A federated CometCloud infrastructure to support resource sharing

Ioan Petri[1], Tom Beach[1], Mengsong Zou[2], Javier Diaz-Montes[2], Omer Rana[3] and Manish Parashar[2]
[1]School of Engineering, Cardiff University, UK
[2] Rutgers Discovery Informatics Institute, Rutgers University, USA
[3] School of Computer Science & Informatics, Cardiff University, UK
contact author: petrii@cardiff.ac.uk

***Abstract*—One of the key benefits of Cloud systems is their ability to provide elastic, on-demand (seemingly infinite) computing capability and performance for supporting service delivery. With the resource availability in single data centres proving to be limited, the option of obtaining extra-resources from a collection of Cloud providers has appeared as an efficacious solution. The ability to utilize resources from multiple Cloud providers is also often mentioned as a means to: (i) prevent vendor lock in, (ii) to enable in house capacity to be combined with an external Cloud provider; (iii) combine specialist capability from multiple Cloud vendors (especially when one vendor does not offer such capability or where such capability may come at a higher price). Such *federation* of Cloud systems can therefore overcome a limit in capacity and enable providers to dynamically increase the availability of resources to serve requests. CometCloud provides an overlay that enables multiple types of Cloud systems (both public and private) to be federated through the use of specialist gateways. We describe how two physical sites, in the UK and the US, can be federated in a seamless way using this system.**

Keywords*—*Cloud Computing, Cloud Federation, CometCloud, Tuple-Space, Task Outsourcing.

I. INTRODUCTION

With the emergence of federation in cloud systems it has become possible to connect local infrastructure providers to a global marketplace where participants can transact (buy and sell) capacity on demand. The mechanism of cloud federation can bring substantial benefits for service providers by offering facilities for accessing global services instead of increasing costs associated with building new infrastructure. More importantly, organisations with spare capacity in the data centre are now provided with a simple way to monetize that capacity by submitting it to the marketplace for other providers to buy, creating an additional source of revenue [1]. Federation in cloud systems has led to a real *democratisation* of cloud markets – enabling businesses to make use of a variety of different cloud providers in different geographic areas [5]. When two or more sites come together, it is important to identify not only the submitted workflow from each site but also the cost of outsourcing additional resources, the revenue obtained from outsourcing tasks or the cost of maintaining a reasonable level of utilisation. Identifying a set of such parameters is a challenging task due to the variability in the parameters of a federated environment (such as number of resources allocated to local vs. remote jobs, how many jobs to outsource to another site, the time interval over which access to remote jobs should be allowed, etc) and the fluctuation of resource demand. Depending on the value of such parameters, a site manager must decide whether to outsource resources, compute tasks locally or reject remote task requests [2]. CometCloud [3] has been demonstrated to integrate both public clouds (based on Amazon EC2/S3) and specialist high performance computing environments (such as TeraGrid) [4]. In this work, we develop two different federation models (i) CometCloud Federation and (ii) aggregated CometCloud Federation where the tasks can be processed exclusively "in house" – using local capabilities or outsourced to remote federated locations. We approach the problem of federation by implementing a real federation Cloud system based on CometCloud facilitating task processing by an "aggregated tuple-space mechanism". This mechanism referred to as "CometSpace" is a distributed, Linda-like tuple space implemented using a Peer-2-Peer overlay network. In this way, a virtual shared space for storing data and computational capability can be implemented by aggregating capacity across a number of different compute resources. CometCloud therefore provides a scalable backend deployment platform that can combine resources of different types across a number of different providers dynamically.

II. SYSTEM MODEL

In this section we present the two federation models that we have developed by using CometCloud in a federated context. Although we report on two sites, the approach outlined in this work can scale to multiple sites based on the use of the Comet overlay. CometCloud is an autonomic computing engine based on the Comet decentralized coordination substrate, and supports highly heterogeneous and dynamic cloud/grid/HPC infrastructures, enabling the integration of public/private clouds and autonomic cloudbursts, i.e., dynamic scale-out to clouds to address extreme requirements such as heterogeneous and dynamics workloads, and spikes in demands.

A. CometCloud Federation Setup

For the CometCloud federation we consider that a site has a number of available workers and a master that receives requests: (i) locally – identifying tasks received from users at the same site; (ii) remotely – requests from remote users at other sites – via the use of a request handler.

As illustrated in Figure 1, the Master must decide how many tasks to accept based on a number of policies. We assume that there is one worker per compute/data access node. All workers are assumed to be the same – i.e. they can execute tasks on resources that are identical. The only differentiating factor, therefore, is the number of workers allocated to local vs.

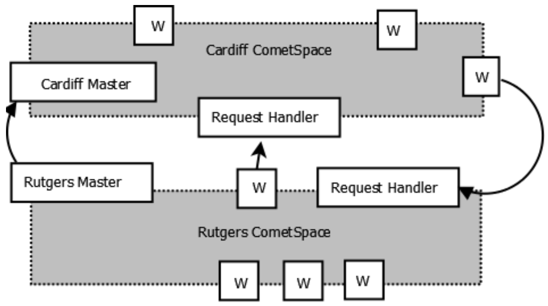

Fig. 1: Federation model

external/remote requests. When one site has a high workload and it is unable to process tasks from its local users within their deadlines it negotiate for the outsourcing of tasks to other remote sites. This could range from two cloud systems sharing workload (as in Figure 1) to a cloud outsourcing some of its workload to multiple other cloud systems. Conversely this ability allows systems with a lower workload to utilise spare capacity by accepting outsourced tasks from other cloud systems. Practically, this process of task exchange is undertaken by the master nodes of the two clouds negotiating how many tasks to be exchanged. Once this has been completed the master node on the receiving cloud informs its workers (using CometSpace) about the number of tasks it is taking from a remote site, and the connection details of the request handler from where the task is to be fetched. Subsequently, when a worker comes to execute a task from an external cloud system, it then connects to the request handler of the remote cloud to collect the task and any associated data.

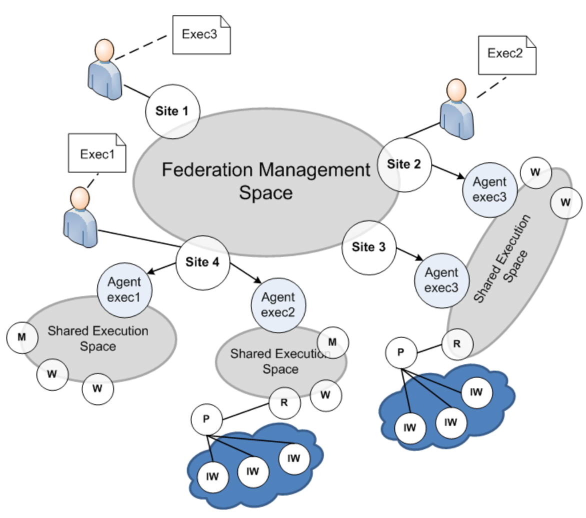

Fig. 2: Aggregated Federation model. Here (M) denotes a master, (W) is a worker, (IW) an isolated worker, (P) a proxy, and (R) is a request handler.

B. Aggregated CometCloud Federation Setup

The aggregated CometCloud federation model is designed to be dynamically updated as it is created in a collaborative way, where each site communicates with others to identify itself, negotiates the terms of interaction, discovers available resources, and advertises its own resources and capabilities. In this way, a federated management space is created at runtime and sites can join and leave at any point. This federation model does not have any centralized component and users can access the federation from any site, which increases the fault tolerance potential of the overall federation, see Figure 2. Another key benefit of this model is that since each site can differentiate itself based on the availability of specialist capability, it is possible to schedule tasks to take advantage of these capabilities. As shown in Figure 2, each shared execution space is controlled by an agent that creates such space and coordinates the resources for the execution of a particular set of tasks. An agent can act as a master node or delegate this to an alternative master (M) when some specific functionality is required. Moreover, agents deploy workers to actually compute the tasks. These workers can be in a trusted network and be part of the shared execution space, or they can be part of external resources such as a public cloud and therefore in a non-trusted network. The first type of workers are called secure workers (W) and can pull tasks directly from the space. Meanwhile, the second type of workers are called isolated workers (IW) and cannot interact directly with the shared space. Instead, they have to interact with a proxy (P) and a request handler (R) to be able to pull tasks from the space.

III. Conclusions

Federated cloud systems provide the means for individual, possibly competing, Clouds to cooperate so as to optimize costs and cope with variations of demand. In a community of cloud providers federation can also enable the process of trading resources, thereby greatly encouraging the emergence of federated markets. In this paper, we have investigated the problem of cloud federation by devising a real federation framework based on CometCloud. The framework provides the advantage of an aggregated tuple-space and enables mechanisms to allocate resources based on a number of different policies. In our CometCloud based system, providers can optimise a number of performance parameters such as delay, utilisation, cost and reputation by using the mechanism of outsourcing tasks. The main advantage of aggregated CometCloud federation is the use of a centralized mechanism which enables users to access any site involved in the federation.

References

[1] I. Goiri, J. Guitart, and J. Torres. Economic model of a cloud provider operating in a federated cloud. Information Systems Frontiers , pages 1-17, 2011

[2] Adel Nadjaran Toosi, Rodrigo N. Calheiros, Ruppa K. Thulasiram, and Rajkumar Buyya, Resource Provisioning Policies to Increase IaaS Provider's Profit in a Federated Cloud Environment. In Proceedings of the 2011 IEEE International Conference on High Performance Computing and Communications (HPCC '11). IEEE Computer Society, Washington, DC, USA, 279-287.

[3] CometCloud Project. http://www.cometcloud.org/. Last accessed: August 2013.

[4] Hyunjoo Kim, Yaakoub el-Khamra, Shantenu Jha, and Manish Parashar, An Autonomic Approach to Integrated HPC Grid and Cloud Usage, the 5th IEEE International Conference on e-Science, Oxford, UK, Dec. 2009.

[5] E. Gomes, Q. Vo, and R. Kowalczyk, Pure exchange markets for resource sharing in federated Clouds, Concurrency and Computation: Practice and Experience, vol. 23, 2011.