PROCEEDINGS OF SCIENCE

Seasonal Ensemble Forecasting Application On Dependable Sumegha Scientific Cloud Infrastructure

Ramesh Naidu Laveti[1], **Prahladarao B B, Vineeth Simon Arackle, Arunachalam B**
Center for Development of Advanced computing
C-DAC Knowledge Park, Byappanahalli, Bangalore, India
E-mail: `{rameshl, vineeth, barunachalam}@cdac.in, prahlada.rao@iiitb.ac.in`

Despite several advances in understanding the behaviour of monsoon variability, innovations in the numerical modeling and the availability of higher computational capabilities, accurate prediction of Indian summer monsoon still remains a serious challenge. Seasonal Forecast Model (SFM), developed for seasonal forecast and climate research, is used for forecasting the Indian summer monsoon in advance of a season. Ensemble forecasting method helps us in finding and minimizing the uncertainty inherent in seasonal forecast. The inherent parallel nature and the bursty computational demands of the ensemble forecasting method allows it to effectively utilize the Infrastructure-as-a-Service (IaaS) model on the cloud platform. However, realizing huge scientific experiments is still a challenge to the cloud service providers as well as to the climate modeling community.

To start with prototype experiments using SFM model were conducted at T-62 resolution (~ 200 km x 200 km grid). The experience gained from the prototype runs were used by the SuMegha operational team to fine tune the configuration of SuMegha Cloud resources to improve the quality of service. High resolution SFM at T-320 (~ 37 km x 37 km grid) was also configured and experiments were conducted to understand the scalability, computational performance of the application and the reliability of SuMegha Cloud.

In this paper, we use SFM as a case study to present the key problems found by climate applications, and propose a framework to run the same on SuMegha Cloud infrastructure to allow a climate model to take advantage of these cloud resources in a seamless and reliable way. The framework uses classification and outlier detection techniques to classify the resources and also to identify the faulty resources. It addresses the challenges such as unexpected hardware failures, power outages, failed porting and software bugs. We share our experience in conducting the ensemble forecasting experiments on SuMegha Cloud using the proposed framework. We also attempt to provide a perspective on the desirable features of a scientific cloud infrastructure, for easier adaptation by the climate modeling community to conduct large scientific experiments.

International Symposium on Grids and Clouds 2016
13-18 March 2016
Academia Sinica, Taipei, Taiwan

[1]Speaker

PoS(ISGC 2016)005

1. Introduction

Weather prediction is an estimation of the future state of the atmosphere by using the current state of the atmosphere and then calculating how this state will evolve in time using a weather prediction model. The current state of the atmosphere can be estimated only with certain accuracy. As the atmosphere is a chaotic system, very small error in the estimation of the initial state can lead to large errors in the forecast. Ensemble forecasting methodology is a popular technique to minimize the effect of uncertainties in the initial conditions [1] [2].

To obtain real time seasonal forecasts using ensemble forecasting methodology, we conduct many number of experiments using slightly different but similar input conditions. It requires enormous reliable computing power and storage which typically cannot be found in commodity servers, thus there is a need for infrastructure like Grid or Cloud. The ensemble forecasting is a set of tasks which can run independently, where each task uses the same application but different parameter and/or input files [3]. Hence, it has inherent parallelism which makes it suitable application that can effectively board on Cloud or Grid Infrastructure.

Seasonal forecast model (SFM) is an Atmospheric General Circulation Model (AGCM) developed by Experimental Climate Prediction Center (ECPC), USA. SFM is a stable, efficient and state-of-the-art model designed for seasonal prediction and climate research. It is an open resource available to the research and academic communities under research license [4].

The work presented in this paper focus on the experience of conducting large ensemble forecast experiments with SFM on the Sumegha Cloud Infrastructure with high availability and reliability. Sumegha is a cloud infrastructure designed and built by Centre for the Development of Advanced Computing (C-DAC). The Sumegha Cloud provides us on-demand access to a shared pool of resources such as virtual HPC clusters, Distributed Mass Storage Resources (Cloud Vault), Scientific Tools and Instruments, with high bandwidth and low latency network [5].

The rest of the paper is organized as follows. In section 2, we briefly discuss the design and portability details of SFM. The details of Sumegha Cloud Infrastructure and the schematic structure of Sumegha in user's point of view are described in section 3. The details of the implementation of low resolution as well as high resolution SFM on Sumegha are presented in section 4. We also illustrate the framework used to do the ensemble experiments on Sumegha in the same section. Furthermore, the benefits, the challenges faced and the requirements of the weather forecasting community from a cloud computing infrastructure are discussed in section 5. In the same section, we present few results obtained from the experiments of SFM. Finally, we present our concluding remarks and future work in section 6.

2. Portable design of Seasonal Forecast Model

In the following subsections, we describe the components of the SFM, the parallelization design used and the portability details.orm and Legendre transform.

2.1 Design of SFM components

Seasonal forecast model consists of three major components: Model Libraries, Model Source Code, and Model configure & run system. Model Libraries contain the necessary APIs, utilities, and climatological or constant fields. Model libraries are machine dependent and need to build again if we change the machine. Model Source code is used to create the model executables which are used for forecasting. For different resolutions of the model or for different modeling options, we need to compile the source code separately. Run scripts contain Bash shell and Perl scripts to run the model, post process and store the model output files in a specified location.

The parallelization strategy used in the implementation of SFM provides us the flexibility to allow the same code to run on sequential, on shared memory parallel as well as on distributed memory parallel machines. This was attained by incorporating a preprocessing stage of the parallel code before the compilation. 2-D domain decomposition method is used to partition the data to perform parallel computations.

Seasonal Forecast Model uses spectral model. To efficiently design a parallel algorithm to implement the spectral model, the data distribution should provide a facility to perform computations with less communication among the processes or if possible, without communication. To avoid communication among proceses, it needs the whole data in one of the three dimensions (X or Y or Z) to reside in one processor to perform computations without communication. For example, Fast Fourier Transform (FFT) requires entire data in the X-direction to reside in one processor, but the data in Y and Z directions can be separated into different processors. This design provides us massive scaling of the application if we have huge computing power and data storage. This method has been widely used in many global spectral models. 2-D decomposition is flexible in the choice of number of processors, almost any number can be chosen, except a prime number. But, the efficiency may be affected for multi-way node machines, where inter-node communication is faster than intra-node communication [6].

2.2 Portability details of SFM

SFM can be ported on several types of machines or clusters. It can run using a single processor as a sequential application. It can also run on a massively parallel processor (MPP) machines using Message Passing Interface (MPI) and OpenMP programming paradigms. The code was designed carefully in such a way that reproducibility of the computations is possible on all types of processor configurations.

3. The SuMegha Cloud

Sumegha is C-DAC's scientific cloud which provides convenient access to reliable, high performance clusters and storage to Indian researcher and scientists without the need to purchase and maintain sophisticated hardware [5]. C-DAC is a pioneer in HPC in India, and its HPC, Grid and Cloud Infrastructures which are linked using National Knowledge Network (NKN) [7]. The high level architecture of Sumegha is shown in Figure 1.

3.1 Services offered by SuMegha

Sumegha mainly provides Infrastructure as a Service (IaaS) and Storage as a Service (StaaS). The following are the few important offerings.

PoS(ISGC 2016)005

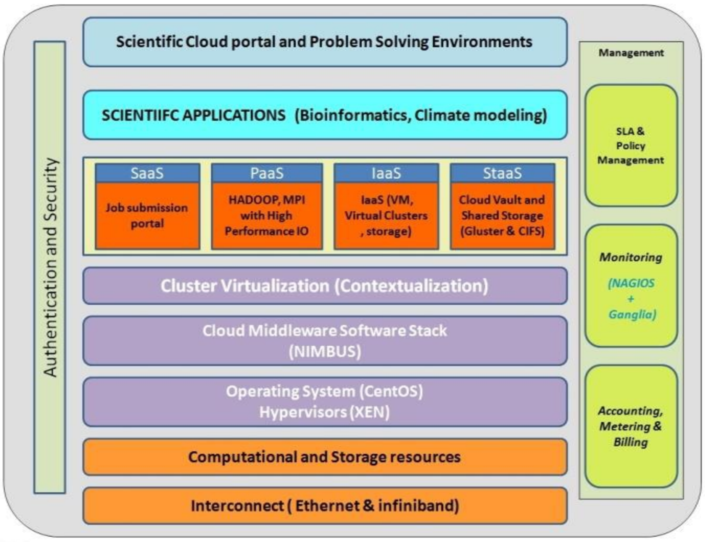

Figure 1. Architecture of SuMegha

(a) Cloud Portal
It offers users a self-service portal that can be accessed through Internet. User can register using simple online registration form and request for resources when he requires. It also provides a Linux client to access the resources using command line interface.

(b) On-demand computong resources
Sumegha offers high performance virtual Clusters on demand with CentOS as the OS. The virtual machines are available in 3 variants: small, medium and large based on the user requirements. It also supports the user to configure and create MPI and HADOOP [8] Clusters on demand.

(c) Storage as a Service (StaaS – Cloud Vault)
SuMegha offers Storage as a Service named as “Cloud Vault”. Users or Organizations can use Cloud Vault to store large amount of data efficiently, safely, and cheaply. It is based on Openstack Swift and Amazon Web Services (S3) - Simple Storage Service. It offers 99% availability to the users with 3-way redundancy using data-replication. It is highly elastic and reliable.

(d) Customized reliable virtual machines
SuMegha allows users to give user requirements and it preconfigures the virtual machines before the virtual cluster is created. It also provides logs and resource history to the user. It allows the user to check the health of the resources, details about the jobs and the usage metrics of the resources.

3.2 SuMegha Cloud – A user's view

In a HPC cluster or in a commodity server, a user is accustomed to local system environments, where all the resources are homogeneous and can be accessed through a unique user account. In a Cloud environment, resources can be distributed and heterogeneous in nature.

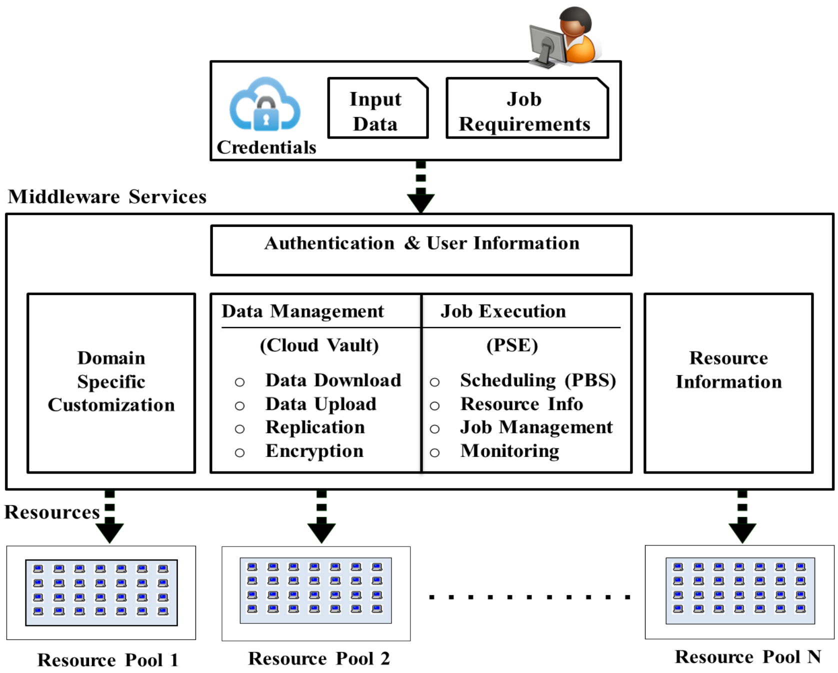

Figure 2. Schematic representation of SuMegha Cloud – User's View

Cloud middleware aggregates all the cloud resources and present them as a single infrastructure from which a user can choose the required resources. The user's view of Sumegha cloud is shown in Figure. 2.

From the user's perspective, the job submission process to the cloud is same as the job submission process of a local cluster or there of a supercomputer: the user fills or prepares the job requirements form and submits the job to the allocated resources using Access Portal or Command Line Interface (CLI). Job scheduling is done using local Torque scheduler.

The storage and access to data are done through a distributed file system - Cloud Vault. The data can be replicated and distributed to different sites and Cloud Vault can select a copy to be used for a particular execution according to, for instance, proximity to the compute resource. The user can transfer/download files to/from Cloud Vault through the portal or command line interface. After the successful completion of the execution of the job, user can post-process and visualize the data using the preinstalled visualization tools such as Grid Analysis and Display System (GraDS) [9].

4. Implementation of SFM on SuMegha

A framework is designed to do the ensemble forecasting experiments using SFM on Sumegha Cloud Infrastructure with high availability and reliability. Initially, the low resolution configuration of Seasonal Forecast Model (T62, approx. 200Km x 200Km) is implemented on Sumegha as a prototype. Here, 'T62' stands for spherical harmonic expansion truncated at wave number 62 and it uses triangular truncation. After the completion of these prototype experiments, high resolution configuration of the model (T320, approx. 40Km x 40Km) is implemented and conducted many experiments to understand the scalability of the application and to investigate the reliability and the availability of the Sumegha Cloud resources.

4.1 Proposed framework

The execution of SFM on Cloud is divided into two parts: Sequential and Parallel. In sequential part, we do the model pre-processing, configuration and creating the necessary data files required by the model to generate forecast. The second part is forecast, which runs in parallel to produce the forecast. The framework to conduct ensemble experiments on Sumegha is shown in Fig. 3. The following is the step wise procedure to conduct ensemble experiments using the framework.

(a) Login using Sumegha Cloud Portal or Command Line Interface using the credentials provided by Sumegha Operations Team.

(b) Create a job requirements file which will have the details about the application, such as the list of ensemble members, number of processing elements required, the amount of the memory required, the amount of storage required, the guestimate of the wall clock time required to complete the job and the list of initial/input conditions.

(c) Provide the list of ensemble members using which the model executes and generates forecast.

(d) Classify the available resources using historical information and train the resource ranking model. SFM was executed with different combinations of the resources available on Sumegha and the required metrics will be captured and stored. The metrics include the total turnaround time, number of bytes in and the number of bytes out in a specified time, memory usage patterns, time required for stage-in and stage-out processes, number of jobs failed etc.

(e) Find out the unreliable or sluggish resources using outlier detection technique, with the help of the metrics obtained in step (d) and the current state of the resources.

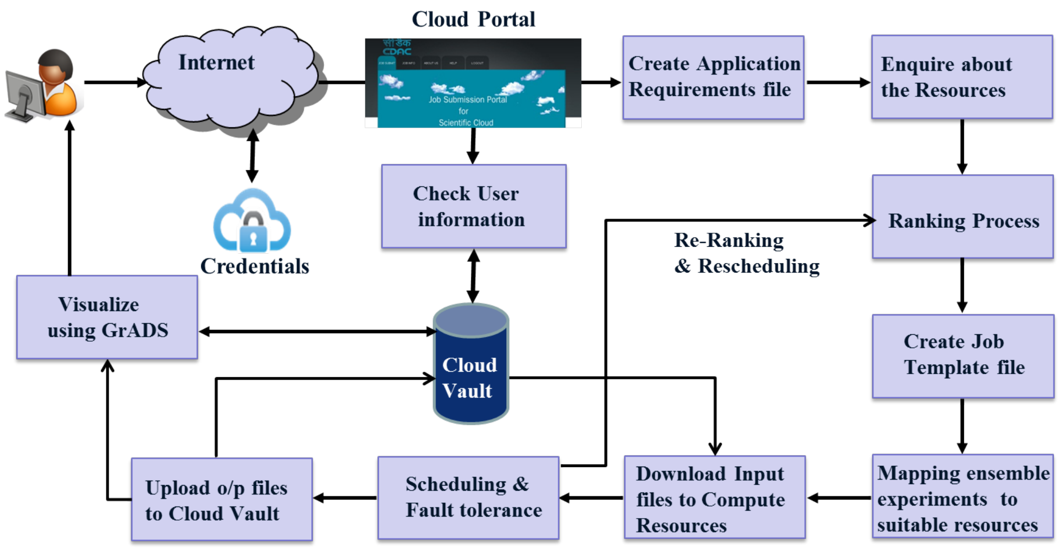

Figure 3. A framework to conduct ensemble experiments with SFM on Sumegha Cloud Infrastructure

(f) Assign ranks to the resources according to the information obtained in steps (d) and (e).
(g) Create a job template file which will be used by the cloud resource manager.
(h) Upload the necessary input files corresponding to each ensemble member and the SFM executables to Cloud Vault. These files will be used by the compute resources while executing the model.
(i) Schedule the jobs according to the ranks given to the resources in step (f). The first ensemble member experiment will go to the resource having rank 1.
(j) Monitor the jobs using SuMegha Job Portal or using Command Line Interface.
(k) If a job fails, the job will be automatically rescheduled with the same ensemble member as the input to the next available best resource. If it fails again then the job aborts. User/SuMegha Operations team's intervention is required to trace and fix the problem. Then user should resubmit the job. And, the ranking system needs to be executed again.
(l) Upload the output files to distributed storage system (Cloud Vault).
(m) Visualize the output data using GrADS which is integrated in the job submission portal. The resulting output plots can be downloaded by the user.

5. Discussions and Results

Initially, SFM was configured to obtain 3-days forecast with 30 minutes time step, to run on 8 processors of SuMegha virtual clusters. Several runs were made on each of the resources available with SuMegha and the results were validated. After validating, the model was configured to do the forecast of Indian summer monsoon season that spans from June to September, using the same number of processors. Several experiments were conducted on various combinations of resources and performance variations are observed. These observations were communicated to SuMegha operational team to fine-tune the configuration of the resources to improve the performance in terms of availability and reliability.

Forecast Length	Total Turnaround Time		
	Physical Cluster	SuMegha Virtual Cluster	Performance Loss
3 - Days	2 min. 42 sec	4 min. 35 sec.	69.70%
165 - Days	60 min. 16 sec.	80 min. 40 sec.	34.00%

T*able 1: Performance of SFM T-62 using 62 CPU cores*

There is a significant fall-off in the performance from physical cluster to virtual cluster. The performance metrics are tabulated in Table 1. The physical cluster and the SuMegha virtual cluster have similar hardware and software resources, with small variations. In case of 3-Day forecast experiment, the total turnaround time of the model run on virtual cluster (4m 35s), which is almost double the total turnaround time of the physical cluster (2min 24s). If we increase the problem size to 165 days forecast, then the total turnaround time of the model run on SuMegha virtual Cluster (80m 46s) is almost 25% more than that of the independent cluster (60m 16s). The reason for the degradation of the performance in both the cases is investigated and was found out that this is due to the extra time taken by the stage-in, stage-out processes on the virtual resources and also due to the overheads incurred by ranking mechanism. We also observed that the variations in the performance of different combinations of virtual resources with similar type of resources. This happened due to several factors including the small differences in the resources such as CPU speed, wall time spent in queue, different versions of MPI libraries, the differences in bandwidth and restart due to errors/problems during the execution. These observations are very useful to the cloud service providers to fine-tune the resources to improve the performance.

Later, high resolution SFM was configured for a 1-day forecast with 2 minutes time step, to run using 64 processors (8 nodes x 8 Processors). Many runs were made on five virtual clusters of SuMegha to validate the porting of the model. We also configured and compiled the model to run it for seasonal forecast using 64 processors. These configurations are used to understand the scalability and the health metrics of the cloud resources. Later, the proposed framework was used to do the ensemble experiments across the virtual clusters of SuMegha.

The ensemble forecasting consists of numerous seasonal forecast runs. A seasonal run produce the forecast for a period from June to September. This period is called as Indian summer monsoon season (ISMR). A seasonal run using a single ensemble member needs around 30 GB of disk space and around 85 Hours of total turnaround team. Total CPU time of all these runs was more than 8500 hours i.e. around 1 year. Using ten virtual clusters of cloud with 64 processors each, model ensemble runs can be completed within 15 days. This shows that weather and climate forecasting applications can harness the power of cloud computing even though there is a degradation in computational performance.

The experience gained from conducting ensemble experiments with simple ranking and restart mechanism gave us an opportunity to the user to conduct the experiments without worrying about the various hardware and software failures. It allowed us to conduct large experiments with less than 8% of failure rate, whereas, without using the proposed framework, the failure rate is around 24%. It also gave a perspective on the desired features of a cloud middleware for easier uptake to cloud computing by the weather and climate modeling community. The main challenges are – large number of parallel computations, long runs takinh several hours to several days and huge volumes of output data. For example, to compute the climatology for the period of 1984-2015, i.e. 32 years, using six ensemble members using SFM-T320, one would require 5.5 TB of disk space and around 660 days of wall clock time if the model is run on a single virtual cluster using 64 processors (8 nodes, each node with 2 x Quad core Xeon @ 3.16 GHz and 64 GB memory). To address these challenges, Cloud middleware should have the following features.

- It should provide a mechanism to address the issues such as non-uniform memory sizes that are available on the virtual clusters of the cloud.
- It should be able to identify the failed jobs as early as possible.
- It should hide the virtualization layer completely from the application.
- It should seamlessly transfer the huge output data files to the user from cloud during the experiment which will avoid the accumulation of huge data on the compute clusters.
- It should allow automatic migration of failed jobs to other reliable resources.
- It should provide dynamic scaling of resources without user's intervention.

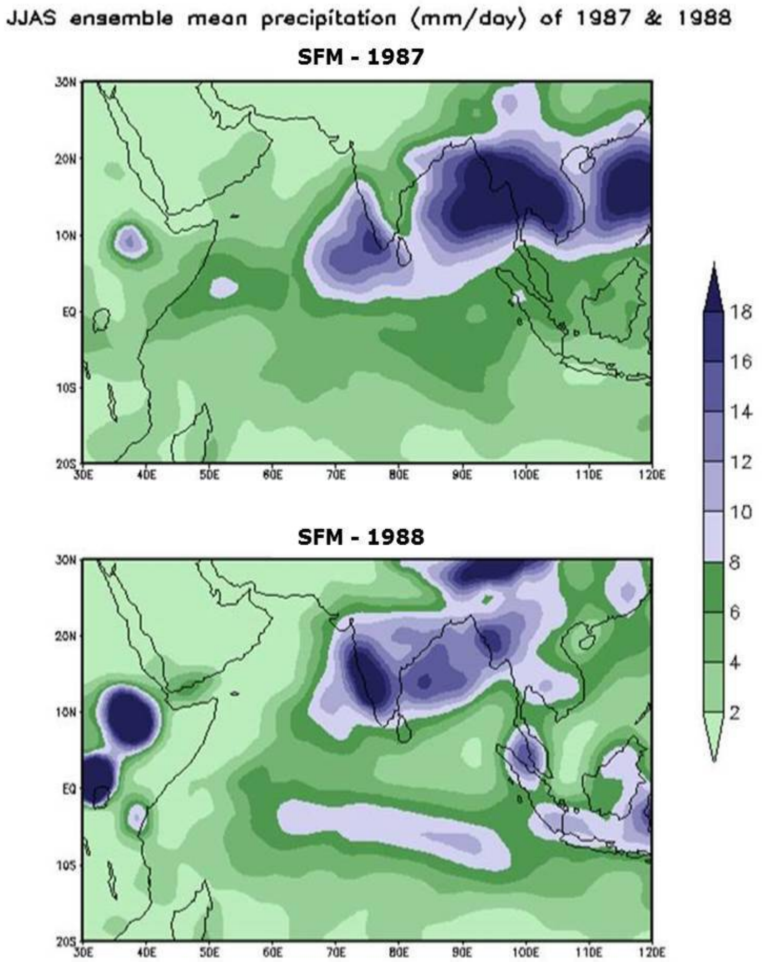

Figure 4. The ensemble mean rainfall of Indian Summer Monssoon of 1987 and 1988

We also present few of the results obtained from SFM-T320. In Figure 4, the top panel depicts the ensemble mean ISMR, i.e. mean rainfall of June to September of the year 1987 and the bottom panel shows the ensemble mean ISMR for the year 1988. We are interested in assessing the skill of SFM by checking whether SFM is capable of simulating drought and excess rainfall years. SFM is capable of predicting the excess rainfall occurred in 1988, and the drought occurred in 1987.

6. Conclusions and Future work

In this paper, we presented our experience in conducting the time critical large ensemble forecast experiments using an atmospheric general circulation model SFM, on SuMegha scientific cloud infrastructure using a framework with ranking and restart mechanism. We have observed that ensemble forecasting is a suitable application from climate modeling domain which can harness the power of cloud computing paradigm. The proposed framework gave us a foundation to conduct large experiments on cloud with improved reliability, on which we can build a reliable cloud environment for climate applications which are time sensitive and critical. Our experiences with building and deploying the climate model (SFM) on Cloud emphasize the need for intelligent middleware technologies which need to be implemented in Cloud infrastructure. With the contribution of this paper, we will further enhance our framework to deal with the complexity and the instability of the SuMegha cloud, which can help the climate community to port their complex applications with more comfort.

References

[1] M. Leutbecher et al, "Ensemble Forecasting", Journal of Computational Physics, Vol. 227, pp. 3515-3539, 2008.

[2] John M. Lewis, "Roots of Ensemble Forecasting", Monthly Weather Review, Vol. 133, pp 1865-1885, 2005.

[3] V. Fernadez etal, "Benefits and requirements of grid computing for climate applications. An example with the community atmospheric model", Journal of Environmental and Modelling software, Vol. 269, pp 1057-1069, 2011.

[4] Kanamitsu et al.,"NCEP dynamical seasonal forecast system", BAMS, 83, 1019-1037, 2000.

[5] Sumegha – CDAC Scientifi Cloud, http://www.sumegha.in/cloud-services/. (Accessed on 22nd September, 2015).

[6] J. G. Sela, "Weather forecasting on parallel architectures", Journal of parallel computing, Vol. 21, Issue 10, pp 1639-1654, ISSN 0167-8191, 10.1016/0167-8191(95)00039-1, October 1995.

[7] Natioanl Knowledge Network – Connecting Knowledge Institutions, http://www.nkn.in/. (Accessed on 22nd September, 2015).

[8] Raj, A.,Kaur, et al, "Enhancement of Hadoop Clusters with Virtualization Using the Capacity Scheduler", Third International Conference on Services in Emerging Markets(ICSEM), December 2012.

[9] Grid Analysis and Display System – An interactive visualization tool for Weather and Climate,http://iges.org/grads (Accessed on 22nd September, 2015).