

Dynamic Alterable Computing Environment System ---- Building Based On Virtualization

REN Xiang^{1, a}

¹No.99 Houbiaoyin Rd, Qinhuai District, Nanjing Jiangsu 210007, China

^arenxiang0214@126.com

Keywords: Virtualization; Computing Environment/Resource; Dynamic Building; vSphere API

Abstract. For quickly, realistically and effectively building up the computing environment in the emulational network and electromagnetic circumstance, how to utilize the underlying virtualization technology to realize dynamic building has become a hot spot and difficulty for researching. Prompt a dynamic building method based on the vSphere API provided by VMware. Experimental results show that achieving pre-configuration and high automation deployment for scalable computing environment.

The Background of Computing Environment's Dynamic Building

Along with the rapid development of cloud computing, virtualization has been paid more attention than ever. One of the most important reasons is that number of large-scaled organization meets a series of problems which come along with rapid growth of giant servers as well as the amount of information and business need to deal with expands exponentially. The cost of servers is expensive while usually one server used only for one purpose which is lack of flexibility while the hardware performance of servers becomes more powerful. With help of virtualization, each organization is able to improve performance and efficiency of IT framework dynamically. Virtualization breaks up traditional model of "one app running on one server" [1].

The powerful ability of virtualization mentioned above is the important advantage of model of cloud computing, where that when making the cost of buying kinds of computing and network hardware as lower as possible, the performance of existing equipment could be made fully use of at the same time, which meanwhile could provide unprecedented flexibility of changing the basic framework [2].

Network Electronic Space (Cyberspace) has become a new hot point. American Cyberspace combat troop already has powerful capabilities [3]. Our security of Cyberspace is meeting serious threat.

For the information infrastructures used in military, if we directly update or modify the system leading to failures, the consequences could be devastating. Even though we could, because of the complexity of system, procedures of testing couldn't be controlled effectively, which will affect the evaluation of testing results. In conclusion, we could not process effective testing in the real environment [4].

Building simulation experiment field of network and electromagnetic circumstance with virtualization could utilize limited hardware resources to simulate testing platform which is close to or equal to real environment while not affect the operations of real systems, then we could process reliability test of new technologies or new systems in this simulation platform. With the flexibility of virtualization, we could also realize the rapid transferring between different experiment environments. In traditional way of experiment, each PC client, each switch, each router has to use the real hardware for testing. The cost is huge while the hardware used in one experiment could hardly be reused in other experiment. Once using virtualization, besides saving cost of buying a lot of hardware devices, we could change strategy rapidly so as to utilize one suit of equipment to build platforms for different experiments of different kinds as well as constructing hardware environment rapidly with few devices

[5]. This paper is searching about the methods of building dynamic alterable computing environment system, hoping to give out an innovative feasible scheme.

System Framework of Building Dynamic Alterable Computing Environment System

Resources generating and configuration is the key technology of building dynamic alterable computing environment, the overall requirement is utilizing virtualization to build a customized, self-deploying and process controllable simulation experiment field of network and electromagnetic circumstance including network, compute and application resources, the system framework is showed in figure 1:

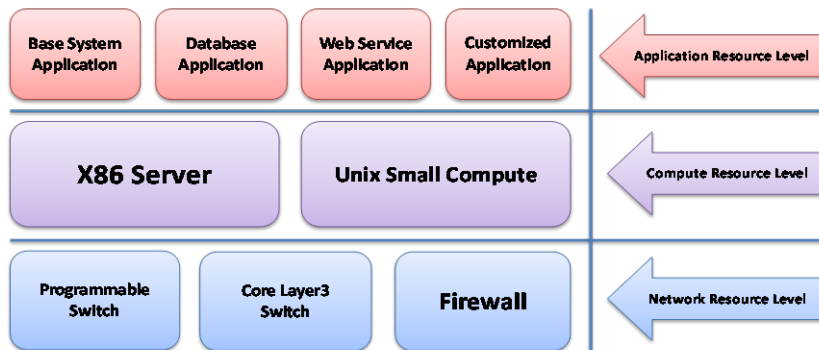


Figure 1 System Framework of simulation experiment field of network and electromagnetic circumstance

The experiment environment should meet following requirements:

- 1) Clone Network System with at least 500 nodes;
- 2) Generate typical Info System with at least 100 compute nodes, and parameters modified on demand;
- 3) Could configure and generate resources including Network, Compute, Application System and so on;
- 4) Based on above, could configure attacks including simulated virus, DDOS, wireless injection and so on.

According to demands of experiment, the procedure is showed in figure 2:

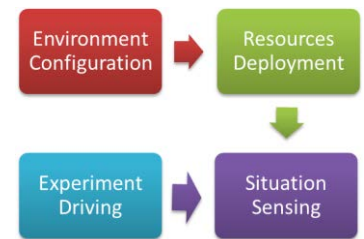


Figure 2 Procedure of building, generating, driving of

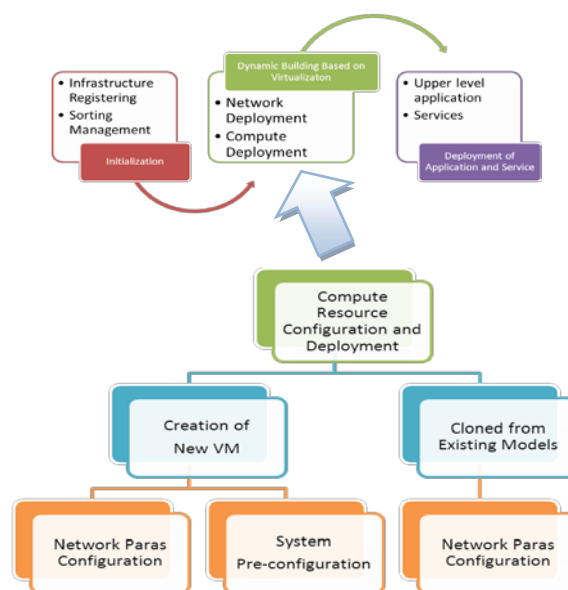


Figure 3 Building procedures of simulation experimental field and Details of compute resource deploying

Configure scale of target experiment through environment configuration model and generate the corresponding configuration document, then deliver the configuration information to resources deploying software for realizing the dynamic deploying and tuning of kinds of resources under

different scales. Meanwhile, according to deploy schedule, related experiment driving information will be loaded simultaneously as the input of generated experiment environment.

In the building procedures above, environment configuration and resources deploying have compact relation with building of dynamic alterable computing environment. The details of procedures are as follows: Environment initialization, Generating of target experiment environment of network, compute, application system and so on. The environment initialization includes registration and sorting management of infrastructure, while network and compute resources deploying could be considered as hardware resources generating of simulation experiment. At last is the deploying of all upper level application and services. The procedures' relation is showed in figure 3 above:

This paper focuses on rapid deploying and configuring of compute resource. The logic framework is as showed in the lower part of figure 3 and it contains creating new virtual machine (VM) and copying from existing typical models. The key technology adopted of the two parts above is based on the vSphere API provided by ESX production of VMware, which utilizes the creation and copy of VM and models, customization of OS and interfaces of hardware reconfiguration of VM to re-program, and combines with virtualization function of ESX's powerful platform to realize the abilities mentioned above.

The corresponding software implementation of procedures above has been completed, and next section will give out detailed information about invoking procedure of vSphere API in configuration and deploying of compute resource within the software implementation of the hardware resource deploying.

Method of Building Compute Environment On Demand

The key technology of building procedures mentioned above is according to real status of compute resource's building in deploying, and then dynamically changing the deployment due to pre-set rules. The basement of it is right the API provided by VMware. Compared with the complexity of open source program, this set of SDK with APIs is already mature, while the difficulty is the using of API. Although the official introduction covers all functions and meanings of parameters, it doesn't give out the detailed method of usage as well as the details of parameters' setting including customization modification of VM's OS like IP changing, while explicit explain and using case of corresponding invoking relations and invoking method are not given out, and the references of error code are rather limited. For rapid and vivid building of experiment environment, fast deployment of compute resource is key point. Among all procedures, after creating of needed compute node, how to configure OS's IP and deploy application automatically has always been a hard problem. Through API mentioned above, we could customize the OS and make the copied VM to begin working in the target environment immediately after deploying.

Through continuous practicing is the development, combined with configuration document generated in the management platform of VMware, as well as "VMware VI and vSphere SDK" [6] wrote by Steve Jin, related invoking method of OS pre-configuration with C# language has been captured. During the way to research the realization of the dynamic building, we do deep researching and practicing of related API, and will introduce some important APIs which are lack of developing references and using cases in the next part.

VM cloning API -- cloneVM_Task(), the data structure VirtualMachineCloneSpec used by this API has already included data structure CustomizationSpec needed by customization. Meanwhile, we could also customize the cloned VM through the API -- customizeVM_Task(), which will need the data structure CustomizationSpec as well.

For realizing the automatic customization of OS, firstly install the VMware Tools (middleware of managing the customization of VM in the ESX platform) into the VM or models needed to be cloned. And then for Windows OS, you need to install Microsoft Sysprep tools into specified folder on the PC which vCenter is installed, meanwhile, if the Windows OS deployed is not volume licensed, it will

need to be reactivated. And for Linux, VMware provides several Perl scripts which located in specified folders of vCenter, when customizing they will be copied into ESX platform for executing. Customization could also be done in the GUI of vSphere Client, while related setting could be exported in XML format, which would be much helpful for understanding how to use the APIs.

After finishing preparing work of platform, parameters of API could be set. Firstly create an object of CustomizationSpec structure, and parameters have to be defined and set are as follows: Network Configuration. Multi-NIC is acceptable while IP, gateway, netmask and DNS of each NIC could be set respectively. For example:

```

customization.nicSettingMap = new CustomizationAdapterMapping[2];
string[] gateway = { "192.168.11.1" };
string[] dnsServer = { "192.168.11.1", "192.168.11.254" };

customization.nicSettingMap[0] = new CustomizationAdapterMapping();
customization.nicSettingMap[0].adapter = new CustomizationIPSettings();
customization.nicSettingMap[0].adapter.ip = new CustomizationIpGenerator();
CustomizationFixedIp newIP = new CustomizationFixedIp();
newIP.ipAddress = "192.168.11.204";
customization.nicSettingMap[0].adapter.ip = (CustomizationIpGenerator)newIP;
customization.nicSettingMap[0].adapter.subnetMask = "255.255.255.0";
customization.nicSettingMap[0].adapter.gateway = gateway;
customization.nicSettingMap[0].adapter.dnsServerList = dnsServer;

customization.nicSettingMap[1] = new CustomizationAdapterMapping();
customization.nicSettingMap[1].adapter = new CustomizationIPSettings();
customization.nicSettingMap[1].adapter.ip = new CustomizationIpGenerator();
CustomizationFixedIp newIP2 = new CustomizationFixedIp();
newIP2.ipAddress = "192.168.11.207";
customization.nicSettingMap[1].adapter.ip = (CustomizationIpGenerator)newIP2;
customization.nicSettingMap[1].adapter.subnetMask = "255.255.255.0";
customization.nicSettingMap[1].adapter.gateway = gateway;
customization.nicSettingMap[1].adapter.dnsServerList = dnsServer;

```

Once all parameters being set, corresponding API could be invoked and passed to ESX to start mission. All above is the realization of single Compute Node pre-configuration as well as the key point of whole building procedures. In pure physical environment, when processing large scale deploying, each Compute Node has to be pre-configured by hands while could not be automatic. Key technology realized above makes automatic customization of OS in large scale deploying feasible which improve efficiency of building vastly.

Compared with single point compute resource pre-configuration, when dynamically building the large scale experiment field, authentication strategy and OS information protection should be paid more attention while deploy efficiency has to be considered as well. Taking these two sides into account, we use the mutual API of ESX to verify information of each compute node created to make sure consist of identity, and coordinate with multi-thread having lock mechanism to invoke APIs mentioned above. On condition that mutual isolation could be guaranteed, realize effective dynamic building of multi-point compute resources.

Case Study

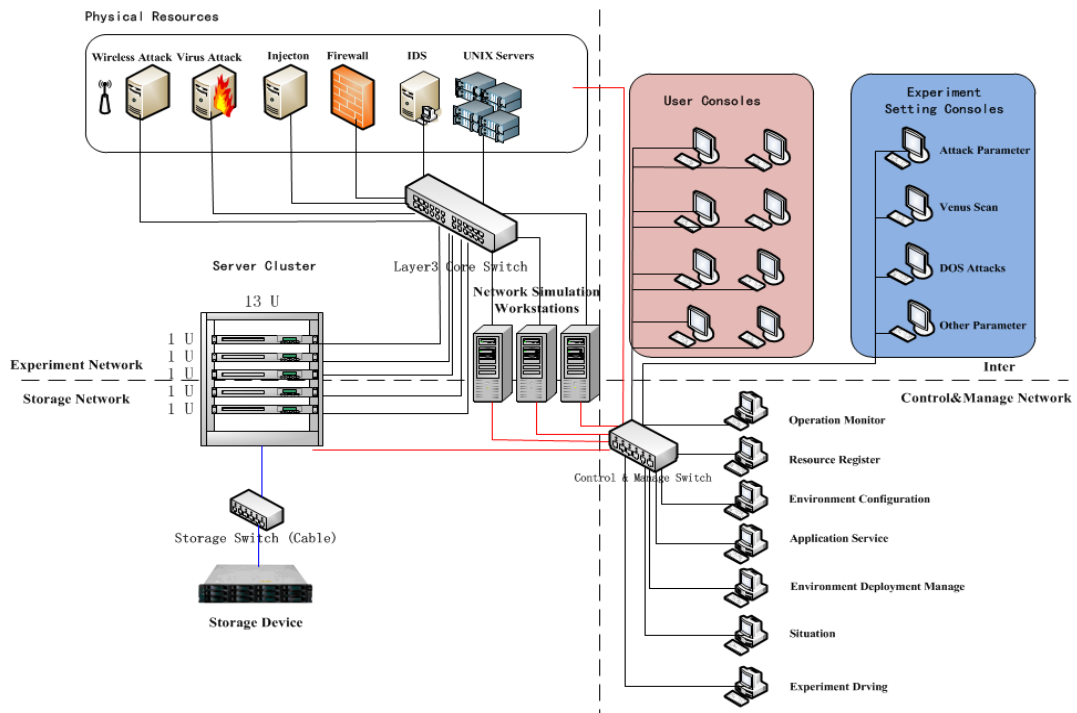


Figure 4 Hardware Framework

Hardware resources used by our program contains the followings: virtualized compute resources group including 5 IBM 3650M3 servers, simulated network server including 3 Lenovo D30 workstations, 1 IBM V3512 NAS, 1 Juniper programmable switch, 4 Cisco layer3 switches, 5 HP UNIX servers, 2 Huawei Firewalls, 1 Venustech IDS and 7 ThinkStation PCs, which are showed in figure 4 above:

Next part will introduce how the system mentioned above could support the dynamic building, so as to verify the feasibility and applicability of the system. With the corresponding configure and deploy software developed, at least 500 network nodes (routers, switches) and 100 compute nodes (Windows XP, CentOS, Backtrack OS) could be created, configured and deployed within 60 minutes. Snapshot of software's operation is showed in figure 5:

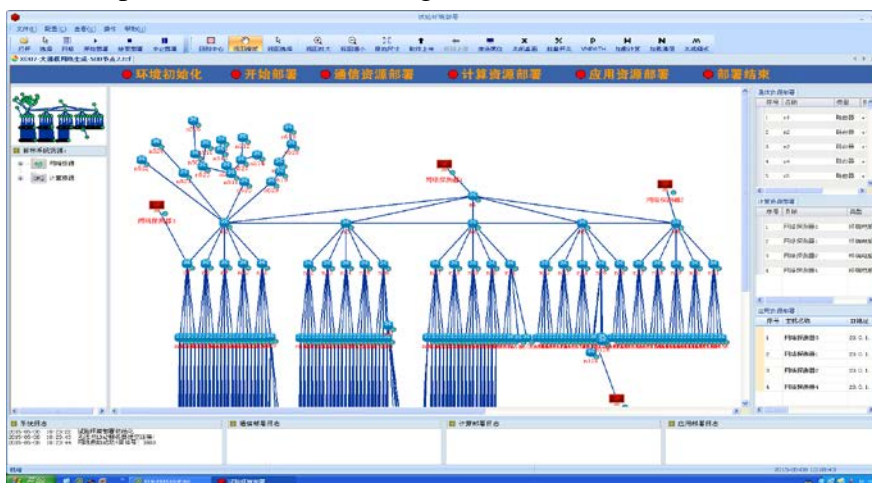


Figure 5 Snapshot of Software's Operation Window

Details of implementation: read in resources information needed by experiment field, and count key needs of performances like total CPU frequencies and total memory sizes. Compared with hardware resources mentioned above and communicate with ESX through related API to test whether the need could be satisfied. If test is OK, then start to deploy: Firstly create each compute node by VM APIs, and verify successful or not through ESX so as to control the schedule of deployment. After all

resources being created, read detailed customization requirements of each OS by turns and then complete the pre-configuration by using APIs mentioned in Section 3, while remember to verify successful or not immediately after each configuration. After finishing all verification we could get an experiment field matching all hardware requirements. Based on this, deploy specified application and services on VMs and then start doing experiment.

Conclusion

At present, there's no similar researches could be referenced inland while compared to American National Cyber Range (NCR) program, the number of network nodes supported by method of this paper is the same while the number of compute nodes is 20% less, and the deployment speed is the same as NCR's 60 minutes. In the building of dynamic alterable computing environment system, besides automatic configuration of network and system introduced in this paper, how to combine automatic configuration of different OS like Windows and Linux is still a hard problem. In the future study, we will continue to focus on development of related technologies.

References

- [1] Yan Yanna. A study on adaptive allocation strategy and mechanism of virtual resource on virtual machine platform. Dissertation Submitted to Hangzhou Dianzi University for the Degree of Master,2009.
- [2] Wang Xiaomei, Ma Fang. Analysis on Application of Computer Virtualization. *China Computer&Communication*, 2012,5:136-137
- [3] Sun Xin, Fei Hong, Zhao Feng, Zhu Hong. U.S.Military Foundational Cyberwarfare Plan X. *Command Information System and Technology*, 2013, 4(3): 25-28.
- [4] Hu Lin. Brief Introduction of Application of Cloud Storage and Virtualization in Military Security. 2011.
- [5] Xu Li, Zeng Zhibin, Yao Chuan. Study on virtual resource allocation optimization in cloud computing environment. *Journal on Communications*, 2012,33(Z1):9-16.
- [6] Steve Jin. *VMWare VI and Vsphere SDK – Managing the VMWare infrastructure and Vsphere*. NEW YORK, 2010.