International Innovative Research Journal of Engineering and Technology
ISSN NO: 2456-1983

International Conference on Emerging Innovation in Engineering and Technology
ICEIET-2017

An Architectural Framework for Scheduling and executing large scientific workflows in cloud environment

S. Balamurugan[1], S. Saraswathi[2]

[1]Assistant Professor, Department of Information Technology,
Perunthalaivar Kamarajar Insitute of Engineering and Technology(PKIET), Karaikal, India.
[2]Professor, Department of Information Technology,
Pondicherry Engineering College(PEC), Puducherry, India.
[1]balaprof@gmail.com
[2]swathi@gmail.com

Abstract— Execution of scientific workflows in a Cloud environment is an evolving paradigm for modeling and formalize scientific applications to speed up the scientific discovering [1] process. These days' cloud computing is getting more popular in the field of data science with which applications which involve large amount of scientific data in the pattern of scientific workflows are scheduled and executed in cloud resources. Scientists are working on large datas and trying various methods to deal with analyzing the large volume of datas. The popular examples for scientific applications are cybershake, montage, Epigenomics etc. Scheduling the execution of such scientific workflows is a major issue in the area of data science because it involves various essential supporting tasks such as management of data and task dependencies, task scheduling and execution, provenance tracking etc., to scientific applications. So, it is very important to design a scientific workflow management system for the cloud environment to overwhelm the large set of scientific data and the complexity involved in analyzing the scientific data. In this paper we proposed a referral architecture for managing the execution of scientific workflows into cloud platform which can be used as a reference model for developing application which schedules the workflow on the cloud.

Keywords—Workflow; data-science; scientific applications; resource management; architectural framework; workflow scheduling; provenance tracking.

I. INTRODUCTION

Cloud computing [2] has been a growing interest and a possible solution in almost every field providing a flexible, on-demand computing infrastructure for number of applications. As cloud computing is considered as an emerging computing paradigm, it is gaining remarkable impetus in both academic and industry. Many industrial and scientific communities are facing the situation where the volume of new data being generated is overwhelming the capacity of those industries to manage it. Usually the data comes from products, sensors, satellites experiments and simulations. Scientists who analyze the large data are attempting many types of methods to deal with increasing volume of data and storage issues.

Today's scientific applications became even more complex. These applications involve different types application modules, developed by different developers which targets heterogeneous set of resources. The applications also involve many computational steps that may require custom execution environments. These applications process very large volume of data and generate large results. Apart from these, the applications also face computational challenges. These Applications need to be designed so that it can take the benefit of small and completely encapsulated modules. It is very important that these applications need to execute the computations consistently and efficiently among any number and type of resources in the Cloud. During the execution process there may exist tradeoff between cost, availability, reliability, and ease of use and access.

One of the best possible resolution to the management of applications in heterogeneous execution environments is to structure the application as a workflow and let the workflow management system [3] manage the scheduling and execution of the any scientific application in different computing environments. Workflows establishes relationship between different computational tasks and bind the modules together and solemnize the order in which the tasks need to complete the execution.

In our work, we are proposing an architectural framework for executing scientific applications in cloud environment. This architecture can be used to map and execute complex scientific workflows on a number of different resources. In this context, the application is described in terms of logical components and logical data and the interaction between different modules. Also the application depiction is independent of the execution platform. Mappings can be developed that can pick the right type of resources in a

number of different execution environment, that can improve workflow execution, and that can recover from execution failures. Section II describes different types of scientific applications that are most prevalent today, Section III describes architectural requirements and Section IV concludes the paper with future scope.

II. SCIENTIFIC WORKFLOWS

Scientific workflows permit the users to easily define multi-step computational tasks, such as retrieving data from any product or a database, reorganize the data, analyze and run. Scientific workflows are accepted useful paradigm to describe, manage, and share complex scientific analyses. Scientific workflows have arisen to challenge the problem of excessive complexity in scientific experiments and applications. Different types of tasks that can be performed in a single workflow. Each task is responsible for a small piece of functionality such that many tasks need to be chained in a pipeline in an order to obtain a workflow that performs some useful task.

The process of linking all the task of a single e-science application is known as workflow composition, as a result this composition a conceptual model of any given scientific application is developed. This conceptual model is often represented as a graph-like structure called Directed Acyclic Graph (DAG) that defines the flow of data within a single workflow. In DAG the nodes are tasks and the edges denote the task dependencies. In a simple manner we can say a scientific workflow is that it manages data flow and dependencies of the tasks. The tasks in a scientific workflow can be varied from small serial tasks to very large parallel tasks. Here is some of the very popular workflow that are most predominant.

A. Cybershake

The Cybershake [4] workflow is used to characterize earthquake hazards in a region using the Probabilistic Seismic Hazard Analysis (PSHA) technique. The Cybershake DAG is as shown in the fig. 1. This workflow is used by used by the Southern California Earthquake Center (SCEC). CyberShake workflows has more than more than 800,000 jobs to execute.

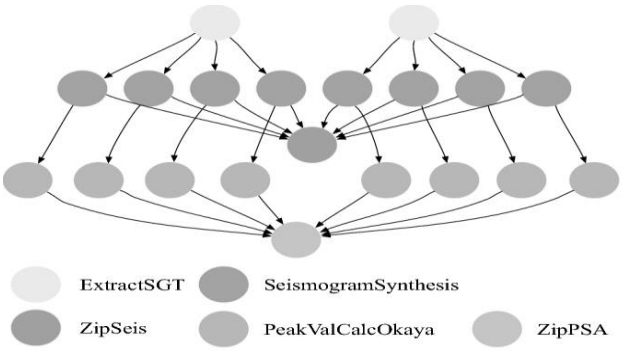

Fig1. Cybershake workflow DAG

Cybershake has seven processing steps, each processing steps has specific computational and workflow implications. 1. Select a hazard curve of interest. 2. Use Earthquake Rupture Forecast to identify all the possible ruptures in the range of 200 km from the hazard curve of interest. 3. Convert ERF into suite of rupture variation. 4. Calculate Strain Green Sensors (SGT) for two horizontal components from the data containing all the ruptures, and then save the data. 5. Calculate Seismogram Synthesis for each rupture variation. 6. Calculate Peak intensity measure of interest. 7. Using the Peak Intensity measures and rupture probabilities, calculate the probability hazard curve.

B. Ligo

LIGO Stands for Laser Interferometer Gravitational Wave Observatory (LIGO) [4]. The LIGO workflow DAG is as shown in fig. 2. This workflow attempts to detect gravitational waves produced by several natural events in the universe. LIGO is the world's largest gravitational wave observatory and a cutting edge physics experiment. The LIGO workflow is used to analyze the huge volume of data obtained from the merging of compact binary systems such as binary neutron stars and black holes. LIGO is an aspiring effort to detect gravitational waves produced by powerful events in the universe, such as collision of two black holes, or outburst of supernovae. Approximately more than one Terabyte of data is recorded per day which is analyzed by scientists in a partnership from four different continents. LIGO and its workflow has already developed over the past decade and the analysis policies accepted by LIGO scientists have been strongly inclined by the increasing complexity of tools to manage distributed resources and the workflows that run of the resources.

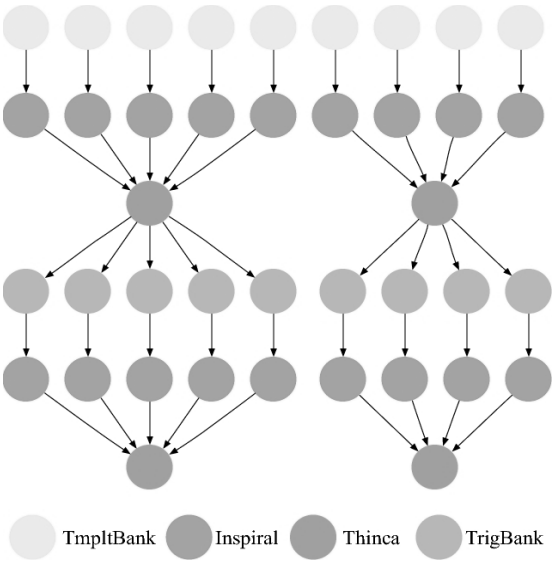

Fig2. Ligo workflow DAG

C. Epigenomics

The Epigenomics [4] workflow is created by USC Epigenome Center. The Epigenomics workflow DAG is

as shown in fig. 3. It is currently involved in the mapping of the epigenetic state of human cells on a genome-wide scale. It is used to automate various operations in genome sequencing processing. This workflow is being used by the Epigenome Center in the processing of production DNA methylation and histone modification data.

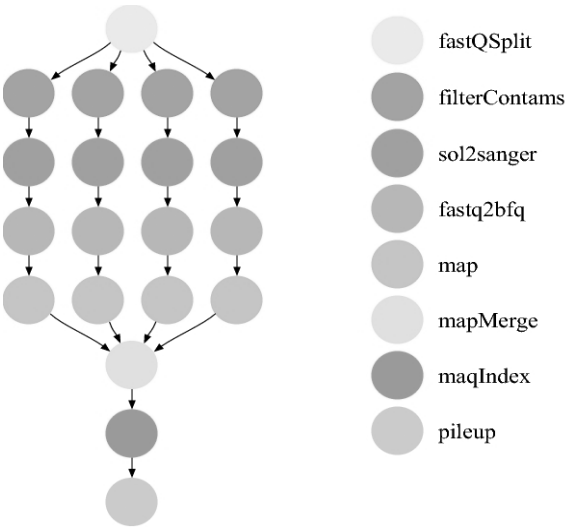

Fig3. Epigenomics workflow DAG

D. Montage

Montage [4] is created by the NASA/IPAC Infrared Science Archive used to generate custom mosaics of the sky using input images in the Flexible Image Transport System (FITS) format.

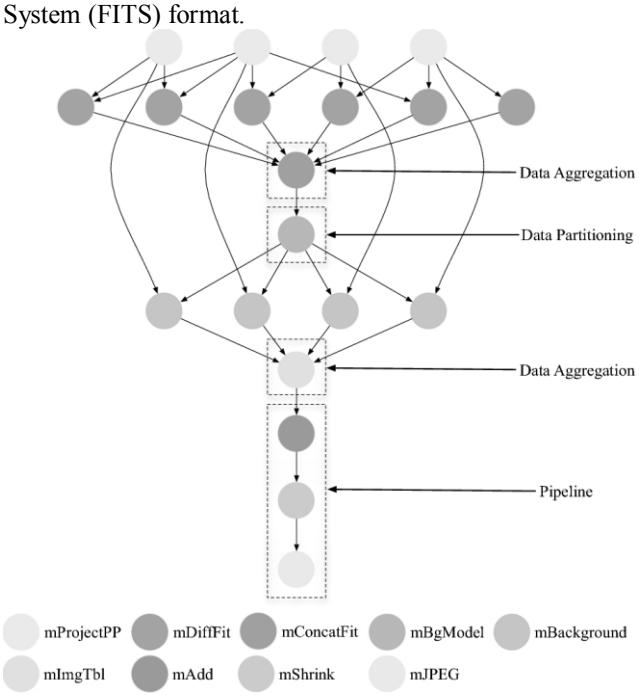

Fig4. Montage workflow DAG

Fig. 4 shows the DAG of montage workflow. During the production of the final mosaic, the geometry of the output is calculated from the geometry of the input images. The inputs are then re-projected to be of the same spatial scale and rotation. The background emissions in the images are then corrected to be of the same level in all images. The re-projected, corrected images are co-added to form the final mosaic. The Montage application has been represented as a workflow that may be executed in Grid and cloud environments.

In the astronomy field there are various projects that uses telescopes to collect high resolution images covering the entire sky. These sky images are stored in centralized repositories and databases to facilitate research by the science community. The primary users of these databases are astronomers interested in studying images of celestial objects such as galaxies or nebulas. Individual images stored in some survey databases often cover only a portion of an object being studied. To create a complete picture of the object multiple images, need to combined into a mosaic. Montage is a set of image processing tools that enable the creation of high quality image mosaics by stitching together smaller images.

E. SIPHT

The SIPHT stands for the sRNA identification protocol using high-throughput technology (SIPHT) [4]. DAG for SIPHT is shown in the fig. 5. It is the bioinformatics project developed by Harvard University and it is conducting a wide search for small untranslated RNAs (sRNAs) that regulate several processes such as secretion or virulence in bacteria. SIPHT uses a workflow to automate the search for sRNA encoding-genes for all of the bacterial replicons in the National Center for Biotechnology Information (NCBI) database.

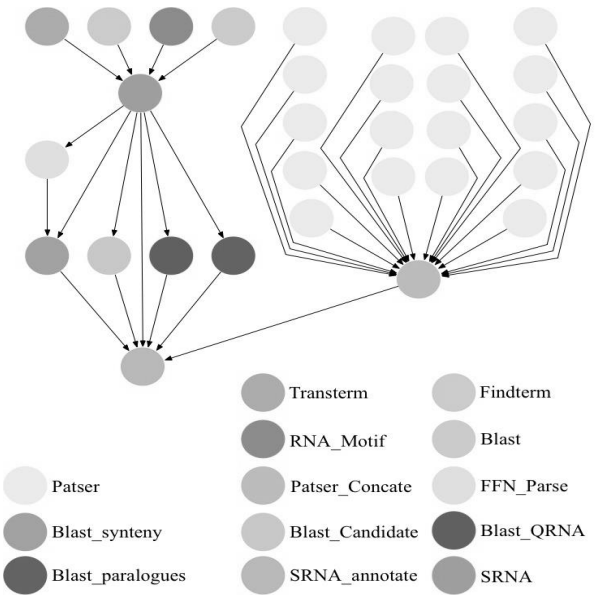

Fig5. SPIHT workflow DAG

III. ARCHITECTURAL REQUIREMENTS

Based on the comprehensive study of various workflow literature from the perspective of framework and architecture, we have come up with the following key

architectural requirements to the scientific workflow execution in cloud. The requirements are Workflow Instance, Workflow composition, Mapping, Resource computation, Execution, Provenance composition and Provenance capture . The Architectural framework of workflow management system is as shown in the figure 6. The architecture shows the various modules and their interaction between modules of the workflow management system.

A. Visualization and Presentation

Visualization and Presentation [5] module is used for the user to interact with the system and provide the supporting functionalities. In most cases only the users will design, modify and run the scientific workflows. The Workflow management system will be more usable only if the visualization and presentation of the system provides user friendly interfaces. This module should also support capability that are very specific to domain to support the overall capability of the system. Our main aim of the design is to easily do the process of achieving a proper workflow design with necessary parameters, values and input data sets. Therefore, a crucial architectural requirement is the flexibility of customizing the user interface according to different user and technological need.

B. Workflow Composition

The Workflow Composition module is responsible for the design and modification of scientific workflows. When appropriate tasks and datasets are given through the presentation module, this module generates a workflow in terms of specifications and dependencies. This representation is generally given in any specification language that the workflow model supports. Also, a scientist can design a workflow using a standalone application or web designing method with supports either graphical or scripting methods. After the workflow composition, it produces the instance of workflow in general specification language such as scripting or graphical representation.

C. Workflow Engine

Workflow Engine is the heart of the workflow management system. Workflow engine is as shown in the diagram fig.6 it coordinates the activities of workflow composition, workflow monitoring, provenance management and computing resources
modules. The main task of workflow engine is to get the specification from workflow composition module and translate it into executable workflow representation. Another important task of workflow engine is to separate the control flow and data flow of the scientific workflow scheduling so that it will improve the overall performance of workflow execution. After doing the translation part and separation and dataflow and control flow part, this module is responsible for finalize the schedule for the execution according the QoS requirement of the user. The Workflow engine will hold all the scheduling algorithms designed by scientists and automatic selection of scheduler will be done according the user requirement. Workflow engine coordinates its task with the workflow monitoring modules to take corrective and preventive measures upon receiving failure messages from the workflow monitoring module.

D. Workflow Monitoring

Workflow monitoring module is responsible for the monitoring the execution of workflow and provides the mechanism for failure handling. Many scientific workflows are designed and run normally in unclear manner for the purpose of experiments. In such cases it is very important for a workflow management system to have a control on the executions by monitoring the status of the workflow, handle failures and reporting the status of the system. Therefore, a very crucial architectural requirement is to provide support for the status and failure monitoring at different level and provides mechanism for storing logs and localizing the error automatically.

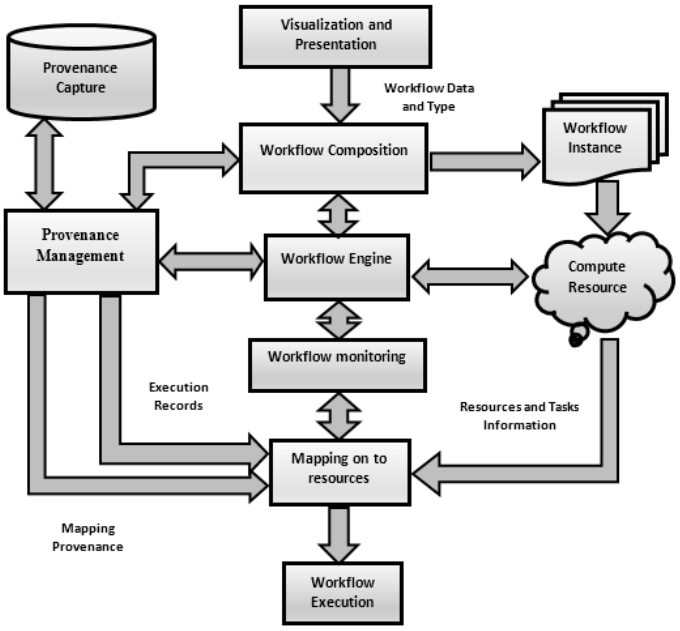

Fig6.Architectural framework for Workflow Management System

E. Provenance Management

Generally, provenance is all about the information of any particular entity, activities and task. Here the provenance management subsystem [6] is responsible for the collection of data about the workflow and its execution and maintain the information in terms of ontologies. In simple terms the provenance management module will collect the workflow execution data convert it into ontological representation and store it in provenance repository. The layered architecture of provenance manager is as shown in the fig. 7. The provenance management will include four sub functions provenance capture, provenance model mapping, provenance storage and provenance querying. It consists of three layers viz. provenance manager, Provenance model mapping and

provenance storage. The provenance manager layer takes care of representation of workflow run provenance by means of ontologies that fill in as vocabularies to depict and serialize provenance metadata. It is having two sub components provenance model manager [6] and querying manager. Provenance Model Manager deals with the ontologies that incorporate both general provenance vocabularies and ontologies particular to a domain which is used to represent knowledge in any specific field.

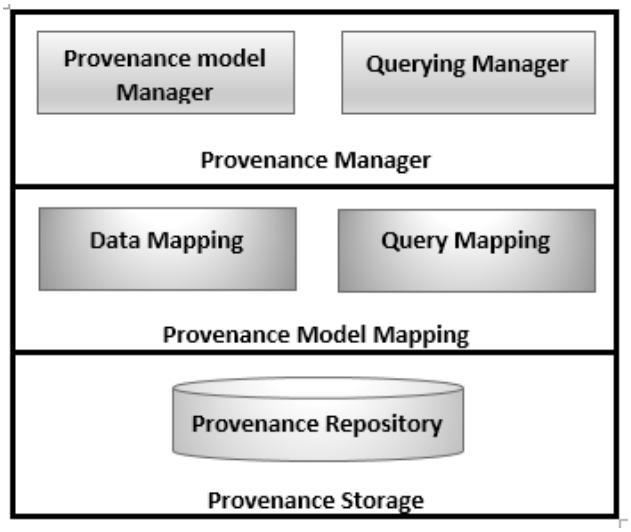

Fig7. Architecture of Provenance Manager

To address the necessities of provenance representation like interoperability, extensibility, and semantic reconciliation. We utilize Semantic Web [7] innovations for provenance representation. Specifically, the language called Web Ontology Language (OWL) [7] is used to express ontologies. OWL languages are regarded as by formal semantics and they are built upon W3C XML standard objects called Resource Description Framework (RDF). A relational RDF store is used for Provenance Querying and it is expressed by RDF query language called SPARQL [7].

The provenance model mapping acts as an intermediary medium for the provenance manager and the provenance storage. It has three different mappings, the first mapping is between OWL and Relational Schema. This mapping generates a relational database schema based on the provenance metadata. The second mapping is between RDF and relational tuples. This mapping is stored into a relational database. The third mapping is between the RDF query language SPARQL and SQL. This mapping converts the SPARQL queries into Structured Queries. These Structured Queries are then executed by Relational Data Base Management System (RDBMS) [7].

F. Compute Resources

Once the workflow engine has completed the task of finding out the necessary resources for each task, it is the job of Compute resource module to check whether the appropriate resources are available on the cloud to execute the workflow. It can lock the resources for execution if it is available. Otherwise if the resources are not available, it can wait for finite amount of time to claim the resource. Otherwise it can return the task to workflow engine to raise appropriate error.

G. Mapping onto the resources

After calculating the resources, the Compute Resource module lock the resources and give the handle to the next module called Mapping onto the resources. This module is responsible for mapping the task to appropriate resources for execution. Once the mapping is done the workflow execution starts and produces the results after the completion of workflow.

H. Provenance Repository

Provenance Repository is the Storage used to store the information related to data provenance. Data provenance is an important form of metadata which tells us how a particular data product was produced, including the system and the processing steps in the computational process. Also it gives the user responsible for its time of execution, and nature of resources used, such as settings, inputs and software tools. Provenance information also provides transparency and supports in the process of auditing and data interpretation. In general, the benefits and applications of provenance include quality, data management, fault identification, data discovery, validation, acknowledgement and reproducibility of scientific workflow executions.

I. Workflow Executor

Workflow executor supports the execution of tasks in a broad range of heterogeneous platform. This module separates the running of tasks and management of task execution. Workflow Executor can also be used in a sense of distributed workflow executor to avoid the problem of single point failures in centralized architecture. Different tasks belong to single workflow can be executed parallel at distributed system to improvise the efficiency and performance. Workflow Executor module provides the functions of run execution, run model mapping and run storage.

IV. CONCLUSION AND FUTURE WORK

Scientific workflows are the important tools in managing large scale and high throughput e-science applications. In this paper we proposed an architectural framework for the remote submission and execution of large scientific workflows in cloud environment. This architectural framework clearly shows all the modules and the interaction between different modules in processing and executing the scientific workflow. When compared to the existing Workflow Management system in the literature, the proposed framework incorporates the workflow Scheduling module integrated in the workflow engine. From which the user can select a better Scheduling criteria according to their needs. The proposed architecture also gives us a better visualization and clear separation of various modules present in the Workflow management system. Using this architectural framework one can develop a Scientific Workflow Management System

(SWMS) to support the design of workflow, submission of workflow in the appropriate cloud resources, storage and finally getting the results from the submission. This architecture is also scalable from the point of execution environment. Future work includes the incorporation of security and trust management features to the existing architecture.

V. REFERENCES

[1] Tsalgatidou, G. Athanasopoulos, M. Pantazoglou, C. Pautasso, T. Heinis, R. Grønmo, H. Hoff, A. Berre, M. Glittum, and S. Topouzidou, "Developing Scientific Workflows from Heterogeneous Services," SIGMOD Record, vol. 35, no. 2, pp. 22-28, 2006.

[2] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared," Proc. IEEE Grid Computing Environments Workshop, pp. 1-10, 2008.

[3] Cui Lin, Artem Chebokto, Zhaoquiang Lai, Farshad Fotouhi, "A Reference Architecture for Scientific Workflow Management Systems and the VIEW SOA solution", IEEE Transactions on Services Computing, Vol. 2, No.1, pp. 79-92, January-March 2009.

[4] Pegasus Documentation and workflow gallery https://pegasus.isi.edu/workflow_gallery/

[5] C. Lin, S. Lu, Z. Lai, A. Chebotko, X. Fei, J. Hua, and F. Fotouhi, "Service-Oriented Architecture for VIEW: A Visual Scientific Workflow Management System," Proc. IEEE Int'l Conf. Services Computing (SCC '08), pp. 335-342, 2008.

[6] Cuevas-Vicenttín, V, Kianmajd, P, Ludäscher, B, Missier, P, Chirigati, F, Wei, Y, Koop, D, Dey, S, "The PBase scientific workflow provenance repository", International Journal of Digital Curation, Vol. 9, Issue 2, pp. 28-38, Feb. 2014.

[7] A. Chebotko, X. Fei, C. Lin, S. Lu, and F. Fotouhi, "Storing and Querying Scientific Workflow Provenance Metadata Using an RDBMS," Proc. Second IEEE Int'l Workshop Scientific Workflows and Business Workflow Standards in E-Science, pp. 611-618, 2007.