COSCA: a component-based and scalable PaaS platform

Abstract – Frühjahrstreffen 2013 der Fachgruppe Betriebssysteme

Steffen Kächele and Franz J. Hauck

Institute of Distributed Systems
University of Ulm, Germany
{steffen.kaechele,franz.hauck}@uni-ulm.de

Introduction Platform as a service (PaaS) is a cloud service model that supports a very high abstraction to tenants. Application developers have to use a certain programming model, e.g. Web Servlets, offered by the platform. The application is uploaded to the platform and typically configured using cloud management tools. In contrast to infrastructure as a service (IaaS) tenants do not need to care about virtual machine images, software installations, configurations and updates. Especially scaling-out of an application on IaaS is rather complex. In this case the application has to be split into pieces that can be replicated on multiple virtual machines. Furthermore, the corresponding virtual machine images have to created, deployed and configured in order to achieve seamless scaling. Unlike IaaS, PaaS platforms typically offer (almost) entirely transparent scaling.

PaaS-based cloud computing allows cloud customers to concentrate on application development, maintenance and operation while the underlying platform is managed by a cloud provider. Yet, the programming model in early PaaS systems is typically limited, e.g. to Web Servlets, and often very restricted (no threads, no files, no sockets, ...) so that legacy applications can hardly be ported to the cloud. Additionally, most PaaS systems treat application as self-contained entities and do not provide a sophisticated support for applications that are composed from multiple subcomponents (e.g. during scale-out).

COSCA This talks presents COSCA, a PaaS platform with support for component-based applications [2]. We argue that the *component structure* of well-design software can be exploited for distribution and for horizontal scalability on the basis of replicating individual components [3]. Applications can be uploaded as in other platforms. Furthermore, the components within an applications can be reconfigured at runtime by installing, starting, stopping and updating of particular components. Thus, it becomes possible to adapt applications in real-time to changes in load, infrastructure and objectives.

Components are expected to interact by using services offered by other components. Our framework manages the distribution of components on cloud nodes and maps service interaction to an appropriate RPC-based communication in case that components reside on different nodes. A monitoring service observes both the resource consumption of components and the interaction between software modules. Thus, the platform is able to decide when it is the ideal time to scale in or out. Our prototype uses Java and a component model equivalent to OSGi. Nevertheless, we expect COSCA concepts to be applicable to other programming and component models.

COSCA can host applications of multiple tenants in a single Java virtual machine. The *runtime layer* isolates applications by virtualising resources as network sockets, files and CPU. This is achieved by injection of COSCA-specific JDK classes, separate class loaders and specific configurations of Java security managers.

Sockets are fully virtualized so that COSCA applications do not need to stick to HTTP but can implement arbitrary application protocols in the cloud. A network layer called COSCAnet provides full abstraction of replicated components, load balancing and scalability by a dedicated routing scheme. File access can be configured to transparently access a distributed file system, e.g. HDFS and XtreemFS.

Preliminary evaluations show that OSGi applications not designed for the cloud can be put into cloud and can be scaled at high load. Replicating individual components has its advantages when some components are not overloaded and have to be shared or need a large amount of resources (e.g. memory). Further evaluations show that COSCA has performance characteristics known from native setups that go beyond current cloud platforms.

In the talk, we will briefly address future work that will integrate fault-tolerance support into the platform [1]. This may affect the service platform as well as the network layer.

References

[1] Franz J. Hauck, Steffen Kächele, Jörg Domaschka, and Christian Spann. The COSCA PaaS platform: on the way to flexible and dependable cloud computing. In *Proc. of the 1st European Workshop on Dependable Cloud Computing*, EWDCC '12, pages 1:1–1:2, NY, 2012. ACM.

[2] Steffen Kächele, Jörg Domaschka, and Franz J. Hauck. COSCA: an easy-to-use component-based PaaS cloud system for common applications. In *Proc. of the First International Workshop on Cloud Computing Platforms*, CloudCP '11, pages 4:1–4:6, NY, 2011. ACM.

[3] Steffen Kächele and Franz J. Hauck. Component-based scalability for cloud applications. In *Proc. of the Thrid Workshop on Cloud Data and Platforms*, CloudDP '13, NY, 2013. ACM.