Q-Aware: A new approach towards the task scheduling process using nature-inspired meta-heuristic algorithms

Research Project
MSc Cloud Computing

Midhun Kumar Nemani
x15040216

School of Computing
National College of Ireland

Supervisor: David Hamill

National College of Ireland
Project Submission Sheet – 2016/2017
School of Computing

Student Name:	Midhun Kumar Nemani
Student ID:	x15040216
Programme:	MSc Cloud Computing
Year:	2017
Module:	Research Project
Lecturer:	David Hamill
Submission Due Date:	16/08/2017
Project Title:	**Q-Aware**: A new approach towards the task scheduling process using nature-inspired meta-heuristic algorithms
Word Count:	4961

I hereby certify that the information contained in this (**Q-Aware**: A new approach towards the task scheduling process using nature-inspired meta-heuristic algorithms) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are encouraged to use the Harvard Referencing Standard supplied by the Library. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action. Students may be required to undergo a viva (oral examination) if there is suspicion about the validity of their submitted work.

Signature:	
Date:	17th August 2017

PLEASE READ THE FOLLOWING INSTRUCTIONS:

1. Please attach a completed copy of this sheet to each project (including multiple copies).
2. **You must ensure that you retain a HARD COPY of ALL projects**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. Please do not bind projects or place in covers unless specifically requested.
3. Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Q-Aware : A new approach towards the task scheduling process using nature-inspired meta-heuristic algorithms

Midhun Kumar Nemani
x15040216
Research Project in MSc Cloud Computing

17th August 2017

Abstract

Cloud Computing is the modern invocation being recited by the IT industry, and has set an example for accessing as well as configuring large-scale distributed computing applications across the clusters in datacenters. Today, it is in control of the greater percentage of computing quite recently, subsequently physical assets are being over-stacked. In-order to manage and assign tasks for the physical or virtual resources, techniques of clustering, scheduling and load balancing are being used. The current techniques are profoundly complex for non-pre-emptive tasks, which has remained as an irrecoverable restraint on the part of scheduling mechanism. In recent times, researches have implemented nature-inspired algorithms into the field of computing to solve the problems associated to complexity and to provide optimal solutions. In this paper, our focus lies in computing the makeSpan and costs incurred for resource management in the task scheduling process with a new methodology. We are proposing a heuristic approach namely Q-aware algorithm which takes the decisions based on analytical scheming. It works by forming a cluster of VMs using K-means algorithm based on the memory, CPU, and bandwidth. This proposed technique uses a pool of nature-inspired algorithms namely SA,GA,IWO,PSO,ABC for the scheduling of tasks. This experiment is performed using cloudsim as simulator in-order to create a datacenter environment and schedule the tasks onto the virtual resources. The results show that the proposed system successfully performs the hyper-analytical scheming and measures the makespan, and calculates the resource utilization during the process.

Keywords: Task scheduling, cloud computing, makespan, meta-heuristic algorithms, nature inspired Algorithms, clustering

1 Introduction

Cloud computing has been rapidly expanding and is among valuable research fields. It is comprised of various services for the users to access the infrastructure, platform and software where resource provisioning and Service Level Agreements (SLA) play a vital role. The problem to deal is how the huge number of resources are allocated by the scheduler efficiently (Mandal and Acharyya; 2015). Load balancing is very much necessary in managing the resources such as to scale up when the demand grows, by dynamically allocating the work among the resources. Cloud computing is backed by virtualization where Virtual machine (VM) runs on top of the physical hosts by sharing the physical resources such as compute, memory, networking, storage, etc. where the total processing power is a critical constrain among these computing units.

1.1 State of the Art

A number of deterministic algorithms also known as rule-based algorithms have been widely used for scheduling of jobs to the resources but were found to be under-performing for the cloud which is a room for infinite resources which is an optimization problem. These algorithms are simple and have been proved to be efficient for small scale to medium scale computing, but many researchers have termed them to be inefficient for large scale or complex scheduling strategies or problems as the results have not been satisfactory.

This implies that there is a space for the current researchers to optimize or develop a new scheduling strategy. So one of the possible solutions would be applying modern heuristics which has also attracted several researches. We have extended our research by studying various heuristics based nature inspired algorithms: ABC, GA, PSO, IWO, SA. Along with these a novel method is developed by leveraging the strengths of these 5 algorithms along with the popular clustering algorithm K-Means.

The research question: **Can the new approach namely "Q-aware technique" be integrated with k-means algorithm for clustering along with meta-heuristic algorithms namely: IWO,GA,ABC,SA,PSO to determine the makespan value and the cost incurred by the resources?**

Aim of the project is to select an optimal algorithm during the process of scheduling and to calculate the makeSpan by choosing a task deployment strategy during the allocation of tasks to the VMs in the clusters bearing in mind the existing static and dynamic load balancing techniques and the heuristic approaches proposed in various papers into consideration. We will keep our focus on the research by taking resource requirements into account, while stressing on evolutionary algorithms discussed in the literature review to date.

1.2 Traditional Scheduling

Traditionally, scheduling of jobs to resources has been attributed as finding a solution to the optimization problem for a set of tasks T=T1, T2 ,T3..,Tn, for a given set M=M1, M2, M3,..,Mm Machines bound to predefined set of constrains. When machine (m) = 1, then the scheduling problem as per (Pinedo; 2012) is referred to as uni-processor or single machine scheduling problem. When machine (m) = 2 then it is referred as parallel processing or multi-processor scheduling problem. Functions such as Makespan, flowtime, lateness are generally being used to measure the performance of the scheduling algorithms. One of the popular ways of representing the scheduling problem is the makespan Cmax(s), mathematically denoted as,

$$\text{Minimize } f(s) = C_{\max}(s),$$

Where s denotes candidate solution, Cj denotes the completion time of job j, $C_{\max}(s) = \max_j C_j$ is the completion time. The scheduling problem later has extended from single tasking to multitasking.

1.3 Scheduling on Cloud

Scheduling the resources on Pay as you use model is attributed to cloud computing, and the problem associated to it is classified to workflow scheduling problem. This can be further divided as service level and task level. Service level can be described as static i.e. at platform layer and task level designates the dynamic nature at resource layer (Wu et al.; 2013).Cloud and grid computing often include heterogeneous resources with virtual machines running on them, and the fundamental issues were quite different as of traditional scheduling. Along with latency costs such as the bandwidth costs are also added and is also defined to be as DAG problem by couple of research studies. As per (Chrétienne and Picouleau; 1995) scheduling in cloud is denoted as

$$\begin{aligned} \text{minimize} \quad & f(s) = C_{\max}(s) + \sum_{i=1}^{n}\sum_{j=1}^{m} C_{ij}, \\ \text{subject to} \quad & C_{\max}(s) \leq U(s), \\ & C(s) \leq B(s), \end{aligned}$$

Where f(s) denotes objective function, $C_{\max}(s)$, the completion time of the last job or the total execution time of all the jobs, n as number of tasks ,m as number of machines, C_{ij}, cost of the processing the i^{th} task on the j^{th} machine. U(s) denotes the pending tasks, C(s) is the total cost of s and B(s) is the budget constrain for the tasks on (s).

1.4 Scheduling Types

A scheduling system is directed by the scheduling algorithm to process the task requests. For a real-time scheduling system, usually each task is consigned with a deadline interval, description, and priority. It also determines how the priorities are mapped to a specific task. A scheduling algorithms can be categorized as static scheduling or dynamic scheduling.

1.4.1 Deterministic scheduling algorithms

Static algorithms have been in use conventionally for the scheduling purposes. They are deterministic and schedule the tasks in a known environment i.e. it contains all the task information and has a pre-plan strategy in mapping the resources prior to execution process. Static scheduler generates patching table. In contrary, Dynamic Scheduling does not just rely on the submitted tasks to the cloud resources but also the present state of systems and computer machines to make a scheduling decision. As stated in (Magoulès et al.; 2012) it makes decisions during the run-time as opposed to the static scheduler which takes the decisions pre-run-time. We will be further discussing about the advantages and dis-advantages while comparing with Non-deterministic algorithms in the next section.

1.4.2 Meta-Heuristic scheduling algorithms

Meta-heuristic algorithms uses transition, evaluation, and determination unlike the static algorithms which are rule based. The "Transition" works by constructive or perturbative method or both to create a solution which will be followed by evaluation operator to evaluate it by using a pre-defined value. Determination will further enhance the solution space to search for a better solution. As said by (Yang; 2010) in Nature-inspired optimization algorithms simple heuristic and meta-heuristic algorithms are classified from Stochastic algorithms which are very close and share a marginal difference. Heuristic is driven by to find and discover by using trial and error methods to find quality solutions in a reasonable amount of time to tough problems. These have been further enhanced or developed as Meta-heuristic algorithms which are considered to be higher level than the heuristic algorithms. With the evolving trends all the algorithms and techniques which are bound by randomization have been classified as meta-heuristic which are intended to move away from local search and work towards finding a solution in global scale. This leads to an optimized solution for the global optimization problem.

In practice, although deterministic algorithms (DA) as per (Morton and Pentico; 1993) have shown improved results in comparison with traditional exhaustive algorithms due to its nature of design they have an issue when handling large-scale scheduling. Meta-heuristic algorithms work with iterations to find the optimal solutions quickly when compared with the traditional techniques. Plethora of research has been done on these techniques which was discussed in (Mandal and Acharyya; 2015) and were proved to show better scheduling results than traditional scheduling algorithms towards the optimization problem.

Algorithms by Category	Pros	Cons
Gradient based	Convergence rate is higher	Not easy to estimate
Deterministic	Efficient for local search	No guarantee for optimality
Nature-inspired meta-heuristics Stochastic Swarm Intelligence	Global optimality Simple and flexible Can address hard and diverse range of problems Easy to implement in parallel	Higher computational costs Solutions cannot be the same May not be efficient for local search and simple problems

Table 1: Comparitive Analysis of Algorithms

The research has set a couple of objectives to achieve the desired output.
Below are the pre-requisites that are to be satisfied:
a) Cluster formation from the available set of virtual machines based on its characteristics.

b) Meta-heuristic algorithms which are to be considered for scheduling the tasks onto resources which are proved to work towards the optimization problem in cloud environments.
c) Creation of self-adaptive methods for the analysis and improvisation of the existing solutions.
We hope that this research and the experiments are intended to benefit the cloud service providers in providing the Quality of Service(QoS) to the consumers while meeting their SLAs with reduced costs and fellow researchers to set the benchmarks for reliability in the process of task scheduling. The research paper is organized as follows, section 2 brings about various scheduling strategies which have been followed and their limitations. section 3 introduces a new methodology to overcome few of these limitations.

2 Related Work

2.1 Scheduling Strategies

Cloud computing is certainly expedient for the tasks which are deliberated to be challenging or time-consuming or which can be expensive in some cases depending on the various scenario. As described by (Magoulès et al.; 2012), Scheduling has become more complicated as the computing and storage resources are decoupled and the service scheduling makes it a unique case when compared to other computing standards. A single generic scheduler might not fit the whole environment but there is a need for a centralized scheduler which works efficiently. Section 1 discusses various types of algorithms and will contrast as well as cherish the related work and efforts put front by the previous researchers. This will help in improvising the existing work by bridging the gaps.

2.1.1 Traditional Algorithms

Traditional algorithms which have been used in the past have been regarded as time consuming. Popular full search algorithms which are in this category are dynamic programming branch-and-bound. This is due to the numerous checks they have to undergo during the scheduling process and in case of deterministic algorithms such as EDD, they are trapped into local optima (Garey et al.; 1976) (Parsa and Entezari-Maleki; 2009)proposed a new task scheduling algorithm RASA (Resource-Aware-Scheduling algorithm). It is composed of two traditional scheduling algorithms; Max -min and Min-min. It utilizes the rewards of Max-min and Min-min algorithms and envelops their drawbacks. Although parameters such as the limit of every task, incoming rate of the tasks, expenditure of the task execution on every resource, cost of the transmission have not been considered.

2.1.2 Nature-Inspired Algorithms

Many researches and experiments are conducted to find an optimal solution to the complex problems like NP hard problem in cloud. One such survey towards this is conducted by (Said; 2016) about the nature-inspired algorithms and their contribution towards minimizing the makespan and finding solutions for various optimization problems. Further we will discuss about various heuristic algorithms in this area, and also the experiments conducted by various fellow researchers. The overall view is depicted in in figure 2

(Pandey et al.; 2010) describe Particle Swarm Optimization(PSO) which belongs to meta-heuristic algorithms, being efficient in various applications such as data mining, pattern recognition and also NP-Hard problems in the scheduling. PSO works by adjusting the trajectory in every generation inorder to find the local best position and the global best position in the total population. The main idea behind this algorithm is to reduce the communication costs.The author by his experiments have proved to reduce the total communication costs by 3 times when compared to Best Resource Selection(BSS)algorithm. PSO has also been discussed and implemented for the purpose of task scheduling in cloud by (Ramezani et al.; 2014)and evaluated against traditional algorithms. During the scheduling process it has decreased the task migration time, which has an immediate impact on the memory utilization. This has resulted in less memory utilization, and also the overall costs incurred. The paramaters considered for this test can be classified into vm properties and task properties. (Zuo et al.; 2014) have continued the research by introducing a Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud, for the scenario of Public clouds which provide the users with IaaS which works on multiplexing of tasks in-order to deal with the challenge of scheduling. The algorithm has to meet the demands during the peak times while the QoS has to be preserved at all times as the providers are bound with SLAs. In the past, many studies have discussed regarding the active procurement of resources in the cloud via cloud federations as a solution for the sake of fulfilling these demands. However, this is not a viable solution and that be practiced as it involves complicated process of setting the SLA's between the business. To deal this, the author has used the PSO(Particle Swarm Optimization) a stochastic algorithm which is based on swarm intelligence theory, encompassing population based technique and which is already proved by (Xiao and Yu; 2008) in solving various complex optimization problems and has termed to be advantageous in finding a quick convergence with easy implementation. The parameters which have been considered are instance types, cost incurred with private cloud, and various VM details and job details. The key issue of optimally allocating of jobs to resources is formulated and dealt by an integer programming (IP) model in a hybrid cloud environment where a task is represented by a "particle" and task priorities as a whole and a scheduling strategy to update the velocity of each particle thus ensuring its diversity and robustness. The experimental approach has proved to be efficient in dealing with the issue.

(Lin and Hsu; 1990), have considered Simulated Annealing (SA) as an alternative algorithm to assign the tasks for the scheduler in-order to deal with the problems involved. They have observed that few parameters included have impacted on performance. The proposed annealing algorithm has proved to provide the efficient solution in searching an near optimal solution by in-cooperating various constraints. (Abdullah and Othman; 2014) has also done similar work and considered the real-execution time and various QoS parameters such as deadline and penalty cost. The experiments have done against the GA(Genetic Algorithm) and have concluded that when the task list is within the threshold of 100 then the performance of SA is better in terms of execution time. But when the threshold increases then the genetic algorithm performs better. In further sections we will discuss about the Genetic algorithm.

For the optimization problem (Liu and Liu; 2016) have used the Simulated Annealing to schedule the tasks in the cloud. Experiments are conducted to check the performance of SA. One such experiment was conducted and have compared the performance of SA with the traditional algorithms to determine the usage of resources by the physical machines.

The results proved that SA provides better performance when compared to the traditional algorithms in finding an approximate optimal solution with higher speeds during the process of annealing. The author has put forward that the simulated annealing need to be used with other algorithms to find an optimal solution.

Genetic Algorithm(GA) is a heuristic algorithm inspired from the nature, which generates the initial population randomly. It Selects-the-best and discards-the-rest principle, from a population. The task scheduler works to search an optimized solution from large solution spaces with the polynomial time. It is a directed search algorithm which has tried to be enhanced by many researchers in order to achieve optimal results and speed of convergence Minimum Execution Time(MET) and Min-min heuristics have been proposed in (Kaur et al.; 2010). GA has also been applied for the work scheduling as in (Singh and Kalra; 2014) and (Yu and Buyya; 2006)and proven that GA has outperformed the traditional algorithms like max-min.

In order to process the tasks efficiently among the available resources in the cloud, Ant Bee Colony(ABC) Algorithm which also comes from the family of meta-heuristic algorithms works by honey Bee swarms nature. This algorithm is known to reduce the job execution time and also removing the complexities involved in the job allocation process. (Le Dinh et al.; 2013)has proposed an algorithm where ABC algorithm is integrated with Greedy algorithm during the assigning of tasks to each processor in order to reduce the makespan with respect to hybrid cloud environments. One more experiment is conducted by (Elhady and Tawfeek; 2015) where the total number of tasks and makespan are the parameters considered inorder to compare the performance with FCFS which is a deterministic algorithm. The results proved to be positive over the traditional FCFS algorithm.

Invasive Weed Optimization(IWO) algorithm, is a population based intelligent stochastic algorithm which is inspired by the invasive behaviour of the weeds growing in the nature. For scheduling the tasks in heterogeneous as well as homogeneous clusters. High computational overhead and poor convergence are the problems addressed by (Li et al.; 2014) in his paper. They have improved the IWO algorithm and has been applied it to resolve the discrete task scheduling problems. It has shown good results by covering a large search space when compared to deterministic scheduling algorithms with higher speedup of execution. Another such paper is (Fan et al.; 2015) where K-Means clustering technique has been used in conjunction with IWO algorithm. The results have shown the cluster separation ratio and tightness as evaluation criteria, the performance of IWO combined with K mean algorithm has outperformed the traditional techniques.

Scheduling on resources are popularly classified into two categories such as hierarchical and partitioning. Numerous studies regarding the works related to the clustering of virtual machines, have concluded that k-means clustering based on partitioning introduced by (Hartigan and Wong; 1979) finds a quick solution for most clustering and location search problems. Key aspects in the process of clustering are defined to establish high homogeneity within the cluster and maintain heterogeneity outside the clusters.

Recently, alternative methods have been proposed such as Hybrid-Heuristics, which work by combining more than one heuristic algorithm in search of optimality. It is depicted in figure 1. (Laha and Chakraborty; 2009) have used the hybrid method by combining GA along with the algorithm proposed in (Nawaz et al.; 1983). This hybrid methodology have shown better results than other heuristics algorithms when used single, but has lead to increase in computational costs when compared to running of a single heuristic algorithm.

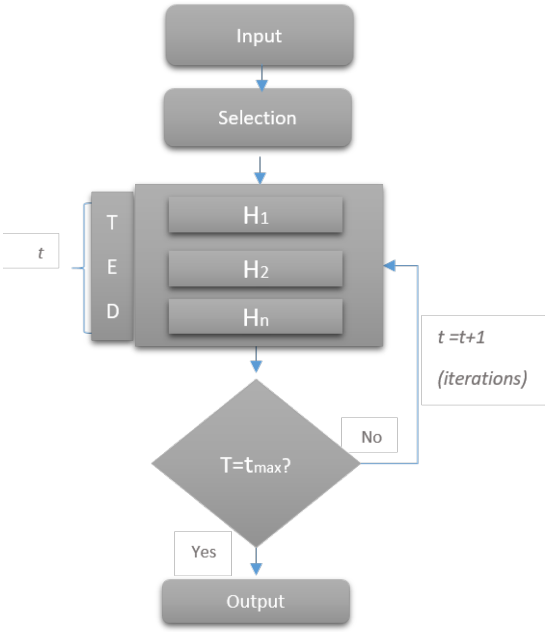


Figure 1: Hybrid heuristic method

Algorithms	Simulation Parameters	Findings
GA	Completion time calculated using communication and computational cost	Minimizes the workflow completion time. Better than FCFS Scheduling
ABC	Total processing time, Processing time, Total time saved, Number of process saved, Efficiency	Simplicity, robustness, high flexibility, Performing better than GA
PSO	File Size, Output Size, No. of VMs, MIPS, VM Memory, Bandwidth, No. of Datacenters and Hosts	Simplicity, Achieved Low computational costs
SA	Physical machines, task information, Execution time, cost parameters	Better than Mapping algorithm, and GA upto 100 tasks threshold
IWO	Initial population size, Maximum population size, Maximum iteration number, Seeds maximum Seeds minimum, Initial standard deviation, Final standard deviation, Modulation index	Solves economic load dispatch problem which minimizes the total running cost

Table 2: Comparison of Various Nature Inspired Task Scheduling Algorithms

2.2 Summary of Related Work

There are numerous articles in the related work that do not advocate the efficient task scheduling algorithms in generating better results. Although there are some specific circumstances where a few of the heuristic algorithms gave comparatively better results, but these results cannot be relied upon at all times hence, also it is complex to be benchmarked as each has their own set of parameters. After scrutinizing the related work, the meta-heuristic algorithms used in the field of computing has proved better in providing an optimal solution than the deterministic algorithms.

3 Methodology

The tenacity of this research is to create a task scheduling algorithm, and the task scheduling processes are categorized into three stages. Each stage has its own set of activities as described in Figure 2which will lead to the implementation of the project. The methodology is outlined in the following figure by three stages:

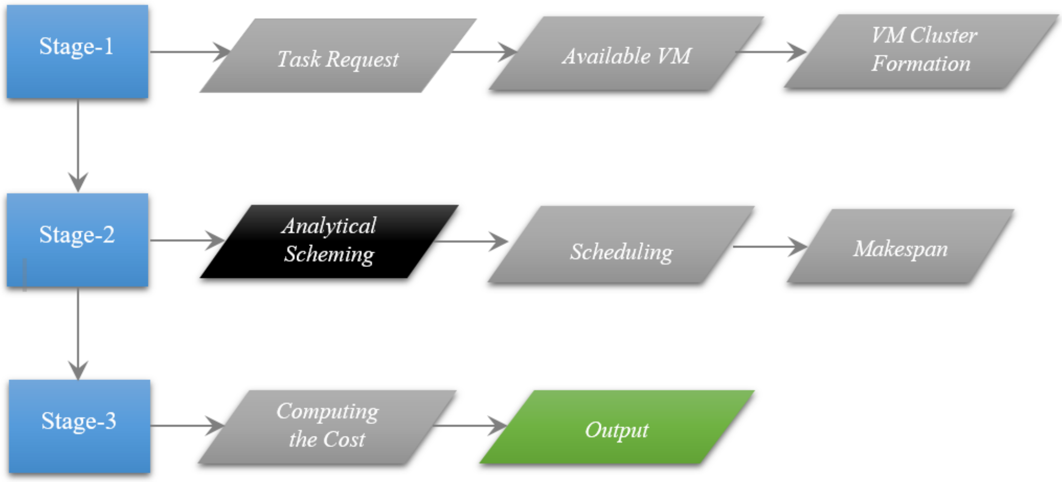


Figure 2: Design of Methodology

3.1 Methodology for First Stage: - VM Cluster Formation

Stage 1 is planned in such a way that the available virtual machine resources such as CPU, memory, bandwidth are considered as the input for the next step.
(Shidik et al.; 2016)and (Fan et al.; 2015) talks about K-Means clustering algorith, that it is proved to be energy efficient when compared to Random Choice(RC) and Minimum Migration time(MMT) in the cloud datacenters. The objects i.e. the Virtual machines are formed in sets and the recalibration is performed based on the distance to the present cluster centroid. The process of clustering oscillates until it reaches a fixed point possibly when reaching the fixed number of iterations or the centroid positions remain constant, although the number of iterations which are performed by the k-means clustering are quite a large number number.

So for clustering technique, K-means is chosen which was introduced by (Hartigan and Wong; 1979) to group the Virtual machines into K-Groups based on the mean value as showin in figure 3. The algorithm is known for its simplicity and for solving the k-means problem by finding ways to minimize the distance between the objects to the closest centre, which are denoted as centroids which was expressed below.

Steps for k-means clustering:

Let $X = \{x_1, x_2, \ldots, x_n\}$ be the set of data point and $V = \{v_1, v_2, \ldots, v_c\}$ be the set of centers.

Step1: Randomly select c cluster centers.

Step 2: Calculate the distance between each data point and cluster centers.

Step 3: Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..

Step 4: Recalculate the new cluster center using:

$$v_i = \frac{1}{c_i} \sum_{j=1}^{c_i} x_i$$

Step 5: Recalculate the distance between each data point and new obtained cluster centers.

Step 6: If no data point was reassigned then stop, otherwise repeat from step 3.

Figure 3: K-Means Algorithm

As outlined in the above steps, a set of clusters encompassing virtual machines will be formed. This can be seen in figure 8

3.2 Methodology for Second Stage: Analytical Scheming

Analytical scheming uses the diversity function and improvement function. This is to intensify the search and balance the diversification factor during the solution search, while in the process of convergence. The proposed technique consists of a pool of heuristic algorithms such as genetic algorithm, simulated annealing, particle swarm optimization, Invasive weed optimization and Ant bee Colony. So the basic idea would be the integrate the following algorithms which will take the strengths of each of them. As discussed in the previous section picks just one algorithm at every iteration unlike the hybrid algorithm which includes all the heuristics on the convergence process. In-order to do this, the two functions which are further discussed in detail in section 3.2.1 and section 3.2.2 are crucial in determining whether to use the current solution D(z) or the next best solution within the population of solutions (Z). As illustrated in figure 4 The main idea would be to search a highly diversified solution without increasing the overhead by finding an optimal solution during the process of iterations.

3.2.1 Improvement (Z)

The improvement function(F1) helps to take decisions during the process of decision making. The decisions include whether to change algorithm or not. This is being done by comparing the makespan obtained using (Hi) with the current makespan. If the makespan obtained in the (Hi) is greater than current makespan i.e. it does not improve after ni iterations, it results true. This operator returns false for the following conditions
1. If the solution of the selected algorithm(Hi) is not value-added after ni iterations
2. While maximum number of iteration is reached
3. When stop condition is triggered
Improvement function can be depicted as follows

$$F_1 = \begin{cases} false; & \text{current makespan is not improved after } \phi_{ni} \text{ iteration} \\ & \text{true; otherwise} \end{cases}$$

3.2.2 Detection of Diversity (Z)

Diversity detection function(F2) is also crucial and is used by analytical scheming to detect the need of shifting to the next algorithm Hi as described below to perform the scheduling of tasks. The diversity Z for the first solution H0 will be used as a threshold value (), i.e. = D(Z0)and compare the values with the current solution. If the current value is greater than , it returns true, else it returns false. Diversity Detection Function can be depicted as follows

$$F_2 = \begin{cases} true\ ; D(Z_0 > \omega) \\ false\ ; otherwise \end{cases}$$

3.2.3 Perturbation Process (Z)

Perturbation denoted as (F3) below, is used to perturb the solutions obtained in the selection Hi , before they are selected onto the next (Hi)level algorithm for the process of effective scheduling. It works by returning false when both the improvement function F1 discussed in section 3.2.1 and diversity detection function(F2) discussed in section 3.2.2 returns true, else it returns true. If it is triggered by the value true, it does the following procedure:

1. It assigns Lowest Priority to the current algorithm
2. It selects the algorithm with the highest priority for the next schedule.

$$F_3 = \begin{cases} false\ , F_1 = true\ and\ F_2 = true \\ true\ , \text{otherwise} \end{cases} \quad *$$

After grouping the VMs into number of clusters, the Q-Aware task scheduling algorithm is applied to allocate the task to the particular cluster that computes the makespan of tasks.

In figure 5, the algorithm makes use of the improvement(F1) and diversity detection function(F2) to find the solution during the process of convergence and based on the perturbation function(F3) it assigns the priority. The highest prioritized algorithm in the pool of Hi is repeated until the termination criteria is met followed by the next one in the pool.

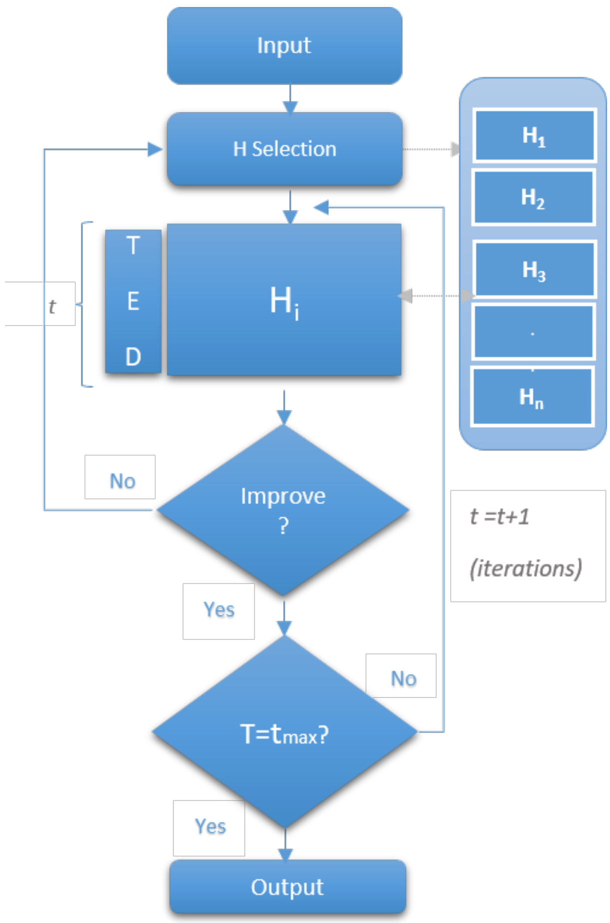


Figure 4: Q-Aware Heuristics Method

3.3 Methodology for Third Stage: - Computing the cost

Productive estimation and basic First in First Out policy is used to calculate the cost of the resources being utilized. The process is well described in the below pseudo code.

Procedure:
While (TaskList is not null)
{
Takes the first Task Ti (first task that allocated to vm using FIFO)
Obtain the utilization values of resources of each task(cpu,memory and bandwidth)
Compute the overall price
Remove task Ti
}

The pseudo code for the above process can be depicted as below

Step 1: *Input the Task and VM details*

Step 2: *Initialize the population* $Z = \{z_1, z_2, \ldots, z_n\}$. *(Based on the Task)*

Step 3*: Is it the first run? (First Execution?)*

Step 4: *If Yes =>*

Step 4.1: *Randomly select an algorithm from the candidate pool (ABC, SA, GA, IWO and PSO)*

Step 5: *If No =>*

Step 5.1:*Select the algorithm with highest priority for scheduling.*

Step 6: *While the termination criterion is not met.*

Step 6.1:*Update the population of solutions Z by using the selected algorithm.*

Step 7: *Compute F1 = Improvement Function (Z).*

Step 8: *Compute F2= Diversity Detection Function (Z).*

Step 9: *Compute F3=Pertub Function (Z)*

Step 10: *Compute If Pertub (Z) is true*

Step 10.1*:Assign "Lowest Priority" to the current algorithm.*

Step 10.2: *Select another algorithm with the "Highest Priority".*

Step 11: *Repeat the scheduling process until maximum number or iteration is reached.*

Step 12: *End*

Figure 5: Pseudocode for Analytical Scheming

4 Implementation

As detailed in the previous sections about the technical aspects of the Q-aware algorithm, below is the skeleton view of the methodology. It comprises of the stages discussed in Section 3 Q-aware technique method shall provide a different approach to endorse the research goal. Consequently, following study provides the details of system Q-aware task scheduling which would be applied across the cloud environment. However, before understanding the working of the system it is essential to understand the internal architecture of the system.

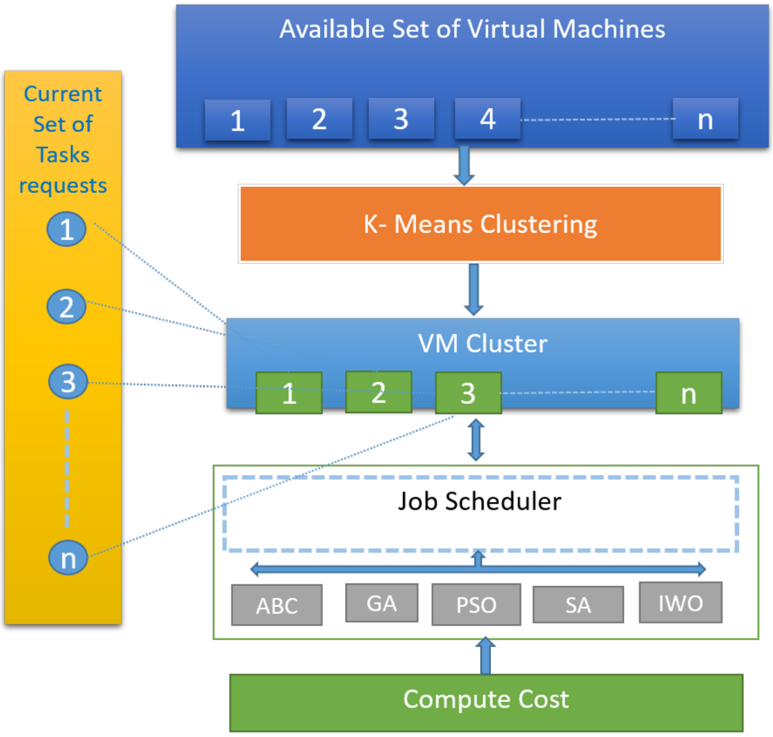


Figure 6: System architecture of Q-Aware Scheduling

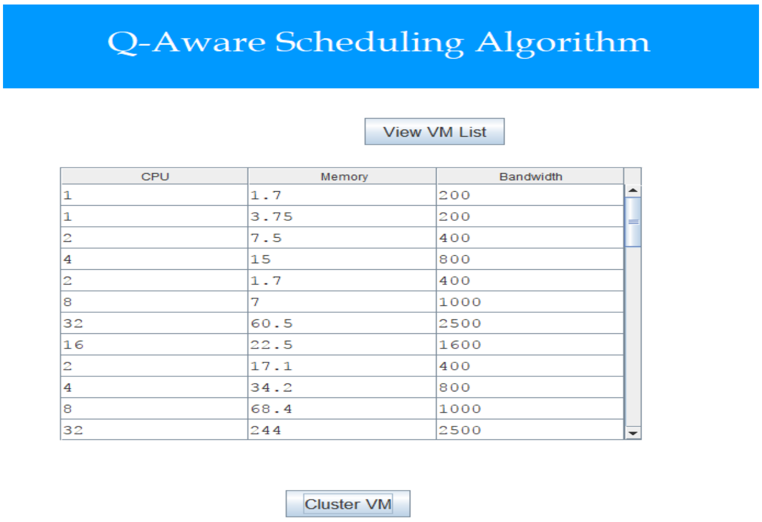
4.1 Tools

The simulation environment is created on java platform (jdk-8u144-windows-x64, jre-8u144-windows-x64). In-order to simulate the datacenter environment, cloudsim 3.0.2 framework has been integrated with the netbeans IDE 8 and JDK. Cloudsim is a toolkit introduced by (Calheiros et al.; 2011)for the purpose of modelling and simulating lot of entities such as Virtualized hosts, tasks, network policies, and message queuing in the cloud environments. The reason for choosing this simulator is, for its easy integration, and it has proved to be efficient and is open source.

5 Evaluation

5.0.1 Dataset

The dataset has been taken from the Amazon web services, which includes the resource information with the CPU, Memory, Bandwidth of various instances as shown in figure 7. It includes wide range of instances which have various combinations and includes wide range of choice of resources depending on the application type.



CPU	Memory	Bandwidth
1	1.7	200
1	3.75	200
2	7.5	400
4	15	800
2	1.7	400
8	7	1000
32	60.5	2500
16	22.5	1600
2	17.1	400
4	34.2	800
8	68.4	1000
32	244	2500

Figure 7: List of Virtual Machines

The following dataset is taken as an input as shown in figure 4,where in real scenario it goes to the cloud provider. The virtual machines are grouped into clusters by using k-means clustering which was detailed in section 3.1. This is depicted in figure 8.

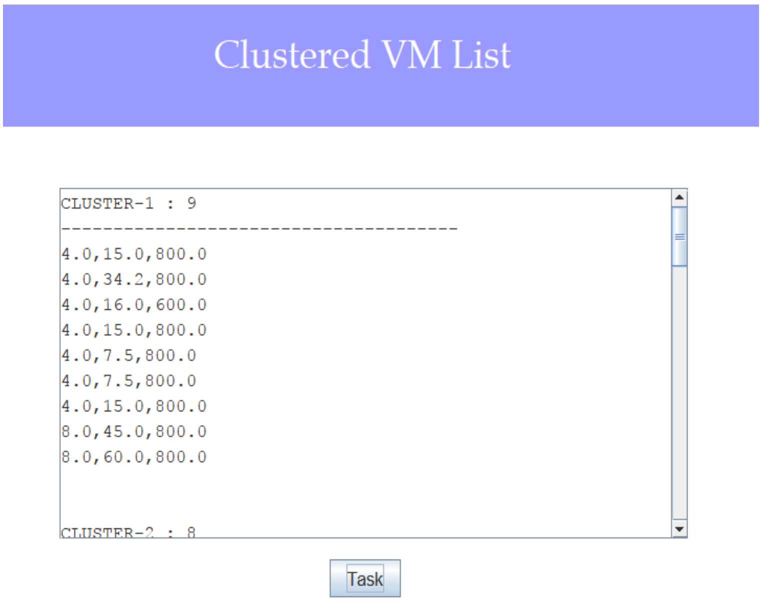
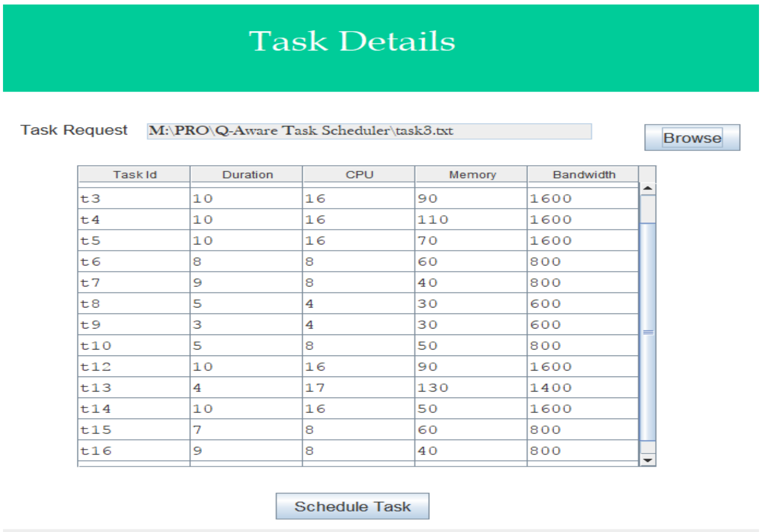


Figure 8: Clusterd VM's

Once the clusters are formed in the order in which the tasks are to be scheduled on the clusters, the tasks information will be considered as an input which consists of task parameters such as taskid and duration i.e. the bursttime as depicted in figure 9



Task Id	Duration	CPU	Memory	Bandwidth
t3	10	16	90	1600
t4	10	16	110	1600
t5	10	16	70	1600
t6	8	8	60	800
t7	9	8	40	800
t8	5	4	30	600
t9	3	4	30	600
t10	5	8	50	800
t12	10	16	90	1600
t13	4	17	130	1400
t14	10	16	50	1600
t15	7	8	60	800
t16	9	8	40	800

Figure 9: Task Details

5.1 Output

Once all the information is provided, it is processed and the Q-aware works by picking an algorithm using the random technique and applies the analytical scheming on the further iterations by calculating the diversity factor. The heuristic algorithms work by selecting and assigning tasks to VMs. The time for the last job executed is calculated and is computed as makespan value as shown below in figure 9. The cost is calculated and is generated based on the resources utilized.

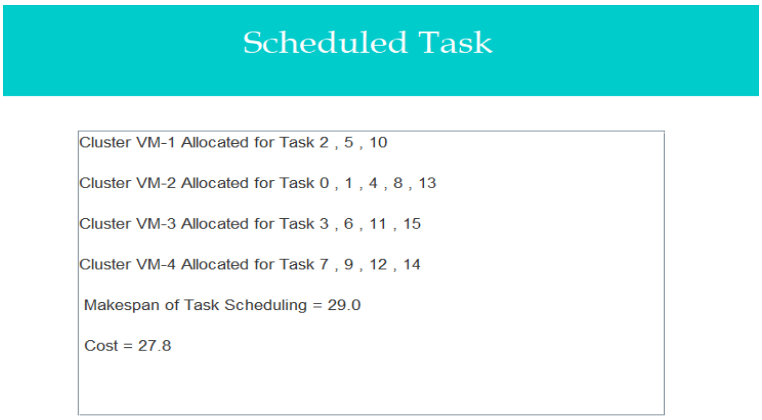


Figure 10: Output

6 Conclusion and Future Work

This research is intended to leverage the strengths of various heuristics which have been inspired from the nature, such as genetic algorithm, particle swarm optimization, simulated annealing, ant colony optimization and Invasive weed optimization. Various researchers have tested the use of these heuristics in various fields and also in the field of computing and has concluded that these work towards the optimization problems in cloud scheduling. The ideology behind this project is to present an alternative method to hybrid-heuristics which merges more than one algorithm to obtain an optimal solution in a single iteration for the scheduling problem in the cloud. This new method uses five heuristic algorithms and work towards finding better solutions thus improving the existing solution during the iteration process by analytical scheming technique and determining the makespan. The amalgamation process of using the discussed powerful heuristics can recompense the deep-down feeble points within each of the meta-heuristic algorithm. The solution has been implemented in java and simulated using the CloudSim and has been successfully tested on NetBeans. This research can be further analysed in detail and compared with the industry used techniques and algorithms as a future work.

Acknowledgement

I wish to express my sincere gratitude to Mr. David Hamill for providing me constant support and encouragement throughout my thesis. It was my privilege to work with Mr. Hamill, who helped me to overcome the pressure by being friendly in nature. This has enriched my ability to work towards the research which I have initially proposed and complete it successfully within the timelines. I would also like to thank my friends for supporting me during the tough times. I am grateful to my parents for being my strength at all times. Importantly, I would like to thank National college of Ireland for helping me to accomplish my dream.

References

Abdullah, M. and Othman, M. (2014). Simulated annealing approach to cost-based multi-quality of service job scheduling in cloud computing enviroment, *American Journal of Applied Sciences* **11**(6): 872.

Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. and Buyya, R. (2011). Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Software: Practice and experience* **41**(1): 23–50.

Chrétienne, P. and Picouleau, C. (1995). Scheduling with communication delays: a survey, *Scheduling theory and its applications* pp. 65–90.

Elhady, G. F. and Tawfeek, M. A. (2015). A comparative study into swarm intelligence algorithms for dynamic tasks scheduling in cloud computing, *Intelligent Computing and Information Systems (ICICIS), 2015 IEEE Seventh International Conference on*, IEEE, pp. 362–369.

Fan, C., Zhang, T., Yang, Z. and Wang, L. (2015). A text clustering algorithm hybriding invasive weed optimization with k-means, *Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), 2015 IEEE 12th Intl Conf on*, IEEE, pp. 1333–1338.

Garey, M. R., Johnson, D. S. and Sethi, R. (1976). The complexity of flowshop and jobshop scheduling, *Mathematics of operations research* **1**(2): 117–129.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm, *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **28**(1): 100–108.

Kaur, K., Chhabra, A. and Singh, G. (2010). Heuristics based genetic algorithm for scheduling static tasks in homogeneous parallel system, *International Journal of Computer Science and Security (IJCSS)* **4**(2): 183–198.

Laha, D. and Chakraborty, U. K. (2009). A constructive heuristic for minimizing makespan in no-wait flow shop scheduling, *The International Journal of Advanced Manufacturing Technology* **41**(1): 97–109.

Le Dinh, L., Vo Ngoc, D. and Vasant, P. (2013). Artificial bee colony algorithm for solving optimal power flow problem, *The Scientific World Journal* **2013**.

Li, K., Li, S., Xu, Y. and Xie, Z. (2014). A dag task scheduling scheme on heterogeneous computing systems using invasive weed optimization algorithm, *Parallel Architectures, Algorithms and Programming (PAAP), 2014 Sixth International Symposium on*, IEEE, pp. 262–267.

Lin, F.-T. and Hsu, C.-C. (1990). Task assignment scheduling by simulated annealing, *Computer and Communication Systems, 1990. IEEE TENCON'90., 1990 IEEE Region 10 Conference on*, IEEE, pp. 279–283.

Liu, X. and Liu, J. (2016). A task scheduling based on simulated annealing algorithm in cloud computing, *International Journal of Hybrid Information Technology* **9**(6): 403–412.

Magoulès, F., Pan, J. and Teng, F. (2012). *Cloud computing: Data-intensive computing and scheduling*, CRC press.

Mandal, T. and Acharyya, S. (2015). Optimal task scheduling in cloud computing environment: Meta heuristic approaches, *Electrical Information and Communication Technology (EICT), 2015 2nd International Conference on*, IEEE, pp. 24–28.

Morton, T. and Pentico, D. W. (1993). *Heuristic scheduling systems: with applications to production systems and project management*, Vol. 3, John Wiley & Sons.

Nawaz, M., Enscore, E. E. and Ham, I. (1983). A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem, *Omega* **11**(1): 91–95.

Pandey, S., Wu, L., Guru, S. M. and Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments, *Advanced information networking and applications (AINA), 2010 24th IEEE international conference on*, IEEE, pp. 400–407.

Parsa, S. and Entezari-Maleki, R. (2009). Rasa: A new task scheduling algorithm in grid environment, *World Applied sciences journal* **7**(Special issue of Computer & IT): 152–160.

Pinedo, M. (2012). *Scheduling*, Springer.

Ramezani, F., Lu, J. and Hussain, F. K. (2014). Task-based system load balancing in cloud computing using particle swarm optimization, *International journal of parallel programming* **42**(5): 739–754.

Said, G. A. E.-N. A. (2016). Science publishing group.

Shidik, G. F., Sulistyowati, N. S. and Tirta, M. B. (2016). Evaluation of cluster k-means as vm selection in dynamic vm consolidation, *Communications (APCC), 2016 22nd Asia-Pacific Conference on*, IEEE, pp. 124–128.

Singh, S. and Kalra, M. (2014). Scheduling of independent tasks in cloud computing using modified genetic algorithm, *Computational Intelligence and Communication Networks (CICN), 2014 International Conference on*, IEEE, pp. 565–569.

Wu, Z., Liu, X., Ni, Z., Yuan, D. and Yang, Y. (2013). A market-oriented hierarchical scheduling strategy in cloud workflow systems, *The Journal of Supercomputing* pp. 1–38.

Xiao, R. and Yu, J. (2008). A newly self-adaptive strategy for the pso, *Natural Computation, 2008. ICNC'08. Fourth International Conference on*, Vol. 1, IEEE, pp. 396–400.

Yang, X.-S. (2010). *Nature-inspired metaheuristic algorithms*, Luniver press.

Yu, J. and Buyya, R. (2006). Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms, *Scientific Programming* **14**(3-4): 217–230.

Zuo, X., Zhang, G. and Tan, W. (2014). Self-adaptive learning pso-based deadline constrained task scheduling for hybrid iaas cloud, *IEEE Transactions on Automation Science and Engineering* **11**(2): 564–573.