Enabling computational geoscience

Literature review

Project Members (Group-14)

A.I. Ranatunga - 090423F
M.H. Kumara - 090269L
M.S.H Jayaratna - 090205N
L.N.P.T Perera - 090377P

Project Supervisors

Dr. Shahani Markus Weerawarana

Coordinators

Dr. Malaka Walpola

10th May 2013

Acknowledgement

This research project would have not been possible without the support of many people. Apart from the efforts of the team members, the success of the project depends largely on the encouragement and the guidance of our project supervisor Dr. Shahani Markus Weerawarana. We are grateful for her constant support towards the project work and feedback given in every event. Her suggestions, attention and time invested to achieve the expected target were crucial during the period. We would also like to take this opportunity to thank Dr. Malaka Walpola; the final year project coordinator, Dr. Chandana Gamange; the Head of the Department and the staff members of the Department of Computer Science and Engineering of University of Moratuwa for providing us with a convenient environment and resources to complete this project.

In addition we would also like to thank personals from different scientific communities for providing us detailed information and support throughout the project work. We would like to show our greatest appreciation to Jun Wang of Indiana University , Dr.Chris Mattmann; Senior Computer Scientist - NASA Jet Propulsion Laboratory Pasadena, Suresh Marru; Chair - Apache Airavata project, Martin Desruisseaux, Sanjaya Medonsa, Danushka Menikkumbura, Pavithra Kulathilaka and the members of Apache Airavata team for their valuable advice and the corporation at every stage of the project.

Table of Contents

1. Introduction 1
1.1. Background 1
1.1.1. Scientific computing 1
1.1.2. Workflows 1
1.1.2.1. Scientific workflows 1
1.1.2.2. Scientific workflow management systems 2
1.1.3. Science gateways 3
1.1.4. Geoscience 3
1.1.4.2. Geoscientists and their focused areas 3
1.1.4.3. Geoscience research and experiments 4
1.2. Research problem 4
1.3. Research objectives 5
2. Literature review 6
2.1. Computational Geoscience 6
2.1.1. Geospatial data & data sources 6
2.1.2. Geosciences standards 7
2.1.3. SOA influence in Geo sciences 7
2.1.3.1. Scientific computing and SOA 7
2.1.3.2. SOA in Geoscience 8
2.1.3.3. Challenges with meeting geoscience requirements in SOA 8
2.1.4. Geographic Information Systems (GIS) 9
2.2. Workflow Management Systems 9
2.2.1. Existing Workflow Management Systems 9
2.2.1.1. Kepler 9
2.2.1.2. Taverna 10
2.2.1.3. Triana 10
2.2.1.4. Pegasus 11
2.2.1.5. Apache Airavata 11
2.2.2. Desirable features of WfMS 11
2.2.2.1. Handling dynamic workflows 11
2.2.2.2. Interoperability 12
2.2.2.3. Data management 12

2.2.2.4. Quality of Service 13
2.2.2.5. Ease of use 13
2.2.2.6. Provenance tracking 14
2.2.2.7. Sharing and reuse 14
2.2.2.8. Monitoring and Error handling 15
2.2.3. Evaluation on existing solutions 16
2.3. Science Gateways 16
2.4. Apache Airavata 17
2.4.1. Apache Airavata architecture 18
2.5. OGC 19
2.5.1. OGC's WPS (Web Processing Service) 20
2.5.2. OGC's WCS (Web Coverage Service) 21
2.5.3. OGC's WMS (Web Mapping Service) 22
2.5.4. OGC compliant and implementing products 22
2.5.4.1. MapServer 22
2.5.4.2. GeoServer 24
2.5.4.3. ZOO Project 24
2.5.4.4. PyWPS 25
2.5.4.5. GRASS 25
2.6. Geoscience gateway 25
2.6.1. Limitations 27
2.6.1.1. Requisite for computational expertise 27
2.6.1.2. Regenerating common geoscience computations 28
2.6.1.3. Difficulties in handling data-intensity 28
2.6.1.4. Difficulty in effective tool selection 28
2.6.2. Computational geoscience concerns 29
2.6.2.1. Spatial data visualization 29
2.6.2.2. Collaboration 29
2.6.2.3. Reuse 30
2.6.2.4. Scalability 30
2.6.2.5. Publishing and Retrieving Services information 31
2.6.2.6. Ease of use 31
2.6.2.7. Security 31

2.6.2.8. Reproducibility 32
2.6.2.9. Interoperability 32
2.6.3. Framework 32
References 34

Table of Figures

Figure 1: Functionality of science gateway 17
Figure 2 : Apache Airata Architecture [48] 19
Figure 3 : GEON gateways 26
Figure 4 : Data access points 26
Figure 5 : Selecting the area for data 27

Table of Tables

Table 1: Comparative Summary of Workflow Management Systems 15
Table 2: OGC Support in Geo-tools 23
Table 3 : Analysis Framework for Architectural Concerns 33

1. Introduction

Geoscience is one of the most prominent research areas involved in modeling and analyzing the evolvement of Earth systems for the betterment of mankind. Advancements in scientific computing such as the advent of High Performance Computing (HPC) has great impact in geoscience researches that involve data-intensive applications, complex computations and modeling. Expansion of powerful tools provides the environment to create a convenient platform that enables scientists to practice complex researches. Evolvements of workflow systems has witnessed a substantial growth in simplifying scientific research processes hiding computational complexity and provide scientists with the convenient platform to engage with.

The project Dhara mainly focuses on enabling geoscience in scientific computing. Thus discussing about fundamentals of scientific computing and geoscience has significant is a key focus in the initial phase of this thesis. The following chapter gives a brief introduction on scientific computing, geoscience and workflows.

1.1. Background

1.1.1. Scientific computing

The Computer Science area where computing deals intensively and extensively with scientific data and calculations, is known as 'Scientific Computing'. Scientific computing evolved as means of solving mathematical problems numerically using a computer [1]. Since then, the bond between the science and computing has become stronger and stronger. Advent of massively parallel computers and the resultant high performance computational (HPC) capabilities can be considered as a result of this bond. The ultimate goal of HPC infrastructure is to create an easily accessible platform for scientists, who may have low-levels computer science expertise, to effortlessly carry out their experiments that involve computer-intensive tasks.

1.1.2. Workflows

A workflow is an abstract model of a series of steps connected to represent a real world process where each step defines a specific task or functionality. According to [2], "A program (or script) is to a workflow what an unstructured document is to a (structured) database". Workflows are capable of hiding the complexity of the execution process. They provide a representation of complex analysis composed of diverse models [3]. A workflow is typically authored using a visual front-end or can be hard-coded, and their execution is delegated to a workflow execution engine that handles the invocation of the remote applications [4]. Workflow systems can be categorized mainly into two sections; business workflows, scientific workflows.

1.1.2.1. Scientific workflows

We are focused on scientific workflows which are significantly different from the business workflows. One of the major differences between business and scientific workflows are that, business workflows are control flow oriented while scientific processes are data flow oriented.

The data flow does not impose an order of execution, it only specifies the input and output of components; component B should use component A's output as input. Three motivations for scientific workflow have been identified as follows: [5]

- Some complex e-science applications often require the creation of a collaborative workflow
- Many e-scientists lack the necessary low-level expertise to utilize the underlying computing infrastructure
- Workflow specifications can be reused, modified and shared

With the availability of vast computational power, scientific communities are engaged in solving interconnected problems spread over multiple disciplines. These activities have accelerated by mapping them onto a workflow. For an example, in earthquake science, workflows are used to predict the magnitude of earthquakes within a geographic area over a period of time. Scientific workflow is the procedure of combining data and processes into a structured set of steps which can operate as a solution to a scientific problem. It includes declarative description about each component and its input and output. [2] Workflows can be generated manually by scientists or using third party tools to assist larger workflows. They utilize distributed resources in order to access, manage and process large amounts of data from a higher-level [6]. Processing and managing such large amounts of data require addressing proper techniques for storage facilities and to handle scaling resources.

1.1.2.2. Scientific workflow management systems

Scientific workflow management systems act as a middleware for creating, combining, executing workflows and data management tools. They have evolved with the aim of producing a high level platform enabling the end-user scientists to create, manage, compose, execute and test scientific workflows without concentrating on the low-level computational technology. In the paper [2] scientific workflow management systems are recognized as providing means to; model and specify processes with design primitives, re-engineer developed processes such as verification and optimization, automating the execution of processes by scheduling, controlling and monitoring the tasks.

Before the invention of workflow management systems, programs and scripts were used as means to run scientific process with predefined steps. But workflow management systems provide advantages over the earlier mechanism for constructing and managing computational tasks. Main advantage is that they provide a simple programming model to compose a sequence of tasks simply by connecting the outputs of one task to the inputs of another. An added advantage is the intuitive visual programming interfaces of workflow management systems, which make them more suitable for users without ample programming expertise. [7]

Currently there are many workflow management systems developed targeting various scientific communities such as Triana, Pegusas, Taverna and Kepler [8]. Most of these systems are designed to solve problems in certain domains.

1.1.3. Science gateways

Science gateway is a collection of tools, applications and data that allows scientists to run their applications having minimum concern of where the actual processing takes place or the underlying complexity of computational resources. Science gateways open up opportunities to access all services in one particular place without considering where the services are located.

Science gateways are expected to serve a common objective of a community. There are various communities who are interested in different research areas such as bioinformatics, astrophysics, oceanography and geoscience. Development of a gateway for a particular community costs time, money and effort significantly. As research paper [9] describes such a development should focus on funding, project goals, tools, community engagement, rewards and recognition.

1.1.4. Geoscience

1.1.4.1. What is Geosciences?

"Geoscience includes all the sciences (geology, geophysics, geochemistry) that study the structure, evolution and dynamics of the planet Earth and its natural mineral and energy resources" [10]. Geosciences investigations range from analyzing the processes that have shaped the Earth through its 4600 million year history to predicting the behaviors of earth systems. In a nutshell, geoscience can be generalized as the science which enables people to analyze and understand the behavior of our planet from core level to atmospheric level. Explorations of planets in the galaxy have widened the scope of geoscience experiments. Geoscientists argue the fact that geoscience is not limited to planet Earth but go beyond.

"Modern Geosciences is founded on plate tectonic theory which states that the outer part of the Earth (the lithosphere) is composed of a series of interlocking plates in relative motion" [10]. Geo-scientists found that origination of mountains; volcanic activities and natural disasters are causes of movements between earth plates. "Exploration and responsible development of natural resources (oil, gas, coal, minerals, construction aggregate, water, soil), preservation of the natural environment, restoration from environmental damage, mitigation of Geo hazards such as earthquakes and landslides, and exploratory research like the Mars space mission" [11] are at the top emerging applications in geoscience. In the recent past, the involvement of computing technologies and resources to enhance the geoscience experiments and research has been increased. The effect of industrial revolution during past decades and the rapid growth of world population strongly affect the behavior of the earth systems. Therefore geoscience has now become a prominent research area which mainly focuses on providing a better and protective environment for all living beings.

1.1.4.2. Geoscientists and their focused areas

"Geoscientists study the Earth's physical composition, structure, history, and the natural processes. They provide information to society for use in solving problems and establishing policies for resource management, environmental protection, public health, safety, and welfare." [11].The main concern of geoscientists is about earth related issues. Geoscientists are extensively

involved in behavioral analysis of earth system, disaster predictions, discovering and developing supply mechanisms of fossil fuels, groundwater, construction materials and mineral ores, studying and mitigating Geo hazards such as volcanic eruptions, earthquakes, floods, and landslides and exploring new ideas about the natural world from the depths of the oceans and the core of the Earth to the outer reaches of space.

Geoscientists' work and experiments heavily depend on long term analysis of earth systems and high data intensive calculations. Geoscience experiments in modeling, analyzing and predicting require high performance computing devices for processing large amount of geospatial data along with data visualization tools. But the lack of the low level computational knowledge of geoscientists acts as a barrier to gain maximum benefit of underlying computational resources. There are many emerging software systems such as scientific workflow management systems to overcome the above barrier by providing understandable user friendly interfaces while hiding the complex computational resources.

1.1.4.3. Geoscience research and experiments

Geoscientists perform numerous amounts of researches for analyzing the behaviors of earth systems to predict the future evolvement of the earth. Researchers are often engaged in popular research areas, such as world temperature variations, wind flow, pressure variations at a spatial point, disaster prediction and oceanographic analysis. Intricate researches in the area of disaster prediction have succeeded in helping thousands of lives. “Geoscience research and applications often involve a geospatial processing workflow. This workflow includes a sequence of operations that use a variety of tools to collect, translate, and analyze distributed heterogeneous geospatial data” [12]. The nature of experiments are often long running, involving massive scientific computations, data intensive and consist of heterogeneous, “multi-scale and multidisciplinary, geospatial processing workflows involve a number and variety of structured activities and computations in a distributed and heterogeneous environment” [12]. Integration of different sets of geospatial data with geo processing workflows needs a high degree of interoperability.

1.2. Research problem

In the past few decades researches in geoscience field have achieved a massive success with the help of software systems in the industry. Nature of the experiments in geoscience are often long running, data intensive, complex and consist of heterogeneous geospatial data and use sophisticated high performance computational resources to perform such computations. Geoscientists often face difficulties in gaining and accessing complex computational resources because of their low level knowledge in computing. Due to this problem critical geoscience researches are getting delayed.

Lack of cohesive open community and meritocratic contribution models and a neutral venue for core features causes science experiments to be continually reinvented [13]. The isolated researches in geoscience domain cause geoscientists to spend time in inventing the same process due to lack of awareness of the available geoscience resources.

Advancement of high performance computers boost up the speed of geoscience experiments in the world. Many geoscience related software systems have been built and are still evolving with the latest technologies to support ongoing geoscience experiments. With the revolution of SOA (Service Oriented Architecture), many international organizations tend to expose their computational resources, geo spatial data and services through a web interface. Collection of these services leads to building online Geographic Information Systems (GIS), by which many geoscientists are benefited. There are a number of geo spatial standards issuing organizations which focus on creating standards to provide uniform access to geoscience related services. Many software tend to build on those standards. Nevertheless geoscientists perform their experiments in an isolated manner.

While having significant support from computer science related resources for geoscience related researches and experiments, it needs to build a uniform platform by using open standards with a strong, open and committed community. With this uniformity and standard interfaces, geoscientists will work collaboratively in geoscience researches without reinventing the wheel with less concern about underling sophisticated computational resources.

1.3. Research objectives

The research project of geo enabling Apache Airavata by integrating Open Geospatial Consortium (OGC) standards has given rise to multiple research objectives. This research area is consisted with cutting edge technologies in the world. Research mainly spans into three major areas: science gateways, scientific workflows and management systems and geoscience related standards (i.e.: OGC).

When analyzing science gateways, it is expected to understand the domain of scientific computing, underlying technologies, community interests, impact on science and its applications related to geoscience. Such analysis will allow understanding the kind of environment resulted in developing Apache Airavata.

Another main focus area is scientific workflows and workflows management systems. It is intended to investigate the difference between business workflows and scientific workflows along with the impact on developing science gateways using Workflow Management Systems (WfMS). It is further required to analyze the different types of WFMS and their characteristics. It will assist to identify major design decisions in Apace Airavata.

Exploring geoscience related standards is another major area in this project. It is essential to identify major tools used as GIS and the focus of each tool in geoscience researches. Further identifying the geoscience standards used and importance of OGC among them are also vital. This will allow understanding the importance and pathways of integrating OGC standards to Apache Airavata.

2. Literature review

This literature review explores technologies we identified as related to enabling computational geoscience. It starts by discussing the state-of-the-art technology support in geoscience. The section 2 and 3 under literature review discuss Scientific Workflow Management Systems and Science Gateways respectively. Then this is focused on tools identified to be used in our project, 'Project Dhara'. Therefore next section explores Apache Airavata, followed by a section on Open Geospatial Consortium and its standards. Final section presents a discussion on a geoscience gateway based on the current limitations and requirements we have identified so far.

2.1. Computational Geoscience

Emerging technologies in scientific computing enhance the performance of scientific applications. Geoscience is a vast area and it is benefited from the innovative technologies and software systems in computer science. Geoscience experiments and researches are often engage with data intensive and time consuming scientific computations. Enhancing computational capabilities in geoscience related software systems from geoscientist's perspective will be an excessive benefit for them. Geoscience experiments mostly engage with creating workflows which contains a sequence of operations that uses variety of tools to collect, translate, and analyze distributed heterogeneous geospatial data [12]. Scientific workflow management systems (WfMS) arise as a solution for scientists to create, manage, execute and monitor workflows without considering underlying complex computational resources. But existing WfMSs deal with issues in supporting geoscience workflows. The technical requirement for invoking geoscience workflows in an asynchronous manner is vital due to their data-intensity and time-consuming features. But current scientific WfMSs are not efficient enough in handling asynchronous concerns.

There are number of geoscience data sources which are used in geoscience applications. Worldwide organizations and scientists create standards to offer an interoperable environment within geoscience applications, through defining standard interfaces for geoscience processes and data services. Computational geoscience has been involved in numerous amounts of geoscience experiments.

2.1.1. Geospatial data & data sources

Geospatial data plays a vital role in geoscience experiments and researches. Spatial data and non-spatial data are the two major types of data available in geoscience applications. Non spatial data are known as raster data. Spatial data contains details about a particular point which can be temperature, pressure or other characteristic. Usually spatial data is known as vector data. Raster data includes geo reference data in an area. Raster data is mostly imaginary where the data contains in image files. Geo data sources provide geo reference data about the locations of the earth. There are online and offline geo data sources. Online Geo data sources can be accessed through the internet while offline data sources act as a database which can be embedded in geo

related applications.

Estimations indicate that terabytes of geospatial data is collected within a day from sensor networks, satellites and other geo reference systems, while it increases daily in the rate of gigabytes. People tend to use the internet frequently to obtain geospatial data. There is a major requirement for a common set of standards for geospatial data because of the high growth and access. Establishing standards to publish data in specific geospatial data formats are major requirement of the modern society. There are well known geospatial data formats existing "namely, ESRI Shapefiles, Microsoft Excel files, HTML files, and GML files" [14]. But the world still lacks the usage of common data formats in exchanging geospatial data which leads to interoperability issues. JPEG, PNG and GIF are some raster data types are common image file formats. OpenDAP, NetCDF and RAMADDA are popular geospatial data sources.

2.1.2. Geosciences standards

Geosciences experiments are often involved with using high performance computational resources. With the revolution of SOA, there are many service providers tend to expose their services as a web services to general public. Geo data and processing services are also among the available web services. Hence providing a common standard to publish and access web services is a major requirement. As a solution to this, people create standards for various geospatial applications.

Standards are distributed into so many areas and limited to specific communities. For an example Canadian Geoscience Standards Board [15] is one community with one set of standards. And another is Australian geoscience standards [16]. When considering standards OGC is the largest geo related standards formulated so far. Open Geospatial Consortium is a geospatial standard creating organization which is a collaboration of more than 400 geosciences related organizations. But some standards are developed on top of some OGC standards.

GeoSciML [17] was built on top of OGC GML. There are also organizations dedicated for interoperability between geoscience systems. One is Group on earth observations [18]. It provides tools for decision making for variety of users for different aspects. It provides standards to bring these services together and provides portal to access these data in user friendly manner. There are lots of projects involved with Group on earth observations like disasters, health, climate, weather .etc.

2.1.3. SOA influence in Geo sciences

2.1.3.1. Scientific computing and SOA

The new scalable SOA operates as a framework of services and service-based development [12]. It is also emerging as the basis for distributed computing and large networks of collaborating applications [12]. SOA has become the ultimate solution for service reuse. Business organizations tend to expose their services on line, targeting various users to earn profits by providing a quality service. In scientific computing environments, reusing existing services is a crucial requirement. Scientific experiments often engage in long running processes with massive

data sets using high performance computing devices. It's not feasible for a single person or a group of people to build an entire platform to perform scientific computations due to costs in acquiring scientific computing devices and maintaining them. Nevertheless scientists often require high performance computing resources to perform complex computations associated with their experiments. Further, most of the experiments are critical in nature and the time factor is a major concern. However as scientists are not proficient in dealing with high performance computing resources, computer engineers need to build software platforms, frameworks and libraries. Such convenient environments abstract the underline high performance computational resources by providing uniform interface for scientists to perform their experiments.

2.1.3.2. SOA in Geoscience

Geoscience is an intricate field which often involves massive data intensive computations. Geoscientists use distributed geo processing services available in the Web, Cloud and Grid computing environments. Distributed geo processing is required by the distributed nature, size, and sophisticated evaluation algorithms of geo data [19]. Geoscientists are not proficient in dealing with high performance computer resources and it will be a disadvantage to gain the maximum benefit of computational resources. However SOA has the capability of embedding distributed processing into some overall service which hides the complex data evaluation tasks behind simple, easy-to-use geo-services [19]. In early days, desktop GIS systems were widely adapted in geoscience experiments. But recent advances in SOA allow users to migrate from dedicated desktop solutions to on-line, loosely coupled, and standard-based services which accept source data, process them, and pass results as basic parameters to other intermediate services and/or to the main model, which also might be made available on-line [20]. Service-oriented application is becoming further useful in handling issues of data accessibility and service interoperability for environmental models [20]. Geospatial experiments need advanced algorithms in processing data which runs on super computers. Using SOA, designers expose complex algorithms and processing resources as online services and scientists use these services to create complex computation patterns. SOA make geoscience experiments manageable and efficient.

2.1.3.3. Challenges with meeting geoscience requirements in SOA

In spite of the excellence of SOA, geoscience applications cannot get the maximum benefit due to many limitations. SOA use Extensible Markup Language (XML) for Web Service Description Language (WSDL) and Simple Object Access Protocol (SOAP) for message processing. However both WSDL and SOAP are not specialized for HPC. Developing geospatial services has been constrained by either API-based or URL-based services [21].The major keystones of geoscience are spatial analysis and vector overlay computation. The issue is that geospatial data are not simple objects. Vector data contain rich information about both spatial shapes and non-spatial attributes [21].SOAP (Simple Object Access Protocol) deals with strings, characters and numbers. Hence SOA is not very efficient in handling complex geospatial data types. Geoscience applications are highly data intensive applications with gigabytes of data flowing within computer networks. Therefore SOA can be a bottleneck for transmitting massive data sets across the computer networks. When large volumes of data are transmitted over the computer network, it is a major concern how to retrieve the input data in an efficient manner [21]. Furthermore it

will definitely be a challenge, as it may not be feasible for a service to maintain the network connection for a long time to transfer and read the input data [21]. Another major challenge is to construct the spatial index framework dynamically based on the input datasets of a web service. There are ongoing researches to handle these drawbacks of SOA. Moreover parallelism seems to be a promising solution and a research direction towards efficient service-oriented GIS [21].

2.1.4. Geographic Information Systems (GIS)

Geographic Information Systems (GIS) is a specialized software which is essential for geospatial analyzing, managing and interpreting geospatial data [22]. These systems can also centralize distributed geospatial data and make them accessible in several forms such as image maps, digital maps and raw data files [23]. Modeling, simulation and visualization can be considered as major functionalities required in geospatial applications. GISs cater the needs of scientists, policy makers and general public in geographic analysis and geographic modeling [22] .Models can be based on GIS in different levels. GIS is used; to prepare data, to visualize the model output, to implement model using its functionality, to input and visualization of output [24].A variety of open source and proprietary GIS products have been implemented to cater the basic needs of GIS users. GRASS GIS is a popular open source GIS while ArcGIS is an example for a proprietary GIS.

2.2. Workflow Management Systems

2.2.1. Existing Workflow Management Systems

Currently there are many workflow management systems developed targeting various scientific communities such as Apache Airavata, Triana, Pegusas, Taverna and Kepler [8]. Most of these systems are designed to solve problems in certain domains.

2.2.1.1. Kepler

“The Kepler scientific workflow system provides domain scientists with an easy- to-use, yet powerful system for capturing scientific workflows (SWFs)” [25]. Streamling workflow creation and execution process enable scientists to design, execute, monitor, re-run, and communicate analytical procedures repeatedly with minimal effort. Kepler is based on mature data flow oriented Ptolemy II system which focuses on visual and module oriented programming along with targeting on multiple component interaction semantics. “Ptolemy is the only available system which allows one to plug in different execution models into workflows” [25]. The controllers of workflow execution process are known as directors. Individual components of a workflow are designed as reusable actors which can represent data sources, sinks, data transformers and etc. Actors and directors are the main building blocks of Kepler's SWFs. Actors can have multiple input and output ports along with parameters which specifies certain behaviors of SWFs. Kepler supports both runtime and design checking of workflow while providing versioning, exchanging and archiving capabilities. Kepler provides user friendly GUI for creating workflows and workflows can be exchanged in XML using Ptolomy's own Modeling Markup Language(MoML). Actors run as local java threads. “But are extended to spawn

distributed execution threads via web and Grid services, as well as through Java's foreign language interface (Java Native Interface)" [25]. Kepler has inbuilt actors to support generic workflows while providing a capability of writing its own actors that suits to a particular works flow. It has a good support for execution of workflows in computer grids. Kepler is extensively used in biology, ecology, geology, astrophysics and chemistry fields.

2.2.1.2. Taverna

Taverna is a powerful, scalable and domain independent workflow management system [26].It has the main objective of porting incompatible services. It is consisted of a rich UI where users can drag and drop the components and link them together.It has the facility to obtain web services via WSDL, port them via UI and run the workflows. Then the output will be displayed via the defined format. Taverna is an open source and free software. No server is required to be installed. Workflows can be shared in myExperiment [27] or in BioCatalogue [28]. Taverna can also be executed in command line. It is currently used by a large community and widely used for multiple purposes such as heart simulations, high throughput screening, Genotype/Phenotype studies, and astronomy [29]. Thus researchers from these domains are capable of abstracting data querying various databases, model them and analyze those combining local and remote resources.

Furthermore it is possible to run Taverna workflows on a grid which should be accomplish with Taverna server or command line tool. For each new grid, a separate plugin needs to be installed. As an example; to access caGrid services, caGrid plugin needs to be installed. It is more secure to write a particular web service and hide the protocol level details (i.e. how remote method invocation has been done) by writing a relevant plugin for that particular service.

2.2.1.3. Triana

Triana is an open source graphical workflow based environment developed at Cardiff University which combines an intuitive visual interface with a wide variety of built-in tools. It is used by scientists for a range of tasks, such as signal, text and image processing, gravitational wave detection project, GEO 600 [4] etc..

It has been extended to gain benefits of Grid computing environments with the aim of providing better integration of existing Grid technologies. It has broadened the range of functionalities of the system allowing users to interact with services running in a Grid environment, coordinating transparent resource sharing between collaboration parties. The basic functional components in Triana are called units which are connected via directed cables that support input and output data. A unit hosts the web service representing the underlying protocol. The cable implementations are resolved at runtime based on the types of units that are connected [4]; in one case data may transfer between local file locations while in another case between two remote locations. Ability to modify and republish nodes is a unique feature of Triana. This was achieved via viewing the source code of each node and allowing users to modify and recompile it. Triana supports several looping and conditional selections when executing nodes and data flow.

Original tools used with the application are written in Java. Abstraction of Grid workflows from the underlying Grid technology workflow has achieved using GAT (Grid Application Toolkit) shielding domain scientists from individual resource manager or file transfer protocols.

2.2.1.4. Pegasus

Pegasus is another Workflow Management System used in number of scientific domains including astronomy, bioinformatics, earthquake science, gravitational wave physics, ocean science and others [30]. For an example; Pegasus is used by earthquake scientists to generate more accurate hazard maps that can be used by civil engineers to design new construction in earthquake-prone areas [31].

A significant characteristic in Pegasus is the concept of abstract workflows. This concept is used to describe and model abstract job computations in distributed environments [32]. The abstraction means the lack of details on the actual execution, such as resource locations in the workflow description. User describes the workflow in resource independent way and Pegasus maps them into appropriate heterogeneous resources [32]. This concept is significant because it has become the base to most of Pegasus's key features.

The abstract workflow description provided by the user is mapped into distributed resources through the workflow mapper by bridging the scientific domain and the execution environment automatically [30]. The mapping process includes locating appropriate software and computational resources along with data indicated in the workflow description. It may also include workflow restructuring and transformation in order to optimize overall performance and data management respectively [1]. As the optimizations are taken care of the workflow mapper, Pegasus allows the scientists to construct workflows in abstract terms without considering about the underlying execution environment by shielding the [1]low-level details [1] [32].

2.2.1.5. Apache Airavata

Apache Airavata is a pure open source software framework for executing and managing computational jobs and workflows on distributed computing resources such as national grids and super computers. Since Apache Airavata will be the target WfMS of the project, a comprehensive description has included in the later part of the document.

2.2.2. Desirable features of WfMS

2.2.2.1. Handling dynamic workflows

According the nature of experiments, the vision of supporting dynamic, adaptive workflows is accelerated. Capturing mechanisms to create certain results via reproducibility is one aspect of handling dynamic workflows [33]. The demand on dynamic workflows increases due to the nature of scientific methodologies, such as; the runtime decisions on later steps of a workflow may need the initial steps' results. Dynamic workflows may respond for external events [33] and may depend on the results of data analysis computation. In addition, dynamic workflows could

have taken place due to observation and modification of different scientists and researchers in same experimental procedure. In contrast the workflow execution may determine its path dynamically at runtime.

Thus managing a dynamic workflow is a challenge in the application. The managing process of a workflow is evolving through cycles. Workflow can be share, refine and rerun to check the result. The process will continue until scientists get a satisfied result. Hence a proper user interaction should handle via an appropriate user interface and the results of execution should be query and display in an understandable manner. Real time workflow status monitoring and dynamic notification is also plays a key role in the dynamic workflow handling.

2.2.2.2. Interoperability

Emerging of several workflow management systems with specific features has stimulated the scientific researches in recent past. The collaborative nature in scientific researches results multiple contributors from geographically distributed locations taking part in developing a single workflow experiment. Almost all of the widely used workflow systems have been developed with different communities, targeting different domains and exhibiting specific features within relevant area. They use different workflow engines, description languages, and formalism which makes it less interoperable [34]. Due to that differentiation it is hard to express workflow of one system using a description language of another [34].

Thus a key architectural requirement in SWfMS is to facilitate the interoperability between different SWfMS, so that one system can take benefits of tools and unique features of another system [35]. This enables the reusing and sharing of workflows among systems. Accordingly in order to achieve interconnection within different systems, workflows should be interoperable.

The interoperability can be achieved in 3 levels [35];

1. Task-level
2. Workflow-level
3. Subsystem-level

GEMLCA service [34] has achieved interoperability of heterogeneous workflow systems via workflow engines integration which can be listed under the 3rd level achievement. The proposed solution executes workflow in the native platform and exposes it to run in non-native platform. The basic idea in the integration of workflows is the ability to use a workflow as a one node/task in the system.

2.2.2.3. Data management

Data management is a one of major requirement in the workflow lifecycle. Data management includes data inputs in creation and execution workflows with handling results. These steps access terabytes of data at initials steps, intermediate levels and in end results as well. Data produced and used in the workflow execution process deals with several data types, such as provenance data for collaboration purposes, metadata to query or store workflow information,

input data from remote data sources, data from intermediate levels to be analyzed and feed the later tasks [34].

There are several procedures that need to be following in data handling. Metadata or provenance data need to agree to a community based standard which is rich enough to describe the data sets and information on data [34]. In addition maintaining metadata catalogues to query and store metadata is another requirement. Use of common metadata catalog independently from the systems will provide good coordination of data access and security.

In contrast identifying the input data formats in various remote processes for the execution of workflows is a major concern. Schedulers in the system responsible for selecting data sets and appropriate computational resources to run tasks [34].Required data need to feed consistently and fast to places where computation takes place. Thus a proper data management section is a major requirement in a workflow management system, specially using for data-intensive applications.

2.2.2.4. Quality of Service

Requirements of Quality of Service (QoS) in workflow management systems need to be specified and optimized. There are several concerns that took into account in recent past such as; reducing execution time, maximizing bandwidth etc. [33]. In addition to time constraints there are some other QoS parameters; responsiveness, fault tolerance, security, and costs [33]. Performance, reliability, security and fault tolerance are also related to QoS. The WfMS need to meet the QoS requirements via exposing them to demanding techniques. Accessing cloud services opposed to traditional reservation of resources is one aspect of optimizing resource allocation with dynamic scaling up and down requirements.

2.2.2.5. Ease of use

Scientific workflow execution requires high computational power, which is gained through cyber infrastructure such as Grid computing and Cloud computing. Open Science Grid and TeraGrid are science gateways which provide widely used and well tested computational capabilities and services [33]. Nevertheless effective and optimized usage of these computing resources requires expertise of tools such as Grid Toolkits. For the average scientist or researcher who does not have ample expertise in such tools, effective utilization of the underlying computational infrastructure becomes a tedious task.

Computational complexities in scientific domain require undivided focus of the scientist. Therefore it is a key requirement of scientific workflow management systems to provide its users with optimum level ease of use for handling underlying resources. Separating the science-focused and technology-independent problem solving environment from the underlying advanced computing infrastructure is considered as the potential approach for this issue in the paper [35].

2.2.2.6. Provenance tracking

Provenance tacking has become the main focus in many research projects as it is a critical component in workflow sharing, reusing and end result reproducibility. Provenance provides answers to: "Who created this data product and when? When was it modified and by whom? What was the process used to create the data product? Were two data products derived from the same raw data?" [7]. Scientists and researchers often require putting in substantial effort for managing the large amount of provenance information related to their work. Therefore scientific workflow management systems should provide automated capabilities to capture metadata, log the sequence in applied steps, parameter settings and intermediate data products [36].

Sharing and reusing of workflows are common practices in scientific communities. Tracking and efficient capturing of provenance information provides important information that is the key to preserving data, determining the data quality and authorship [7]. Reproducibility is an essential feature of computational scientific experiments as same as conventional laboratory experiments [36]. When provenance information is effectively embedded in workflows fellow scientists can redo a particular experiment using the same data, following the same steps, evaluating the intermediate data products and finally reproducing the experiment results. Holistically provenance information enables the scientists to reproduce experiment results and evaluate the validity of each other's hypotheses [33].

2.2.2.7. Sharing and reuse

Sharing and reuse in various aspects is widely used practice within the scientific communities. A Workflow is a great way to electronically capture a process, which paves the way to sharing and reusing them [33] . Encouraging researchers to include workflow usage into their practices would contribute to rapid advancements. Nevertheless there are several workflow management systems in use at present. Therefore it is essential that reuse is supported among them. Support for reuse has to be done in two different aspects: workflow semantics and infrastructure [33].

Researchers have to agree on process semantics used in workflows. Workflow management systems (WfMS) should provide capabilities and emphasis on constructing workflows in formal and explicit ways. Sharing is the major intention behind workflow reuse. Nonetheless reusing is often followed by refinements. Abstractions allow scientists to identify the level of description useful to share workflows so that other scientists could refine and use [33]. Hence abstraction and refinement capabilities require being present in workflow management systems.

There is a range of distinct computational environments such as Open Science Grid, TeraGrid and Amazon EC2, which are used to run workflows. As a result workflow sharing and reuse often lead to instances where the same workflow requires to be run on different and heterogeneous environments. Workflow management systems should allow user created workflows to be easily run in these different environments without alteration.

2.2.2.8. Monitoring and Error handling

Long running and data intensive scientific workflows are common scenarios. These workflows are often collaboratively developed and even modified. Then these workflows are constructed with various distributed tasks over network communications [35] . All in all above characteristics of scientific workflows impose additional challenges regarding errors and failures. Therefore scientific workflow management systems require the capability to monitor status and failure of a running workflows in various levels including mechanisms for catching and handling errors automatically [35].

Pegasus workflow management system provides error handling which has been identified as a key feature of it. It tries error handling in various levels: retrying tasks, retrying entire workflow, providing workflow-level check-pointing, re-mapping portions of the workflow, trying alternative data sources and providing a rescue workflow containing a description of only the work remaining as a last resort [30].

	Apache Airavata [37]	**Taverna** [26]	**Triana**[38]	**Kepler**[39]	**Pegasus**[40]
Workflow engine	Airavata Workflow Engine	Freefluo	Triana engine	Ptolemy II framework	DAGMan
Workflow description language	WS-BPEL	Scufl (language using XML)	BPELL, Triana specific language	Ptolemy's Modeling Markup Language	DAX - based on XML
User Interface	XBaya GUI	Taverna UI	Triana UI	UI from Ptolemy system	Pegasus web portals, WINGS
Features	Scalability, Performance, Monitoring, Interoperability, On demand service creation	Scalability, Reuse, Interoperability, Error handling, Security, Concurrency	Scalability, Performance, Error handling, Support automating repetitive tasks	Scalability, Reuse, Process and data monitoring, Provenance tracking	Scalability, Performance, Reuse, Provenance tracking, Error recovery, Reliability
Involvement in geoscience	OLAM project	Used in PML	GEO 600 project	support Earth-Grid access	CyberShake project
Support for geoscience standards	-	WPS	-	OpenDAP	-
Specific areas	biology, chemistry, oceanography	bioinformatics, geoscience, astronomy, chemistry	gravitational wave detection	biology, ecology, geology, astrophysics and chemistry	astronomy, bioinformatics, physics

Table 1: Comparative Summary of Workflow Management Systems

2.2.3. Evaluation on existing solutions

In the recent past, a growing effort has been made in enhancing workflow systems with an aim of improving support for scientific research. Workflow systems often consist of a visual front-end and a back-end for handling the complex execution processes. They act as a middleware for research process in multiple domains. Hence they continue developing to fulfill requirements of a specific domain.

The inner execution techniques in the system such as workflow engine operation, handling data and workflow formalism etc. differs from one to another. Functional components and the connectors have different purposes and control flows illustrate different features. Triana support viewing, modifying and compiling source code of each node [4] while Pegasus support the concept of abstract workflow which can be effectively mapped to distributed resources for the execution. Table 1 above demonstrates an overview on technical details of each system identified above.

2.3. Science Gateways

Science is the studying how nature behaves [41]. Nevertheless it is consisted of experimental science, theoretical science and computational science. Nowadays scientific researches involve massive calculations and large amount of data storage and transfer. Computational science plays a major role in today's researches which can be considered as an amalgamation of computer architecture, scientific algorithms involving mathematics and application of science.

Today scientific researches involve advanced applications of science, demanding massive usage of algorithms and mathematical calculations (i.e. DNA sequencing technologies). Conversely no single scientist is capable of handling all the computational tasks individually. Therefore more advanced architectures are needed to cater these requirements.

Scientist should be able to reuse what other scientists have implemented in the same domain. This is one of the key motivations for building science gateways. There are several places where scientists can consider for access high performance computing. It can be regional grids such as NWICG [42], Open Science Grid [43], TeraGrid (Now the project has been extended as XSEDE [44]) and DOE [45]. Most of the science gateways have 3 layers; application layer, middleware layer (SAGA) and resource layer (Teragrid).

With Teragrid there are four main gateways; Computational Chemistry Grid (GridChem), Linked Environments for Atmospheric Discovery (LEAD), nanoHUB.org, Cancer Biomedical Informatics Grid (caBIG).

GridChem was built targeting molecular sciences and biologists, physicists and chemists contribute to make the application layer contains with various interfaces. Further there are programming experts and HPC experts who maintain the distribution of these services. In contrast LEAD was built for scientists to help on demand forecast and analyze weather data. It is basically a web services based cyber infrastructure build using Teragrid as its backend resources. Further nanoHUB was designed for nanotechnology. It can visualize 3d wave

functions, energy states, absorption etc. as output [46]. Further caBig was designed to facilitate researches related to cancer. Main challenge that caBig encountered was the variety of community including researchers, doctors and patients struggled with understanding the manipulation of computing resources. Hence the concept of virtualizing access to resources via a science gateway has overcome this challenge. Access is provided by a web portal or a downloadable client. Figure 1 shows the basic functionality of a science gateway.

Plus another aspect of science gateways are VO (Virtual Organization) [46]. VO's are formed to solve a particular problem. Mostly it is a temporally group solving small problem and hence needs to share data and information.

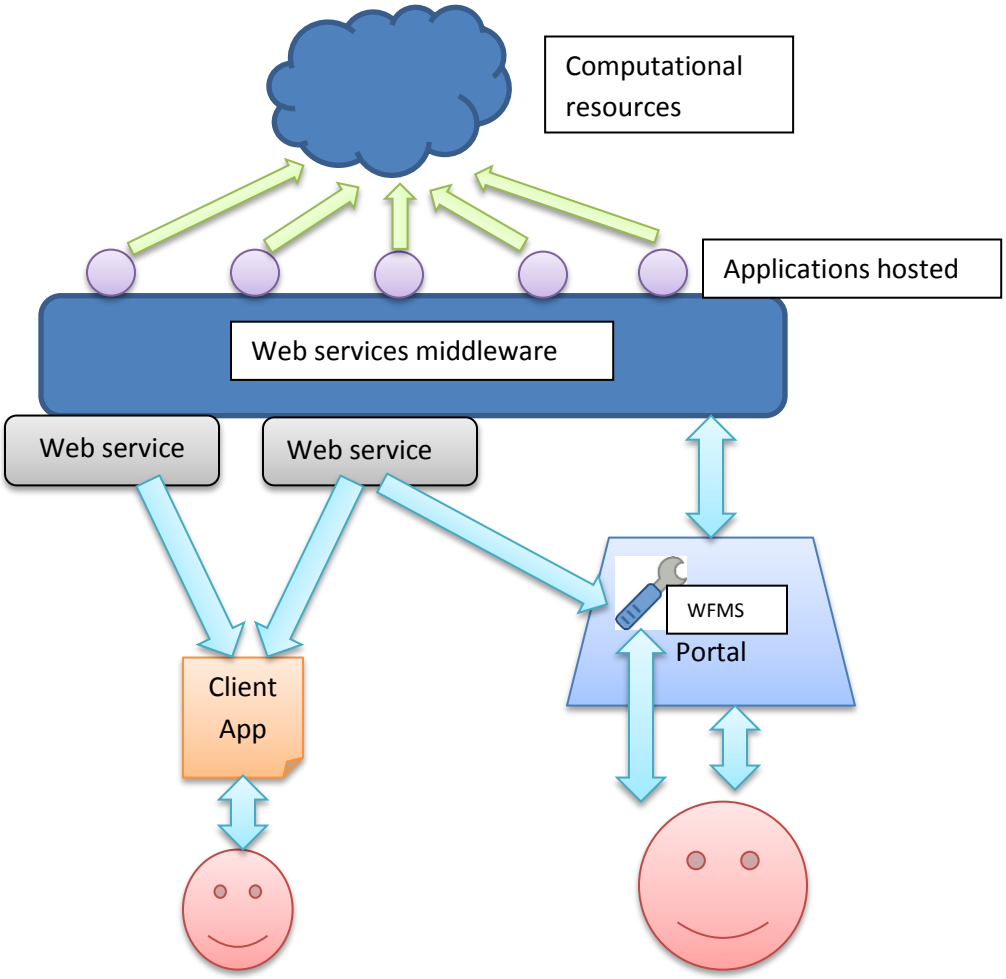

Figure 1: Functionality of science gateway

Users typically access the science gateways via a web portal, where they can download many tools such as customized WfMS (Workflow Management System), access data and other related resources. Middleware layer plays a key role in science gateways. It hosts the applications in backend resources, handles data and metadata, exposes the scientists researches output as web services to be consumed by application clients or other scientists. Other scientists may or may not use WfMS. WfMS can use those services and can give a valuable output which can again be

hosted. Difficulties with science gateways are; it takes significant effort and time to build, it needs experts in HPC to operate, the rate of failure after building is really high.

2.4. Apache Airavata

"Apache Airavata is a software tool kit to build, manage, execute and monitor workflows that executes on computing resources varying from local resources to grid or cloud computing resources" [13]. Apache Airavata is designed and built upon Open Gateway Computing Environment (OGCE) workflow engine. The origin of Apache Airavata occurred at the Extreme Computing Labs at Indiana University. "The software was initially developed to meet the challenging goals of the Linked Environments for Atmospheric Discovery (LEAD) [47] project". Airavata's main focus is to mitigate the reinvention of existing scientific solutions for the absence of a cohesive open community with meritocratic contribution models. As Apache Airavata belongs to Apache Software Foundation, it is moving forward with the collaboration of an intelligent set of open source contributors and users. Apache Airavata aims to provide a set of comprehensive tools such as baseline tools of application, workflows, job and data management systems, while synergistically working with external portals, user management and security frameworks to build science gateways with an active committed user community [13]

2.4.1. Apache Airavata architecture

Apache Airavata has a minimalistic and conceptually easy to understand architecture which is focused on supporting long running applications and workflows on distributed computational resources [13]. Its modular architecture is comprised of a set of integrated modular components based on SOA. Desktop and browser based tools for managing workflows, server side application for managing and execution of scientific application and supporting interoperability are main features of Apache Airavata.

There are four major types of modular components available in Apache Airavata and each of them is specific to provide a well-defined set of services. Modules of Apache Airavata are developed, built and packaged as standalone components as well as integrated into the Apache Airavata suite. [13]

XBaya workflow suite: Xbaya is known as the user interface provided for composing and monitoring scientific workflows. Apache Airavata has a workflow engine of its own which is will be replaced by Apache ODE in the future.

GFac: Gfac can be considered as the heart of Apache Airavata which wraps the command line driven scientific and other applications, and wrap them as network accessible services. This service layer supports several communication protocols such as SOAP, REST and JSON.

WS-Messenger: WS-Messenger is a web service-based messaging system which is responsible for reliable, scalable and efficient message delivery [13]. WS-Messenger contains a message broker which notifies subscribers about various important events occurred during the workflow execution.

Registry API: Registry is a one of major component in Apache Airavata which provides access to store and retrieve documents and workflows remotely. Present registry is mounted upon MySQL database.

Provenance aware workflow processing support is available in Apache Airavata. At the workflow processing, it's natural to have an output of a previous workflow component which remove the requirement of re-running tasks. In this situation, Provenance information stored in registry will help to find out tasks that have been previously executed from which output is available on a repository. Apache Airavata consists of a set of built-in constructs to support parallelism. ForEach workflow component in the default Apache Airavata component list is used to achieve parallelism within a workflow. This is known as Parametric Sweeps where same algorithm is applied in distributed processes to achieve a common target.

Apache Airavata is still undergoing heavy developments and is improving the support for creating scientific gateways.

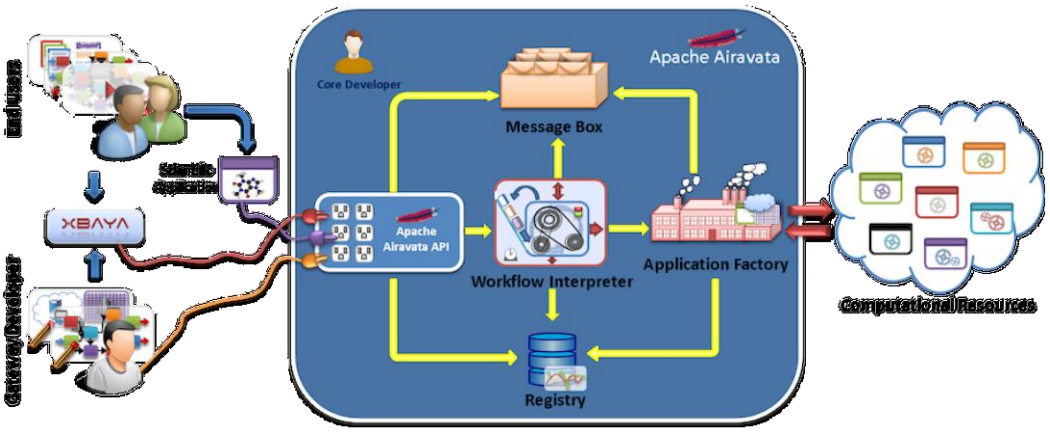

Figure 2 : Apache Airata Architecture [48]

2.5. OGC

OGC (Open Geospatial Consortium) is not for profit organization which makes geospatial standards. Objective of OGC is to "a world in which everyone benefits from the use of geospatial information and supporting technology". At the time of writing this, OGC is consisted of 445 companies, research organizations, public organizations, and universities getting together developing interface standards to make almost all geospatial applications interoperable by going through a consensus process. Set of OGC standards are so large and no single software in the world has implemented almost all the standards. To give an overview, some of the standards will be discussed here.

OGC standards expand in many domains. For an example in aeronautics Aeronautical Information Exchange Model (AIXM) and the Weather Information Exchange Model (WXXM) standards are based on OGC [49]. Spatial data for built environment for civil engineering aspects

is also a major usage where 3d modeling is heavily used. Geographical location finding is heavily used in business intelligence systems where lot of applications needs to be interoperable and common standard is a must. In defense systems OGC standards are used because of no proprietary system can provide such a broad range of interoperability cost effectively. Disaster management systems also use different kind of raster data, vector data and route data where many systems needs to be interoperable (not only IP based). Similarly it has a broader scope in many other fields as well.

These software may be OGC implementing or OGC compliant. Only OGC compliant products can wear official OGC logo. For a product or a software to become OGC compliant it needs to pass CITEE tests [50]. CITE tests are essentially about Web Services.

OGC has number of standards spread in disciplines as mentioned above. From the web services point of view we can find WMS (Web Map Service), WFS (Web Feature Service), WPS(Web Processing Service) and (WCS)Web Coverage Service. WMS basically focus on providing the relevant map. Client can request the specific map by addressing the relevant URL with standard defined parameters with coordinates. Then map is returned in PNG, GIF or JPEG, or sometimes as vector-based graphical elements in Scalable Vector Graphics (SVG) or Web Computer Graphics Metafile (WebCGM) formats.

WFS provides features over a map. After obtaining the map, client might need to show specific geographical features on top of a map. For an example it may be locations where some wild fire occurs. WMS server cannot provide that. And it should be obtained as a feature. This data is received by (Geographical Markup Language) GML format. Then those specific locations can be displayed by client software.

WPS is specified for geospatial processing. For an example in the above scenario if some client needs to identify which towns are affected by wildfire he has to call Web Processing service which can calculate the buffer area, impact on intersections .etc. So client can then display the towns affected by the wild fire. WPS can be used to provide even the simplest calculation and very sophisticated calculations.

WCS is more related to space/time phenomena. Not like WMS and WFS providing static So this dynamic information can be used for client side rendering and use more rich set of syntaxes than WMS and WFS.

Likewise OGC standards are in wider range and they are widely adopted by larger community. And it is continuously evolving until almost everybody gets benefit out of it.

2.5.1. OGC's WPS (Web Processing Service)

Web Processing Service is one of major standards in OGC. WPS defines standard interface to access geospatial processing services. WPS is used to provide wide range of geospatial services from simple mathematical calculations to larger geospatial analysis. WPS service discovery and execution happens basically in SOA strategies.

Basically client can send request to process service with specifying data required. Then user can obtain the output by a defined format. Data can be of image data format or GML (Geography Markup Language) which is an xml based language.

WPS atomic operations have been standardized to achieve interoperability among processes. There are 3 main methods defined in WPS: GetCapabilities, DescribeProcesses and Execute.

GetCapabilities method provides what processes are available and description of each. For an example to calculate an intersection area of two buffered areas, client might expect the *intersect* to be returned.

Client send it as HTTP request with KVP(key/value pair) service as WPS and request as GetCapabilities. Then client can identify whether the process desired is available or not in the relevant WPS server.

If the desired process is found, client can request more detailed information of that process. That is can be achieve through DescribeProcess. This is need of two areas as inputs name firstarea and *secondarea* where output to be named as *thirdarea*. Basically theree types of input output formats are defined: Literal data, Complex data and bounding box data [51]. Literal data can be primitive like character, string, date, etc. Complex data is vector or raster data. And the bounding box format specifies the area of a map as a box. An XML schema is defined for the co-ordinates.

Execute will trigger the process to be executed for obtain the output in GML format. Request is similar to KVP HTTP GET request where input parameters defined accordingly. If the time taken to a process is greater than the time taken for live HTTP connection then client can request to use poll method for retrieve output results of a particular process. In poll method client frequently poll and see whether the data is available.

Output can be specific MIME schema or reference URL for the result. The output can be anything. Even it can be an image where client need to specify for raw data output.

Requests can also be wrapped using SOAP structure where SOAP body is used to define request parameters of WPS. This promotes platform independent, language independent and highly extensible messaging framework support for WPS.

2.5.2. OGC's WCS (Web Coverage Service)

WCS is a OGC standard for define Web-based retrieval of coverages [52] which represents digital geospatial information representing space/time-varying phenomena. WCS exposes coverage data in convenient forms that are useful in client side rendering and as an input for scientific models. One of the major uniqueness of WCS is it allows users to query portion of geospatial information in server according to spatial constraints or query criteria.

"The WCS suite is organized as a Core, which every WCS implementation must support, and a set of extensions defining additional functionality". WCS protocol implementers can choose which extensions to support in their implementation. But implementation should include support for at least one data encoding format and communication protocol. WCS core establishes a basic

spatial and temporal extraction. WCS core establishes three requests types which are known as GetCapabilities, DescribeCoverage and GetCoverage WCS requests and responses can be make via SOAP and REST protocols.

2.5.3. OGC's WMS (Web Mapping Service)

OGC's WMS (Web Map Service) is a standard protocol for obtaining the geographical map images by a map server.Output can be obtained in PNG, GIF, JPEG OR SVG (Scalable Vector Graphics), and Web Computer Graphics Metafile (WebCGM) formats. There are basically three types of information available; service level metadata, map and features on the map. Three operations are handled by WMS spec; GetCapabilities, GetMap, getFeatureInfo.

As response for 'getCapabilities' operation, client receives services description and request parameters accepted via a XML document. In 'GetMap' request user should specify the format, width & height, SRS (Spatial Reference System), Bounding box co-ordinates etc. Height, width, Style, format and transparency can also be defined. Map can also be obtained in as layers from different WMS servers. The response can even be an exception in the case of standard violation. 'GetFeatureInfo' operation returns a feature on the map if that is enabled on the desired map server. Client has to specify the relevant X, Y parameters in the request.

WMS is the most widely used standard among WPS, WMS, WCS and WFS. Vast variety of clients obtains their map services via this standard. Therefore map servers those who do not implement OGC at its core tend to implement WMS converters to increase interoperability.

2.5.4. OGC compliant and implementing products

2.5.4.1. MapServer

MapServer is an open source founding project of OSGeo [53]. Its objective is to display spatial dynamic maps. Furthermore it is capable of querying hundreds of vector, raster, database formats and image formats. MapServer is fast and works as a high quality rendering engine. This is a CGI (Common Gateway Interface) program that sits inactive on the Web server. When a request is sent to MapServer, it uses information passed in the request URL and the Mapfile to create an image of the requested map [53]. Raw data from different sources will be input for a basic MapServer application.

A Map file, which has a .map extension, is the main configuration file for data access and styling for MapServer. It can be edited in a simple text editor and defines where data is located and the details of the target output map [53]. A map can contain different symbols that represent specific features of an area. The user can set a group of parameters where some are mandatory to describe the features of the map. Moreover the user can decide in which color the image map should highlight the desired area or locations.

The main data input will be geographic data which is utilized by the MapServer and the output map will be displayed on a HTML page which is the interface between the user and the MapServer. MapServer provides support for many OGC specifications as follows; WMS (Web

Map Server) allows use of data from several different servers and enables the creation of a network of MapServers from which clients can build customized maps. It works as a CGI and has a number of request types, which has a set of query parameters and associated behaviors for each.

Tool category		Available Products		OGC support				Other
				WMS	WFS	WPS	WCS	
Desktop GIS - Desktop GIS software provides complete and powerful set of GIS capabilities to assist in performing complex spatial analysis, spatial data creation, and visualizing data mostly in maps		Open source	GRASS GIS	✓	✓	✓	✓	Raster Data formats/Vector Data format
			uDig	✓	✓	✓	✗	KML, GeoRSS
			QGIS	✗	✓	✗	✓	WCS-T, WFS-T, GML
Web map servers - Provide capabilities for viewing, editing and sharing geospatial data		Open source	GeoServer	✓	✓	✓	✓	Only implements WPS
			MapServer	✓	✓	✗	✓	WMC, Filter Encoding, SLD,GML,SOS,OM
			OpenMap	✓	✗	✗	✗	Plugin support for WMS
Spatial database management systems – Address the issue of processing and analyzing spatial data, providing convenient front-end for visualizing and manipulation of them		Open source	PostGIS	✗	✗	✗	✗	SFS
			SpatiaLite	✗	✗	✗	✗	SFS
			TerraLib	✗	✓	✗	✗	Supports WMS and WCS with the TerraOGC extension
Software development frameworks and libraries - Support building geoscience applications with features such as visualizing, processing and analyzing	**Web**	Open source	MapFish	✓	✓	✗	✗	-
			OpenLayers	✓	✓	✗	✗	-
	Non-web	Open source	GeoTools	✓	✓	✗	✗	WFS plugin, WMS plugin
			GDAL	✓	✓	✗	✓	WMS file format recognition, Separate drivers for WFS & WCS
OGC implementation frameworks - Serve user written functionality in OGC standards		Open source	PyWPS	✗	✗	✓	✗	-
			ZOO	✗	✗	✓	✗	-

Table 2: OGC Support in Geo-tools

Mapserver can retrieve and display data from WFS (Web Feature Server) which publishes geospatial data to the web. These data can be used as a data source to render the map. It is XML-Encoded geospatial data. Data downloaded from remote locations will be saved in the directory specified in the .mapfile.

WCS (Web Coverage Service) can be referred as a raster equivalent of WFS, which allows the publication of digital geospatial information representing space/time varying phenomena. WCS can be enabled by adding metadata to the .mapfile.

MapServer is capable of processing multiple requests. Once a request is made, it works as a separate MapServer instance and has no service interruptions. The user only need to have knowledge of HTML and geographical data for a typical application and may need java, DHTML and JavaScript for a complex application.

2.5.4.2. GeoServer

GeoServer is a community driven project which focuses on sharing and editing geo spatial data. It was implemented using java language. Geo server supports [54]OGC implementations of WMS (Web Map Service), WCS (Web Coverage Service) and WFS (Web Feature Service). WPS does not derive with GeoServer core. Hence it has to be installed as a separate plugin. Furthermore it supports many backend data formats such as ArcSDE, Oracle Spatial, DB2, SQL Server, shapefile, GeoTIFF, MrSID, and JPEG2000. It also supports multiple output formats such as GML, shapefile, KML, GeoJSON, PNG, JPEG, TIFF, SVG, PDF, GeoRSS.

GeoServer is not just a piece of software but interoperable with a collection of multiple client and servers with databases.

Geo server consists of OpenLayers [55]as a component to visualize maps. And also GeoServer has the ability to integrate its data with various map API's like Google maps, Yahoo maps and Microsoft Virtual Earth.

2.5.4.3. ZOO Project

ZOO is an Open Source WPS (Web Processing Service) platform. It is a developer-friendly framework for creating and chaining web services and also OGC (Open Geospatial Consortium) compliant [56] .

ZOO project consists of three parts, namely: ZOO kernel, ZOO services and ZOO API. ZOO kernel is the core of the ZOO project, which is capable of loading dynamic libraries and handling them as on-demand web services. ZOO services are composed of a metadata file, which contains a description of available functions to be called by WPS Exec request and the corresponding code of implemented function. ZOO API is a simple and concise JavaScript library designed to call and chain the ZOO Services easily. ZOO Kernel is capable of communicating with cartographic engines and web mapping clients. It provides WPS support to spatial data infrastructures and web mapping applications [56].

Major advantage of the ZOO project is that the users can easily expose a service in WPS standard. ZOO project currently consists of few example web services based on various existing Open Source libraries and provides simple Web processing functions such as GIS format conversion and GIS file re-projection. Users can implement services in C/C++, Fortran, Java, Python, PHP, Perl and JavaScript. Then with the metadata file also provided by the user, ZOO creates a WPS compliant web service [56]

2.5.4.4. PyWPS

PyWPS Web Processing Service: is a Python program which implements the OGC's WPS 1.0.0 standard (with a few omissions) [57]. PyWPS makes it easier for people to interface geospatial calculations into SOA. PyWPS is completely written in Python which is considered a better choice of implementation language because it is very flexible language to develop in and allows others to easily integrate existing processing even if they are written in other programming languages

2.5.4.5. GRASS

GRASS (Geographical Resources Analysis Support System) is a comprehensive GIS with raster, topological vector, image processing, and graphics production functionalities [58] .GRASS used for geospatial data management and analysis, image processing, graphics/maps production, spatial modeling, and visualization. GRASS GIS is currently used in academic and commercial settings around the world, as well as by many governmental agencies and environmental consulting companies [59]. It is and open source software and it adopt several existing geographical software which are results of individual researches. But GRASS GIS is mature open source GIS software which extensively use in geoscience research and experiments.

2.6. Geoscience gateway

"Geoscience includes all the sciences (geology, geophysics, geochemistry) that study the structure, evolution and dynamics of the planet Earth" [10].Geoscience has become fast growing and multibillion dollar worth field due to the high demand for conducting day today activities in human life. Earlier day's geoscience was limited only to those who are in the geoscience field. But today the situation has been changed. Everyone on the planet Earth needs to know what is going on around for almost all day today activities. This has enforced the most powerful tools man has ever made to be used in the field of geoscience. From the perspective of computations many researches have been carried on this area and emerged the subject on computational geoscience. Geoscience typically deals with extremely large amount of data. Hence highly scalable computational resources are needed to deal with geoscience experiments.

Geoscientists do not have enough domain knowledge of manipulating such systems. Thus it is a major requirement to build science gateways in this domain. Quakesim [60] is one of the use case of a science gateway in geoscience perspective. It provides access to Web services modeling codes, data sources for researching and other aspects of earthquakes. Useful data can be obtained via the web services for modeling purposes.

It is moreover essential for scientists to work collaboratively and share the common geoscience processes. GEON is a good example for that. Science gateway aspects of GEON will be discussed here. GEON is a geoscience gateway implemented on top of TeraGrid resources. Typically science gateways need to authenticate users that need access. Once authenticated GEON provides a collection of gateways for various purposes.

OpenTopography is a gateway that provides topographical data obtained from **LIDaR** (**LI**ght **D**etection and **R**anging or **L**aser **I**maging **D**etection and **R**anging) sensors. LIDar is a method of obtaining remote data from a target by illuminating a laser light [61]. This portal access is also shown in Figure 3. Once entered users are provided with set of data which can be filtered according to the requirement. Data points are shown in Figure 4

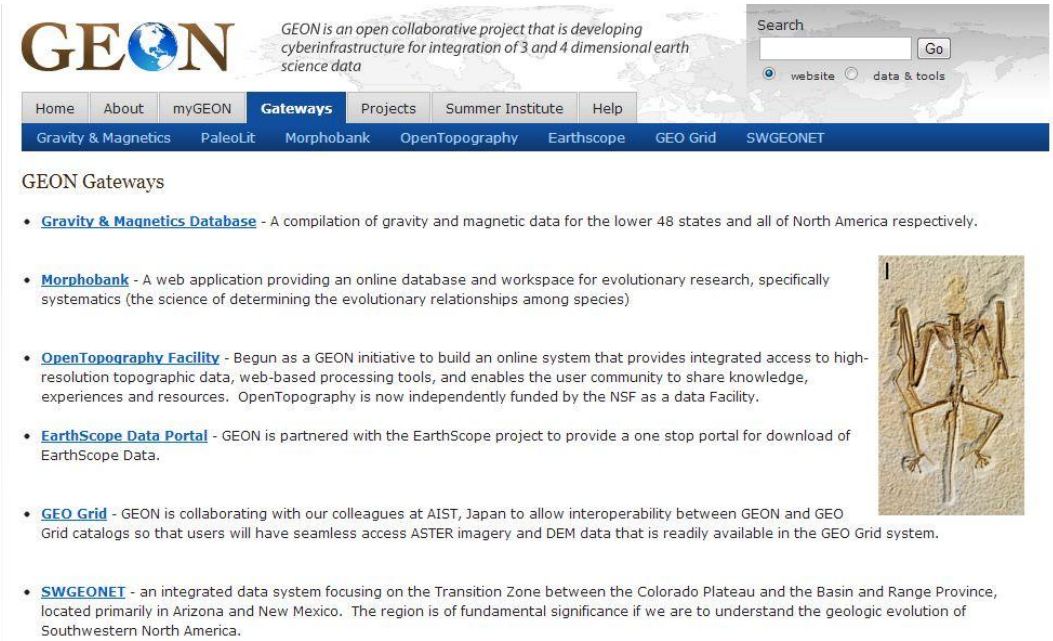

Figure 3 : GEON gateways

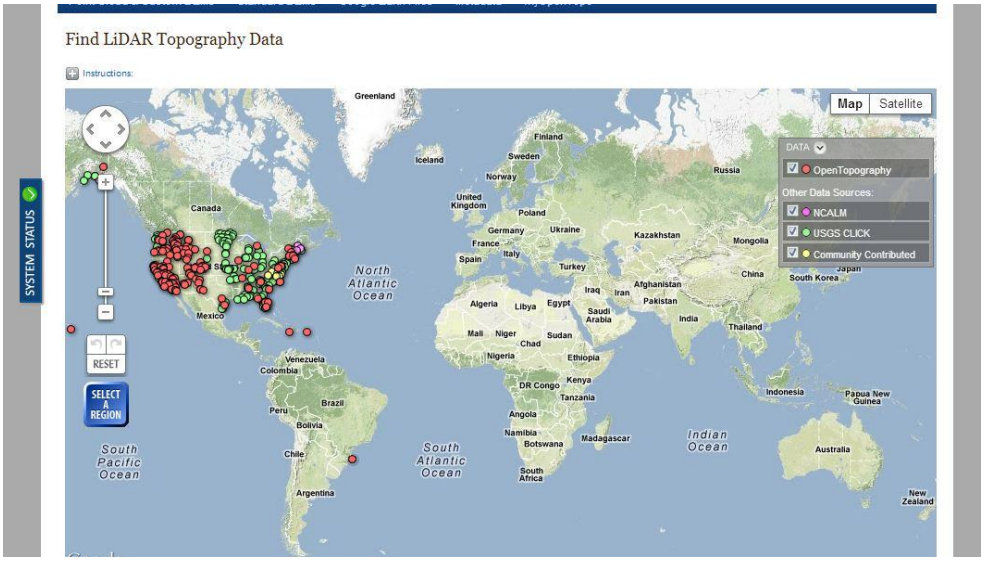

Figure 4 : Data access points

Data can be manipulated by an online web portal or by a downloadable tool. Once these points are accessed user can either select various parameters according to their requirements or can

request to process them on cloud or download the data file and process with an available tool locally. Figure 5 shows this.

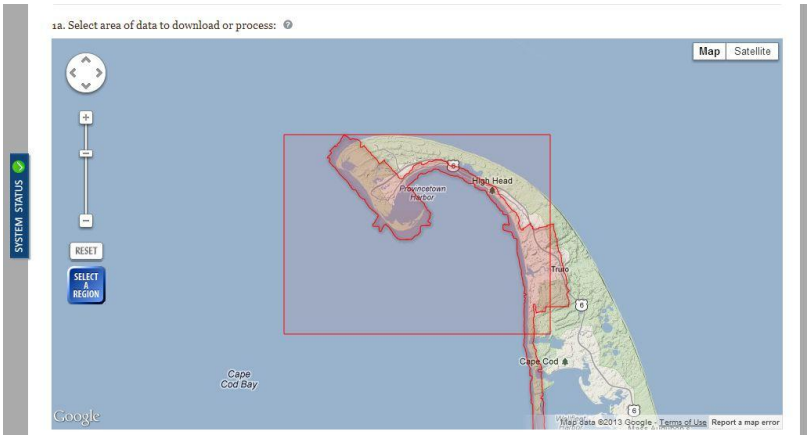

Figure 5 : Selecting the area for data

There are number of tools listed such as GRASS (Geographic Resources Analysis Support System), GDAL (Geospatial Data Abstraction Library). Some of those tools have discussed earlier. Similarly gateway provides a common portal to community support and education support.

Equally science gateways play a major role in geoscience domain. After doing some processes newly processed data can be available for other scientists via MYGEON. Many other science gateways also provide such personalization on work carried out.

2.6.1. Limitations

E-science has become an advanced tool provider for scientists, allowing them to easily engage in research and experiments which require high computational power and resources. However there are certain features that have not been fully realized through the enhancements of technological support for geoscience computing. Furthermore less focus on end-user interactions has caused negative impacts in achieving the expected outcome of these modern tools. Below we discuss the discovered weaknesses of existing systems.

2.6.1.1. Requisite for computational expertise

Geoscience experiments often deal with extreme technologies involving complex computations running in HPC. This makes end users reluctant to use the available sophisticated tools due to lack of required geoscience knowledge and expertise in underlying computational infrastructure. An ideal solution should focus on delivering a user friendly environment by hiding underlying complex computational resources that enable users to get familiar with the tool set with low computational expertise. Instead the currently popular applications include a steep learning curve. To get the maximum benefits of scientific experts, they should be allowed to focus on experiment and calculations without concerning about handling underlying resources and infrastructure.

2.6.1.2. Regenerating common geoscience computations

Several popular Workflow Management Systems are used by the science communities including Kepler, Triana, Taverna and Pegasus. These systems are used by different scientific communities according to their requirements. They differ in several aspects including type of workflow engines and workflow description languages. Therefore a workflow created for a certain workflow management system is not compatible to be used in another system [34].

Lack of reuse can be identified as a cause for significant drawback in scientific advancements. For an example, Pegasus and Taverna have been used as workflow management tools in many bioinformatics projects [40][26]. Therefore it is very likely that a project using Pegasus has already constructed a workflow that can be useful in the project implemented using Taverna. But due to the differences and incompatibilities in workflow description languages and workflow formalisms etc. a workflow cannot be reused [62]. Therefore each set of scientists has to construct similar workflows according to their own workflow management system. In many of the existing workflow management systems scientists has to deal with low level computational infrastructures to some extent. If the workflows can be shared and reused among different scientific communities dealing with similar problems it would tremendously contribute to the advancement in scientific research and experiments in the respective domains.

2.6.1.3. Difficulties in handling data-intensity

Geosciences as with the other sciences, most researches and experiments are heavily engaged with massive data sets. As computations grow in complexity running on larger computers, the reliable analysis of the data they generate becomes more challenging [14]. When it comes to geoscientists, effective processing of geospatial data is a major requirement. But currently software support available for handling and processing geospatial data is a major concern. Geo scientific research and discovery is delayed or missed due to unavailability of data intensive handling resources. Every geoscientist in the world does not have the privilege of accessing high performing super computers to deal with high data intensive geo spatial experiments. This has become a major concern where talented researchers do not get the benefit of performing high data intensive applications. Existing software and technologies aren't perfect and more comprehensive analysis data would help in the discovery and identification of unanticipated phenomena, and also help expose shortcomings in the simulation methodologies and software. Gap between computations and I/O operations often affects to data generation rates and scientists struggle to reduce their data output to cost effective data writing at runtime and analyze the data subsequently [14]. Simply data access and processing operations dominate the power of the computational resources. Climate sciences are part of geosciences climate change research isn’t only a scientific challenge of the first order, but also a major technological and infrastructure challenge [14].

2.6.1.4. Difficulty in effective tool selection

There are lots of tools used by Geoscience experiments related to various research requirements. Some of these tools are publicly available and some are not. Some of them are online tools while majority are in the form of downloadable client. There are separate tools for different aspects of

geosciences. Some of these are Databases, web services based tools using desktop clients, web services based tools using on browser clients, geo processing applications, spatial tools which focus on co-ordinates handling, navigation and mapping tools, crisis management tools, mapdata and geospatial libraries.

For an example if some novice researcher needs to do some simple co-ordinate transformation where should he go? Answer is definitely hidden among the vast number of tools available online. But surfing internet is not enough to capture the correct tool.

There are larger collection of open source geospatial software in OpenSourceGIS [63] and many other proprietary software systems. Even the OSGeo-Live DVD [64] a collection of software which implements OGC standards contains more than 60 different software systems. It is practically not possible to go through each and every one of them to check their capabilities.

2.6.2. Computational geoscience concerns

As we identified the limitations of existing technological support in geoscience domain, it is also necessary to find remedies to overcome these limitations and challenges. In this section we explore the features that should be presented through technology towards geoscience in order to achieve a growth in data analysis and understanding, proportional to the exponential growth in computing, data storage and other performance elements.

2.6.2.1. Spatial data visualization

Geospatial data visualization capabilities are one of major requirements in geoscience world. Distributed GeoVisualization systems enable collaborative synchronous and asynchronous visual exploration and analysis of geospatial data via the Web, Internet, and large-screen group-enabled displays [62]. The results of high data intensive, long running geoscience experiments itself make no sense until the output data is mapped into a visualization tool for viewing and further analysis. Geoscientists often use visualization tools in their experiments where they maps large amount of data sets to geographic maps available in visualization tools to visually analyze geo spatial data sets. There are two major types of geo spatial data available in the world which are known as raster and vector data [65]. Geospatial data formats act a vital role when it comes to visualize data. There are well recognized visualization tools, both browser-based and desktop available as commercial and open source which accept different geospatial data formats for visualization aspects. These tools are capable of visualizing data in 2 dimensions up to 4 dimensions.

2.6.2.2. Collaboration

Modern geoscience experiments are consisted of large scale researches. There is an involvement of lot of stakeholders in this process. NASA Jet Propulsion laboratory recently had an attempt to enhance its usability of observational data. Its main focus was for NASA level 2 observatory data products.This program was called CDX. It involves community from JPL and Program For Climate Model Diagnosis And Intercomparisons (PCMDI [66]). This kind of collaborations has resulted in more formalization and standardization of data products. Simultaneous analysis of the

data is hence needed. In the above scenario there are different ways that these products differ. Another good example is oil drilling, where geo scientists have to deal with engineers and come up with solutions. These two groups of people are in two different domains. There are instances that both of these working groups use the same tools. In this situation target company will have to take the responsibility to train both groups. Linking all the computational resources with each other makes a heterogeneous computing environment to work with. This makes a lot of overhead in a large scale project.

Each of the research group uses their own tools for analysis. Some of the commonly used formats are ESRI shape files, GML files, Microsoft Excel files and HTML files as well [14].Since data distribution also so wide spread, it is a desirable feature of converting of these formats to one another. This is not only a format conversion, but also there is an issue with designing common model for data which supports simultaneous access of databases and interoperability between heterogeneous databases.

Stakeholders of this area are most probably working in a university in geology, physics or biology department while geophysicist might be with larger technical staff. It is obvious that those people have differing software experience and different expectation for a common cyber infrastructure. For collaboration to be successful it needs to have simple interfaces and well documented step by step processes.

2.6.2.3. Reuse

Sharing information about researches and experiments has been in practice among scientists for a long time. Medium of sharing has evolved with the technology from pen and paper to digital cameras, e-mail, Web and software [33]. From scientists' perspective, process sharing can contribute to accelerate the advancements in science research and discovery. Alternatively it will help to broaden the knowledge of students by sharing experimental resources [67].

Today in SOA domain, service computation is a key mechanism which allows creation of value-added services by integrating and reusing existing services [20]. This mechanism exposes multiple reusable services to a wider community to be consumed for different purposes. Interoperability issues related to reuse can be overcome by adapting specific standards for modeling geospatial data and standards for service interfaces [20].

2.6.2.4. Scalability

Scalability is a fundamental requirement for large-scale scientific experiments and has a great impact on the application performance and completion. The recent progress in modeling and virtualizing technologies has paid reasonable effort for utilizing resource pools for on demand and scalable scientific computing. The nature of geoscience experiments is changing of resource quantities and characteristics to vary at runtime. Thus integration of cloud computing into workflow engine is a desirable feature that applications need to address dynamic scaling up and down according to the resource requirements. In addition, frameworks excessively need to access a range of computational resources in a geoscience application to manage distributed

applications. Adopting the current tools to benefit from cloud-specific services and proliferating of cloud specifically for science applications are emerging in academia.

2.6.2.5. Publishing and Retrieving Services information

Capability of publishing service information and retrieving the metadata of a service is a major requirement of the system. Geoscience experiments are highly depended on available online processing services due to the need of high performance computational resources. "Discovering suitable geo-processing services is a major challenge in this endeavor" [68]. System needs to facilitate service providers to publish their services along with service descriptions. Service provider can be a researcher who publishes his services which can be used by some other researcher to support with his experiments. Service provider provides service endpoints along with input and output formats of service parameters. Along with emerging SOA, the online Geo Information(GI) Systems ,data and processing services are increased [68]. Standard way to publishing information about services is a major necessity of the system where it can be effectively used in building applications and workflows with existing service information rather than searching for available services from different places through Internet.

2.6.2.6. Ease of use

Nowadays scientists are provided with sophisticated high level tools to deal with complex computations and experiments. One of the weak points that need to be satisfied on scientific research tools is its complexity and difficulty of adapting to the technology. Generally the resource sharing and scalable facilities, accessing Grid infrastructure need to have steep learning curve and many efforts have then to be made focusing on them. Thus while expanding the facilities and capabilities of the applications, users should provide in the easiest way with the assistance of intuitive graphical user interfaces [69]. For an ideal gateway researches could have focus totally on their research studies with little concern of the underline technology. To achieve this status the application need to hide the complexity and expose user to desired set of features in an accessible manner.

2.6.2.7. Security

Security is the one of key features in a framework which support simple access to cyber infrastructure (CI) by providing advanced interfaces to collaboration, analysis, data management, and other tools for students and researchers [70]. Providing a collaborative environment and distributed resource access from one place often increases the need of an efficient security principle within a framework. Systems need to access the resources, computing cycles, instruments and data on researcher's behalf. Resource access often requires use of the researcher's security credentials; in some cases exposing the researcher's long-lived password to potential compromise in the system [70]. Therefore effective handling of security in framework is a major requirement. Usually systems tend to access grid and cloud computing resources to handle grid and cloud security. Users demand that their research and personal information are kept protected, and resource providers demand that their computing and storage resources are used appropriately by authorized users [70]. When it comes to collaborative development of

workflows and experiment, it often requires properly maintaining research resources and workflow files in a manner which does not expose critical information about a particular research to the unprivileged. OAuth protocol is widely adopted in providing a security in science gateway systems.

2.6.2.8. Reproducibility

Reproducibility is a key factor in scientific analysis and processes. It enables scientists to evaluate the validity of each other's hypothesis and finalize [33]. Provenance information on data derived and derivation procedure is essential in order to achieve reproducibility in scientific computations. Provenance information, as mentioned earlier, includes important information on preserving data, determining the data's quality and authorship and reproducing results [7] .

In a highly collaborative environment, where many scientists are involved provenance data gets highly fragmented in e-mails, wikis, journal references etc. [33]. Therefore provenance information vanishes especially while sharing processes. Workflow-management systems must capture and generate provenance information as a part of the workflow-generated data. Nevertheless it is critical to share provenance information systematically and explicitly with the process sharing instances as well.

2.6.2.9. Interoperability

Issues of service and data discovery, composition, communication within end users and distributed services are critical in the application domain. Thus following proper standards on interfaces will avoid any restriction, access or additional implementation in the future. Adhering to a set of widely accepted standards rather than limiting to a unique module, accommodates accessing remote services in a shared manner. For an instance OGC has provided specific interface descriptions for web services. Such a solution should allow scientists to communicate with service instances, connect to corresponding distributed services and directly invoke available OGC-based services via OGC standard specifications [20]. Hence it is important to serve perceived needs of expert users while maintaining common service accessing principles.

2.6.3. Framework

Table 3 offers categorization of above analyzed tools and application on the basis of their support to accomplish discussed features. Further it gives an overview on the technical approach under architectural and application level concerns.

Architectural Concerns	Type of tool	Approach	Limitations
Spatial data visualization	Desktop GIS	Provide views by rendering different types of images, animations and raster maps	Difficulties in handling heterogeneous computing environments
	Web Map Servers	Publish geospatial data through standards which can be used by visualization software and tools such as Google Maps	Limited OGC support (separate plugins are required in some instances)
	Software development frameworks and libraries	Facilitate the environment for building visualization applications by providing tools, APIs and standards support	Many software frameworks and libraries are specialized in specific areas
Collaboration	Scientific workflow management systems	Provide a platform for scientists to collaboratively create, execute and monitor workflows by using the services provided by various communities	Complex to build, No unified method of generating workflow, Limited to specific domains
Reuse	Scientific workflow management systems	Create interoperable workflows with other software systems by generating generic representations of workflows	Difficulties in service discovery and finding appropriate process
Ease of use	All tools	Hide complex computational resources and provide user friendly environments, Support universal standards, Provide standard APIs while hiding complex processing algorithm	Complexity of the domain make it difficult to create intuitive interfaces
Security	Scientific Workflow Management systems	Use security frameworks and security protocols	No framework addresses all the security concerns
Reproducibility	Scientific Workflow Management systems	Manage provenance information of workflows	Security concerns may raise when handling provenance data
Scalability	Scientific Workflow Management systems	Scale up and down resources according to runtime requirements with proper usage and handle cloud services	Requires substantial domain knowledge
Interoperability	All tools	Support universal standards to access services	Continuous change of standards

Table 3 : Analysis Framework for Architectural Concerns

References

[1] David Kincaid and Ward Cheney, "Numerical Analysis: What is it?," in in *Numerical Analysis: Mathematics of Scientific Computing*, 3rd ed., American Mathematical Society, 2002, pp. 1–2.

[2] Ustun Yildiz, Adnene Guabtni, and Anne H. H. Ngu, "Business versus Scientific Workflow A Comparative study," in in *Proceedings of the 2009 Congress on Services - I (SERVICES '09)*, 2009, pp. 340–343.

[3] E. Deelman and Y. Gil, "Managing large-scale scientific workflows in distributed environments: Experiences and challenges," in *e-Science and Grid Computing, 2006. e-Science'06. Second IEEE International Conference on*, 2006, pp. 144–144.

[4] V. Curcin and M. Ghanem, "Scientific workflow systems-can one size fit all?," in *Biomedical Engineering Conference, 2008. CIBEC 2008. Cairo International*, 2008, pp. 1–9.

[5] J. Chen and W. M. P. van der Aalst, "On scientific workflow," *Ieee Tech. Comm. Scalable Comput. Newsl.*, vol. 9, no. 1, 2007.

[6] S. Pandey, D. Karunamoorthy, and R. Buyya, "Workflow engine for clouds," *Cloud Comput. Princ. Paradig. R Buyya J Broberg Goscinski Eds Isbn-13*, pp. 978–0470887998, 2011.

[7] S. B. Davidson and J. Freire, "Provenance and scientific workflows: challenges and opportunities," in in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 2008, pp. 1345–1350.

[8] I. Altintas, J. Wang, D. Crawl, and W. Li, "Challenges and approaches for distributed workflow-driven analysis of large-scale biological data: vision paper," in *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, 2012, pp. 73–78.

[9] N. Wilkins-Diehr and K. A. Lawrence, "Opening science gateways to future success: The challenges of gateway sustainability," in in *Gateway Computing Environments Workshop (GCE), 2010*, 2010, pp. 1–10.

[10] "What is Geoscience?," *The Department of Geosciences*. [Online]. Available: http://www3.geosc.psu.edu/prospective/whatis.php. [Accessed: 10-Apr-2013].

[11] "Geoscience Career Frequently Asked Questions," *AGI American Geosciences Institute*. [Online]. Available: http://www.agiweb.org/workforce/faqs/#1.1. [Accessed: 10-Apr-2013].

[12] P. Zhao, L. Di, and G. Yu, "Building asynchronous geospatial processing workflows with web services," *Comput. Geosci.*, vol. 39, pp. 34–41, 2012.

[13] S. Marru, L. Gunathilake, C. Herath, P. Tangchaisin, M. Pierce, C. Mattmann, R. Singh, T. Gunarathne, E. Chinthaka, and R. Gardler, "Apache airavata: a framework for distributed applications and computational workflows," in *Proceedings of the 2011 ACM workshop on Gateway computing environments*, 2011, pp. 21–28.

[14] J. Sankaranarayanan, E. Tanin, H. Samet, and F. Brabec, "Accessing diverse geo-referenced data sources with the SAND spatial DBMS," in *Proceedings of the 2003 annual national conference on Digital government research*, 2003, pp. 1–4.

[15] "Geoscientists Canada." [Online]. Available: http://www.ccpg.ca/main/index.php?lang=en. [Accessed: 20-Apr-2013].

[16] "Australian geo science standard." [Online]. Available: http://www.ga.gov.au/products-services/data-applications/data-standards-symbols/geoscience-data-standards.html. [Accessed: 20-Apr-2013].

[17] "Group on earth observations." [Online]. Available: http://www.earthobservations.org/geoss.shtml.

[18] "Geoscience BC, Project 2011-012." [Online]. Available: http://www.geosciencebc.com/s/2011-012.asp. [Accessed: 22-Apr-2013].

[19] M. Owonibi and P. Baumann, "Heuristic geo query decomposition and orchestration in a SOA," in *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, 2010, pp. 890–894.

[20] C. Granell, L. Díaz, and M. Gould, "Service-oriented applications for environmental models: Reusable geospatial services," *Environ. Model. Softw.*, vol. 25, no. 2, pp. 182–198, Feb. 2010.

[21] X. Shi, "High performance computing: fundamental research challenges in service oriented GIS," in *Proceedings of the ACM SIGSPATIAL International Workshop on High Performance and Distributed Geographic Information Systems*, 2010, pp. 31–34.

[22] Xiaoyan Li, Liping Di, Weiguo Han, Peisheng Zhao, and Upendra Dadi, "Sharing geoscience algorithms in a Web service-oriented environment (GRASS GIS example)," *Comput. Geosci.*, vol. 36, no. 8, pp. 1060–1068, Aug. 2010.

[23] M. Zapatero, R. Castro, G. Wainer, and M. Houssein, "Architecture for integrated modeling, simulation and visualization of environmental systems using GIS and cell-devs," in *Proceedings of the Winter Simulation Conference*, 2011, pp. 997–1009.

[24] T. C. Vance, N. Merati, S. M. Mesick, C. W. Moore, and D. J. Wright, "GeoModeler: tightly linking spatially-explicit models and data with a GIS for analysis and geovisualization," in *Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems*, 2007, p. 32.

[25] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock, "Kepler: an extensible system for design and execution of scientific workflows," in *Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on*, 2004, pp. 423–424.

[26] "Taverna Official Site." [Online]. Available: http://www.taverna.org.uk/. [Accessed: 02-Apr-2013].

[27] "Myexperiment." [Online]. Available: http://www.myexperiment.org/. [Accessed: 16-Apr-2013].

[28] "BioCatalogue." [Online]. Available: http://www.biocatalogue.org. [Accessed: 15-Apr-2013].

[29] P. Fisher, "An Introduction to Web Services and Scientific Workflows."

[30] "Pegasus: workflow management system." [Online]. Available: http://pegasus.isi.edu/. [Accessed: 20-Apr-2013].

[31] E. Deelman, "The Pegasus Workflow Management System." 2008.

[32] N. Mandal, E. Deelman, G. Mehta, M.-H. Su, and K. Vahi, "Integrating existing scientific workflow systems: the Kepler/Pegasus example," in *Proceedings of the 2nd workshop on Workflows in support of large-scale science*, 2007, pp. 21–28.

[33] Yolanda Gil, Ewa Deelman, Mark Ellisman, Thomas Fahringer, Geoffrey Fox, Dennis Gannon, Carole Goble, Miron Livny, Luc Moreau, and Jim Myers, "Examining the Challenges of Scientific Workflows," *Computer*, pp. 24–32, 2007.

[34] Tamas Kukla, Tamas Kiss, Gabor Terstyanszky, Peter Kacsuk, and Gergely Sipos, "Enabling the execution of various workflows (Kepler, Taverna, Triana, P-GRADE) on EGEE."

[35] Cui Lin, Shiyong Lu, Xubo Fei, A. Chebotko, Darshan Pai, Zhaoqiang Lai, F. Fotouhi, and Jing Hua, "A Reference Architecture for Scientific Workflow Management Systems and the VIEW SOA Solution," *Ieee Trans. Serv. Comput.*, vol. 2, no. 1, pp. 79–92, Jan. 2009.

[36] B. Ludäscher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao, "Scientific workflow management and the Kepler system," *Concurr. Comput. Pr. Exp.*, vol. 18, no. 10, pp. 1039–1065, 2006.

[37] "Apache Airavata." [Online]. Available: http://airavata.apache.org/. [Accessed: 17-Apr-2013].

[38] "Triana." [Online]. Available: http://www.trianacode.org/index.html. [Accessed: 17-Apr-2013].

[39] "Kepler." [Online]. Available: https://kepler-project.org/. [Accessed: 17-Apr-2013].

[40] “Pegasus.” [Online]. Available: http://pegasus.isi.edu/. [Accessed: 17-Apr-2013].

[41] “Overview of computational science.” [Online]. Available: http://www.shodor.org/chemviz/overview/compsci.html. [Accessed: 11-Apr-2013].

[42] Preston Smith, “Northwest Indiana Computational Grid.” [Online]. Available: https://indico.fnal.gov/getFile.py/access?contribId=28...10... [Accessed: 26-Apr-2013].

[43] “Open Science Grid,” *Open Science Grid*. [Online]. Available: https://www.opensciencegrid.org/. [Accessed: 11-Apr-2013].

[44] “TeraGrid Archives,” *XSEDE : Extreme Science and Engineering, Discovery Environment*. [Online]. Available: https://www.xsede.org/tg-archives. [Accessed: 13-Apr-2013].

[45] “Advanced Scientific Computing Research (ASCR),” *Advanced Scientific Computing Research (ASCR)*. [Online]. Available: http://science.energy.gov/ascr/. [Accessed: 13-Apr-2013].

[46] D. Gannon, “Programming E-Science Gateways,” in in *Making Grids Work*, Springer, 2008, pp. 191–200.

[47] D. Gannon, B. Plale, S. Marru, Y. Simmhan, and S. Shirasuna, “Dynamic, adaptive workflows for mesoscale meteorology,” *Work. E-Sci.*, pp. 126–142, 2007.

[48] “Airavata Architecture.” [Online]. Available: http://airavata.apache.org/architecture/overview.html. [Accessed: 30-Apr-2013].

[49] “About OGC.” [Online]. Available: http://www.opengeospatial.org/ogc. [Accessed: 21-Apr-2013].

[50] “CITE tests,” *CITE tests*. [Online]. Available: http://cite.opengeospatial.org/teamengine/. [Accessed: 18-Apr-2013].

[51] Adam Leadbetter and Roy Lowry, “NETMAR : Open service network for marine environmental data.” .

[52] “Web Coverage Service.” [Online]. Available: http://en.wikipedia.org/wiki/Web_Coverage_Service. [Accessed: 30-Apr-2013].

[53] N. H. Vaidya, M. N. Mehta, C. E. Perkins, and G. Montenegro, “Delayed duplicate acknowledgements: a TCP-Unaware approach to improve performance of TCP over wireless,” *Wirel. Commun. Mob. Comput.*, vol. 2, no. 1, pp. 59–70, 2002.

[54] “GeoServer.” [Online]. Available: http://geoserver.org/display/GEOS/Welcome. [Accessed: 18-Apr-2013].

[55] “OpenLayers,” *OpenLayers*. [Online]. Available: http://openlayers.org/. [Accessed: 13-Apr-2013].

[56] “ZOO 1.2 Documentation.” [Online]. Available: http://zoo-project.org/docs/. [Accessed: 20-Apr-2013].

[57] “PyWPS documentation.” [Online]. Available: http://pywps.wald.intevation.org. [Accessed: 13-Apr-2013].

[58] M. Neteler, “Introduction to GRASS.” .

[59] “GRASS GIS.” [Online]. Available: http://grass.osgeo.org/. [Accessed: 30-Apr-2013].

[60] “QuakeSim,” *QuakeSim understanding the earthquake process*. [Online]. Available: http://quakesim.org/. [Accessed: 04-Dec-2013].

[61] “LIDAR.” .

[62] T. M. Rhyne, A. MacEachren, and T.-M. Rhyne, “Visualizing geospatial data,” in *ACM SIGGRAPH 2004 Course Notes*, 2004, p. 31.

[63] “OpenSourceGIS.” [Online]. Available: http://opensourcegis.org/. [Accessed: 18-Apr-2013].

[64] “OSGeoLive.” [Online]. Available: http://live.osgeo.org/en/index.html. [Accessed: 28-Mar-2013].

[65] P. Capolsini and A. Gabillon, “Security policies for the visualization of Geo Data,” in *Proceedings of the 2nd SIGSPATIAL ACM GIS 2009 International Workshop on Security and Privacy in GIS and LBS*, 2009, pp. 2–11.

[66] Chris A. Mattmann, Amy J. Braverman, and Daniel J. Crichton, “Understanding architectural tradeoffs necessary to increase climate model intercomparison efficiency,” in in *ACM SIGSOFT Software Engineering Notes*, 2010, pp. 1–6.

[67] N. Wilkins-Diehr, D. Gannon, G. Klimeck, S. Oster, and S. Pamidighantam, “TeraGrid Science Gateways and Their Impact on Science,” vol. 41, pp. 32–41, 2008.

[68] M. Lutz, “Ontology-based service discovery in spatial data infrastructures,” in *Proceedings of the 2005 workshop on Geographic information retrieval*, 2005, pp. 45–54.

[69] R. Barbera, G. La Rocca, R. Rotondo, A. Falzone, P. Maggi, and N. Venuti, “Conjugating Science Gateways and Grid Portals into e-Collaboration environments: the Liferay and GENIUS/EnginFrame use case,” in *Proceedings of the 2010 TeraGrid Conference*, 2010, p. 1.

[70] J. Basney, R. Dooley, J. Gaynor, S. Marru, and M. Pierce, “Distributed web security for science gateways,” in *Proceedings of the 2011 ACM workshop on Gateway computing environments*, 2011, pp. 13–20.