ITRI OpenStack Distribution

Chun-Chieh Huang, Guo-Hong Lai, Ling-Kang Wu, Ming-Shan Deng and Jaime Alvarez

***Abstract*—OpenStack is a new and emerging open source software stack for creating and managing clouds. It provides the flexibility and agility to meet the business need. Though the benefits of OpenStack, it still misses some important features like scalable and comprehensive bare metal provisioning, and high availability support for every OpenStack system component, etc. ITRI OpenStack Distribution is a large scale data center management system providing a turnkey, OpenStack based, and carrier-grade IaaS solutions for service providers. It is applicable for enterprise-grade private clouds, public clouds and also hybrid clouds. On top of OpenStack, ITRI OpenStack Distribution integrates two plug-ins DISCO for Cinder, and Peregrine for Neutron. Moreover, high availability on both core components of ITRI OpenStack Distribution and switches is also constructed. And, it adds Hybrid Data Center Management (HDCM) module for the business model of hybrid, physical and virtual, leasing. As a result, ITRI OpenStack Distribution incorporates all the key elements of an IaaS and is extendable by leveraging OpenStack and market-centered enhancements. Moreover, ITRI OpenStack Distribution has lower entry barriers for deployment.**

***Index Terms* — OpenStack, ITRI OpenStack Distribution, DISCO, Peregrine, Hybrid Data Center Management**

I. Introduction

THE cloud computing stack consists of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). The definition of cloud computing from NIST's viewpoint is as follows [4]: it is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., network, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. And, from the cloud computing stack viewpoint, IaaS provides the fundamental building blocks of computing resources. It abstracts hardware including servers, networks, and storages into a pool of computing, networking, and storage that are delivered as services. It does resource scheduling to manage the resources for virtual machines. It does server load balancers to manage Internet traffic. It also enables VM auto-scaling while detecting some resource usages by VM which meet the criteria. The goals of IaaS are to provide a flexible, standard, and virtualized operating environment for supporting PaaS and SaaS. PaaS is a multi-tenant architecture where multiple concurrent tenants share the same development platform. It delivers application execution services and provides an efficient and an agile approach to do deployment and to manage and scale applications in the cloud. The last one is SaaS. It delivers business processes and applications, such as Customer Relationship Management (CRM), e-mail to end users. Recently, SaaS enables cost effective value added services for many Internet of Things (IOT) applications such as intelligent factories, smart home, and healthcare.

ITRI Cloud OS [3], one of cloud solutions from CCMA in ITRI in Taiwan, integrates computing virtualization, storage virtualization, and network virtualization. ITRI Cloud OS introduces a new concept called Virtual Data Center (VDC). VDC provides an abstraction layer that allows each tenant can define their own virtual data center totally isolated from those of other tenants. A VDC consists of one or multiple virtual clusters (VCs), each of which in turn comprises multiple virtual machines (VMs). ITRI Cloud OS provides block storage solution with strong data protection through N-way replication, local and remote backup. ITRI Cloud OS supports high availability (HA) and does absolute isolation among virtual data centers. And, from Internet edge logic viewpoint, ITRI Cloud OS provides inter-VM load balancing within a VC and supports south-north bound traffic shaping. Inside ITRI Cloud OS, physical data center management (PDCM) provides comprehensive server/switch/disk/software monitoring and unified event log collection and analysis. Moreover, during run time, the security subsystems of ITRI Cloud OS enforces the policies of a VC by properly shaping network traffics and Web based filtering according to the shaping policy, the firewall policy, and WAF in load balance policy respectively. The below figure, Fig. 1, describes the components of ITRI Cloud OS.

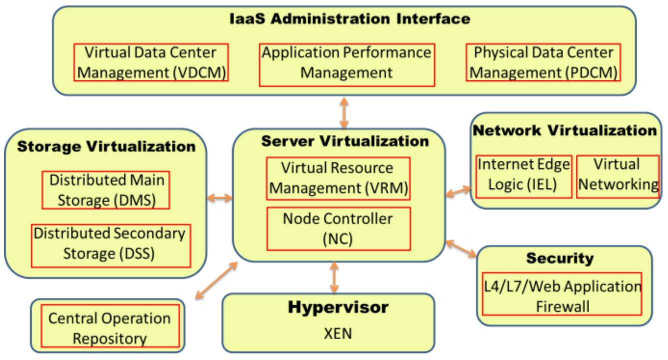

IaaS Administration Interface
Virtual Data Center Management (VDCM)
Application Performance Management
Physical Data Center Management (PDCM)
Storage Virtualization
Distributed Main Storage (DMS)
Distributed Secondary Storage (DSS)
Server Virtualization
Virtual Resource Management (VRM)
Node Controller (NC)
Network Virtualization
Internet Edge Logic (IEL)
Virtual Networking
Security
L4/L7/Web Application Firewall
Central Operation Repository
Hypervisor
XEN

Fig. 1. The components of ITRI Cloud OS

OpenStack is a collection of open source software projects such that service providers can use it to setup and run their cloud infrastructure. In 2010, both Rackspace and NASA have jointly launched OpenStack and are the key initial contributors to the stack. Austin is the first OpenStack release on October 21st in 2010. In 2011, OpenStack has three releases, they are: Bexare, Cactus, and Diablo. After that year, each year OpenStack has two releases. The core services of OpenStack are NOVA, NEUTRON, SWIFT, CINDER, KEYSTONE, and GLANCE. It also provides additional services such as identity management, orchestration, and metering accessed in the same programmatic manner through the API. Fig. 2. The OpenStack projects in Kio version and their relationships [6]
, except the red blocks, describes the OpenStack projects in Kilo version and their relationships.

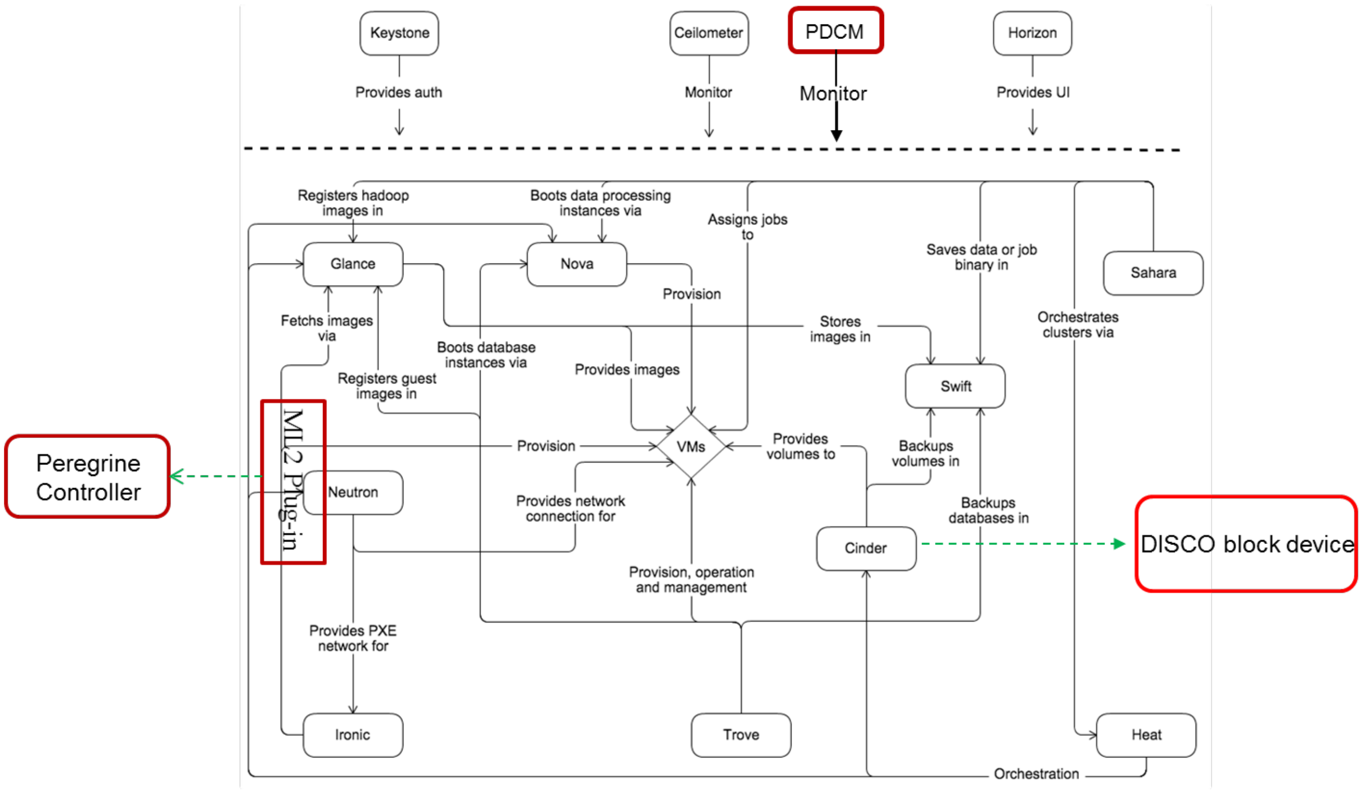


Fig. 2. The OpenStack projects in Kio version and their relationships [6]

ITRI OpenStack Distribution, IOD, combines the accumulated knowhow and deployment experience of ITRI Cloud OS and that of OpenStack, both of which have been developed simultaneously in earliest over the past six years. IOD integrates all key functionalities required to offer IaaS service into a single program. The red blocks in Fig. 2. The OpenStack projects in Kio version and their relationships [6]
, are the plug-ins developed and deployed by ITRI IOD. IOD integrates two plug-ins - DISCO (Distributed Integrated Storage with Comprehensive Data PrOtection) for Cinder, and Peregrine for Neutron - hardens high availability, and adds Hybrid Data Center Management (HDCM) module for the business model of hybrid (Physical and Virtual) leasing. The physical machines and services in IOD can be monitored through the plug-in module - PDCM. IOD deploys the IaaS system from bare-metal to the HA mode of the core components. IOD also provides the tools and technologies to abstract the underlying infrastructure in an easy and

standardized consumption model. The rest of the paper is organized as follows. Section II presents the missing in OpenStack from the service provider viewpoint. In Section III, we briefly describe the architecture of IOD, while in Section IV we present the components of IOD. Finally, Section V concludes the work and presents the future work.

II. What Is Missing In The OpenStack

OpenStack is a new and emerging open source software stack for creating and managing clouds. It provides a neutral and agnostic mechanism to manage an agile infrastructure in the data center. Over the past six years, the cloud is still an incredibly dynamic, rapidly evolving marketplace. New features and approaches are being introduced all the time. If the cloud system is vendor lock-in, we will be on the vendor's timeline to bring those new capabilities to our business. And, OpenStack provides the flexibility and agility to meet the business need. Though the benefits of OpenStack, it still misses some important features. The following subsections will describe the missing parts.

A. Scalable and Comprehensive Bare Metal Provisioning

OpenStack is a powerful but not an easy deployment IaaS system. No matter how many times you have read the official installation guide, for an OpenStack beginner, we may still take several days to deploy it. Fortunately, we learn lots of experiences in the Cloud OS in the past six years. Even we have deployment knowledge like cobbler and puppet, etc., we should do trial-and-error multiple times. To simplify the installation and to reduce the deployment time, a bare metal provisioning and an automated deployment tool is a must.

If we google bare metal provisioning, Ironic will show up. It is the OpenStack component which is for bare metal provisioning, and is officially integrated into OpenStack after the Kilo release. Its aim and use-cases are to provision the physical hardware of compute resource and rapidly deploy a cloud infrastructure, etc. [7]. Ironic service is in OpenStack controller node. And, through Ironic conductor and TFTP server, we can deploy the bare metal node. The problem is back to that bare metal provisioning and automated deployment tool is still needed for native OpenStack platform. Therefore, Ironic could not provide service without the OpenStack platform. In addition, the bare metal deployment through Ironic does not include the hardware setting on that bare metal node. The operators should do manually BIOS configuration and BMC configuration first before deployment. It would take a lot of time.

Moreover, Ironic is not good at scaling when administrators want to add additional network nodes, control nodes or compute nodes.

B. HA Support for Every OpenStack System Component

High availability (HA) is an important element in OpenStack. Through HA, it provides automated fail-over and avoids the single point of failure (SPOF). And, the system downtime, mean time between failures (MTTF), could be reduced significantly. Currently, in the paper [1], OpenStack relies on the Linux HA, corosync and pacemaker, and load balancing stack like HAProxy to manage HA.

Two problems occur while setting up HA in OpenStack. One is that HA is not a natural component of OpenStack. For an administrator who is not expertise in HA configurations, he/she has to spend lots of time to learn and do HA setting. Another one is that not all components apply Linux HA plus HAProxy to do HA deployment. We should do different combinations of high availability mechanisms to meet the HA requirements of each component in OpenStack.

C. Standard Operating Procedures (SOPs) and Tools for Change Management

OpenStack would not have been possible without the hard work of a dedicated community of contributors. OpenStack is composed of a set of interrelated projects, which make up the various components of a cloud computing platform. Components communicate with each other through lots of complex configurations. Vendors develop easy installation tools to make the OpenStack deployment more convenient and faster.

Some components will be broken after running services for a while. Even though native OpenStack has a debug mode to log errors, it is still hard to do troubleshooting and to solve them especially for those issues across multiple components at the same time. For this, standard operating procedures (SOPs) and tools (such as scripts) are needed. Some issues like services or agents in service down or error state, are easier to be solved if we apply some management tools to recover the platform immediately.

In addition, SOPs are also important for OpenStack in daily operations. As mentioned above, changing or modifying configurations in one component may affect other components. To avoid such issue, we need SOPs and some tools to implement these SOPs for changing management.

D. Scalability for Internet-facing Packet Processing

In data center, network consists of three parts, (1) the Internet facing IP routers, (2) the internal fabrics that tie together the compute servers, control servers, and storage servers, and (3) the Internet edge logic that typically sit between the IP routers and the internal fabrics and is responsible for examining, filtering, distributing and transforming incoming and outgoing packets.

The cluster based network nodes in OpenStack, with HA configuration like VRRP (Virtual Router Redundancy Protocol), is organized as a ring of servers. They could manage large user loads afforded by increasing number of network nodes. In large scale system, tenants create many virtual networks for Internet facing purposes. Currently, the cluster based network nodes serve the Internet requests in an inefficient way. This is because the loading of each network node is not considered while doing the virtual router assignment. This makes some network nodes should serve many requests and could not move those requests to others on the fly.

E. Overhead-minimizing Network Virtualization

The VMs on two different hosts could communicate with each other through port-based GRE (Generic Routing Encapsulation) tunnels. Open vSwitch (OVS) comes embedded with Linux Kernel 3.3 and up, it became the cornerstone of network virtualization. OVS can use software GRE tunnels between two hosts as a way of encapsulating traffic and creating an overlay network. With tunneling, especially in the software based tunneling, both additional overhead in the packet and the need to maintain the distributed page table are the most important issues.

The authors [2] propose OVS with DPDK (Data Plane Development Kit) to improve OVS performance in 2015 OpenDaylight summit. Though the performance can be improved after applying DPDK, the packets still need to be encapsulated. In addition, we only can get significant performance improvement when a large number of packet sizes are small. The assumption flow patterns are not the same as the real ones in cloud environment.

F. Physical Data Center Administration Tool

Physical data center management (PDCM) is a hardware monitoring system and, moreover, it is a service management system. It monitors both hardware and low level software components simultaneously, which can send real time notifications to alert operational team and tenants in a timely fashion to minimize the delay of response when system dysfunction occurs. Currently, OpenStack has 19 projects. None of them are doing the same work. It is hard to monitor the health states on both hardware and software components.

In addition, log analysis is also an important feature in the log system. Firstly, it draws the system graph of the IaaS platform. Secondly it does the event correlation analysis from the collected logs. Finally, from the system graph and the event clues, the log analysis could easily figure out the possible failure point.

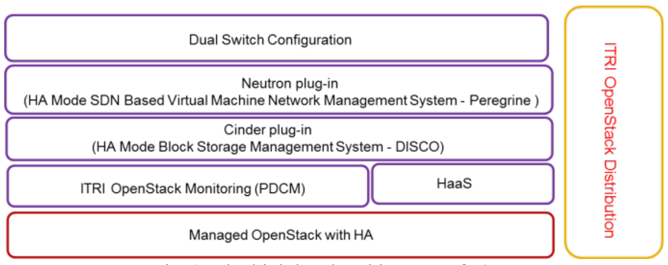
G. Physical Machine (Hardware) Leasing Service

Hardware as a Service (HaaS) is a service model that allows a tenant could lease physical machines. With HaaS, we could provide physical machine leasing service. The procedures of leasing service are leasing, provisioning and management.

After applying the HaaS service in OpenStack, the service provider could able to provide not only virtual machines, but also physical machines leasing service, which we called Hybrid Data Center Management (HDCM).

III. ITRI OPENSTACK DISTRIBUTION ARCHITECTURE

ITRI OpenStack Distribution (IOD) incorporates all of the key elements of the IaaS operation, such as computing, storage, networking, load-balance, security, and management interface capabilities. This system can help us on IaaS solutions by shielding customers from the tedious tasks of resource integration, configuration and validation; organizations can therefore focus on their value-added to meet client’s business demands for IaaS, PaaS, and SaaS services. Fig. 3 depicts the high level architecture of IOD.


Dual Switch Configuration
Neutron plug-in
(HA Mode SDN Based Virtual Machine Network Management System - Peregrine)
Cinder plug-in
(HA Mode Block Storage Management System - DISCO)
ITRI OpenStack Monitoring (PDCM)
HaaS
Managed OpenStack with HA
ITRI OpenStack Distribution
Fig. 3. The high level architecture of IOD

In IOD, we have the following features, they are: managed OpenStack with HA capability, physical data center management, two plug-ins - DISCO for Cinder, and Peregrine for Neutron - hardens high availability on these two plug-ins, and HDCM module for the business model of both Physical and Virtual leasing. In addition, dual switch configuration is for network high availability.

The below picture, shows the IOD stack. The yellow blocks are OpenStack itself and dotted blocks are from ITRI. The lower left corner is related to Peregrine plug-in, the lower right corner is about DISCO plug-in, and the upper right corner is for PDCM. The next section will describe these features.

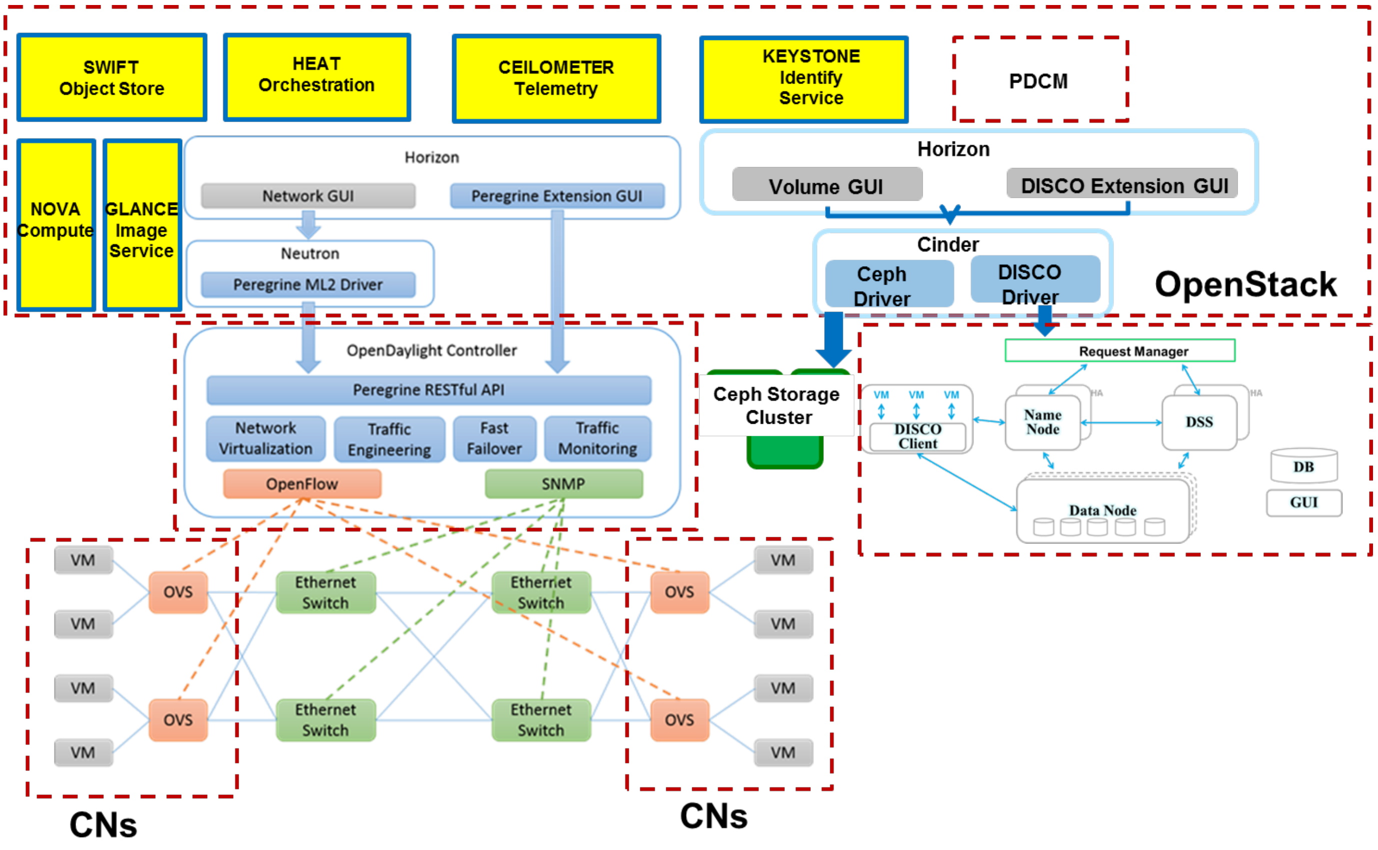


Fig. 4. ITRI OpenStack Distribution stack

TABLE I describes the missing parts in native OpenStack and the corresponding solutions in IOD.

TABLE I
The Missing Parts in Native OpenStack and the Corresponding Solutions in IOD

OpenStack Missing Items	Native OpenStack	ITRI OpenStack Distribution
Scalable and comprehensive bare metal provisioning	No automatic deployment tool support	With BAMPI, MAAS, and Juju, IOD supports automatically deployment
HA support for every OpenStack system component	Manually configure HA	IOD does automatically HA configuration
Standard Operating Procedures (SOPs) and tools for change management	Lack of SOPs and management tools for change management	IOD provides SOPs and management tools to recover the failure nodes
Scalability for internet-facing packet processing	Network nodes could not support load share on internet-facing payloads	IOD will do this feature in the road map
Overhead-minimizing network virtualization	OpenStack provides tunnel-based network virtualization	Peregrine in IOD provides VLAN-based network virtualization
Physical data center administration tool	No package support	PDCM in IOD provides both hardware and components monitor
Physical machine (hardware) leasing service	Ironic service cannot do bare metal deployment from scratch	HaaS service in IOD can do bare metal deployment from scratch

IV. Components in ITRI OpenStack Distribution

This section describes the features of IOD. They are (1) scalable, manageable and easy deployment; (2) HA support for every component; (3) Cinder plug-in – DISCO; (4) Neutron plug-in – Peregrine; (5) Physical data center management module; and (6) Hardware as a service.

A. Scalable, Manageable and Easy Deployment

To achieve true automatic deployment, IOD integrates BAMPI (Bare Metal Provisioning from ITRI), MAAS (Metal as a Service), and Juju into deployment processes. The work starts from bare metal to IOD services ready with a few clicks. Setting up hardware, network, firmware in BIOS and first OS on fresh servers correctly is a pain point for operators. BAMPI does the above work with ease way. And, BAMPI will be introduced in detail in the HaaS subsection. Fig. 5 depicts the

IOD deployment. Through this deployment process, IOD can do automatically deployment from bare metal to IOD services ready. From the network architecture, it shows that the IOD is deployed as HA mode and the management, storage, and external networks are separated.

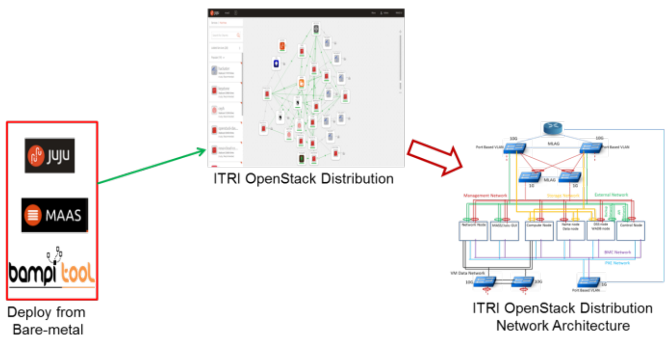


Fig. 5. The IOD deployment from bare metal to IOD services ready

As attested by our customers, IOD reduces the deployment time in the field significantly (tens of times). As a result, the deployment of IOD with indispensable HA mode becomes extremely easy. Once the system is up running with applications deployed on top of IOD, PDCM, in collaboration with Horizon, will provides real-time monitoring of hardware and low-level software components, event collection and alarms. Hence, the heath of the system is assured with status available at the fingertips of the administrators and tenants.

From the OpenStack survey report of OpenStack foundation in October 2015, Ubuntu is the most popular operating system for OpenStack in the world [5]. In IOD, we use Ubuntu MAAS as bare metal provisioning tool and Ubuntu Juju as the OpenStack services deployment tool and those of them are supported by Canonical under the Ubuntu Advantage service program [8][10]. MAAS and Juju will be described in detail in the following subsections:

1) MAAS

MAAS is the bare metal provisioning tool and treats physical machines as virtual machines in cloud [8]. MAAS provides RESTful API, Web UI, and command-line to achieve zero-touch OS installation. Currently, MAAS supports Ubuntu, CentOS, Windows and even customized image for OS installation. After a sequence of operations in MAAS server, like enlist and commission, a node will be in ready state for deployment. In addition, MAAS provides flexible operations like the change management which can be done on the fly. The main components of MAAS are region controller, cluster controller(s), nodes as shown in Fig. 6. And, MAAS manages the nodes for IaaS. In the small testing environment, we can setup both region controller and cluster controller on the same server.

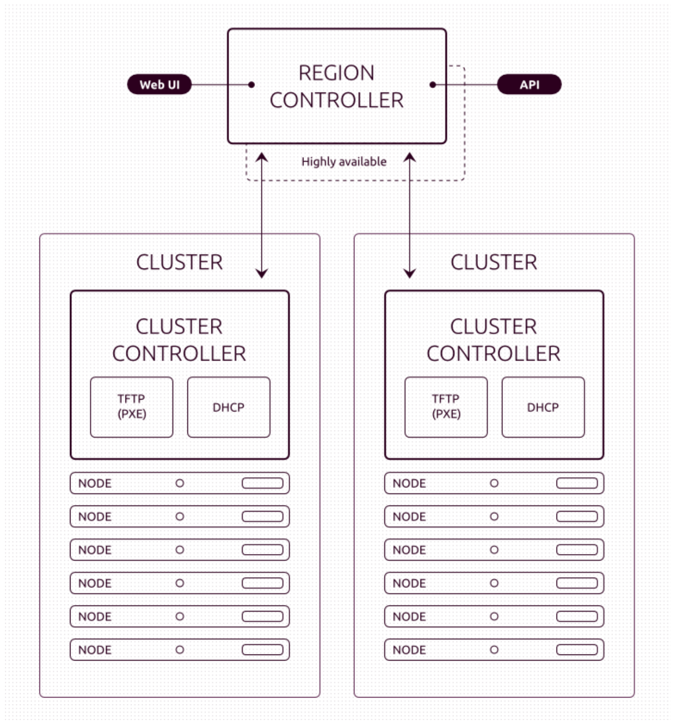


Fig. 6. MAAS components with structure diagram [9]

2) Juju

Juju is an open source based modelling tool integrated with MAAS [10]. Some similar tools like Puppet and Chef can do the same things as Juju does. However, Juju not only does the service deployment but also supports dynamic configuration, some templates for common services deployment, and the maintenance of deployed services through Juju GUI.

In addition, Juju has a great ecosystem. It has full of charms that can deploy all kind of different services which not limited to OpenStack components. Therefore, the combinations of MAAS and Juju with GUI interface enable us to deploy a node and the corresponding services easily.

To setup Juju, Juju-core service will be installed and configurations should be setup in advance [11]. Then Juju bootstrap chooses one of ready nodes in MAAS to deploy Juju API service. Juju-GUI services are used to manage charm installation shown in Fig. 7.

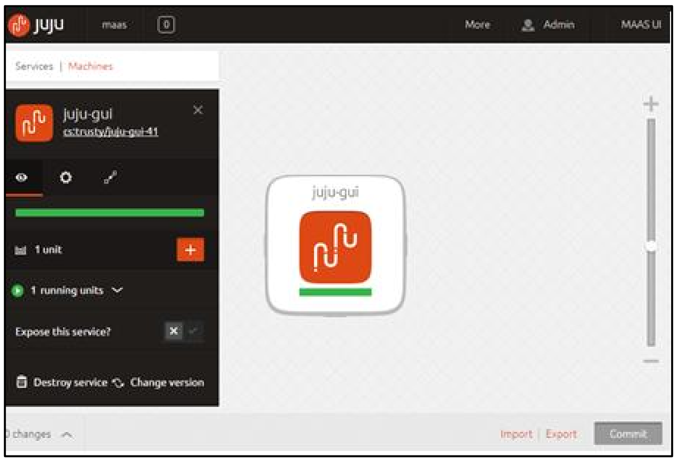


Fig. 7. Juju-GUI example

OpenStack components can be easily deployed into the platform through Juju-GUI or command line interface. The components can be deployed into different modes like LXC (Linux Container) or on the physical host [12]. Besides, components with HA could also be easily deployed through the same operations with different yaml files.

B. HA Support for Every Component

IOD assures overall system's high availability through securing high availability on each component. IOD applies different types of HA into the cloud system to render different scenarios. The HA type decision is based on the best practice and the nature of each component. For example, a combination of Linux HA and HAProxy serves the high availability well for OpenStack stateless API module, while a Galera cluster is used for MySQL, etc. In addition, dual switch configuration is adopted to provide hardware based redundancy in IOD. These settings allow the cloud system to survive under any single point failure conditions.

C. Cinder Plug-in – DISCO

DISCO is a cinder plug-in, and a high-performance distributed block storage solution which provides a distributed storage pool on a set of commodity hardware that can scale out with ease and on demand. DISCO provides block-level storage service to VMs similar to iSCSI. Name node, DSS node, DISCO client node, and Data node are the components of DISCO. Fig. 8 depicts DISCO components and the corresponding relationships among them in IOD. The DISCO Cinder driver is responsible for sending and receiving requests to and from the Cinder and is certificated in OpenStack Mitaka formal release. The request manager is an API server as an entry point of DISCO and provides a set of storage APIs to manage DISCO. DISCO name node maintains various disk-based mapping tables to map large number of blocks in different address-spaces, and its workload consists of a large number of requests that either read an existing entry, write a new entry, or modify an existing entry in the mapping-table. DISCO DSS node helps in backing-up data from primary storage to secondary storage at periodical intervals. And, DSS performs several incremental backups and full-backups. DISCO client node, located in each computing node, captures the I/O requests. For an incoming I/O request, it will consult DISCO name node for address mapping and forward I/O request to mapped data nodes. Finally, DISCO data node is responsible for storing tenants' payload.

The features of DISCO include high availability, self-healing, N-way replication, de-duplication, fast volume clone, datacenter-to-datacenter/wide area data backup (WADB), and user friendly GUI for monitoring, management, and diagnostic shown in Fig. 9.

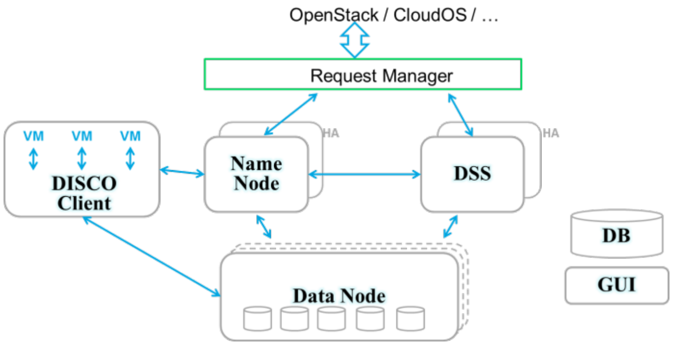
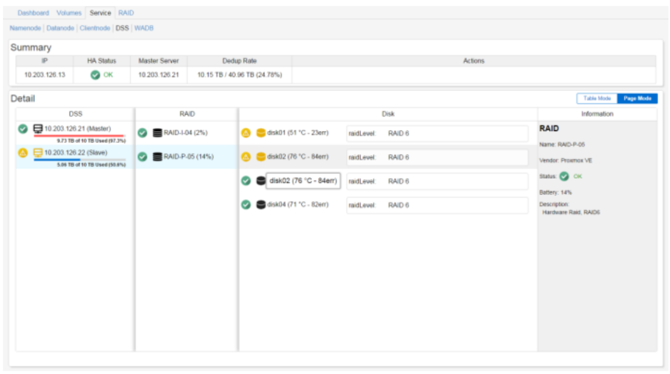


Fig. 8. DISCO components and the corresponding relations among them



Summary

IP	HA Status	Master Server	Dedup Rate	Actions
10.203.126.13	✓ OK	10.203.126.21	10.15 TB / 40.96 TB (24.78%)	

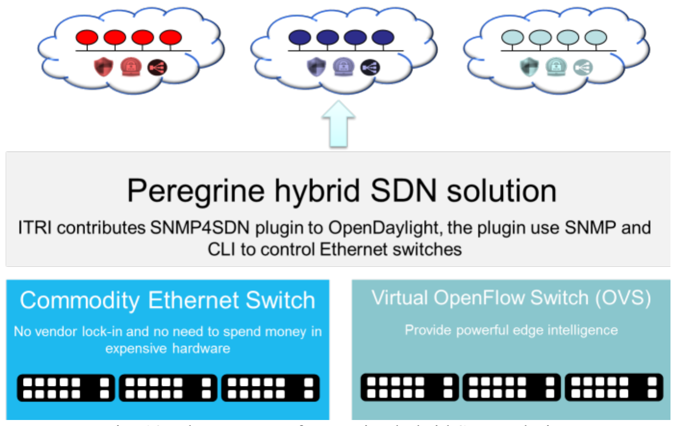
Detail

DSS	RAID	Disk		Information
✓ 10.203.126.21 (Master) 9.73 TB of 10 TB Used (97.3%)	✓ RAID-I-04 (2%)	⚠ disk01 (51 °C - 23err)	raidLevel: RAID 6	RAID
⚠ 10.203.126.22 (Slave) 5.06 TB of 10 TB Used (50.6%)	✓ RAID-P-05 (14%)	⚠ disk02 (76 °C - 84err)	raidLevel: RAID 6	Name: RAID-P-05
		✓ disk02 (76 °C - 84err)	raidLevel: RAID 6	Vendor: Proxmox VE
		✓ disk04 (71 °C - 82err)	raidLevel: RAID 6	Status: ✓ OK
				Battery: 14%
				Description: Hardware Raid, RAID6

Fig. 9. GUI based diagnostic tool

D. Neutron Plug-in – Peregrine

Peregrine is a neutron plug-in which, among other salient features, supports dynamic traffic engineering and fast failover. Fig. 10 illustrates the concept of Peregrine hybrid SDN solution. As a result, it significantly increases network resource utilization and throughput in an intelligent and responsive way. Peregrine achieves this using VLAN-based network virtualization, as opposed to the default way of tunneling by OpenStack. This has the advantage of allowing the use of commercial Ethernet switches without OVS capability – with drastic cost saving. This "lightweight" approach has additional benefits of lower CPU load as well as transparency of the link traffic contents. Fig. 11 draws the network topology according to current network connectivity. A comprehensive GUI-based debugging and monitoring utility has also being developed as shown in Fig. 12, where, for example, administrators can be alarmed for overloaded link, who can then drill in to investigate the information such as the VM end points, VALA tags, bi-directional flow and even packets, etc. An intelligent pre-calculation of fail-over paths under any single point of link failure allows Peregrine to complete fail over within one second – a fraction of any current practice.



Peregrine hybrid SDN solution
ITRI contributes SNMP4SDN plugin to OpenDaylight, the plugin use SNMP and CLI to control Ethernet switches
Commodity Ethernet Switch
No vendor lock-in and no need to spend money in expensive hardware
Virtual OpenFlow Switch (OVS)
Provide powerful edge intelligence

Fig. 10. The concept of Peregrine hybrid SDN solution

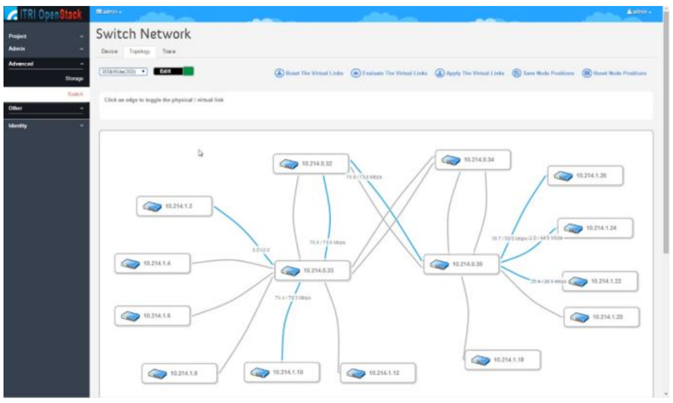


Fig. 11. The network topology

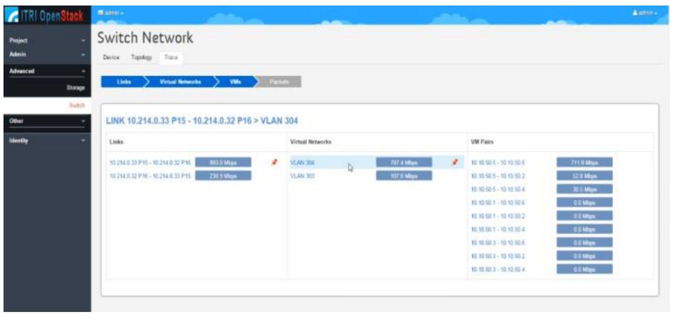


Fig. 12. GUI-based debugging and monitoring utility

E. *Physical Data Center Management Module*

PDCM is one of IOD components. Inside PDCM, by constant checking the operation status and performance properties (such as CPU utilization, etc.), all physical devices will be monitored over time. It also collects logs from various components in IOD into a central repository, analyzes them and, as a result, is able to send alerts to allow operators to respond in no time under significant system faults.

In addition, PDCM provides restful API interface for programming flexibility. PDCM could monitor many IOD infrastructure endpoints; it implies that the components, for examples, nova services, neutron agents, etc., are constantly monitored over time. PDCM authenticates itself through Keystone system and hence is authorized to communicate with all components of IOD to realize its role. This is why IOD is shipped and deployed with PDCM. After setting, PDCM can immediately start monitoring IOD. Fig. 13 illustrates PDCM request flow in IOD.

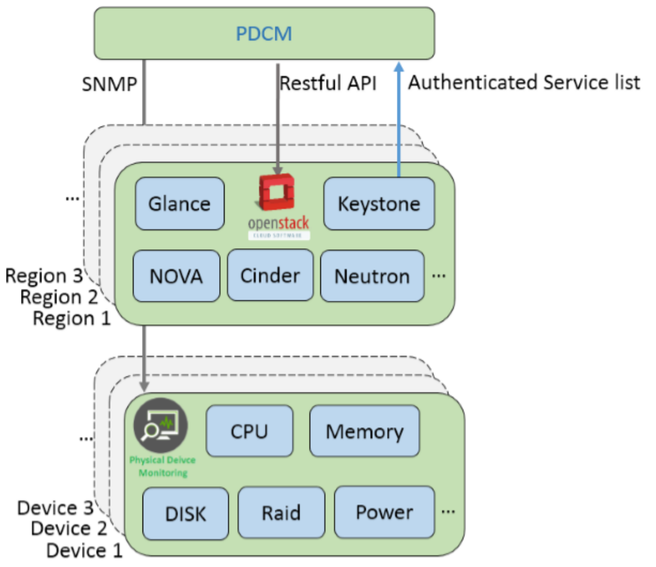
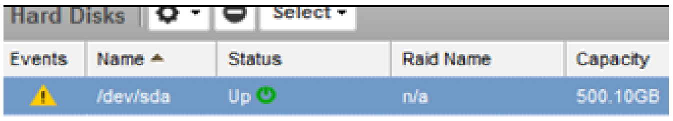


Fig. 13. PDCM request flow in IOD

The system administrator can also develop new plugins for PDCM to extend its functionality and features. For example, we can develop new installable plugins that can: monitor new types of devices that are currently not supported; monitor additional OpenStack components that are currently not supported; and create additional views that display certain visual data, etc.

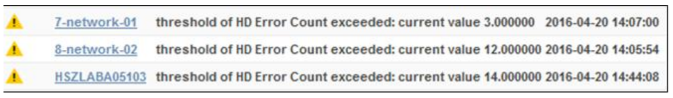
With PDCM, administrators can make sure that both environment and infrastructure are performing in the right way. And, if problems or inconsistencies arise, we can pinpoint the source and fix it in a rapid manner. For example, if a hard disk drive in a device is experienced some errors or malfunctioning, we can easily see this in PDCM displayed as visual alerts (as Fig. 14).


Hard Disks | Select

Events	Name	Status	Raid Name	Capacity
⚠	/dev/sda	Up	n/a	500.10GB

Fig. 14. Disk error shown in PDCM

At the same time, these alerts also create Events stored in PDCM and give more detailed description of what is happening and where it is. Fig. 15 shows the disk error events and could be shown in PDCM. In addition, these events can be configured as E-mail alerts and be sent to administrators.



⚠	7-network-01	threshold of HD Error Count exceeded: current value 3.000000	2016-04-20 14:07:00
⚠	8-network-02	threshold of HD Error Count exceeded: current value 12.000000	2016-04-20 14:05:54
⚠	HSZLABA05103	threshold of HD Error Count exceeded: current value 14.000000	2016-04-20 14:44:08

Fig. 15. Disk error events shown in PDCM

The advantages of PDCM are as follows:

1) *Hardware Device Monitoring*

PDCM is able to monitor the hardware devices in IOD. The monitoring items include CPU utilization , memory utilization , network routes , interfaces , file systems , hard disk state , raid card state , power consumption , fan , current and voltage , and thermal.

2) OpenStack Resources and Services Monitoring

PDCM is able to monitor the OpenStack resources such as: regions , availability zones , instances , hosts , hypervisors , flavors , images , networks , subnets , routers , ports , floating IPs. PDCM also provides Nova, Cinder and Neutron service monitoring. In addition, the service states of both DISCO and Peregrine plug-ins are also monitored by PDCM.

3) Event Management

With a single event console, PDCM allows you to view all events throughout your entire cloud environment. PDCM is also able to send alerting mail to different person according to severity of the events.

4) Easy to Add and Monitor Multiple OpenStack Clouds

As a data center monitoring entry point, PDCM provides an easy way to allow administrators to add new OpenStack cloud to be monitored. Users only need to put their OpenStack keystone information and then PDCM would automatically get all resources including physical devices and then start to monitor them.

F. Hardware as a Service

As described in the previous section, in Section II, Hardware as a Service (HaaS) provides leasing, provisioning, and management services. Fig. 16 illustrates the concept of HaaS. For leasing, the administrator can do inventory and resource management. For example, according to the leasing request, we can allocate a set of physical machines called as a PDCI (Physical Data Center Instance) to a registered tenant. Each tenant could own multiple PDCIs. And, each PDCI has its own network setting.

For provisioning, server, network and storage provisioning are included. On every leased server, server provisioning enables the user to deploy application and middleware components, install OS, configure and check BIOS and BMC. Network provisioning provides automated VLAN-based configuration for the PDCI. Storage provisioning can create volumes on a shared storage server and associate the volumes with leased servers.

In this paper, we integrate both BAMPI and Peregrine into IOD HaaS service. BAMPI is a powerful tool to do machine setting and OS provisioning. It could significantly reduce the time for doing the above work. In the real customer environment for 20-server provisioning, we reduce the setting time from one day to 3 hours. And, BAMPI can reduce time consuming significantly when we increase the number of requests for provisioning servers.

Regarding the management part, the administrator can monitor machine status, network topology and resource management in the PDCIs. Furthermore, in HaaS project, PDCI isolation in different tenant is one of the important items. Thus, the PDCI in tenant A could not affect another PDCI in tenant B.

With IOD, we are able to provide not only virtual machines, but also physical machines leasing service, which we called Hybrid Data Center Management (HDCM). The customer can lease both virtual and physical machines according to their needs. Additionally, HaaS also provides a comprehensive resource management and monitoring service such that we can easily install, run and get a quick view of their PDCI status.

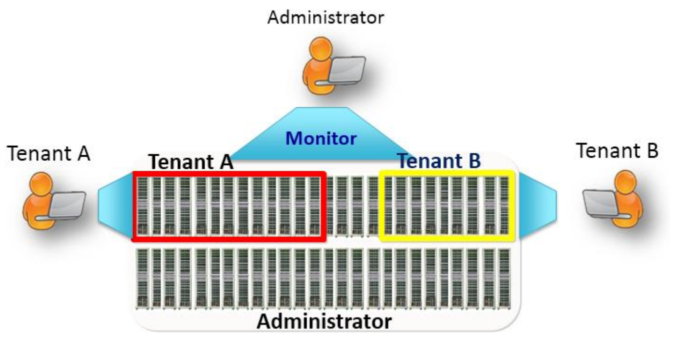


Fig. 16. The concept of HaaS

V. Conclusion

IOD offers all of the advantages for IaaS from a first-class open source project - OpenStack – to prevent the customers from proprietary vendor lock-in; has the full support of a seasoned team; is enhanced by two plug-ins – Peregrine for Neutron and DISCO for Cinder – with additional features and performance enhancement; supports hardened high availability and real-time upgrade. The ITRI IOD team is dedicated to serve the industry of private, public and hybrid clouds and serve as the solid foundation for PaaS, SaaS and various vertical applications which need scalability with flexibility and cost-effectiveness.

Acknowledgment

We thank those whom participated in ITRI/ICL for providing review comments for this paper.

References

[1] D. Mane, "Building a high availability – OpenStack", *International Journal of Engineering Research and Applications*, 3(4), Jul-Aug 2013, pp.269-277.

[2] S. Mishra, E. Multanen, D. Zhou, and R. Koodli, "Enabling DPDK accelerated OVS in ODL and accelerating SFC", presented at OpenDaylight summit in 2015. Available: http://events.linuxfoundation.org/sites/events/files/slides/ODL%20Summit%202015%20DPDK.pdf

[3] T.C. Chiueh, E. J. Chang, R. Huang, H. Lee, V. Sung, M. H. Chiang , Security considerations in ITRI Cloud OS, *2015 International Carnahan Conference on Security Technology (ICCST)*, Sep 21st – 24th, 2015, Taipei, Taiwan.

[4] NIST Cloud Computing Program. [Online]. Available: http://www.nist.gov/itl/cloud/

[5] OpenStack user survey: A snapshot of OpenStack users' attitudes and deployments. pp. 32, October 2015. [Online]. Available: http://www.openstack.org/assets/survey/Public-User-Survey-Report.pdf

[6] OpenStack Installation Guide for Ubuntu 14.04. [Online]. Available: http://docs.openstack.org/kilo/install-guide/install/apt/content/ch_overview.html

[7] The OpenStack Ironic installation guide website. [Online]. Available: http://docs.openstack.org/developer/ironic/deploy/install-guide.html

[8] The Ubuntu MAAS website. [Online]. Available: http://maas.io/

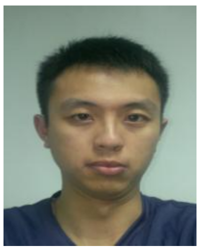
[9] The Ubuntu MAAS orientation website. [Online]. Available: ttps://maas.ubuntu.com/docs/orientation.html#setup

[10] The Ubuntu Juju website. [Online]. Available: http://www.ubuntu.com/cloud/Juju

[11] The Ubuntu Juju Quick Start website. [Online]. Available: http://maas.ubuntu.com/docs/Juju-quick-start.html
[12] The Ubuntu wiki - OpenStackHA. [Online]. Available: https://wiki.ubuntu.com/ServerTeam/OpenStackHA

Chun-Chieh Huang received a Ph.D. degree from the Institute of Information Management at National Chiao Tung University, Taiwan. His research interests include network security, cloud security, data protection, network virtualization, and large scale system design. He is a senior engineer at ITRI now.

Guo-Hong Lai is currently working as deputy engineer in ICL, ITRI and mainly focus on deployment of Cloud OS and OpenStack. Lai graduated from Electrical Engineering department, National Taiwan University of Science and Technology.



Ling-Kang Wu is currently working as deputy engineer in ICL, ITRI. Wu graduated from Compute Science and Information Engineering department, National Central University.

Ming-Shan Deng is currently working as a software research and developer in ICL, ITRI. Deng focus on developing the monitoring and management system for both hardware and software component in ITRI OpenStack Distribution.

Jaime Alvarez is currently working as a software engineer in ICL for ITRI's OpenStack Distribution's PDCM platform. Mr. Alvarez is a Masters graduate from National Tsing Hua University and also focuses on development of web applications and web services.