Multi-cloud distributed task scheduling for cost and QoS optimization

Cloud computing offers the ability to provision resources as required on a per-task basis. Due to the growth of Cloud computing, there are many challenges faced today. One of these challenges is scheduling tasks to minimize cost in respect to runtime. Task scheduling is considered an NP hard problem with $O(m^n)$ complexity when running $n\ tasks\ on\ m\ resources$.

In a multi-cloud scenario we still want to select the best resource to run a task on. However due to cost models that scale independently and the fact that measured performance that can vary from reported performance, this problem isn't as simple as it sounds. Optimally we want our algorithm to select the resource that would run at the highest performance to cost ratio.

The algorithm I am proposing is a modified scheduling algorithm that has been adapted to factor in the cost of a VM and it's actual performance versus the performance as reported by the cloud provider. The first step is to initialize a queue of all the tasks we want to run. We then start by building a cost to performance ratio for each machine that we want to run the current task on. After some variable initialization, we identify the highest performance to cost ratio. This will not specifically select the fastest or the cheapest machine, but the machine that will run the fastest for the lowest cost. We then identify the cloud provider that our selected VM is running on and attempt to schedule our task to that Virtual Machine. In the case that the scheduling fails, we attempt to reschedule.

A large selection of algorithms to schedule tasks currently exist. I will be comparing my proposed algorithm to these algorithms to test its performance.

I will experiment with this algorithm first on CloudSim to get a working model and then scale it to run on the three main cloud providers, amazon, google, and microsoft.