

# “Powered by VCL” - Using Virtual Computing Laboratory (VCL) Technology to Power Cloud Computing

Mladen Vouk, Sam Averitt, Michael Bugaev, Andy Kurth, Aaron Peeler, Henry Shaffer, Eric Sills, Sarah Stein, Josh Thompson,

North Carolina State University, Raleigh, NC 27695

{vouk, sfa, mabugaev, andy\_kurth, fapeeler, hes, edsills, sstein, jfthomps}@ncsu.edu

## ABSTRACT

Virtual Computing Laboratory (VCL) – <http://vcl.ncsu.edu> is an award-winning open source implementation of a secure production-level on-demand utility computing and services oriented technology for wide-area access to solutions based on virtualized resources, including computational, storage and software resources<sup>1</sup>. Currently, NC State VCL is serving a student and faculty population of more than 30,000. There are VCL pilots with a number of University of North Carolina campuses, North Carolina Community College System, as well as with a number of out-of-state universities – many of which are members of the IBM Virtual Computing Initiative. The VCL technology can host practically any other overlay and virtualization solution, from single-seat desktops with an operating system of choice, to high-performance computing clusters. NC State has been researching this technology and operating it in production settings since 2004. It is an ideal technology for hosting “cloud” solutions of almost any type. In this white-paper we describe how VCL can (and already does) host some of the “cloud” computing environments proposed by various IBM and other initiatives.

## 1. Introduction



The model is a simple one, yet a very effective one. It was originally described by Averitt et al. in February 2004 [1]. VCL technology was developed to address the mission needs of the university and was implemented<sup>2</sup> by the NC State College of Engineering and Information Technology Division<sup>3</sup>. It had its first production users in the Fall 2004. Currently, NC State VCL is serving a student and faculty population of more than 30,000 [e.g., 2, 3, 7]. There are VCL pilots with a number of University of

<sup>1</sup> This work is supported in part by NC State University, NC Community College System, UNC GA, East Carolina University, UNC Greensboro, NC Central University, State of North Carolina, IBM, Intel, NetApp, SAS, Sun

<sup>2</sup> <http://vcl.ncsu.edu>

<sup>3</sup> Now Office of Information Technology (OIT)

North Carolina campuses, North Carolina Community College System, as well as a number of out-of-state universities – many of which are members of the IBM Virtual Computing Initiative. VCL technology can host practically any other overlay and virtualization solution, from single-seat desktops with an operating system of choice, to a collection of Globus<sup>4</sup>, Hadoop<sup>5</sup>- or Condor<sup>6</sup>-based platforms, to high-performance computing clusters. It is an ideal technology for hosting “cloud” solutions of almost any type [7]. In this paper we describe how VCL can (and to a large extent already does) host such concepts<sup>7,8,9,10,11</sup>.

## 2. “Cloud Computing”

“Cloud” computing builds upon decades of research in virtualization, distributed computing, „grid computing,“ utility computing, and more recently networking, web and software services. It implies a service oriented architecture, reduced information technology overhead for the end-user, greater flexibility, reduced total cost of ownership, economy of scale, on-demand services and many other things.

Some of the principal hallmarks of a “cloud” are abstraction and hiding of complexity, use of virtualized resources, and efficient (re-)utilization of distributed

<sup>4</sup> Globus: <http://www.globus.org/>

<sup>5</sup> Hadoop: <http://hadoop.apache.org/core/>

<sup>6</sup> Condor: <http://www.cs.wisc.edu/condor/>

<sup>7</sup> IBM Cloud Computing, 8 October 2007, white paper

<sup>8</sup> IBM, “IBM Introduces Ready-to-Use Cloud Computing,” Nov 15, 2007,

<http://www-03.ibm.com/press/us/en/pressrelease/22613.wss>

<sup>9</sup> IBM, “IBM, EU Launch Research Initiative for Cloud Computing,” in Grid today, February 5, 2008, <http://www.gridtoday.com/grid/2102294.html>.

<sup>10</sup> Amazon Elastic Compute Cloud: <http://www.amazon.com/gp/browse.html?node=201590011>

<sup>11</sup> The NetApp Kilo-Client: [http://partners.netapp.com/go/techontap/tot-march2006/0306tot\\_kilo.html](http://partners.netapp.com/go/techontap/tot-march2006/0306tot_kilo.html)

(Published in ICVCI 2008, May 16-17, Proceedings, pp 1-10)

resources [7]. The key to the solution is an integrated framework that allows reliable, flexible and reconfigurable aggregation, sharing, and allocation of software, computational, storage and networking resources on-demand.

The key characteristic of such a framework and its elements is its component-based nature [8,9,10]: **Reusability** (elements can be re-used by others), **substitutability** (alternative implementations are easy to insert, very precisely specified interfaces are available, run-time component replacement mechanisms exist, there is ability to verify and validate substitutions, etc), **extensibility** (ability to readily extend system component pool, increase capabilities of individual components, have an extensible architecture that can automatically discover new functionalities and resources, etc), **customizability** (ability to customize generic features to the needs of a particular scientific domain and problem), **composability** (easy construction of more complex functional solutions and workflows using basic components, reasoning about such compositions, etc.) and **scalability** (ability to scale solutions to achieve economy of scale).

The framework must be dependable, we should be able to use it to provide individual and workflow-based services, and must allow a range of couplings among its building blocks – from loose to tight. It also must be fault-tolerant, and it must have data- and process-aware service-based delivery as well as an ability to audit processes, data and results. Virtual Computing Laboratory (VCL) represents such a framework.

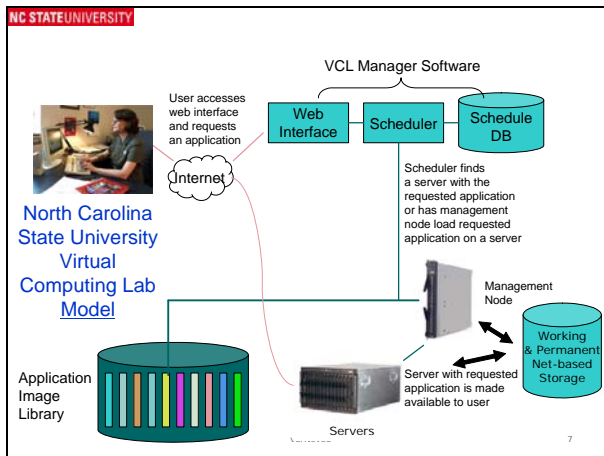


Figure 1. Illustration of VCL physical architecture (single site).

### 3. Virtual Computing Laboratory

VCL delivers a range of functionalities and services that map well onto the cloud computing requirements and expectations. VCL services range from single real or virtual computer laboratory “seats” or desktops, to single applications on-demand, to classroom size groups of seats,

to enterprise server solutions, to homogenous and non-homogenous server clusters for research and production, to high-performance computing services and clusters (including grid-based services).

There are four principal components in the VCL architecture (Figure 1): a) an end-user **access interface** (web-based or API based), b) a **resource-manager** (or VCL Manager) which includes a scheduler, security, multi-site coordination, performance monitoring, virtual network management, etc., c) an **image repository**, and d) computational, storage and networking **hardware**.

#### 3.1 Users

In describing the VCL architecture and the mapping of it services, we will focus on four principal non-exclusive categories of its users: end-users of services such as computational “desktops”, service provisioning and integration personnel (people that help manage access to resources and perhaps compose them into workflows), authors of services (or image creators), and VCL framework developers.



Figure 2. VCL web portal.

A typical user accesses VCL through a web interface (portal – see Figure 2) to select from a menu a combination of applications, operating systems and services that are needed (Figure 1). If a specific combination is not already available as an “image”, an authorized user can construct one’s own from the VCL library of components. This customization capability is very much in the spirit of what services engineering and management is all about. VCL

(Published in ICVCI 2008, May 16-17, Proceedings, pp 1-10)

Manager software then maps that request onto available software application images and available (possibly heterogeneous) hardware resources, and schedules it for either immediate use (on demand) or for later use.

End-users have a limited set of privileges. Their menu may look something like what is shown in Figure 3. They can make reservations for “now” or for sometime in the future (Figure 3), check on current reservations (Figure 4), set her user and Windows RDP preferences (Figure 5), view system statistics (Figure 6), and access help.

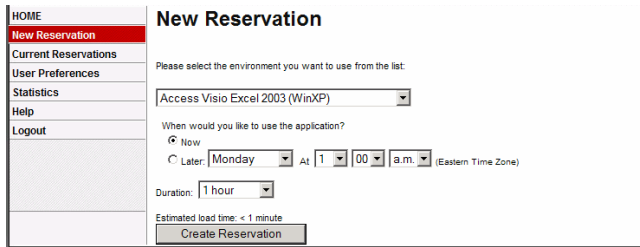


Figure 3. Make an “image” reservation.

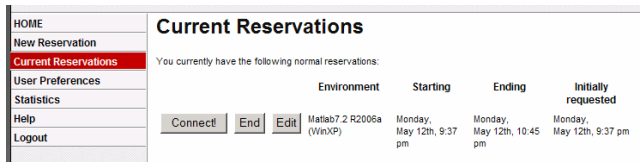


Figure 4. View current resource reservations

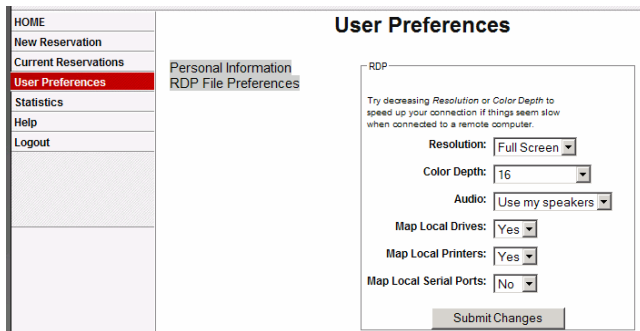


Figure 5. Setting RDP file preferences

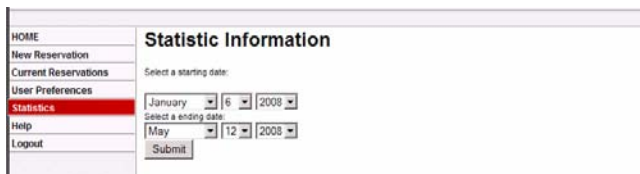


Figure 6. Select the range for statistics.

The mode of access to the resources will depend on what the service offers. It may range from RDP or VNC type of access to a remote desktop, to an ssh-based or X-Win access to a Linux service, to web-based access, to proxy access to a computational cluster.

### 3.2 Virtualization

VCL architecture is intended to support the abstraction of resources at several levels [6]: at the application and

operating system level (e.g., via images and hypervisors), at the hardware location level (VCL manager), and at the network level (virtual networks, VLANs, VPNs, etc.). The VCL technology bundle is in many ways a “shim” or “driver” between more traditional virtual environments, operating systems, and networks and the underlying hardware. It abstracts software from hardware, but from the end-user perspective it still delivers the composite of the software and hardware stacks (Figure 7). It is therefore agnostic to both end-user access platform (e.g., laptop, PDA) agnostic, and (to a very large extent) software and service manufacturer/provider agnostic.

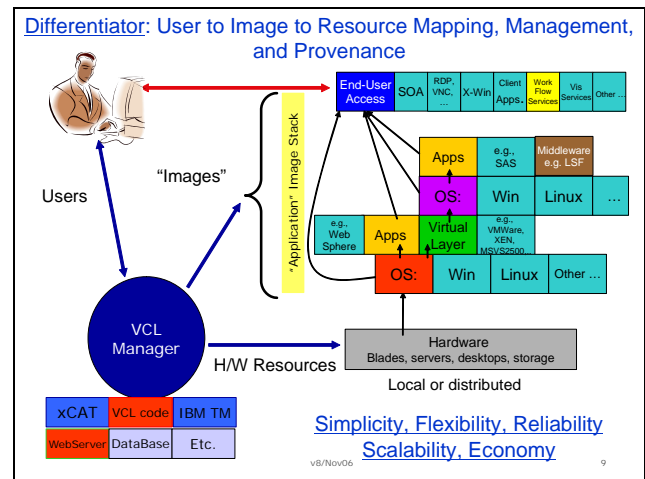


Figure 7. Illustration of the VCL logical architecture.

An “image” (Figure 7) is a software stack that incorporates

- any base-line operating system, and if virtualization is needed for scalability, a hypervisor layer,
- any desired middleware or application that runs on that operating system, and
- any end-user access solution that is appropriate.

Images can be loaded on “bare-metal”, or into an operating system/application virtual environment of choice.

When as user has the right to create an image, that user usually starts with a “NoApp” or base-line image (e.g., Win XP or Linux) and extends it with his/her applications. Similarly, when a user constructs composite images (aggregates of two or more images), the user extends service capabilities of VCL.

A user can have either sole use of one or more hardware units, if that is desired, or the user can share the resources with other users. Scalability is achieved through a combination of multi-user service hosting, application virtualization, and both time and CPU multiplexing and load balancing.

By design, VCL subsumes and allows any of the “software as a service” solutions, grid solutions, virtualization solutions, and terminal services solutions available today –

(Published in ICVCI 2008, May 16-17, Proceedings, pp 1-10)

e.g., VMWare, XEN, MS Virtual Server, KVM, Virtuoso, and Citrix to mention a few. It also allows any of the suitable access/service delivery options – e.g., from RDP or VNC desktop access, to X-Windows, to a web or other services.

**VCL Manager** software was developed by NC State University using a combination of off-the-shelf IBM products (such as IBM xCAT<sup>12</sup> and IBM Tivoli Monitoring<sup>13</sup>), in-house developed “glue” or middleware software (including vclsd), an open source web server (Apache), and an open source data base (MySQL). All components of the VCL can be (and are) distributed (Figure 8). There are two principal parts to a manager – an image loader and platform manager (in the current implementation the latter is xCAT), and a resource and image scheduler and mapper. However, the architecture allows substitution of different (possibly proprietary) shim components. For example, xCAT, while used in most of our current implementations can be (and is in several of our pilots) replaced by another image loading solution, including an interface to VMware management software.

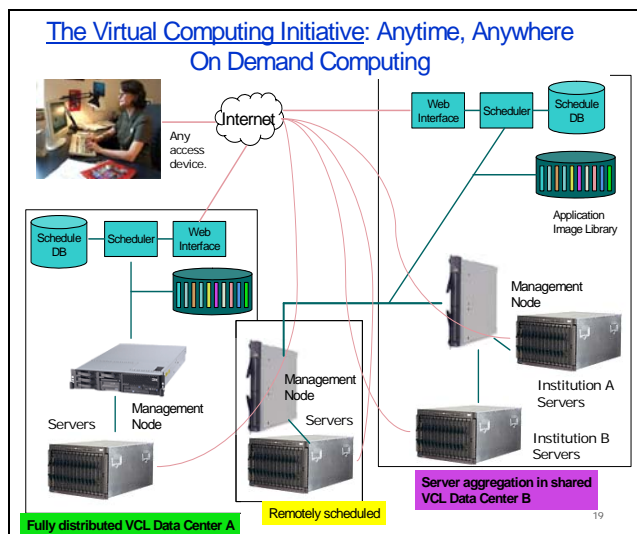


Figure 8. Distributed VCL

### 3.3 Advanced User Functions

Advanced users can access and manage an increasingly larger number of VCL functionalities and resources.

For example, a high-end user’s left-hand main menu bar may look something like Figure 9. Such a user can, in addition to the functions already mentioned, create extended reservations (by date) rather than over just a few hours (Figure 9), create block reservations (Figure 10), manage groups (Figure 11), manage images (Figure 12, 13), manage availability schedules of resources used by

images (Figure 14), manage resources (Figure 15), manage VCL management nodes, view resource time-tables, set privileges for other users (Figure 16), and identify users.

#### 3.3.1 Block Reservations

**Create and Edit Block reservations (Figure 10).** The block reservation interface is used to create scheduled preloads for classroom sessions, workshops, or specific group events in which a large number of compute resources are required to be preloaded with an application environment and then be ready for immediate use at the desired time. This process guarantees that the selected user group will have priority access to the allocated resources during the time of the block reservation. At the end of the defined block reservation time and when the users have completed their individual reservations, the compute resources are returned to the general compute resource pool.

Figure 9. Single-seat reservation interface for a service author (image creator), integrator and provisioning manager.

#### 3.3.2 Create and Edit User and Resource Groups

The Manage Groups interface provides the ability to create, edit and delete groups (Figure 12). There are two basic groups, **users** and **resources**. The user groups simply contain locally created user groups that can be managed by a third party service or through the VCL admin interface. The user groups are assigned throughout the VCL admin interface, in the privilege tree, as owners of other groups, block reservations, etc. VCL also utilizes groups defined in LDAP, i.e. if a site has a general LDAP service normally the users are assigned to a site group. Within the privilege tree, this group can be used to provide authorization to check out VCL application environments.

A **resource group** can be one of four types: computer, image, schedule, and management node. The computer and image groups are the most common. The description of each is below:

- **Computer group** – contains physical or virtual machines, used in Manage Computers and Manage Images, and Privilege Tree.

<sup>12</sup> <http://www.xcat.org/>

<sup>13</sup> <http://www-306.ibm.com/software/tivoli/products/monitor/>

(Published in ICVCI 2008, May 16-17, Proceedings, pp 1-10)

- Image group - contains images, used in Manage images and Privilege Tree
- Schedule group – simply a defined schedule group that contains schedules created under Manage schedules. The schedule group is listed in the privilege tree so other VCL administrators can manage the schedule.

Management node group – a special computer group that contains one or more management nodes, used in Management nodes and privilege tree.

Figure 10. Block reservation page

User Groups

Name	Owner	Editable by	Initial Max Time (minutes)	Total Max Time (minutes)	Max. Extend Time (minutes)	Max. Overlapping Reservations
10 Reservations@NCSU	rlromps@NCSU	vcldev@NCSU	240	360	30	10
2 Reservations@NCSU	lapeleler@NCSU	vcldev@NCSU	240	360	30	2
30 Reservations@NCSU	rlromps@NCSU	vcldev@NCSU	240	360	30	30
Accessibility_start@NCSU	lprms@NCSU	vcldev@NCSU	240	360	60	0
ArcGISwSAS@NCSU	lprms@NCSU	vcldev@NCSU	240	360	60	0
BAE-fault@NCSU	lq@NCSU	vcldev@NCSU	240	360	60	0
BAE-stung@NCSU	lq@NCSU	vcldev@NCSU	240	360	60	0
BAE_anderson_agents@NCSU	lq@NCSU	vcldev@NCSU	240	360	60	0
BioChem image managers@NCSU	lapeleler@NCSU	vcldev@NCSU	240	360	60	0
Blaze@NCSU	lq@NCSU	SCUL_users@NCSU	240	240	120	0
DMES@NCSU	rlromps@NCSU	vcldev@NCSU	240	360	60	0
DME_LINCarrng200@NCSU	lapeleler@NCSU	vcldev@NCSU	240	360	60	0

Figure 11. Manage User and Resource Groups.

### 3.3.3 Image Management

The **Manage Images interface** has four main functions:

- **Create / Update an image** – This tool is very similar to the general VCL reservation function but with the image flag enabled. Provided a user or group of users have permission to create images, they can select from a base “No apps”

template or use an existing environment they can manage, the user then proceeds to create a new environment (OS + applications) or update an existing image.

- **Edit Image Information** – Provides easy method to change the attributes of an image, the image name, owner, hardware requirements, deletion, etc. Image attributes (metadata) are an important mechanism for image association and management (Figure 14).
- **Edit Image Grouping** – This matrix allows adding individual images to logical image groups. For example, a unit or department (image group) might own 20 application environments or images. By grouping the images, VCL administrators can easily define who can access all images or what hardware those images can be loaded on to, etc.
- **Edit Image Mapping (Figure 13)** – Maps image groups to physical computer groups. In this way VCL generically handles an extensive range of OS’s and hardware. Once an image group is mapped onto a computer group, this mapping informs the VCL scheduler that any environment in a mapped image group can be loaded onto any computer in the mapped computer group.

Image Group	CNR blades	ECU blades	ECU vmmware vms	Education Blades	ENGR blades	ENGR blades 2	ENGR blades 3	ENGR blades 4	ENGR linux lab	ENGR solaris lab
Accessibility FC5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Accessibility	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Alice NCCU	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ArcGISwSAS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Autocad	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Autodesk	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
BAE images	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
BioChem images	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CCE Images	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ced images	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
CHASS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figure 12. Excerpt from Image Group to Hardware mapping table

Image creation mimics the procedure of installing or changing software on one’s own computer. Once chooses an existing image (“No Apps” or one with apps) and then installs, updates, uninstalls, etc, as one would do on one’s computer. Then the image is saves, using standard imaging methods, tested and then released into production use. The production image, along with its previous versions is stored in the image library space.

The VCL currently supports multiple image formats and has the potential to use many more. The most common types are kickstart based installs for Linux distributions that

(Published in ICVCI 2008, May 16-17, Proceedings, pp 1-10)

use kickstart, disk images using partimage<sup>14</sup>, vmware disk images for both Vmware Free server and ESX standard. In addition to these image types, VCL also brokers remote access to standalone machines, by interacting with a vclclient daemon installed on the remote device. This is ideal for utilizing traditional computing lab machines in an academic setting. In a classical lab setting a student may not have root or administrative control over the machine. In full VCL setting, they typically do (but then after the use, the machine is re-loaded with a clean image). Similarly, this more of access – non-root mode – is a good way of approaching integration of resources such as z-Series LPARS.

Normal academic student use of VCL is to broker access to any of the environments for a relative short time period ranging 1-6 hours, the advanced usage of VCL allows for long-term renewable sessions. This scenario is primarily used for resources owned by research projects or for special case use.

Figure 13. Illustration of a multi-image “image”, or an ENVIRONMENT.

### 3.3.4 Environment

Figure 13 illustrates a very interesting concept. Composite “images” or environments. Those are groups of images that are always deployed together and act together. In the example, there is a master-image called MPLS Netlabs Testbed 2 (owner fapeeler) that requires at least on GB of memory, at least a 1 GHz processor, at least 100 Mbps

<sup>14</sup> Partimage: <http://www.partimage.org>

network connectivity, does not have any licensing restrictions, if loaded on “bare metal” may take about 10 minutes to load, is allowed for general checkout, and is not keyed to a single user. This image has five child-images or subimages. When the MPLS Netlabs Testbed2 image is requested, it automatically initiates load of additional five other (bare-metal) images. This environment then operates as a unit. When the reservation time is over, all images are unloaded from the system.

### 3.3.5 Resource Availability

**Create a new schedule and add to schedule group (Figure 14).** This interface defines when computer resources are made available, e.g., on 24x7 basis or only at specific times throughout a day. As an example resources in a traditional computer lab are normally open to walk-in users from 9-6pm. A daily schedule would be created with a VCL window of 6pm until 9am the next day, so that those lab machines would be remotely available during the lab closing hours.

Figure 14. Illustration of the interface that schedules the group of computers located in Daniels 200 public lab for use over VCL.

### 3.3.6 Manage Computers

The **Manage Computers** interface has three functions:

- **Edit Computer Information (Figure 15)** – Allows the VCL computer owner or administrator to add, remove and edit attributes about a computer or some other resources. The attributes include memory, process, network speed, schedule, owner, preferred image, etc. This information is very useful in management of both single and composite images. For example, Figure 16 shows an MCNC based computer owned by fapeeler that is currently in use as a VMWare hypervisor host (which is also the preferred image

(Published in ICVCI 2008, May 16-17, Proceedings, pp 1-10)

for that host). It is blade on a 24x7 schedule with 4 GB of memory and two 2.8 GHz processors and 1GB network NIC.

- **Edit Computer Grouping** – Populates individual computers into computer groups. Used in mapping of Image groups to Computer groups, assigning computer management rights to a VCL site administrators group.
- **Computer Utilities** – An administrative tool used to select individual or a large number of compute resources to reload with a specific environment, change the state or change the schedule.

### Edit Computer

**Hostname:**   
**IP Address:**   
**State:**   
**Owner:**   
**Platform:**   
**Schedule:**   
**Current Image:** VMware104hostELsmp  
**Preferred Image:**   
**RAM (MB):**   
**No. Processors:**   
**Processor Speed (MHz):**   
**Network Speed (Mbps):**   
**Computer ID:** 599  
**Type:**

Figure 15. Illustration of the resource attribute interface.

### 3.3.7 Management Nodes

The **management nodes** (MN) interface allows adding of new management nodes to the VCL database, addition of management nodes to management node groups, and mapping of which management node groups can control which computer groups. The specific functions are:

- **Edit Management node Information** – add, change state of MN
- **Edit Management node Grouping** – put management node in a MN group
- **Edit Management node Mapping** – Provides a mapping of MN group to computer groups. Tells the VCL scheduler which MN can control individual compute resources. A management node can control more than one computer group. For example data center blades servers might be grouped by chassis c1,

c2, c3 and management node 1 is in a MN group that is mapped to control all three computer groups. Another example: there are 1000 lab machines in a lab group; there are 4 management nodes in four different MN groups. Each MN group is mapped to manage any of the 1000 machines in the lab group, thus providing a built-in fail over mechanism. The VCL scheduler then makes decisions on which management node a request is assigned to.

### 3.3.8 Privilege Tree

The privilege tree is the core method of providing the different authorization levels required to check out environments and administer a VCL system. It is a directory service based model, in which permissions and resources (image, compute, schedule, management node) can be inherited throughout sub nodes or branches. The tree begins with a simple node containing both user or group level privileges and resource attributes.

The screenshot shows the 'Privilege Tree' interface. On the left is a navigation menu with options like HOME, New Reservation, Current Reservations, Block Reservations, User Preferences, Manage Groups, Manage Images, Manage Schedules, Manage Computers, Management Nodes, View Time Table, Privileges (highlighted), User Lookup, Statistics, Help, Documentation, and Logout. The main area shows a tree view with nodes like ECU, Johnston CC, NCCU, NCSU, UNCG, and WakeTech. Below the tree is a table titled 'Privileges at Selected Node' for 'Users'. The table has columns for various privilege types and rows for users like allenp@ECU, boycee@ECU, creasey@ECU, creechd@ECU, daughtry@ECU, hodge@ECU, norrisj@ECU, and thornstons@ECU. Each cell in the table contains a checkbox, some of which are checked.

Figure 16. Illustration of the interface for setting different levels of user, group and resource privileges.

The user or group privileges are:

- **nodeAdmin** – ability to Add / Delete privilege tree nodes
- **userGrant** – manage user and user group privileges at a given privilege node

The following privileges require a corresponding resource attribute to be meaningful.

- **computerAdmin** - to use the Manage Computers tool
- **groupAdmin** – to use the Manage Group tool

(Published in ICVCI 2008, May 16-17, Proceedings, pp 1-10)

- **imageAdmin** – to create or manage images owned or have administer resource attributes rights over
- **imageCheckout** – most common, for checking out environments
- **mgmtNodeAdmin** - to use the Management Node tool
- **resourceGrant** – Add resource Group and modify resource group privileges
- **scheduleAdmin** – use the Manage Schedules tool

The resource attributes are:

- **available** – grant use access to resources in image or computer groups at a specific privilege node
- **administer** – allow administrative control over resources in listed resource groups at a specific privilege node
- **manageGroup** – allows site admin to perform resource mapping, grouping, and administration of resource groups

### 3.4 Hardware

Computational hardware and storage can be any appropriate platform – from a blade center, to a collection of diverse desktop units or workstations, to an enterprise server, to a high-performance computing engine. Storage, communications and other hardware are subsumed. VCL Manager provides appropriate virtualization (aggregation, dis-aggregation, abstraction, isolation, etc.) of the available hardware resources before mapping the requested image onto that hardware. VCL services focus on controlling the resource at the platform level. As such, VCL subsumes solutions such as Citrix and hypervisor-based implementations (e.g., VMWare, Xen and Microsoft’s Virtual Server based solution). Since many of these solutions are simply just environments (possibly multi-server) that can be dynamically loaded on the BladeCenter servers, these solutions can become just another level in the VCL “image stack”.

A typical VCL installation will have one or more blade chassis with at least one of the blades being designated as the management node. Often another unit may serve as user-interface delivery platform (web) and data-base server. Each blade has at least two networking interfaces – one for the public network, and the other for a private network that is used to manage the blades and load images. Storage is attached either directly through fiber or via network.

### 3.5 HPC

The NC State VCL implementation extends from “single seat” desktops to HPC offerings (Figure 17). In the HPC case blades are loaded with the HPC Linux computational images. The latter group of services operates on clusters of

machines/blades isolated through an HPC VLAN. The only public access is through login nodes. At NC State the scheduler of choice is LSF, but any other job scheduling solution can be used.

The integration of HPC significantly increases the reuse of blades and allows the infrastructure to be shared, leading to greatly enhanced usage levels. This provides significant economies.

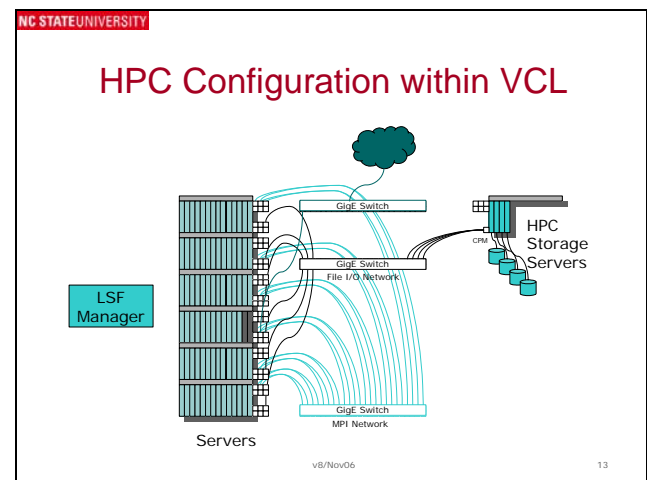


Figure 17. VCL integrates with our HPC structure.

### 3.6 Heterogeneous Resource Clouds

When VCL images are equipped with software that allows formation of overlay networks and clouds (e.g., based on Globus, Hadoop, or Condor), then those images, environments, or clusters can operate as parts of Globus Grids, Hadoop-based clouds, Condor groups, or as members of any other type of cloud.

From the perspective of VCL, it can transform into and support any type of environment so long as an image with the appropriate environment manager is provided.

For example, there is nothing to prevent formation of a z-Cloud, or a grouping of z-Series LPARS that would be accessed through a specialized VCL login node image. LPARS could be assigned as no-root public laboratory resources and their creation and administrative management would be left to z-Series admins. In fact, we do have similar images for teaching about z-Series machines.

A step further is to form workflow-based environments where, for example, a Kepler-based workflow management image [10] controls one or more z-Series LPARS, a cluster of Cell (multicore) processors, and a cluster of classical blades. VCL tools that can be used to form these composite environments are group reservation functionality, HPC cluster functionality, and Environment (parent-child) functionality.



(Published in ICVCI 2008, May 16-17, Proceedings, pp 1-10)

### 3.7 Storage

VCL users can a) access storage on platforms on which images run (typically blades), b) corporate network-based storage, and c) storage on their own access platforms (e.g., laptops).

NC State VCL images are constructed with storage access modules appropriate for the environment and institution they operate in. At NC State University, we equip Windows images with applications that access our corporate/enterprise level storage, as well as any network accessible storage anywhere. WolfCall<sup>15</sup> is an application that works with an AFS Client and the Kerberos for Windows 2000 and XP software that lets our users access their NC State allotted AFS space. Our Linux/SunOS images have a similar access our AFS space. We also construct our images with a more general GUI-based applications, such as SSH Secure File Transfer Clients or PUTTY<sup>16</sup>. Images that use RDU have an added option of using RDU-based remote storage on the access platform storage devices (e.g., end-user drives, memory key, and similar).

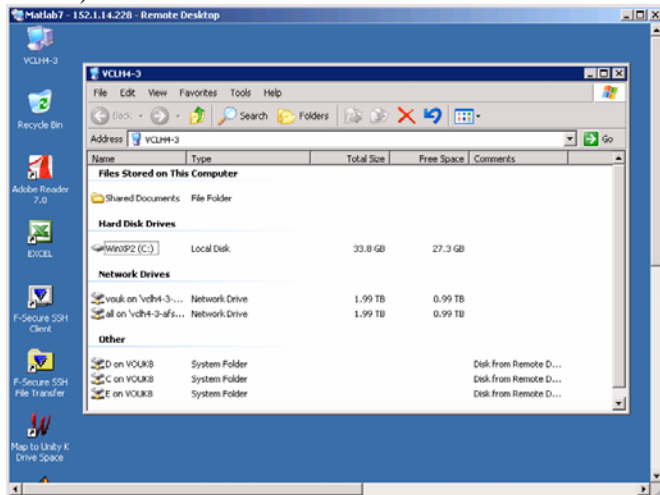


Figure 18. Via RDP VCL Windows images can access NC State Enterprise Storage (Network Drives), local storage (Hard Disk Drives), and remote access platform local drives (D and C) and memory key (E). Via F-Secure SSH File Transfer software they can access any ssh-accessible storage in the world. Similar arrangements exist for our Linux images.

### 3.11 Monitoring and Provenance

VCL offers a rich set of usage and performance monitoring tools. System profiling and provenance information is used to optimize scheduling of the requests and problem resolution. Profiling and provenance is provided through a

combination of image creation logs and retention of older images (allowing rollback if errors are discovered), system logs, image usage statistics and operational profiles.

For example, IBM Tivoli Monitoring software equipped images were used in a pilot to assess image performance needs and thus make appropriate virtualization decisions.

### 3.12 Open Source

VCL is currently available under Eclipse open source license.

### 3.13 Security

As with any networked system, security is of great importance and continuously needs preventive action. The VCL uses simple but effective security measures in both authentication and authorization of services.

Authentication is affiliation based LDAP, VCL supports the ability to bind to different LDAP services where appropriate and separate user's access to resources based on that affiliation. In addition to LDAP, VCL also supports other authentication modules. The environment level authentication is determined at image creation time, depending on the environment, for example the current windows based images make use of single one-time accounts that get created at reservation time. The Linux environments can make use of existing authentication infrastructure or can also use standalone account mechanism.

After a user is authenticated and allowed to make a VCL reservation, VCL IP-locks the provisioned environment to end-user IP address using OS level firewall. During the reservation process, and in the normal reservation mode, the user is required to acknowledge the reservation request. During this acknowledgment the web application captures the visitor's address, the management node then proceeds to modify the OS level firewall by opening the correct service port for that remote IP address to connect to the VCL compute node.

Currently the primary access methods use ssh (secure shell) and remote desktop connection. In the latter case, we also implement one-time passwords as an added security method. Other access methods are being considered.

Another important factor is that the VCL compute blade resources are reloaded once the end-user has completed their reservation removing any residual data from one user to the next.

Still there are other security methods available, such as using VPN's, private VLAN's, ssh-tunneling, etc. Recently NCSU has been awarded federal funding to address security concerns in Open Systems and open-source software, thus the Secure Open Systems Initiative

<sup>15</sup> <http://www.eos.ncsu.edu/wolfcall/>

<sup>16</sup> <http://en.wikipedia.org/wiki/PuTTY>

(Published in ICVCI 2008, May 16-17, Proceedings, pp 1-10)

(SOSI)<sup>17</sup> has been established with one of the goals being a security hardened version of VCL

#### 4. SUMMARY

Virtual Computing Laboratory (VCL) – <http://vcl.ncsu.edu>, is an open source implementation of a secure production-level on-demand utility computing and services oriented technology for access to solutions based on virtualized resources. Currently, it is serving the NC State student and faculty population, nominally 30,000+ users.

The benefit of this scalable and flexible shared open source solution, to which we hope all Virtual Computing Initiative (VCI) participants who wish to do so will also contribute as developers as well, are numerous. They include management and provisioning that can be as centralized or as distributed as one wishes, considerably increased utilization of the resource and thus reduced cost of service delivery, reduced information technology overhead on institutions that may not have appropriate resident personnel for construction and maintenance of very complex information technology services, provision to end-users of a very broad palette of computational services on-demand - thus leveling the field at an affordable cost, individualization and customization that each institution and user group needs, and provision of a viable disaster recovery and business continuity solutions across all participating institutions.

Computing development is moving toward the rich services environment made available through Cloud-like access. VCL is uniquely constructed to meet the needs of education, as well as commercial industries. It takes the capabilities of HPC needed for research and high-end industry operations and combines them in a vastly-customizable, agile, efficient, effective, and extensible computational architecture. It is the way of the future.

#### 5. Acknowledgments

VCL development was supported through multiple IBM Shared University Research (SUR) equipment grants. Additionally, funding from the IBM RTP Center for Advanced Studies (CAS) has supported numerous research projects associated with the VCL. IBM software used in the lab (including the IBM Tivoli Monitoring and WebSphere Application Server middleware) was donated by the IBM Academic Initiative, a program that provides IBM software free for use in the classroom or in research, with an aim toward developing curricula that addresses critical industry skills.

#### 6. References

1. UNCGA grant proposal entitled “Support for Distributed Scalable High Performance and Grid-based Computing and Problem Solving with Emphasis in Science and Engineering,” by Sam Averitt, Mladen Vouk, Henry Schaffer, Eric Sills, and Gary Howell, NC State University, Raleigh, February 2004.
2. Sam Averitt and Mladen Vouk, “The Digital Campus utilizing Virtual Computing Environment,” SUGI31, San Francisco, CA, March 26-29, 2006 (**Invited Talk**)
3. Mladen Vouk, "Automation of Large-scale Network-Based Scientific Workflows," Microsoft eScience Workshop, The Johns Hopkins University, Baltimore, Maryland, October 13 - 15, 2006, p21
4. Peeler, A., S. Averitt, W. Baudoin, M.I Bugaev, G. Howell, E. Sills, H. Schaffer, J. Thompson, M. Vouk. Virtual Computing Laboratory at NCSU. *UNC CAUSE 2006* (<http://www.ecu.edu/cause06/>), Nov 2006.
5. Henry Schaffer, Sarah Stein, Aaron Peeler, Samuel Averitt, Mladen Vouk, Eric Sills, “Connecting to High-End Software Applications,” UNC Teaching and Learning with Technology Conference, Raleigh, NC, March 21-23, 2007.
6. Sam Averitt, Michale Bugaev, Aaron Peeler, Henry Schaffer, Eric Sills, Sarah Stein, Josh Thompson, Mladen Vouk, “The Virtual Computing Laboratory,” Proceedings of the International Conference on Virtual Computing Initiative, May 7-8, 2007, IBM Corp., Research Triangle Park, NC, pp. 1-16.
7. Mladen Vouk, “Cloud Computing –
8. [I. Crnkovic](#) and [M. Larsson](#) (editors), *Building Reliable Component-Based Software Systems*, [Artech House Publishers](#), ISBN 1-58053-327-2, 2002, <http://www.idt.mdh.se/cbse-book/>
9. CCA, <http://www.cca-forum.org/>, accessed February 2006
10. B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger-Frank, M. Jones, E. Lee, J. Tao, Y. Zhao, “Scientific Workflow Management and the Kepler System,” *Concurrency and Computation: Practice & Experience*, Special Issue on Workflow in Grid Systems, Volume 18, Issue 10 (August 2006), Pages: 1039 – 1065, 2006.

---

<sup>17</sup> [http://www.engr.ncsu.edu/news/news\\_articles/sosi.html](http://www.engr.ncsu.edu/news/news_articles/sosi.html)