Network traffic optimization architecture for scalability in academic inter-cloud computing environments

Shigetoshi Yokoyama
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo 101-8430, Japan
+81-3-4212-2726
yoko@nii.ac.jp

Atsushi Matsumoto
SCSK Corporation
Toyosu Front, 3-2-20,Toyosu,Koto-ku,
Tokyo 135-8110, Japan
+81-3-5166-1927
at.matsumoto@scsk.jp

Nobukazu Yoshioka
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo 101-8430, Japan
+81-3-4212-2570
nobukazu@nii.ac.jp

ABSTRACT

In academic inter-cloud environments, virtual machines communicate with other virtual machines in different virtual L2 network segments through a virtual router. As virtual routers are deployed on certain physical machines, communication between virtual machines via virtual routers could not be an optimal in physical sense. We are proposing an academic inter-cloud architecture (AIC) [1] which consists of AIC compute and AIC storage. AIC is a hub of clouds and it provides compute physical resource on-demand for each cloud. It means that AIC provides extension to the clouds. The clouds meet in AIC. In that sense, the virtual machines in each cloud are physically close. However, the communication between virtual machines which are located in different clouds has another level optimization problem. The communication between the virtual machines is through the gateways of each cloud. The latency cloud be much smaller if we can optimize the route. For cloud scalability we may provide bare metal servers at geographically distributed manner but the network latency should be optimized by application requirements. We propose an architecture to solve these issues which has L3 function delegation mechanism to open flow switches to reduce network latency according to users' requirements for short cuts.

Categories and Subject Descriptors

D.2.11 [Software Architectures]: Inter-cloud software architecture.

General Terms

Management, Performance, Design.

Keywords

Cloud computing, Inter-cloud, Open flow, Software defined network.

1. INTRODUCTION

Cloud computing is now widely used in the academic community. Users in the academic community have a lot of opportunities to use public clouds and private clouds operated in universities. The academic inter-cloud connects private clouds operated in universities and the cloud platform shared among the universities via the high performance and secure network. It also offers services for federating/sharing computing resources among clouds in order to support advanced research and education. This paper presents the network traffic optimization architecture for scalability in academic inter-cloud computing environments.

Private clouds get some benefit from the consolidations made possible by using virtualization technology. However an individual organization cannot reduce IT costs significantly through the use of its own private cloud because it must have on hand the maximum IT resources needed to deal with peak traffic.

In order to better utilize IT resources, a hybrid cloud solution is feasible in some situations. A hybrid cloud consists of a private cloud and public cloud; the private cloud deals with flat traffic and the public cloud covers peak traffic. However, when security matters, it is not feasible to send all the peak traffic to the public cloud. It is important to think about sharing IT resources among private clouds to ensure better utilization and security at the same time. This idea can be viewed as a private cloud hosting service. The private clouds are deployed on demand by getting bare metal servers from the server pools. The private cloud can be separated physically by using open flow closed networks. On the other hand, a community cloud is a way to keep clouds independent from one another while getting flexibility and security at the same time. In fact, there has been a lot of activity on ways to establish community clouds. The approaches can be categorized into two kinds. One is standardization of cloud services and the other is multi-cloud federation. The standardization approach would take time; i.e., it would not be effective until there are enough implementations and deployments following the standard specifications. The federation approach places limitations on the functionalities provided to users; they have to be the greatest common divisor of the clouds' functions.

We proposed a new approach, called "cloud on demand", which integrates many private clouds horizontally and shares IT resources among them to accommodate peak traffic. By applying this solution, users can get good IT resource utilization like in a public cloud and have the level of security of a private cloud. [1] If the academic inter-cloud infrastructure cannot provide enough resource when the total usage of the community peak is too big, it will extend to public clouds.

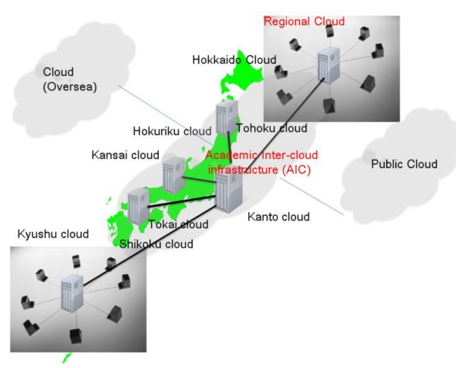

Figure 1 Academic Inter-cloud Infrastructure

In the academic inter-cloud environments, virtual machines communicate with other virtual machines in a different virtual L2 network through a virtual router. As virtual routers are deployed on physical machines similar to the virtual machines, communication among virtual machines via virtual routers could not be an optimal in physical sense. The communication among machines which are different clouds has another level optimization problem. We propose an architecture to solve these issues which has L3 function delegation mechanism to open flow switches to reduce network latency. This paper is organized as follows. Section 2 describes the Academic Inter-Cloud (AIC) software architecture. Section 3 introduces the network traffic optimization architecture. Section 4 shows prototype implementations for the network optimizations and we conclude in Section 5.

2. AIC Software Architecture

Our approach is "cloud on demand". Figure 2 is an overview of our cloud on demand solution. There are two service components. One is called AIC Compute [2], and the other is called AIC Storage [3], [4], [5]. AIC Compute is a service by which users create clusters consisting of physical servers, and it can deploy software components for building IaaSes and PaaSes.

The AIC Storage lets users store objects, such as machine images, as if they were using a local cloud object storage service. Physically, each cloud is connected to a high-speed wide area network, such as SINET-4 [6]. The network connections are made by using network functionalities like L2VPN. The physical servers can be located in the same L2 network segment if the same VLAN–ID is assigned to them. The physical servers that are assigned different VLAN-IDs are securely separated from the other network segments.

Through this design, we can generate physical machine clusters in inter-cloud environments on which we can deploy IaaS software like OpenStack, Eucalyptus, and others in our favorite configurations for each.

In addition, we configure a distributed inter-cloud object storage service using open source software like OpenStack swift for storing machine images.

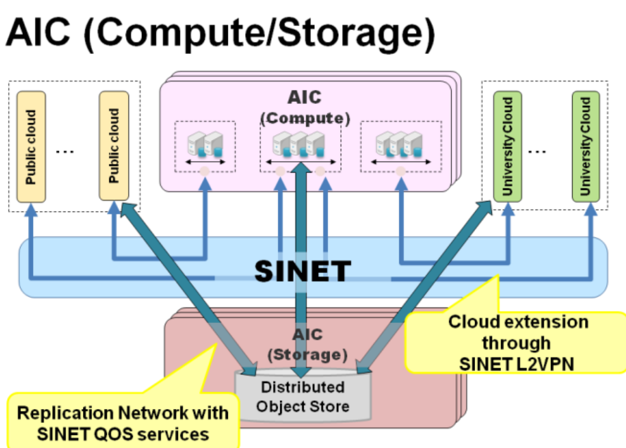

Figure 2 Architecture overview

2.1 AIC Compute

AIC Compute is designed as follows:

1) Two-layer implementation

The lower layer takes care of physical machine cluster management. The upper layer handles virtual machine cluster management. Moreover, each layer is programmable with web APIs.

2) The lower layer

The lower layer handles the operating systems of each node composing a cluster. Nodes can be allocated to clusters dynamically from software and securely separated by using open-flow network technology, which is called cluster as a service.

3) The upper layer

The upper layer deals with deploying IaaS software such as OpenStack and Eucalyptus. It also can deploy PaaS software. The layer has configuration management tools to ease deployment on the nodes of clusters. An actual deployment example is depicted in Figure 3.

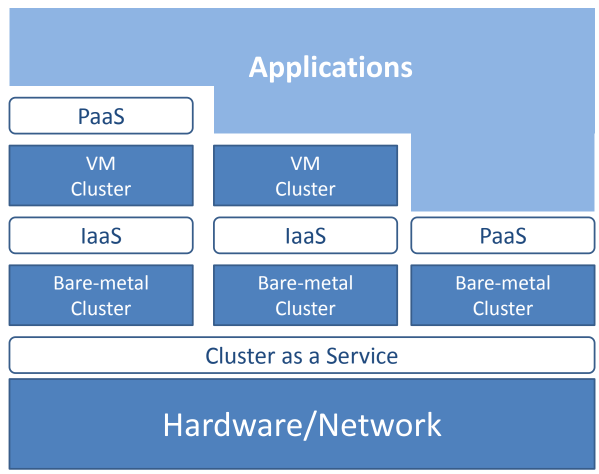

Figure 3 AIC Compute Software Stack

The figure 4 shows a deployment example of two cloud infrastructures by using AIC Compute. When the cloud-A need more resource than the university-A provide in its private cloud, the cloud-A can be extended to AIC using AIC Compute to get more bare metal machines for the extension. The same can be done for cloud-B and other clouds. The users should not know this extension in usual use cases. After the burst, the resource which is allocated to the cloud-A is returned to the AIC machine pool.

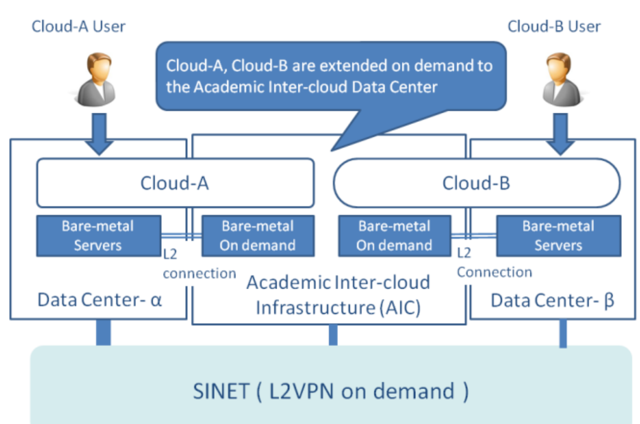

Figure 4 Academic Inter-cloud Infrastructure usages

2.2 AIC Storage

AIC Storage is geographically distributed object store service which is used to allocate the application execution environments like figure 5. We deploy an IaaS cluster on demand on physical servers and deploy the application virtual machine cluster on it. In this sequence, the IaaS cluster uses the AIC Storage to launch virtual machines from machine images that have been prepared for the scientific application cluster. IaaS clusters themselves are not necessarily destroyed after each the application execution. The life cycle of the IaaS is independently controlled by the application execution environment managers.

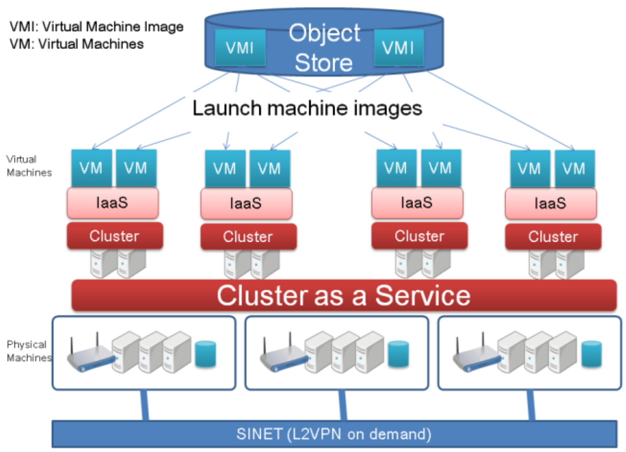

Figure 5 AIC Storage for application execution environments

3. Network traffic optimization in AIC

We implemented AIC compute and AIC Storage based on OpenStack nova and swift [7] and deployed a prototype system which is directly connected to SINET at National Institute of Informatics. It works fine for usual use cases but if applications have critical network latency requests we need to enhance the architecture by which we can establish network traffic optimizations. There are two levels in the optimizations. One is in a cloud infrastructure and the other is among cloud infrastructures.

3.1 Optimizations in a cloud infrastructure

Let's assume Cloud-A in figure 6 is an IaaS like OpenStack and it is extended to from the data center alpha to the AIC. Because the virtual machine vm01 and vm11 are connected to different L2 segment the network connection between them have to be through the router which is located in the data center alpha using two Open vSwitches. The vm01 might be an application server and the vm11 might be a database server. The network latency between the data center alpha and the AIC causes severe performance issues. To solve the problem we have to optimize the traffic between vm01 and vm11. In the case of figure 7, vm01 and vm11 are in the same physical machine at AIC and in this case we delegate some part of L3 functionalities in the router which is at data center alpha side to the Open vSwitche ovs2. By doing this, the network latency between vm01 to vm11 will be the one within a single machine. In the case of figure 8, vm01 and vm13 are in the different machines at AIC. Even in this case, we delegate some part of L3 functionalities in the router to the Open vSwitche ovs2 and ovs3 in order to make the traffic between vm01 to vm13 stays in AIC.

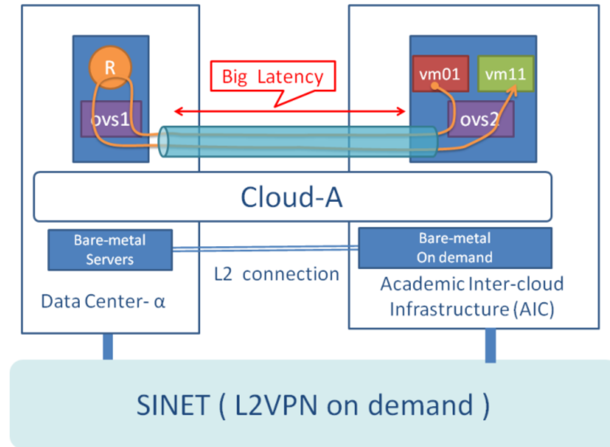

Figure 6 Issue in a cloud infrastructure

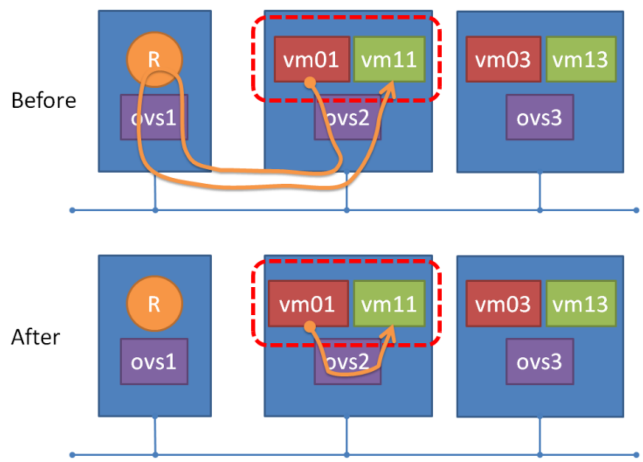

Figure 7 Case in one physical machine

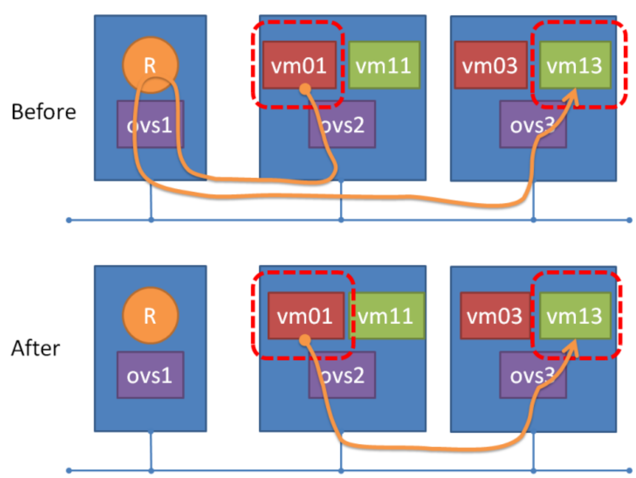

Figure 8 Case in different physical machines

3.2 Optimizations among cloud infrastructures

As described in figure 4, the cloud-A and the cloud-B are both extended to the AIC and the virtual machines in them may have to communicate. For example, let's assume that the cloud-A is the research cloud of the university-A and the cloud-B is the research cloud of the university-B and they have some research collaborations on the clouds in the figure 9. The traffic between vm01 to vm13 goes through the gateways at both universities. That means it goes thought the internet.

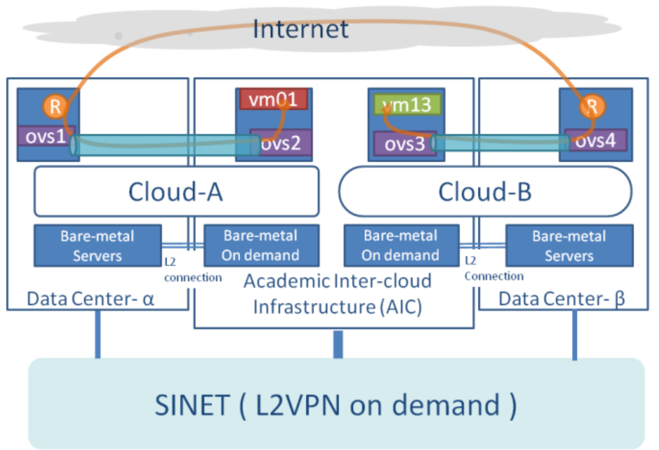

Figure 9 Network traffic among cloud infrastructure

To eliminate the problem, our approach is to develop AIC Network service in AIC. AIC Network will handle several network functionalities like the firewall, VPN and so on as a cloud service. In the figure 10, the upper side is usual network configuration with AIC Compute. Clusters in AIC are extensions of university clouds. FWs and VPN gateways are prepared by the university cloud managers at their sites. The idea is to make left-side right and right-side left, which means putting the network services in the hub. It should be better for network latencies and ease of network managements. If the universities are located in different regions in the globe, the network latencies between virtual machines in AIC can be several hundred milliseconds even if they are close enough in the same date center.

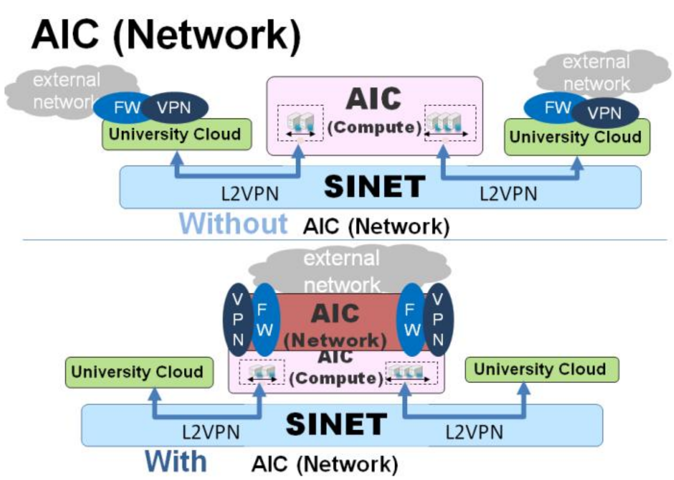

Figure 10 AIC Network

By using routing functionality in AIC Network the problems described above will resolve like figure 11. The traffic from vm01 to vm13 goes to AINC Network router and will not go out from AIC in this configuration.

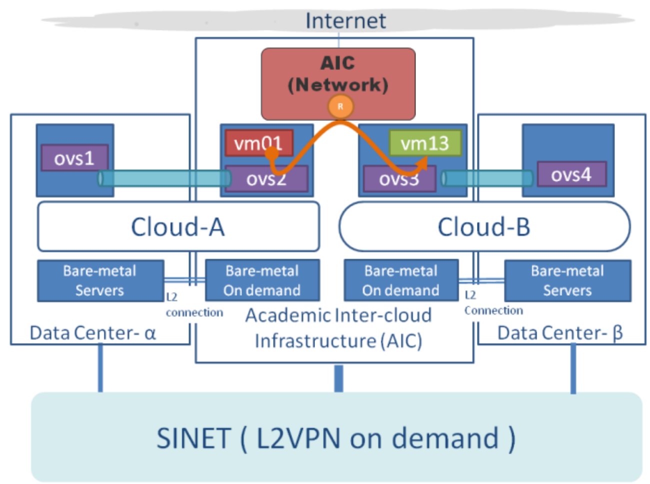

Figure 11 Routes via AIC Network

4. Prototypes

We developed two prototypes for investigating the architecture described in section 3. One is for the network optimization within a cloud infrastructure and the other is for that among cloud infrastructures.

4.1 Prototype for the optimizations in a cloud infrastructure

We set up a prototype using OpenStack grizzly which has the compute node1 at the AIC which is located in Tokyo and the compute node 2 and the network node in Ishikari and they are connected via GRE tunnels on SINET L2 VPN. The distance between Tokyo and Ishikari is about 1,000 kilometers.

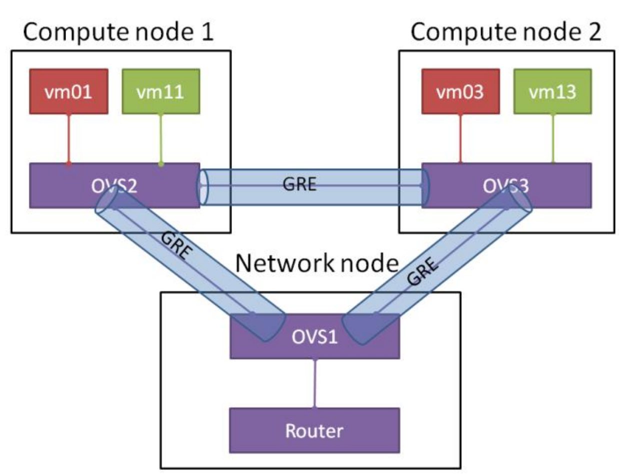

Figure 12 Prototype for the optimizations in a cloud infrastructure

The original round trip time between vm01 to vm11 is more than 43 msec. and that of after optimization is around 1msec.

4.2 Prototype for the optimizations among cloud infrastructures

We set up a prototype using OpenStack grizzly neutron “VPN as a Service” as AIC Network prototype in AIC prototype environment at National Institute of Informatics. We can connect from vm01 in cloud-B to vm13 in cloud-A thought VPN. From this experiment, we can know that by managing AIC Network in the hub we can make short cuts among cloud infrastructure extensions in AIC.

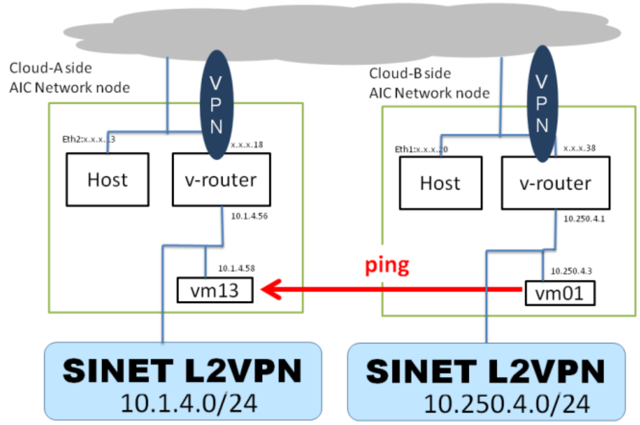

Figure 13 Prototype for the optimizations among cloud infrastructures

In this prototype, the connection between cloud infrastructures is VPN tunnels but we can have another configuration. AIC Computer separates each cloud infrastructure by using open flow controller and switches describes in figure 14.

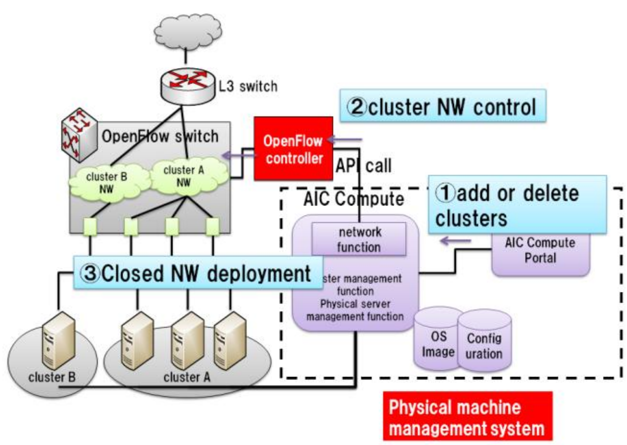

Figure 14 AIC Compute network control mechanism

Because AIC Compute is implemented based on OpenStack nova bare metal with open flow controller and switches, we can delegate some L3 functionalities to the bare metal OpenStack neutron and modify flow tables to make the short route among the cloud infrastructures.

5. Conclusion

We introduced AIC architecture and the network latency issues when we extend the deployment in geographically wide area in order to have scalable cloud infrastructure. There are two scoop levels to optimize network traffic. One is network traffic with in a cloud infrastructure and the other is that of among cloud infrastructures. The former issue will be resolved by managing Open vSwitches in the node delegating some L3 functionalities from virtual router in the cloud infrastructure. The latter issues will be eliminated by applying AIC Network which is new component to AIC.

The prototype systems for evaluating these approaches are built and we feel that they have potential to realize AIC vision in Figure 15. We will continue to design the network optimization in two levels and AIC Network. We are looking forward to sharing the evaluation results after we finish the design and implementations in the near future.

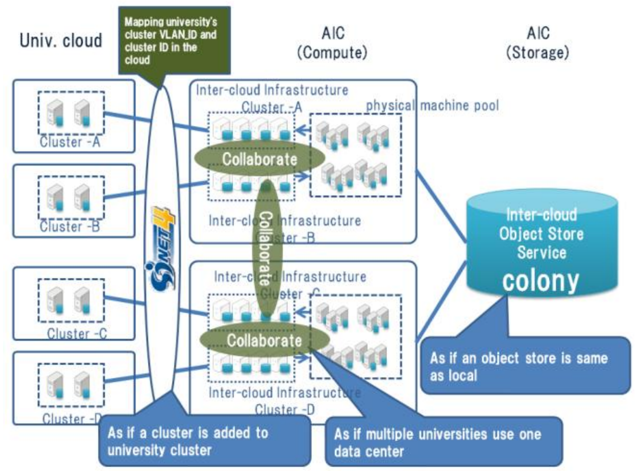

Figure 15 AIC vision

6. REFERENCES

[1] Shigetoshi Yokoyama, Nobukazu Yoshioka, "On-demand Cloud Architecture for Academic Community Cloud", CLOSER2014, 4th International Conference on cloud computing and Service Science, 2014 (to appear)

[2] Shigetoshi Yokoyama, Nobukazu Yoshioka, "Cluster as a Service for self-deployable cloud applications", pp.703-704, Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium, 2012

[3] Shigetoshi Yokoyama, Nobukazu Yoshioka, "An Academic Community Cloud Architecture for Science Applications," pp. 108-112, 2012 IEEE/IPSJ 12th International Symposium on Applications and the Internet (SAINT), 2012

[4] Shigetoshi Yokoyama, Nobukazu Yoshioka, Motonobu Ichimura, "Intercloud Object Storage Service: Colony", pp. 95-98, CLOUD COMPUTING 2012, The Third International Conference on Cloud Computing, GRIDs, and Virtualization, 2012.

[5] Shigetoshi Yokoyama, Nobukazu Yoshioka and Motonobu Ichimura, "A Network-aware Object Storage Service", pp.556-561, The 2nd International Workshop on Network-aware Data Management to be held in conjunction with SC12, 2012

[6] SINET:http://www.sinet.ad.jp/index_en.html?lang=english

[7] OpenStack: http://openstack.org/