Analysis and Evaluation of an Integrated Web Services Framework

THESIS

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Da Cai

Graduate Program in Computer Science and Engineering

The Ohio State University

2012

Thesis Committee:

Dr. Rajiv Ramnath, Advisor

Dr. Jayshree Ramanathan

Dr. Thomas Bitterman
Copyright by

Da Cai

2012
Abstract

Along with increasing demands for using software to deal with model simulation and physical predictor in industry, professional people prefer to choose an effective and efficient way to achieve their goals, using professional software applications. It is necessary to provide a simple way for users to access on-demand software services without having to configure and maintain it by themselves. In this thesis, I propose a web-based framework to integrate multiple software applications for users to access their required software services through E-Commerce, and I show the benefits of this service-oriented architecture. We use two typical “Simulation Software as a Service” (SMaaS) implementations to illustrate how this integrated framework can be implemented, and make surveys for users to evaluate the user experience.
Dedication

This document is dedicated to my family.
Acknowledgments

I would like to express my sincere gratitude to my advisor, Dr. Rajiv Ramnath. His knowledge and critical thinking has been very valuable. His encouragement and help for me has made this thesis possible.

I would also like to thank Dr. Jayashree Ramanathan and Dr. Thomas Bitterman for their continued support, insights and suggestions that helped shape my research.

I also really appreciate the support I got from my parents, and my close circle of friends who kept encouraging me constantly and kept me on track to complete this successfully.
Vita

June 2010 ...................................................... B.S. Digital Media Technology, Harbin Normal University

September 2010 - present.............................. Graduate Student Department of Computer Science and Engineering The Ohio State University

Fields of Study

Major Field: Computer Science and Engineering
# Table of Contents

Abstract ......................................................................................................................... ii

Acknowledgments........................................................................................................ iv

Vita ................................................................................................................................. v

Chapter 1 .................................................................................................................... 1

Introduction .................................................................................................................. 1

  1.1 The Problem ........................................................................................................... 3

  1.2 Solution Approach ............................................................................................... 5

Chapter 2 .................................................................................................................... 8

Related Work ............................................................................................................... 8

  2.1 Collaborative Web Service ................................................................................... 10

  2.2 Web Service Configuration and Customization ..................................................... 11

  2.3 SaaS Implementation .......................................................................................... 12

  2.4 Web Portal Evaluation ....................................................................................... 13

Chapter 3 .................................................................................................................... 15

The Framework .......................................................................................................... 15

  3.1 Web Services Portal ............................................................................................ 19
3.2 On-demand License Service and Accounting Billing ........................................ 20
3.3 Application Software Deployment ..................................................................... 21
3.4 High-Performance Computing and Virtualized Resources ............................... 22
Chapter 4 .................................................................................................................. 24
Case Study .................................................................................................................. 24
4.1 Polymer Portal .................................................................................................. 24
   4.1.1 User Experience ......................................................................................... 25
   4.1.2 Architecture ............................................................................................. 27
   4.1.3 Implementation ......................................................................................... 29
4.2 Manifold Flow Predictor ................................................................................... 32
   4.2.1 User Experience ......................................................................................... 34
   4.2.2 Architecture and Implementation ............................................................ 35
Chapter 5 .................................................................................................................. 38
Evaluation .................................................................................................................. 38
5.1 Surveys ............................................................................................................... 38
   5.1.1 Polymer Portal User Survey ...................................................................... 38
   5.1.2 PolymerOhio Staff Survey ......................................................................... 39
   5.1.3 Manifold Flow Predictor User Survey ..................................................... 40
5.2 Analysis of User Surveys .................................................................................. 40
5.2.1 Cost ........................................................................................................ 41
5.2.2 Simplicity ................................................................................................ 42
5.2.3 Sustainability .......................................................................................... 47
5.2.4 User Satisfaction .................................................................................... 48

Chapter 6 ......................................................................................................... 50
Conclusions ...................................................................................................... 50
References ....................................................................................................... 51
Appendix ........................................................................................................... 55

A.1 Polymer Portal User Response ................................................................. 55
A.2 PolymerOhio Staff Response ................................................................. 57
A.3 Manifold Flow User Response ................................................................. 58
List of Tables

Table 1. Industrial and Academic Use................................................................. 4
Table 2. Web Service Technology Stack............................................................. 8
List of Figures

Figure 1. Architecture of SMaaS for Cloud Computing ........................................ 5
Figure 2. Web Service Models in a Service-Oriented Architecture ............................... 9
Figure 3. Portal and Catalog Layer ............................................................................. 16
Figure 4. Cloud Services .............................................................................................. 18
Figure 5. Logical Architecture ...................................................................................... 27
Figure 6. Polymer Portal Component Architecture ...................................................... 30
Figure 7. Web Portal Inner Architecture ...................................................................... 31
Figure 8. Example of an OpenFoam simulation result ............................................... 33
Figure 9. Manifold Flow Predictor Work Flow ............................................................. 34
Figure 10. Manifold Flow Predictor Architecture ......................................................... 36
Figure 11. Price of Simulation Services on Polymer Portal .......................................... 42
Figure 12. Three Steps to Create a Job of Manifold Flow Predictor .............................. 46
Chapter 1

Introduction

The term “software” was first used by John W. Tukey in 1958 [31], and in subsequent decades, application software has been developed and gradually used in many areas. The growth of software has caused significant changes to the human life style and business models. In daily life, people deal with complicated financial forms using automated application. They also share videos, photographs, and articles through the Internet with families and friends. On the other hand, in business, software can help companies to manage their private data, make a financial prediction, and simulate a process or a model. Web service is increasingly gaining widespread adoption in modern software applications [1]. Many web services have been published by Amazon, Google, and Microsoft. Also, numerous business applications are delivered as web services, such as Enterprise Resource Planning (ERP) and Customer Relationship Management (CRM). Web service is versatile by design, and it can be accessed via a web-based client interface. Furthermore, one web service can be utilized by several clients so that the cost is lowered.

Along with increasing demands for using software to deal with model simulation and physical predictor in the industrial area, professional people prefer to choose an
effective and efficient way to achieve their goals, using professional software applications. “The high cost of creating and maintaining software and hardware infrastructures for delivering services to businesses has led to a notable trend toward the use of third-party services which provide computation power, and data storage space to clients” [5].

Software-as-a-Service (SaaS), a software delivery model in which software and associated data can be accessed by users using a thin client via a web browser [30], offers important benefits, such as lower cost, remote access, rapid deployment, and leveraging cloud resources. The customers can subscribe to a short-term license, rather than a perpetual license to use a software application, so that they save a lot of up-front costs, which can be used to improve business. The service price is also based on “usage parameters,” such as user groups and the period of using the application. The software application’s environment is configured and maintained by a SaaS vendor so that the deployment is very quick. Haiqi Liang, et al. [1] proposes an approach to customize web service through collaboration between service provider and consumer. They state, “As different consumers usually have special requirements for the business application in terms of user interface, business process and data, it is critical for the SaaS services to provide service consumers easy personalization and customization capabilities according to their own requirements.” [1] It is necessary to build a framework to extend web service according to multiple user requirements. This has led to this research on SaaS implementation, which can address both industrial and education needs.
1.1 The Problem

Simulation, through the use of simulation software, is the science of creating statistically accurate models to represent the behavior of real life systems in order to subject them to predictive experimentation. [37] The experiments or scenarios can then enable “what if?” questions to be answered without risk or disturbance to the real life system. [37] In industry, companies use simulation software, which imitates physical phenomena with a set of mathematical formulas [24], to improve product quality and reduce cost. Especially with companies from plastics, rubber and advanced material segments, it is necessary to find a way to simulate the process or predict the result in order to increase productivity and save up-front cost. Simulation software service is a good solution for businesses. For example, General Motors uses parallel computing to simulate the crash testing of automobiles. It claims that it can reduce the number of full-size crash vehicle tests by more than 85 percent, at a cost savings of $500,000 per test [34]. Similarly, supercomputing simulations have reduced the cost that Goodyear spends on physical tire prototypes from 40 percent to 15 percent, while dramatically reducing the time to get a newly designed tire on the market [35]. In education educators, students, and researchers must obtain access to the individual machine that can host the simulation software to do their experiments, research, or product development. Both industrial and academic users are subject matter experts and not computer experts. They need a system that is simple to use, but gives them access to a wide variety of software. Also, it is necessary to provide training courses for simulation software users to learn how to use the software.
<table>
<thead>
<tr>
<th></th>
<th><strong>Industrial Use</strong></th>
<th><strong>Academic Use</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Simulation</strong></td>
<td>Product development testing</td>
<td>Class project research</td>
</tr>
<tr>
<td><strong>Visualization</strong></td>
<td>Visualization and Analyze generated results</td>
<td></td>
</tr>
<tr>
<td><strong>Training</strong></td>
<td>Simulation-related software</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Industrial and Academic Use

Both business and education need this kind of process, which is time-consuming and bewildering for users since they have to hold various authentication certificates for each machine that hosts the service. In particular, many industrial processes are amenable to simulation, and simulating these processes allows for optimization of time, cost, energy, quality, and some other process parameters. However, simulation software is very expensive, hard to use, and needs to be bundled with local supporting hardware. The cost is fairly high, and many potential users cannot justify paying such a high up-front cost for such an uncertain payoff, so users would like to use a relatively cheap way to address the simulation needs. In addition, users might have additional functional requirements or need other kinds of software services, such as wanting to visualize the simulation results after the simulating process. There should be a way to bundle multiple software services together for users to select.

Through this thesis, we aim to develop a web-based framework of measures for delivering multiple software services in order to satisfy users’ requirements. We also intend to discuss the results of a survey we used to evaluate user experience of this framework.
1.2 Solution Approach

Software as a Service (SaaS) is gaining momentum with an increased number of vendors and recent successes from leading players in the market [14]. To solve the above problems, we propose Simulation as a Service (SMaaS) as an extension of SaaS, it provides a combination of general purpose and special usage tools across different platforms.

Figure 1. Architecture of SMaaS for Cloud Computing [10]
SMaaS provides an integrated suite of tools for end-to-end simulation support across multiple platforms, and is built using primarily Common-Off-The-Shelf (COTS) components. Its characteristics include:

1) Long-running simulations require both interactive sessions and batch jobs
2) Expensive third-party solvers required
3) Large data sets required
4) High Performance Computing (HPC) processing required

There can be many different groups involved in a SMaaS system. For example, there are educators who wish to use the system for instruction, students who will learn how to use the tools hosted by the system, software vendors who provide the tools, and the resource provider, who needs to provision and maintain the system. Each group has different and often conflicting goals, and they need to be able to pursue these goals in a flexible and secure manner. SMaaS aims to serve these diverse user communities by providing a scalable, secure, and expandable architecture.

We intend to design a user-friendly web-based platform to deliver software services. The platform will enable users to purchase short-period software licenses according to their actual requirement rather than a perpetual license, which is fairly high in cost. An E-Commerce Catalog will be used to customize user requirements, which means we will provide a bunch software services. The users will be able to select the services they want to use. Furthermore, users will be able to use a virtual machine that is configured and maintained by the service provider to access the software, so the users will not need to consider how to configure and maintain the hardware. We also will
design the features of common entry point and single sign-on for users to log on so that disparate subsystems can be tied together into a cohesive whole that will make sense to the user. Once the user is signed in, they will be able to use any part of the platform for which they are authorized without re-authenticating. All of these features will be integrated into one web-based framework.
Chapter 2

Related Work

Web Services [32] software is built on standard protocols and technologies such as HTTP [46], XML [47], SOAP [33], WSDL [48] and UDDI [49]. Web 2.0 is a revolution [32], which moves from Simple Object Access Protocol (SOAP) towards Representational State Transfer (REST) based communications. Please see the table below.

<table>
<thead>
<tr>
<th>Service Publication and Discovery (UDDI)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Description (WSDL)</td>
</tr>
<tr>
<td>XML – Based Message (XML)</td>
</tr>
<tr>
<td>Service Public and Discovery (TCP/IP, HTTP, FTP)</td>
</tr>
</tbody>
</table>

Table 2. Web Service Technology Stack [3]

People can use different Application Programming Interfaces (APIs) to integrate multiple web services as a new web application. “Web Services define a new paradigm in present-day collaborative application development in enterprises, and span systems,
networks, and organizations.” [2] The following diagram is a completely service-oriented model.

![Diagram of Service-Oriented Architecture]

Figure 2. Web Service Models in a Service-Oriented Architecture [32]

The most common understanding of Cloud Computing is service-oriented, from hardware to application level. In the above diagram, Platform as a Service (PaaS), Software as a Service (SaaS), and Data as a Service (DaaS) are the concrete implementations of Cloud Computing. “Cloud Computing receives significant attention as it can enable the rapid delivery of computing resources as utilities in a dynamic, scalable, and virtualized manner.” [25] There are many cloud products in both industry and research communities, such as Amazon Elastic Compute Cloud (EC2), Google App Engine (GAE), Microsoft Azure, Salesforce.com’s Force.com, VMware, Eucalyptus, and Citrix. Although each of them has its own specialties, they improve IT efficiency
through their scalability, vast computing resources, and the infrastructure support for multiple tenants. People can utilize their Cloud platform to run their own jobs, and there is no need to consider computing resources or deploy running environments, maintenance, and job scheduling.

2.1 Collaborative Web Service

Some research has supported web service collaboration. Zhengxiong Hou et al [10] propose a web-based framework for delivering software as a service based on high-performance computing resources. It provides a web portal for users to access the application software under a pay-for-use mode. This solves the problems of traditional software, which is not service-oriented and is limited by software licenses. This E-Commerce feature is similar to my integrated framework, which lets users select their preferred type of software service license by different cost. Irfan Awan et al [3] propose a new web services-based model that enables different precedence for collaborative applications. Most of the current approaches equally treat all the activities of collaborative applications; however, they proposed a three-layer architecture that processes various treatments for the collaborative applications. My integrated framework does not implement the feature of “different precedence to different services,” because all of the services we provide within the framework are independent. There is no need to implement the strategy of different treatments. The collaboration of web services cannot obscure the integration challenges faced by enterprise infrastructure. Firewall/NAT [55] traversal and security issues often pose a bottleneck as enterprises may not be comfortable running mission critical applications outside the corporate firewall. In our
framework, our services are building on the Linux system as a cluster rather than distributed systems. The issue of collaboration with outside resources, however, will be taken into consideration in the future, so this related work is worth researching. Feng Liu et al [22] present a Proxy-based firewall/NAT traversal solution for web service integration. It allows the web service to integrate with an on-premise application without firewall configuration. It is important to manage a web service platform since more and more of this kind of platform has been widely used. Jianzhong Li et al [28] introduce a component-based approach for the implementation of the integrated management framework for web service platforms. The management framework offers a unified entry point, which is convenient for the administrator to access the management tools of the web service platforms. The feature of “unified entry point” is similar to our framework, which also provides a single entry point for users to log in so that users can seamlessly migrate among different services. Also, our framework is built by different components, which are responsible for different features.

2.2 Web Service Configuration and Customization

Haiqi Liang et al [1] introduce an approach to customize web service through collaboration between a service provider and a consumer in a programmatical way, which is customizing web service with WS-Policy. WS-Policy [38] provides a flexible and extensible grammar for expressing capabilities, requirements, and general characteristics of entities in an XML web service-based system. It defines a framework and a model to express these properties as policies. In contrast, within our integrated framework, users can select the preferred services into their own catalogs on the web page, which is
friendly to users. Although a SaaS application can be accessed by a large amount of clients with multi-tenancy capabilities, many clients still ask for function variants according to their unique business needs. Wei Sun et al [11] propose a competency model and a methodology framework in order to help SaaS vendors plan and evaluate the capabilities and strategies for service configuration and customization. The competency model is described by a range, from “Entry,” “Aware,” and “Capable” levels to “Mature” and “World Class.” Additionally, different levels can be used to identify the configuration and customization through benchmarking with the market’s competency level. [11] This related work is helpful to our framework, since we have different kinds of users, ranging from those in academia and industry, to utilize our high-performance computing service. They have different demands for software services and computing resources, so this kind of evaluation model is useful when we implement our framework.

2.3 SaaS Implementation

Jyoti Namjoshi et al [6] present a cloud-based travel reservation application based on service-oriented architecture. Their solution provides an efficient reservation process, easy-to-use anytime access from multiple locations and from heterogeneous technology environments [6]. This is a typical service-oriented application; however, its functionality cannot satisfy customers’ increasing requirements. Our framework has scalability, which makes it easier for us to extend our services and lets our user customize their catalogs. SaaS users only need to purchase a subscription from a service provider. This contrasts the traditional way, where users buy a perpetual or long-term license. V. Choudhary [7] shows that this property of SaaS licensing model leads to greater
investment and a higher quality of software. We implement this property in our integrated framework, and users benefit from its simple-to-use and cost-saving strategy. In order to make simulation software accessible by researchers and field experts all over the Internet, rather than only on a single machine, Song Guo et al [8] provides specification for simulation software as a service and service-oriented architecture experiment in order to support automatic deployment for simulation services. In contrast, we pre-configured the services provided in our framework, and there is no specification for different software services.

2.4 Web Portal Evaluation

Even as more and more web portals are developed to support the various needs of users, users exhibit a desire for a single site that can satisfy multiple user requirements rather than using a variety of sites for specific requirements. Mario Christ et al. [57] measured Web portal utilization of individual features in the HomeNet [57] project, and developed demographic profiles of groups with different portal utilization levels. In our project, we can leverage this kind of methodology to measure the utilization of simulation software services provided by Polymer Portal for different kinds of users such as academic and industrial users so as to make a rational allocation of computing resources and virtual machine resources. Saeed Nourizadeh Azar et al. [58] propose several criteria to evaluate web portals based their categories (General Portals, B2B Portals, Enterprise Portals, Government Portals and so on). [58] also lists desirable portal characteristics, namely, easy to use, intuitive classification and searching, collaborative information sharing, universal connectivity to information resources, dynamic access to information
resources, intelligent routing and so forth. We plan to utilize these criteria to evaluate features of the Polymer Portal and Manifold Flow Predictor, such as the customization and personalization of the E-Commerce Catalog, flexible permission granting for authentication and authorization, and the server-based architecture for the high performance computing resources.
Chapter 3
The Framework

To expose application software as services, the main components include service encapsulation, description, and deployment for application software, service registration and discovery, service negotiation and selection, service delivery and usage, service composition, and on-demand license service and accounting [10]. Below is the figure of the component architecture.
At the top usage layer, the web platform provides a friendly user interface for potential users. This layer sets the portal developer free from worrying about the technical terms or the specific underlying systems. Therefore, developers can mainly focus on input services and output services. At the second layer, we create an E-Commerce Catalog service that is the inventory of simulation services and training materials from Resource Providers. The user can browse and select available services and course information from the catalog. In the next layer, independent software vendors
are responsible for providing application software, cost, and licenses to our service provider. E-Commerce Service provides billing information to users, simulation authorization and payments to resource providers, and license usage reports and payments to software vendors. Moreover, resource reusability also can be achieved by using remote connection protocol to access an external ready-to-use resource. With respect to increasing usability, since multiple services can be accessed through a single website, the user does not need to know the actual link for each service. Instead, the web platform will redirect the user to each service as long as the user gets authenticated and authorized by the system. The bottom layer is the infrastructure layer, which deploys the virtual machine pool, active directory database, and the application software. A cloud, which is a typical high-performance computing environment running Linux on a cluster in batch mode to support a wide variety of operating systems (all the different versions of Windows, the Linux and Unix variants) without disrupting normal operations. It became possible for us to support one user running a simulation on Linux-based software while another ran a Windows-based visualization program at the same time. Each virtual machine runs on a high-performance node, which guarantees software performance. In addition, we accommodate transient spikes in usage (such as a large class) without letting valuable resources stand idle during periods of low use. The active directory is used to authentication and authorization, which means that it is not only a security node but also a service management node for each user. Below is a figure illustrating workflow.
Figure 4. Cloud Services [42]

As Figure 4 illustrates, there are five main components in our service-oriented platform: E-Catalog Service, E-Commerce Service, Simulation Service Resource Provider, Commercial License Broker, and Independent Software Vendors. E-Catalog Service enables a user to browse and select the available software service, and create his or her own user profile. All of the available software services are listed in the inventory. Once the user selects a kind of software service, he or she can purchase the corresponding software license by credit card or debit card through another component, E-Commerce
Service. There are various kinds of software licenses to be provided and each kind of license has a different price, so users can choose one of them according to their actual requirements. Through this component, the user can get the authorization of the required service. The Simulation Service Resource Provider is the agent that provides simulation software application for users, so that the user can access the simulation service at this component. Behind this component is the actual software vendor. The Commercial License Broker can be regarded as a bridge between the E-Commerce Service component and the Independent Software Vendors component. The broker is used to require application license from software vendors for users and give authorization to the Simulation Service Resource Provider for different services.

3.1 Web Services Portal

Business users want adaptive, flexible, and cost-effective solutions. [12] The mode for using most high-performance computing software is based on command line options or graphic user interface, and the client-server model is usually adopted as the architecture [10]. However, this kind of architecture is not service-oriented. It is necessary to build a web portal that is easily accessed by the users to browse, purchase, and access available software services, as long as they have an Internet connection. The portal can be regarded as a platform that integrates the available software services for users. In addition, through this portal, users not only can select the service they prefer to use, but also customize their service catalog in order to simplify the workflow for later usage.
Since long-running simulations on the system require both interactive sessions and batch jobs, simplicity in user experience is very important so that the portal can be used as a common entry point for log on, rather than having to remember individual routes for each service. The common entry point allows them to access services faster and easier. In addition, users might transfer from one service to another. It is useless and boring if every transfer needs authentication, so a single sign-on (SSO) mechanism [42] should be implemented within the portal. SSO is a property of access control of multiple related but independent components in the portal. With this property, a user logs in once and gains access to all systems without being prompted to log in again at each of them.

To implement this property, our system has to internally translate and store different credentials compared to what is used for initial authentication. The Microsoft Windows Active Directory [41] enables companies and other organizations to store information such as files and applications on one or more networks as well as provides facilities for administratively determining user access rights. Active Directory SSO means that users have only one user name and password to remember. When a user logs on to any network, the Active Directory SSO automatically logs the user on to every authorized resource. The advantages to SSO service are two-fold.

3.2 On-demand License Service and Accounting Billing

The price of simulation software and supporting hardware is fairly high, especially for small and medium-sized companies who might not be able to afford this uncertain up-front cost. In this Simulation as a Service (SMaaS) framework, we implement a “pay-for-use” mode, which means that the users can access the service as
long as they purchase the valid service license. There should be various licenses for users to choose, and the feature of different types of licenses can be implemented by the Active Directory. Unlike a perpetual or one-year license, SMaaS users can subscribe to a short-period license, like a one-day, one-week, and four-week license, all of which are much cheaper than the original license. Every different kind of software service will be regarded as single service group. We use group policies, which are Active Directory Object formally named as Group Policy Object, to assign different access permission for each user. Users can select the software from the service list and move to their own catalog. After the single sign-on, as described above, the user will be redirected to the catalog. In the catalog, we added E-Commerce capability, which means that users can purchase various software licenses with credit card or debit card. After the purchase process is completed, the identifier of the user with the license information will be added to the software service group in the Active Directory. At the same time, the user who bought this service license would be notified through email.

3.3 Application Software Deployment

The application software is deployed on the virtual machine, which is maintained by the service provider, so there is no need for users to care about where the software is running or how the software runs. Users can use a thin client such as web browser VNC client and VMview to remotely access the software service. Resource utilization of the simulation software may be uneven in academic and industrial usage. For instance, resource usage in an educational setting tends to be lower than industrial usage; however, educational usage might be relatively high during class or the day before a due date.
Therefore, it is necessary to take that into consideration since the computing for assigning to each virtual machine resources are limited. The elementary ideas are to assign various performance configurations to different virtual machines of usage, which means to offer industrial users’ virtual machines a relatively higher processing capacity while offering academic users lower computing resources, and to make the service license as short as possible, like the one-day license described above, to not only save the user money but to also save computing resources for the service provider. Most of the simulation process can be done within a short period of time, so that the user can choose the shortest possible license period.

3.4 High-Performance Computing and Virtualized Resources

There is a trend for scientists and engineers to conduct computation-intensive or data-intensive simulations or analytics on large-scale computation systems with multiple processors [56]. Supercomputers are used for high calculation-intensive tasks such as problems in quantum physics, weather forecasting, climate research, oil and gas exploration, and molecular modeling [39]. Simulation software is a kind of computational resource-consuming (e.g., computer nodes, storage, etc) software application, so it is important to run the software and process the jobs on a high-speed supercomputer. For the sake of improving computation speed, we deployed Virtual Machines on Red Enterprise Linux as a cluster; each virtual machine runs on a high-performance computing node, which can satisfy the simulation software’s requirement. The simulation software is pre-configured by the service provider on the virtual machine, so there is no need for users to configure it. The SMaaS framework would support
multiple software services, some of them are Windows-based applications, some are Linux-based, and some are web-based, so it is necessary to make the service a cross-platform. Users can use Remote Desktop Protocol (RDP) to access Windows VM and Virtual Network Computing (VNC) session or Secure Shell (SSH) to access Linux VM. Therefore, providing high-performance computing and virtualized resources is not only about how to allocate computational resources or the storage of high-speed data access [56], but it is also about how to make the user interact with the simulation services.
Two implementations developed by Ohio Supercomputer Center (OSC) that show how the framework can be implemented and used in reality are Polymer Portal and Manifold Flow Predictor.

4.1 Polymer Portal

This portal is developed by OSC’s web application team, and I took the responsibility to develop the frontend using Drupal, which is an open source Content Management System (CMS). Polymer Portal is a “one-stop resource” that bundles access to commercial software modeling and simulation services with training in computation and 3-D modeling [36]. It was developed through a 2007 agreement between PolymerOhio, Inc. and OSC. It features “web-based applications designed to help increase productivity for polymer companies, including companies from the plastics, rubber, and advanced materials segments”. [36] It can be regarded as a single entry point that integrates multiple software services for users to access. The front end interfaces with an E-Commerce service where the various training and simulation services are offered as items in a catalog. Integrated Authentication, Authorization, and Accounting services are provided through a common Active Directory (AD) server. This allows a
variety of tools to be integrated without significant work on either the resource provider or the toolmaker’s part. In the first generation, we offer a simulation software service for users. Industrial users can access the simulation software services through a virtual machine run on the node, which has 4 cores, 16 GBs of memory, and 100 GBs of disk space, as long as they purchase the valid software license.

4.1.1 User Experience

Basically, there are two kinds of potential users in our system: academic users and industrial users. Both kinds of users can access simulation services and training services supported by the service provider. After a user signs in, he or she can browse available services and training opportunities listed as a catalog and add selected software to the shopping cart if they want to purchase them. As the user purchases software, the services can be accessed by clicking on to redirect the user to a certain platform without signing on again. For users who purchase training services, the browser will lead the user to Moodle services, which is an external software service based on Remote Desktop Connection Protocol populated with training course materials. Academic users use Moodle to publish or access courses, while industrial users take advantage of Moodle for updated seminar information. For users who would like to use the service of P4, which is a web-based application, the browser will redirect the user to a web application platform. For users who buy Modex3D services, a browser will launch VMware view to show a remote desktop running Moldex3D. This automatic browser redirection mechanism prevents our user from worrying about the actual address of certain services and, therefore, raises the point of user simplicity.
Although training and simulating are both main components, the resources they require are quite distinct. Different from the training function, the simulating function requires large-scale computational resources that can be provided by various simulation tools that reside on a Virtual Machine (VM) or native hardware (i.e. high-performance computers). For academic use, the workload is uneven and occurs in bursts. For instance, whereas before a submission deadline or during lab time, the usage of our system increases dramatically, during other times, the resource lays idle. Cloud architecture allows computing resources that would have otherwise been idle to be employed serving software for many different classes, each of which have their own usage patterns, thereby evening out load and reducing the total amount of computing resources required. For high-end industrial users, the situation is quite different. Since individual enterprises are more likely to develop and test their product constantly, computational resources need to be assigned to them more evenly and consistently than academic users. Based on these two differences, when individual enterprises or academic users purchase software, individual licenses should be assigned to them. Academic licenses should be assigned to educators and students, while commercial licenses should be assigned to industrial users.
4.1.2 Architecture

There are currently five kinds of Common-off-the-Shelf (CotS) services underlying our Polymer Portal. Polymer Property Predictor Portal (P4) is a simulation service based on a web-platform that allows users to perform calculations on a given polymer molecule. Production Flow Analysis and Simplification Toolkit (PFAST) is also a web-based simulation service that allows both industrial and academic users to automate the manual methods of Production Flow Analysis (PFA). In addition to web-based simulation services, currently Moldex3D, a VMware-based service, and Ludovic, a
client-server mode application, are provided. Moldex3D is an engineering simulation tool designed for plastic injection modeling simulation, while Ludovic is a 1D modeling software dedicated to co-rotating twin screws [40]. Ludovic’s purposes are controlling the extrusion process, improving the material quality, and going deeper in the material or process analysis and optimization. [40] In addition, we can also connect our users to an external Moodle service run by another state agency through Remote Desktop Protocol (RDP). Moodle service is an online training service provided by the service provider. The training courses can help non-IT professional users begin using the simulation software in a short period.

Heterogeneity is the nature of a cloud computing system since underlying resources are based on different hardware architecture. For the sake of simplicity, we use a single sign-on (SSO) [27] mechanism to allow users to login only once and be able to access the individual system without being prompted to be authenticated again for each service. In our system, web-based applications, VMware images, external Moodle service, and native high-performance computing hardware are five distributed services, and they all use a common Active Directory (AD) to implement SSO authentication and authorization [42]. This SSO mechanism addresses the challenge of user simplicity very well since users can simply use the service without knowing the location and other details of the cloud computing infrastructure. For the sake of managing these services, which are based on various platforms running on their own server, a common AD service is served with authorization information to web applications, VMware images and remote Moodle service. Our Active Directory treats available services as objects and an object
forms a group. We divide users based on groups and assign different access permissions to distinct users belonging to each group. In our system, for instance, if a user purchases both PFAST and Moldex3D services, the administrator will add the user account to both PFast and Moldex3D groups in the Active Directory so that the user can access both services via Polymer Portal. In addition, since our users range from those in academia to industry, we need to issue different licenses to different users.

4.1.3 Implementation

There are many powerful modules that can be used to implement our features, such as shopping cart, Customer Relationship Management, and single sign-on. Below is the overall component architecture of Polymer Portal.
The Web Portal can be regarded as a common entry point for users to access services. Once the user logs on, he or she will have permission to access the service. This is implemented by the LDAP module [16], which will look up the user into the Active Directory (AD) in order to authorize the user. The Moodle On-line Training service is provided by a service provider. It helps users learn how to use the service. We deploy a Virtual Machine Pool on OSC’s HPC cluster, which guarantees the performance of each
virtual machine. There are two key organizational features in the architecture: the common Drupal database and the shared AD server. The common Drupal database enables the various components to easily share information about users. For example, having the UberCart module [15] and the CiviCRM module share the database means that it is easy to generate reports detailing purchases made on the site. The use of a shared AD server between Drupal and Moodle enables single sign-on functionality. This improves the user experience by allowing the site to appear as a unified whole rather than a loosely conglomerated bunch of programs. It also removes the need for a lot of time-consuming and error-prone manual work, and enables a much simpler user model.

Below is the inner architecture of the Web Portal component.

![Diagram](image)

Figure 7. Web Portal Inner Architecture
We used a XAMPP suite among web container, database, and web server to build Polymer Portal. It is easy to set up and configure. Drupal Core not only provides nice graphics user interface, but it also can be customized by users to create their own themes. The LDAP module provides the connection to our Active Directory server, used to authenticate and authorize the users who want to log on. The Ubercart Module provides a configurable product catalog and includes catalog pages and a block to display product categories [15]. Our customer can purchase the service license using credit card or debit card. It connects to PolymerOhio’s payment system. CiviCRM [51] is the Drupal module used for Customer Relationship Management (CRM). This primarily consists of generating reports concerning site usage, or which customers bought what products. It can provide many services, including notifying users of upcoming events that they might be interested in, analyzing trends (for example, determining the overall number of visitors for a given time period), determining revenue breakdown, and customized report generation.

4.2 Manifold Flow Predictor

Ohio Supercomputer Center (OSC) collaborated with TotalSim [52] Company to develop a portal named Manifold Flow Predictor (MFP), which leveraged OSC high-performance computing resources and an open source tools named OpenFoam [44] to help P&G analyze fluid flow. I took the responsibility of building the portal by Drupal and visualizing simulation results by Virtual Network Computing session. TotalSim [52] is a company that provides a competitive and flexible approach to analyze Computational Fluid Dynamics (CFD) [53]. Computers can be used to perform calculations required to
simulate the interaction of liquids and gases with surfaces defined by boundary conditions [53]. Because this requires a lot of computational resources, a high-speed supercomputer will be needed. Users can visualize the simulation result generated by OpenFoam. They benefit from this SMaaS architecture because they can process the job and visualize the result only through a web browser; no extra software or hardware is needed. Below is an example of simulation result generated by OpenFoam.

Figure 8. Example of an OpenFoam simulation result [43]
4.2.1 User Experience

MFP users are CFD experts but not computer experts. They do not know how to manage HPC jobs through scripts and command line options, so it is necessary to make the user interface easy to use. Below is the work flow diagram of MFP.

![Web Browser](image)

Figure 9. Manifold Flow Predictor Work Flow

The OSC authenticated user can access MFP services as long as he or she logs on. MFP provides a friendly interface for users to create manifold jobs. Users only need to upload manifold files and fill in fluid flow parameters, which are required by OpenFoam in the web page:

- Manifold geometry
- Fluid properties

34
• Boundary conditions

Once the user finishes these three steps of filling parameters, he or she can click the “Save” button to save this job on the HPC system. The user will then be redirected to the MFP home page, which lists all the manifold jobs that have been created. After that, users can fill in the number of steps, which is a special parameter; the number of steps “tells” OpenFoam how many times the iteration of simulation will be used for simulating. When the simulation result is generated, the “Visualize” button will be activated. Then, the user can visualize the result within the web browser. We use software called Paraview [45] for visualization. Once the CFD expert analyzes the simulation result, he or she can judge what number of iterations will be appropriate for simulating a manifold flow.

4.2.2 Architecture and Implementation

There are two main components in MFP’s architecture, HPC Cluster and Virtual Network Computing (VNC) [19] [23] server. For the sake of simulation performance, we build MFP on the OSC Cluster. Manifold jobs can be processed by OpenFoam on the cluster, and the simulation result will be stored on the system. VNC server is used to provide the visualization feature. The simulation result is loaded by Paraview in the VNC session, and it will be popped up as a Java Applet, which is supported by a modern web browser. Below is the architecture of Manifold Flow Predictor.
We use Portable Batch System (PBS) [13] to schedule jobs. PBS is a resource manager that automates CPU and resource scheduling. Pending jobs will be queued until there is available computing resource to process it.

The frontend of MFP is implemented by Drupal, and the backend is implemented by script languages, which interact with a HPC system and a VNC server. Frontend are typical web forms that are used to collect user inputted Computational Fluid Dynamics parameters, which are required by OpenFoam. Once the job has been created, a script
within the backend will substitute the dummy parameters in the job template using the collected ones. Then, the job template will be submitted to PBS as a job script. After PBS assigns computing resources to the job, the job will be processed. For the visualization part, we use a script to implement two things: starting the VNC server and locating the new generated simulation result. In fact, the “Visualize” button, which is described above, is the link of this script. We also modified the X11[54] startup file in order to start Paraview, which loads the target-generated result when the VNC session starts. Each job will have its own VNC session, so that visualizing will not create conflicts. The user can use a web browser to access a VNC session rather than install a VNC viewer client. This is not only a great benefit of using a VNC session but also the benefit of a SMaaS framework. The user only needs a thin client to access heavy-weight simulation software and to visualize the result.
Chapter 5

Evaluation

In this chapter, we explain surveys that we conducted with users in order to evaluate the actual benefits of the integrated framework. We applied the SMaaS framework to two projects, Polymer Portal and Manifold Flow Predictor. The surveys are listed within this chapter as examples, and all information was collected through questionnaires or interviews. The questions focus on user experience, such as actual cost, simplicity, and satisfaction, and aim to evaluate the usability.

5.1 Surveys

5.1.1 Polymer Portal User Survey

- Are you satisfied with Polymer Portal? In what areas can it improve?
- What are you using Polymer Portal for?
- How would you rate the quality of the guiding documentation? Is it clear enough to go through?
- Did you encounter any problems with sign on, browsing the service catalog, purchasing, or accessing the Virtual Machine (install VMview)?
- Have you ever used Moldex3D or Ludovic before? How much money does it cost (Software & Hardware)? How long does it take to do configuration?
Are you satisfied with the price for different periods of licensing for Moldex3D and Ludovic?

How would you rate the quality of the response times when you performed keyboard and mouse actions within VMview?

How would you rate the transfer speed between USB storage and virtual machine?

How is the color rendering of the simulation result in VMview?

Would you recommend Polymer Portal to others with similar needs?

How did you hear about Polymer Portal?

Do you feel you can perform customization within your catalog?

5.1.2 PolymerOhio Staff Survey

What is the motivation of building Polymer Portal?

Before Polymer Portal was delivered, did the polymer company make simulation and modeling? If so, how to implement simulation? After delivered, does PP actually improve benefits?

Although Polymer Portal users are polymer experts, they are not computer professionals. Simulation software is difficult to use, how would you rate your learning process?

Is simulation software, such as Moldex3D and Ludovic, able to satisfy complicated industrial cases, like modeling and simulation?
Polymer Company purchases software license and supporting hardware and hires IT professionals (one-time investment) versus using Polymer Portal. How do you choose? Why?

5.1.3 Manifold Flow Predictor User Survey

- Have you ever submitted jobs to High Performance Computing System before? If yes, is it complicated?
- Have you ever used OpenFoam?
- Does Manifold Flow Predictor actual resolve your problems?
- Is it easy to go through three steps to create a manifold flow job?
- Is there a compatibility issue when you click the “Visualize” button within the browser?
- How is the browser response time when you start to visualize your data?
- How would you rate the quality of the response speed when you perform keyboard and mouse actions within the browser?
- How is the color rendering of the simulation result?
- Are you satisfied with your experience with Manifold Flow Predictor or does it need any improvements?
- Was using MFP for the first time easy? If not, what would you suggest to help with this?
- Have you ever heard of Virtual Network Computing (VNC)?

5.2 Analysis of User Surveys
We interviewed Polymer Portal and Manifold Flow Predictor users to talk about their using experience. They were most concerned about the cost for using the service and the simplicity to use. Below is the analysis of the survey feedback.

5.2.1 Cost

Our Polymer Portal is built on a SMaaS framework; it is under a “pay-for-use” mode so that users can purchase a short-period service license with lower cost. One of Polymer Portal’s users said that he always uses Moldex3D to simulate plastic injection; however, the price of the Moldex3D license is fairly high. It is about $50,000 for one year, and it does not even include a software update or technical support, which requires an additional $20,000. The supporting hardware also still needs to be paid and maintained. His company is a medium-sized polymer company; they do not want to spend lot of money on such this kind of up-front cost. However, Polymer Portal resolves all these issues. The user can purchase a one-day Moldex3D license, which only costs $180, to access the simulation service within a high-performance virtual machine that is configured and maintained by the Ohio Supercomputer Center. Another Polymer Portal user said that he really benefited from the training course. He wants to analyze the extrusion process of a certain material, but he does not know how to use Ludovic [40], which is a complicated application. He purchased our Ludovic software service at a much lower price and attended the training course. After a one-week training course, he mastered most key skills related to his analysis requirement. However, if he had not used our “on-demand” simulation software service, he would have had to pay $28,000 for a one-year Ludovic Software license and $8,000 for one year of training, which wastes
much money and a lot of time. Below is a diagram that shows the cost of currently available simulation services on Polymer Portal.

<table>
<thead>
<tr>
<th>Product</th>
<th>Solution Type</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ludovic: 1 Day Software Access for Ohio Clients</td>
<td>Extrusion</td>
<td>$225.00</td>
</tr>
<tr>
<td>Ludovic: 1 Week Software Access for Ohio Clients</td>
<td>Extrusion</td>
<td>$900.00</td>
</tr>
<tr>
<td>Ludovic: 4 Weeks Software Access for Ohio Clients</td>
<td>Extrusion</td>
<td>$2,700.00</td>
</tr>
<tr>
<td>Ludovic: 1 Day Software Access for non-Ohio Clients</td>
<td>Extrusion</td>
<td>$250.00</td>
</tr>
<tr>
<td>Ludovic: 1 Week Software Access for non-Ohio Clients</td>
<td>Extrusion</td>
<td>$1,000.00</td>
</tr>
<tr>
<td>Ludovic: 4 Weeks Software Access for non-Ohio Clients</td>
<td>Extrusion</td>
<td>$3,000.00</td>
</tr>
<tr>
<td>Moldex3D eDesign: 1 Day Software Access for Ohio clients</td>
<td>Injection Molding</td>
<td>$180.00</td>
</tr>
<tr>
<td>Moldex3D eDesign: 1 Week Software Access for Ohio clients</td>
<td>Injection Molding</td>
<td>$720.00</td>
</tr>
<tr>
<td>Moldex3D eDesign: 4 Weeks Software Access for Ohio clients</td>
<td>Injection Molding</td>
<td>$2,160.00</td>
</tr>
<tr>
<td>Moldex3D eDesign: 1 Day Software Access for non-Ohio clients</td>
<td>Injection Molding</td>
<td>$200.00</td>
</tr>
<tr>
<td>Moldex3D eDesign: 1 Week Software Access for non-Ohio clients</td>
<td>Injection Molding</td>
<td>$900.00</td>
</tr>
<tr>
<td>Moldex3D eDesign: 4 Weeks Software Access for non-Ohio clients</td>
<td>Injection Molding</td>
<td>$2,400.00</td>
</tr>
</tbody>
</table>

Figure 11. Price of Simulation Services on Polymer Portal

As the figure shows, there are various kinds of licenses for users to select, and they can select a type of license depending on their own need. According to the surveys, users like one-day licenses the best, since they are cheap and give them more flexibility.

5.2.2 Simplicity

Manifold Flow Predictor (MFP) provides a friendly user interface for submitting jobs to High Performance Computing (HPC) resources to process. There is no need for the user to know the details of the internal process, such as job processing and result visualization. Our MFP users are computer non-professionals, but they are fluid flow
experts. According to the survey, they benefited from MFP’s simplicity. One user said that she does not even know what Virtual Network Computing (VNC) is. The simulation result comes up from her “browser,” which is actually a Java Applet, and then she can edit the result using Paraview within the browser. MFP wraps HPC, software application, and technical details into a service that can be easily accessed. However, if we had not provided MFP, the user would have had to learn how to write a job script with parameters and complex options to submit to a HPC system. Below is a snippet of job script.
#PBS -N PGM-RUNXX_OpenFOAM
#PBS -l nodes=2:ppn=8
#PBS -l walltime=0:30:00
#PBS -j oe
#PBS -S /bin/bash

#Initialize OpenFOAM

# for openfoam-1.7.x
# OpenFoam-1.7.x needs gcc.4.4.5 and mvapoich2-1.6i-gnu
module load gcc-4.4.5
module switch mpi mvapich2-1.6-gnu
module load openfoam-1.7.x
# if the module does not work... alternate method to set enviroment...
# need to set env by :
#export FOAM_INST_DIR=/usr/local/OpenFOAM
foamDotFile=$FOAM_INST_DIR/OpenFOAM-1.7.x/etc/bashrc
#source /usr/local/OpenFOAM/OpenFOAM-1.7.x/etc/bashrc

#Move to the case directory, where the 0, constant and system directories reside
# not sure this export command is needed...
export OpenFOAM_case=/nfs/02/ndem0001/PG/CASES/RUNXX
cd $PBS_O_WORKDIR

surfaceCheck constant/triSurface/walls.stl > LOG-
surfaceCheck
..../bin/ManifoldFlow/MFMgr/checkwatertight.pl

blockMesh > LOG-blockMesh
snappyHexMesh -overwrite > LOG-sHM
createPatch -overwrite > LOG-createPatch
checkMesh > LOG-checkMesh

# Command below scales mesh (i.e. "(1e-3 1e-3 1e-3)" to convert mm to m)
transformPoints -scale @@MFP_SCALE@@ > LOG-scale
decomposePar > LOG-decomposePar

touch RUNXX.foam
touch LOG
 chmod 777 LOG

mpiexec simpleFoam -parallel < /dev/null >> LOG

foamLog LOG
As the code snippet shows, it is difficult for users to substitute parameters and use command line options. In contrast, our MFP hides all these processes. The diagram below shows the three steps of creating a manifold fluid flow job.
Figure 12. Three Steps to Create a Job of Manifold Flow Predictor
After the user finishes these three web forms, the job is saved and is easy to submit. Furthermore, our MFP user said that he likes the list of manifold jobs on the MFP home page. It makes simulation results easily to browse, and the user can manage the jobs he submitted merely through moving the mouse.

5.2.3 Sustainability

Polymer Portal can be regarded as a commercial platform that supports users to purchase various kinds of software service. Currently, it supports five kinds of simulation software services all implemented on, and hence dependent upon the High Performance Computing (HPC) system at Ohio Supercomputer Center (OSC). So far, the current number of users using the simulation service is small and (hence) the performance is acceptable; however, as the number of users increases with limited computing resources, performance could be the bottleneck. We plan to collect usage statistics for each kind of simulation service, so that a rational allocation of computing resources to each service may be made <refer to some of Dr. Calyam’s papers here – ask him which ones>.. For example, according to our survey, most of our users use Moldex 3D and Ludovic, so more resources must be allocated to the virtual machines on which these two kinds of software services are deployed. In addition, there will be other companies who want to cooperate with OSC in the future to leverage the HPC resources by using Polymer Portal. The issue of authentication scalability will also emerge as this collaboration grows. OSC uses Active Directory to implement Polymer Portal’s authentication and authorization, while other companies might use other kinds of internal authentication mechanism, such as OpenID or Shibboleth, which cannot be compatible
with Active Directory. This will affect the sustainability of Polymer Portal, and a way to
deal with it must be discovered.

5.2.4 User Satisfaction

So far, most of the feedback we received from their users about Polymer Portal
and Manifold Flow Predictor is positive. These two implementations of SMaaS
integrated framework are usable, which resolves their actual needs in real life. According
to the survey, Polymer Portal users can use Moldex3D to resolve the problem they met in
plastic injection molding within a virtual machine that runs on a high-performance
computing node. The functionality is as usable as the separate components, which need
to be configured and maintained by the users themselves. In addition, users save a lot of
money that can be used to purchase software license and supporting hardware. Manifold
Flow Predictor (MFP) users can simulate fluid flow in different steps to judge the best
number of steps for simulation and then visualize the result through MFP. They not only
benefit from leveraging high-performance computing resources to process simulation
jobs but also from visualization of the simulation result within the web browser. All of
these features simplify the users’ steps. There is no need for users to start a Virtual
Network Computing (VNC) server manually and start Openfoam to load the simulation
result separately.

As described above, users are satisfied with the cost and performance of using
simulation software on Polymer Portal and the simplicity of submitting manifold jobs on
Manifold Flow Predictor. However, according to the survey, some places still need to be
improved in the future:
• VMview and VNC session’s performance should be better. Since the software applications are deployed on our HPC system, the thin client can be regarded as an “interface” to the system for users to use.

• The transfer rate between USB storage and VMview is low. Virtual machines cannot access the Internet in their current stage, so it will be better when the user can use the Internet to transfer files.

• The job cannot be re-processed unless the user goes through the three steps to fill in the parameters again on MFP. A MFP user said that he only wants to make a tiny modification for the manifold job to observe the difference, but he has to submit a new job rather than modify the existing one.
Chapter 6

Conclusions

This thesis creates a framework that integrates different software services and web components into a wrapped site in order to reduce cost, simplify the workflow, and improve the using experience for users. I explain the architecture and list important components of this framework. Furthermore, we apply this framework to two concrete products, Polymer Portal and Manifold Flow Predictor, which provide Simulation Software as a Service for users. We also utilize user surveys to evaluate these two products in usability and user experience. We find that users benefit from this framework from lower cost, software service flexibility, simple user experience, and performance of software.
References


[8] Song Guo, Fan Bai, Xiaolin Xu, “Simulation Software as a Service and Service-Oriented Simulation Experiment” IRI 2011


[22] F. Liu, etc “SaaS Integration for Software Cloud”, CLOUD 2010


52


[33] SOAP, http://www.w3.org/TR/soap/

[34] GM, http://www.gm.com/


[40] Ludovic, http://www.scconsultants.com/En/Ludovic.html


[42] David E. Hudak, etc “Simulation as a Service: A Cloud-Based Framework to Support the Educational Use of Scientific Software”


[46] HTTP, http://www.w3.org/Protocols/


[55] Dan Wing, “Network Address Translation: Extending the Internet Address Space” Internet Computing, IEEE 2010

[56] Xiaorong Li, etc “Building an HPC-as-a-Service Toolkit for User-interactive HPC services in the Cloud” 2011 WAINA

[57] Mario Christ, etc “Measuring Web Portal Utilization” 35th Hawaii International Conference on System Sciences (HICSS) 2002

A.1 Polymer Portal User Response

- Are you satisfied with Polymer Portal? In what areas can it improve?
  
  *Yes.*
  
  *VMview should be faster, but it not an issue. Transfer rate is low between VMview and USB storage.*

- What are you using Polymer Portal for?
  
  *Moldex 3D.*

- How would you rate the quality of the guiding documentation? Is it clear enough to go through?
  
  *Good.*

- Did you encounter any problems with sign on, browsing the service catalog, purchasing, or accessing the Virtual Machine (install VMview)?
  
  *Not yet.*

- Have you ever used Moldex3D or Ludovic before? How much money does it cost (Software & Hardware)? How long does it take to do configuration?
  
  *Moldex3D, I don’t know the cost of it. It is really a complicated configuration process.*
• Are you satisfied with the price for different periods of licensing for Moldex3D and Ludovic?

    Of course, one-day license.

• How would you rate the quality of the response times when you performed keyboard and mouse actions within VMview?

    Should be faster.

• How would you rate the transfer speed between USB storage and virtual machine?

    Very slow, should be faster.

• How is the color rendering of the simulation result in VMview?

    Nice.

• Would you recommend Polymer Portal to others with similar needs?

    Yes.

• How did you hear about Polymer Portal?

    Internet.

• Do you feel you can perform customization within your catalog?

    Yes, I can purchase software services through that.
A.2 PolymerOhio Staff Response

- What is the motivation of building Polymer Portal?

  *We intend to provide Ohio polymer companies a broad range of value-added programs and simulation services.*

- Before Polymer Portal was delivered, did the polymer company make simulation and modeling? If so, how to implement simulation? After delivered, does Polymer Portal actually improve benefits?

  *Yes. They use local computing resources to simulate.*

  *Polymer Portal does improve the work efficiency and reduce the cost.*

- Although Polymer Portal users are polymer experts, they are not computer professionals. Simulation software is difficult to use, how would you rate your learning process?

  *Most of them said the simulation software is easy to learn. Most of the buttons functions are the professional nouns they are familiar with.*

- Is simulation software, such as Moldex3D and Ludovic, able to satisfy complicated industrial cases, like modeling and simulation?

  *Yes.*

- Polymer Company purchases software license and supporting hardware and hires IT professionals (one-time investment) versus using Polymer Portal. How do you choose? Why?

  *We use Polymer Portal. It is not necessary to hire IT staff, the polymer expert can master the simulation software in a short time.*
A.3 Manifold Flow User Response

- Have you ever submitted jobs to High Performance Computing System before? If yes, is it complicated?
  
  *Yes, it completed and you need to write script.*

- Have you ever used OpenFoam?
  
  *Not yet.*

- Does Manifold Flow Predictor actual resolve your problems?
  
  *Yes.*

- Is it easy to go through three steps to create a manifold flow job?
  
  *Yes, it is easy to use.*

- Is there a compatibility issue when you click the “Visualize” button within the browser?
  
  *Since I only use Chrome, there is no compatibility*

- How is the browser response time when you start to visualize your data?
  
  *About 10 seconds of response time.*

- How would you rate the quality of the response speed when you perform keyboard and mouse actions within the browser?
  
  *It is nice.*

- How is the color rendering of the simulation result?
  
  *It looks good.*

- Are you satisfied with your experience with Manifold Flow Predictor or does it need any improvements?
Yes, I am satisfied with this MFP experience.

- Was using MFP for the first time easy? If not, what would you suggest to help with this?
  
  *I need to know the function of each button, after that, everything becomes easy.*

- Have you ever heard of Virtual Network Computing (VNC)?

  *Not yet.*