Cyberaide Creative: On-Demand Cyberinfrastructure Provision in Clouds

Casey Rathbone, Lizhe Wang, Gregor von Laszewski, Fugang Wang
Service Oriented Cyberinfrastruture Lab, Rochester Institute of Technology
102 Lomb Memorial Drive, Rochester, NY 14623-5608
Email: {casey.rathbone, Lizhe.Wang, laszewski, kevinwangfg}@gmail.com

Abstract

As demand for grid and cloud computing solutions increases, the need for user oriented software to provide access to theses resources also increases. Until recently the use of computing resources was limited to those with exceptional knowledge of the system design and configuration. With the advent of grid middleware projects this started to change allowing for new users not familiar with the complex grid infrastructure and client software to use the systems for their own research. The Cyberaide Gridshell demonstrated this by developing a user friendly interface to submit jobs to a grid. Following this theme it is our objective to create a tool that will take another step further by abstracting the creation and configuration of the infrastructure and system software away from the end-user. This will be achieved through the use of cloud resources provided by VMware virtualization and deployment via a web interface. We will show the benefits of deploying cyberinfrastructures, like clusters and grids, on a cloud design by demonstrating the ease of cyberinfrastructure deployment and the versatility of the systems that can be spawned on demand.

1 Introduction

Traditionally a cyberinfrastructure, e.g., a computational Grid, has been a physical system consisting of routers and switches to provide a transport layer for all network communications between a network of common systems. The systems were designed specifically as a set of homogeneous machines that could all be easily maintained and updated so that collective computations would be consistent throughout the cluster. Progressively as hardware has evolved more machines have been added to these initial networks making the entire grid infrastructure a mash-up of a variety of different systems creating issues for developers since specific machines are more capable of specific tasks. The complexity introduced by the variety of hardware and software running within a single grid has developed in a large concern that requires proficient network administrators for an entire range of possible systems. Researchers must also deal with configuring the middleware to use these machines and communicate through the assortment to complete their work. It is our goal of this paper to address this issue by employing a cloud with virtualization to create a common platform abstracted away from the hardware and software resources and then to deploy cyberinfrastructure on the cloud. This will allow for the end-user to specify the necessary resource requirements and have them immediately deployed without needing to understand the configuration and underlying infrastructure.

To demonstrate the capabilities of grid on cloud concept this project has developed a tool to be used within the Cyberaide software package that will be capable of deploying an individual workstation, a self-configuring cluster, and a Grid infrastructure on-demand.

The advantages of building cyberinfrastructure on cloud resources are the ability to easily construct specific computing resources, add and remove resources as demands change and create snapshots to preserve the current state of system progress. These features provide the researcher more flexibility with computing resources with a minimal overhead cost.

2 Background

In order to more accurately understand the information presented in this paper it is important to have a firm grasp of the technologies used to create traditional grids and virtualized hardware. Specifically information about grid middleware and virtualization terminology and software is provided to improve clarity.

2.1 Cloud Computing

At the current stage, the Cloud computing is still evolving and there exists no widely accepted definition. Based on our experience, we propose an early definition of Cloud computing as follows [**?**]:

> *A computing Cloud is a set of network enabled services, providing scalable, QoS guaranteed, normally personalized, inexpensive computing infrastructures on demand, which could be accessed in a simple and pervasive way.*

Cloud computing provides various non-trivial functionalities, such as, Infrastructure as a Service, Software as a Service, Data as as Service. The Cloud computing distinguishes itself from other computing paradigms, like Grid computing, Global computing, Internet Computing in the following aspects:

User-centric interfaces Cloud services should be accessed with simple and pervasive methods. In fact, the Cloud computing adopts the concept of Utility computing. In other words, users obtain and employ computing platforms in computing Clouds as easily as they access a traditional public utility (such as electricity, water, natural gas, or telephone network).

On-demand service provisioning Computing Clouds provide resources and services for users on demand. Users can customize and personalize their computing environments later on, for example, software installation, network configuration, as users usually own administrative privileges.

QoS guaranteed offer The computing environments provided by computing Clouds can guarantee QoS for users, e.g., hardware performance like CPU speed, I/O bandwidth and memory size. The computing Cloud renders QoS in general by processing Service Level Agreement (SLA) with users – a negotiation on the levels of availability, serviceability, performance, operation, or other attributes of the service like billing and even penalties in the case of violation of the SLA.

Autonomous System The computing Cloud is an autonomous system and it is managed transparently to users. Hardware, software and data inside clouds can be automatically reconfigured, orchestrated and consolidated to present a single platform image, finally rendered to users.

Scalability and flexibility The scalability and flexibility are the most important features that drive the emergence of the Cloud computing. Cloud services and computing platforms offered by computing Clouds could be scaled across various concerns, such as geographical locations, hardware performance, software configurations. The computing platforms should be flexible to adapt to various requirements of a potentially large number of users.

2.2 Cyberinfrastructure

Cyberinfrastructure was coined in 2003 by the National Science Foundation [?] to describe the resources, tools and related services for science and engineering research. It includes everything from supercomputers, high-capacity storage systems, and software suites developed to used in conjunction with these resources such as grid middleware. This project continues working towards the objectives of the Office of Cyberinfrastructure with the development of a tool that can be used to provide resources to the scientist while hiding the "complexities and heterogeneity of contemporary cyberinfrastructure." [?]

2.3 Virtualization

One of the key enabled technologies of Cloud computing is the virtualization technology. The benefits of virtualization within the computing industry have been numerous. The ability provided to monitor all hardware access, easily develop software for a platform that has been abstracted from the hardware, and the ability to stop the virtual machine in a specific state and move it another resource and restart it in the same state as before are a few of the key benefits. Within this project we plan to utilize all three in the on demand datacenter with the virtualization software providing a foundation for us to develop an overlay network of computing resources intended to be used within a grid. To make this possible we are using VMware's ESXi software package. The VMware ESXi package is a free license version of ESX virtualization software. It provides a virtualization platform that allows for a network administrators to efficiently manage and deliver a multitude of operating systems. In this project we will be making the creation and networking of the virtual machines automated by the scheduler receiving request from the web service. Specifically ESXi was chosen because it provides a free full system virtualization platform with an available software development kit and a hardware compatibility list supported by a commercial company regularly making updates.

2.4 Grid Middleware

Grid middleware is defined as the glue software the binds together heterogeneous systems to allow for resource sharing within virtual organizations. Within this project we utilize the Condor Project [?] to bring together the virtual machines created on the ESXi server to be the grid middleware. Condor was chosen because of its relative ease of deployment and the ability to easily create clusters of the Condor clusters into flocks for a greater use of resource sharing. The grid infrastructure provided by Condor is going to utilized later as the interface for the grid scheduler while working in coordination with the ESXi server for virtual machine instantiation. This can be modified based upon the requirements of the scientist using the system but for our demonstration purposes Condor was adequate.

3 Related Research

In order to provide better clarity for our project we researched several related projects to determine where shortcomings and benefits had be achieved.

This Grid Appliance [?] provides an example of exactly the small, easily distributable virtual machines that will be most efficiently utilized within our cloud environment. The Grid Appliance is a self-configuring virtual machine that has Condor middleware already installed allowing the machine to immediately come online into a Condor grid to provide resources. Our project will similarly have preexisting virtual machine templates containing grid middleware installed along with any software specific for the completion of the job specified via the web service. This small footprint will provide an excellent platform to allow the resource scheduler to manage and mitigate virtual machines as it sees fit to optimize throughput.

The Nimbus [?] project provides the ability to convert an existing cluster into an IaaS (Infrastructure-as-a-Service) cloud to share the resources externally. It utilizes the virtualization of Xen hypervisor to provide a virtual platform for clients to setup and lease remote resources. This project develops the cloud but is currently limited by the utilization of the resources provided. Scientist using the Nimbus will also have to be familiar with the Globus Toolkit [?] and Globus Workspace Service. Cyberaide Creative creates a virtual cluster on top of a cloud provided by VMware ESXi server that doesn't require an existing cluster infrastructure. This allows the researcher more flexibility the number of actual machines and network hardware.

Eucalyptus (Elastic Utility Computing Architecture for Linking Your Programs To Useful Systems) [?] project has been developed as an open-source software infrastructure aimed at providing an web service based cloud computing environment. It was designed as a development platform for other clouds such as the Amazon Elastic Cloud Compute (EC2) [?]. The cloud developed by Eucalpytus on Amazon's EC2 creates a cluster of virtual machines without any initial middleware pre-configured leaving this task up to the researcher. We realize that the installation and configuration of grid middleware is a daunting task even for an experienced system administrator and therefore integrate the creation of virtual machines with the configuration of the middleware software package making the cluster more accessible to novice users.

Above solutions can provide virtual machine resources from Clouds on demand. Compared with them, our unique contribution lies in that we propose a new cloud computing paradigm: on demand provision of cyberinfrastructure on Clouds and on-demand access the provided cyberinfrastructure. We design and implement an lightweight middleware, Cyberaide creative, a part of Cyberaide project [?], which can combine the functionalities of cyberinfrastructure creation, configuration, management and access in an integrated solution.

4 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.

- 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.

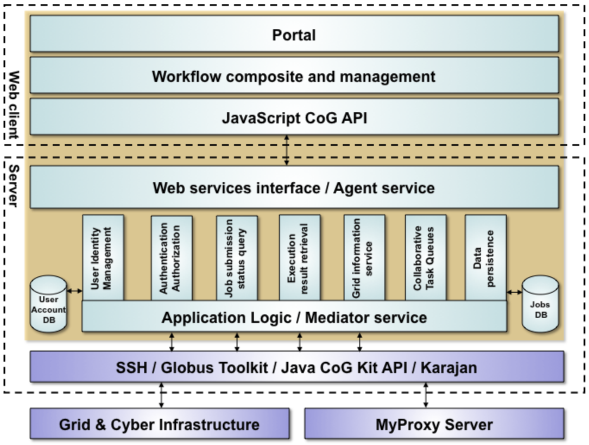
Web client
Portal
Workflow composite and management
JavaScript CoG API
Server
Web services interface / Agent service
User Identity Management
Authentication Authorization
Job submission status query
Execution result retrieval
Grid information service
Collaborative Task Queues
Data persistence
User Account DB
Application Logic / Mediator service
Jobs DB
SSH / Globus Toolkit / Java CoG Kit API / Karajan
Grid & Cyber Infrastructure
MyProxy Server

Figure 1. System Architecture

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).

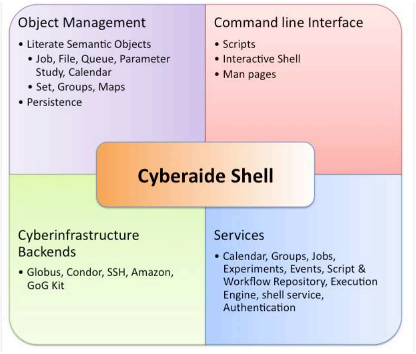
Object Management
• Literate Semantic Objects
• Job, File, Queue, Parameter Study, Calendar
• Set, Groups, Maps
• Persistence
Command line Interface
• Scripts
• Interactive Shell
• Man pages
Cyberaide Shell
Cyberinfrastructure Backends
• Globus, Condor, SSH, Amazon, GoG Kit
Services
• Calendar, Groups, Jobs, Experiments, Events, Script & Workflow Repository, Execution Engine, shell service, Authentication

Figure 2. High level design of Cyberaide Shell

5 Cyberaide Creative Paradigm

The paradigm presented using Cyberaide Creative is an on-demand resource allocation system. This creates an environment where resources can be optimally utilized as demands request them. The Cyberaide Creative system is used as a tool for acquisition of the production grid running either Condor or Globus ToolKit and Cyberaide Gridshell is the operating interface to the grid. The benefits of this paradigm are the ability to outsource resources for less cost than to maintain a complete internal system for peak resource consumption.

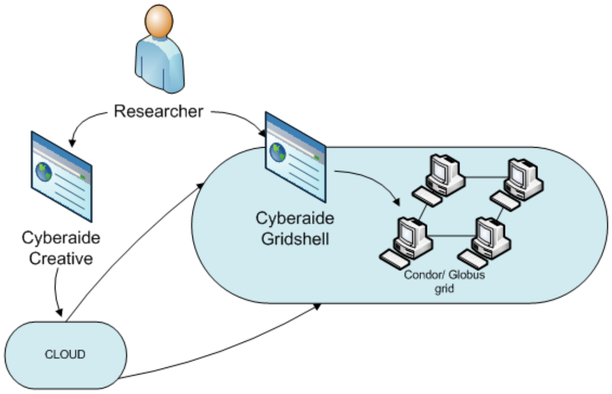

Figure 3. Cyberaide Creative Paradigm

Figure 3 shows the computing paradigm of Cyberaide Creative:

1. Users send requirement to Cyberaide creative to demand cyberinfrastructures from Clouds, for example, a condor cluster, or a computational Grid with Globus Toolkit as a middleware.

2. Cyberaide creative then construct a cyberinfrastructure for users, which is pre-installed some Grid middleware, like condor, Globus and Cyberaide shell.

3. Users then on demand access the cyberinfrastructure with aides of cyberaide shell.

6 Cyberaide Creative Use Case

As previously described the entire virtual data center will be developed on the VMware ESX 3i package as a platform for allocation of resources. The virtual platform will be utilized as a resource pool for dynamic creation of virtual machines for the on demand creation of grid networks to be used for specific jobs submitted to the web service.

6.1 Request Cyberaide Shell Access Interface Use Case

Working within the Cyberaide Framework, Cyberaide Creative will be capable of launching a virtual machine preconfigured to the client's specifications from a web interface to provide access to a pre-existing cluster.

1. The scientist has a pre-existing computing cluster and requires a simple job submission interface. The scientist proceeds to connect to the on-demand web interface to specify the location and access configurations to the cluster.

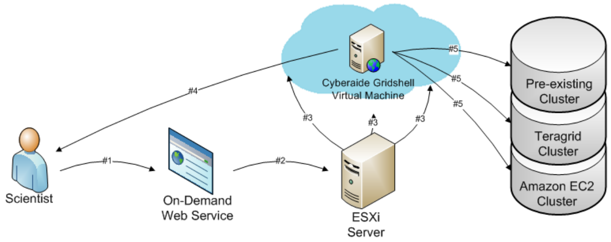

Figure 4. Gridshell creation use case

2. The web service obtains the necessary configurations from the scientist and then requests a Cyberaide Gridshell virtual machine from the ESXi server.

3. The ESXi server creates a new Cyberaide Gridshell virtual machine and passes along the configurations.

4. The Cyberaide Gridshell virtual machine allows for remote connections to be made from the scientist to handle job submissions.

5. The Cyberaide Gridshell handles all coordination to the various computing clusters simplifying the end-user interface .

6.2 Request Single Virtual Machine Workstation

An additional use case demonstrates a simple single user workstation virtual machine.

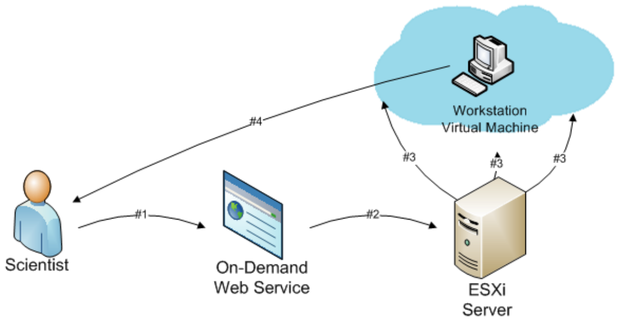

Figure 5. Workstation creation use case

1. The scientist requires a new workstation for a specific task that would burden the resources of the local machine. Therefore the scientist logs onto the on-demand web interface and selects the specifications of the workstation required.

2. The web service uses these specifications to instantiate a workstation on the ESXi server.

3. The ESXi server launches the virtual machine image returning the IP address and login information to the web service.

4. The web service identifies the location of the newly created workstation to the scientist, allowing the scientist to directly login into the new virtual machine.

6.3 Request Cluster Use Case

Another common functionality for the on-demand system will be to generate a cluster based on the needs of the scientist. The following use case demonstrates the regular operation of the entire system.

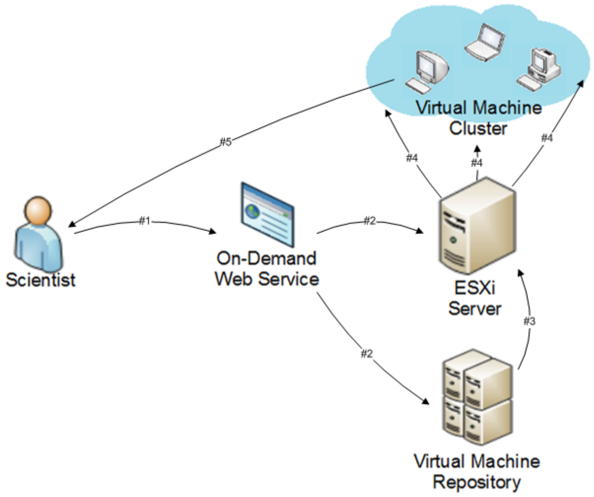

Figure 6. Cluster creation use case

1. The scientist recognizes that the project requires a significant amount of resources to complete in a timely manner. The scientist proceeds to log onto the on-demand web interface and specify the job and required resources.

2. The web service uses these job requirements to determine the allocation of available resources. Then contacting the ESXi server and virtual machine repository.

3. The ESXi server obtains the appropriate virtual machine image.

4. The ESXi server then instantiates the cluster on the cloud identifying a cluster controller for the scientist to interface with.

5. The host controller will be created with a Cyberaide Gridshell and will be exposed directly to the scientist.

7 Cyberaide Creative Design and Implementation

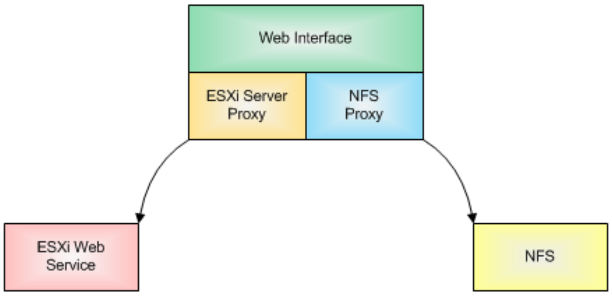

Figure 7. System components

To develop this tool to create on-demand cyberinfrastructure we used several technologies to increase accessibility and flexibility of the system. The three main components of the Cyberaide Creative project are the web interface, VMware ESXi server and the virtual machine repository.

7.1 Design

The web interface is used to provide easy access to deploying resources on the cloud. The web interface is currently developed using web service technology to allow for simple, interoperable platform. The ability to connect different systems to web service makes creating user interfaces and integrated into the Cyberaide software package an easy job for any administrator. The backend of the web service interface is Virtual Infrastructure Java API [?] used to connect to the VMware web service via HTTP protocol that is capable of passing through firewalls and other security measures with minimal modifications. The only configuration required for the web interface to connect to the ESXi server is to specify the server's URL, administrator username and password. This allows for easy re-configuration of the web interface if new ESXi server becomes available.

The next system component is the VMware ESXi server that is used to provide the cloud resources. It provides a scalable platform to create cyberinfrastructure. Currently our installation of ESXi server handles storage of virtual machine images along with the administration controls over the virtual machine instantiations. It is possible to link several of the ESXi servers together to create a larger pool of resources to be used with Cyberaide Creative. The ESXi server handles all of the downloading and configuring of the virtual machine instantiation via commands issued by the web interface.

The communication between the web service and ESXi server is maintained by a Common Information Model (CIM) [?]. CIM is a standard developed by Distributed Management Task Force(DMTF) to provide a method to define managed elements within the IT environment and set relationships between the objects. CIM is a combination of specifications and schema that make the integration of components from different providers easier. The schema defines the objects in a common XML based format much like WSDL documents for web services and the specifications defines the usage of these objects within the communication. We have included this type of protocol between the web service interface and the ESXi server because it will allow us to easily change the backend server to another solution such as Xen or VirtualBox.

The final component within the Cyberaide Creative system is the NFS virtual machine repository. This is optional within our current configuration but an absolute requirement as system demand increases. The purpose of the virtual machine repository is to provide a database that will store virtual machine images that can be updated and deployed by the researcher. Being an NFS disk attached to the ESXi server allows the storage size to change and move as updates are done to the entire system.

The virtual machine repository has been developed to store and manage the open virtual format (OVF) [?]. The OVF is a open standard within the industry that was developed to provide a common configuration that can be accepted by a variety of hypervisors. This allows the repository to be used with any assortment of virtual platform solutions such as Xen, VMware, and VirtualBox. This contributes the usability for scientists by relieving any restraints on specific platforms and makes future upgrades feasible.

The flexibility within the system is maintained with the distributed nature of the web service interface, VMware ESXi server and the virtual machine image repository. By separating these three components it allows for a highly upgradeable and adaptable system capable of being run a variety of configurations regardless of the component network domain, geographical location, or infrastructure.

7.2 Implementation

The Cyberaide Creative implementation uses Java framework to allow for platform operability and simple integration into the existing Cyberaide software package. This also provides a wide variety of developed packaged to interface to the ESXi server and the NFS. The web service is currently developed as a weblet within Apache Tomcat server and incorporates both the Virtual Infrastructure (VI) API and Yet Another Network Filesystem (YANFS) to provide connectivity to the other components within the system. Currently the entire tool has been developed within the Netbeans IDE to minimize web service configuration and integration to an HTTP server but in future work we will use an embedded HTTP server such as Jetty.

The ESXi server has a web service interface with default installation that allows for the VI API easily connect to and issue commands such has power on, power off, create virtual machine and delete virtual machine. The only change made to the default installation of the ESXi server was to include access to the virtual machine repository. The YANFS package is used to view, upload, and delete virtual machine images within the virtual machine repository. The web service combined with these two packages allows the Cyberaide Creative to create an interface that provides all of the features a user would need inorder to setup, configure and maintain a virtual grid infrastructure.

The virtual machine repository is implemented as a NFS server that has been deployed on a default Linux Ubuntu installation with a local area network connection to the ESXi server. We realize that to utilize this as a high performance computing system it is necessary to upgrade this connection in the future to increase system performance.

The GridAppliance was used as our virtual machine demonstration. It was capable of creating a private pool of nodes via Condor middleware. With Condor properly configured on the virtual cluster we then configured a Cyberaide Shell for access returned the login information to the Cyberaide Shell.

8 Cyberaide Creative Performance Evaluation and Discussion

To demonstrate the capabilities provided by Cyberaide Creative we have run some analysis of the performance of the system. These performance aspects include the overhead created by virtualization and the overhead introduced instantiating virtual machines via web services.

8.1 Virtualization Overhead

One of the primary concerns for clouding computing within the high performance computing industry is the amount of overhead created by the hypervisor abstracting the hardware away from the software. We realize this and felt it is necessary to demonstrate with a comparison of two systems the amount of overhead lost.

The system utilized to determine the impact of the overhead created was the same machine used throughout development of the Cyberaide Creative. The specifications of the system are 2x 2 GHz Dual Core AMD processors with 8 gigabytes of RAM and 4 250 gigabyte IDE hard drives.

To complete the analysis of the overhead introduced by the ESXi server we utilized the Pelican HPC [**?**] to load the Debian based system via a USB drive and launch virtual clusters. The USB drive was configured to load all the necessary operating system components into the RAM without modifying any of the existing hard drive data, this allowed us to quickly test the virtualization overhead introduced. The virtual clusters were created using a single node as the head node with an additional 2, 4 or 8 compute nodes respectively. The Pelican HPC is specifically designed for this type of deployment and network boot capability lends itself well to starting the compute nodes.

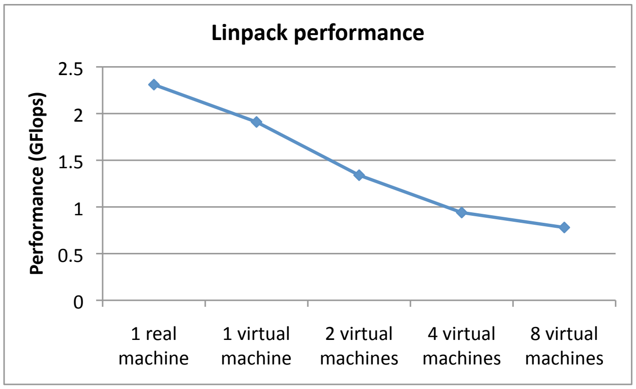

Figure 8. Linpack Results

The results show an expected decrease of approximately 20% decrease in performance from the same host machine running the Pelican HPC natively. This can be attributed to the ESXi server needing to properly emulate the underlying hardware and provide features such as snapshot and migration. Additionally it is obvious from the chart to see that with the creation of a larger cluster on our limited resources the management and message passing between compute clusters consumes a considerable portion of the resources without any added benefits to the system.

8.1.1 Web Service Implementation Overhead

To determine the web service's impact on virtual machine instantiation (power on and power off) we compared the time taken to start a virtual machine using our web service interface and a the ESXi server's local command line interface. To accurately measure the duration taken to instantiate a virtual machine from the web service we integrated a timer within the web service. The timer is first started when a command to start a virtual machine is made and then the timer is stopped upon the detection of the IP address, showing that the virtual machine is available for use. Inorder to detect the IP address of the virtual machine the ESXi server is continually polled requesting the information about the specified virtual machine. The virtual machine used for this comparison is the Grid Appliance virtual machine which would be a common virtual machine run to create a grid.

The graph presented shows the overhead produced by the web service interface in launching virtual machines. As you can see the web service adds a minimal increase the overall time necessary to start a virtual machine.

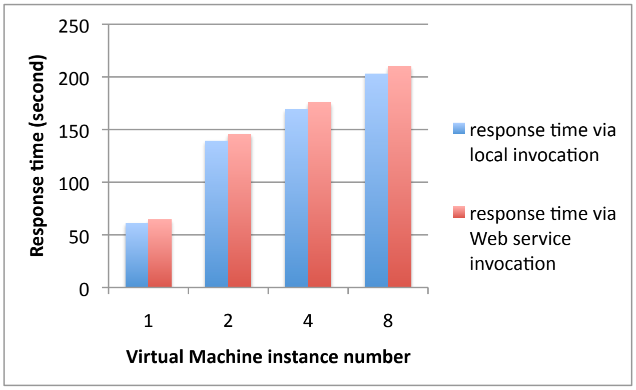

Figure 9. Web service overhead

9 Conclusion and Future Work

Through the use of current technologies, grid computing and virtualization, a new computing type has emerged as virtualized utility computing. The techniques presented within this paper have demonstrated the integration of these technologies to create an on demand cyberinfrastructure capable of providing resources for specific computing tasks. We feel the grid on cloud design has a great potential to serve as a platform for future developments for researchers and that this paper should only be seen as an introduction to its capabilities.

Acknowledgment

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