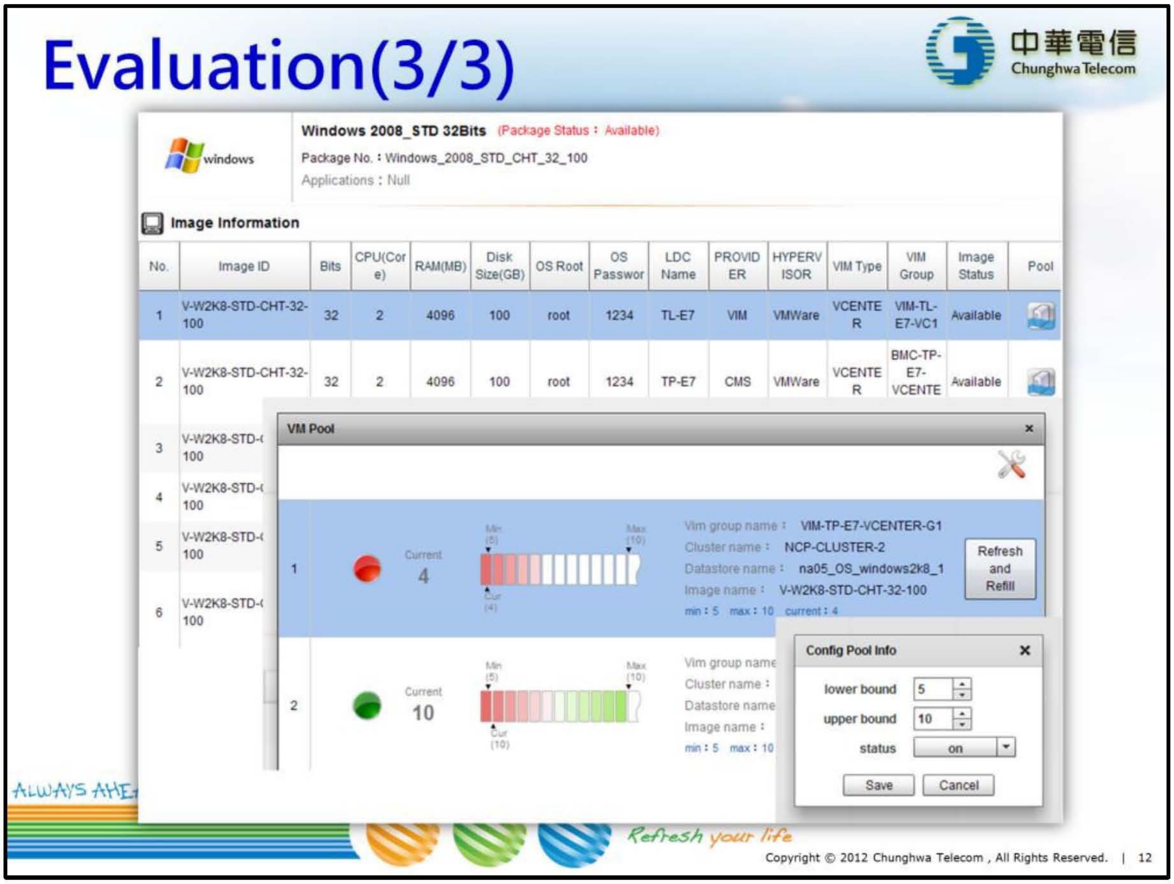


	1	10	30	50	100	50x2
Ratio (%)	94.93	83.57	85.79	82.16	84.57	59.74

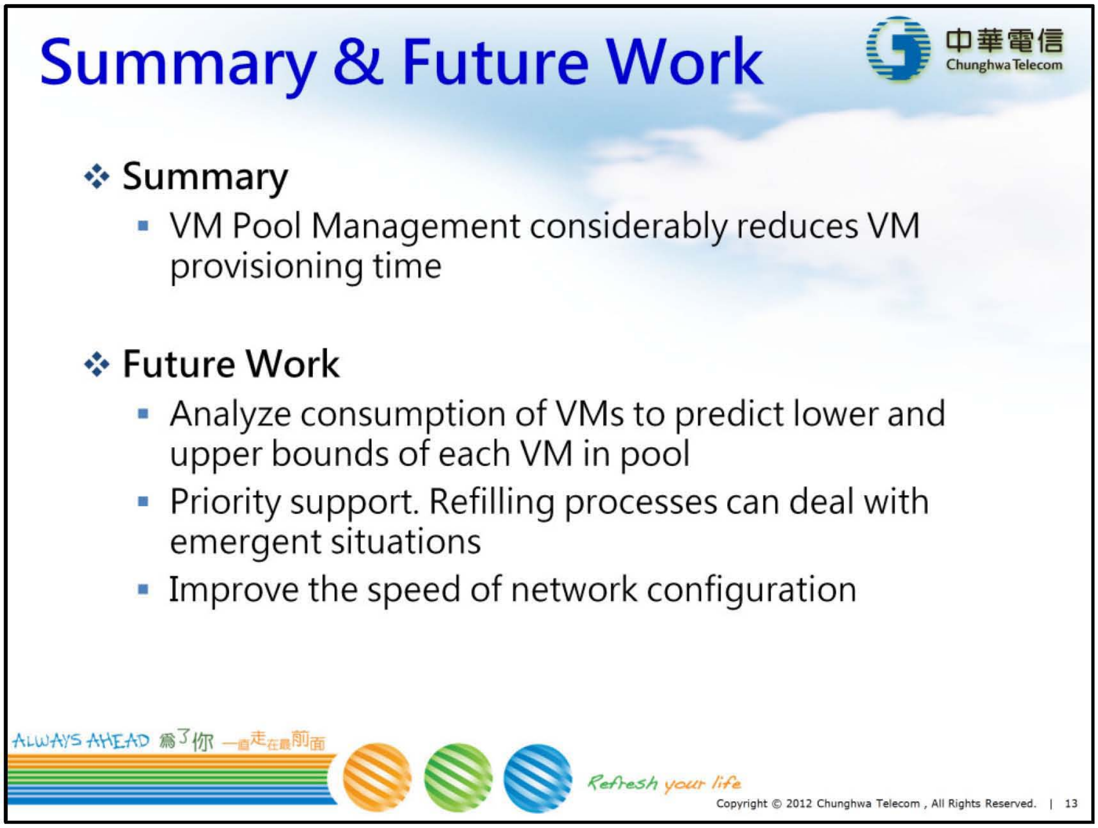
Ratio of Reduced Average VM Provisioning Time

We combine data to present differences graphically. As we can see, the provisioning time with VM pool is much less than directly cloning architecture with any number of VMs. The quantity is larger, the difference is bigger. In comparison with two slowest VM provisioning, we can promise customers to successfully get the workable VMs in 30 minutes instead of over 3 hours if a customer requests 100 VMs services immediately.

We also show the ratio of reduced average VM provisioning time. We successfully reduce over 80% time on provisioning any number of VMs of one specified OS image. Even in provisioning VM of two types of OS, near 60% time is saved in new designed architecture. These results have shown we successfully propose a fine-grained architecture and methodology to realize rapid VM provisioning for VMs on-demand purpose. Our proposed architecture with VPM is better than the direct-cloning one. That means our method does work well. And we have achieved our goal.



We've created a dynamic controller of VPM in graphical user interface mode. Operators in cloud computing data centers can supervise VM pool inventory status through this GUI interface; especially it could be controlled from the input of marketing. We provide the control panel to adjust the upper bound, lower bound and current status as well. The control panel of VPM helps operators to set required parameters conveniently and immediately-in-effect according the changes. In several practical cases, some particular consumers want 400 VMs few days later. However, operators can set VM pool configuration and prepare enough VMs in advance. Therefore, consumers can use VMs immediately whenever they want rather than to wait VM provisioning for hours or days.



In this paper, we introduce a newly-designed architecture and mechanism for delivering VMs in an efficient way. Our main purpose is to speed up VM provisioning in two practical situations such as simultaneous and considerable VM provisioning. With VPM, we have quite great enhancement on the speed of VM provisioning. In addition, we've in-directly resolved the situation of peak times provisioning. The loading of provisioning engine is reduced as well and the increase of waiting time of on-demand customers is avoided. It doesn't suffer from heavy file transmission and operations any more. It also increases the probability of successful VM provisioning significantly via VM generation of VPM before delivering to the customers. From the viewpoint of operation, the graphical user interface of managing VPM is designed for artificial supervision and configuration reflecting the market demand in time.

In rapid provisioning areas of delivering virtualized resource on demand for customers, several important issues we have to improve and realize still exist. Firstly, the trend prediction model of VMs consumption of each category could be evaluated by VPM to configure VM pool intelligently. Secondly, the features of priority mechanism in managing pools could be added to VPM when executing refilling process. The last one of all issues is improving the speed of network configuration on considerable requests by batch processing. We are expecting all these enhanced future works are advanced to reinforce a more intelligent provisioning engine for running a successful cloud computing business.


中華電信
Chunghwa Telecom
ALWAYS AHEAD 為了你 一直走在最前面
Thank you for your attention
Telecom Laboratory
Chunghwa Telecom Co. Ltd.
http://www.chttl.com.tw/web/en/index/
Refresh your life
Copyright © 2011 Chunghwa Telecom , All Rights Reserved. | 14

References

[1] Michael A., Armando F., Rean G., Anthony D. J., Randy K., Andy K., Gunho L., David P., Ariel R., Ion S., and Matei Z., "Above the Clouds: A Berkeley View of Cloud Computing", 2009.

[2] A Weiss, "Computing in the clouds", *COMPUTING*, 2007

[3] hicloud, http://hicloud.hinet.net/

[4] NIST Definition of Cloud Computing Peter Mell and Tim Grance, Version 15, 10-7-09 http://www.nist.gov/itl/cloud/

[5] L. Hanhua, W. Yashi ,M. Lijuan1 ,H. Zhenqi. OSS/BSS Framework based on NGOSS, *IEEE,* 2009.

[6] TMF GB939 , "NGOSS Contract Examples", Release 6.0, November 2005

[7] Jia-Bin Yuan, Yi-Ching Lee, Wudy Wu, Hey-Chyi Young, Kuan-Hsiung Liang, "Building an Intelligent Provisioning Engine for IaaS Cloud Computing Services". *NPNOMS,* 2011.

ALWAYS AHEAD 爲了你 一直走在最前面

Refresh your life

