## Emulab

Mentioned in “Open Cirrus™ Cloud Computing Testbed: Federated Data Centers for Open Source Systems and Services Research”

Note: *PRS = physical resource set* in the quote below.

“EmuLab, the original PRS service, is a single-site testbed where each user can reserve a certain number of machines (typically a few tens) and get exclusive access to bare hardware. Emulab also provides mechanisms to emulate different network characteristics.”

“Emulab also provides mechanisms to emulate different network characteristics”

Emulab - Network Emulation Testbed Home

<http://www.emulab.net/>

Emulab software is installed at more than two dozen sites around the world - <http://users.emulab.net/trac/emulab/wiki/OtherEmulabs>.

Emulab Git Repository Information

<http://users.emulab.net/trac/emulab/wiki/GitRepository>

Emulab Client GUI

<http://www.emulab.net/netlab/client.php3>

“The Emulab software is a full-featured suite for testbed management. It is designed to provide the low-level access to testbeds that system researchers require, as well as higher-level tools that enhance researcher productivity.”

<http://www.newmexicoconsortium.org/probe>

Emulab software information

<https://users.emulab.net/trac/emulab/wiki/Tutorial>

Emulab hardware

<https://users.emulab.net/trac/emulab/wiki/UtahHardware>

Listing of publications either about Emulab or used Emulab in their research

<http://www.emulab.net/expubs.php>

Classes of research found on this page:

* Active Networks
* Adaptive Traffic Equalization
* Application and Experiment Support
* AQM Congestion Control
* Cloud Computing
* Databases
* Debugging Distributed Systems
* Declarative Networking
* DHT & P2P
* Distributed Checkpointing
* Distributed Databases
* Distributed Middleware
* Distributed Systems and Networking
* Dynamic Systems
* Education
* Emulation
* File Transfer & Streaming
* GRID Computing
* Internet Availability
* Measurements, Diagnosis, and Anomaly Detection
* Multicast and Broadcast
* Name Services
* Network Protocols
* Overlay Networks
* Parallel Computing
* Performance of Network Protocol
* Pub/Sub & Dissemination Systems
* QoS
* Resource Allocation
* Robust Scalable Services
* Routing
* Routing Overlay
* Security
* Simulation
* State Machines
* Testing
* Virtualization
* Vision & Architecture
* Wide-area Experiments
* Wireless
* Uncategorized

## PRObE – Parallel Reconfigurable Observational Environment

<http://www.newmexicoconsortium.org/probe>

NSF-sponsored project aimed at providing a large-scale, low-level systems research facility.

Collaborative effort:

* New Mexico Consortium
* Los Alamos National Laboratory
* Carnegie Mellon University
* University of Utah
* University of New Mexico

Will provide:

* Highly reconfigurable, remotely accessible and controllable environment that researchers can use to perform experiments that are not possible at smaller scale.
* Provides – at least two 1,024 node clusters, one of 200 nodes, and some smaller machines with extreme core count. Machines are retired large clusters donated by DOE facilities.

Based on:

* Emulab testbed-management software

It is dedicated to systems research. The organization envisions “this unique system will support research in many related fields such as Operating Systems, Storage, and High End computing.”

**World’s First Large Scale High Performance Super-Computing Systems Research Facility Opening at New Mexico Consortium**

Los Alamos, NM, October 15, 2012

<http://www.newmexicoconsortium.org/news/world2019s-first-large-scale-high-performance-super-adcomputing-systems-research-facility-opening-at-new-mexico-consortium>

“There have been no supercomputers available for computer scientists to work out issues with operating systems or other systems software. As supercomputers get larger and more complex, providing reliable and efficient system level software becomes a daunting task; without a place to test new concepts, advancement in high performance computing will be delayed.”

LANL’s deputy division leader for HPC, Gary Grider, had the idea to take nodes that would have been retired and re-use them for systems research. He worked with colleagues at Carnegie Mellon, and the NSF donated $10MM.

The University of Utah’s Emulab Group provides the testing management software for PRObE.

The idea came out of LANL, but the New Mexico Consortium is operating the center. “The New Mexico Consortium is a non-profit partnership between the University of New Mexico, New Mexico Institute of Mining and Technology, and New Mexico State University.”

The facility was completed in December 2011, with the help of local high school and college students.

PRObE Machines

<http://www.newmexicoconsortium.org/probe/machines>

## Open Cirrus™

https://opencirrus.org/

“Open Cirrus is an open cloud-computing research testbed designed to support research into the design, provisioning, and management of services at a global, multi-datacenter scale.

The open nature of the testbed is designed to encourage research into all aspects of service and datacenter management. In addition, we hope to foster a collaborative community around the testbed, providing ways to share tools, lessons and best practices, and ways to benchmark and compare alternative approaches to service management at datacenter scale.”

## Open Cirrus™ Cloud Computing Testbed: Federated Data Centers for Open Source Systems and Services Research

<http://static.usenix.org/events/hotcloud09/tech/full_papers/campbell.pdf>

“Open Cirrus is a cloud computing testbed for the research community that federates heterogeneous distributed data centers.”

It has “global services such as single sign-on, monitoring, storage, and job submission.”

“However, system researchers, who are developing the techniques and software infrastructure to support cloud computing, still find it difficult to obtain low-level access to large scale cluster resources.”

“The Open Cirrus™ project aims to address this problem by providing systems researchers with a testbed of distributed data centers they can use for systems-level (as well as applications and services) cloud computing research”

Aims to achieve the following goals:

* “Foster system-level research in cloud computing”
* “Encourage new cloud computing applications and applications-level research”
	+ “the potential for developing new application models and using these models to understand the necessary systems level support”
	+ “using the federated nature of Open Cirrus to provide a platform for new kinds of federated applications and service that run across multiple data centers.”
* Collection of experimental datasets.
* Develop open-source stacks and APIs for the cloud.

Three reasons why the sites are working together:

* Increased impact
* Validation through heterogeneity
* Shared innovation

Several high-level architectural choices drove the Open Cirrus design:

* Systems vs. application-only research – Open Cirrus enables research using physical machines in addition to virtualized.
* Federated vs. unified sites – “federates a number of sites with different hardware, services, and tools.” “Commonality is enabled by Open Cirrus global services under development, such as global single sign-on and global monitoring.”
* Data center focus vs. centralized homogenous infrastructure – “Open Cirrus revolves around multiple data centers.”

Open Cirrus Design:

* Global sign-on
* Direct access to physical resources
* Similar operating environments – create similar operating environments by defining a minimum set of services.
* Global services available from any site

Open Cirrus service stack implementation:

* PRS service – lowest level service – *physical resource set* – a set of VLAN-isolated compute, storage, and networking resources. The initial version uses HP Integrated Lights-Out technology (iLO) to remotely manage servers at the firmware level (although it is being generalized to handle other mechanisms such as IMPI).
* Cluster management services
	+ Cells as a Service (CaaS) – infrastructure management system for virtual resources hosted in the cloud focused on the creation and management of secure groupings of virtual resources.
	+ Tashi
* Application framework services – Hadoop, Pig, and MPI, …



The authors group testbeds into those supporting systems research and those supporting applications research:

“Testbeds such as PlanetLab [2], EmuLab [1], DETER Testbed [3], and Amazon EC2 [7], are designed to support systems research, but with diverse goals.”

“PlanetLab consists of a few hundred machines spread over the world, mainly designed to support wide-area networking and distributed systems research”

Open Cirrus™ architecture and configuration

<https://opencirrus.org/content/testbed-architecture-and-configuration>

## DETER testbed

Mentioned in “Open Cirrus™ Cloud Computing Testbed: Federated Data Centers for Open Source Systems and Services Research”

“The DETER testbed is an installation of the Emulab software. It is mainly used for security research, e.g., collecting a large-scale worm trace. Consisting of two heterogeneous sites, DETER may be viewed as a federated Emulab installation. However, the two sites are tightly-coupled, since the controller resides in one site and controls physical resources in both sites.”

Benzel, T., et al., “Design, Deployment, and Use of the DETER Testbed,” Proc. of DETER Workshop, Aug 2007.

## GENI

<http://www.geni.net/>

<http://groups.geni.net/geni/attachment/wiki/GeniSysOvrvw/GENI-AnIntroduction-28Feb2012.pdf>

“Evolving technological and social networks, intertwined and worldwide in scope, are rapidly transforming societies and economies. The Global Environment for Network Innovations (GENI), a project sponsored by the National Science Foundation, is open and broadly inclusive, providing collaborative and exploratory environments for academia, industry and the public to catalyze groundbreaking discoveries and innovation in these emerging global networks.”

“GENI is a virtual laboratory at the frontiers of network science and engineering for exploring future internets at scale. GENI creates major opportunities to understand, innovate and transform global networks and their interactions with society.”

GENI terms - <http://www.protogeni.net/trac/protogeni/wiki/FlackManual#Terms>

*Users* register at an *authority,* which issues them a *certificate* and gives them the *credentials* they need to present when allocating resources, which are advertised at various *managers*. To reserve resources (nodes, links, etc.), users must first create or be added to a *slice*. Slices are containers issued by authorities, which allow users to ask managers for resources. Resources are issued by managers to users in the form of *slivers*. Slivers contain addressable resources assigned to a user's slice from a manager. If a user asks for resources from more than one manager then they would have one slice and multiple slivers from different managers. Resources are described using *RSPEC*s, formally defined lists of resources. *Advertisement* RSPECs are listings of resources at a manager, *request* RSPECs are listings of resources a user wants and *manifest* RSPECs are listings of resources the user gets. Resources described in request RSPECs can be *bound* to a specific resource (ex. pc143) or *unbound* if the user simply desires any resource of a certain type (ex. any exclusive pc).

## ExoGENI

<http://www.exogeni.net/>

“ExoGENI is a new GENI testbed that links GENI to two advances in virtual infrastructure services outside of GENI: open cloud computing (OpenStack) and dynamic circuit fabrics. ExoGENI orchestrates a federation of independent cloud sites located across the US and circuit providers, like NLR and Internet2 through their native IaaS API interfaces, and links them to other GENI tools and resources.”

“Individual ExoGENI deployments consist of cloud site “racks” on host campuses, linked with national research networks through programmable exchange points. The ExoGENI sites and control software are enabled for flexible networking operations using traditional VLAN-based switching and OpenFlow. Using ORCA (Open Resource Control Architecture) control framework software, ExoGENI offers a powerful unified hosting platform for deeply networked, multi-domain, multi-site cloud applications. We intend that ExoGENI will seed a larger, evolving platform linking other third- party cloud sites, transport networks, and other infrastructure services, and that it will enable real-world deployment of innovative distributed services and new visions of a Future Internet.”

Projects that power ExoGENI:

* ORCA-BEN – core development of ORCA features. ExoGENI is controlled by a specific deployment of ORCA tailored to GENI needs and requirements. https://geni-orca.renci.org/trac/
* NeworkedClouds – adapting OpenStack to a networked clouds environment. https://code.renci.org/gf/project/networkedclouds/

ExoGENI papers can be found here:

<http://www.exogeni.net/document-library/>

## Orca

<https://geni-orca.renci.org/trac/>

“ORCA is a Control Framework to provision virtual networked systems via Secure and Distributed management of Heterogeneous Resources over Federated substrate sites and domains.”

“ORCA is the control framework for the ExoGENI testbed. ORCA is tightly integrated with OpenStack and Eucalyptus via special extensions to both for provisioning virtual machines. ORCA is also integrated with xCAT to support bare metal node provisioning. ORCA acts as a multi-layer dynamic network provisioning service for BEN optical network.”

## ProtoGENI

<http://www.protogeni.net/trac/protogeni>

<http://www.protogeni.net/trac/protogeni/wiki/Tutorial>

“ProtoGENI is an NSF-funded and GPO-funded prototype implementation and deployment of GENI, led by the Flux research group at the University of Utah, and largely based on our Emulab software. ProtoGENI is the Control Framework for GENI Cluster C, the largest set of integrated projects in GENI.”

Flack

Flack is a web-based graphical tool for reserving GENI resources.

<http://www.protogeni.net/flack>

## GENI in the Cloud

<http://s3.amazonaws.com/marcoy_thesis/Thesis.pdf>

## GENICloud and TransCloud: Towards a Standard Interface for Cloud Federates

<http://dspace.icsy.de:12000/dspace/bitstream/123456789/370/1/GeniCloud%20and%20TransCloud.pdf>

## GENI Racks

The GPO has GENI rack requirements. Current rack projects can be categorized into the following categories:

* ExoGENI A higher cost, flexible virtual networking topologies solution including OpenFlow, that also delivers a powerful platform for multi-site cloud applications. These racks are typically deployed as an integrated part of a campus network.
* InstaGENI - A mid-range cost, expandable GENI Racks solution that can be deployed at a large number of campuses, delivering Internet cloud applications support, along with OpenFlow and VLAN networking. These racks are normally deployed outside a site firewall.

## OpenFlow

<http://groups.geni.net/geni/wiki/OpenFlow>

<http://www.openflow.org/>

“OpenFlow is an open standard that allows you to run experimental protocols in production networks.”

“OpenFlow enables networks to evolve, by giving a remote controller the power to modify the behavior of network devices, through a well-defined "forwarding instruction set". The growing OpenFlow ecosystem now includes routers, switches, virtual switches, and access points from a range of vendors.”

<http://groups.geni.net/geni/wiki/GENIExperimenter/RSpecs>

“In order to allow interoperability among different Aggregate Managers(AMs), GENI requires a common language for describing resources, resource requests, and reservations. GENI now uses standardized Request Specification (RSpec) documents which are XML documents following agreed schemas to represent resources. The schemas support Aggregate or resource specific extensions. Ongoing work covers agreeing upon ontologies for other resource types.”

Related - Software Defined Networking (SDN).

<http://opennetsummit.org/talks/ONS2012/heller-mon-intro.pdf>

<http://en.wikipedia.org/wiki/Software-defined_networking>

<http://www.slideshare.net/martin_casado/sdn-abstractions>

## FIRST – Future Internet Research for Sustainable Testbed

<http://www.apan.net/meetings/kualalumpur2009/proposals/FutureInternet/2009.07-FIRST-APAN-v02.pdf>

<http://www.computer.org/csdl/proceedings/icoin/2012/0251/00/06164436-abs.html>

FiRST (Future Internet Research for Sustainable Test-bed) is the future internet platform development project being performed in Korea

## NEPI – Network Experimentation Programming Interface

NEPI: using independent simulators, emulators, and testbeds for easy experimentation

<http://dl.acm.org/citation.cfm?id=1713268>

<http://hal.archives-ouvertes.fr/docs/00/39/76/92/PDF/RR-6967.pdf>

Nepi: An integration framework for network experimentation

http://nepihome.net/raw-attachment/wiki/WikiStart/nepi\_integration\_framework.pdf

NEPI tutorial

http://www.nepihome.org/raw-attachment/wiki/nepi/CCNxOnPlanetLabEurope/ccnx\_planetlab\_tutorial.pdf

"NEPI, the Network Experimentation Programming Interface, is a life-cycle management tool for network experiments."

## Toward Replayable Research in Networking and Systems

<http://www.cs.utah.edu/flux/papers/replayable-archive10.pdf>

“Beyond encapsulating the deﬁnition and history of an experiment, a replayable experiment is associated with a mechanism for actually re-executing a system under test.”

Data repositories, testbeds, and experiment management systems

**Papers listed for data repositories:**

* Challenges and opportunities in Internet data mining
* Cooperative Association for Internet Data Analysis (CAIDA). DatCat. http://www.datcat.org/.
* The Internet Measurement Data Catalog.

**Papers listed for network testbeds:**

* L. Peterson et al. A blueprint for introducing disruptive technology into the Internet. In HotNets, Oct. 2002.
* Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In WCNC, pages 1664–1669, Mar. 2005.
* B. White et al. An integrated experimental environment for distributed systems and networks. In OSDI, pages 255–270, Dec. 2002.

**Papers listed for experiment management systems:**

* J. Albrecht, C. Tuttle, A. C. Snoeren, and A. Vahdat. PlanetLab application management using Plush. OSR, 40(1):33–40, Jan. 2006.
* E. Eide, L. Stoller, and J. Lepreau. An experimentation workbench for replayable networking research. In NSDI, pages 215–228, Apr. 2007.
* Y. Wang et al. Automating experimentation on distributed testbeds. In ASE, pages 164–173, Nov. 2005.

“testbeds have focused on supporting control and therefore repeatability (e.g., Emulab) or on deployment of applications on production networks (e.g., PlanetLab)”

These three topics are moving the systems community forward in terms of supporting repeatable research.

“A replayable experiment not only encapsulates the definition of a computer-based activity, but in addition, is associated with a facility for actually re-executing that activity.”

“At a minimum, the repeatability of an analysis requires the collection, dissemination, and preservation of the “raw” input data, and also a thorough definition of the analysis itself.”

“A packaged experiment is **replayable** if it is accompanied by or associated with a mechanism for re-executing the activities described by the experiment.”

“To ensure that already-packaged experiments continue to be replayable, a possible long-term solutions for a testbed to offer virtual equivalents to all of its hardware.”

The testbed should support recording mechanisms to capture measurable attributes of the experiment.

## An Experimentation Workbench for Replayable Networking Research

<http://www.cs.utah.edu/flux/papers/workbench-nsdi07-base.html>

## PlanetLab

<http://static.usenix.org/events/nsdi04/tech/full_papers/bavier/bavier_html/>

<http://nsg.cs.princeton.edu/publication/planetlab_hotnets_02.pdf>

Arch. Overview - <http://planet-lab.org/files/pdn/PDN-06-031/pdn-06-031.pdf>

Guides - <https://www.planet-lab.org/doc/guides>

Four design principles that are not widely supported in existing testbeds

1. Services should be able to run continuously and access a slice of the overlay’s resources
2. Control over resources should be distributed
3. Overlay management services should be unbundled and run in their own slices
4. APIs should be designed to promote application development.

These emerging services are the result of a convergence of two historically separate research communities. :

1. The distributed systems community
2. The network community

The authors want to integrate these two perspectives into a new *service-oriented network architecture*. However, because of the ossification of the Internet, the authors introduce a blueprint for introducing disruptive technologies into the Internet through the use of overlay networks.

Mentioned in “Open Cirrus™ Cloud Computing Testbed: Federated Data Centers for Open Source Systems and Services Research”

“PlanetLab consists of a few hundred machines spread over the world, mainly designed to support wide-area networking and distributed systems research. Although it does not provide access to bare metal hardware, it does provide root access to the OS through a light-weight virtualization similar to FreeBSD jail.”

“EmuLab, the original PRS service, is a single-site testbed where each

user can reserve a certain number of machines (typically a few tens) and get exclusive access to bare hardware. Emulab also provides mechanisms to emulate different network characteristics.”

There are several testbed items listed in the planetlab\_hotnets\_02.pdf paper. These are generally referring to networking testbeds:

* Internet2
* Emulab
* The Grid – Globus
* The ABONE – an overlay testbed that grew out of the Active Networks initiative.
* The XBONE – an overlay network with support for IP-in-IP tunneling.

In the planetlab\_hotnets\_02.pdf paper, there is a section titled “Application-Centric Interfaces” which discusses the following:

* Perhaps the single greatest failure of testbeds, in general, is they do not promote application development.
* Difficult to do research on a testbed and provide a stable foundation within the testbed that is slow to make changes.

NOTE: Initially PlanetLab and the ExoGENI projects both prescribe to the notion using the same hardware within their projects.

<http://nsg.cs.princeton.edu/publication/planetlab_hotnets_02.pdf> - Section 4.

<http://groups.geni.net/geni/wiki/GENIRacksHome/RacksChecklist>

## Plush

PlanetLab Application Management Using Plush

<http://cseweb.ucsd.edu/~snoeren/papers/plush-osr.pdf>

##  Macintosh HD:Users:Jon:Desktop:plush_fig1.png

## A Network in a Laptop: Rapid Prototyping for Software-Deﬁned Networks

<http://klamath.stanford.edu/~nickm/papers/a19-lantz.pdf>

This paper introduced Mininet. A tools that can be used within a VM to do rapid prototyping. Fits in the area of software-defined networks.

## Measurement Lab

<http://www.measurementlab.net/>

<http://www.sigcomm.org/ccr/papers/2010/July/1823844.1823853>

<http://www.cc.gatech.edu/people/home/dovrolis/Papers/mlab-ccr-final.pdf>

Measurement Lab is an open platform for researchers to deploy Internet measurement tools. By enhancing Internet transparency, M-Lab helps sustain a healthy, innovative Internet.

## ViNe

<https://www.acis.ufl.edu/content/user-level-virtual-network-support-sky-computing-0>

<https://www.acis.ufl.edu/content/managed-virtual-networks-grids%E2%80%94-vine-approach>

<http://www.cecs.uci.edu/~papers/ipdps06/pdfs/1568975101-IPDPS-paper-1.pdf>

Tsugawa, M. and J. A. B. Fortes, "A Virtual Network (ViNe) Architecture for Grid Computing," 20th Intl Parallel and Distributed Processing Symposium (IPDPS-2006), 04/06.

“The primarily goal is to create virtual network environments with bi-directional communication capability between any pair of hosts. In order to support unmodified applications, the ViNe architecture is based on IP-overlay on top of the Internet. Components to be virtualized are network interfaces, routers and links between them.”

## Automating experimentation on distributed testbeds (2005)

<http://dl.acm.org/citation.cfm?id=1101934>

## Towards an Experimental Testbed Facility for Cyber-Physical Security Research (2012)

<http://www.osti.gov/energycitations/product.biblio.jsp?osti_id=1038374>

Computer security domain

-Search Google Scholar for “Toward Replayable Research in Networking and Systems”

## Exploring the Virtual Infrastructures as a Service concept with HIPerNET

http://hal.inria.fr/docs/00/49/37/85/PDF/RR-7185.pdf

## Design and Evaluation of a Virtual Experimental Environment for Distributed Systems

<http://hal.archives-ouvertes.fr/docs/00/72/43/08/PDF/RR-8046.pdf>

## The Software-Defined Data Center: Dissecting the Latest Buzzword

<http://www.networkcomputing.com/data-center/the-software-defined-data-center-dissect/240006848>

## VMworld 2012: Software-Defined Networking Becomes a Buzzword

<http://www.eweek.com/c/a/Virtualization/VMWorld-2012-SoftwareDefined-Networking-Becomes-a-Buzzword-346692/>

## The Xen-Blanket: Virtualize Once, Run Everywhere (2012)

<http://jamjoom.net/publications/jamjoom-EuroSys-12.pdf>

Open Virtualization Format – one approach to standardize virtualization

The authors take a *user-centric* approach. Users can implement or deploy hypervisor tools or management services on clouds.

## Supporting Experimental Computer Science

<http://hal.archives-ouvertes.fr/docs/00/72/26/05/PDF/RR-8035.pdf>

401.pdf

## The Open Cloud Testbed: A Wide Area Testbed for Cloud Computing Utilizing High Performance Network Services

<http://arxiv.org/pdf/0907.4810.pdf>

The Open Cloud Testbed is managed and operated by the Open Cloud Consortium.

[www.opencloudconsortium.org](http://www.opencloudconsortium.org)

OCC Cloud Resources - <http://opencloudconsortium.org/2012/01/10/occ-cloud-resources/>

* OCC-Y Cluster consists of 4 racks, 928 cores and 1 PB of storage and is a data cloud based upon the Hadoop.
* The OCC Open Cloud Testbed has approximately 1 rack of equipment at three different locations (Chicago, Livermore and Miami) that can be used for experimental studies and testing of cloud computing software and services.
* OSDC listed below.

Testbeds available for a variety of research:

* High performance computing applications (e.g. the Teragrid)
* Network research (e.g. PlanetLab)
* Network security research (e.g. DETER)

The Open Cloud Consortium has:

* Open Cloud Testbed -
* Open Science Data Cloud (OSDC)

<http://opencloudconsortium.org/2012/01/10/occ-cloud-resources/>

## Open Science Data Cloud

<http://www.opensciencedatacloud.org>

Managed by the Open Cloud Consortium

The OSDC is a petabyte scale science cloud for researchers to manage, analyze and share their data and to get easy access to data from other scientists.

Sponsor (and donation) information:

* The OCC-Y Hadoop Cluster (approximately 1000 cores and 1 PB of storage) was donated by Yahoo! in 2011.
* Cisco provides the OSDC access to the Cisco C-Wave, which connects OSDC data centers with 10 Gbps wide area networks.
* The OSDC is supported in part by the National Science Foundation under Award Number 1129076. Any opinions, findings, and conclusions or recommendations expressed in OSDC materials are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. This five year (2010-2016) PIRE Award is used to train scientists to use the OSDC and to further develop the underlying technology.
* The StarLight Facility in Chicago allows OSDC to connect to over 30 high performance research networks around the world at 10 Gbps or higher, with an increasing number of 100 Gbps connections.
* Yahoo! donated 5 racks of equipment to the OSDC in 2008. These racks provided over 800 cores and 0.6 PB of data for the Proof of Concept OSDC (2008-2010).

<http://opencloudconsortium.org/2012/01/10/occ-cloud-resources/>

OSDC currently consists of:

* Six racks containing 159 nodes, 1248 cores and 1.2 PB of usable storage.
* Divided into a utility cloud called OSDC-Adler that is based upon Eucalyptus and a utility cloud called OSDC-Sullivan that is based upon OpenStack

## VISION Cloud Project

<http://www.visioncloud.eu/>

VISION Cloud is a FP-7 funded EU research project. The focus is on innovative cloud storage.

## Cloud Testing Tools (2011)

<http://www.computer.org/csdl/proceedings/sose/2011/0411/00/06139087-abs.html>

Testing-as-a-Service (TaaS) is mentioned in this paper.

The authors present an high-level architecture of what should be included to create a TaaS solution.

According to [40], TaaS concept was initially introduced by Tieto in Denmark in 2009. TaaS has received wide attention due to its scalable testing environments, cost reduction, utility based service models, and on-demand testing services.

As shown in Fig. 4, Yu, et al., [12] deﬁned a 5-layer TaaS framework based on cloud infrastructure services, including:

* Test service tenant and contributor layer: This layer provides functionality that supporting testing service tenant and contributor to interact with TaaS.
* Test task management layer: This layer is a middleware layer, supporting service registry and repository, scheduling and dispatching test tasks, and some other functionality.
* Testing resource management layer: This layer acts as the Cloud infrastructure, taking the responsibility of resource management and monitoring, test task provisioning.
* Test layer: This layer is the kernel part of the platform, consisting of service composition, service pooling and test-reduce sub layer.
* Testing database layer: This layer is used to store the test task of tenants, targets-under-test, service images, and bug tracking results.



## Expertus: A Generator Approach to Automate Performance Testing in IaaS Clouds (2012)

<http://www.istc-cc.cmu.edu/publications/papers/2012/expertus.pdf>

Expertus is designed to automate performance and scalability testings in cloud platforms. Expertus was created by extending our previous work on code generation for Infopipes [21]. Expertus takes an experiment specification (application parameters, target cloud, and test scenarios) as an input and creates the resource (e.g., scripts) to fully automate the testing process (i.e., deployment, configuration, execution and data collection). Expertus is designed as a multi-pass compiler, which leverages template-driven code generation to modularly incorporate different software applications and clouds. As documented by our results, the flexibility and extensibility of Expertus have enabled cloud experiments at a scale that is beyond manual test execution.

We tackle experiment challenges by automating complete experiment process i.e., application deployment and configuration, experiment execution, data collection, and data analysis. The cornerstone of experiment automation is the code generation,

## How a Structured Testbed Enables the Rapid Development and Deployment of Cloud Services: the VISION Cloud Use Case (2012)

<http://www.computer.org/csdl/proceedings/iscc/2012/2712/00/MC22-abs.html>

## MalStone and MalGen

<http://code.google.com/p/malgen/>

<http://malgen.googlecode.com/files/malstone-TR-09-01.pdf>

Generate synthetic site-entity log data for testing and benchmarking applications requiring large data sets.

## Developing highly complex distributed systems: a software engineering perspective

<http://www.springerlink.com/content/6382652856g55431/fulltext.pdf>

## A Survey of Software Testing in the Cloud (2012)

<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6258440>

## Software Testing in Cloud Platform: A Survey (2012)

<http://www.ijcaonline.org/archives/volume46/number6/6912-9304>

## Improving Cloud Datacentre Scalability, Agility and Performance Using OpenFlow (2012)

<http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6338039>

## Large Scale Data Analytics on Clouds (2012)

<http://dl.acm.org/citation.cfm?id=2390026>

## Automated Software Testing as a Service (2010)

<http://dslab.epfl.ch/pubs/taas.pdf>

## An environment for evaluation and testing of service workflow schedulers in clouds

<http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5999838>

Tashi project - Apache project