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

Roy Campbell,[5] Indranil Gupta,[5] Michael Heath,[5] Steve Ko,[5] Michael Kozuch,[3] Marcel Kunze,[4] Thomas Kwan,[6] Kevin Lai,[1] Hing Yan Lee,[2] Martha Lyons,[1] Dejan Milojicic,[1] David O'Hallaron,[3] and Yeng Chai Soh[2]

[1]*HP Labs,* [2]*IDA,* [3]*Intel Research,* [4]*KIT,* [5]*UIUC, and* [6]*Yahoo!*

Abstract

There are a number of important and useful testbeds, such as PlanetLab, EmuLab, IBM/Google cluster, and Amazon EC2/S3, that enable researchers to study different aspects of distributed computing. However, no single testbed supports research spanning systems, applications, services, open-source development, and datacenters. Towards this end, we have developed Open Cirrus, a cloud computing testbed for the research community that federates heterogeneous distributed data centers. Open Cirrus offers a cloud stack consisting of physical and virtual machines, and global services, such as sign-on, monitoring, storage, and job submission. By developing the testbed and making it available to the research community, we hope to help spur innovation in cloud computing and catalyze the development of an open source stack for the cloud.

1. Introduction

There is growing interest in cloud computing within the systems and applications research communities. However, systems researchers often find it difficult to do credible work without access to large-scale distributed datacenters. Application researchers could also benefit from being able to control the deployment and consumption of hosted services across a distributed cloud computing testbed.

Pay-as-you-go utility computing services by companies such as Amazon, and new initiatives by Google, IBM, and NSF, have begun to provide applications researchers in areas such as machine learning and scientific computing with access to large scale cluster resources. 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. (Open Cirrus™ is a trademark of Yahoo!, Inc.). The project is a joint initiative sponsored by HP, Intel, and Yahoo!, in collaboration with NSF, the University of Illinois (UIUC), Karlsruhe Institute of Technology, and the Infocomm Development Authority (IDA) of Singapore. Additional Open Cirrus site members are expected to join in 2009.

The Open Cirrus testbed is a collection of federated datacenters for open-source systems and services research. As shown in Figure 1, the initial testbed is composed of six sites in North America, Europe, and Asia. Each site consists of a cluster with at least 1000 cores and associated storage. Authorized users can access any Open Cirrus site using the same login credential.

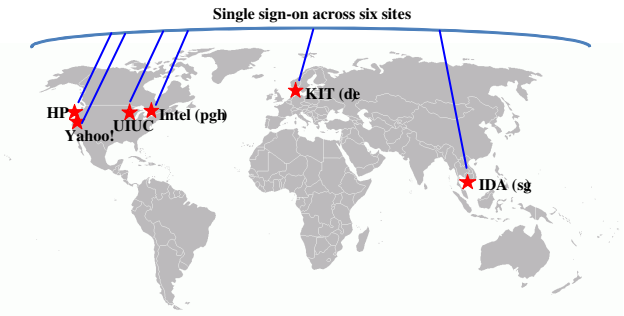

Figure 1. Open Cirrus testbed circa Q1 2009.

2. Motivation and context

Open Cirrus aims to achieve the following goals:

Foster systems-level research in cloud computing. In the current environment, only big service providers such as Yahoo!, Google, and Amazon have access to large scale distributed datacenters to develop and test new systems and services. Researchers must typically rely on simulations or small clusters. In creating Open Cirrus, we hope to help democratize innovation in this area by providing researchers with the resources they need to do credible systems research. Open Cirrus provides two unique features that we believe are essential to enabling systems-level research. First, Open Cirrus sites allow access to low-level hardware and software resources (e.g., install OS, access hardware features, and run daemons). Second, the testbed comprises heterogeneous sites in different administrative domains around the world, so researchers can study issues in leveraging multiple datacenters.

Encourage new cloud computing applications and applications-level research. Providing a platform for real world applications and services is an important part of Open Cirrus. Particularly exciting are (1) the potential for developing new application models and using these

models to understand the necessary systems level support, and (2) using the federated nature of Open Cirrus to provide a platform for new kinds of federated applications and services that run across multiple data centers.

Collection of experimental datasets. Researchers in cloud computing often lack datasets that would enable them to conduct high-quality experimental evaluations. Open Cirrus sites will enable researchers to import, store, and share large-scale datasets such as web crawls and datacenter workload traces. With such facilities, we hope that Open Cirrus will become a "watering hole" where researchers with similar interests may exchange datasets and develop standard cloud computing benchmarks.

Develop open-source stacks and APIs for the cloud. If cloud computing is to become widespread, it will be important to have a non-proprietary and vendor-neutral software stack. We envision Open Cirrus as a platform that the open source community can use to design, implement, and evaluate such codes and interfaces for all levels of the cloud stack. Open source is as much about community as it is about software, and we see Open Cirrus as a foundation of a larger open cloud community.

There are three reasons the participating Open Cirrus sites are working together to provide a single federated testbed, as opposed to each site building and operating a separate cluster:

- *Increased impact.* Collaborating on a single larger effort provides us with greater impact than we could achieve individually.
- *Validation through heterogeneity.* The quality of software and services can be improved by testing in the different site environments.
- *Shared innovation.* We expect that pooling resources and collaborating on a larger testbed will improve efficiency because the sites will be sharing innovations.

One measure of efficiency is management cost. Figure 2 shows the basic idea using ballpark cost figures gleaned from the current Open Cirrus sites. While the costs for running a cloud infrastructure increase with the number of sites, the savings from sharing software development and operational methods reduces the overall costs.

For example, Yahoo! has invested multiple engineer-years of effort in Hadoop and HDFS. Intel Research is a major contributor to the Apache Software Foundation's Tashi project, an open source infrastructure for managing and scheduling virtual machines. HP is developing a physical resource set allocator. UIUC is developing new monitoring and storage management infrastructures. KIT is creating new interactive services for HPC-on-demand. IDA conducts research in virtual networks, programming models, and robust resource allocation and management. By sharing these new systems and the lessons learned in deploying them, all of the sites benefit.

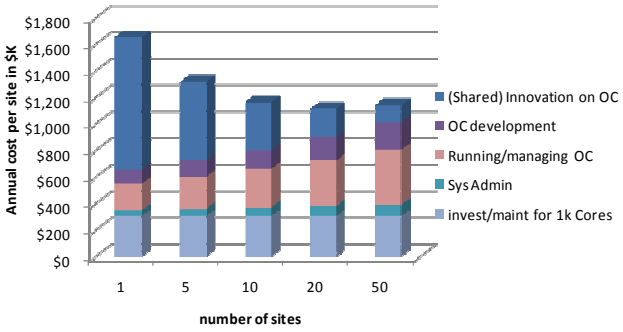

Figure 2. Annual cost per site for different number of sites.

3. Architecture, design, and implementation

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

Systems vs. application-only research. In contrast to clusters, such as IBM/Google and Amazon EC2/S3, Open Cirrus enables research using physical machines in addition to virtualized. This requires provisioning of the bare metal, enabling root access to provisioned servers, providing isolation at the network level, and reclaiming access in case of fraudulent or erroneous behavior.

Federated vs. unified sites. In contrast to a unified architecture such as PlanetLab, Open Cirrus federates a number of sites with different hardware, services, and tools. The sites exist on different continents, under different regulations and privacy concerns. Commonality is enabled by Open Cirrus global services under development, such as global sign-on and global monitoring. Some local services may be different across sites, but common practices and regulations will be established to promote consistent administration and oversight.

Data center focus vs. centralized homogeneous infrastructure. Compared to a centralized approach, such as EmuLab, Open Cirrus revolves around multiple data centers. This data center focus enables independent research, while sharing resources. It has implications on security, enforcing authorizations between users and individual sites, and integration with existing organizational regulations.

Open Cirrus design. The Open Cirrus design is guided by a desire to create a unified and coherent resource, rather than several completely separate clusters that only share a name. The major design goals include:

Global sign-on. Each Open Cirrus user has a single login name and password that will work at any site that they are authorized to use, which is necessary for a coherent and unified testbed. To provide this facility, Open Cirrus supports a centralized database that maintains a global

username and access key for each user. Because each site is expected to provide user access through an *ssh* gateway machine, ssh public keys are a natural fit for the user access keys. Getting an account on one Open Cirrus site does not automatically grant you accounts on all sites; each site makes access decisions independently. However, when users have been granted access by more than one site, the same login credentials will work on all access-granting sites. Open Cirrus also maintains a database of revoked access keys and a notification service that will distribute information about undesirable or suspicious user behavior to all Open Cirrus site administrators.

*Direct access to physical resources. S*ystems research is supported by allowing direct access to physical resources on the machine. For example, researchers can have root password, install kernel images, and access processors, chipsets, and storage. However, some resources, particularly network resources needed for proper isolation such as switch VLAN configurations, may be virtualized or unavailable.

Similar operating environments. Given that the Open Cirrus sites are managed by different organizations with different practices, it is not feasible for each site to have identical operating environments. However, we can create *similar* operating environments by defining a minimum set of services that every site must offer. For example, at a minimum, each Open Cirrus must offer Hadoop and an HDFS repository, and must support global sign-on.

Global services available from any site. A small set of global services are available from any Open Cirrus site. Examples include a common subversion repository, global monitoring, and a moderate scale storage service for configuration files, intermediate results, or binaries.

Open Cirrus service stack implementation. A typical Open Cirrus site consists of a number of services:

PRS service. The lowest level service is based on the notion of a *physical resource set (PRS*). A PRS is a set of VLAN-isolated compute, storage, and networking resources. At any point in time, a cluster (datacenter) is partitioned into one or more PRS *domains,* dynamically allocated and managed by a *PRS service,* at the request of *PRS clients*. Each PRS domain is VLAN-isolated from the others, and all applications and services on the cluster run on some PRS domain. For example, Figure 3 shows a snapshot of the PRS domains in a typical cluster. In this example, the cluster is partitioned into four domains. From left to right, the first domain is used for low-level systems research, where researchers have installed their own OS kernels and are running their own experimental codes and services. The second domain runs a VM management system that provides users with virtual clusters of VMs that share the physical nodes and storage in the PRS domain. Users build their own services and applications on top of these virtual clusters. The third and fourth domains are storage and workload and trace collection infrastructure services that are accessed by user services and applications running on the second partition.

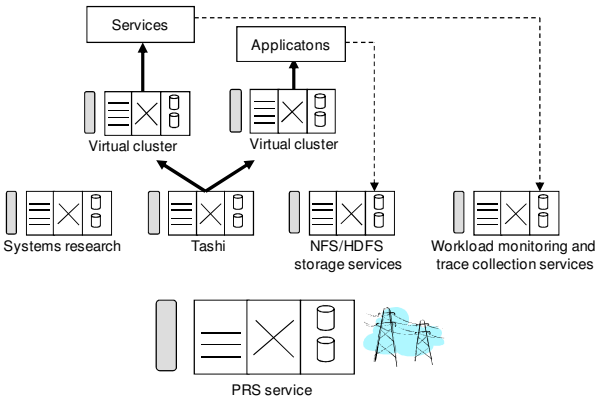

Figure 3. PRS domains.

HP is leading the development of the PRS service as a monetary system based on physical machine allocation. The initial version uses HP Integrated Lights-Out technology (iLO) to remotely manage servers at the firmware level (although this is being generalized to handle other mechanisms such as IPMI). This allows us to image the operating system, reboot, shutdown, etc., regardless of the server's operating system. In addition, we use VLAN technology to isolate different users and provide custom firewalls for each user.

Cluster management services. We currently run several different cluster management services on Open Cirrus sites. The first service, *Cells as a Service* (CaaS), is an infrastructure management system for virtual resources hosted in the cloud focused on the creation and management of secure groupings of virtual resources, called Service Cells. Within cells customers can instantiate and operate the services of their choice. The second service, *Tashi*, is an open-source cluster management system for cloud computing on massive internet scale datasets (Big Data). The system is being developed through the Apache Software Foundation incubator by Intel, Yahoo, and Carnegie Mellon University. Similar to systems such as CaaS, Eucalyptus, and EC2, Tashi manages logical clusters of virtual machines. The key research focus is the high-level co-scheduling of computation (in the form of VMs), storage (distributed across the local disk drives of the cluster), and power. Other systems, such as Eucalyptus, are likely to be supported as well.

Application framework services. Open Cirrus sites also provide higher level services, such as Hadoop, Pig, and MPI, that support user-level applications and services.

Figure 4 shows the high-level view of a typical Open Cirrus site (the Intel Research Pittsburgh cluster) and Table 1 summarizes some of the basic characteristics of the initial six Open Cirrus sites.

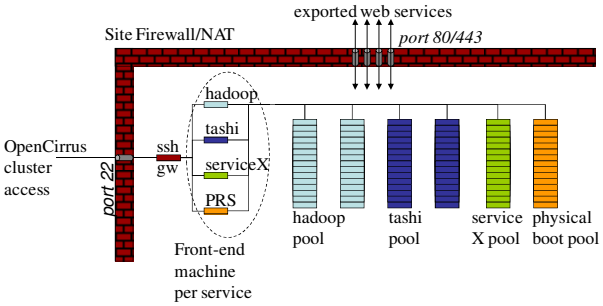

Figure 4. A typical Open Cirrus site.

Site	Characteristics							
	#Cores	#Servers	Public partition	Memory Size	Storage Size	Spindles	Network	Focus
HP	1,024	256	178	3.3TB	632TB	1152	10G internal 1Gb/s x-rack	Hadoop, Cells, PRS, scheduling
IDA	2,400	300	100	4.8TB	43TB+ 16TB SAN	600	1Gb/s	Apps based on Hadoop, Pig
Intel	1060	155	145	1.16TB	353TB local 60TB attach	550	1Gb/s	Tashi, PRS, MPI, Hadoop
KIT	2048	256	128	10TB	1PB	192	1Gb/s	Apps with high throughput
UIUC	1024	128	64	2TB	~500TB	288	1Gb/s	Datasets, cloud infrastructure
Yahoo	3200	480	400	2.4TB	1.2PB	1600	1Gb/s	Hadoop on demand

Table 1. Summary of the initial Open Cirrus sites.

4. Open Cirrus Economic Model

The emergence of each individual site in Open Cirrus and the expected growth of the federation are driven by the economy in today's cloud computing environment. This section derives explicit breakeven points for the choice between renting vs. owning a cloud infrastructure, thus implicitly justifying Open Cirrus' economic rationale.

Single Site: Consider a medium-sized organization (e.g., a startup or a university department) wishing to provide a web service to a client population. The service will run in a cloud, accessing stored data and consuming CPU cycles. Suppose this service is identical to the UIUC Open Cirrus site: 128 servers (1024 cores) and 524 TB. The organization's dilemma is: should it rent the infrastructure from a cloud provider (e.g., Amazon Web Services' [7] EC2 and S3), or should it own (buy and maintain) a cloud?

First, the option of renting: at current AWS rates of $0.12 per GB-month and $0.10 per CPU-hour, our service incurs monthly: (1) storage cost of 524*1,000*$0.12, or $62,880; (2) total cost of $62,880 + 1,024*24*30*$0.10, $136,608. Second, for the option of owning, the split of amortized monthly costs is 45%:40%:15% for hardware:power:network [8,9,10,11]. If the service's lifetime is M months, it would incur monthly: (1) storage cost (assuming $300 1 TB disks and scaling for power and networking) of 524*$300/0.45/M, or $349,333/M; (2) total cost (based on actual systems cost and salary of one sysadmin for about 100 servers [9,10]) of ($700K/0.45/M + $7,500), or ($1,555,555/M + $7,500).

This allows us to calculate the breakeven points for (1) storage as 349K / M < 62,880, or M > 5.55 months; (2) overall as 1,555K / M + 7,500 < 136,608, or M > 12 months. *Thus, if the service runs for over 12 months, it is preferable to own infrastructure than to rent it. Similarly, it is better to own storage if it is used for over 6 months.*

Clouds are typically under-utilized [8]. With x% resource utilization, the above breakeven time becomes 12*100/x months. Since 36 months is the typical lifetime of hardware, the breakeven resource utilization is 12*100/x < 36, or x > 33.3%. Concretely, even at currently CPU utilization rates of 20% observed in industry, a storage utilization of > 47% would make it preferable to own (since storage and CPU account evenly for costs).

Federated Sites: Federation can help absorb overloads due to spikes (e.g., at conference deadlines) or under-provisioning [8,11]. Figure 5 plots the costs incurred by a single under-provisioned cloud for three options: offloading only to AWS (Existing DC), offloading to 5 federated clouds (Open Cirrus 6) and AWS, offloading to 49 federated clouds (Open Cirrus 50) and AWS.

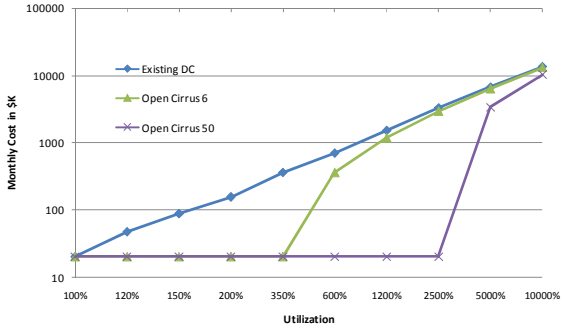

Figure 5. Overload Under-provisioned Site to AWS v. 6/50 Sites

It is clear that a federation of 6 sites is able to defer costs up to 250% overload, while with 50 sites, the breakeven point is ~2,500% (assumption is that other sites are utilized 50% and are not idle, otherwise, the breakeven would have been 500% and 5,000% respectively). The detailed data and spreadsheet for this calculation are available from http://opencirrus.org.

Finally, we state the caveat that the above calculation is only a starting step, e.g., it can be expanded by accounting for economic costs of disasters such as massive failure, project cancellation, time to start up, etc.

5. Related Work

Existing testbeds can be broadly grouped into those that mainly support applications research and those that can support systems research. Testbeds, such as the Google-IBM cluster [5] and TerraGrid [4], focus on supporting

computing applications research. Thus, these testbeds do not enable access to bare metal hardware or root access to the OS. Instead, services such as MPI and Hadoop are installed for ease of access to the resources. For example, the Google/IBM cluster is configured with the Hadoop service and targets data-intensive applications research, such as large-scale data analytics. TerraGrid is a multi-site infrastructure mainly used for scientific research. The Open Cloud Testbed [6] focuses on cloud computing middleware research, and it is currently configured as a small-scale testbed with four 32-node sites (at the time of this writing).

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. 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. Open Cirrus provides Emulab-like flexibility for systems research with federation and heterogeneity, which are crucial for cloud computing.

The DETER testbed is an installation of the Emulab software. It is mainly used for security research, e.g., colleting 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. In Open Cirrus, all sites are loosely-coupled.

Amazon EC2 provides virtual machines on the pay-as-you-go basis. Though it allows complete control over the virtual machines, users cannot control the network resources, reducing the flexibility as a systems research testbed. Garth Gibson is leading an effort to recycle LANL's retiring clusters (typically with a few thousand machines) by making them available for systems research. Testbeds are compared in the Table below. There are also other efforts, such as Reservoir [13] and RightScale [14], but their description is beyond the scope of this paper.

6. Conclusion

In this paper we presented Open Cirrus, a federated testbed of distributed clusters for systems and applications research. Open Cirrus offers unique opportunities for conducting research that none of the previous or current testbeds have offered (federation of heterogeneous sites, systems and applications research, and datasets). In addition, it offers an open stack with non-proprietary APIs for Cloud Computing. Through shared innovation it offers an economical model for an increased impact on communities around the globe.

Acknowledgements

Partial funding for the Open Cirrus UIUC site was provided by NSF. We would like to recognize a number of people who have made significant contributions to Open Cirrus, including A. Chien, R. Gass, K. Goswami, C. Hsiung, J. Kistler, M. Ryan, C. Whitney, and J. Wilkes. Wilkes in particular was instrumental in formulating the original Open Cirrus vision, as well as the notion of the PRS.

References

[1] White, B., et al., "An Integrated Experimental Environment for Distributed Systems and Networks," OSDI, Dec.2002.
[2] Peterson, L., et al., "A blueprint for introducing disruptive technology into the internet," Proc. HotNets-I, Oct. 2002.
[3] Benzel, T., et al., "Design, Deployment, and Use of the DETER Testbed," Proc. of DETER Workshop, Aug 2007.
[4] Catlett, C. et al. "TeraGrid: Analysis of Organization, System Architecture, and Middleware Enabling New Types of Applic.," HPC and Grids in Action, Amsterdam, 2007.
[5] http://www.google.com/intl/en/press/pressrel/20071008_ibm_univ.html
[6] http://www.opencloudconsortium.org
[7] http://aws.amazon.com/
[8] Armbrust, M., et al., "Above the Clouds: A Berkeley View of Cloud Computing," UCB/EECS-2009-28
[9] Hamilton, J., "Cost of Power in Large-Scale Data Centers," http://perspectives.mvdirona.com/2008/11/28/CostOfPowerInLargeScaleDataCenters.aspx
[10] Hamilton, J. Internet-Scale Service Efficiency. Proceedings of the Large-Scale Distributed Systems and Middleware (LADIS) Workshop, September 2008.
[11] Greenberg, A., et al., "The Cost of a Cloud: Research Problems in Data Center Networks", ACM SIGCOMM CCR, Vol. 39, No. 1, January 2009.
[12] http://OpenCirrus.org/
[13] https://sysrun.haifa.il.ibm.com/hrl/reservoir
[14] http://www.rightscale.com/

Chara-cteristics	Testbeds							
	Open Cirrus	IBM/Google	TerraGrid	PlanetLab	EmuLab	Open Cloud Consortium	Amazon EC2	LANL/NSF cluster
Type of research	Systems & services	Data-intensive applications research	Scientific applications	Systems and services	Systems	interloper. across clouds using open APIs	Commercial use	Systems
Approach	Federation of heterog. data centers	A cluster supported by Google and IBM	Multi-site heterog. clusters supercomp	nodes hosted by research instit.	A single-site cluster with flexible control	Multi-site heterog. clusters	Raw access to virtual machines	Re-use of LANL's retiring clusters
Participants	HP, Intel, IDA, KIT, UIUC, Yahoo!	IBM, Google, MIT, Stanford, Washington	Many univ. & organizations	Many univ & organizations	University of Utah	4 centers –	Amazon	CMU, LANL, NSF
Distribution	6 sites	Centralized, one DC in Atlanta	11 partners in US	> 700 nodes world-wide	> 300 machines University@Utah	480 cores, distrib. in four locations	Several unified DCs	1000s older, still useful nodes at 1 site