Next Generation Clouds, The Chameleon Cloud Testbed, and Software Defined Networking (SDN)

Joe Mambretti
International Center for Advanced Internet Research
Northwestern University
Chicago, USA

Jim Chen
International Center for Advanced Internet Research
Northwestern University
Chicago, USA

Fei Yeh
International Center for Advanced Internet Research
Northwestern University
Chicago, USA

Abstract— **Next generation clouds, based on highly programmable, high performance networks, especially those supported by Software-Defined-Networking (SDN) have attracted significant interest by research communities. In recognition of the increasing importance of advancing cloud services and technologies, especially for providing Internet services, the US National Science Foundation (NSF) established a project, the NSFCloud initiative, to enable the computer science research community to develop and experiment with novel cloud architectures and create new, architecturally enabled innovative applications for cloud computing through empirical research experimentation by using large scale distributed cloud testbeds. This paper provides an overview of one of those testbeds, the Chameleon Cloud testbed, with an additional description of the integration of that testbed with high programmable, high performance networks, based on SDN. The Chameleon project is designing, deploying, and operating a large scale, highly distributed experimental environment for empirical cloud research, integrated with high programmable networks as a foundation resource.**

Keywords— Cloud testbeds, NSFCloud, Chameleon Cloud, federated clouds, programmable networking, network testbeds, Software Defined Networking (SDN), Software Defined Network Exchanges (SDXs), distributed clouds, Software Defined Infrastructure (SDI), highly distributed environments, Global Environment for Network innovations (GENI), international SDXs, programmable clouds

I. Introduction

In the last few years, cloud based services have become important resources for all major economic systems. Therefore, increasingly, cloud based services are an inherent part of the critical infrastructure for all key sectors of 21st century societies, which are now primarily based on the management and transfer of large scale digital information. Cloud services are now ubiquitous as a support system for all major 21st-century economic activities. Consequently, it is important to enable cloud services and technologies to continue to evolve. However, a persistent barrier to such further advancement has been the lack of large-scale experimental cloud research platforms that could be used to conduct such research.

In recognition of the increasing importance of advancing cloud services and technologies, especially for providing Internet services, the US National Science Foundation (NSF) established a project in 2014, the NSFCloud initiative, to enable the computer science research community to develop and experiment with novel cloud architectures and create new, architecturally enabled innovative applications for cloud computing through research experimentation by using large scale distributed cloud testbeds. Such applications include those used by scientific communities, including those in the important area of cyber-physical systems, which integrates computation with physical infrastructure, real-time systems, and safety-critical applications such as those used to support medical devices, power grids, and transportation systems. The NSFCloud program was developed by the NSF's Directorate for Computer and Information Science and Engineering (CISE) as part of its Research Infrastructure: Mid-Scale Infrastructure program. In part, this program was established to complement the NSF's Global Environment for Network Innovations (GENI) testbed, which was established for network science researchers [1]. The success of this project provided an informative context for the design of the NSFCloud initiative. In early 2014, the NSF distributed a solicitation for proposals to create two large scale distributed testbed environments which could be used to conduct the required empirical cloud experimentation. Subsequently, two major awards were announced, one for the Chameleon Cloud testbed and one for CloudLab [2, 3].

This paper provides an overview of the Chameleon Cloud testbed, with an additional description of the integration of that testbed with high programmable, high performance networks, based on Software Defined Networking/OpenFlow (SDN). The Chameleon project is designing, deploying, and operating a large scale, highly distributed, experimental environment for cloud research. The two primary sites for the physical resources of this facility are co-located at the University of Chicago and The University of Texas at Austin, which are interconnected by 100 Gbps paths, provided by national, regional, and enterprise research and education networks. The International Center for Advanced Internet Research (iCAIR) at Northwestern University in Chicago is

978-1-5090-0144-6/15 $31.00 © 2015 IEEE
DOI 10.1109/ICCCRI.2015.10

the lead institution for Chameleon Cloud networking. In addition, with other Chameleon partners, iCAIR has undertaken to develop a networking testbed for developing and prototyping cloud networking architecture, protocols, and technology to experiment with concept that will inform the Chameleon environment. This testbed will not be available to experimenters. Other Chameleon Cloud partners include The Ohio State University and the University of Texas at San Antonio.

II. Chameleon Facilities

The Chameleon Cloud is comprised of 18,000 processor cores in more than 500 cloud nodes with 5 petabytes of storage. Researchers using Chameleon, are able to find, integrate, and configure assemblages of resources – "slices" – across the Chameleon environment as customized clouds, using either pre-defined or custom software stacks to develop and test the attributes of different type of cloud architecture to address challenges such as efficiency, optimization, usability, scalability, security, machine learning, and adaptive operating systems. In addition, the environment is being used to create new types of cloud based services, including those for data intensive science.

Chameleon supports "bare-metal access" to researchers, an alternative to only using virtualization technologies that are commonly deployed in order to share cloud hardware. By providing bare metal access also, Chameleon enables researchers to design, develop, and experiment with innovative virtualization technologies – that is it enables them to create new virtualization techniques that are not deployed in today's clouds, for example, developing and experimenting with new types of fundamental primatives to be used as building blocks for novel layers of abstraction. Also, unlike standard cloud environments, Chameleon is deeply instrumented to allow for individual researchers to profile and view their custom environments in a highly granulated way, with detailed traces for all processes, with an exceptional degree of scientific fidelity, a key consideration for an empirical research environment. Chameleon support for experimental repeatability and reproducibility is a major project goal because this attribute is an absolute requirement for empirical researchers.

Within its environment, Chameleon supports heterogeneous architecture, including those related to low power processors, general processing units (GPUs), and field-programmable gate arrays (FPGAs), storage devices, multiple network resources, including SDN/OpenFlow, and specialized interconnects and instrumentation devices. The Chameleon environment is sufficiently flexible to enable the integration of other types of resources also. An project expectation is that many novel devices will be created during the multi-year lifecycle of this initiative that will be incorporated into the Chameleon testbed.

Among the reference architecture contexts for the Chameleon Cloud architecture have been multiple programmable infrastructure established for highly distributed multi-domain Grid environments, including the FutureGrid, Grid5000, and related environments. The Grid initiatives advanced the concept of highly programmable networking in Grid environments [4].

III. Design Objectives Responding To Research Community Requirements

The design of Chameleon has been shaped by considerations of different research communities [5]. Three primary communities have been defined by the project organizers. For example, one group consists of researchers who are primarily focused on higher level considerations (from an architectural perspective), such as services, applications, and educational activities. These experimenters primarily require access to persistent, reliable standard cloud configurations and options for deploying pre-defined environments implemented as virtual machines (VMs). They also occasionally would like to implement their own pre-packaged VMs, incorporating higher-levels services, for example, using container services such as Docker, which an open platform that allows developers and system administrators to create, and run distributed applications, based on the Docker Engine - a portable, lightweight runtime and packaging tool, and Docker Hub, a cloud service for applications sharing and workflows automation.

A second group consists of researchers who would like to select from among several pre-defined environments in a reserved partition, for example, those deployed as hardware images having configurations integrated with basic services, and those who would like to use specialized clouds, such as highly secure clouds, highly available clouds, federated clouds, special purpose clouds, etc. Such services are made available as specialized implementations of OpenStack, Hadoop, and related stacks.

A third group of researchers consists of those who want to investigate topics that are closer to infrastructure foundations, for example, new types of virtualization technology, novel operating systems, non-standard networking, novel devices and components, and infrastructure level experiments. These researchers would like to deeply change many aspects of these cloud foundation resources, for example, configuring directly and reconfiguring the software stack, undertaking bare metal provisioning, selecting OSs, configuring and reconfiguring those OSs, configuring and reconfiguring storage systems, and file systems, and configuring and reconfiguring network devices, paths and characteristics, including the interconnects and topologies. A reference architecture for these types of experimental network capabilities has been the GENI environment and related network testbeds, which are also based on SDN/the OpenFlow and related programmable networking techniques.

The Chameleon Cloud provides mechanisms that allow experimenters to easily create images, implement them initially in a limited partition, test them for that implementation, and then further implement them at a lerger scale within wider deployments. These experimenters have root access, and they can reboot common or customized OSs, configure and reconfigure additional components of the stack, implement a limited number of changes to the BIOS, and access such management capabilities as manipulating the power on and off components, and they also can use console access. Basically, an experimenter's environment consists of allocated nodes that are isolated on a specific dedicated L3 or L2 private network and a gateway node, in part, to enable ssh connections to the Chameleon account, accessible by the public network and the dedicated private network.

In the Chameleon environment, fundamental processes are those that, within a policy context, discover quantified resources that can be allocated, claim such resources, integrate them, deploy customized software environments based on the resources, implement customized software configurations, utilize the selected resources for specific periods of time, and then measure results through analytic processes. Consequently, a resource reservation system is provided, along with requisite authentication/authorization processes and tools for utilization monitoring and analytics. In part, supporting these capabilities requires resource partitioning and network isolation for the individual environments.

IV. Chameleon Experimenter Workflows

The next section of this paper describes a general Chameleon Cloud experiment workflow [6]. Below is a general schematic depicting the experiment workflow processes. (Figure 1)

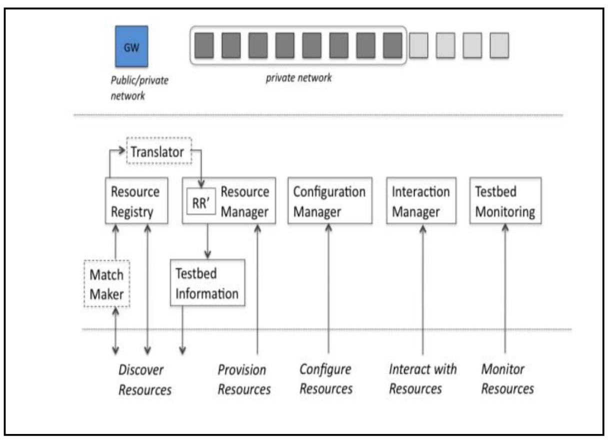

Figure 1: Conceptual Diagram of Chameleon Experimental Environment Creation Workflow and Services Used for Environment Creation and Management.

In this figure, the Chameleon experimental environment is shown as a collection of nodes that are isolated on a unique L3 or L2 private network together with a gateway node that is on the public as well as a private network. The gateway node allows experimenters to ssh to their Chameleon account and access storage partitions of the global storage via the Chameleon account. This gateway also allows experiments to ssh to the lease nodes using the private network and to transfer data. The gateway node can be allocated on a per-user/lease basis, or it can be shared among experimenters.

1) Access to Chameleon: Initial processes are used to enter the portal into the Chameleon environment, to establish a Chameleon account, using appropriate policy based procedures, which allow access to the overall environment, including tool options, an overview of available resources, resource management procedures, etc
2) Resource Discovery: A key initial step is to view the range of potential resources within the environment. To determine whether or not resources exist, an experimenter browses a form of catalog – the Resource Registration Component (RR), which hosts representations of the Chameleon resources at a highly granulated level, including the type, version, and other attributes.
3) Availability: After browsing the resource catalog, determining what resources are required, discovering desired resources, the experimenter next determines whether or not the required resources are available, by using using the Resource Management Component (RM). This component of the Chameleon system provides data on reservations and options for selecting resources and reserving them for specific periods of time, which are determined by the experimenter. This process returns a reservation ID that is then used to manage the resources that have been selected.
4) Environment Use With the Configuration Manager: At the time that the selected resources become available under the terms of the reservation, the experimenter can then configure and reconfigure the environment using the Configuration Management Component. Options exist for selecting from an image repository, persevering customized images, returning to earlier images, and changing images. These processes are supported by a configuration server, which manages image distribution.
5) Interaction Manager: As noted, a set of experimenters require special capabilities that provide for more options than root, such as options for reconfiguring BIOS (e.g., enable/disable hyperthreading), customizing I/O components, changing accelerator card settings, powering computers on and off, obtaining access to consoles, providing for specialized network capabilities, and performing other low level functions. These tasks are accomplished through the Interaction Manager.
6) Testbed Metrics Collection (Monitoring): Testbed instrumentation is a particularly important component of the Chameleon Cloud testbed. As noted, Chameleon was designed as testbed to support empirical experimentation, and, therefore, reproducibility and repeatability of specific experiments with scientific fidelity are key requirements. Therefore, it is important to have common, specialized, and custom sensor mechanisms for gathering highly granulated metrics for all aspects of the experimenter environment, including hardware nodes, software processes, network devices, storage systems, file systems, PDUs, and other components. Custom sensors can be used within the

testbed environment to monitor specialized capabilities and processes that are unique to the experiment being conducted.

V. Chameleon Cloud Network Architecture

Networking for clouds is a particularly important area that has been evolving especially quickly in recent years, and to a large degree is motivating the transition to network programming based on SDN/OpenFlow [7]. Similarly, the Chameleon Cloud network architecture is being designed, in part, with GENI and related network testbeds as a reference architecture, as a highly programmable, deeply instrumented network based on SDN/OpenFlow.

SDN has fundamentally transformed networking services and infrastructure, and for the foreseeable future it will continue to motivate major network advances. Therefore, SDN is an essential experimental resource component for the Chameleon environment. Also, the design of Chameleon is directly to closely integrating compute and networking resources into one seems environment and not separate resources. In part thos goal can be accomplished because the SDN architecture utilizes virtualization techniques that allow low level functionality to be expressed as fairly high level abstractions. Consequently, networking services and foundation resources can be significantly more programmable and dynamic than when using traditional approaches.

Today, many new benefits are being realized by deploying the SDN architectural model, including faster implementations of new and enhanced services. Another benefit is that this approach enables for deep viewing into all network flows through an OpenFlow switch. This enables enhanced direct management and control over those flows, and therefore, enhanced network service and infrastructure management and control. Another benefit is that this model allow for many more options for dynamic provisioning, including real time provisioning and automated dynamic response to conditions based on flow analytics. This approach also allows enabling applications, edge processes and even individuals to directly control core resources. In addition, it provides for substantially improved options for creating customizable network services and infrastructure.

SDN architecture separates the control plane from the data plane, abstracting the forwarding path. This approach enables a controller, with a secure channel connecting network devices, to directly manipulate network functions. OpenFlow switches host repository of cached information (in flow tables), detailing information about the traffic streams in the switch. This information can be used to examine traffic flow and to manage flows based on such analysis. Consequently, SDN has been a major benefit to networks in cloud data centers and on WANs interconnecting such data centers, especially with regard to resource optimization.

Changes in architecture and technology for programmable networking, including SDN/OpenFlow, are expected to accelerate in the next few years even faster than they have during the current pace of rapid change. Therefore, the Chameleon Cloud network is being designed to ensure that it has the flexibility to accommodate ongoing future changes and to provide a maximum range of feasibilities for cloud innovation experiments based on programmable networking.

OpenFlow 1.3 has been deployed within the Chameleon environment, to enable experimenters with a full range of programmable resources. All switches within the Chameleon research network are fully OpenFlow 1.3 compliant programmable switches, although at this time (mid-2015), not all 1.3 functions have been implemented. Each node connects to the SDN network at 10 Gbps, and from each unit four 40 Gbps uplinks provide 160 Gbps per rack to the Chameleon core aggregations 100 Gbps Ethernet links, which in turn connect to DWDM-based WAN 100 Gbps networks, which interconnect facility networks (at a University of Chicago facility at Argonne National Laboratory and at the Texas Advanced Computing Center - TACC), campus networks, regional networks, and national R&E networks.

A separate one Gbps Ethernet network supports a management plane and control plane network, that extends to all nodes, to maintain monitoring and connectivity when the research network is isolated from the public networks or is in an experimental mode. The Chameleon initiative is currently also developing a plan to federate testbed resources, including experimental network testbeds, a topic discussed in a later section of this paper.

From the perspective of Chameleon environment management, there are three types of general networks to address: 1) the network inside each Chameleon Cloud rack, 2) the network interconnecting the Chameleon Cloud racks and rack sites, and 3) the networks that connect Chameleon with non-Chameleon Cloud resources and facilities.

For each type of network, data planes as well as control planes need to be implemented so that all the components are integrated in a complementary way to support the Chameleon Cloud experimenters. In addition, it is important to note that the overall Chameleon environment is comprised of three types of network resources that transverse each of the three networks described in the previous paragraph, the foundation network resources used by Chameleon, the network resources that support Chameleon cloud environment management (e.g., updating switch software images and reconfiguring of switches as part of supporting Chameleon cloud operations), and the network resources directly used by experimenters.

These networks support four categories of network services in the Chameleon Cloud: standard L2 networks, standard L3 networks, SDN networks, and hybrid networks (e.g., integrating SDN and non-SDN). For most experimenters, only three network services may be of interest, L2 and L3 networks, and SDN networks. The hybrid network is only of interest to experimenters who are engaged in investigations of hybrid mode experiments. The basic network that the majority of researchers use is an end to end network with single domain and mono layer networking. Network tunneling is a particularly important technology for cloud networking, especially for shared multi-tenants distributed cloud environment, and it is supported in the

Chameleon environment, including tunnel based on SDN. OpenStack is being used for the majority of network configurations.

VI. Chameleon Cloud Network Control Plane

The Chameleon Cloud control plane is the network required to program, control, and monitor network equipment and virtual network devices for Chameleon Cloud experiments. The control plane's access security allows experimenters and Chameleon managers and technical staff to connect internal or external network controllers and processors with minimum impact on required network performance. For the 1st phase of the project, the control plane within the racks primary uses OpenStack (Neutron)'s built in functions and features. The control plane among the racks and sites also uses OpenStack (Neutron) built in functions and features, with necessary enhancements provided by the Chameleon team. For federations with non-Chameleon resources, customized functions are being developed to providing the requirement federation components. The GENI – Chameleon federation is the initial federation requirement. The design and implementation plan for this federation are currently being developed.

The majority of Chameleon experiments will not use the control plane. However, for experimenters who would like use the control plane, utilizing OpenStack built in capabilities is an option, with a few pre-set configurations to allow experimenters to toggle between different types of network configurations. The experimenter accessible control plane supports pre-configured VXLAN, VLAN, and GRE solutions, with capabilities to allow experimenters to select from among different configurations.

In the case of experiments that need to use SDN/Openflow, the control plane provides capabilities to connect allocated switches, OVSs with SDN/OpenFlow network controllers for SDN control protocols. In such cases, the SDN/OpenFlow network controllers are implemented inside the Chameleon Cloud as well as over the external network for different experiments.

VII. Chameleon Cloud Network Data Plane

The Chameleon Cloud data plane is the network used to transit all regular network traffic, including regular in band network control traffic, management traffic, testing traffic, and monitoring traffic as well as data plane traffic defined by SDN/OpenFlow networking. Inside the rack, the data plane can be based on standard routed networking, GRE, VLANs, VXLANs, and OpenFlow with Hybrid networking. Between racks and sites, regular routed networking, VLANs, VXLANs, and OpenFlow co-existences are implemented. Federations with non-Chameleon Cloud resources use standard routed networking, VLANs and SDN.

VIII. Chameleon Cloud and Federation

The The Chameleon Cloud initiative is exploring opportunities to federate with other testbeds, including international testbeds. As noted, a current initiative is implementing a federation with the GENI testbed. Plans are underway also to federate with CloudLab and the Open Science Data Cloud (OSDC), which is managed by the Open Cloud Consortium (OCC) [8].

Internationally, multiple opportunities exist for cloud testbed federation. Preliminary discussions have taken place with managers of cloud testbeds in several countries about such federations. For example, the Chameleon internal testbed may be integrated with the International GENI (iGENI), a large scale, highly distributed infrastructure environment based on SDN/OpenFlow [9].

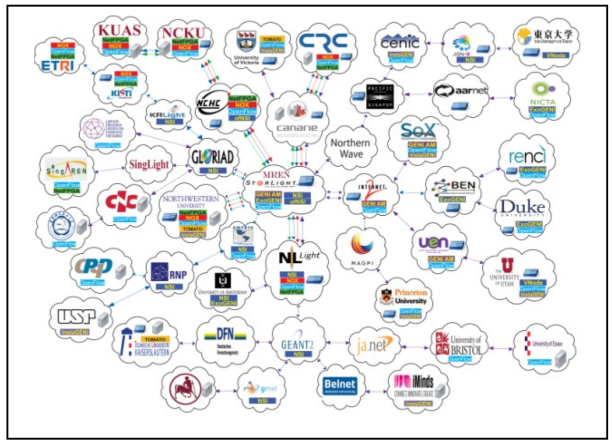

Figure 2: International Advanced Networking Research Distributed Environment Based On OpenFlow/SDN

The networks for these international federations will be implemented on the Global Lambda Integrated Facility (GLIF) [10]. The GLIF is global optical networking distributed facility that enables the creation of customized production and testbed networks. (Ref: Figure 3).

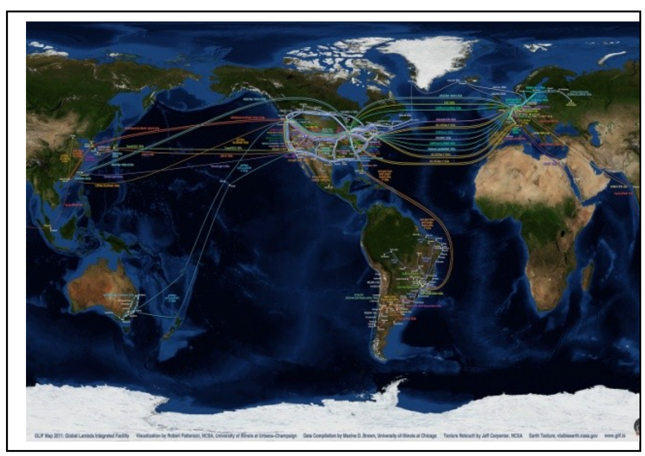

Figure 3: Global Lambda Integrated Facility (GLIF).

The GLIF community has been supporting an interdomain architecture development project that enables dynamic interdomain provisioning of L1 and L2 paths among the GLIF exchange points (GLIF Open Lambda Exchanges – GOLES). The GLIF community, with the Open Grid Forum, a standards organization, develop an API, the Network Services Interface Connection Service (NSI CS - published as NSI CS 2.0) [11] . The figure below (Figure 4) depicts a number of individual inter-domain VLANs around the globe that are directly addressable using the NSI CS API.

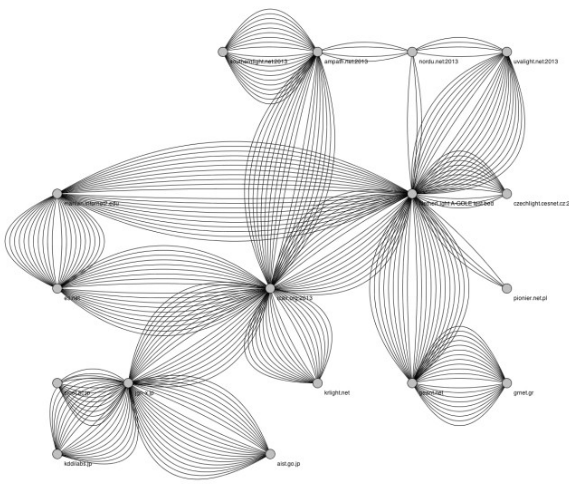

Figure 4: Global Dynamic VLANs Accesible As AutoGOLE Resources

Another resource that is available to enable these types of federations is a prototype Software Defined Networking Exchange (SDX) that has been implemented at the StarLight International/National Communications Exchange Facility in Chicago, Illinois (on the Chicago campus of Northwestern university. The StarLight Exchange Facility is a major exchange facility for world-wide international, national, and regional research and education networks, data intensive science networks, federal agency networks, and large scale national and international network research testbeds [12].

Essentially, the StarLight SDX is a large scale virtual switch within which resources can be customized to meet the requirements of specialized applications and service, including international testbed interconnections. The StarLight SDX is based on a foundation of interconnected high performance servers (controllers) and SDN/OpenFlow switches, which are depicted in the image below. (Ref: Figure 5)

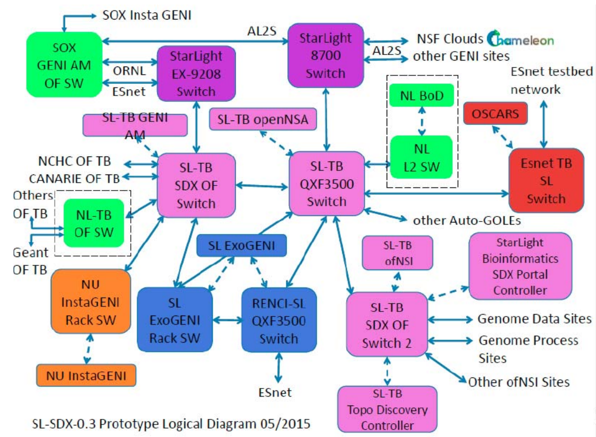

Figure 5 : StarLight SDX Prototype

X. Chameleon Cloud and Future Initiatives

There is a wide array of potential cloud computing research topics, almost an unlimited number. In the near term, as noted some of the new initiatives will focus on federations with other national and international testbeds. Another initiative is exploring the potential for HPC partitions within the Chameleon Cloud environment. Another initiative is exploring the potential to federate with distributed HPC environments such as the NSF's XSEDE (Extreme Science and Engineering Discovery Environment) [13]. Also, the StarLight consortium has been awarded a grant under the NSF 14-554 International Research Network Connections (IRNC) program, for Infrastructure and Innovation of U.S. R&E Open Exchange Points (IRNC: RXP), which is focused on developed international SDX capabilities for global science research. This StarLight international SDX facility will be designed and implemented to support multiple domain sciences including computer science focused on cloud research, including activities based on the Chameleon Cloud testbed.

XI. Summary

Next generation distributed infrastructure, especially computational clouds using highly programmable networks, including Software-Defined-Networking (SDN), comprise important investigated research topics. Recognizing the transformational changes enabling by advancing cloud services and technologies, and motivated by the need for continuous advancement, the US National Science Foundation (NSF) established a project, the NSFCloud initiative, as a resource for the computer science research community interested in developing and experimenting with innovative cloud architectures technologies by using large scale distributed cloud testbeds. This paper provides a general overview of the Chameleon Cloud testbed, and it also describes its high programmable high performance foundation networks, based on Software Defined Networking/OpenFlow (SDN). The Chameleon Cloud initiative plans to continually enhance the cloud testbed and to federate with other types of testbeds, including international testbeds. Partners are invited to participate in the Chameleon Cloud testbed. Below is an image of the Chameleon web site. (Ref: Figure 6)

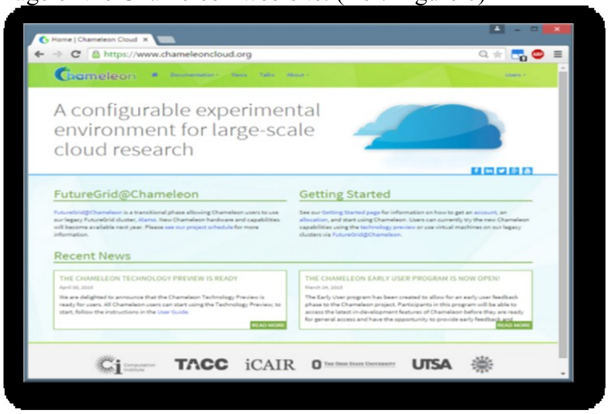

Figure 6: Chameleon Cloud Web Site

ACKNOWLEDGMENTS

For their excellent support, the authors would like to thank the US National Science Foundation, especially, the Computer and Information Science and Engineering (CISE) Directorate, and its NSFCloud and Global Environment for Network Innovations (GENI) programs, and the NSF Division of Advanced Cyberinfrastructure and its International Research Network Connections (IRNC) program, the Computational Institute at the University of Chicago, the Texas Advanced Computing Center at the University of Texas at Austin, The Ohio State University, the University of Texas at San Antonio, the Math and Computer Science Department at Argonne National Laboratory, the members and partners of the StarLight International/National Communications Exchange Facility Consortium, the Metropolitan Research and Education Network (MREN), the Global Lambda Integrated Facility (GLIF), other national and international research and networking partners, and other members of the iCAIR research partnership consortia [14].

REFERENCES

[1] www.geni.net

[2] www.chameleoncloud.org.

[3] www.cloudlab.us

[4] F. Travostino, J. Mambretti, G. Karmous-Edwards (editors), Grid Networks: Enabling Grids with Advanced Communication Technology, John Wiley & Sons, July 2006.

[5] This section is based on draft Chameleon architecture planning documents, unpublished

[6] This section is based on draft Chameleon workflow description documents, unpublished

[7] N. McKeown, et al., OpenFlow: Enabling Innovation in Campus Networks, ACM SIGCOMM Computer Communication Review, 2008, 2, 69-74.

[8] www.opencloudconsortium.org

[9] http://groups.geni.net/geni/wiki/IGENI

[10] Global Lambda Integrated Facility (GLIF) www.glif.is

[11] G. Roberts, T. Kudoh, I. Monga, J. Sobieski, J. MacAuley, C. Guok, NSI Connection Service V2.0, Open Grid Forum, GFD-R-P.212, NSI-WG June 2014.

[12] J. Mambretti, T. DeFanti, M. Brown, StarLight: Next-Generation Communication Services, Exchanges, and Global Facilities, Advances in Computers. 01/2010; 80:191-207.

[13] www.xsede.org

[14] www.icair.org