Advances in Computer Science and its Applications (ACSA)
Vol. 1, No. 4, 2012, ISSN 2166-2924

www.worldsciencepublisher.org

Multi Level Adaption for Efficient Workflow Management in Cloud Computing

M. Sudha[1], M. Monica[2]

[1] Assistant Professor Senior, School of Information Technology & Engineering,
[2] Assistant Professor, School of Computing Sciences &Engineering,
VIT University, India

Email: msudha@vit.ac.in

Abstract - Scientific computations from various research and business endeavors are composed as workflows that access shared data, shared pool of resources on demand. Predicting the dynamic properties of the resources and coordinated control of resources and workflows is critical especially for deadline sensitive workflows. Existing approaches for executing workflows are mainly focused on homogeneous environments whereas in the cloud, workflow has to manage new issues such as scheduling over different cloud environments, initialization of virtualized instances and impact of data transfer. The investigation of cloud workflow systems, is a timely issue and worthwhile for increasing performance efforts. This research work considers the problem of design and optimization of cloud workflows.

Keywords - Cloud Computing; virtualization workflow; multi-tenant; middleware

1. Introduction

Most recently scientific and business applications are leveraging the power of distributed computing infrastructure and storage resources through utility, grid and cloud computing as an integral part of mainstream business models. These trends are changing the software services and the interaction of higher-level software with distributed systems. The advent of the internet has also resulted in the emergence of sophisticated end-user tools such as web interfaces and portals that enable the end-user to access distributed information and resources. Workflow tools have emerged at the applications later that allow users and businesses to compose work units as a sequence of automated or semi-automated operations.

Workflow systems are designed to support the process automation and task sharing of large scale business and scientific application. Workflow refers to the activity of defining the sequence of tasks required to manage a business or scientific process, workflow is a template for such an orchestration. In recent year most of the workflow systems are deployed through distributed computing platforms and commercially hosted services such as clouds. Cloud Computing environments facilitate ease of resource sharing in terms of scalable IT infrastructure, application development platforms and value-added business application services. The operation models may include pay-as go utility models, free infrastructure services with value added platform services, fee-based infrastructure services with beneficial application services, or free services for providers allowing sharing of revenues generated from consumers (Liang-Jie Zhang et.al. 2009). It has become extremely important for scientific and business communities to be efficient enough to develop share and use the distributed applications in a highly dynamic fashion. Since most of these resources are self-contained and organized in a heterogeneous way, resource scalability, making a system itself to expand and contract with its resource pool to accommodate complex and simpler workloads, is very tedious and inefficient. On some situation it requires to address „resource insufficiency. During peak periods; on the other hand, it cannot provide services to others during off-peak periods.

In current computing paradigms, workflow systems have to maintain their own high performance Computing infrastructures rather than to employ them as services from a third party according to their real needs. Meanwhile, most of the resources are idled except for bursting resource requirements of large scale workflow applications at peak periods. In fact, many workflow systems deal with a large number of conventional less demanding workflow applications for large proportion of time.

Therefore, resource scalability is becoming a critical problem for current workflow systems. However, such an issue has not been well addressed by current computing paradigms such as cluster, grid and cloud computing. Workflow planning techniques so far have focused on performance based resource selection and mapping in conjunction with run-time systems handling failures and variations. In addition, as these workflows run in resource environments such as utility computing and cloud computing, it is vital to account for the diversity and variability of the resources in planning techniques. When resources are procured across a group of users or workloads, workflow planning needs to be preceded by a separate resource acquisition phase and followed up with adaptation as resource properties change during workflow execution. Therefore a well coordinated, dynamic and adaptive approach to workflow planning that works with user requirements and variable resource characteristics while being shielded from specific resource policy or systems is always wanted in this situation. In this context, a workflow orchestration handles planning and execution management that includes resource acquisition, workflow scheduling and real-time adaptation is always required.

This research intent is to investigate the existing workflows to identify the issues affecting the efficiency of workflow management for cloud environment and to propose a framework with multi level adaptability in workflows for scientific application. Section 2 and 3 describes generic system model and related works on cloud. Section 4 states the current issues in this field. Current gaps and design is clearly defined in section 5 section 6 describes the design criteria and the proposed system design in section 7 followed by our current design conclusion in section 7.

2. Generic System Model

Cloud computing system model can be divided it into two different software components: the front end and the back end and it is connected with each other through a network, usually the Internet. The front end is usually a client or the user end. The back end is the "cloud" section of the system which can be data center too. [16] The front end includes the client's computer (or computer network) and the application required to access the cloud computing system. Not all cloud computing systems have the same user interface. Services like Web-based e-mail programs leverage existing Web browsers like Internet Explorer or Firefox. Other systems have unique applications that provide network access to clients. Back end of the system are the various computers, servers and data storage systems that create the "cloud" of computing services in hypothesis, a cloud computing system could include almost any computer software as you can imagine, from data processing to video games.

Typically, each application will have its own associated servers. A centralized server is expected to administer the system, monitoring traffic and client demands to ensure everything runs smoothly. It follows a set of rules and agreed upon practices referred as protocols and uses a special kind of software called middleware. Middleware is a sophisticated software program that allows networked computers to communicate with each other. This model does not seem to address end-to-end management. Ultimately, the cloud service infrastructure must provide end to-end service assurance to meet both service creation and service delivery platform user requirements. The service creators must be able to develop services rapidly using reusable and collaborating service components available globally.

The infrastructure must also accommodate billions of users globally who will contribute to wildly fluctuating workloads. Cloud computing represents an ever demanding profitable IT trend. earlier to cloud computing, obtaining such resources the initial IT investment in purchasing the computers themselves and the significant resources devoted to maintaining the infrastructure, upgrading the software was an expensive and unlikely proposition for various organizations and simply impossible for individuals in most of the cases. At the present, cloud computing has the potential to benefit both providers and users. Cloud providers gain additional sources of profits and are able to improve their large data centers and the expertise of large scale data management with less human effort , on the whole cost is reduced through consolidation, while capital investment in physical infrastructure is reduced across many customers.

Independent cloud users can store, access, and share information in previously unimaginable ways. Managerial cloud users no longer have to worry about purchasing, configuring, administering, and maintaining their computing infrastructure, which allows them to focus on their core competencies.

A scientific workflow is a description of a process, usually in terms of scientific computations and their dependencies .It can be visualized as a directed graph, whose nodes (also called actors) represent workflow steps or tasks, and whose edges represent dataflow and/or control-flow. A workflow support the integration of components for data acquisition, transformation, and analysis to build complex data-analysis frameworks from existing building blocks, including algorithms that are available as locally installed software packages or globally accessible web services. Workflows describe the relationship of the individual computational components and their input and output data in a declarative way Workflow management is very essential because much of the benefits of cloud computing approach is the effective process swiftness and simplicity with which IT resources can be created and put into effective usage.

2.1 Generic Workflow Modeling - DAG

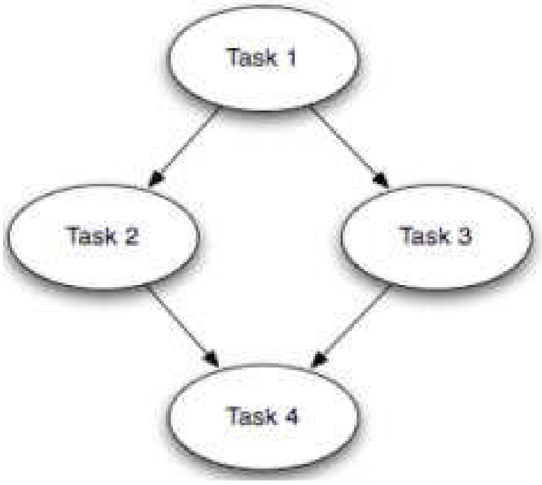

Fig. 1. DAG representation of workflows

Workflows can be expressed as directed acyclic graphs (DAGs).Tasks in a DAG is arranged in a hierarchy according to their dependencies. Each task has an outgoing edge connecting it to all other tasks that are dependent on it. Edges typically represent data dependencies such as input and output files, but they may also be used to impose a specific ordering on task execution or to represent non-data dependencies, such as dependencies on shared resource.

2.2 Characteristics Classifying Workflow Types

- Dataset structure and size (decide the resource requirements)
- Pattern – Control flow design (Sequence, parallel, split)
- Computation – Time involved, Data – different types (including input data, backend databases, intermediate data products, output data products).

The combination of the structural and pattern characteristics, the computational and data sizes helps in understanding the workflow requirements when making planning and adaptation decisions.

3. Related Works

Bhaskar Prasad et.al. (2010) reported that the scientific workflow in the context of multi-tenant cloud environment provides control flow, data flow, and new requirements of system development and concluded two methods, semantic based workflow and policy-based workflow to implement scientific workflow.

Daniel de Oliveira (2010) have stated that the large scale scientific experiments modeled as scientific workflows produce a huge amount of data and require simultaneous workflow s to reduce process execution time.

Cloud computing implies a service oriented architecture, reduced information technology complexity for the end user, great elasticity, reduced total cost of investments, on demand service and many other things(Mladen A. Vouk , 2009)

Guy Bunker et al.(2009) described that managed cloud services that include OS level support can result in improved reliability and security.

Christina Hoffa et al. (2008) Stated that a workflow with short job runtimes, the virtual environment can provide good compute time performance and clustering small jobs together can greatly reduce the scheduling overheads.

Blythe et. al. (2005) identifies and evaluates two resource allocation strategies for workflows - task-based and workflow-based. The task-based algorithm greedily allocates tasks to resources. Workflow-based algorithms find an optimal allocation for the entire workflow and perform better for data-intensive applications.

The demand for cloud-based consumer and business applications currently necessitate the next generation computing powers of datacenters that must be massively scalable, efficient, agile, reliable and secure. The basic requirements for a cloud computing environment are to scale cloud services reliably to large number of service developers and billions of end users with efficient workflow tools for the next generation cloud computing infrastructure to follow an evolution similar to the one that a to the formation of scalable internet works.

4. Current Issues and Analysis

Existing Workflow planning techniques focused on resource selection process and mapping in combination with run-time systems handling failures and variations. Many currently existing workflow models are designed for set of specified resources and the workflow techniques only partially interoperate with the available shared pool of computing resources. Current workflow scheduling and planning mechanisms are confirmed to be incompetent for the frequent dynamic changes and scalable requirements of multi user cloud applications.

Current cloud evolution is limited to the following three areas:

- The Virtualization of servers, load balancers, and some server IP address management services.
- The replacement of SAN infrastructure with large commodity server farms that support virtual applications using Direct Attached Storage (DAS) or File Systems (distributed or otherwise).
- Efficient management of workflows in a cloud environment to allow fast scaling up and scaling down.

- Storing scalability/ compressibility options for every node in the workflow
- Input events and output events of every node in workflow

Transparency

a) An activity is a discrete step in a business process (workflow).

b) Activities range from calling a remote service to perform a task, e.g. calculating taxes, performing currency conversions, looking up inventory, to custom-defined services.

c) Actual Implementation" of services obscured an another version of virtualization

d) Transparent load-balancing and application delivery

e) Solution to be automated and integrated in workflow process.

Scalability

a) Scale up and build "mega data centers"

b) Need configuration or re-architecting

c) Potential of interrupting services is huge and the ability to transparently scale the service infrastructure and the solution and On-demand, real time scaling.

f) Control node provides dynamic application scalability

g) Integration with virtualization solution or orchestration with workflow process to manage provisioning.

Intelligent Monitoring

a) Control node – intelligent monitoring capabilities

b) Server overwhelming or application performance affected by network conditions – behavior outside accepted norms

c) More than knowing when a service in trouble what action should be taken

Capacity management

a) From buckets to rivers

b) Constrained set of resources – predict peak usage and have in-house data center to manage them

c) Not upper limit of computing power but speed at which new services can be provisioned and put into production

d) Initiate new system, transfer data, connect existing system, test combined system, manage complete life cycle

5. Current Status and Gaps

Most of the existing workflows are designed for set of specified resources and the workflow techniques only partially interoperate with the available shared pool of computing resources. It is most wanted in this state of interaction to have an optimized workflow systems that could provide space for creating workflow portal where users can interact using a simple GUI based editor tools for choosing their own menus or options for application submission interface, application monitoring interface etc.

6. Proposed Design Criteria

In a multi-tenant cloud environment where multiple workflows are submitted by each user, priorities, budget and other constraints need to be accounted for across the workflows. When resources are acquired across a group of users, workflow planning can be performed as a Separate phase which focus on the required resource acquisition and followed up with resource adaptation phase.

6.1 Multi Level Scheduling

In multi level scheduling the workflows involves two distinct activities such as management of resources and the scheduling of actual application jobs, these two functions are divided between the user and the provider. Providers can retain their influence over resources and users are given control over scheduling. This division is accomplished by creating personal temporary collection of computational resources provisioned from a cloud site and managed by the user referred as personal stub. After a personal stub is created it can be used to schedule and execute application jobs using policies specified by the user based on the needs of the application.

6.2 Multi Level Adaptations

To support a dynamic environment of Cloud Computing, we need Multi-level Adaptation that supports local changes while balancing the global state of the system. Support for synchronized planning of resources and services with workflow characteristics to meet the needs of the user. Ability to react to some of the user-initiated changes in run time and to adjust to available personal stub and shared pool of underlying configurable resources and should be able to adapt the availability variations.

7. Proposed System Design

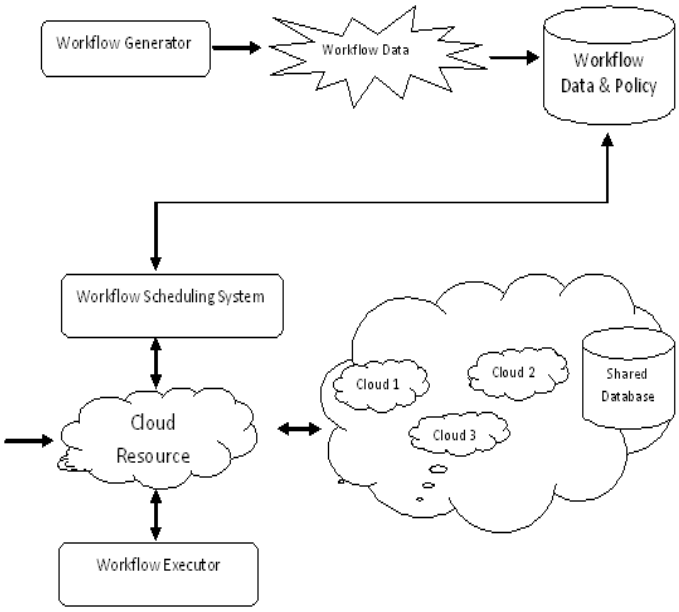

Fig. -2 Workflow Systems for HPC.

7.1 Workflow Generator

In this proposed architecture consists an orchestration system and execution system .Orchestration function is to plan for the successful execution of the application, It will determine a set of tasks to be performed in a cohesive manner. This is done as two level processes, level one will get the resource requirement and does the actual resource selection using some programming models and resource mapping is done in the second level. Workflow generator produces the workflow data which include the resources selected and the mapped resource; this is determined by an iterative procedure. Then the workflow data will be checked to comply with the underlying systems policy.

7.2 Workflow Scheduler

Workflow scheduling system will generate the actual execution schedule based on heuristic techniques the major difference between the workflow generator and the scheduler is the generator mainly performs the resources acquisition in user space and the scheduler will receive the workflow data with detailed resources slots and will produce a systematic updated execution schedule (directed acyclic graphs) a DAG scheduling approach can be used.

7.2.1 Dynamic Adaptive Scheduling

This Dynamic Adaptive scheduling [17] adapts a new model of scheduling which is based on multiple resource data retrieval approach, this dynamic scheduling heuristic assign interdependent tasks to compute resources based on both multi-source parallel data retrieval time and task-computation time using tree based resource retrieval procedure. Hence, with a combination of data retrieval and task-resource mapping technique the proposed model can achieve efficient schedules that are better than existing heuristic based techniques, for scheduling application workflows. A scheduling mechanism should be able to select the data sources and transfer data to a compute host with an optimal transfer time. Similarly, the selection of the compute host, in relation to the selected set of data hosts, should be such that the execution time is minimal.

A list-based scheduling heuristic is followed based on a ready-list, where tasks are added as they become available for scheduling. In dependent-task scheduling, child tasks become ‘ready’ only when their parents have successfully completed execution. The ready-list is filled by the scheduling loop, starting from the root task, as tasks are scheduled and get completed.

Adaptive Dynamic scheduling model provides better solution by integrating the availability of multiple resources and data retrieval mechanism on tree structure pool of collected similar resources in the cloud infrastructure enabling parallel acquisition of resources for workflow executions, the implementation of the scheduling algorithm on top of these scheduling infrastructures enables successful workflow completion in cloud with better performance and minimal time and supports efficient data management in cloud computing

7.3 Workflow Executor

Workflow executor is the execution system that executes the set of operations involved in the tasks as produced by the scheduler however in the real time scenario it is expected to experience significant effects on reliability and efficiency. The inputs for the workflow mostly depend on the type of application for example if a scientific weather forecasting application is involved then the input data will be a preprocessed flow of data from the sensors, therefore the input data will be based on the nature of the application executed. .Executor will have to maintain a work queue it is proposed to have a simple priority queue data structure to manage the work priorities. The execution is DAG dependent. Thus the workflow can be managed in efficient manner with the expected perform ability.

8. Conclusion

Recent advances in computing, fast coming out of virtualization technologies, web-services and multi-core architectures have accelerated advances in utility and cloud computing business models. These changes in turn necessitate the need to closely examine the software systems and the workflow tools that runs atop these systems and services offered by several distributed data centers to provide predictable optimized services to scientific and business applications.

References

[1] Bhaskar Prasad Rimal, Mohamed A.E I Refaey "A Framework for Scientific Workflow Management Systems for Multi-Tenant Cloud Orchestration Environment" IEEE -2010 workshops on Enabling Technologies: Infrastructure for collaborative Enterprises.

[2] J. Blythe, S. Jain, E. Deelman, Y. Gil, K. Vahi, A. Mandal, and K. Kennedy. Task Scheduling Strategies for Workflow based Applications in Grids. In CCGRID, 759–767, 2005.

[3] C. Berkley, S. Bowers, M. B. Jones, B. Ludäscher, M. Schildhauer, J. Tao, "Incorporating Semantics in Scientific Workflow Authoring", 17th International Conference on Scientific and Statistical Database Management, 27-29 June 2005, Santa Barbara, California.

[4] Christina Hoffa, Gaurang Mehta, Timothy Freeman, Ewa Deelman, Kate Keahey, Bruce Berriman, "On the Use of Cloud Computing for Scientific Workflows", Fourth IEEE International Conference on eScience, 2008.

[5] Daniel de Oliveira, Eduardo Ogasawara, Fernanda Baiao, "SciCumulus: A Lightweight Cloud Middleware to Explore Many Task Computing Paradigm in Scientific Workflows" 2010 IEEE 3rd International Conference on Cloud Computing.

[6] David Abramson, Blair Bethwaite, Colin Enticott, Slavisa Garic, Tom Peachey , Anushka Michailova, Saleh Amirriazi and Ramya Chitters " Robust Workflows for Science and Engineering" ACM 2009.

[7] Ewa Deelman, Dennis Gannon, Mathew Shields and Ian Taylor "Workflows and e-Science: An Overview of workflow systems features and capabilities", Future generation Computer Systems 25(2009) 528-540.

[8] Guy Bunker, Farnam Jahanian, Aad van Moorsel, Joseph Weinman," Dependability in the cloud: Challenges and opportunities", IEEE 2009.

[9] Lavanya Ramakrishnan and Beth plale "A multi-Dimensional classification model for Scientific Workflow Characteristics" WANDS 2010, ACM 978-1-4503.

[10] Lijun Mei, W.K.Chan, T.H.Tse, "A Tale of Clouds: Paradigm comparisons and some thoughts on research issues", IEEE Asia-Pacific Services Computing Conference, 2008.

[11] Mladen A. Volk "Cloud computing-Issues, Research and Implementations", Journal of computing and Information Technology, CIT 16 2008,4, 235-246.

[12] Rajkumar Buyya1, Rajiv Ranjan2 and Rodrigo N. Calheiros "Modeling and simulation of scalable cloud computing Environments and the CloudSim Toolkit: challenges and opportunities" IEEE, 2009

[13] S. Ouyang, "Implementation of Policy Based Management in Workflow Management", Lecture Notes in Computer Science, pp 675-666, Springer Berlin/Heidelberg, 2007.

[14] The Force.com Multitenant Architecture, derstanding the Design of Salesforce.com.s Intermet Application Development Platform. Technical White Paper, 2008

[15] M.sudha and M.Monica "A Simplified Network manager for Grid and Presenting the Grid as a Computation Providing Cloud", International Journal of Advanced Research in Computer Science (IJARCS), Volume 01, Number 03, October 2010, pp 173-176.

[16] M.sudha and M.Monica "Investigation on Efficient Management of Workflows in Cloud Computing Environment", International Journal of Computer Science and Engineering (IJCSE), Volume 02, Number 05, August 2010, pp 1841-1845.

[17] M.Sudha, M.Monica: "Dynamic Adaptive Workflow Scheduling Mechanism for Instance Intensive Cloud Applications", Journal of Expert Systems (JES), Vol. 01, No. 01, pp 31-36, 2012.