

Development of Cyber-Technology Information for Remotely Accessing Chemistry  
Instrumentation

by

Craig A. Richards

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master

in the

Computing and Information Systems

Program

YOUNGSTOWN STATE UNIVERSITY

March 2011

Development of Cyber-Technology Infrastructure for Remotely Accessing Chemistry  
Instrumentation

Craig A. Richards

I hereby release this thesis to the public. I understand that this thesis will be made available from the OhioLINK ETD Center and the Maag Library Circulation Desk for public access. I also authorize the University or other individuals to make copies of this thesis as needed for scholarly research.

Signature:

---

*Craig A. Richards*, Student

Date

Approvals:

---

*Dr. Graciela Perera*, Thesis Advisor

Date

---

*Dr. Allen D. Hunter*, Committee Member

Date

---

*Dr. John R. Sullins*, Committee Member

Date

---

Dr. Peter J. Kasvinsky, Dean of School of Graduate Studies and Research Date

Craig A. Richards

© 2011

## **Abstract**

There exists a wide variety of technologies which allow for remote desktop access, data transfer, encryption, and worldwide communication through the Internet. These technologies, while independently solving unique problems, can be combined into a project which would resolve all of the unique problems with one single system. Youngstown State University's Chemistry Department required a high reliability unified system to provide remote access, web cam feeds, user security, and encrypted file transfer for computer equipment operating scientific instrumentation. A suitable software project solution was developed at Youngstown State University in collaboration with Zethus Software through analysis of technological resources and project requirements, and a process of software development.

This thesis describes the cumulus::CyberLab project developed in order to resolve the above requirements. The cumulus::CyberLab project allows students, faculty, and scientists to remotely access millions of dollars of scientific equipment offered by our university from anywhere in the world. To best describe this project, this thesis outlines the overview of the project, work in the project, and how this project created unique software which is valuable to not only our university but also to other worldwide users.

## **Acknowledgements**

I would like to acknowledge Dr. Allen D. Hunter for being the Principal Investigator responsible for the initial creation of the cumulus::CyberLab project, Dr. Graciela Perera for her assistance as advisor during the thesis creation process, the Army Research Labs for funding the project, Zethus Software (founded by Eric Parker and Andrew Reinhardt) for their development of the cumulus::CyberLab project software as well as providing me with an internship, the Cushwa Family for providing scholarships for academic success and university enhancement, Dr. Matthias Zeller for his assistance during the cumulus::CyberLab project, and finally the various faculty members at Youngstown State University who provided me with a valuable education which made my participation in this project possible.

## Table of Contents

Abstract.....	iv
Acknowledgements.....	v
List of Figures.....	viii
Chapter 1: Introduction.....	1
1.1: Background.....	1
1.2: Motivation and Problem.....	4
Chapter 2: Foundation.....	7
2.1: Cyber Technology.....	7
2.2: Remote Access.....	11
2.2.1: Technical Background.....	11
2.2.2: Analysis of Prior YSU Research.....	12
2.2.3: Motivation of Remote Access.....	17
2.3: Cloud Computing.....	22
2.4: Case Study – X-Ray Diffraction Scientific Equipment.....	27
Chapter 3: Cyber Technology Software.....	31
3.1: Zethus Software Overview.....	31
3.2: History and Vision.....	34
3.3: Rational of cumulus::CyberLab.....	37
3.4: Development Principles and Phases.....	40
3.5: Lessons Learned.....	43
Chapter 4: Development and Evaluation of cumulus::CyberLab.....	45
4.1: System Administration.....	45
4.1.1: Utilized Hardware.....	45
4.1.2: Operating Systems and Installation.....	48
4.1.3: Deployment and Diagnostics.....	52
4.1.4: System Administration Summary.....	53
4.2: Testing and Documentation.....	53
4.2.1: Description of Testing.....	54
4.2.2: Overview of cumulus::CyberLab Client software.....	54
4.2.3: Testing Process Overview.....	57
4.2.4: Software Documentation System.....	60
4.2.5: Testing and Documentation Summary.....	61
Chapter 5: Conclusions.....	63

Works Cited ..... 65

## List of Figures

Figure 1 -- cumulus::CyberLab system overview.....	2
Figure 2 -- Network Types.....	26
Figure 3 -- Initial main screen of cumulus::CyberLab.....	55
Figure 4 -- eGroupware dialog box.....	57



## **Chapter 1: Introduction**

This chapter serves as an introduction to this thesis detailing a project, called the cumulus::CyberLab project, which has benefited our university by solving notable problems with remote access to scientific equipment. Background information, the problem and motivation behind this project, and an outline of future chapters will be covered to appropriately introduce this project. This chapter has fundamental information necessary to comprehend the chapters that follow and the entirety of the thesis.

### **1.1: Background**

Modern society has provided many advanced pieces of scientific instrumentation which make possible modern studies of various scientific questions. Such instrumentation is expensive to purchase and to operate. In the current economy such purchases and operations are being scrutinized more than ever to justify the costs required. Such scrutiny can have a negative impact on provisioning scientists, researchers, and teachers the necessary scientific equipment to educate future generations of students. The cumulus::CyberLab project was designed as a solution to this problem. The project provides a secure, simple, and non-intrusive remote access technology to allow for increased collaboration between educational institutions and scientific professionals to share scientific equipment and thus justify its' cost.

This recently developed software project was designed in collaboration with Zethus Software, a company currently within the Youngstown Business Incubator (YBI). Zethus Software, in collaboration with Youngstown State University (YSU), developed the cumulus::CyberLab software and continues to develop the software by providing

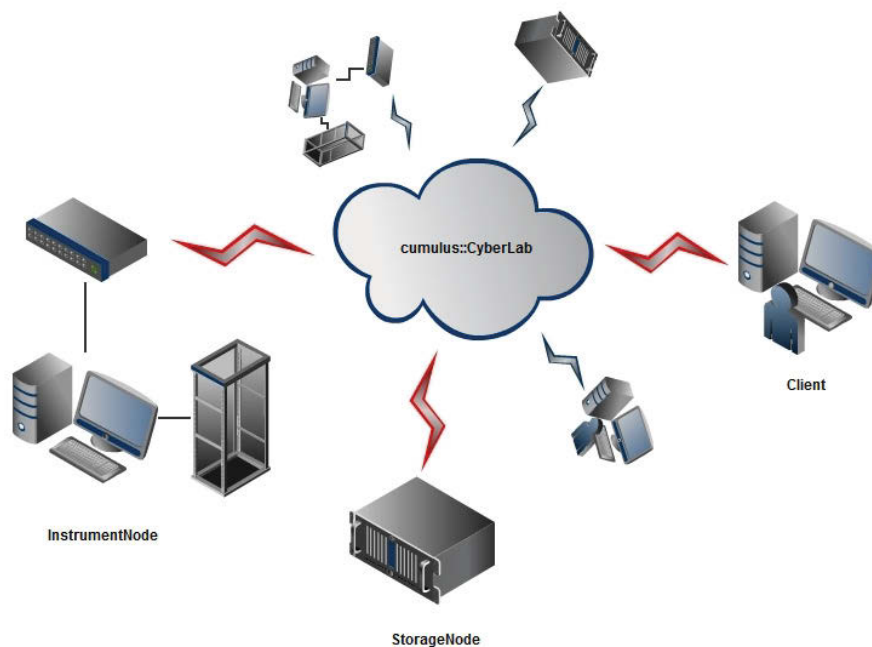


Figure 1 -- cumulus::CyberLab system overview

support, new features, and expanding the network of cumulus::CyberLab devices. This software provides many advantages to YSU including sharing of expensive scientific equipment with other institutions, increasing collaboration between educational institutions and scientific professionals, and providing an infrastructure for remote classroom teaching as well as training. The project is easy for both campus staff to deploy and maintain as well as clients, such as students, to use.

Figure 1 shows an overview of the devices and networks involved in a typical cumulus::CyberLab setup. The Client connects via the red network paths to the resources requested (in this case an InstrumentNode and a StorageNode) and the client does not have to worry about the network itself or the other participants in that network. The cumulus::CyberLab system provides a vital link between the scientific equipment, secure data storage archiving, and the user in a remote location. The equipment is attached to the cumulus::CyberLab InstrumentNode (one of three varieties of symbiotes) which

provides equipment remote access service through the Internet in a secure fashion. The equipment can store files securely on the StorageNode which can then be accessed by the end user. The remote location user is able to use this remote access service via a piece of software operating on the remote location computer which requires no installation and is launched from a web browser. The software allows for access to any other remote access services made available by other cumulus::CyberLab Symbiotes installed and participating in the collaborative network. The user is able to operate scientific equipment from a remote location and securely store scientific data using the cumulus::CyberLab system.

The cumulus::CyberLab InstrumentNode provides a remote access link to any scientific equipment that provides standard computer outputs (keyboard, video, and mouse) via a KVM over IP device built into the InstrumentNode. Installation of the InstrumentNode only requires connecting a few additional cables to the control computer. Any Symbiote deployed can easily be installed by connecting a few cables and powering on the device. This provides for a simple hardware installation of the InstrumentNode, SensorNode, and StorageNode Sybiotes.

The remote location user is able to connect to cumulus::CyberLab by running the client software through their browser. The client software utilizes the Java Runtime Environment (JRE) which must be installed on the client's computer (this is common and the JRE is free). The application is started when the user clicks on a website link to a Java WebStart definition file which initializes the program by downloading it and running it. No installation is required to run the client software and therefore the software can run from any computer with the appropriate Java runtime environment

installed. This provides for a simple operation of the client software allowing for remote access to scientific equipment.

The cumulus::CyberLab system currently provides remote access capabilities and secure data archiving to instrumentation in the X-Ray facility at Youngstown State University. The X-Ray facility has three instruments with purchase and upgrade costs of more than two million dollars that additionally require a constant operational budget.<sup>1</sup> Being housed in a climate controlled environment, these instruments require periodic maintenance and optimal operating conditions. Each instrument is controlled by a computer providing standard outputs (keyboard, video, and mouse) and lacking secure remote access capabilities as well as data archiving before the cumulus::CyberLab project. These instruments are vital for chemistry research and provide essential scientific data.

The motivation as to why this project was brought into existence as well as the problems encountered before the project's inception will be discussed in the next section.

## **1.2: Motivation and Problem**

Youngstown State University is a state university consisting of roughly 15,000 students fortunate to have various specialized scientific equipment on-site.<sup>2</sup> However, our university cannot possibly afford all types of specialized scientific equipment needed to carry out the large variety of chemistry experiments. Furthermore, our equipment is not 100% utilized due to the limited number of students and faculty in the Chemistry Department. In collaboration with various universities across the nation, sharing of resources, including scientific equipment, benefits the students at each institution. The

---

<sup>1</sup> (Youngstown State University Department of Chemistry)

<sup>2</sup> (The Vindicator)

problem is that students and professors at various universities cannot easily travel to access scientific equipment when such equipment is not available on-site.

The scientific equipment at YSU needs periodic monitoring and observation. This observation can be done on-site but this limits the staff involved in the observation from carrying out other duties and travel from the university without a skilled substitute. The proper observation and control of scientific instrumentation is required to prevent time and money wasted by analyzing samples that are not showing favorable results thus making further analysis worthless. Since the staff cannot remain onsite 24 hours a day due to payroll costs, there are situations where money is wasted analyzing worthless samples.

The equipment utilized to analyze those samples requires periodic maintenance and occasional troubleshooting. Such tasks are often done off-site by manufacturer technicians or specialists who can handle such tasks but are commonly not on-site. Such maintenance and repairs can be done remotely if a mechanism exists where they access the machines without traveling to physically access them. Sending those technicians or specialists to the on-site location costs a significant amount of money for each incident.

Those situations can be mitigated if a remote access solution is in place allowing for the sharing of these scientific instruments as well as the remote monitoring of these instruments as needed. The cumulus::CyberLab project provides such a remote access solution being secure enough to be utilized for both situations. While the project has allowed for specific problems to be resolved, the availability of this remote access solution also opens up new opportunities in the realms of remote classroom education,

training, and collaboration with industry. This allows for our university to help achieve its' mission statement goals.

The cumulus::CyberLab project involved the participation of the author over the past three years in assisting Zethus Software and Youngstown State University in adapting, testing, and developing the project for Youngstown State University's specific needs as well as providing features valuable to other institutions, industry, and other interested clients. Chapter 2 describes the foundation for this project and thesis, Chapter 3 describes cyber infrastructure software, Chapter 4 details and the development and evaluation of this project, and Chapter 5 concludes the thesis.

## **Chapter 2: Foundation**

To further clarify the background and motivation for this project, a detailed analysis of the various technologies involved as well as the foundation of each technology is required. This chapter will introduce each of these topics and it will clarify the motivation for this project. We will discuss the term cyber technology and how it applies to the goals of this project, the benefits and drawbacks of other current remote access technology along with the motivation for this project and the technical background, analysis of cloud computing in relation to this project, and a case study of specific scientific equipment receiving benefits from this project.

### **2.1: Cyber Technology**

Cyber Technology is the convergence of communication, control, and computation technologies in systems that allow users to transparently share and access resources. In the context of this thesis, the term cyber technology describes the overall goal to collaboratively share resources and academic expertise amongst a collective partnership of universities and industrial companies. To help explain how Cyber Technology relates to cumulus::CyberLab we will review the technologies and history of the Internet and how it relates to cumulus::CyberLab, the motivation and problems behind sharing scientific equipment, and the project that developed to resolve those motivations and problems.

One means of providing connectivity for systems in cyber technology is by using the Internet. The foundation of the Internet was ARPANET whose design was the foundation of the modern Internet. Some founding goals of the Internet was to have a wide array of computers speaking a common language (protocol) to each other and to

allow them to share various services.<sup>3</sup> Services shared at the onset of ARPANET included sharing files and sending email. The primary goal of such computer communication was to share recently developed software and research results in the Computer Sciences field.

Today, the Internet is comprised of many interconnected local area networks. These local area networks allow users to share among themselves resources or equipment (e.g., printers) by having computer communication protocols built for such purposes. A printer shared between many different computers allows more people to have printer access and requires that less people have a separate printer. Access to the printer can be restricted depending on how the network is set up and operated. In the end, the printer's usefulness is increased as more people can use it through the network and as a person's need for an individual printer is reduced. This allows for better use of purchased equipment.

Equipment has been expanded to include more than just printers as the demand for remotely accessing the capabilities of various computer accessible devices increased. Today scientific instrumentation is controlled by computers which have the capability of being remotely accessed and thus sharing the capabilities of the connected scientific instrumentation with the remotely connected party. Therefore, you do not have to be in the vicinity of the scientific instrumentation to use such instrumentation.

While the Internet grew in size and complexity over the years, this basic concept of sharing resources potentially across the whole network has remained intact. Therefore, it is possible to share a resource across the currently 1.9 billion Internet users worldwide,

---

<sup>3</sup> (Markoff)



if desired.<sup>4</sup> Identifying the resources to share, who should have access to those resources, and finally sharing those resources is all that is needed to make a single piece of equipment more useful than it was previously. Sharing resources amongst the academic community is one of the goals of a higher education institution.

To that end, our university has equipment which provides valuable services in the field of Chemistry. The equipment in the X-Ray facility analyzes the composition of a sample material to determine what elements and/or compounds are present. Therefore, a topic such as “Does this paint have lead in it?” can be determined quickly as long as the operator of the machine is a skilled scientist. More complicated inquiries about the compositions of various samples require increasing amounts of skill from the scientist running the experiment as well as increased experimentation time.<sup>5</sup> To that end, while this equipment is very useful in resolving scientific inquiry, it also requires operation by a skilled scientist in order to be useful. A solution is needed to allow for a scientist to remotely access such equipment so that experiments can always be operated by skilled personnel.

The solution toward the problems described previously are investigated in this project and it combines several off-the-shelf and open source software solutions, commercial hardware solutions, and proprietary networking and security technologies into one concise system called cumulus::CyberLab. Its’ goal is to allow remote scientists access to shared scientific resources. While each piece of the system has previously existed independently, the combination of all of these pieces into a concise and operational product has not been attempted before by any other company or institution.

---

<sup>4</sup> (Miniwatts Marketing Group)

<sup>5</sup> (Department of Geosciences at Indiana University-Purdue University Fort Wayne)

This combination makes it relatively easy and safe to share resources with the target group of scientists.

With hardware designed to intercept and transmit video signals while sending keyboard and mouse signals to the target computer (KVM), the critical controlling signals from the end user's computer are communicated to the server computer and the video signals are sent back to the user's computer by the project's software. The system then communicates that information to the end user through its' cloud network protocol for reliability reasons and uses AES encryption for security. The end user views the transmitted KVM signals via a remote desktop protocol known as VNC. They can also view laboratory conditions by a web camera which is also shared through the network. Files collected during the session are read by the project's software, uploaded to the main Cloud Network protocol, stored in a secure manner on multiple nodes of that network, and later retrieved by the remote scientist after their experiment is complete. In short, each technological piece is combined into an end user product which allows for equipment to be shared with remote scientists.

Allowing remote access to the resources available at our university to remote scientists further benefits the educational community. Research can be done remotely, students can learn without being on campus, and instrumentation monitoring can also be done without being on campus. Just as printer sharing allowed individuals to print documents without being located right next to the printer, cyber technology allows for the usage of various devices without the user being located right next to the device and, in this case, allows for the remote access of a computer controlling a piece of scientific

equipment. This makes the equipment more accessible and therefore more useful than it was previously.

## **2.2: Remote Access**

Many programs to remotely access computers are available on the market. These software packages provide the needed functionality to allow a person to remotely access a computer system. While these programs provide solutions for various real world problems, they usually do not provide a solution adequate for the needs of a scientific laboratory by lacking sufficient encryption, user account level security, scheduling, and operating system independence. To better understand the products on the market including their limitations, this sub-chapter will discuss the technical background of the various remote access programs as well as the historical usage and motivation behind those products.

The facilities at Youngstown State University are like facilities at any modern university in that they provide expensive scientific equipment which solves specific problems. Giving these pieces of equipment remote accessibility has been the goal of our university's X-Ray Facility. To that end various solutions were attempted with various degrees of success and failures. By reflecting upon past events, cumulus::CyberLab was developed to resolve the specific needs of scientific facilities and other interested parties across the nation.

### **2.2.1: Technical Background**

Remote Access can be broken down into several various products released by various vendors. Microsoft releases a Windows Remote Desktop Connection product with every new version of their operating system, Symantec has available software called

PCAnywhere, and other open source software products exist such as VNC. While each product has differences in what features they provide for remote access, they all provide remote access capability and were used interchangeably.

All of the previously mentioned products provide remote access to a computer. You install the “server” part on the instrument control computer and the “client” part on the computers which will access the instrument control computer to the remote “client” user. The “client” computers access the “server” computer by providing a connection password and the connection is generally unencrypted. Still, you can access the computer remotely but it is on a first come first serve basis where only one person can be controlling or viewing the shared computer at once. Additional features provided by some software products include file transfer of varying degrees of usability and security. The price of each software product is different, with some products being free, thus making pricing a potential deciding factor when determining a solution.

The core aspect of providing a way to remotely access a computer is present. The other features such as file transfer, security, scheduling, communication, and providing information require solutions to be found by the end user. Those features are generally outside of the scope of remote access. Various other solutions can be researched and used to solve each problem individually and then used by the end user in combination with the remote access products to do what is required.

### **2.2.2: Analysis of Prior YSU Research**

An integral feature of the cumulus::CyberLab project is the remote desktop aspect. Prior research into the realm of remotely accessing scientific instrumentation was carried out at our university. This research, while seeking to resolve a similar problem

that is addressed in this thesis, addresses a significantly different aspect of remote desktop access which is undesirable for this project. In order to best explain the undesirable qualities, further analysis of this prior research is required. The general goals and software programs analyzed, the reasons why these programs were inappropriate for the needs of our instrumentation facilities, and the differences between proprietary and open source solutions shall be discussed to clarify why VNC was selected for this project.

Youngstown State University (YSU) in collaboration with the Ohio Supercomputer Center (OSC) researched the network performance of remote access programs with the goal of finding performance metrics in relation to transferring text, 2D graphics, and 3D graphics and they published a paper based on their findings. The OSC, utilizing prior experience in remote access programs, selected various emerging remote desktop control technologies to analyze. The programs selected were Microsoft RDP (w/MMR), HP RGS, and Teradici's PCoIP. These were selected over older remote desktop protocols including Microsoft RDP, Citrix ICA, and VNC RFB because the older protocols were only suitable for transmitting text based displays, in the opinion of the paper being reviewed. The differentiation is in the mechanism each protocol uses to transmit information from server to client.

The newer protocols differed in that they analyzed what information was being displayed on the screen at a software level and transmitted multimedia information differently (either by sending it via a separate channel or by transmitting such information via proprietary protocols). Older protocols would transmit the entire visible screen with no regards to separating text based content from multimedia content. The notable problem with older protocols is that they are usually unable to transmit 3D

images because they are not able to communicate effectively with the video card. In regards to transmitting high quality multimedia content from source to client, newer protocols are more efficient and thus more tolerant of bad network conditions than older protocols.<sup>6</sup> However, while the paper shows clear advantages to selecting newer protocols over older ones, it does not address the needs of the current project.

Due to the nature of the instrumentation involved, installing remote access software on them is often not possible. Changes to the installed software on the instrumentation computers could cause unforeseen complications and was not worth the risk from the instrumentation operators. Additionally, older operating systems would not support installation of recent versions of remote access software. A new approach was needed; one where the signals transmitted from the computer to the physical monitor and from the physical mouse and keyboard back to the monitor would be intercepted and translated into a suitable remote desktop protocol. While hardware solutions existed to carry out such a conversion, those solutions were unable to easily determine at that point the difference between multimedia content and text based content. Beneficially, this hardware based approach can transmit any information on the screen including 2D, 3D, and text based content. Unfortunately, any network and speed performance gains possible with the newer protocols are likely to be ineffective since the difference between content cannot easily be determined.

The protocol on the hardware solutions is VNC. The reasoning behind this at first seems somewhat unclear being that the other protocols show beneficial speed and network bandwidth characteristics when compared to older protocols such as VNC. One reason behind this selection is that the hardware solution can only analyze signals sent to

---

<sup>6</sup> (Hudak and Calyam)

the monitor and thus cannot determine what content is being transmitted, therefore likely eliminating any speed benefits of the other protocols. The other reason behind this is that the VNC protocol is an open source protocol whereas the others mentioned in this section are proprietary solutions. This not only effects the development time for the developer of the hardware solution but would also significantly impact the development of cumulus::CyberLab. These impacts influence the decision on which protocol to choose.

A proprietary solution is best considered as a closed box. The performance of that solution can be measured and compared with other solutions but you have to use the box provided to determine performance. In order to be integrated into other programs and hardware platforms, a programmatic API may be available allowing a programmer to talk to the closed box instructing it to operate in the larger programming project. Access to such an API, as well as usage of the API, is subject to licensing fees paid to the proprietary owner. The problem is when such an API is unavailable or prohibitively costly. Therefore, utilization of such a proprietary protocol is only feasible if the protocol fulfills necessary requirements for the project or can otherwise justify the extra effort and financial costs by major improvements over other solutions.

The other solution involved is VNC, which is an open source protocol. In this situation, open source software programs are best described as open boxes. A programmer can use this open box like the closed box described in the prior paragraphs and they can additionally use this box in any other way they can imagine by literally making it an integral part of their software project, if they so choose. If an API is not available, they can create one. Being open source, the availability of such a program is free. In summary, they are guaranteed access to this remote desktop protocol and it will

function with a minimal amount of effort and financial cost. The hardware platform has the flexibility required to closely link this software to their hardware. The cumulus::CyberLab software additionally has the flexibility to link the client software closely to their own software providing additional benefits such as encryption. The benefits outweigh the costs in practical use.

For the initial project specifications, VNC is appropriate to provide remote desktop accessibility. While not the best solution in terms of network bandwidth or transmission speed of multimedia content, the flexibility of using such a protocol, in the larger project with a minimum amount of effort and financial cost, makes the solution good enough for remote instrumentation use at this time. Further development efforts in the realm of utilizing high speed multimedia remote desktop protocols are possible and feasible if such a need arises. However, for the current needs of YSU's instrumentation labs, VNC is sufficient to provide a good balance between costs and features.

While research into other forms of remote desktop protocols was carried out at YSU, such research fell outside of this project. The manner in which information is collected from the host computer differs from the approach mentioned in the literature review undertaken. The ability to use an open source solution when compared to a proprietary solution made the difference in selecting an appropriate remote desktop protocol. In the end, the VNC protocol was selected to be a part of the larger cumulus::CyberLab project which involves the programming of several other features to make a complete project which can serve the needs for YSU's instrumentation labs.



### 2.2.3: Motivation of Remote Access

To have an entire collaborative network of universities sharing computers, which have access to scientific equipment, requires the utilization of remote access technology at a minimum (data archiving is desirable as well). Available vendor solutions for remote access problems do not provide certain features which would provide benefit to the collaborative network. A small selection of vendor solutions will be briefly discussed as well as the desired features not provided.

Remotely accessing the computers controlling the equipment in the X-Ray Facility is possible by utilizing appropriate technical solutions. To that end, we will consider Symantec PcAnywhere, Cisco WebEx, Citrix GoToMyPC, and Windows Remote Desktop. Each product exists to solve a specific problem and provides a general subset of desirable features to accomplish such tasks. By analyzing those items, we can determine a general overview of what features are available to a facility utilizing any of those software products.

Symantec PcAnywhere is a self-proclaimed world leading remote control solution. It provides the ability to manage computers, resolve helpdesk issues, and connect to various devices in a simple and secure manner.<sup>7</sup> Multiple platforms are supported (Windows, Linux, Mac) which allows for remote control of any computer device even if you are unable to enter the building which it is in. Various encryption algorithms are available as well as various authentication mechanisms for individuals wanting to connect to the system (such as network authentication). Remote connection is possible utilizing a variety of connections including cellular connections. With a package

---

<sup>7</sup> (Symantec Corporation , Remote VPN, Remote Server Access Software, Remote PC Manager | Symantec pcAnywhere)

type installer, installation can be done across a network in an automated fashion.<sup>8</sup> In summary, this product provides a remote access solution for an entire corporation requiring remote access to a large variety of server and end user computer devices.

Cisco WebEx provides a variety of solutions to best cater to specific needs and requirements. The product most relevant to the needs and requirements of a scientific facility is the Remote Access product. Designed to allow employees to manage and support remote computers, this product allows for employees to remotely view and control remote computers via a centralized dashboard and allows for remote access to be operated right from the browser. The dashboard allows for centralized administration and it is powered by the Cisco WebEx Collaboration Cloud which provides for IP address blocking, computer specific access codes, phone authentication, and user account access rights. With highly customizable access permissions, user accounts can automatically expire at predefined times. The remote computer can be restarted, system upgrades can be implemented, and file transfers are possible as well.<sup>9</sup> In summary this product allows any technical support problems to be resolved remotely by various company employees providing all necessary features to get the job done.

Citrix GoToMyPC Corporate Edition provides the necessary features to allow for the sharing of a large variety of computers with a large variety of end users all with appropriate administration allowing end users to access familiar computer environments without having to physically be at their computer. Only requiring installation on the server, the clients view the remote computer through their browser.<sup>10</sup> The connection is encrypted and allows for file transfers with copy and paste support, sound transfer,

---

<sup>8</sup> (Symantec Corporation, Symantec pcAnywhere™ 12.5)

<sup>9</sup> (Cisco)

<sup>10</sup> (Citrix Online, LLC., PC Remote Access | GoToMyPC)

remote printing, and multi-monitor support. Usage reports allow for billing and observation of what users are doing.<sup>11</sup> Corporate features allow for assigning groups to members and for multiple members to share access to one computer. With a high degree of control, an employee can set up one time use username/password combinations so they can manually restrict when a computer is used. One can integrate the solution to the company's LDAP, Active Directory, or other account management systems.<sup>12</sup> In summary, this product is a way to allow for employees to access their PCs remotely as needed.

Windows Remote Desktop is a built in, already installed, remote access solution built into various supported versions of Windows. This allows for employees to access computers remotely and also powers the Remote Assistance functionality of Windows. This product does not work on any other operating systems. Features include encryption, file transfer, printer sharing, and clipboard sharing. Users can disconnect at any time and resume their session as needed. There is also support for 24 bit color, smart card authentication, transmitting special key combinations, and transmitting sound.<sup>13</sup> User authentication is dependent on the remote computer where the connecting user can login on the remote computer as if they were actually on that computer. As a free solution, it allows employees to remotely assist or remotely connect to computers with a minimum of fuss.

Overall, all solutions provide for file transfer, remote connection, a variety of user authentication and permissions, a casual amount of scheduling and control of when a person can access a computer as well as what computers they can access, and finally all

---

<sup>11</sup> (Citrix Online, LLC., Remote File Access | GoToMyPC)

<sup>12</sup> (Citrix Online, LLC., Business Remote Access | GoToMyPC)

<sup>13</sup> (Microsoft Corporation.)

are easily installable and usable. Therefore, they do provide features that are necessary and vital for an individual desiring remote access capabilities. With these products designed to allow for remote technical assistance and to provide remote access to computers, they do their tasks well. However, they do not necessarily provide enough features for an instrumentation facility.

An instrumentation facility would find additional features useful for remote collaboration with other interested parties. These features are not available in other commercial products and are features of which cumulus::CyberLab provides. Each feature will be compared to the vendor solutions. To that end, we will discuss file transfer vs. file storage, scheduling vs. on demand access, sensor and data sharing, and software installation vs. hardware signal interception.

Vendor solutions all provide for file transfer where files on the remote computer would be transferred to the remotely connected party. In this situation, files consist of valuable scientific data that would require storage back-up. The vendor solutions do not provide a mechanism to store files on a different server computer, much less in an encrypted manner. The cumulus::CyberLab software handles file transfers by uploading files to an intermediate StorageNode and then handling the transfer from that StorageNode to the end user. Therefore, encrypted user files are redundantly stored in the cumulus::CyberLab cloud and thus available for transfer at a later point. While a vendor solution might let clients connect via a cell phone connection, cumulus::CyberLab will let clients store files in the cloud until they are on a faster connection and can download their files.

Some vendor solutions allow for granular user access where a user can only access specific equipment. Such accounts can be set up with 1 time passwords allowing for access only when so granted by the owner of a computer and usage reports can indicate how long the person was utilizing the equipment for billing purposes. However, this still lacks a nice scheduling solution where a user can request blocks of time to access a machine in which the user can be billed. This allows for groups of users to collaboratively schedule time on a computer and cooperatively share the equipment. This scheduling feature is available in cumulus::CyberLab yet is not a mature feature in some vendor solutions and simply not present in other vendor solutions.

Vendor solutions allow for remote access to a computer system and allow for the remote user to do what is required on the end system. However, certain devices such as web cameras or temperature sensors can transmit far less information if they simply transfer the information to the end client. This is not possible on vendor solutions yet these devices can transfer their web camera or sensor data directly to the client via the cumulus::CyberLab system. Future work can potentially allow for even more scientific instrumentation which provides point by point data to transfer the data directly to the end users without the need for a comparatively bandwidth intensive remote desktop session.

All of the vendor solutions analyzed involved software installation of some kind and certainly required software installation on the computer to which remote users would access. This is sometimes not possible in an instrumentation lab due to older hardware (small sized hard drives), operating systems, or due to unknown software incompatibilities. While some vendor solutions exist allowing for hardware interception of keyboard, mouse, and video signals (KVM) leaving the computer, these solutions

provide very little features outside of direct remote access to the system in question. The cumulus::CyberLab system integrates such a hardware solution so no software installation is required on the server computer and it delivers the client software required to the remote users through the browser. It solves a problem in that no installation is required at either end and features are not sacrificed.

These additional features are beneficial to an instrumentation facility and are features lacking in vendor solutions and yet are provided by the cumulus::CyberLab project. This project brings together various technologies and provides features which other vendors do not provide simply because they do not cater to the market of scientific facilities sharing instrumentation to remote users. The motivation to modernize instrumentation laboratories and to allow for remote collaboration between various universities is the driving factor of the cumulus::CyberLab project and therefore the additional features provided set it apart from the other solutions available in the market and further define and validate the reasoning behind the formation of the project. Yet, to better understand the project as a whole, it is a good idea to study the technology behind it.

### **2.3: Cloud Computing**

The sharing of resources remains a vital feature of the Internet and the manner of which those resources are shared determines the positive and negative issues that can be encountered in daily use. To help understand why Cloud Computing is a vital aspect of the cumulus::CyberLab project, an introduction and comparison will be made of the three mechanisms of sharing information in-between Internet nodes. The cumulus::CyberLab project will be compared to each solution as well as reasons why certain solutions would

not have been ideal choices. Coverage will include Client Server, Peer-to-Peer, and Cloud computing in the exploration of data sharing techniques.

Early computer technology was too expensive for most of the public or businesses to own. When a computer was purchased and used, it was common that it would connect to a larger, more expensive, and more powerful computer which would do computationally expensive tasks and transmit the results back to the slower, personally owned, computer. The more powerful computer became known as a Server and the less powerful personal computer was called a Client. A Server would offer resources in the form of Services to any requesting Client which would access the offered Services. The system was simplistic and worked well enough to share information. Services provided eventually expanded to include such things as the services of various connected devices to the server computer.

Benefits of the Client Server model is that, being a simplistic model, any problems such as network connection issues could easily be isolated as a problem at the Client, Server, or the Network that the communications was traveling. Thus, the reliability of the system could be personally reviewed and checked. Additionally, a server machine would rarely be used for non-server tasks so the separation of computer general users from computer back end processing was helpful to security.<sup>14</sup>

The cumulus::CyberLab project stores information in a network database for retrieval by any connecting clients. While each node is separate and unique, the user administration has to be synchronous across all nodes. Additionally, shared information such as file transfer materials and remote desktop sessions would have to exist on the Server and it could suffer bandwidth issues, reliability issues, or otherwise be

---

<sup>14</sup> (Kurose and Ross 84)

troublesome to maintain. In short, since multiple client and server computers are available in the cumulus::CyberLab project's network and since the server computers may need to take on client roles for certain scenarios, there needs to be another solution for this project.

Peer-to-Peer (P2P) networking is a different approach to networking and provides features which are of interest to the cumulus::CyberLab Project. A young college student, by the name of Shawn Fanning, became a notable developer of P2P software technology after releasing the generally illegal software program called Napster. That program best illustrates the Peer-to-Peer concept where a personal computer and a remote computer both act as servers and clients. As a client, the computer makes requests to other nodes connected for mutual sharing benefits. As a server, the computer answers any incoming requests for files and transfers them to the requesting machine. No one machine is the central server for the group and each individual entity is able to be a downloading (client) and hosting (server) platform as needed.

A core aspect of P2P programming is the removal of the dedicated server environment and replacing it with member pieces that work together taking on client or server roles as needed to form a low maintenance network that is extremely resistant to shut down. Collectively, each node (computer that is part of this network) passes along requesting information to each of the other nodes to help locate requested materials. When requested material is located, the node hosting the information and the node requesting the information connect directly to each other to facilitate the necessary data transfer. Individually, a node is likely to share little but collectively the group of nodes



usually offers desirable services to some other node in the P2P network. This ability to dynamically form a network is appealing to the project at hand.<sup>15</sup>

What was previously described is an unstructured P2P network. A structured P2P network minimizes disruption of the network by sharing a hash table around to other nodes indicating where information is located. This allows a node to more easily locate resources and for those resources to be available even in the dynamic network structure where nodes are often entering and leaving. However, this does not provide any sort of duplication of data which means that accessing specific data is impossible when the node containing the data goes offline so long as no copies of that data exist elsewhere. This P2P network type still makes the assumption that any node can leave the network at any time.<sup>16</sup>

While both unstructured and structured networks have resistance from being forcibly destroyed, especially when compared to a Client Server setup, there are no guarantees or likelihood that a particular service will be available or online when its' needs are requested. Participation in the group generally has low restrictions and it is treated as a more or less open network allowing for member nodes to come and go as they please. While the technical merits of a self-assembling network are desirable, the lack of security and stability in having a resource online when needed makes this network methodology unsuitable for the project.

Cloud Computing takes the approach of P2P networks and expands the concept so that the problems of having unavailable resources and low security are avoided. Nodes participating in this network are expected to remain a part of the network and, while they

---

<sup>15</sup> (Kurose and Ross 84)

<sup>16</sup> (Kurose and Ross 152-156)

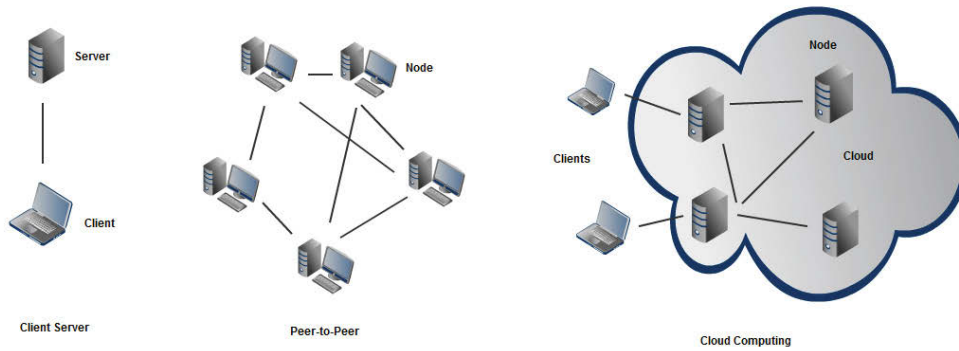


Figure 2 -- Network Types

can take on client and server roles, they are expected to usually be acting as servers.

Nodes communicate amongst each other so that the loss of nodes would only reduce the services offered and would not cause any major network problems. Connecting Clients communicate with their choice of node and access resources provided by the cloud and provide no resources in return. The end result is that the client only sees one unified server resource which they connect into to access the services located within the cloud.<sup>17</sup>

Benefits to this system involve easier developing networks, controllable security across the whole network, controlled access to the network, and the improvised resistance to network degradation by inaccessible servers. Therefore, one resource shared with the cloud can be accessed by any other cloud piece which also can accept any incoming Client connections. A direct path to the node physically sharing the resource is not needed as a path will be determined from any node to any other node.

Drawbacks include the expected requirement that any node in the cloud should have the best uptime possible. Since a node may serve as the only method of accessing one cloud area from another cloud area, technical issues in that node can cause limited cloud functionality. Additionally, access to such a network is generally limited because

---

<sup>17</sup> (Buyya, Yeo and Venugopal)

that the target market is businesses which can provide high reliability computers as opposed to the general populace. None of these issues greatly influence the project since the nodes in the network are going to be used in a business-like environment where best possible uptime is a goal.

The development of newer techniques to share information has developed a method which is useful for this project. While the paradigms of Client Server and P2P is used commonly, Cloud Computing is a service most companies are adopting due to the benefits provided and the ability to easily compensate for the drawbacks. With the Cloud Computing network in place, we now have the foundation for the rest of the project.

#### **2.4: Case Study – X-Ray Diffraction Scientific Equipment**

Members of the scientific community such as educators, scientists, and students commonly carry out experiments using equipment that is not easily obtainable or portable. Various solutions exist to work around those problems but the integrated solution that this project introduces provides a unique alternative. We will discuss one of the pieces of scientific equipment in detail, problems facing scientists who wish to use such systems, and how this project intends to resolve those difficulties.

One model of X-ray Diffraction equipment that YSU currently utilizes is the Bruker AXS D8 Advance. The D8 is a room sized piece of equipment utilizing a variety of water cooling, extremely low temperature fluid cooling, computer control equipment, as well as precise operation by a skilled equipment specialist.<sup>18</sup> Costing well over \$200,000, the equipment is neither portable nor inexpensive to obtain. The machine is designed to measure the diffraction of an X-ray beam as it passes through a sample in

---

<sup>18</sup> (Bruker AXS Inc, Diffraction Solutions D8 Advance)

response to the compounds contained within it. A skilled technician with a good sample can get valuable information about a sample's composition by utilizing this equipment.

The D8 is controlled in part by a software program written for the control computer connected to the machine. This control computer is running Windows XP and is dedicated to the operation of the machine's control program. The software program controls the complicated measuring variables needed to collect information that the scientist desires. Without access to that program on that computer, the instrument cannot be controlled.

Utilization of that computer is required for an experiment to be run on a sample. A local scientist will insert a sample to be measured into the D8, and then collect data from the sample by utilizing the computer to operate the scientific instrument, and finally transfer the data to another workstation to process. The data collection part of the process is beneficial to the scientist because the scientist conducting the experiment on that sample is present to control the data collection. The scientist instructs the machine to collect the specific information that is desired which may change during data collection. Additionally, data collection can be observed and terminated or modified if the information being collected is not useful due to the quality of the sample, the information desired, or new information determined early on in the data collection process. This value added ability to control the data collection process either requires the Scientist to travel to the instrumentation facility or for another solution to be developed.

Depending on the connections the facility has with the outside world, samples will be sent in from various parts of the world to be measured at the facility. It is possible for a scientist located close to the facility to travel with minimal financial costs and time

constraints. That option is not possible for a scientist located far away from the facility. The scientists who cannot travel to the facility have to rely on the local staff to run the data collection efficiently and hope that standard data collection procedures yield the desired data. If the information returned is not useful or if there are still questions remaining then the same sample may be sent multiple times to be run again with slightly different data collection requests. This technique, while not ideal, was the best possible solution until recently.

The cumulus::CyberLab project provides a remote desktop access window to the D8 controlling software. The machine can start data collection on the preloaded data sample and the scientist, who is remotely connected to the machine, can control the equipment. When the procedure is done, the collected information is provided back to the scientist. Overall, the scientist gets far more value from a facility that offers the ability to remotely control the machine and control the operation of the equipment.

The additional equipment that the cumulus::CyberLab project provides connects to the control computer of the D8 in a non-invasive manner by hooking into the keyboard, video, and mouse signal cables. Those signals are translated to and from a form, which is easily transmitted across the cloud network that the project uses, to and from the intended recipient. At the facility, the only thing that is required is for the project's Symbiote hardware device to be plugged in and turned on. After that, it does everything else necessary to integrate into the larger cloud and provide its' services.

The project has security measures in place to ensure security to the expensive equipment at the X-Ray facility as well as ensure security of the session and data collected. This is done by an all encrypted communication network which is encrypted

by AES. Access to various pieces of hardware is done via username and password which has fine grained user permissions which specify what resources they can access, when, and which members have priority. Detailed logging provides detailed information on what transpired and when to allow for billing, auditing, and analysis of machine use patterns.

The cumulus::CyberLab project provides more features for the facility owner than a simple remote desktop application can provide. The cumulus::CyberLab project also provides more features to the remote user. The whole project is designed as a good alternative to sending a physical person to operate next to the very immobile and expensive physical machine. To that end, this project fulfills those goals very well.

While there are several ways to solve a problem, better solutions can be developed by pursuing promising ideas. By leveraging the power of the Internet, a problem of bringing a scientist to scientific equipment can be resolved in a new manner. While the D8 is quite expensive, this project allows for more scientists to use previously purchased equipment as opposed to having to purchase equipment for their own organizations. Overall, a facility with a remote desktop advantage is more useful than one without.

### **Chapter 3: Cyber Technology Software**

The development of cyber technology software was made possible by a Youngstown Ohio company by the name of Zethus Software. Their company has made the cumulus::CyberLab project possible and has allowed for the resolution of the problems and motivations previously introduced in this thesis. The following section describes in detail Zethus Software.

#### **3.1: Zethus Software Overview**

Zethus Software who developed the cumulus::CyberLab technology in collaboration with Youngstown State University is a successful technology start-up company who made software which fulfilled the needs of our university. The success of their company deserves mention and is best described by analyzing the Youngstown Business Incubator and Zethus Software in more detail. These companies help explain why Entrepreneur magazine listed Youngstown Ohio as a top 10 city to start a business in 2009.<sup>19</sup> These companies helped make the software project explained in this thesis and helped put Youngstown in the news.

Where it all began was the Youngstown Business Incubator (YBI). Formed in 1995 and adopting a mission statement that focused the organization on information technology in 2000, YBI provides starting businesses with the resources required to become successful companies. Providing building space, free internet access, cheap telephone service, and video conferencing solutions to businesses helps alleviate the requirements for those businesses to acquire business reliable communication mechanisms. By proving on site entrepreneurial counseling as well as networking

---

<sup>19</sup> (Daley)

opportunities, companies in the YBI can develop and find customers. Companies have at their disposal many resources they require to grow.

The business incubator, a non-profit corporation, is located in a donated building. The 5 floor building was constructed in 1915 and required renovations before becoming operational. The renovations were funded by the Ohio Department of Development (ODOD) and current operating costs are also provided by ODOD. Due to core values of ODOD and analysis of the local business environment, the YBI became an incubator which mainly services the information technology business field. In general, business to business relations are encouraged over business to client operating models. The stage was then set for various companies to take hold and develop through the YBI providing information technology related products to any market from local to worldwide. This allowed Youngstown, formerly a steel producing powerhouse, to have a bright future in the field of information technology.<sup>20</sup>

Zethus Software is part of that bright future. Being founded in 2003 by Eric Parker and Andrew Reinhardt,<sup>21</sup> Zethus became a company focusing on Grid and Cloud computing solutions to solve real world business problems. The core software that Zethus produces was developed over the last several years to solve a variety of business problems relating to sharing of storage space, computer resources, and to provide remote access in a seamless cloud. The core software resolves those problems in a secure manner with low administrative overhead. Zethus uses rapid development techniques; thus, software solutions are developed quickly and reliably delivered. The company has the core team required to develop software projects for businesses and universities.

---

<sup>20</sup> (Youngstown Business Incubator, History)

<sup>21</sup> (Zethus Software LLC., Zethus Software Overview)



Dr. Allen D. Hunter, working with the YBI, was able to learn of Zethus Software's company focus and was able to collaborate with them to make cumulus::CyberLab a reality. The university provided a project, now known as the cumulus::CyberLab project, funded by an Army Research Laboratories grant as well as university interns to Zethus Software who developed the software to meet the real world needs of YSU. Through several years of development and testing, the cumulus::CyberLab project came to fruition. The project was developed to share remote access to various pieces of scientific equipment that were developed by a company called Bruker AXS. In this regard the university can collaborate with other universities sharing resources and benefiting from the exchange. This was all made possible by the networking opportunity given to Zethus Software by the YBI to collaborate with YSU.

The YBI mentions networking as a core component of the services they provide.<sup>22</sup> Allowing businesses to meet and interact allows both to benefit from the exchange. Bruker AXS is a global market and technology leader in scientific equipment<sup>23</sup> and Zethus Software is on the cutting edge of Cloud and Grid computing development. Through Youngstown State University who was introduced to Zethus Software through the Youngstown Business Incubator, Zethus Software met and networked with Bruker AXS. By providing software which fulfilled the needs of Bruker AXS, Zethus Software and Bruker AXS signed a Worldwide Strategic Partnership on March 23, 2010.<sup>24</sup> Through collaboration, our university made an impact providing business opportunities previously not available to either company. Additionally, our university now has the use

---

<sup>22</sup> (Youngstown Business Incubator, About Us)

<sup>23</sup> (Bruker AXS Inc , About us)

<sup>24</sup> (Zethus Software LLC., Bruker AXS and Zethus Software Signs Worldwide Strategic Partnership)

of cutting edge cloud software which allows our university to collaborate better with other universities.

This region has been looking for a bright future ever since the collapse of the region's important industry, the steel industry. That bright future is becoming a reality in part due to the actions of this university, the Youngstown Business Incubator, and Zethus Technologies. Entrepreneur magazine is correct in saying that Youngstown is one of the top 10 cities to start a business since this city is waking up into a new era of business development. Youngstown has all the pieces required to develop into a new era of business, job, and economic development and Youngstown State University is proudly a part of that future.

### **3.2: History and Vision**

Zethus Software, founded in 2003, has continued its' mission statement goals year after year to develop a wide range of software products which eventually lead to the cumulus::CyberLab project. Their employees made the company successful over the years. Various technologies utilized in the company have resulted in Zethus developing intellectual property which made the cumulus::CyberLab project a reality. As such, their company was a perfect fit to fulfill the needs of Youngstown State University.

The mission statement of Zethus Software is quoted below.

“Mission: Design and develop software tools that allow users to concentrate on their work, not operate computer systems. There is a sweet spot in software development where innovation and theoretical work can be applied to real-world problems to produce an amazing, positive impact

on a person's ability to perform a task – the system becomes transparent and the task illuminated.”<sup>25</sup>

Software products were developed by Zethus Software which solved real world business problems. Each successive software product expanded the amount of software tools available to Zethus Software for them to develop future products. Over the lifetime of the company, the products Coherent Modeling, WebTest, Prospect Catalyst, zsJobQuote, GRIP and cumulus::Archive were developed.

The developed software products used cloud and grid computing technologies. Such grid computing technologies were utilized with the goals of providing low system administration overhead and to allow for a loose collection of computing devices to provide services in a seamless manner to the end user. With grid computing and cloud computing being embraced over the company's lifetime, these technologies help define the core of Zethus Software products. Around that core technology surrounds the necessary software programming to make that technology come to life.

The programming languages used for the current generation of Zethus Software products is ObjectPascal, Java, JavaScript, Python, and PHP with a transition to using more Java and to incorporate C and to use the JavaScript, Python, and PHP programming languages in special cases. The current cumulus::CyberLab software utilizes for embedded code JavaSE, JavaSWT and Swing for user interface coding, and PHP for web page design. Overall lines of code in Zethus Software products, while only a minor glimpse into the complexity of the software programs, is around 300,000 lines. The

---

<sup>25</sup> (Zethus Software LLC., Zethus Software Overview)

programming of each software product reflects the experience of the Zethus Software staff in resolving real-world problems with practical software products.<sup>26</sup>

One of the co-founders of Zethus Software, Eric Parker is the current lead software architect. He was also the lead software architect for the cumulus::Quorum third generation grid/cloud framework. His current work with cumulus::CyberLab was made possible in part by the existence of cumulus::Quorum. Eric was the lead architect and project leader for distributed engineering systems for Bridgestone/Firestone before his role in founding Zethus Software. He also has a B.S. with honors in Computer and Electronic Sciences from the College of Wooster.<sup>27</sup>

The other co-founder of Zethus Software is Andrew Reinhardt who is the current Vice President of Engineering. His focus is on the application development of the various cumulus products. He was an integral part of the cumulus::Quorum development and continued work into the cumulus::CyberLab development. Before founding Zethus software with Eric Parker, Andrew was a Project Analyst and Senior Engineer for Bridgestone/Firestone. Andrew also earned his M.S. and B.S. degrees in Mechanical Engineering from Case Western Reserve University.<sup>28</sup>

The Zethus Software staff in carrying out their jobs to fulfill their mission statement developed intellectual property which is the concepts and code incorporated into various products empowering the latest products from Zethus Software. This intellectual property is licensed upon contractual agreement with Youngstown State University to allow for the development of CyberLab. Collaboration between both Zethus and Youngstown State University utilizing existing intellectual property helps

---

<sup>26</sup> (Zethus Software LLC., Zethus Software Overview)

<sup>27</sup> (Zethus Software LLC., Eric Parker, CTO & co-founder)

<sup>28</sup> (Zethus Software LLC., Andrew Reinhardt, V.P. Engineering & co-founder)

develop new intellectual property beneficial to both parties. Therefore, both parties benefit from the shared collaboration established with the foundation of the cumulus::CyberLab project.<sup>29</sup>

With the foundations present from Zethus Software, a new software product was developed. This new product, the already mentioned CyberLab, was developed after essential rational was established. This project was made possible in conjunction with the mission statement, technologies developed and utilized, the staff involved, and the existing intellectual property. To that end, Zethus Software provides a stable foundation for new software development.

### **3.3: Rational of cumulus::CyberLab**

With the collaboration of Youngstown State University, Zethus Software had a new goal in regards to new software development. In regards to the desired goals for the project, characteristics about this new project were noted and recorded. Additionally, elements of the overall project were drafted to allow for development of individual pieces. The total representation of this information explains the rational of cumulus::CyberLab and explains the project as it eventually was developed.

On February 26, 2008 Dr. Allen D. Hunter, a YSU Professor of Chemistry, became the principal investigator for the cumulus::CyberLab project as YSU entered into a contract with Zethus Software for software development for the project. By allowing complicated chemical analyses to be carried out remotely from multiple locations, the functionality and usability of testing equipment located at YSU would be increased. This would decrease the cost per sample of operating the equipment and increase the number

---

<sup>29</sup> (Zethus Software LLC., Zethus Software Overview)

of users that can access the equipment by allowing for remote access. Thus started the development of the project as well as the definition of what the project entails.

This project as defined would allow the sharing of expensive scientific instrumentation with members dispersed geographically and organizationally in the manner of a virtual community. A goal was to have near zero negative impact on the instrument control system and to be intuitive for the members in the virtual community. With very little administrative effort, the system should be robust flexible, and able to easily grow. The systems involved should be inexpensive using commercial grade hardware and networks. In the end, future use as appliances should be possible.<sup>30</sup>

Elements of the new project define the various sub-sections of the project which would come together to form a unified system which would be utilized by various members in the virtual community. The sub-sections also define how work would be distributed and how the project would be developed. The whole system entails every element being present but does not in itself specify the amount of work needed for any one piece to be finished completely.

A core part of the new project was the Instrument Remote Desktop Tool which provides to a member the ability to remotely control scientific instrumentation remotely. Reservations for time on the instrumentation would be handled in this element as well as all general user interaction. Scientific data collected during their session would also be handled by this element. This element developed into the cumulus::CyberLab user tool.

In relation to the prior element, the KVM+ InstrumentNode element defines the unique hardware necessary to provide the capabilities to the Instrument Remote Desktop Tool allowing that tool to remotely control an instrument control system (aka instrument

---

<sup>30</sup> (Zethus Software LLC., Zethus Software Inks Deal with Youngstown State University)

computer) with near zero negative impact. Additionally, this element constitutes the abilities to transport data from the session to the end user, scheduling, and characterizing the instrument to define its' overall role in the system.

Another vital part of the project is the Administration Tool which is utilized by the system administrators to manage the cumulus::CyberLab system. This allows for analysis of the cumulus::CyberLab system to pinpoint any problems, configure the system to fit specific needs, and to add or remove users as needed.

Empowering the previously mentioned elements is the cumulus::Quorum base technology element previously developed at Zethus Software. Similar to the operating system of a computer, this element provides a base for the rest of the cumulus::CyberLab software to function. Defined in this element are authentication, auditing, security, organization, alarms, and monitoring.

Another element empowering the previously mentioned elements is the cumulus::Archive element providing data archive technology. Allowing for scalable, efficient, secure, and cost effective data storage, this element will allow the large datasets collected during scientific instrumentation operations to be stored, sorted, and organized. When requested, the information can then be transmitted to the end user and deleted from the system.

The Website Portal element provides a gateway to members in the virtual community interested in accessing the cumulus::CyberLab system. Providing a mechanism for members of the virtual community to retrieve the necessary software required for authorized interactions with the system, this Portal allows for the members of

the community to run the software involved with this project without any local installation on their own computer and to run the software anywhere as needed.<sup>31</sup>

The elements, when combined, form the cumulus::CyberLab system which carries out the goals and motivations requested by the principal investigator. In the end, the overall software product is a seamless unified system which only hints at the elemental approach to assembling the overall project. This is accomplished by following the development principles and phases as defined by Zethus Software.

### **3.4: Development Principles and Phases**

With the overall goals and elements of the project in place, development principals and phases are utilized to help the implementation of the project. With specified development principals, the goals of the project are emphasized and kept in focus. With development phases, the development effort of an element is known and followed to assist the element in being developed. These principals and phases are therefore important to the development of cumulus::CyberLab.

The specific development principals of cumulus::CyberLab were defined in part in the rational of cumulus::CyberLab. These goals are kept in mind as development of each element takes place to ensure that the goals are met. To that end, each development principal will be analyzed to help define and specify cumulus::CyberLab.

The project involves several nodes which should be allowed to come or go, be added or removed, or otherwise exist in a highly distributed and loosely organized network. This allows for no single point of failure or attack point. This is additionally specified in the grid/cloud like nature of the network that the nodes form and is a desirable feature for cumulus::CyberLab.

---

<sup>31</sup> (Zethus Software LLC., CyberLabNet Overview)



Well defined interfaces between modules and elements allows for a Service Oriented Architecture. This allows for extensible collaborative development efforts. Coding becomes more sustainable and flexible. User interfaces, being that they issue service calls to the system, can be replaced with automation. In general, programming changes and maintenance is easier with this kept in mind.

Programming is carried out with modules with well-defined interfaces and decoupled functionality. This allows for better design and as necessary refactoring and extensibility down the line. This minimizes programming effort required in the later phases of the project and allows for more focus on features and requests.

The program is responsible for moving a large volume of data quite frequently. A data centric design, where the system acts like a pipeline for the data, moves the data through the system while having an understanding for the data that is flowing through. Using meta-data, data is described by more data which gives the ability to identify data easily and move data around efficiently.

The program, while being accessible through a web browser, is provided as a web launched rich internet application. This has the benefits of no installation being required and providing better capabilities than an in-browser solution which would be restrained by the browser's capabilities.

Open source code and frameworks were utilized when possible preventing lost time and effort in re-implementing solutions already available. Licensing issues were avoided in comparison to utilizing closed source solutions.

Time and effort were also saved by having a cross platform code foundation for this project. This allows for additional flexibility in hardware construction and for

browser software deployment to a wide variety of virtual community members in the finalized system.

System state and commands are to be communicated via messages that have semantic data (e.g. XML). This is in part related to the Service Oriented Architecture but has additional emphasis for the purposes of internal system communications.<sup>32</sup>

These design principals are iterated and remembered during the design phases that each element has to go through. Each element can be blurred between two adjacent development phases and multiple elements can be in differing development phases at the same time. The phases help with design and collaboration to help speed up development efforts.

The Design phase is where the functional and technical documentation for the element is generated and finalized. Those documents serve as the blueprint for the element. In this phase, design models, functional specifications, guidelines, and technical methodology are detailed. The overall idea is to go from defining high level overview of the element to the technical how to carry out construction of the element before the next phase begins.

The Proof of Concept phase follows where proof of concept examples of the software is created to study the mechanisms and concepts of the project. These examples do not have to be interactive. The goal is to ensure that all design phase work was accurate and desirable and to choose what works best before continuing.

Prototypes are developed utilizing what was learned from the prior two phases. These prototypes are interactive and undergo internal and alpha testing and specify continuing work towards a finalized product while still allowing for non-major changes.

---

<sup>32</sup> (Zethus Software LLC., CyberLabNet Development Principles)

Developed elements start to take form and definition. This sets the foundations for the final phase.

In the development phase, the elements are constructed utilizing the work from the prior three phases. Extensive internal testing and beta testing takes place as well as early adopter testing. The elements are operational, working, and have all features requested in the design phase completed. Fixing software bugs is considered an aspect of the end parts of this phase. In general, this phase is the final phase of the development phases.<sup>33</sup>

With the conclusion of all phases for each of the elements, keeping in mind the development principals previously described, the cumulus::CyberLab project reaches conclusion as a finalized product reaching distribution. Future work would likely utilize the same approach utilizing the existing code base as a foundation to permit such work. In conclusion, this approach was successful leading to the project as it stands today.

### **3.5: Lessons Learned**

The cumulus::CyberLab project was additionally an internship for a fortunate Cushwa Fellow scholarship recipient at Youngstown State University who is the author of this paper. Their work over the past few years with this project has taught valuable lessons to be utilized in future work opportunities. Below is a brief summary of lessons learned while at Zethus Software and lessons which will remain useful for years to come.

The departure from classroom experiences and the entry into a long term internship brings with it an aspect of long term development. A typical class session lasts 15 weeks and this project lasted several years. While a classroom experience begins and ends within that 15 week period and a project can easily fail to conclude, the software

---

<sup>33</sup> (Zethus Software LLC., CyberLabNet Development Phases)

project at Zethus Software concluded on time and was successful. To that end, the company itself is of interest.

Being a small startup company, there was a small number of employees (3) and interns (2) at Zethus software. This emphasized the need for each individual to assist as requested and where possible to help the project achieve success. The efforts put into the project by the various Zethus employees are both commendable and remembered as a lesson that hard work pays off. Their techniques in handling situations and proceeding onwards with their current project eventually leading to a successful completion is a testament to any company with a project to work hard and see it through to completion.

The author's time with Zethus Software has not only assisted Zethus but also provided valuable lessons for the future. These lessons will assist in future job opportunities and will assist in the selection of wise decisions as needed. To that end, the internship opportunities provided to students at Youngstown State University should not be passed up and should be pursued where possible.

## **Chapter 4: Development and Evaluation of cumulus::CyberLab**

During the development of cumulus::CyberLab, Youngstown State University's involvement in the project provided assistance to both our university and Zethus Software. An evaluation of the system administrative tasks performed, the various testing done, and of the overall project is described in the following sections.

### **4.1: System Administration**

Deployment and support of the cumulus::CyberLab system involved several system administration tasks. These tasks ranged from initial construction of each node to end user technical support. This specific section of the thesis describes hardware assembly, operating system installations, installation of various required support programs, and end user troubleshooting of deployed systems. Each of those activities are activities that a system administrator is expected to carry out and activities which are required for the project's success.

#### **4.1.1: Utilized Hardware**

At the heart of each Symbiote is standard computer hardware now consisting of Mini-ITX form factor motherboards, laptop and desktop hard drives, and various form factor power supplies in various case sizes. Several different hardware and case designs were tested and utilized before a standard form factor was determined. Each case variation was replaced for reasons of construction problems or general usability issues. In the end, two form factor cases were standardized being a general Symbiote node for the InstrumentNode and SensorNode devices and a general StorageNode type case.

Each variation of case design had specific strengths and weaknesses. The weaknesses of earlier case designs lead to the development and utilization of later case

designs. Each case design allowed the Symbiotes to be manufactured and tested in the early beta phases of the project and provided a foundation for later case designs to be researched and utilized. Notable features of each case will be mentioned, however, each case design will only be briefly reviewed.

Initial case design consisted of generic Pico-ITX and Mini-ITX case hardware. The Mini-ITX cases carried out the role of the InstrumentNode which allowed for remote access to another computer while the Pico-ITX case carried out the role of the SensorNode which allowed for web cam observation of the environment. The Mini-ITX cases lacked necessary case holes for the KVM hardware and additionally had optical drive bays which would not be used in this system. The modification of those cases to allow for the KVM hardware to be installed was time consuming and not aesthetically pleasing to a customer. The optical drive area had to be blocked off so as to avoid confusion by an end user. Additionally, ports which should not generally be accessed by the end user which are built into each motherboard were made available by virtue of the case design which would lead to end-user confusion since the ports were similar to the ones offered on the KVM hardware. The Pico-ITX cases also had issues.

The Pico-ITX cases, being much smaller than the Mini-ITX cases, initially seemed like an ideal solution for allowing a web camera to be shared in the cloud. However, the hardware required to utilize this case was different than the hardware used in the Mini-ITX cases and required different programming libraries to properly utilize. This made this case and hardware another platform which needed support. Having less platforms would be ideal for keeping support costs low and for troubleshooting end user problems.

Another case was selected which was the size of a pizza box. This case was selected to be the initial case design of a storage node which consists of several hard drives providing a large amount of storage compared to the other two hardware platforms. These boxes were capable of having two drives for storage capabilities. The problem is that they could not be expanded to include additional drives. Additionally, these cases also exposed ports to the end user that should not be used. Still, these cases had relatively few problems compared to the other cases.

While the initial two cases (KVM and Sensor) were sufficient for initial testing, there were many problems which needed resolving or which caused construction time or programming time to be increased. Future iterations of cases would have to provide an ideal solution for these problems without introducing new problems of their own. These ideas were kept in mind when future case designs were selected. To that end, the Storage case became the new KVM and Sensor case designs resolving problems by not having a place for an optical drive and being a standardized platform for development. However, the remaining port accessibility problems (ports being available that should not be and needing to physically make ports for the KVM hardware) means that a better solution was needed.

The next iteration of cases were custom cases purchased from an outside vendor. Having appropriate port outlets for the KVM hardware, ports hidden which should not be used, sufficient internal space for all additional components, and being completely appropriate for the custom specifications for the cumulus::CyberLab project, these cases are ideal. Still, the needs for a SensorNode and InstrumentNode differ from the needs of a StorageNode. The StorageNode case has room for 5 internal hard drives and the

SensorNode/InstrumentNode case is significantly smaller only having room for the specialized hardware required for those uses. With no known problems, these cases work perfectly for current uses.

With all case iterations, assembly was required. The cases did not have the necessary hardware installed in them and such installation was required before the cases could be useful. The key factor in each case design is that space is greatly limited and optimal placement of the internal hardware is important in terms of internal cooling and lowering the time required to assemble each Symbiote. Wiring inside the cases had to be kept to a minimum where possible to promote airflow inside the case and wire length had to be sufficient for all components. Each system needed construction to working specifications with all possible errors during construction avoided. Through several iterations of cases, all major construction problems have been addressed and resolved in the manners previously indicated.

With the appropriate construction of each Symbiote, the next phase of preparation was to install the software required to allow them to function. Software installation required the installation of an operating system and then the specific Zethus Software to provide the cumulus::CyberLab service. Details about the installation of such software are included in the following section.

#### **4.1.2: Operating Systems and Installation**

All constructed Symbiotes provide one or more USB ports, a network port, and internal or external video, keyboard, and mouse ports for initial installation and various troubleshooting tasks. Operating systems were installed with USB CD-ROM drives utilizing one of two techniques and requiring video, keyboard, mouse, and network



access utilizing traditional KVM hardware to allow many installations in parallel.

Various scripts and installation documents outlined software pre-configuration required before Zethus specialized software was installed on a device to make it into a Symbiote.

The operating systems currently in use or used in the past for this project are various versions of Ubuntu and Open Solaris. For Ubuntu, a monitored automated install was created and Solaris requires a manual installation utilizing a USB flash memory device for quick installation times. Both installation techniques minimize installation times and provide a known static operating system upon completion of the installation procedure pending further software configuration.

Ubuntu is based on a Linux operating system known as Debian. In turn, Ubuntu supports an automated installation technique known as preseeding where a file is placed onto a CD image defining how the installation should proceed and what additional software or steps are needed before installation is considered completed.<sup>34</sup> A preseeding automated installation procedure for Ubuntu was developed to allow for automated installations and appropriate documentation was created to allow for this procedure to be duplicated on future Ubuntu versions as needed. The installation procedure requires no keyboard or mouse interaction and is fully automatic only requiring the keyboard, monitor, and mouse for any problems during installation such as a missing network connection or a corrupt CD. In the end, an Ubuntu operating system is installed with SSH access and a specially configured Java configuration allowing for software installation to begin.

If the specific device is a Sensors device then an automated script is executed which installs required software to enable the yet to be installed cumulus::CyberLab

---

<sup>34</sup> (Ubuntu documentation team)

software to control the web camera and remote sensors. The scripting obtains known versions of the supporting software and compiles them from source with installation following. The system is then instructed to start those software services on system start-up. Within approximately ten minutes, the script will have installed all the required support software for the cumulus::CyberLab system. Future versions of this script will include Samba file sharing support which is in early testing.

At this point on all systems an automated script is run, different than the one mentioned in the last paragraph, which installs the cumulus::CyberLab software. The script determines if everything has been installed correctly and ensures that no programs will cause the installation to fail. When that is confirmed, the cumulus::CyberLab software is installed and the operating system is directed to start the software when the Symbiote is launched via Ubuntu's Upstart service. When this procedure is completed, the Symbiote is fully setup.

An Additional operating system was used during the development of the StorageNode. This operating system, known as Open Solaris, provided an interesting file storage system, known as ZFS, which provides increased performance when compared to RAID as provided by Ubuntu. The procedure for installing Open Solaris differs from Ubuntu. These differences increase the amount of work needed to support all Symbiote platforms. Therefore, Open Solaris provided an alternative to Ubuntu and was tested to see if it would be a suitable operating system for a StorageNode. While testing eventually proved this to not be worthwhile because of the added support costs not justifying the minimal performance increases, this other operating system is worth discussing.

The Solaris operating system has only experimental support for automated installations. The installer for Solaris is very slow causing significant delays in operating system installation. Various types of CD media were tested to determine transfer rates and installation times concluding with a test utilizing USB flash memory sticks to compare their rates of installation. It was determined that the flash memory sticks were the fastest at installation so they are currently used for general installation. Configuration of this operating system concludes with an Zethus installation document which outlines the configuration of the ZFS file system and other software services on Solaris. The steps are not automated but they take less than five minutes to manually execute. The time required for installation of the operating system is significantly longer so installations are usually done in parallel using a KVM to switch between installations.

The finalized system has cumulus::CyberLab software installed by Zethus employees utilizing an automated script similar to the Ubuntu install. The software configures itself and allows itself to be run by the operating system when installation is complete utilizing the Open Solaris services. The final end product is then tested and receives further configuration as needed before being sent to the end customer. The end product is designed to be as robust as possible while needing minimal servicing.

The instructions, while similar to describe, were not as automated as the Ubuntu install. This required more interaction from Zethus employees and increased the amount of time required to install all required software. Additionally, having multiple operating systems to support raises support costs in that testing and documentation would have to describe both platforms. In the end, the Open Solaris platform was abandoned in favor of

Ubuntu due to the speed of installation and the ease of working with Ubuntu in setting up installation cases.

When all software installation procedures are completed, the finished Symbiote is ready for testing and end user deployment. Such testing and deployment is a detailed process carried out by system administrators. As such, they are described in the following section.

#### **4.1.3: Deployment and Diagnostics**

During the development process, various prototype systems were deployed onto the campus of Youngstown State University which required periodic troubleshooting and analysis. The cumulus::CyberLab tool displays various statistical information on each node to indicate its' general health in relation to current operating system version, hard drive space available, and general operational status in relation to the cloud. When this functionality is impaired, remotely accessing the system using the SSH protocol allows for a technician to talk to the underlying operating system to further diagnose the problem. Additionally, a diagnostic USB memory stick can have important system logs transferred to it automatically when it is inserted into the cumulus::CyberLab Symbiote. Finally, if needed, the system can be physically accessed by Zethus technicians utilizing hidden ports on the motherboard for keyboard, mouse, and monitor access in cases of severe problems.

Additionally, an on-site technician can observe technical problems with the cumulus::CyberLab system as well as the instrument computers and determine which is at fault and implement basic troubleshooting techniques to try to resolve the problem. Common problems will be outlined in the cumulus::CyberLab user manual being

developed for technicians to refer to. Additionally, problems relating to the instrument computer can inaccurately be associated with cumulus::CyberLab and the on-site technician can diagnose those problems too. Zethus Employees can provide additional technical support for complicated problems. On-site technicians will resolve common problems by utilizing the previously mentioned manual and receive guidance from Zethus Employees where needed.

#### **4.1.4: System Administration Summary**

From the initial construction to the end user troubleshooting of each Symbiote, being involved in every step of the Symbiote's life consists of being a system administrator to these devices. While various different skills and procedures come into play for each specific part of the device's lifespan, the ability to carry out any and all of those specific tasks was required to allow for the deployment of this system. Such is the role of a system administrator; they are required to carry out tasks involving the deployment and continued operation of the system as a whole. To that end, system administrators are vital to the construction and any required troubleshooting of each Symbiote.

#### **4.2: Testing and Documentation**

One of the tasks carried out during the development of the cumulus::CyberLab software was testing individually and with a group. The goal of testing was to find critical bugs not found during initial development and before the first end users found them. To this end, various group members were instructed in the operation of the cumulus::CyberLab software and they additionally carried out various tasks without user documentation or instruction in order to analyze how those tasks were carried out. Such

testing provided useful feedback for Zethus which assisted them in correcting bugs and designing the software to be user friendly.

#### **4.2.1: Description of Testing**

One design goal of the program was to have a program which was simple to use. One part of making a simple interface is analyzing how people use the interface and making modifications based on common mistakes people ran into during testing. As tasks were attempted, the user was observed attempting the task to see what they naturally tried to carry out that task and future versions of the software allowed for that technique to work. While this approach is not appropriate for every computer user who will use the software, the idea was to cover some of the most common approaches to a problem and have the software respond as expected.

Testing involved a group of initial beta test users who mainly consisted of faculty members at Youngstown State University as well as a local testing team of Computer Science and Chemistry majors. The beta testers would use the software in everyday tasks. The testing team would be given tasks to carry out and do focused testing on specific subjects deemed worthy of testing by Zethus Software. Any problems encountered were entered into a bug tracking system ran by the open source eGroupware software system<sup>35</sup> and the same system was used to also note down any special directions, mistakes, or other performance metrics. Those reports would be analyzed before making another version of the cumulus::CyberLab software.

#### **4.2.2: Overview of cumulus::CyberLab Client software**

Figure 3 shows the initial main screen of the cumulus::CyberLab UserTool software. The software is designed to have various sections which are visible in tabs at

---

<sup>35</sup> (Stylite)

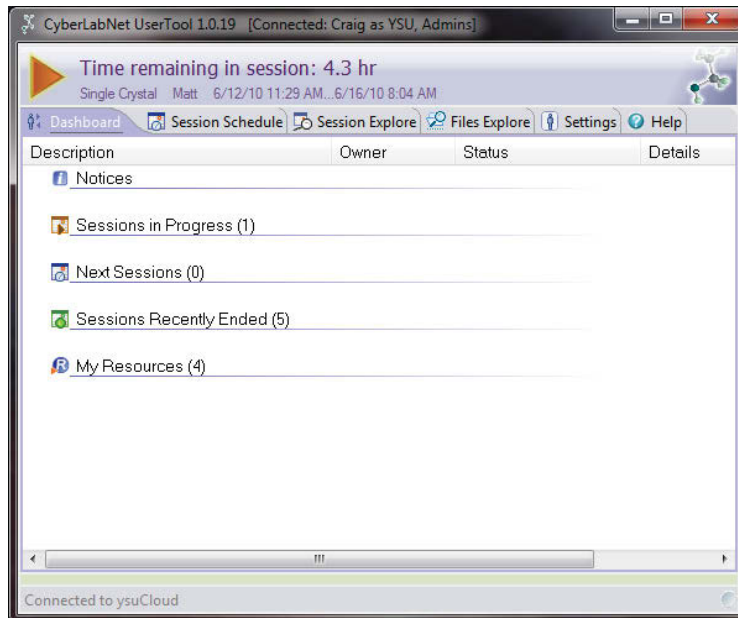


Figure 3 -- Initial main screen of cumulus::CyberLab

the top. Visiting a section by left clicking on a tab will show information relating to the section in the main area below the tabs. The interface allows for left and right clicking carrying out expected tasks and the interface automatically refreshes to show current information on screen. Important details, in this case that an active session can be joined by the user, are shown above the tabs bar. The user can exit the program by closing it like a normal application by the X icon at the top right. The application, designed for ease of use, can have user documentation made easily.

The standard operations of the program involve scheduling time on a target machine, viewing scheduling information, joining a session in progress, and retrieving files from a previous session. These tasks are accomplished by their respective tabs in the interface. To help illustrate a typical user's session in the program, we will cover how those tasks would be carried out by a typical user. The steps necessary to carry out those tasks are designed to be logical, simplistic, and efficient.

After logging into the system, you are presented with the Dashboard screen. On this screen you are able to see the current or upcoming session information in the top area of the program. You can optionally double click on the play icon in this top area to join a session in progress. By clicking on the arrow to the left of the “Sessions in Progress” and “Next Sessions” text, you can see related session information for each area. Right clicking on those entries will allow you to see detailed information about the session and alter information in the case of next sessions. You can also view and set security details which specify who can access your sessions and what access rights they would have. You can also right click on the “Sessions in Progress” text to create a session which starts immediately (ad-hoc session), useful for equipment administrators.

The “Session Schedule” tab allows you to schedule a session in the future. Drop down detail lists on the left allow you to specify criteria to select the calendar information to display in the main area to the right. Past times and dates appear shaded and unavailable and any current or future times already scheduled will also appear shaded and unavailable. You can schedule a new time on the calendar by right clicking in any spare time slot and filling out the requested information. You can also right click on a previous schedule and alter its' details from this tab.

Typically a user would schedule a session in advance, access the system on their scheduled time, and collect any files transferred in the tool after the session ends. Scheduling the equipment comes on a first serve basis with the equipment administrator having the ability to modify any sessions to override priority as needed. The system keeps track of when they accessed the equipment for auditing reasons. Additionally, security restrictions ensure that only the appropriate people access the equipment at the



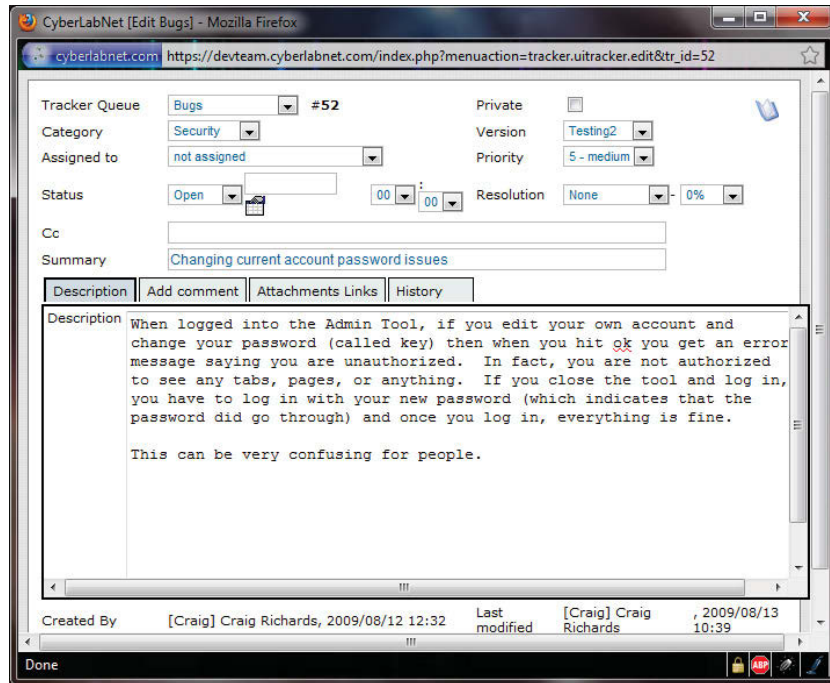


Figure 4 -- eGroupware dialog box

appointed time. Equipment administrators can be contacted via a chat program built into the program or via other means as desired or needed.

### 4.2.3: Testing Process Overview

As part of the development process for this program, testing was needed by both individuals knowledgeable in computer operations (Computer Science majors) and expected typical users of the system (Chemistry and Biology majors). Those individuals would need to carry out all feasible operations of the system, both typical and abnormal, and report information back to the software developers in respect to any problems encountered and any features requested to make the steps they carried out easier.

The testing of this program with the testing team involved various sessions where tasks were carried out by each team member. Some tasks involved each member doing identical steps and others involved members doing multiple tasks at the same time so as to simulate a multi-user situation. Each task was defined in the eGroupware system in a

specific manner so as to describe what needed to be done but with enough detail omitted to allow the users to try to creatively resolve the problem they were presented with.

When the task was completed, a comment was left in the task to give a short summary of how the task execution was handled. An example task is shown in Figure 4.

When a bug was discovered, a new bug report was filed which used a similar form to the one shown in Figure 4. Each item in eGroupware has a unique number (shown as the bold number beside Tracker Queue). This unique number is used to help link tasks and bugs together and to also help identify which tasks need executed next. The unique number was also used to help track general progress through each testing session. Priority numbers were used to group tasks together in order of difficulty to execute helping to separate single user and multiuser tasks. Finally, attachment links allowed for screenshots to be submitted to tasks or bug reports to help describe problematic situations.

A member of the testing team would either carry out tasks independently or in coordination with the team leader. In either situation, the eGroupware system would be accessed and the “Tracking System” area would be checked. The “Tracker Queue” drop down area would be set to Testing Tasks and the individual tasks would then display on the screen. A desired task would be clicked which would then cause a new window to display (Figure 4) with details relating to the task. With the directions read, the testing member would carry out the task as instructed filling in necessary omitted details with logical steps that they feel would accomplish the task.

Testing results would be reported in two different areas depending on what results needed reported. The general task report would always be submitted by adding a

comment to the testing task by clicking the “Add Comment” tab in the specific task. These would indicate that an individual completed the task as well as how the task generally went indicating if feature requests or bugs were being filed. Additional comments can be submitted as needed to provide additional details on the task at hand. Software developers would later read these comments and would retrieve the feature requests and bugs as needed.

For Feature Requests and Bugs, the related Tracker Queue would be selected on the main browser page and the user then had the ability to create a new entry by clicking on “Add” near the Tracker Queue selection area. Related fields would be filled in, attachments would be uploaded as needed by visiting the Attachment Links area, and then the new entry would be submitted generating a new identifier number which can be used to uniquely identify the Feature Request or Bug. Additionally, related tasks would be provided where applicable. Fixes or solutions to those entries would be commented on by submitting new comments to the entries themselves using a technique similar to the one used in Testing Tasks.

User input is highly desired. The methods used to execute a task as well as user frustrations or confusions can be seen as a usability bug. Since most users have over time become desensitized to user interface problems, observation of a user carrying out a task is often needed so that a user can be questioned about how they initially carried out the task and what seemed natural to them. Such discussion then leads to feature requests filed in the eGroupware system of things that can be implemented to help a new user feel comfortable with the system. If the end user does not read the software documentation, a

possible scenario, then they should still be able to use the software in a relatively efficient manner.

#### **4.2.4: Software Documentation System**

Software documentation, while not completed, was in development during the testing phase. As steps were carried out during testing, the steps taken to carry out the various tasks were recorded in an online documentation system by the various testing team members. The end goal was to have this documentation available on a website for any users of the software to access when needed. While the documentation remains incomplete, the mechanism for storing such documentation as well as generating such documentation is valid.

Testing team members accessed a Wikipedia like software system called DokuWiki to store documentation pages. DokuWiki was selected since it creates internal program files which can easily be modified to create a physical manual.<sup>36</sup> Each page utilized a simplistic Wiki syntax with a built-in editor which easily provided access to various formatting features. As new pages were needed, they were created by first creating chapter links from the main page, following those links, and then creating the page as needed. Revision control allowed for mistakes to be undone and revisions to be compared. Each testing team member had an individual login so that their activities could be monitored. Their work was critiqued as needed.

Testing tasks were also filed in the eGroupware system under the Tracker Queue “Wiki”. In a similar fashion to the Testing Tasks, each Wiki task would have specific goals and objectives with completion being indicated by a comment added to the task. While the documentation system could be expanded as needed without requiring a task

---

<sup>36</sup> (DokuWiki Community)

for each expansion, the tasks allowed for directed expansion of the documentation system as desired. Each new Wiki task would indicate areas which needed attention.

The documentation system itself stored entries as plain text files which can easily be formatted in various ways to allow for such future options as printing user manuals. The documentation system could also be configured for deployment use by disabling all revisions and user access systems making a static documentation website suitable for end user access. In this phase, a development documentation system would be maintained so that the documentation can be updated as needed as transferred to the production system. Future upgrade possibilities remain possible because the plain text pages could be converted into a format that can be read by any other end system.

Future work in this area would require an individual or group dedicated to the generation of user documentation. The group previously generating the documentation was responsible for general testing tasks and therefore would only generate documentation when no tasks were pending. This led to the documentation being incomplete and missing technical details which are not immediately apparent to the end user as well as missing documentation on the administration system which the end user would generally not use but the instrument administrator would likely use. When documentation resumes, the technical system required is already in place.

#### **4.2.5: Testing and Documentation Summary**

The software testing carried out by the testing team sought to point out flaws and problems that either should be caught before a beta testing user encountered the problem or to help identify usability issues that would likely not be reported by a beta tester or end user. Consequently, having Chemistry students as part of the team helped expand the

testing to individuals without a strong computer background. This testing assisted the software into developing into the version it is now and testing is sure to be a part of future software developments with cumulus::CyberLab. Additionally, the documentation system and already created documentation provides a foundation for future work in creating software documentation required for this project. Testing and documentation are required parts of any software program or project and this project is no exception.

## **Chapter 5: Conclusions**

The cumulus::CyberLab project developed a unified system allowing for remote access, webcam visibility, encrypted file transfer, user account security, and high reliability to the computers operating the millions of dollars of scientific equipment at Youngstown State University. By providing such a system, individuals from across the nation and the world can access YSU's scientific equipment in a safe and secure manner. Such a project was made possible by the various computer technologies developed and invented thus far.

Cyber technology, remote access, cloud computing, and computer controlled scientific equipment are all brought together to provide services across the internet from our university to interested professionals in the world. By bringing together these technologies into a unified package, Zethus Software was able to provide the software program that Youngstown State University required.

Zethus Software is a Youngstown, OH based company developing cutting edge cloud computing software. In collaboration with Youngstown State University and with the assistance of the Youngstown Business Incubator, this company was able to create the latest in cloud computing software and assist the Youngstown region in entering the technological future. Their work on this project has made this project a reality.

System administration tasks from hardware assembly to testing and documentation were carried out in a methodical and efficient manner allowing for the development of Symbiotes running cumulus::CyberLab software which enables the cloud to provide the required services. By working with various Youngstown State University students, Zethus software was able to save time and money and develop the desired

software which in turn helps Youngstown State University by providing software which assists YSU's educational goals.

In conclusion, the cumulus::CyberLab project provides beneficial features to not only our university but also to other interested parties. This project expands the realm of cloud computing so that it provides remote accessibility and security to millions of dollars of scientific equipment. Therefore, scientists can continue to run experiments even when not on campus and become more efficient at their jobs. Overall, this project has fulfilled the motivation of this university and contributed a unique software program to the world.



## Works Cited

- Bruker AXS Inc. About us. n.d. 26 October 2010 <[http://www.bruker-axs.de/about\\_us.html](http://www.bruker-axs.de/about_us.html)>.
- . "Diffraction Solutions D8 Advance." 2006. Bruker AXS. 26 October 2010 <[http://www.bruker-axs.com/uploads/tx\\_linkselectorforpdfpool/01626\\_ND\\_D8\\_ADVANCE\\_e\\_72dpi\\_05.pdf](http://www.bruker-axs.com/uploads/tx_linkselectorforpdfpool/01626_ND_D8_ADVANCE_e_72dpi_05.pdf)>.
- Buyya, Rajkumar, Chee Shin Yeo and Srikumar Venugopal. "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities." n.d. 3 November 2010 <[http://www.cloudbus.org/~raj/papers/hpcc2008\\_keynote\\_cloudcomputing.pdf](http://www.cloudbus.org/~raj/papers/hpcc2008_keynote_cloudcomputing.pdf)>.
- Cisco. WebEx: Online meetings and video conferencing. n.d. 24 November 2010 <<http://www.webex.com/product-overview/support-center/remote-access.html>>.
- Citrix Online, LLC. Business Remote Access | GoToMyPC. n.d. 24 November 2010 <[http://www.gotomypc.com/remote\\_access/business\\_remote\\_access](http://www.gotomypc.com/remote_access/business_remote_access)>.
- . PC Remote Access | GoToMyPC. n.d. 24 November 2010 <[http://www.gotomypc.com/remote\\_access/pc\\_remote\\_access](http://www.gotomypc.com/remote_access/pc_remote_access)>.
- . Remote File Access | GoToMyPC. n.d. 24 November 2010 <[http://www.gotomypc.com/remote\\_access/remote\\_file\\_access](http://www.gotomypc.com/remote_access/remote_file_access)>.
- Daley, Jason. "Where to Be an Entrepreneur." August 2009. Entrepreneur Magazine. Entrepreneur Media, Inc. 26 October 2010 <<http://www.entrepreneur.com/article/202666-9>>.
- Department of Geosciences at Indiana University-Purdue University Fort Wayne. "X-Ray Diffraction." 11 November 2003. 3 November 2010 <<http://www.geosci.ipfw.edu/XRD/techniqueinformation.html>>.
- DokuWiki Community. dokuwiki [DokuWiki]. n.d. 26 October 2010 <<http://www.dokuwiki.org/>>.
- Hudak, David E and Prasad Calyam. "Remote desktop control and large-scale data transfer and management: Survey and Experiments." DRAFT OSC Project Report for the YSU Metrology Project. 16 September 2010.
- Kurose, James F. and Keith W. Ross. Computer Networking A Top-Down Approach. Ed. Michael Hirsch, Lindsey Triebel and Jeffrey Holcomb. 4th. Pearson Education, Inc., 2008.
- Markoff, John. "An Internet Pioneer Ponders the Next Revolution." 20 December 1999. The New York Times. 26 October 2010 <<http://partners.nytimes.com/library/tech/99/12/biztech/articles/122099outlook-bobb.html>>.
- Microsoft Corporation. Remote Desktop Protocol (Windows). 28 October 2010. 24 November 2010 <<http://msdn.microsoft.com/en-us/library/aa383015%28VS.85%29.aspx>>.
- Miniwatts Marketing Group. World Internet Usage Statistics News and World Population Stats. n.d. 26 October 2010 <<http://www.internetworldstats.com/stats.htm>>.
- Stylite. EGroupware - Online Groupware, CRM: Home. n.d. 23 November 2010 <<http://www.egroupware.org/>>.

- Symantec Corporation. Remote VPN, Remote Server Access Software, Remote PC Manager | Symantec pcAnywhere. n.d. 24 November 2010 <<http://www.symantec.com/business/pcanywhere>>.
- . "Symantec pcAnywhere™ 12.5." October 2008. 24 November 2010 <[http://eval.symantec.com/mktginfo/enterprise/fact\\_sheets/b-datasheet\\_pcanywhere\\_12\\_5\\_10\\_2008.pdf](http://eval.symantec.com/mktginfo/enterprise/fact_sheets/b-datasheet_pcanywhere_12_5_10_2008.pdf)>.
- The Vindicator. "YSU enrollment is at highest point since 1990." 9 September 2010. Vindy.com. 26 October 2010 <<http://www.vindy.com/news/2010/sep/08/ysu-enrollment-is-at-highest-point-since/>>.
- Ubuntu documentation team. Automating the installation using preseeding. n.d. 26 October 2010 <<https://help.ubuntu.com/10.04/installation-guide/i386/preseed-intro.html>>.
- Youngstown Business Incubator. About Us. n.d. 26 October 2010 <<http://www.ybi.org/about>>.
- . History. n.d. 26 October 2010 <<http://www.ybi.org/about/history>>.
- Youngstown State University Department of Chemistry. "Instrumentation Grants from YSU Science Faculty." n.d. YSU Structure & Chemical Instrumentation Facility. 26 October 2010 <<http://www.as.ysu.edu/~scic/Grants.html>>.
- Zethus Software LLC. "Andrew Reinhardt, V.P. Engineering & co-founder." n.d. Zethus Software - c u m u l u s :: C y b e r L a b. 26 October 2010 <<http://www.zethussoftware.com/company/team/andrew-reinhardt-v-p-engineering-co-founder/>>.
- . "Bruker AXS and Zethus Software Signs Worldwide Strategic Partnership." 26 March 2010. Zethus Software - c u m u l u s :: C y b e r L a b. 26 October 2010 <<http://www.zethussoftware.com/news-events/bruker-axs-and-zethus-partnership/>>.
- . "CyberLabNet Development Phases." 2008. CyberLabNet - Development Team. 26 October 2010 <<http://www.cyberlabnet.com/Docs/CyberLabNet-DevelopmentPhases.pdf>>.
- . "CyberLabNet Development Principles." 2008. CyberLabNet - Development Team. 26 October 2010 <<http://www.cyberlabnet.com/Docs/CyberLabNet-DevelopmentPrinciples.pdf>>.
- . "CyberLabNet Overview." 2008. CyberLabNet - Development Team. 26 October 2010 <<http://www.cyberlabnet.com/Docs/CyberLabNet-Overview.pdf>>.
- . "Eric Parker, CTO & co-founder." n.d. Zethus Software - c u m u l u s :: C y b e r L a b. 26 October 2010 <<http://www.zethussoftware.com/company/team/eric-parker/>>.
- . "Zethus Software Inks Deal with Youngstown State University." 26 February 2008. Zethus Software - c u m u l u s :: C y b e r L a b. 26 October 2010 <<http://www.zethussoftware.com/news-events/zethus-software-inks-deal-with-youngstown-state-university/>>.
- . "Zethus Software Overview." 2007. CyberLabNet - Development Team. 26 October 2010 <<http://www.cyberlabnet.com/Docs/ZethusSoftware-Overview.pdf>>.