PROCEEDINGS OF SCIENCE

Context-aware cloud computing for HEP

Randall J. Sobie[*]
Department of Physics and Astronomy, University of Victoria, Victoria, British Columbia, Canada V8W 2Y2
E-mail: rsobie@uvic.ca

The use of cloud computing is increasing in the field of high-energy physics (HEP). Both dedicated and opportunistic cloud resources are delivering a growing fraction of the requirements of large projects. The use of context-aware designs, such as those used in the mobile communication industry, can improve our distributed systems and make them more responsive to changing demands and dynamic resources. We show how context-aware service have been implemented and their impact on the system. We describe our plans to add new context-aware capabilities in the future.

International Symposium on Grids and Clouds 2016
13-18 March 2016
Academia Sinica, Taipei, Taiwan

[*] Speaker.

1. Introduction

Cloud computing is having a significant impact in the field of high-energy physics (HEP) [1, 2, 3, 4]. One can attribute this, in part, to the widespread adoption and investment in cloud computing by the commercial world. This large investment is leading to new and exciting developments in cloud software and services. One of the interesting areas is the use of context-awareness in distributed systems. In particular, the mobile communications industry is using context-aware systems to provide a tailored or customized response to users based on information such as their location, time, environment and previous use.

We have been using distributed cloud centres for HEP workloads [5]. Many of the clouds we use are opportunistic resources, hosted by research institutions that are not involved in HEP research or commercial clouds. These clouds do not have the usual HEP services, software caches or data repositories. As a result, the virtual machines and application jobs need to find these resources from external sources. One can preselect remote sites with the required services but often the choice is not optimal or the site may be offline. Instead we want a dynamic and flexible way to set up clouds, configure virtual machines, retrieve software and data. We argue that adopting a design that utilizes context-aware systems results in a more robust and efficient distributed cloud system.

In this paper, we briefly describe the basic concepts of context-aware computing and the design of our distributed cloud computing system. We show how context-aware service have been implemented and their impact on the system. We describe our plans to add new context-aware capabilities in the future.

2. Context-aware systems

A system is considered to be context-aware if it uses context to generate an action that provides information or services to the user [6]. The key elements of a context-aware computing system are shown in fig. 1 and summarized below. Further detail can be found in ref. [6, 7].

Entities are persons, places or objects that are considered relevant to the interaction between a user and an application;

Context is any information that can be used to characterize the situation of an entity;

Context-broker is a mediator service between entities providing the exchange of information between the entities;

Context-aggregator is a service that stores context for future use by the context-broker;

Action (context-triggered action) is an interaction between entities based on context provided by the context-broker.

A context-aware computing system has the ability to act on information about its environment such as location, time, and other data. For example, consider the example of a bus stop timetable service. A user arrives at a bus stop; the service anticipates the needs of the user based on the context of the situation and automatically starts a timetable application on their mobile device [8].

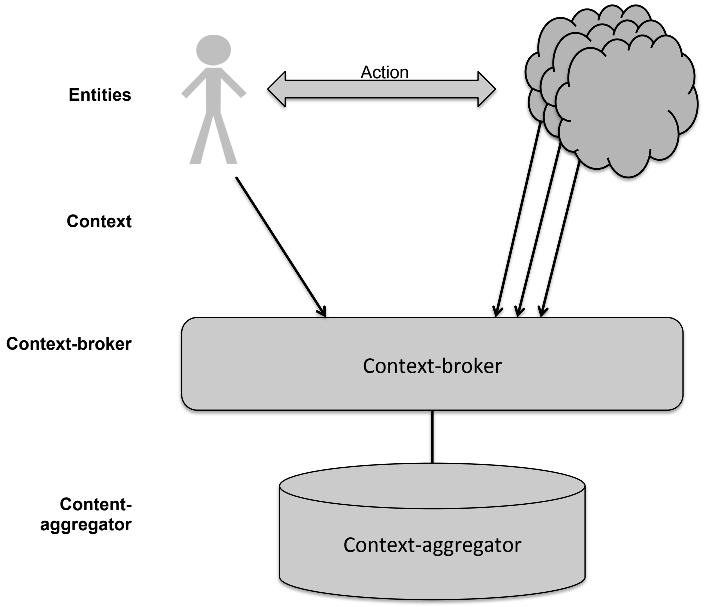

Figure 1: An outline of the features of a context-aware system. "Entities" are persons, places or objects that are considered relevant to the interaction between a user and an application; "context" is information that can be used to characterize the situation of an entity; "context-broker" is a mediator service between entities providing the exchange of information between the entities; "context-aggregator" is a service that stores context for future use by the context-broker; and "actions" occur between entities based on context provided by the context-broker.

3. Distributed cloud computing

A number of methods have been developed to utilize clouds by the HEP community. The design and implementation of the systems is usually driven by the requirement to integrate the cloud services into an existing local infrastructure. For example, the Belle II experiment at the KEK Laboratory in Japan uses four different methods to exploit clouds [4]. This paper will focus on our system that integrates private and opportunistic clouds into a single batch computing environment [5].

A high-level diagram of the system is presented in fig. 2. The system is designed for high-throughput batch jobs such as those used in HEP. The applications reconstruct raw analog and digital information into physical quantities such as particle momenta and energy. Other applications generate simulated data samples that are used to understand the detector performance and develop analysis algorithms. The applications usually require single-core but the number of multi-core applications (within a single machine) are increasing.

A user or project workload management system submits jobs to the batch job scheduler of the distributed cloud (we use HTCondor [9] as the batch scheduler). The VM-provisioning service reviews the status of the remote clouds and requests the remote cloud boot a user-customized virtual machine (VM) that meets the requirements of the user's job. When we developed our distributed cloud in 2010, there were no VM-provisioning services. As a result, we developed the CloudScheduler service to manage the VMs on remote clouds [10]. CloudScheduler makes requests to boot or delete VMs on the remote clouds, and also monitors their status.

The distributed cloud was designed to simultaneously exploit dedicated private clouds and opportunistic private and commercial clouds. We consider private dedicated clouds to be those operated on behalf of the HEP community; they have all the required HEP services, software caches and data repositories available locally. Private opportunistic clouds may also be funded with research dollars but have no HEP services. Commercial clouds are widely available from many companies.

The introduction of context-aware computing was originally conceived for dynamic and distributed systems (such as mobile communication). Similarly, context-aware services will help us make better and more effective use of remote clouds that have none of the HEP services or data.

Our system has been in production operation for many years using clouds in North America, Australia and Europe, completing many millions of jobs for ATLAS and Belle II [5]. The number and types of clouds has evolved due to the rapid development in cloud technologies; for example, the commonly used OpenStack software did not exist when CloudScheduler was developed.

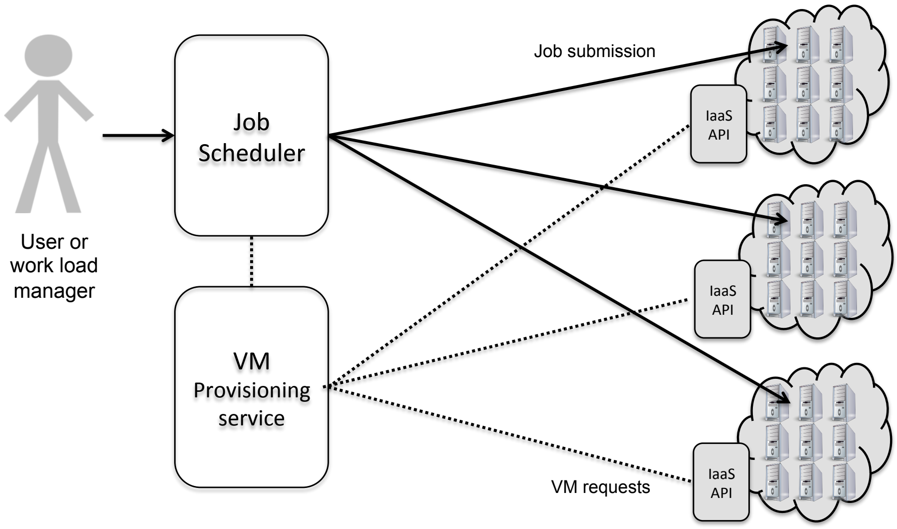

Figure 2: A distributed cloud computing system for HEP applications. The user or workload management system submits jobs to a job scheduler (such as HTCondor). A VM-provisioning system (such as Cloud-Scheduler) reviews the job queue and the availability of remote clouds. If a remote cloud has the resources and capacity for a user job, then the VM-provisioner request the cloud boot the user-customized VM. Once the VM is running, then it registers with the job scheduler. The job scheduler submits the application to the VM. One the job is finished and there are no more user jobs, then the VM-provisioner shuts down the VM.

4. Context-aware services

Many of the services in our distributed cloud can be considered to be context-aware. We briefly describe the CloudScheduler VM provisioning service and the Shoal software discovery service as two examples of contact-aware services. We are currently working on two additional services: a monitoring and alerting service, and a data federation service. We discuss the motivation and design of these two planned services.

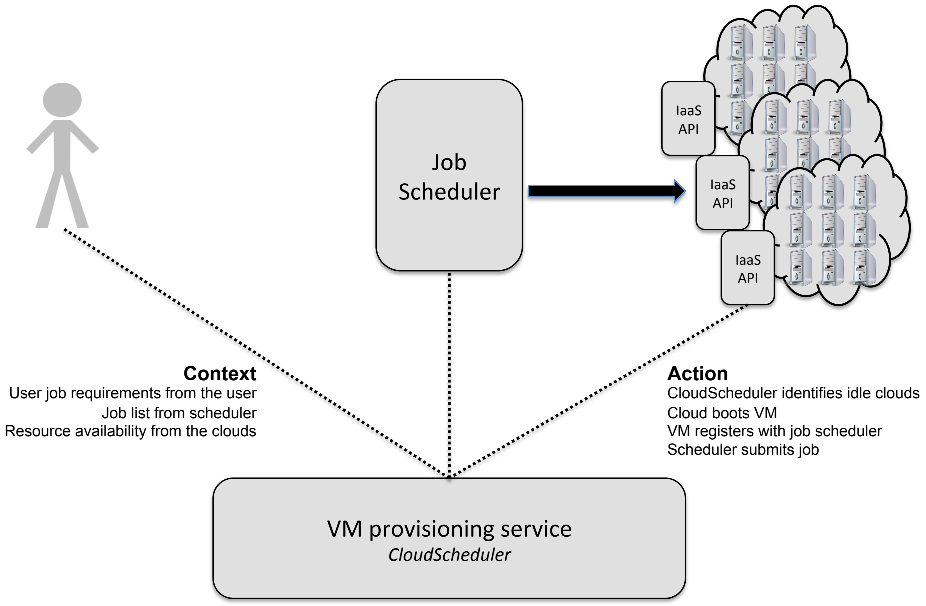

Figure 3: The CloudScheduler VM provisioning system is a context-aware service. The context from the user, the HTCondor queue, and the clouds are used to generate actions that boot a user-customized VM on a free cloud.

4.1 CloudScheduler - a context-aware VM provisioning service

CloudScheduler is a VM provisioning service that our group developed in 2009 [10]. CloudScheduler follows a context-aware design as shown in fig. 3, It uses context from the user (their job and VM requirements), the HTCondor job scheduler (the user specifications and job priority) and the remote clouds (availability of resources). As a context-aware broker, CloudScheduler uses the context to initiate actions; it identifies the next highest priority job in the HTCondor queue and requests that the selected cloud boot the VM required by the user. The VM registers with HTCondor pool and the user job is submitted to the resource. Once there are no further user jobs in HTCondor queue, CloudScheduler uses this new information to retire the virtual machine and free the resources for other users.

4.2 Shoal - a context-aware software delivery service

The HEP community uses the CernVM file system for storing operating system and application software [11]. The software is further cached in a set of globally distributed HTTP Squid web caches [12]. For traditional sites, the Squid cache is a local installation and specified in the site configuration. This model is not ideal for opportunistic resources, and as a consequence, we developed Shoal. Shoal is a context-aware HTTP Squid discovery service [13] as shown in fig. 4. Shoal collects context from the HTTP Squid caches (their location, activity and connectivity) and context from the virtual machines (location). Shoal links the VM with the optimal HTTP Squid cache providing fast access to both the operating system and application software.

4.3 Plans for new context-aware data services

At the moment, we use the opportunistic clouds for low-I/O applications such as the simulation of particle collisions. High I/O applications can be run on the dedicated private clouds if there is a local storage system with the required data. Our goal is to run high I/O applications on the opportunistic clouds and the challenge is to deliver the data to the virtual machines. Our plan is to utilize an HTTP data federation that unifies the HEP storage facilities. The VM would query a data federation service for the location of the data, the load on storage facility and the network utilization. The data would be copied to a local cache on the VM. Once the application has successfully completed, the data would be deleted. The HEP community has established HEP-specific data federation services but we favour a open-source service so that it can be used by researchers in other fields. Currently, we have been studying the Dynafed HTTP data federation service [14].

In addition, we are collecting monitoring data from all our services and clouds using standard (open-source) and customized services. We currently rely on staff to review the information and identify faults or issues, however, this is a tedious and error-prone task. Our goal is to have a context-aware service that collects the data and takes corrective action. For example, we want to disable a cloud if there are network or other issues; we want to identify VMs that are non-responsive or have been lost to the VM provisioning service; or identify application jobs that are stuck in a waiting state due to a network or cloud issue. The monitoring and alerting service will allow us to scale the level of our system to the ten thousand core level and beyond by significantly automating many features.

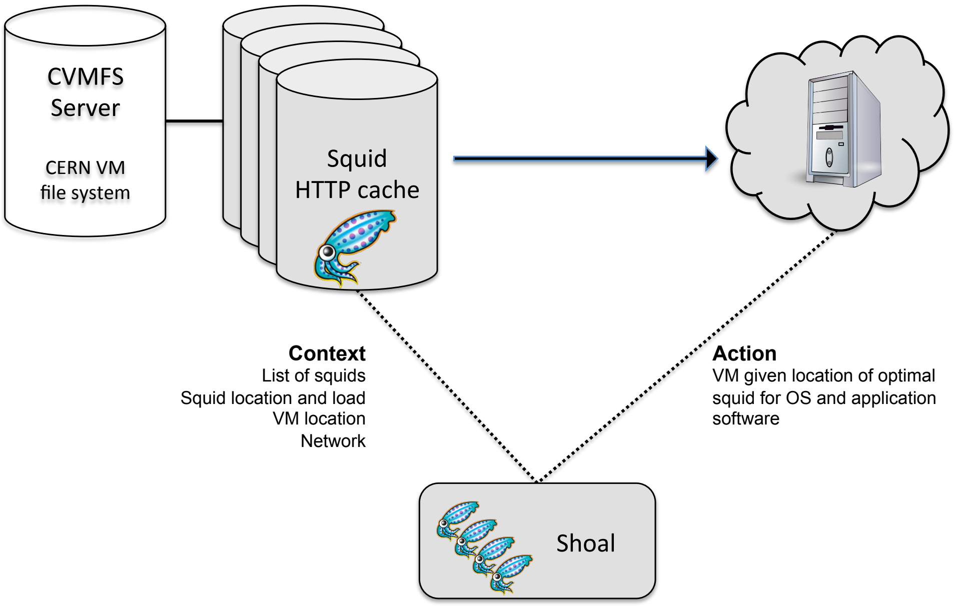

Figure 4: Shoal is a context-aware HTTP Squid discovery service. Shoal collects context from the HTTP Squid caches (their location, activity and connectivity) and context from the virtual machines (location). Shoal links the VM with the optimal HTTP Squid cache.

5. Summary

We have shown that context-aware services can significantly benefit the utilization of distributed and opportunistic cloud resources. Further, the addition of context-aware data federation services should enhance the capabilities and allow one to run data-intensive applications on all clouds.

References

[1] R. P. Taylor *et al. The Evolution of Cloud Computing in ATLAS,* J. Physics: Conf. Series 664 022038 (2015). http://stacks.iop.org/1742-6596/664/i=2/a=022038

[2] A. Andronis *et al. The Diverse use of Clouds by CMS,* J. Physics: Conf. Series 664 022012 (2015). http://stacks.iop.org/1742-6596/664/i=2/a=022012

[3] A. McNab *et al. LHCb experience with running jobs in virtual machines,* J. Physics: Conf. Series 664 022030 (2015). http://stacks.iop.org/1742-6596/664/i=2/a=022030

[4] R.J. Sobie *et al. Utilizing clouds for Belle II,* J. Physics: Conf. Series 664 022037 (2015). http://stacks.iop.org/1742-6596/664/i=2/a=022037

[5] R.J. Sobie. *Distributed cloud computing in high energy physics,* ACM SIGCOMM workshop on Distributed cloud computing (DCC '14). ACM, New York, 1. http://dx.doi.org/10.1145/2627566.2627578

[6] A.K. Dey. *Understanding and Using Context,* Personal Ubiquitous Comput. 5, 1 (2001), 4-7. http://dx.doi.org/10.1007/s007790170019.

[7] E. Badidi and L. Esmahi. *A Cloud-based Approach for Context Information Provisioning,* World of Computer Science and Information Technology Journal (WCSIT) Vol. 1, No. 3, 63-70, 2011.

[8] A. Schmidt. *The Encylopedia of Human Interaction*, Chapter 14 Context-aware computing. https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/context-aware-computing-context-awareness-context-aware-user-interfaces-and-implicit-interaction

[9] HTCondor: High Throughput Computing. https://research.cs.wisc.edu/htcondor

[10] P. Armstrong *et al. Cloud Scheduler: a resource manager for distributed compute clouds,* arXiv:1007.0050 2010.

[11] J. Blomer *et al. The Evolution of Global Scale Filesystems for Scientific Software Distribution,* Computing in Science and Engineering 17, 61, 2015. doi: 10.1109/MCSE.2015.111.

[12] Squid: Optimising Web Delivery. http:://www.squid-cache.org

[13] I. Gable *et al. Dynamic web cache publishing for IaaS clouds using Shoal,* J. Physics: Conf. Series 513 (2013) 032035. http://stacks.iop.org/1742-6596/513/i=3/a=032035

[14] F. Furano *et al. Towards an HTTP Ecosystem for HEP Data Access,* J. Physics: Conf. Series 513, 032034 (2014). http://stacks.iop.org/1742-6596/513/i=3/a=032034