WORK FLOW SCHEDULING BY OPTIMIZE AND AI APPROACHES IN CLOUD: REVIEW

Dr. Shamsudeen. E

Assistant Professor of Computer Applications, EMEA College of Arts and Science, Kondotty, Kerala, India

ABSTRACT - Applications that use workflow include carrying out a series of actions in a sequential order, which allows the data to be analysed in a way that is both well-structured and dispersed. Because of the interdependence of these jobs, there is a significant amount of data transfer that occurs throughout the execution of the workflow between the tasks that come before and after it. Processing of workflow applications that include terabytes or petabytes of data is required in many branches of scientific research, such as biological engineering, ocean sciences, earthquake science, and many more potentially fruitful fields of study. The processing and analysis of such data requires more complicated computational resources, which, if performed in a standard computer environment, would be excessively convoluted. Implementing workflow applications in the cloud might be the solution to the problem that was discussed above because of the scalability aspect of cloud computing environments, which enables infinite resources for execution. Cloud computing, sometimes referred simply as "the cloud," is a well-liked Distributed System paradigm that provides customers with on-demand, utility-based information technology services on the basis of a pay-per-use payment model. As a result, it makes it possible to execute workflow applications at a minimal cost since it eliminates the need of personally owning any infrastructure. To be more precise, Infrastructure as a Service (IaaS) offers access to heterogeneous limitless resource pools, which makes it possible to install process applications in a cloud environment that is both scalable and adaptable.

Keywords: loud, scheduling, workflow, optimization

I. INTRODUCTION

The advent of microminiaturization of technologies with the ubiquitous networking made the computing resources more influential and cheaper than ever before. This enabled a new computing model called Cloud Computing. Cloud computing is a widely preferred Information Technology that offers resources dynamically in a subscription based service. It consist of a pool of virtualized resources readily available, which can be reconfigured according to the user requirements in terms of scalability and load balance, hence permit opportunities for optimal resource utilization. Workflows are used to represent the scientific and business applications which specify the overall structure and behavior of applications in a platform-independent way which involves high computation with complex large scale data analysis. As a result scientific workflows progressively adopt cloud computing for computation. Figure 1.1 represents the Cloud Workflow Execution Environment

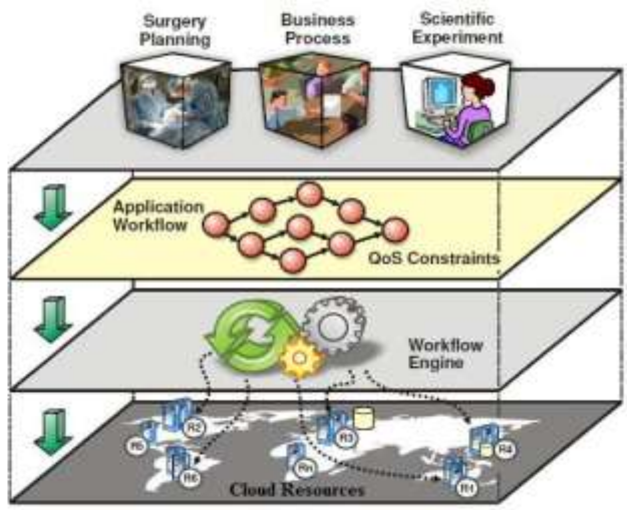


Figure 1.1 High level view of a cloud workflow execution environment

Workflow is generally represented as DAG (Directed Acyclic Graph) consists of nodes and edges, which represents control dependency between tasks. Workflow computation involves several processes such as transferring data, running computations, result analyses and management of output result Systems. The advancement of Workflow Management System made the workflow execution simple and efficient by automating and masking the orchestration of the entire workflow process. In such applications scheduling plays a vital role, as it maps the workflow tasks on to the available resources by preserving data dependencies. For the past several decades there are numerous scheduling algorithms have been widely studied and implemented. A primary issue of scheduling is how the application task needs to be mapped for execution in a resource so as to satisfy the user defined QoS (Quality of Service) objectives. The success rate lies in the fulfillment of QoS requirements, which in turn depends on the effective use of the available resources.

Cloud computing provides more control over the different types and number of the resources employed. This feature along with the abundance of resources enables the need of resource provisioning strategy that works with the scheduling algorithm; a process that determine the number and types of the resources to use and when to acquire and release them. Another challenging issue is that the scheduler should tradeoff between the performance and cost to avoid paying unnecessary costs. The algorithm should be aware of the dynamic nature of cloud platform.

The thesis focuses on the problem of efficient scheduling of large-scale workflow applications in cloud environment. It investigates scheduling and resource provisioning strategy that addresses the challenges involved in resource provisioning. This is attained by a detailed taxonomy and survey of state-of-the art scheduling algorithms. Additionally, a set of scheduling algorithm is proposed to schedule the workflow applications with a notion to reduce the execution time and cost, minimizing the energy consumption and to maximize the resource utilization of running workflow applications in the cloud environment.

II. BACKGROUND

This section presents the fundamental concepts related to the addressed research problem in the thesis.

Cloud Computing Architecture and Types

Cloud computing provides a dynamic and scalable computing resources over the Internet on a pay-per-use basis. NIST defined the cloud computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction". Cloud allows its user to access number of resources using any type of devices with internet access. Cloud provides three types of service models to its users such as Infrastructure as a service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). The cloud service models are represented in a Figure 1.2 Cloud Computing Service Offerings.

(Source: https://www.bodhost.com/blog/the-concept-of-cloud-computing-design-principles-and-paradigms/)

Figure 1.2 Cloud Computing Service Offerings

Software as a Service (SaaS) delivers software and applications through internet. Users can make use of the services via web or APIs and the users are free from the software and hardware management. Platform as a service (PaaS), offers access to the cloud-based environment in which developers can create, deploy and deliver their own applications. Infrastructure as a Service (IaaS), provides the fundamental computing resources such as storage, networking and servers. This could be possible by leasing Virtual Machines (VMs) with a preferred CPU, memory, storage, and bandwidth capacity. Different types of resources are available to suite various application requirements. Various IaaS cloud providers available to serve its user in a better way. IaaS providers manage and maintain the entire environment; only it outsources the required Infrastructure services to the users. The thesis preferred IaaS cloud for resource provisioning.

Types of cloud computing

Public clouds are owned and maintained by the third party service providers. They offer seamless cloud services to the general public. As infrastructure costs are spread across all the users, each customer is benefit from the economy of scale allows operating on a low cost with pay-as-you-go model. All the users on public cloud shares the same resources pool with limited security protection, configuration and SLA specificity, which makes this model not suitable for the services with sensitive data. Private cloud is owned exclusively by the individual organization which provides scalability, flexibility, automation and monitoring. It also addresses data security and control, which is often missing in a public cloud environment. The two variations of private cloud are: On-Premise Private Cloud and Externally-Hosted Private Cloud.

On-Premise Private Cloud is also known as the Internal Cloud, that is hosted inside the organization with an own data center. It provides protection and standardization. It is best suitable for the application that requires absolute control and configurability of the security and infrastructure.

Externally-Hosted Private Cloud is hosted by an external cloud provider that facilitates elite cloud environment with full privacy.

Hybrid cloud combines the benefit of both public and private clouds. An organization can leverage third-party cloud providers in full or partial manner, which increases the flexibility of computing. It has the advantage of providing on-demand resources and externally-provisioned scalability. Augmenting a private cloud with the public cloud resources can be used to manage any unexpected surge in workload. Added advantage is that the individual applications or portions of applications can be migrated to the Public Cloud during peak hours to balance the load.

III. WORKFLOW MANAGEMENT SYSTEM

Workflows

Workflow is a concept that is originated in business world to automate the business logic and tools. Later, the scientific domain acquired the concept from business world to automate the scientific process. Workflow representation of a complex scientific process comprises of large number of dependent tasks requiring higher computation resources for the execution. Scientific workflows are designed to support complex scientific processes. They are used to prove scientific hypothesis and conducting series of experiments by simulating, managing, analyzing and visualizing scientific data.

Workflow applications are modeled as Directed Acyclic Graph (DAG), where nodes corresponds to tasks and edges indicates dependency between the tasks. A significant property of a workflow is that, it manages the flow of data. A Workflow application ranges from a simple serial task to very large complex parallel tasks bounded by large number of small and serial tasks used for pre and post-processing. In general, workflow consists of an automated set of procedures, where file and data are passed between the tasks according to the defined set of policies to achieve the overall objective. Figure depicts the sample workflow with nine tasks. In the example task T1 produces four intermediate output files, which becomes the input for the tasks T6 and T7. Tasks T2, T3, T4 and T5 cannot start their execution till T1 finishes its execution and produces output data.

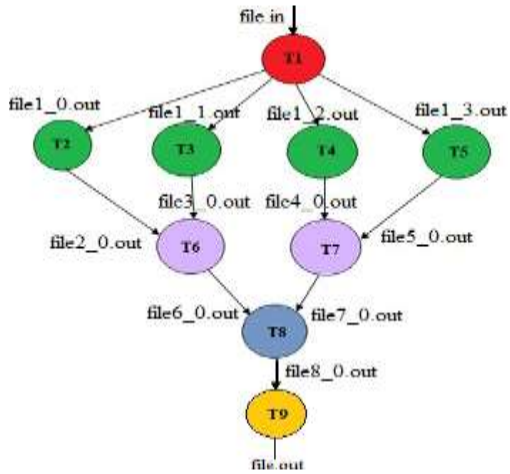
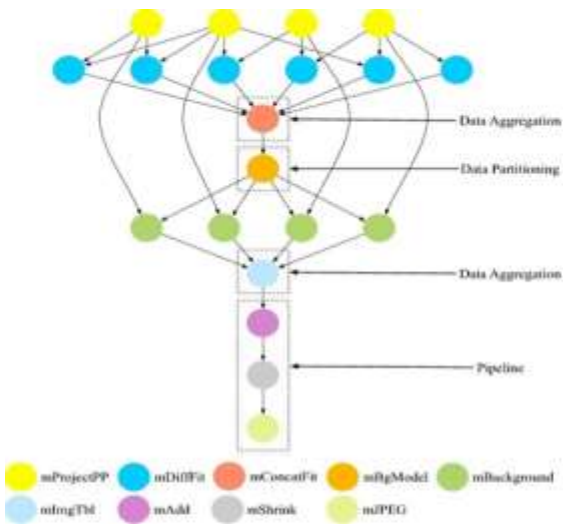


Figure 1.3 Sample Workflow with nine tasks. Nodes represent tasks and edges represent data dependencies between the tasks

Pautasso and Alonso (2006) described the characterization of various computation models that can be used for the optimization of large scale scientific workflow optimization. Workflow applications can be Memory intensive, CPU intensive, I/O Intensive or data intensive based on the nature of problem they are intended to solve. Memory intensive workflow requires high physical memory usage. CPU intensive workflows spend more time in computations. I/O intensive workflow application spend majority of their time in I/O operations and data intensive workflow applications has higher workloads to manage than computational load.

Sample Workflow Applications

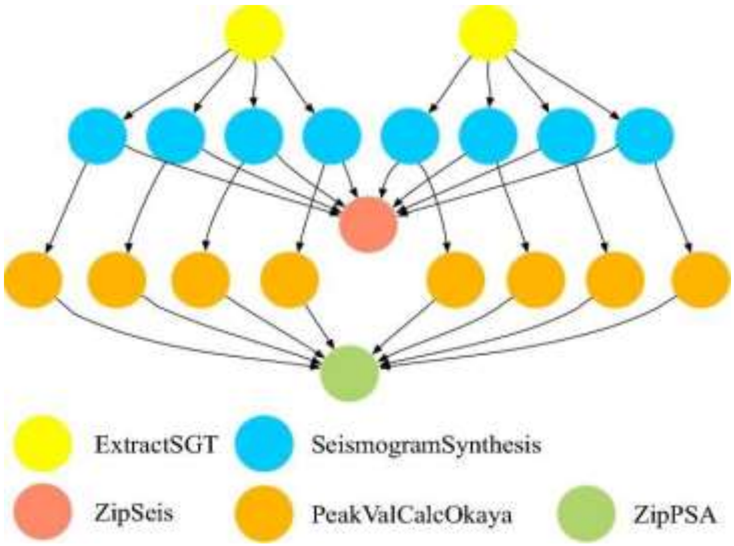
Scientific areas embrace workflows for expressing various intense computational problems that can be processed in distributed systems.



Source:https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator

Figure 1.4 Montage Workflow

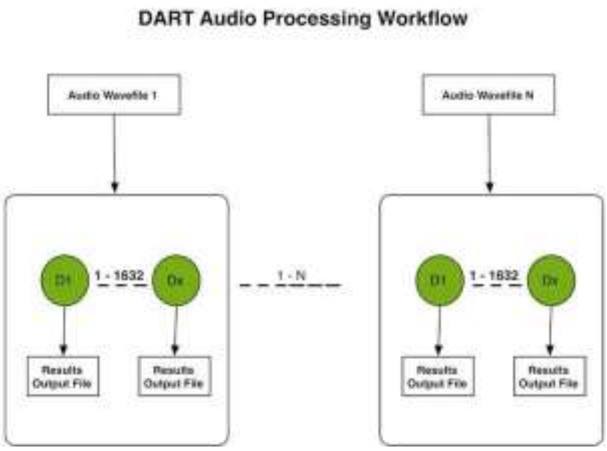
Montage is an astronomical application created by NASA/IPAC Infrared Science Archive as open source toolkit. It is an I/O Intensive application used to create the mosaic images of the sky based on the set of input images. It facilitates the production of composite images of a sky, which will be difficult to capture by the astronomical cameras. At the time of workflow execution, the output image geometry is calculated from that of the input images. The images are then re-projected to be the same spatial scale & rotation and background emissions are adjusted to be of the same level in all images. The application has been represented as a workflow as shown in Figure 1.4 and that can be run in a distributed environment. Cybershake is an earthquake hazard characterization workflow which is both data and memory intensive application used in SCEC – Southern California Earthquake Center. A sample workflow structure of cybershake is depicted in the Figure 1.5.



(Source: https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator)

Figure 1.5 Cybershake Workflow

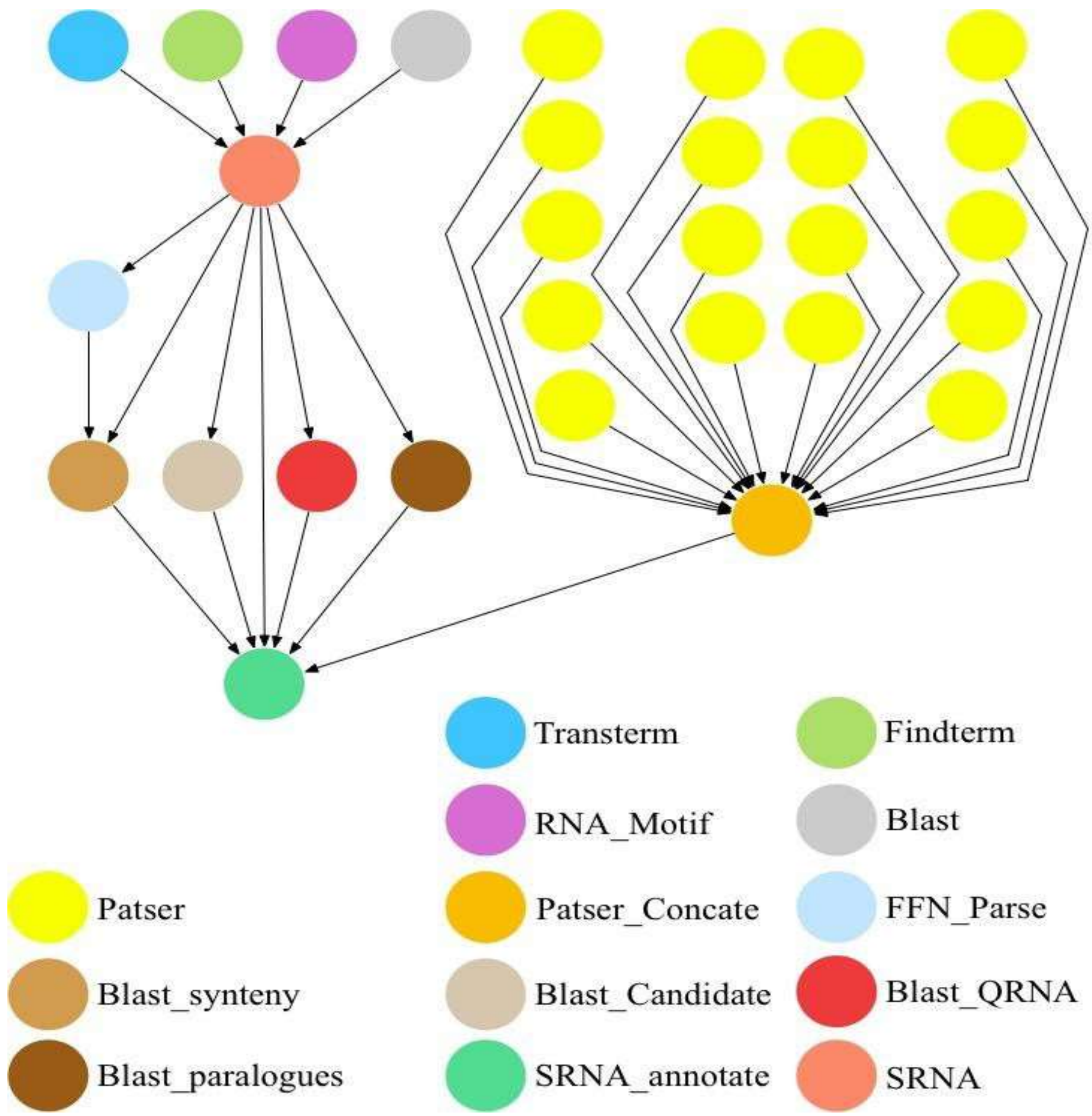
Another example of workflow is Distributed Audio Retrieval using Triana (DART) is application framework developed for the audio analysis, with a Musical Information Retrieval (MIR). It uses DART MIR platform to determine the optimal parameter setting. DART is a flexible platform for conducting MIR research and experiments. The DART workflow is depicted in the Figure 1.6.



(Source: https://pegasus.isi.edu/workflow_gallery/)

Figure 1.6 DART Audio Processing Workflow

SIPHT is a bioinformatics workflow which, is used to automate the searching process of sRNA encoding genes for bacterial replications in National Center for Biotechnology Information (NCBI). The Structure of the workflow is represented in the Figure 1.7.



(Source: https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator)

Figure 1.7 SIPHT Workflow

Other workflow examples include Glimmer, Gene2Life, MotifNetwork and MEME-MAST. Glimmer (Gene Locator and Interpolated Markov ModelER) is a system to find genes in microbial DNA using Interpolated Markov Models. Gene2Life is a Biomedical Workflow application used in molecular biology analysis. It takes DNA sequence as Input and searches the gene database to find the matched DNA Sequence. MotifNetwork is a collaborative project between RENCI and NCSA. It is a compute intensive biomedical workflow application. It provides access to the domain analysis of genome sized collection of input sequences.

MEME-MAST is a sequential biomedical workflow allows user to find motifs in DNA or protein sequences and then search the sequence database for the recognized motifs.

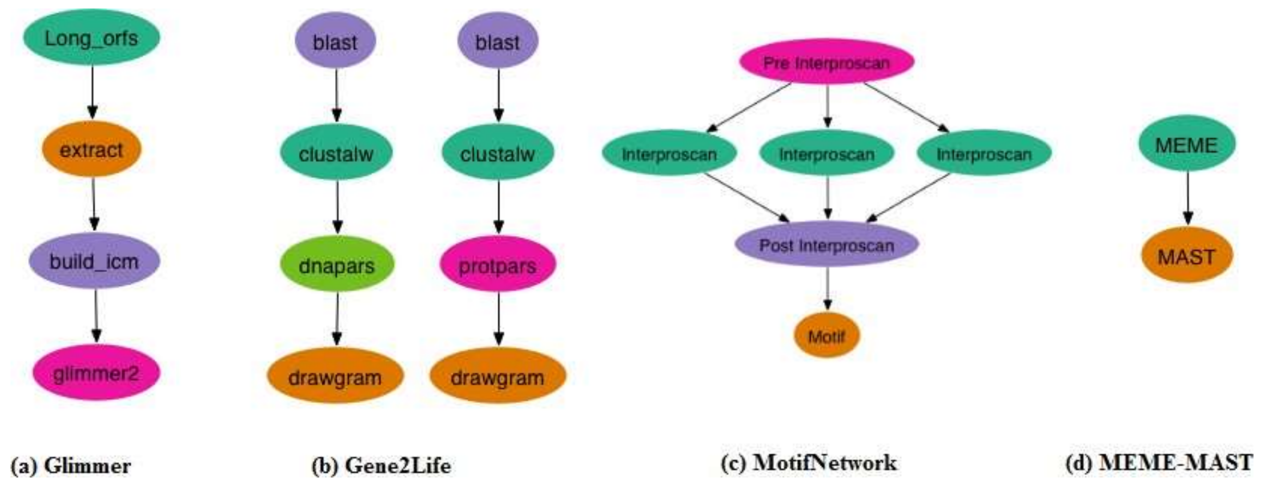


Figure 1.8 Structure of Workflow

Ramakrishnan & D. Gannon (2008) has investigated the detailed characteristics of the biomedical workflows. Figure 1.8 depicts the approximate structure of the above discussed Workflow applications. The Directed Acyclic Graph in XML (DAX) file for all the workflows applications is available in the Pegasus Workflow Gallery <https://pegasus.isi.edu/ workflow_gallery/>.

IV. STATE OF ART CLOUD WORKFLOW SCHEDULING ALGORITHMS

Indrajeet Gupta and colleagues (2016) suggested a two-phase workflow scheduling method with a priority system with the goal of reducing the amount of time needed for total processing while simultaneously increasing the amount of time that the cloud was used on average. The first phase consists of task prioritisation, which places each task in a global queue for scheduling after assigning priority for each task in a workflow based on the average ratio of communication cost to the average computation cost. This is done so that the tasks can be completed in the most efficient order possible. In the second step, the relevant virtual machine (VM) for the chosen task is determined according to the job's priority, and the task is mapped to the appropriate virtual machine that provides the quickest possible execution time. Malawski et al. (2015) developed a novel workflow scheduling algorithm that is based on the static and dynamic strategies for task scheduling and resource provisioning for workflow ensembles in cloud ensembles with a notion to maximise the completion of user-prioritized workflows within a given budget and deadline. The algorithm was published in the journal Computers in Industry. To find a solution to the issue, they came up with Dynamic Provisioning Dynamic Scheduling (DPDS), Workflow Aware Dynamic Provisioning and Scheduling (WA-DPDS), and Static Provisioning Static Scheduling (SPSS). The DPDS algorithm is an online scheduling technique that consists of a provisioning operation and a scheduling procedure. The provisioning procedure is based on the utilisation of resources based on a threshold, and the scheduling procedure schedules the ready task from the priority queue based on the priority of workflow and maps it to the idle VM. Both of these procedures are determined by the utilisation of resources based on a threshold. It ensures that a task with a lower priority will be postponed when a job with a higher priority is available, despite the fact that the lower priority jobs will continue to consume the resources. When there are no tasks that have a high priority. Since there is no preemption method being employed, tasks with a lower priority are allowed to take up more time, which causes jobs with a higher priority to be delayed. Since it does not utilise the structural information of the workflow for scheduling, it is possible that it is unaware of the budget and deadline limits. As a result, it schedules the task with the lower priority and delays the job with the higher priority. The Workflow aware DPDS is an extension of the DPDS that prevents this situation from occurring by initiating the workflow admission mechanism whenever it detects the first tasks of a new workflow in the priority queue. It makes an estimate of the amount of money that is still available, and if there is not enough money, the workflow will not be accepted, and the tasks will be withdrawn from the priority queue. If there is enough money, the job may be executed. As a result, it is able to properly manage the tasks with a lower priority and reject the workflow that has a high cost and is budgeted. SPSS is responsible for the creation of the provisioning and scheduling plan in advance, and this plan is designed in such a way that it only permits the workflow that meets the specified budget and time limit.

Amandeep verma and Sakshi Kaushal (2015) developed a Budget and Deadline Constrained Heuristic (BDHEFT) for scheduling workflow activities in the cloud environment. This heuristic is based upon the heterogeneous earliest finish time. During the scheduling of tasks among the available resources, it takes into account the limits of the budget and the deadline, and it displays the tradeoff between the amount of time it takes to complete the work and how much it costs. A service level scheduling phase and a task level scheduling phase make up BDHEFT. Both phases are separate from one another. During the service level scheduling phase, tasks are chosen based on their rank, and the appropriate resource for each task is constructed with the help of six defined variables. These variables include the current task budget (CTB), the current task deadline (CTD), the budget adjustment factor (BAF), and the deadline adjustment factor (DAF) (DAF). After then, the phase of task-level scheduling chooses the most appropriate resource for each individual job, taking into account both the available funds and the impending deadline.

Both an optimum and a heuristic scheduling approach were presented by Moise and Jerry (2016) in order to optimise the cost of scheduling DAGs on the IaaS platform. By breaking the issue down into its component parts—namely, the number of allocated VMs, the size of the allocated VMs, and the scheduling of workflow tasks on allocated VMs—it is possible to discover the most cost-effective method for scheduling workflow activities. The heuristic approach uses the results of the brute force technique as an evaluation baseline. The brute force algorithm works to lower the execution cost by repeatedly going through all of the possible scheduling possibilities. Since the optimum scheduling method has a high time complexity, the heuristic scheduling approach is offered for the scheduling tasks and the VM size selection in order to lower the time complexity. This was done so that the optimal scheduling algorithm could be used. After putting the task on the VM size, the task cost should be as low as possible; however, the amount of time spent by the VM idle and by the job making span should be kept to a minimum in order to ensure that the total amount of time spent executing the work is distributed evenly across the VMs. Workflow application maketime may be cut down by more efficient scheduling of processes using DAG levels.

In their 2015 paper, Arabnejad and Bubendorfer introduced a Proportional Deadline Constrained (PDC) algorithm for workflow scheduling in the cloud. Their goal was to reduce costs while still adhering to deadline requirements.

PDC is comprised of four distinct processes, including workflow levelling, deadline distribution, task selection, and instance selection for the purpose of scheduling workflows. Workflow levelling distributes tasks over many levels in order to optimise parallelism while still maintaining dependencies, and the deadline distribution step sends a portion of the user's deadline to each of the levels that have been identified. The tasks that will be completed are chosen during the task selection phase based on the priority, which is determined by a downward rank, and the optimal resource for scheduling is chosen during the instance selection phase by concentrating on the balance that must be struck between cost and time. Heterogeneous Budget Constrained Scheduling (HBCS) is a method that was described by Arabnejad and Barbosa (2014). Its purpose is to reduce the amount of time spent executing a task while simultaneously ensuring that its associated costs remain within a user-specified budget. The algorithm consists of a task selection phase and a processor selection phase, and it begins by computing two schedules for each workflow task: one schedule with a minimum execution time and a high cost, and another schedule with the lowest cost. Using this scheduling information, the user decides whether or not the resource will execute the task within the user-defined deadline or budget. During the task selection step, the task is chosen based on the priority, which is determined by utilising an ascending rank, and during the processor selection phase, the resource is chosen that will reduce the overall amount of time and money spent executing the work. A Just-in-Time (JIT-C) scheduling algorithm was suggested by Jyoti Sahni and Deo Prakash (2015) for the purpose of scheduling processes in a cloud environment. The aim of the system was to reduce costs while still fulfilling user-defined deadlines. In order to arrive at the correct scheduling choice, it takes into consideration the variable performance of the virtual machines as well as the time it takes to acquire new instances.

The particle swarm optimization scheduling approach that was introduced by Suraj Pandey and colleagues (2010) was designed to minimise the cost of operating workflow applications in the cloud. When optimising for cost, both the cost of calculation and the cost of transmission are taken into account. The workflow scheduling optimization process consists of two components: I a scheduling heuristic, and ii) task-resource mapping optimization. Both of these components are included in the optimization process. By maintaining the interdependence between workflow tasks, which is calculated by PSO, the tasks are chosen for selection based on the amount of time it takes to both execute and communicate the job. Since it is an online scheduling calculation, and because communication costs are updated frequently in the scheduling heuristic, the suitable resource is chosen according to the most current network and resource circumstances.

SaaS Cloud-Partial Critical Paths (SC-PCP) method was introduced by Abrishami and Naghibzadeh (2011). This algorithm is a QoS-based workflow scheduling system that seeks to reduce the execution cost while still fulfilling customer demands. Specified deadline. The PCP algorithm makes an effort to schedule the critical task, also known as the tasks that are present in the critical path, to the resources that can carry out the task with the lowest possible cost of execution. This is done with the goal of reducing the total cost of the path while ensuring that all of the tasks are completed before the finish time. When it comes to scheduling work, it utilises three distinct policies, which are the optimal policy, the lower cost policy, and the fair policy. The optimised strategy assigns each job to the resource with the highest rate of productivity, ensuring that the work is completed far before its latest target completion time, which is costly in general. The reduction cost policy is used in order to pick the most cost-effective resource for workflow scheduling, with the goal of ensuring that the job is completed in a timely manner without going over its allotted budget. A fair policy is the same as a policy that reduces costs; in addition, it has the advantage of rescheduling the activities, which increases dependability.

Pareto optimum scheduling heuristic (POSH) was presented by Su et al. (2013). It is based on the Heterogeneous Earliest Finish Time (HEFT) algorithm and uses the Pareto dominance notion as its foundation. The Weighting phase, the Prioritizing phase, and the Mapping phase are the three steps that are included in the POSH scheduling process. During the weighting phase, the weight of each node is determined by the amount of time it took to complete the job. The weight of each edge is determined by the amount of time it took to transmit data. In the prioritisation phase, the tasks are sorted according to the ascending priority of each job, which is determined by the weight of the node in conjunction with the execution time of each child task. The mapping step allocates resources to the chosen task using the pareto dominance notion in such a manner as to reduce the amount of time needed to complete the chosen activity while simultaneously lowering its associated cost.

Elzeki et al. (2013) suggested a revised version of the Max-Min scheduling method, which takes into account the influence of RASA as well as the Max-Min approach. It chooses jobs to do based not on how long a task takes in total but rather on how long it is predicted to take to execute each task. The predicted amount of time it will take to complete each of the tasks that have been submitted is determined by the algorithm. After then, the activity with the highest priority execution time is selected for mapping on the resource that has a minimum overall completion time and the scheduled tasks are eliminated from the Meta tasks list and the calculated execution times are updated.

Toktam Ghafarian & Bahman Javadi (2015) proposed the scheduling algorithm which maximizes the resource utilization and increases proportion of workflows that meets the deadline. The idea is to partition the workflows to minimize its dependencies and schedule it to the distributed resources according to the proximity and load balancing policies of resources. The execution time is estimated prior to the resource allocation and tasks are allocated to the appropriate resource. If any sub workflow misses its sub-deadline then that task is reallocated to the public cloud resources for the betterment of system performance.
Ghasemzadeh et al. (2016) proposed a Deadline-Budget Workflow Scheduling (DBWS) for the minimization of cost and time of workflow execution in cloud environment. Tasks are divided into various levels depends on their depth of rank and the selection is based on the priority of upward rank. The maximum execution length of each task in appropriate level are computed and assigned. Resource selection depends on the tradeoff between time and cost and is computed by defining a sub deadline time for each task, which is computed from the applications deadline.

Ghorbannia Delavar & Yalda Aryan (2014) proposed HSGA a hybrid heuristic method for scheduling workflows based on a genetic algorithm with a notion to minimize the makespan and loadbalance. Workflow tasks are prioritized depends on graph topology, as this is an effective approach for completion time reduction. Resources are chosen based on the execution time. A best resource which executes the selected task in minimum time is selected based on the fitness value, which is computed using the time and failure rate of task execution on each resource.

Choon Lee et al. (2015) presented a Maximum Effective Reduction(MER) algorithm for a better resource efficient solution which optimizes the usage of resources in scheduling environment. MER optimizes the workflow scheduling by task consolidation approach in two ways: Idle time slots filling and squeezing. Resources with lesser execution tasks are identified and are likely to be shifted to other resources, if shifting enhances the efficiency of resource. Consolidation degree is adjusted by changing the delay limit in makespan. It engage three phases for scheduling which includes

(1) delay limit identification, (2) task consolidation and (3) resource consolidation for scheduling. Delay limit identification phase identifies the subset resources with makespan increase bounded by delay limit and the task consolidation phase considers the remaining resources for consolidation. The aforementioned resources are sorted in decreasing order of their resource efficiency and considered for consolidation with other resources. The resource which incurs the minimum makespan increase is selected for task execution. Resource consolidation phase identifies the unused resources and synthesizes it for the effective use of other tasks which leads to the increased performance of resource utilization.

V. CONCLUSION

The need of efficient algorithms for the selection of optimized resources from the heterogeneous resource pool was the main inspiration behind this research and so in line with that the thesis presented the development of multi-objective workflow scheduling algorithms and the experimental results for the same. The thesis also introduced the Utilization factor which will be helpful in reducing the host energy consumption. The Focus of the research was directed to the reduction of time, cost and energy which had also concentrated on the improved resource utilization. The thesis records experimental evaluation with the performance metrics such as Execution time, Execution cost, Energy consumption and resource utilization which reveals the implemented algorithms‟ performance to be comparatively better than the existing algorithms. Diverse applications with different tasks have been used for the analysis of the considered algorithms.

VI. REFERENCES

[1]. Allcock, I. Foster, V. Nefedova, A. Chervenak, E. Deelman, C. Kesselman, J. Lee, A. Sim, A. Shoshani, B. Drach, & D. Williams,2001,‟High-performance remote access to climate simulation data: a challenge problem for data grid technologies‟, in Supercomputing ‟01: Proceedings of the ACM/IEEE conference on Supercomputing. New York, NY, USA: ACM, pp. 46–46.

[2]. Amandeep Verma, Sakshi Kaushal 2015,‟Cost-Time Efficient Scheduling Plan for Executing Workflows in the Cloud‟ in Journal of Grid Computing, vol. 13, no. 4, pp 495

[3]. Anju Bala & Inderveer Chana,2015,‟Multilevel Priority-Based Task Scheduling Algorithm for Workflows in Cloud Computing Environment‟ in Advances in Intelligent Systems & Computing, vol. 1, no.408, pp. 685- 693.

[4]. Anton .Beloglazov, R. Buyya, Y.Lee, A.Zomaya, (2011) „A Taxonomy & Survey of Energy-Efficient Data Centers & Cloud Computing‟, Journal of Advances in Computers. 82, 47-111.

[5]. Anton Beloglazov, Jemal Abawajy, & Rajkumar Buyya. 2012,‟Energy- aware resource allocation heuristics for efficient management of data centers for Cloud computing‟ in Future Generation. Computer Systems. 28, 5 (May 2012), pp. 55-768. DOI=http://dx.doi.org/10.1016/ j.future.2011.04.017.

[6]. B. Arun Kumar, T. Ravichandran, „Time & Cost Optimization Algorithm for Scheduling Multiple Workflows in Hybrid Clouds, „ in European Journal of Scientific Research vol. 89 no. 2 October 2012, pp.265-275.

[7]. Beloglazov, A.; Buyya, R. 2012, „Optimal online deterministic algorithms & adaptive heuristics for energy & performance efficient dynamic consolidation of virtual machines in cloud data centers‟ in Concurrency And Computation: Practice And Experience. 2012,vol 24, pp. 1397–1420.

[8]. C. Isci, J. Liu, B. Abali, J. O. Kephart, & J. Kouloheris, „Improving server utilization using fast virtual machine migration,‟ IBM J. Res. Develop., vol. 55, no. 6, pp. 1–12, Nov. 2011.

[9]. Ciufudean, C., Neri, F, 2014,‟Open research issues on Multi-Models for Complex Technological Systems‟, WSEAS Transactions on Systems, vol.13.

[10]. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good,A. Laity, J. C. Jacob, and D.S. Katz, 2005,‟Pegasus: Aframework for mapping complex scientific workflows onto distributed systems‟, in Scientific Programming, vol. 13, no. 3, pp. 219–237.

[11]. Ehab Nabiel Alkhanak, Sai Peck Lee, Reza Rezaei, Reza Meimandi Parizi, 2016, „Cost optimization approaches for scientific workflow scheduling in the cloud & grid computing: A review, classifications, & open issues, Journal of Systems & Software‟, vol. 113, March 2016, pp. 1-26, http://dx.doi.org/10.1016/j.jss.2015.11.023.

[12]. Ewa Deelman, Karan Vahi, Mats Rynge, Gideon Juve, Rajiv Mayani, Rafael Ferreira da Silva,2016‟,Pegasus in the Cloud: Science Automation through Workflow Technologies‟, IEEE Internet Computing, vol. 20, no. , pp. 70-76, Jan.-Feb.

[13]. Fang Liu, Jin Tong, Jian Mao, Robert B. Bohn, John V. Messina, Mark L. Badger, Dawn M. Leaf 2011,‟ NIST Cloud Computing Reference Architecture‟ in Special Publication (NIST SP),pp. 500-292.

[14]. Faris Llwaah, Nigel Thomas, Jacek Ca la,2015,‟Improving MCT scheduling algorithm to reduce the makespan & cost of workflow execution in the cloud‟ in 31st UK Performance Engineering Workshop 17, pp. 1 – 8.

[15]. G. Juve, A. Chervenak, E. Deelman, S. Bharathi, G. Mehta, & K. Vahi, 2013, „Characterizing & profiling scientific workflows,‟ Future Generation Computer Systems, vol. 29, no. 3, pp. 682–692.

[16]. G. Mehta, E. Deelman, J. A. Knowles, T. Chen, Y. Wang, J. Vöckler, S. Buyske, & T. Matise, 2011,‟Enabling Data & Compute Intensive Workflows in Bioinformatics‟, in 2nd International Workshop on High Performance Bioinformatics & Biomedicine (HiBB 2011) in conjunction with Euro-Par Conference.

[17]. Garg, R. & Singh, A.K. 2016‟ Energy-Aware Workflow Scheduling in Grid Under QoS Constraints‟ in the Arabian Journal for Science & Engineering vol. 41, no.2,pp. 495-511. doi:10.1007/s13369-015-1705-y

[18]. Ghorbannia Delavar, A. & Aryan, Y. 2014, „HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems‟ in Cluster Computing vol 17:pp. 129-. 137 doi:10.1007/s10586-013-0275-6.

[19]. Guangjie Han , Wenhui Que , Gangyong Jia & Lei Shu, 2016,‟An Efficient Virtual Machine Consolidation Scheme for Multimedia Cloud Computing‟ in the journal of sensors , vol 16,no. 246; pp.1 – 17, doi:10.3390/s16020246.

[20]. GueArout T, Monteil T, Da Costa G, Neves Calheiros

R, Buyya R, Alexandru M. Energy-aware simulation with DVFS. Simulation Modelling Practice & Theory. 2013; vol. 39: pp. 76-91. doi: 10.1016/j.simpat. 2013.04.007

[21]. Gupta, M. S. Kumar & P. K. Jana, 2016, „Compute-intensive workflow scheduling in multi-cloud environment," International Conference on Advances in Computing, Communications & Informatics (ICACCI), Jaipur, pp. 315-321.doi: 10.1109/ICACCI.2016.7732066

[22]. Gupta, M. S. Kumar & P. K. Jana, 2016,"Transfer time-aware workflow scheduling for multi-cloud environment," in International Conference on Computing, Communication & Automation (ICCCA), Noida, pp.732-737.doi: 10.1109/CCAA.2016.7813824.

[23]. H. Topcuouglu, S. Hariri, M.-y. Wu, 2002," Performance-effective & low-complexity task scheduling for heterogeneous computing", IEEE Trans. Parallel Distrib. Syst. 13 (3) (2002). pp.260–274. http://dx.doi.org/10.1109/71.993206

[24]. Hamid Arabnejad, Jorge G. Barbosa,2014,"A Budget Constrained Scheduling Algorithm for Workflow Applications" in Journal of Grid Computing, vol. 12, no. 4, pp. 665-679.

[25]. HeeSeok Choi, JongBeom Lim, Heonchang Yu, & EunYoung Lee, 2016, „Task Classification Based Energy-Aware Consolidation in Clouds," Scientific Programming, Article ID 6208358, pp. 1-13, doi:10.1155/2016/6208358.

[26]. https://confluence.pegasus.isi.edu/display/pegasus/WorkflowGenerator

[27]. J. D. Ullman, „Np-complete scheduling problems",1975, In the Journal of Computer & System Sciences, vol. – 10, no.-3, pp 384-393.

[28]. J. Yu & R. Buyya, 2005,"A Taxonomy of Workflow Management Systems for Grid Computing", in the Journal of Grid Computing, vol.3, no. 3, pp. 171–200.

[29]. Jyoti Sahni, Deo Vidyarthi, 2015, „A Cost-Effective Deadline- Constrained Dynamic Scheduling Algorithm for Scientific Workflows in a Cloud Environment", in IEEE Transactions on Cloud Computing, pp. 1, 5555, doi:10.1109/TCC.2015.2451649.

[30]. L. Hongyou, W. Jiangyong, P. Jian, W. Junfeng & L. Tang, 2013"Energy-aware scheduling scheme using workload-aware consolidation technique in cloud data centres," in China Communications, vol. 10, no. 12, pp. 114-124, Dec. 2013. doi: 10.1109/CC.2013.6723884.

[31]. L. Ramakrishnan & D. Gannon, 2008, „A Survey of Distributed Workflow Characteristics & Resource Requirements", Indiana University Technical Report TR671.

[32]. L. Ramakrishnan, D. Gannon, 2008, A survey of distributed workflow characteristics & resource requirements, Technical Report TR671, Indiana University.

[33]. Li, J, Ma, T, Tang, M., Shen, W, Jin Y,2017," Improved FIFO Scheduling Algorithm Based on Fuzzy Clustering in Cloud Computing" in journal of Information vol. 8,no.1,pp. 25.DOI: 10.3390/info8010025.

[34]. Liu. K, J. Hai, C. X. L. Jinjun, Y. Dong, 2010, „A compromised time- cost scheduling algorithm in SwinDeW-C for instance-intensive cost- constrained workflows on cloud computing platform",in Journal of High Performance Computing Applications , vol.,no.24, no.4, pp: 445– 456.

[35]. M. Adhikari & T. Amgoth, 2016, „Efficient algorithm for workflow scheduling in cloud computing environment," Ninth International Conference on Contemporary Computing (IC3), Noida, 2016, pp. 1-6. doi: 10.1109/IC3.2016.7880222.

[36]. M. H. Kumar & S. K. Peddoju, 2014 „Energy efficient task scheduling for parallel workflows in cloud environment," 2014 International Conference on Control, Instrumentation, Communication & Computational Technologies (ICCICCT), Kanyakumari ,pp.1298- 1303.doi: 10.1109/ICCICCT.2014.6993161.

[37]. M. J. Fairman, A. R. Price, G. Xue, M. Molinari, D. A. Nicole, T. M. Lenton, R. Marsh, K. Takeda, & S. J. Cox, 2009,"Earth system modelling with Windows Workflow Foundation",in Future Generation Computer Systems, vol. 25, no. 5, pp. 586–597.

[38]. M. Malawski, G. Juve, E. Deelman & J. Nabrzyski, 2012, „Cost- & deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds," High Performance Computing, Networking, Storage & Analysis (SC), 2012 International Conference for, Salt Lake City, UT, pp. 1-11.doi: 10.1109/SC.2012.38

[39]. M. Rynge, G. Juve, J. Kinney, J. Good, B. G. Berriman, A. Merrihew, & E. Deelman, 2013," Producing an Infrared Multiwavelength Galactic Plane Atlas using Montage, Pegasus & Amazon Web Services", in 23rd Annual Astronomical Data Analysis Software & Systems (ADASS) Conference.

[40]. Moise W. Convolbo, Jerry Chou, 2016, „Cost-aware DAG scheduling algorithms for minimizing execution cost on cloud resources"in The Journal of Supercomputing, vol. 72, no.3, pp 985.

[41]. Mozhgan Ghasemzadeh & Hamid Arabnejad & Jorge G. Barbosa, 2016," Deadline-Budget constrained Scheduling Algorithm for Scientific Workflows in a Cloud Environment" in 20th International Conference on Principles of Distributed Systems (OPODIS 2016, vol 70, pp 19:1-19:16, DOI:

10.4230/LIPIcs.OPODIS.2016.19

[42]. N.Mohanapriya, Dr.G.Kousalya, & Dr.P.Balakrishnan, 2016, „Cloud Workflow Scheduling Algorithms: A Survey", in International Journal of Advanced Engineering Technology, vol. VII, Issue-III.,Pg.no.:188- 195.