

# Cyberaide Virtual Appliance: On-Demand Deploying Middleware for Cyberinfrastructure

Tobias Kurze<sup>1</sup>, Lizhe Wang<sup>3</sup>, Gregor von Laszewski<sup>2</sup>, Jie Tao<sup>1</sup>, Marcel Kunze<sup>1</sup>, Fugang Wang<sup>3</sup>, David Kramer<sup>4</sup>, Wolfgang Karl<sup>4</sup>, and Jaliya Ekanayake<sup>2</sup>

<sup>1</sup> Steinbuch Center for Computing, Karlsruhe Institute of Technology, Karlsruhe 76344, Germany

<sup>2</sup> Pervasive Technology Institute, Indiana University at Bloomington, Bloomington, IN 47408

<sup>3</sup> Service Oriented Cyberinfrastructure Laboratory, Rochester Institute of Technology, Rochester, NY 14623

<sup>4</sup> Department of Computer Science, Karlsruhe Institute of Technology, Karlsruhe 76131, Germany

**Abstract.** Cyberinfrastructure offers a vision of advanced knowledge infrastructure for research and education. It integrates diverse resources across geographically distributed resources and human communities. Cyberaide is a service oriented architecture and abstraction framework that integrates a large number of available commodity libraries and allows users to access cyberinfrastructure through Web 2.0 technologies. This paper describes the Cyberaide virtual appliance, a solution of on-demand deployment of cyberinfrastructure middleware, i.e. Cyberaide. The proposed solution is based on an open and free technology and software – Cyberaide JavaScript, a service oriented architecture (SOA) and grid abstraction framework that allows users to access the grid infrastructures through JavaScript. The Cyberaide virtual appliance is built by installing and configuring Cyberaide JavaScript in a virtual machine. Established Cyberaide virtual appliances can then be used via a Web browser, allowing users to create, distribute and maintain cyberinfrastructure related software more easily even without the need to do the “tricky” installation process on their own. We argue that our solution of providing Cyberaide virtual appliance can make users easy to access cyberinfrastructure, manage their work and build user organizations.

**Keywords:** JavaScript, Cyberinfrastructure, Virtual Appliance.

## 1 Introduction

Research topics of modern cyberinfrastructure cover the development of a wide range of computing and information processing infrastructures and services, for example, advanced data acquisition/storage/processing/visualization, high performance computing environments for advanced scientific & engineering applications, and other networked services. There are a lot of scenarios where an advanced cyberinfrastructure is needed. Especially in the scientific domain such

infrastructures are of great use. Nevertheless, users who want to use an advanced cyberinfrastructure may find it hard to use or are just unable to use it due to the complexity involved. This is where the Cyberaide toolkit becomes handy. Cyberaide is a lightweight middleware for users to access advanced modern cyberinfrastructure. It provides various user interfaces and toolkits for users to access resources, submit jobs and manage user organizations. However it takes some time and effort to set it up. For users of Cyberaide it would be much easier and more convenient to have a tool that installs and deploys Cyberaide automatically. As the services provided by Cyberaide can be accessed through a web interface it is an obvious choice to deploy Cyberaide as a virtual appliance.

A virtual appliance provides a lot of benefits compared to “traditional software”: simple installation and setup; very easy deployment process; helping augment the efficient utilization of underlying hardware [1]. The appliance can be tested to ensure that all the software is working correctly on the hardware described by the virtual appliance itself. Additionally a virtual appliance is maintained by the creators of the appliance and not by the users, allowing to create correct updates with higher confidence [2].

We propose to use virtual appliances to simplify users’ work in accessing cyberinfrastructures. The solution is to create a Cyberaide Virtual Appliance on the fly, then to deploy and configure it, and finally to expose its services to the users. With the cyberaide virtual appliance, users can on-demand build middleware to access production cyberinfrastructure, like TeraGrid, and organize their work on the cyberinfrastructure.

The rest of the paper is organized as follows. Section 2 introduces related work and provides background information. This is followed by an overview of the Cyberaide Toolkit in Section 3. Section 4 illuminates the proposed Cyberaide Virtual Appliance solution, followed by some test results in Section 5. Section 6 concludes with a short summary and an outlook about future work.

## 2 Background and Related Work

Advanced cyberinfrastructure facilitates the development of new applications, allows applications to interoperate across institutions and disciplines, insures that data and software acquired at great expense are preserved and easily available and empowers enhanced collaboration over distance, time and disciplines. Cyberaide is a service oriented architecture that enables users to access and manipulate cyberinfrastructure resources. The Cyberaide toolkit originates from grid portal framework and Globus Toolkit project. The development of grid portals has started some time ago and one of the first usable libraries supporting grid portal development was the Java CoG Kit [3]. Another important resource is the TeraGrid portal [4] that is used to obtain access to grid resources. TeraGrid itself uses the Globus Toolkit [5] to manage resources. Unfortunately this solution doesn’t integrate JavaScript. To tackle this issue Cyberaide Javascript [6] has been developed. It is a grid abstraction framework that enables the easy access to grids through JavaScript.

Another important technology base for building the Cyberaide Virtual Appliance is virtualization. One of the earliest VM solutions has been provided by IBM in 1972 [7]. Nowadays different solutions are available that can be classified into several categories according to the virtualization technique used. One solution is XEN [8], which uses the paravirtualization [9] technique. Another virtualization technique is Full Virtualisation. VMware Server [10], KVM [11] and Microsoft Virtual PC [12] are representatives of this category.

To reduce the complexity of software development a relatively new approach is to use Virtual Appliances. Some software systems are difficult to compile, to link, and to install and have been well tested just on a specific version of tools and platforms. A software publisher can bundle the necessary tools in an appliance and distribute it to users [13].

The Grid Appliance is an example of a Virtual Appliance. It is a *virtual machine based system which enables an execution environment in which users are given the opportunity to voluntarily share resources and run unmodified x86/Linux applications* [14]. Another interesting appliance is Cern VM [15] which is built using rBuilder [16] and provides a minimal Linux base to run LCG (LHC Computing Grid [17]) applications. CernVM is available as a raw system image file or as a VMware image.

Some related work on cloud computing [18, 19] uses virtual machine and virtual appliance as basic building blocks. Eucalyptus [20] is a cloud computing system that implements what is commonly referred to as *Infrastructure as a Service (IaaS)* [21]. It provides the possibility to run and control virtual machine instances which are deployed across a variety of physical resources and offers an entry-point into a cloud for users and administrators. A computing cloud is a pool of network enabled virtualized resources and services that can be dynamically reconfigured to adjust to a variable load, providing scalable, QoS guaranteed computing platforms on demand. There are multiple definitions for the term 'cloud computing', please refer to [18, 19] for more details. OpenNebula [22] decouples a server not only from the physical infrastructure but also from the physical location by enabling the dynamic deployment and re-placement of virtualized services within and across sites. It's a tool that can be used to manage clouds, even public clouds when combined with a cloud interface. The Xen Grid Engine [23, 24] is an extension of the Sun Grid Engine cluster management system based on the Xen Hypervisor. It supports for virtual machine management and offers increased usability and security features for cluster environments.

### 3 Cyberaide: A Light Weight Middleware for Production Grid

As introduced in the first section, there are a lot of scenarios where an advanced cyberinfrastructure is needed, but it might be difficult to use one. A possible solution to this dilemma is provided by Cyberaide. Several tools have been developed under the Cyberaide logo; well-known examples are Cyberaide toolkit and Cyberaide Shell. Consecutively Cyberaide toolkit's architecture will be shortly introduced and explained.

Cyberaide enjoys the following essential features:

- *Ease of use*: make the JavaScript based API and interfaces useful for Grid and Web developers.
- *Low installation footprint*: support fast downloads as well as an easy maintenance through a small manageable code base.
- *Security*: gain access to Grid resources in order to avoid compromising the system. This is especially important due to known limitations of JavaScript.
- *Basic Grid functionality*: is provided for developers to create Grid-based client applications.
- *Advanced functionality*: is offered as many developers do not want to replicate functionality provided by other Grid middleware and upperware.

The framework is designed in layers and comprised of different components. (see also Fig. 1). A web client that provides access to Grid functionality and components that can be deployed in a web server are provided. A service called “mediator service” mediates tasks to the Grid and basically is a secure server that provides most of the functionalities in regard to the Grid.

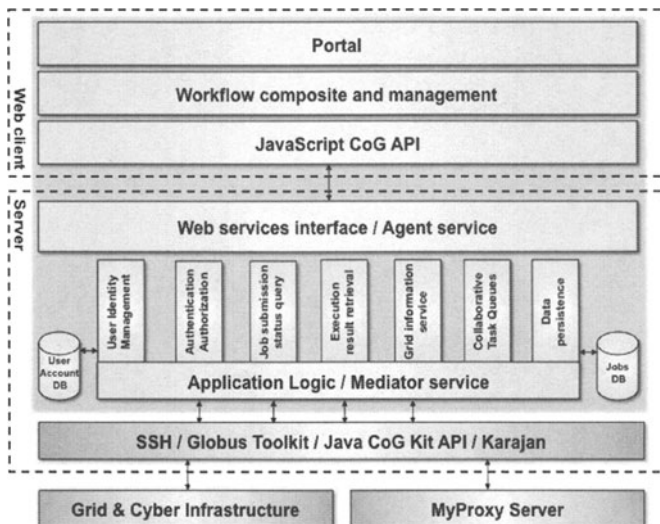


Fig. 1. System Architecture

- Web client: provides elementary functionality to access the Grid through a portal user interface.
- Server: contains two logical parts:
  - Agent service: is the intermediate service between Web client and mediator service; works as proxy for users to interact with the mediator service.

- Mediator Service: is the bridge between the Grid and the client library. The mediator service offers different functionalities and contains the application logic.

Because of the separation between the service and the client the development of Cyberaide shell was possible. this is a system shell that facilitates the use of cyberinfrastructures. It contains four high level design components: object management, cyberinfrastructure backends, command line interpreters, and services (see Fig. 2).

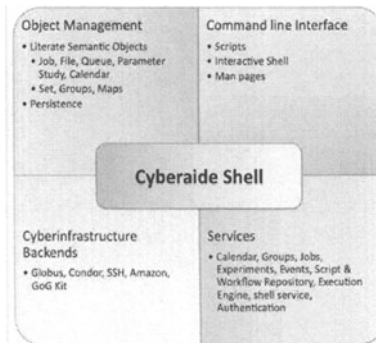


Fig. 2. High level design of Cyberaide Shell

#### 4 Cyberaide Virtual Appliance: On Demand Accessing Production Grids

To help users focus on their work and to also enable unexperienced users to work with Cyberaide, a virtual appliance for Cyberaide that configures itself for the most part might be a good solution. There are several tools available that facilitate the creation of a virtual appliance. Below, some information about two different tools usually used to create the Cyberaide virtual appliance is listed.

- VMware Studio [25] is a tool provided by VMware and is an appliance itself. It provides a Web Interface through which an appliance can be configured easily with the following steps:
  - Configure a “virtual hardware”, for example, memory, hard disk, network.
  - Choose an operating system (an ISO image has to be provided for the selected OS, by default only a few operating systems are supported)
  - Configure the software that should be installed on the virtual machine. Packages that are available on the installation medium (.iso) can be selected and will be installed automatically.
  - Set a target machine and directory. The appliance will then be copied to the specified location. VMware Studio is also able to automatically start the created appliance on a specified and correctly configured VMware

Server [10]. The created virtual machine consists of a vmx and a vmdk file that can be used with any hypervisor solution supporting these formats. (ESX, VMware Workstation, etc.)

- The second commonly used tool is *VMBuilder* [26]. VMbuilder basically is just a script that automates the process of creating a ready to use VM based on Ubuntu [27]. VMbuilder is part of JeOS 8.10 (Ubuntu: Intrepid Ibex). It offers the same abilities as VMware Studio does, except for the web interface used to configure the appliance. Nevertheless the configuration is still quite easy and done via some configuration files. In addition to that, there is no need for an ISO image on the local machine; VMbuilder automatically downloads all necessary files from the Internet. The operating system that will be installed (by default) is JeOS [28], a very efficient variant of the Ubuntu Server operating system, configured specifically for virtual appliances.

Since both tools show their own features, it is not easy to decide on one for building our Cyberaide virtual appliance. In order to make a decent choice we have evaluated and tested both tools. The following subsection outlines this evaluation process.

#### 4.1 Evaluation of Tools

The two possible solutions to create the Virtual Appliance as presented above are free and offer all needed functionality. To finally make up a decision the pros and cons of each tool have been evaluated. A listing of some evaluation criteria and the corresponding results are given in Table 1.

**Table 1.** Evaluation of virtual appliance creation tools

Criteria	VMwareStudio	JeOSVMBuilder
<i>ease of use</i>	very good, web-based interface	less comfortable, only CLI
<i>supported OSs</i>	Ubuntu, <sup>1</sup> SUSE, RedHat, CentOS	Ubuntu JeOS <sup>2</sup> only
<i>supported hypervisors</i>	VMware only	KVM, Xen and VMware
<i>auto-start on hypervisor</i>	supported	not supported
<i>encountered problems</i>	a lot <sup>3</sup>	no serious problems

<sup>1</sup> different distributions available.

<sup>2</sup> efficient variant of the Ubuntu Server operating system.

<sup>3</sup> see succeeding description.

Even though VMware Studio has some features ahead of JeOSVMBuilder there occurred a lot of problems while using it. For example: In theory an automated transfer of the created appliance to a target host, which is running VMware Server, is supported, but didn't work properly. Another drawback of VMware's solution is the package management. To directly install a certain package into the appliance it's necessary to also add in all dependencies of this package. This results in a tricky and less comfortable configuration.

Despite JeOSVMBuilder only supporting Ubuntu as operating system, its support for a large range of hypervisors is much more important. In addition, the applied operating system (Ubuntu JeOS [28]) is light and allows a very small virtual image file size and a good performance of the virtual machine.

On the other hand, JeOSVMBuilder doesn't provide functions to copy the created image on a target machine nor does it allow to start the created appliance on a hypervisor. However, this missing functionality can be added easily with a small script.

Overall, JeOSVMBuilder is better for our use case. The details about how adaptable this tool is can be seen in the following subsection.

## 4.2 Solution Description

After having evaluated the different possibilities we selected JeOSVMBuilder to create the Cyberaide virtual appliance. This command line tool requires just two basic parameters<sup>1</sup> to create a trivial virtual appliance. In the presented solution VMware is used as hypervisor and Ubuntu JeOS as operating system. All available parameters may either be passed to JeOSVMBuilder on the command line directly or by using some configuration files. The proposed solution uses four configuration files that control and set up the building process of the virtual appliance:

- A basic configuration file that allows to define some basic parameters such as: platform type (i386), amount of memory of the virtual appliance, packages that should be directly installed, etc.
- A hard-disk configuration file that defines the size of each available (virtual) hard-disk and the number and size of all the partitions that will be created on these hard-disks.
- Boot.sh: Shell script that will be executed during the first boot of the new appliance.
- Login.sh: Shell script that will be executed after the first logon in the new appliance.

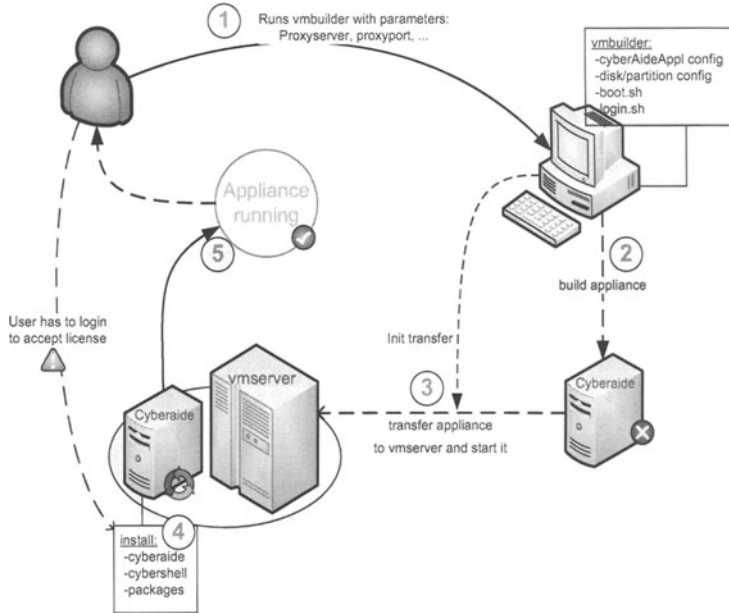
The essential part of the build process of the Cyberaide appliance is located in the two shell scripts. The boot.sh script sets some system environment parameters and installs and downloads required software. As some of the packages require user input (for example: user has to accept license of Sun's JDK) a completely unattended installation is not possible, hence those installations that require user input or rely on packages that need user input are started using the login.sh script.

Fig. 3 depicts the complete installation process:

- The user starts a script and passes some parameters such as proxy-host and proxy-port to it. This adapts the VMbuilder configuration files and starts the VMbuilderscript.

---

<sup>1</sup> 1.)target OS (always Ubuntu) and 2.)target hypervisor.



**Fig. 3.** Cyberaide Virtual Appliance: Build Process

- Vmbuilder then creates a virtual machine and installs some basic packages in it.
- The virtual machine files are moved to the VMserver and the appliance is started for the first time.
- Boot and Login scripts are executed. Because of some licensing issues the user has to login into the new appliance and accept the Sun license. Then the rest of the installation is completed and the appliance is completely created.
- The appliance is running and the services are ready to be used.

Besides the Boot and Login scripts, another two scripts were written, one for adapting the Vmbuilder configuration files and the other for transferring the appliance to the target host and starting it on the specified hypervisor. To ensure a secure communication between the host running Vmbuilder and the target host (and the hypervisor (host) (if different)) ssh is used. As the connections between the hosts mentioned above have to be established in an automated manner it's preferable to use certificates instead of passwords for ssh.

### 4.3 Scenario

A possible scenario where an advanced cyberinfrastructure may be of use is a grid and additional tools simplifying the use of it. To set up and to configure a grid middleware is not an easy task, so additional tools that provide an abstraction layer may be introduced to make it possible to use the grid services through a web interface, for example. One tool that provides such functionality is Cyberaide.



As Cyberaide itself relies on some other tools its configuration is not an easy task either. A (scientific-)user may find it too complicated or may just not be able to install all necessary tools. This is where the auto-install and deployment process of Cyberaide, exactly what we do, comes into play.

This scenario assumes the existence of a grid (e.g. TeraGrid) and a user that wants to run some jobs on this grid. The user may be a physicist and not familiar with all details of the grid and how to configure his tools to use the grid. Because he prefers to focus on his work and not to loose a lot of time setting up his tools nor tinkering around with the grid, he decides to leave this time consuming step to someone else. The IT department informs him, that there is a toolkit available named Cyberaide and that it can be installed by just running a deployment and installation script available on a machine running in the computing center. The user, keen to try this tool (and leaving the work to it) starts the script from his local machine and continues working. In the meantime a virtual appliance is created and set up. After the process has been finished the user receives a message that contains a description how to use the Cyberaide appliance and how to configure it. This last configuration step is quite easy and should not take a lot of time. As illustrated in this scenario the use of Cyberaide toolkit and the automated installation and deployment process saves a lot of time and helps users focus on their actual work.

## 5 Performance Evaluation and Discussion

### 5.1 Test of Cyberaide Virtual Appliance

The motivation was to provide a solution that facilitates the creation and the deployment of the Cyberaide toolkit. The proposed solution provides a very simple interface (see Fig. 4) that allows the creation of the appliance in an easy way. It also includes an efficient way to transfer the appliance to a target system from where it can automatically be started on a hypervisor.

There still are some inconveniences such as the need to log in during the installation phase to accept the license or the small effort it takes to set up the secured communications between the hosts. But finally the installation and deployment of Cyberaide is really easy and comfortable for an user. Some facts about the creation process are given in table 2.

The time to generate the appliance as well as the time needed to transfer it to the computer where the hypervisor is running strongly depends the network (and/or Internet) speed of the concerned systems. The numbers related to the building time (as given in table 2) have been measured by using a system which has a very fast internet connection. They may be extremely different on other systems or in another network environment.

Once the basic appliance (OS & basic packages only) is generated and transferred to a hypervisor, the second part of the installation is performed by the appliance itself. It downloads and installs all necessary packages which takes more time than the first part of the installation does.

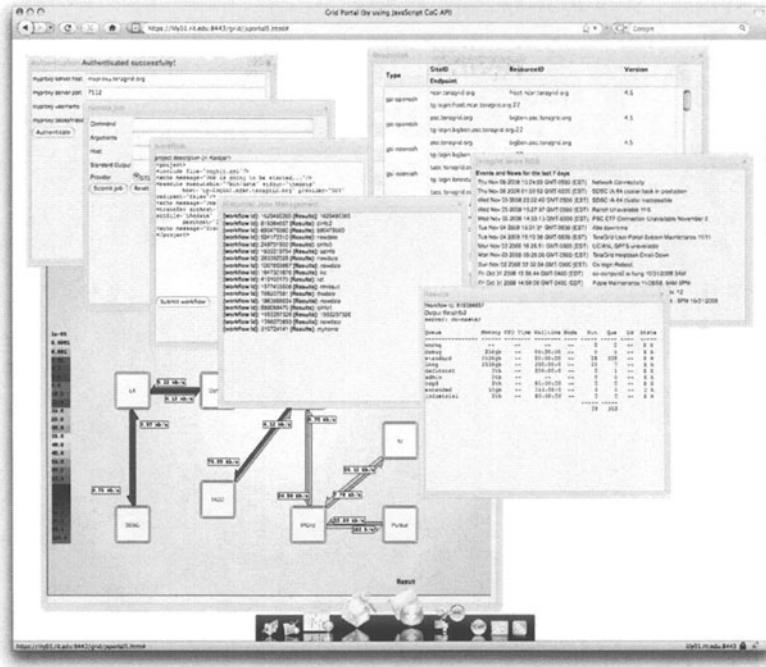


Fig. 4. Cyberaide Web Interface

The last installation steps are performed after the first login and basically install Cyberaide and all related components.

Finally the user has just to set up his certificates and keyfiles. The complete procedure takes in total about one hour (depending on the hardware) and needs just 2 user interactions (first login: accept license and set up certificates user login).

## 5.2 Test of Cyberaide Virtual Appliance on TeraGrid

To demonstrate that the appliance works, the LINPACK 1000x1000 benchmark program in single precision (1000s) and in double precision (1000d) have been tested on one node on TeraGrid. The benchmark is available at <http://www.netlib.org/benchmark/>.

Please note that this is just an exemplary program and the purpose of this test is not to measure the performance of the grid or any part of it.

The test has been performed using Cybershell which is included in the appliance. First step is to retrieve credentials. This can be done in Cybershell with the myproxylogon command: `security myproxylogon -u USERNAME`. Cybershell also provides an easy way to list available nodes (`execution listnodes`) and running jobs (`execution listjobs`). To finally submit a job the following command can be used: `execution s -id 12 -cmd myscript.sh`.

**Table 2.** Some key numbers of the creation process

building time (OS & basic packages only)	ca. 10 minutes <sup>1</sup>
building time (until first login)	ca. 20 minutes <sup>2</sup>
installation & deployment time after first login	ca. 15 minutes <sup>3</sup>
total time	ca. 45 minutes to one hour
virtual image file size (OS & basic packages only)	ca. 400 MByte <sup>4</sup>
virtual image file size (total)	ca. 2,8 GByte

<sup>1</sup> includes download time of OS & packages (without local repository).

<sup>2</sup> installation continues after login for license reasons.

<sup>3</sup> remaining packages will be downloaded and deployed after login.

<sup>4</sup> less than 200 MByte (ca. 190 MByte) zipped size.

#### Technical Info:

Host machine: Core i7 3,2 GHz, 6 GByte DDR3, fast Internet ( 2MB/s)

VMbuilder: Ubuntu 8.10 on VMware Server 2 (1 Core, 384MB assigned)

The submitted job is then scheduled and run on one (or more) of the execution nodes for example the node tg-c254. On this particular node the unoptimized 1000d benchmark reported about 66 MFlops. A more optimized version (-O2) reports 133 MFlops. As mentioned above this should not be considered as a real performance measurement of the grid, but only demonstrates that principally Cyberaide Virtual Appliance works.

## 6 Conclusion and Future Work

The Cyberaide Virtual Appliance provides a lot of functionality and in the same time is quite easy to use even for inexperienced users. The web interface as well as the secure shell (ssh) access to the Virtual Appliance provide a comfortable way to interact with it.

Cyberaide Virtual Appliance improves the Cyberaide Toolkit with a simple user interface and is a step forward that provides a faster, more reliable and easier installation, setup and configuration phase which finally results in a more productive working environment.

However, there are still some problems with the prototypical implementation:

- Large amount of data has to be downloaded during the installation. This may cause high cost in data transfer and long creation time depending on the Internet access speed.
- Uncomfortable start of building process.
- DHCP server has to be available and leasing addresses to Virtual Appliances.

Future work on Cyberaide Virtual Appliance may attack these problems. Possible solutions might be:

- To reduce the amount of data that has to be downloaded a *local repository* for software packages and the JeOS image could be created. This will accelerate the creation process of the Virtual Appliance as well as reduce costs

by avoiding transfers from the Internet. Another point may be that in the future some of the necessary packages are no longer available in the Internet repositories due to updates, etc... A local repository guarantees the presence of all necessary packages and versions.

- At the moment the installation process is started via the console on a local machine. It might be even more comfortable to have a web interface where a user can request the Cyberaide Virtual Appliance to be installed on a certain machine and to be started on a specified hypervisor. Once the machine is up and running all information the user may need to proceed (e.g. IP address of Virtual Appliance) could be displayed directly in the browser.
- By default, the Virtual Appliances tries to use the bridged networking interface of the hypervisor it's running on and expects to receive a DHCP lease while booting. Hence a DHCP server and a virtual bridged networking adapter has to be provided. In a future version NAT support might be included.

In summary, we use the virtual appliance approach to automate the process of installing the middleware for accessing cyberinfrastructures. The initial product is available for users. Problems mentioned above will be solved in the near future.

## Acknowledgment

Work conducted by Gregor von Laszewski and Lizhe Wang is supported (in part) by NSF CMMI 0540076 and NSF SDCI NMI 0721656.

## References

1. Herrod, S.A.: Future of virtualization technology
2. Sapuntzakis, C.P., Lam, M.S.: Virtual appliances in the collective: A road to hassle-free computing. In: Jones, M.B. (ed.) HotOS, pp. 55–60. USENIX (2003)
3. von Laszewski, G., Foster, I.T., Gawor, J., Lane, P.: A java commodity grid kit. *Concurrency and Computation: Practice and Experience* 13(8-9), 645–662 (2001)
4. Teragrid portal, <http://www.teragrid.org/userinfo/portal.php>
5. The globus toolkit, <http://www.globus.org>
6. von Laszewski, G., Wang, F., Younge, A., He, X., Guo, Z., Pierce, M.: Cyberaide javascript: A javascript commodity grid kit, pp. 1–10 (2008)
7. Vm history, <http://www.vm.ibm.com/history/>
8. Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T.L., Ho, A., Neugebauer, R., Pratt, I., Warfield, A.: Xen and the art of virtualization. In: Scott, M.L., Peterson, L.L. (eds.) SOSP, pp. 164–177. ACM, New York (2003)
9. Whitaker, A., Shaw, M., Gribble, S.D.: Denali: Lightweight virtual machines for distributed and networked applications. In: Proceedings of the USENIX Annual Technical Conference (2002)
10. Vmware server, <http://www.vmware.com/products/server/>
11. Kvm - kernel based virtual machine, [http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page)
12. Microsoft virtual pc, <http://www.microsoft.com/windows/virtual-pc/default.aspx>

13. Sapuntzakis, C.P., Brumley, D., Chandra, R., Zeldovich, N., Chow, J., Lam, M.S., Rosenblum, M.: Virtual appliances for deploying and maintaining software. In: LISA, pp. 181–194. USENIX (2003)
14. Wolinsky, D.I., Figueiredo, R.J.: Simplifying resource sharing in voluntary grid computing with the grid appliance. In: IPDPS, pp. 1–8. IEEE, Los Alamitos (2008)
15. Cernvm, <http://cernvm.cern.ch/cernvm/>
16. rbuilder, <http://www.rpath.com/rbuilder/>
17. Lcg, <http://lcg.web.cern.ch/LCG/>
18. Wang, L., Tao, J., Kunze, M., Castellanos, A.C., Kramer, D., Karl, W.: Scientific cloud computing: Early definition and experience. In: HPCC, pp. 825–830. IEEE, Los Alamitos (2008)
19. Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M.: A break in the clouds: towards a cloud definition. SIGCOMM Comput. Commun. Rev. 39(1), 50–55 (2009)
20. Eucalyptus, <http://www.eucalyptus.com/>
21. Nurmi, D., Wolski, R., Grzegorzczak, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The eucalyptus open-source cloud-computing system. In: Proceedings of Cloud Computing and Its Applications (October 2008)
22. Opennebula, <http://www.opennebula.org/doku.php?id=start>
23. Xge - xen grid engine, <http://mage.uni-marburg.de/trac/xge>
24. Fallenbeck, N., Picht, H.J., Smith, M., Freisleben, B.: Xen and the art of cluster scheduling. In: First International Workshop on Virtualization Technology in Distributed Computing, VTDC 2006, p. 4 (2006)
25. Vmware studio,  
<http://www.vmware.com/support/developer/studio/index.html>
26. Jeosvmbuilder, <https://help.ubuntu.com/community/JeOSVMBuilder>
27. Ubuntu, <http://www.ubuntu.com/>
28. Jeos, <http://www.ubuntu.com/products/whatisubuntu/serveredition/jeos>.
29. Kleinrock, L.: Ucla to be 1st station in nationwide computer network (July 1969), <http://www.lk.cs.ucla.edu/LK/Bib/REPORT/press.html>
30. von Laszewski, G., Younge, A., He, X., Wang, F.: Cyberaide shell: Interactive task management for grids and cyberinfrastructure,  
<http://cyberaide.googlecode.com/svn/trunk/papers/08-gridshell/vonLaszewski-08-gridshell.pdf>, <mailto:laszewski@gmail.com>
31. Foster, I.: What is the grid? a three point checklist (June 2002)
32. Oasis soa reference model,  
[http://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=soa-rm](http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm)