Future Generation Computer Systems 41 (2014) 65–78

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Comparison of a cost-effective virtual cloud cluster with an existing campus cluster

A. Stephen McGough [a,*,1], Matthew Forshaw [b], Clive Gerrard [b], Stuart Wheater [c], Ben Allen [b], Paul Robinson [d]

[a] Durham University, Durham, DH1 3LE, United Kingdom
[b] Newcastle University, Newcastle upon Tyne, NE1 7RU, United Kingdom
[c] Arjuna Technologies Ltd, Newcastle upon Tyne, United Kingdom
[d] Red Hat Inc., Newcastle upon Tyne, United Kingdom

HIGHLIGHTS

- We simulate a Cluster computer running on the Cloud.
- We compare this to a Cluster running on Campus.
- We show that the cost of running a Cloud Cluster is inversely related to the make-span of work on the cluster.
- We compare the cost of using Cloud vs local clusters.

ARTICLE INFO

Article history:
Received 1 July 2012
Received in revised form
4 June 2014
Accepted 7 July 2014
Available online 12 July 2014

Keywords:
Cloud
Economic
Simulation

ABSTRACT

The Cloud provides impartial access to computer services on a pay-per-use basis, a fact that has encouraged many researchers to adopt the Cloud for the processing of large computational jobs and data storage. It has been used in the past for single research endeavours or as a mechanism for coping with excessive load on conventional computational resources (clusters). In this paper we investigate, through the use of simulation, the applicability of running an entire computer cluster on the Cloud. We investigate a number of policy decisions which can be applied to such a virtual cluster to reduce the running cost and the effect these policies have on the users of the cluster. We go further to compare the cost of running the same workload both on the Cloud and on an existing campus cluster of non-dedicated resources.

1. Introduction

Cloud Computing [1] provides a new model for computational processing and data storage removing many of the access barriers to large-scale computing (often referred to as High Throughput Computing (HTC)) by eliminating the need for capital expenditure on large private infrastructures. Instead users pay only for the computational power or data space they use – more than they could afford to buy though enough to meet their immediate needs from an apparently infinite (henceforth we just say infinite) pool of resources – transferring capital expenditure to operational cost. This allows the user to work in-spite of local resource availability. Large collections of resources can be provisioned in a short period of time, quicker than many organisations can offer, for a relatively small operational outlay, and at a fraction of the capital cost. This approach has been used in scenarios with significant temporal variation in requirements, alternating between periods of little (or no) activity to periods of high activity and jobs which require low data transfers, to mitigate the data transfer times and costs.

Traditionally many organisations such as universities or companies have provided HTC through a dedicated centralised cluster of computers, where capital expenditure is committed to a fixed number of computational resources and data storage. This has the advantage of economies of scale as most users of the HTC facility will not need full access to the facility at the same time. The size of such a facility is dominated by two factors: the anticipated load on the cluster and the available budget. The aim is to provision enough

* Corresponding author. Tel.: +44 01913341749.
E-mail addresses: stephen.mcgough@durham.ac.uk (A.S. McGough), m.j.forshaw@ncl.ac.uk (M. Forshaw), stuart.wheater@arjuna.com (S. Wheater), ben.allen@ncl.ac.uk (B. Allen), paul.robinson@redhat.com (P. Robinson).
[1] Work carried out whilst based at Newcastle University.

http://dx.doi.org/10.1016/j.future.2014.07.002
0167-739X/

resources to deal with all but the exceptional load scenarios placed on the resources. The exceptional load is dealt with either by failing to achieve the required level of Quality of Service or by outsourcing work, for example to a Cloud provider [2,3]. Excess jobs which cannot be handled on local resources are sent to a (public) Cloud for execution—thus allowing the owners to temporarily increase the size of their own cluster.

Here we explore an alternative use case—moving the entire cluster onto the Cloud. We investigate a number of polices which can be applied over an existing HTC management service for determining the number of Cloud instances which should make up the virtual Cloud cluster. We further investigate whether there are advantages in all HTC users within an organisation sharing resources to help reduce costs.

We evaluate our approach through the use of two metrics: the financial cost of using the Cloud (based on the number of hours consumed along with data transfer charges) and the impact on job overheads. We define overhead as the difference between the total time a job spends within the system and the actual execution time for the job, a more formal definition for overhead is given in Section 3. The overheads include both the time to upload and download data to the Cloud along with any other delays incurred from using the Cloud. This data transfer also has implications for the cost of using the Cloud as most Cloud providers charge for data transfers.

We use a trace-driven simulation [4], using trace logs from the HTCondor [5] (formerly called Condor) desktop cluster based at Newcastle University [6,7], to evaluate the effectiveness of our approach. In order to evaluate our policies more thoroughly we have generated a number of synthetic trace logs based on increasing the number of users submitting work into the HTC cluster. These synthetic loads represent approximately one to five times the workload from our real logs, allowing for evaluation of our policies under greater workload. Using just the submission times for jobs to the cluster, their execution times and the data ingress/egress volumes allows us to submit jobs into the simulated Cloud cluster where jobs will either receive service immediately, if virtual computational instances (referred to here as *instances*) are idle, or enter a queue awaiting execution. A Policy can then be enacted to determine if (and when) a new Cloud instance should be started or unused instances terminated. As the main focus of this paper is a comparative evaluation of a number of policies we do not concern ourselves with how users would have changed their usage patterns on the Cloud, instead using these trace-logs for comparison only—real deployment would almost certainly alter usage patterns. We acknowledge here that the execution times of workload on the Cloud would vary in comparison with the execution times observed on our local desktop cluster. However, our aim here is to compare the different polices for optimising our use of the Cloud hence we do not take this variation into account. Further, Gillam et al. [8] observe over 100% variation in performance of Cloud instances advertised as being the same thus making any scaling process highly inaccurate.

An alternative approach used by many organisations is to make use of their existing computational resources for a secondary purpose, thus exploiting the idle time on these computers for HTC workload. However, as computers are used by the HTC system speculatively, computational work may need to be sacrificed in the case when the user requires his/her computer. This has the advantage that although these resources are no longer dedicated for the processing of computational workload it does allow the organisation to make use of a large collection of computers for little (if any) capital expense. This form of desktop cluster, often referred to as a desktop grid, can therefore be seen as an alternative to using the Cloud.

We have previously shown that ∼120 MWh of energy was consumed in 2010 to power the Newcastle HTCondor desktop cluster [7]. This being made up from ∼43 MWh from good HTCondor work which completed and ∼77 MWh from bad HTCondor work which did not complete. In order to fairly compare the use of a desktop cluster with the Cloud we additionally factor in the other charges which would be required for running this service, those of staff costs, carbon emissions and dedicated server costs.

We see our key contributions from this work as being:

- an evaluation of the feasibility and cost of moving an entire HTC cluster into the Cloud based around real trace logs and trace logs generated from synthetic users;
- an evaluation of a number of policies for minimising the cost of using the Cloud for HTC workload along with the effect that this will have on the overheads observed by the user;
- a comparison of the cost implications of running large HTC workloads on a Cloud as opposed to using a non-dedicated HTC desktop cluster.

The rest of this paper is set out as follows. Section 2 discusses related research to the work we propose. In Section 3 we describe in more detail the cluster we are modelling. We present a number of policies for optimising the cost for using the Cloud in Section 4 along with the perceived benefits of these policies. The simulation environment is described in Section 5 with the simulation results being presented in Section 6 where we also compare the cost of using the Cloud to the cost, in terms of both energy and hardware, for using the campus based cluster at Newcastle when executing the same workload. Finally our conclusions are presented in Section 7.

2. Related work

There is currently great interest in Cloud Computing [1]. This has led to a number of investigations into the applicability of the Cloud as a tool for aiding users in their work. A number of simulation approaches to model the benefits of Cloud computing have been performed. Deelman [9] evaluated the cost of using Amazon's Elastic Compute Cloud (EC2) [10] and Amazon's Simple Storage Service (S3) [11] to service the requirements of a single scientific application. Here we seek to service the requirements of multiple users and multiple applications.

De Assuncao [2] proposed the use of Cloud computing to extend existing clusters to deal with the exceptional load. This work was further extended by Mattess [12] by proposing the use of Amazon Spot Instances, supply-and-demand driven pricing of instances, to further reduce the cost of Cloud Bursting. Our approach differs from these in the sense that we seek to deploy our entire cluster to the Cloud. The approach of using Spot Instances, however, could easily be included in our approach and would allow for the same cost reduction as proposed by Mattess. Van den Bossche et al. [13] uses Binary Integer Programming to select which workflows should be burst to the Cloud. This approach is computationally expensive to determine the optimal approach and does not address the issue of when to terminate instances. To address the computational expense Van de Bossche et al. extend their work by developing scheduling algorithms for bag-of-tasks applications in hybrid cloud environments [14]. It may be naively assumed that our approach here is no more than the degenerative case with no local resources. However, these papers discuss when Cloud resources should be brought in, whilst our work discusses how to best manage the starting/termination of instances. These two approaches can therefore be seen as complementary.

Marshall [15] proposes policies for how to extend the number of Cloud instances to use along with simulations of a small number of short running synthetic jobs to evaluate overhead times. Here we