Matching INSPIRE Quality of Service Requirements with Hybrid Clouds

Authors:

Bastian Baranski[1], Theodor Foerster[2], Bastian Schäffer[3] and Kristof Lange[2]

[1] con terra GmbH, Münster, Germany

[2] Institute for Geoinformatics, University of Münster, Germany

[3] 52°North GmbH, Münster, Germany

Keywords:

Spatial Data Infrastructure, INSPIRE, Quality of Service, Cloud Computing, Hybrid Cloud

Corresponding Author:

Bastian Baranski

con terra GmbH

Martin-Luther-King-Weg 24

48155 Münster

Germany

Phone: +49 251 74745 - 0

Mail: b.baranski@conterra.de

Abstract

A lot of effort has been invested into Spatial Data Infrastructures (SDIs) during the last decade regarding interoperable standards for services and data. But still the scalability and performance of SDI services is reported to be crucial especially if they are accessed concurrently by a high number of users. Furthermore, laws and provisions such as the INSPIRE directive specify challenging requirements regarding the performance, availability and scalability of SDI services. This paper presents a Hybrid Cloud architecture for matching INSPIRE related Quality of Service (QoS) requirements without investing in rarely used hardware in advance by occupying external third-party resources on a pay-as-you-go basis. The presented Hybrid Cloud is a composition of a local IT-infrastructure (Private Cloud) and the computational resources of third-party vendors (Public Cloud). The local infrastructure is laid out to handle the average main load of a service and in lasting peak times additional resources of external providers are allocated and integrated on-demand into the local infrastructure to automatically provide sufficient service quality. A proof-of-concept implementation of the proposed Hybrid Cloud approach is evaluated and benchmarked with respect to INSPIRE related QoS requirements.

1. Introduction

A lot of effort has been invested into Spatial Data Infrastructures (SDIs) during the last decade regarding interoperable standards for services and data (Bernard, Craglia, Gould, & Kuhn, 2005; Bernard, Kanellopoulos, Annoni, & Smits, 2005; Masser, 2005; Onsrud, 2007; Scholten, Klamma, & Kiehle, 2006). But still the scalability and performance of SDI services is reported to be crucial especially if they are accessed concurrently by a high number of users or if huge amounts of data need to be processed (Scholten et al., 2006; Tu et al., 2004; Yang, Cao, & Evans, 2007; Zhang, Pennington, & Michener, 2007). Furthermore, laws and provisions such as the INSPIRE directive (EC, 2007) – a legal framework for establishing and operating a harmonized Spatial Information Infrastructure (SII) in the European Union (EU) - specify challenging requirements regarding the performance, availability and scalability of current SDI services.

Cloud Computing promises to fulfill such requirements. First experiments with Cloud Computing for geographic applications demonstrated that Cloud Computing is suitable for building on-demand GIS hosting platforms (for instance ESRI ArcGIS Server on Amazon EC2) as well as for realizing high-available and high-scalable geoprocessing services (Baranski, B., Schäffer, B. & Redweik, R. 2009). However, the application and the reliability of so-called Hybrid Clouds for SDIs have not been investigated yet. Finally, by allowing SDI service providers to up- and down-scale their infrastructure dynamically in the cloud without investing in rarely used hardware in advance, applying Cloud Computing to SDIs is also promising to operate SDIs cost-efficiently.

This paper presents a Hybrid Cloud architecture and implementation for matching INSPIRE related Quality of Service (QoS) requirements without investing in rarely used hardware in

advance by allocating external third-party resources on-demand and on a pay-as-you-go basis. The presented Hybrid Cloud architecture is a composition of two or more Cloud infrastructures that remain unique entities but are bound together to one organizational unit. The Private Cloud part of the Hybrid Cloud approach handles the average main load of an INSPIRE related use case within the service providers own infrastructure. Whereas in lasting peak times, additional resources of external Public Cloud providers are allocated and integrated on-demand into the service provider´s infrastructure to automatically match specific QoS requirements (e.g. as specified by the INSPIRE directive).

The proposed Hybrid Cloud architecture is implemented based on common as well as tailored Free and Open Source Software (FOSS). The approach is demonstrated and tested for the INSPIRE Coordinate Transformation Service (INSPIRE, 2009).

Section 2 presents required background information and the motivation for the conducted research. Section 3 describes the design and the implementation of the proposed Hybrid Cloud architecture. Section 4 presents a real world use case that is used in the same chapter as a foundation for a detailed performance and scalability evaluation of the proposed Hybrid Cloud approach. In Section 5 the research outcome is summarized and an overview about open issues and interesting topics for further research is given.

2. Research Context

This chapter presents background information and the motivation for the conducted research.

2.1 INSPIRE and Quality of Service (QoS)

The Infrastructure for Spatial Information in the European Community (INSPIRE) directive (EC, 2007) defines a legal framework that aims to create a common SDI for the European

Union (EU). To ensure that the SDIs of the EU member states are compatible, the INSPIRE directive describes so-called Implementing Rules (IRs) in the following five areas: Metadata, Data Specifications, Network Services, Data and Service Sharing, and Monitoring and Reporting. There are several technical guidance documents available for INSPIRE Discovery Services (based on OGC Catalogue Services, CSW), INSPIRE View Services (based on OGC Web Map Services, WMS), INSPIRE Download Services (based on OGC Web Feature Services, WFS) and INSPIRE Transformation Services (based on application profiles of OGC Web Processing Services, WPS). The technical guidance documents specify different requirements such as mandatory service operations, interoperable data encodings and specific QoS requirements.

From an INSPIRE service providers perspective, especially the specified QoS requirements are very challenging in terms of their implementation. An INSPIRE View Service (INSPIRE, 2010) for instance must be available 99% of the time (availability), the maximum initial response time of a GetMap request with 640x480 pixel must be 5 seconds (performance) and a service instance must be able to fulfill both of these criteria even if the number of served simultaneous service requests is up to 20 per second (capacity). Prior the Hybrid Cloud evaluation (Section 4), the scalability of a Open Source implementation[1] of the OGC Web Map Service (WMS) specification (OGC, 2006) was tested on a single machine regarding the INSPIRE QoS requirements for INSPIRE View Services. In the experiment a randomized request (quite simple in terms of requested geometry) was sent to the service. Figure 1 visualizes the average response time compared to the numbers of served simultaneous requests. The figure shows that the INSPIRE performance criteria could only be matched if the number of served simultaneous service requests exceeds least of all 5 per second. The benchmark was performed with the same testing tool (Section 4.2) that was used in all presented benchmarks in this document. The FOSS4G WMS Benchmarking experiments

[1] http://www.geoserver.org

(FOSS4G, 2010) optimized different WMS implementations and performed much more detailed benchmarks.

[FIGURE 1]

However, fulfilling the INSPIRE QoS requirements with classical service deployment technologies seems to be crucial. Cloud Computing as a new concept for virtualized infrastructures and for software deployment promises to fulfill exactly such performance and capacity requirements by automatically scaling the underlying infrastructure for instance in case of high amounts of users requesting a service.

2.2 Cloud Computing

Cloud Computing is one of the latest trends in the mainstream IT world (Gartner, 2008) (Gartner, 2009) and several mainstream IT companies such as Amazon, Google, Microsoft and Salesforce have already built up significant effort in this direction. The term Cloud Computing describes an approach in which applications and data are no longer located on desktop computers, but distributed over remote resources facilities operated by third party providers (Foster, 2008).

Cloud Computing replaces the classic multi-tier architecture of web services and creates a new set of layers (Sun Microsystems Inc., 2009). Software as a Service (SaaS) is the top layer offering web-based applications. Platform as a Service (PaaS) is the middle layer and encapsulates development and runtime environments (e.g. operating systems, databases or web application frameworks). Infrastructure as a Service (IaaS) is the bottom layer and delivers basic computational infrastructures as standardized services over the network. The bottom layer is then based on virtualized hardware provided to realize a Cloud Computing

infrastructure. The term "Virtualization" – one of the key technologies for realizing Cloud Computing - refers to the creation and execution of Virtual Machines (VMs). A VM is a software implementation of a computer containing an isolated operating system and additional application software. A so-called "Hypervisor" (a virtualization techniques mostly realized as software installed on a server) allows multiple copies of the same or different virtual machines (guest machines) to run concurrently on one physical computer (host machine).

The key characteristics of Cloud Computing are the ability to scale and provide computational power and storage dynamically in a cost efficient and secure way over the web. From a provider perspective, Cloud Computing enables IT companies to increase utilization rates of their existing hardware significantly. By outtasking software and data (even whole business processes) to facilities operated by third parties, Cloud users do not need to operate their own large-scale computational infrastructure anymore. Furthermore, cloud resources (i.e. storage or computational power) are allocated nearly in real-time and most Cloud Computing providers offer advanced mechanisms for scaling the deployed applications automatically on-demand (e.g. in case of high amounts of users requesting an application or a service). This allows cloud users to handle peak loads very efficiently without managing their own infrastructure. Therefore, Cloud Computing provides scalable infrastructures for outsourcing applications based on pay-as-you-go revenue models and is consequently enabling new business models with less up-front investments.

In essence, Cloud Computing is not a completely new concept. It moreover collects a family of well-known and established methods and technologies such as Virtualization under the umbrella of the term Cloud Computing (Sun Microsystems Inc., 2009).

When resources and services are made available through Cloud Computing in a pay-as-you-go manner to the general public, it is called a Public Cloud (Armbrust, 2010). When methods and technologies for realizing Cloud Computing are used to manage the internal data center of a company and when such a data center is not made available to the general public, it is called Private Cloud. In a so-called Hybrid Cloud, a local data center managed by a Private Cloud is combined with resources and services of a Public Cloud to handle tasks that cannot be performed in the local data center due to general hardware limitations and temporarily heavy workload (Armbrust, 2010).

[FIGURE 2]

2.3 Cloud Computing and GIS

Cloud Computing overlaps with some concepts of Distributed Computing and Grid Computing (Hartig, 2008). Both, Grid and Cloud environments provide a network infrastructure for scaling by sufficient storage and computational capabilities. In the geospatial domain, the area of Grid Computing has been addressed very early at various levels. This is reflected for instance through a Memorandum of Understanding (MoU) between the Open Geospatial Consortium (OGC) and the Open Grid Forum (OGF). Many research projects investigated how to merge the SDI concept and Grid Computing infrastructures (Di, Chen, Yang, & Zhao, 2003; Fleuren & Muller, 2008; Hobona, Fairbairn, & James, 2007; Lanig, Schilling, Stollberg, & Zipf, 2008; Padberg & Kiehle, 2009). The GDI-Grid project focused on solutions for the efficient integration and processing of geospatial data in the German national Grid Computing infrastructure D-Grid through OGC Web Services (Maué & Kiehle, 2009). A proof-of-concept was developed for "flood simulation", "noise propagation in urban areas" and "emergency routing for relief units" scenarios. The SEE-GEO (Secure Access to Geospatial Services)

project[2] investigated the means of making geospatial data securely available in the UK National Grid Service (NGS) for use by the UK academic sector. Most of conducted research focused on efficient and secure on-demand distributed processing and transfer of huge amounts of geospatial data in Grid Computing infrastructures through the SDI standards layer; as for instance presented in (Di et al., 2003) and (Woolf et al. ,2009). Other important aspects such as the scalable and efficient hosting of geospatial services, the economical efficiency of an underlying infrastructure and therefore the opportunity to realize sustainable business models for SDI service providers have not been addressed. Cloud Computing promises to provide solutions in this context.

On a conceptual level the interoperability of geospatial web services and the Cloud Computing paradigm (Everything as a Service, EaaS) was studied for instance in (Ludwig et al., 2010) and (Schaeffer et al., 2010a). The licensing concept presented in (Schaeffer et al., 2010b) for realizing access control and payment models in SDIs, was further developed in (Baranski et al., 2010a) to a general framework for realizing sustainable pay-as-you-go business models for geoprocessing service hosted in Public Clouds. First experiments with Cloud Computing for geographic applications have demonstrated that Cloud Computing can be used for on-demand available GIS hosting platforms. ESRI for instance offers a variety of Cloud-based applications and services[3]. ArcGIS Online allows users and developers to access maps (e.g. the Community Maps) online via Cloud-hosted web applications and services. Furthermore, customers of ESRI can access ArcGIS Server hosted at the Amazon Elastic Compute Cloud (EC2). So customers use the ESRI components on a Public Cloud instead of hosting the software internally. WeoGeo (USA) for instance provides a Cloud-based marketplace for the

[2] http://edina.ac.uk/projects/seesaw/seegeo
[3] http://www.esri.com/technology-topics/cloud-gis/index.html

transformation, storage and sale of geospatial data[4]. Although scaling aspects are considered, open SDI standards and security requirements are not addressed yet. Furthermore, currently a cloud-based "The Geospatial Platform" infrastructure is under development under the leadership of the US Federal Geographic Data Committee (FGDC), to promote the sharing of geospatial data, services, and applications to support all levels of government[5]. These examples show, that especially the commercial sector achieved considerable progress towards cost-efficient hosting of geospatial applications, services and data based on Cloud Computing. Furthermore, some experiments showed that Cloud Computing is suitable for realizing high-available and high-scalable geospatial services. In (Blower, 2010) an OGC WMS implementation was deployed at the Google App Engine (GAE). In (Baranski et al., 2009) an OGC WPS implementation was deployed at GAE and at Amazon EC2. Most of the conducted research focused on the performance behaviour of the deployed services and vendor-specific limitations for acquiring virtualized resources on-demand and in real-time.

The application of Hybrid Clouds for geospatial web services was first described in (Foerster et al., 2010) based on the example of geoprocessing services. In this paper, the research is extended by analyzing how Hybrid Clouds can achieve specific QoS such as the INSPIRE availability, performance and capacity requirements.

3. The Hybrid Cloud

This chapter introduces the design and the implementation of the proposed Hybrid Cloud for matching the INSPIRE QoS requirement without investing in rarely used hardware in advance.

3.1 Architecture

[4] http://www.weogeo.com
[5] http://www.geoplatform.gov

The proposed Hybrid Cloud supports SDI service providers to realize a failsafe and cost-efficient operation of SDI services. The major goal of the proposed Hybrid Cloud is always to provide sufficient computational resources to achieve a constant average service response time, independent of the numbers of users requesting a service. By incorporating external third-party resources into the local IT-infrastructure, the proposed architecture offers potentially unlimited resources on-demand and nearly in real-time. The proposed Hybrid Cloud combines the limited hardware resources of a local IT-infrastructure (local servers) and potentially unlimited resources at a Public Cloud provider (virtualized servers).

This section presents an implementation-independent view on the organizational units oft he proposed abstract Hybrid Cloud architecture (Figure 3). Deployment and concrete software recommendations for implementing the proposed abstract architecture are presented in the following section (Section 3.2).

- The *Proxy* component is the main entry point for all clients (users, applications and other services) and it controls access to the whole (local and third-party) server infrastructure. It receives all incoming service requests, forwards them to the Load Balancer at the Gateway, and returns the delivered response as if it was itself the origin.

The Gateway is an organizational unit containing a Load Balancer, a Cloud Controller, a Cloud Manager and a Virtual Machine (VM) Repository.

- The *Load Balancer* component contains a registry of all running service instances, the so-called IP Pool. The Load Balancer receives all forwarded requests from the Proxy and equally distributes them across all available service instances.

- The *Virtual Machine (VM) Repository* component realizes a local storage containing a set of prepared *Virtual Machine (VM)* images. Each VM image belongs to an offered service and contains a guest operating system, all required software components and related configurations. In the proposed Hybrid Cloud architecture two different types of VM images exists. One image type is dedicated for use in the local infrastructure. The other image type is dedicated for use at the Public Cloud. Anyway, for each offered service, one VM image for each of the two types must be provided at the VM Repository.
- The *Cloud Controller* component manages the virtualized local IT-infrastructure by providing an interface for starting and stopping VM instance on the local servers. Therefore, on each of the local servers a host operating system together with a *Hypervisor* must be installed. The only task for the Hypervisor is to run the guest operating systems (VM).
- The *Cloud Manager* component monitors the CPU load on each running VM in the architecture. If the overall CPU load of the system goes beyond a configured threshold, the Cloud Manager starts a new VM instance and adds the new running VM to the IP pool of the Load Balancer. In the ideal case, the VM is started via the Cloud Controller at a local IT-infrastructure. In the case that all local servers are busy, the VM is started at the Public Cloud. If the overall CPU load of the system goes below a configured threshold, the Cloud Manager stops the running VM instance with the lowest CPU load (with a priority for running VM instances at the Public Cloud). Before the Cloud Manager stops a running VM, the VM is removed from the IP pool of the Load Balancer.

[FIGURE 3]

Each time a new VM instance is added/removed from the IP pool of the Load Balancer, the Load Balancer must be restarted to notice the new resources. To avoid connection interruptions between the Proxy and the requesting clients (in the case the Load Balancer is not available for a short period of time), the Proxy component re-sends the forwarded requests to the Load Balancer until they could be processed successfully.

3.2 Implementation

Nearly all described components of the Hybrid Cloud are common Open Source software packages.

- The Proxy component is realized through an Apache HTTP Server[6] combined with the mod_proxy module[7]. The Apache HTTP Server is configured to act as a reverse proxy and therefore appears to the requesting client just like an ordinary web server.
- The Load Balancer is realized through the nginx web server[8] that is also configured to act as a reverse proxy.
- The Open Source Kernel-based Virtual Machine (KVM) hypervisor[9] was used in the local infrastructure. Therefore, the KVM image type is dedicated for use in the local infrastructure. Since Amazon EC2[10] was used as a Public Cloud provider, the Amazon Machine Image (AMI) image type is dedicated for use at the Public Cloud.
- Eucalyptus[11] is a software package for implementing Private Cloud infrastructures in local computer clusters. The Cloud Controller component uses the Eucalyptus Cloud Controller (CLC), the Eucalyptus Storage Controller (SC) and the Eucalyptus Cluster

[6] http://httpd.apache.org
[7] http://httpd.apache.org/docs/2.0/mod/mod_proxy.html
[8] http://nginx.org
[9] http://www.linux-kvm.org/page/Main_Page
[10] http://aws.amazon.com/ec2
[11] http://open.eucalyptus.com

Controller (CC) to manage the virtualized local IT-infrastructure. Via the Eucalyptus interface (that is compatible with the Amazon EC2 and Amazon S3 services), the Cloud Controller is able to start and stop VMs in the local IT-infrastructure. The local servers run Ubuntu 9.10, a KVM hypervisor and the Eucalyptus Node Controller (NC) to offer a cloud abstraction at each local server.

The Cloud Manager component is Open Source software developed from scratch to suit the INSPIRE-specific requirements and is publicly available at 52°North[12].

3.3 Parameters and Rules

The Cloud Manager is the core component and provides the scalability of the system either by starting/stopping VM instances in the local IT-infrastructure (Private Cloud) or the Public Cloud.

[TABLE 1]

The configuration parameters of the Cloud Manager affect the dynamic behaviour and therefore the scalability of the overall system (e.g. how fast new VM instances are available in case of high user demands). Since the resources at the Public Cloud are allocated based on a pay-as-you-go manner, these parameters affect also the financial efficiency of the proposed approach (e.g. how much the service provider has to pay for a more dynamic scalability).

4. Performance Evaluation

[12] http://www.52north.org

In this chapter a detailed evaluation of the proposed Hybrid Cloud approach is performed. Different INSPIRE-related QoS indicators such as the availability, the performance and the capacity of deployed services are benchmarked.

4.1 Scenario

The National Spatial Data Infrastructures (NSDIs) of the European member states are expected to be compatible regarding data content and data encodings. To ensure that geospatial data from different sources across the member states is interoperable, the INSPIRE directive recognizes the INSPIRE Transformation Service as "a special auxiliary service, designed to be used together with the other recognized service types (Discovery/View/Download) in order to make those services compatible with the common European specifications" (Kovanen et al., 2009).

In Section 2.1 the QoS requirements for INSPIRE View Services are described. Following the regarding technical guidance document, an INSPIRE Transformation Service must be available 99% of the time (availability), the maximum initial response time must 0.5 seconds per each Megabyte (MB) of input data (performance) and a service instance must be able to fulfill both of these criteria even if the number of served simultaneous service requests is up to 5 per second (capacity). The performance of an INSPIRE View Services implementations is shown to be crucial in times of high request rates (Section 2.1), the scalability and therefore the performance of INSPIRE Transformation Services implementation is expected to be crucial as well. In Section 4.3 a classic (single server) deployment of an INSPIRE Transformation Service is benchmarked. Furthermore, in Section 4.4 and Section 4.5 different aspects of the presented Hybrid Cloud are evaluated against INSPIRE QoS requirements.

The INSPIRE Coordinate Transformation Service is defined as an Application Profile (AP) of the OGC Web Processing Service, WPS (OGC, 2007) and first implementations of such a service interface are presented in (Kubik et al., 2009) and (Lehto, 2009). To perform the benchmarks and to evaluate the scalability of the proposed Hybrid Cloud architecture, the WPS implementation of 52°North was extended to support INSPIRE-compliant coordinate transformation.

In our benchmarks the Private Cloud consists of 4 instances, based on 2 servers with multi-core processors containing 2 cores each. The underlying virtualized hardware acquired at the Public Cloud (Amazon EC2) is unknown and only the family of instance types is known. In all benchmarks where the Public Cloud was activated, the Hybrid Cloud was configured to start up to 6 instances at Amazon EC2 from the "Small Instance" family.

4.2 Benchmarks

There are several Open Source tools available for performing web server benchmarks or load testing; for instance Apache JMeter[13] or ApacheBench that is part of the Apache HTTP Server Project. To control the amount of workload that is sent to a web service in a specific period of time (the number of parallel requests) and to control the logged benchmark indicators, a simple benchmark tool tailored to the particular needs of the intended demonstration scenario and the presented Hybrid Cloud approach was developed.

The tailored benchmark tool is implemented in Java and is public available as Open Source software at of 52°North. It sends a specific number of requests per sequence to a web service and slightly increases/decreases the number of requests per sequence over time. The benchmark tool supports the configuration of its behavior through the following list of

[13] http://jakarta.apache.org/jmeter

parameters (including default parameter values used in most of the presented experiments). A visualization of the process of increasing/decreasing the number of requests from one to the next sequence could be found in Figure 4.

[TABLE 2]

To get representative results, all of the experiments are repeated several times with both the same and slightly different benchmark configurations (e.g. faster/slower up- and down-scaling of the Hybrid Cloud). The outcomes are presented in the following sections. Relevant deviations from normal behaviour are described explicitly.

4.3 Classic Deployment

First, a benchmark for a classic (single server) deployment was performed. The WPS and the executed process run on a single server inside a web application container (Apache Tomcat 5.5) that was installed on a machine in the Private Cloud without allowing the Private Cloud to start new VMs. The web application container was configured to use local memory as much as possible, accept incoming HTTP connections as many as possible and create local threads as many as possible. No clustering, load balancing or other advanced distributed computing technologies were used to make use of multi-core processor or other servers in the network. Such a deployment is similar to server environments for production as in most of the governmental agencies offering INSPIRE services.

The visualized average response time of a request send to the classic deployment could be found in Figure 4.

The benchmark shows that the average response time of all requests that are sent to an INSPIRE Transformation Service that is deployed on a non-scaling single server increases significantly according to the number of served simultaneous requests. The average response time increases from 2.7 seconds (2 requests in 10 seconds) up to 61.3 seconds (40 requests in 10 seconds). This behaviour is not surprising. The processing time at the server for a single request is approximately 2.5 seconds (plus the overhead for receiving the input data and sending the response data) and during this time period the average CPU load of the targeted single-core machine is near by 100% most of the time. When 5 requests within 10 seconds are sent to the service (uniformly distributed over time), the server is still able to handle all requests one after another without interfering. If more than 5 requests within 10 seconds are sent to the service, the server has to queue arriving requests or has to pause and restart already processing requests. However, in our experiment a single-core machine is not able to handle more than 5 requests within 10 seconds without increasing the average response time significantly. Furthermore, the average response time increases faster than the number of requests due to a lot of computational overhead (pausing and restarting processes, memory allocation and copying, etc.).

[FIGURE 4]

Furthermore, in nearly all conducted experiments the web services throws internal server errors (e.g. XML parsing errors, out of memory exceptions) during peak load and a few times the whole server crashed under overload and further requests could not be processed anymore.

4.4 Private Cloud Deployment

Next, the Private Cloud was tested. The benchmark was performed without allowing the Cloud Manager to acquire additional resources at Amazon EC2. Therefore, the scalability of a Cloud-managed internal data center is reviewed in this section.

The visualized average response time of a request sent to the Private Cloud in comparison to the local hardware utilization rate is depicted in Figure 5.

[FIGURE 5]

The benchmark shows that the average response time of all requests that are sent to an INSPIRE Transformation Service that is deployed in the Private Cloud still increases according to the number of served simultaneous requests. The average response time increases from 1.6 seconds (4 requests in 10 seconds) up to 5.9 seconds (40 requests in 10 seconds). Compared to the benchmark results of the single server deployment (Section 4.3), this is a significant improvement. Even in times of peak load, the average response time is at maximum by the factor of 3.6 higher than in idle times (compared to maximum 22.7 times higher than in idle times for the single server deployment). Furthermore, in none of the conducted experiments the web services threw internal server errors or crashed.

4.5 Hybrid Cloud Deployment

Finally, the complete Hybrid Cloud was tested. The benchmark was performed with allowing the Cloud Manager to use all 4 local instances (Private Cloud) and to acquire up to 6 additional VM instances at Amazon EC2 (Public Cloud). Therefore, the scalability of the complete Hybrid Cloud architecture is reviewed in this section.

The visualized average response time of a request send to the Hybrid Cloud in comparison to the local and third-party hardware utilization rate could be found in Figure 6.

[FIGURE 6]

The benchmark shows that the average response time of all requests that are sent to an INSPIRE Transformation Service deployed in the Hybrid Cloud still slightly increases but stays nearly constant independent of the number of served simultaneous requests. The average response time increases from 1.7 seconds (4 requests in 10 seconds) up to 3.3 seconds (40 requests in 10 seconds). Compared to the benchmark results of the single server deployment (Section 4.3) and also compared to the benchmark results of the Private Cloud deployment (Section 4.4), this is a significant improvement. Even in times of peak load, the average response time is maximum 1.9 times higher than in idle times (compared to 22.7 times for the single server deployment and 3.6 times higher for the Private Cloud deployment). Furthermore, in none of the conducted experiments the web services throws internal server errors or crashed.

5. Discussion

The proposed Hybrid Cloud combines the limited server infrastructure of a Private Cloud with the potentially unlimited resources of a Public Cloud to realize specific Quality of Service (QoS). Depending on the overall system load, the Hybrid Cloud starts/stops either local or external Virtual Machines (VM) and distributes incoming service requests across all available VM instances. The Hybrid Cloud was implemented with common Open Source software and tailored components for realizing QoS regarding the INSPIRE requirements.

The benchmarks show that the proposed Hybrid Cloud is able to provide sufficient computational resources to scale a deployed service regarding INSPIRE requirements and to guarantee a (nearly) constant response time independent of the number of simultaneous served requests. Furthermore, the benchmarks show that there is a mutual dependency between the utilization rate of the local as well as the external third-party infrastructure (the number of running VM instances) and the overall workload of the deployed services. Therefore, the Hybrid Cloud helps service providers to operate their local data center as efficient as possible (e.g. only minimum number of required resources is allocated to meet the specific QoS) and to allocate third-party resources provide sufficient QoS in an economic efficient manner (e.g. on a pay-as-you-go basis).

In the conducted benchmarks the deployed service always failed to meet the claimed QoS requirements for the INSPIRE Coordinate Transformation Service (Section 4.1) as a result of really challenging INSPIRE requirements (compared to the other INSPIRE service types). The accomplished WMS experiment (Section 2.1) certainly showed that failing to fulfill the INSPIRE related QoS requirements is not related to a specific implementation or service type. The described general performance behaviour (significant higher response time for an increased number of simultaneous requests) could be observed for all types of services. However, the Hybrid Cloud helps SDI service providers to scale the SDI service infrastructure to realize (nearly) constant response times and therefore to fulfill INSPIRE related QoS requirements in the first place.

The most limiting factors of the Hybrid Cloud are the time needed for starting new VMs, the network overhead when forwarding requests to the Public Cloud and the complex configuration of the Cloud Manager. Depending on the configured threshold the Cloud Manager starts or stops new VM instances at the Private or the Public Cloud. New instances

are not available in real-time (it takes up to 30 seconds in the Private Cloud and up 20 seconds at Amazon EC2) and stopped instance may cause connection interruptions between the Proxy and the Load Balancer/VM, so that a request must be repeated. Forwarding requests to running VMs in the Private Cloud is not that time consuming, but the network bandwidth between the Private Cloud and the Public Cloud is an limiting factor that influences the overall response time and affects the QoS experience of the requesting user. The configuration of the Cloud Manager directly affects the dynamic behaviour and therefore the scalability of the overall system (e.g. how fast new VM instances are available in case of high user demands). Therefore, a fine-tuning of the Cloud Manager is required. Furthermore, the local hardware and the virtualized hardware acquired at the Public Cloud provider (e.g. the family of instance types at Amazon EC2) is of much importance as well.

One topic for further research would be the analysis, how much of a process can be parallelized (would be easy for that specific coordinate transformation algorithm), how to distribute each oft the parallelized sub-processes over multiple Cloud instances and how much the processing speed could be increased with such an approach as for instance examined in (Baranski, 2009) for Grid Computing infrastructures and geoprocessing services.

There are still some general open issues and legal barriers regarding the adoption of Cloud Computing. Most of them deal with data security and trust aspects. How could service and data providers ensure that their data and services are protected against unauthorized access? How could the outsourcing of (potentially) personal data to third-party infrastructures be aligned with the restrictions and provisions of federal state law? Since geospatial data often contains personal information, these issues arise especially in the context of the geospatial domain. To get a real benefit from the Cloud Computing paradigm, these open issues have to be tackled on different levels. First, new methods and technologies have to be established to

support the secure and trusted storage and processing of rights protected and personal data in Public Cloud environments (e.g. concepts like the Amazon Virtual Private Cloud). Second, existing laws and provisions must be modified in order to allow the outsourcing of personal data to external data centers operated either by governmental agencies or classical market-oriented companies. A trust relationship between data owners and external data centers could be established for instance through a 'trust certificate' that is granted by a confidential third-party and that requires a standardized trust validation and monitoring processes.

Furthermore, the economic efficiency of the utilization of Public Cloud infrastructures is frequently propagated. But a detailed and use-case specific comparison between the return on investment of operating local data centers and the overall costs for utilizing third-party infrastructure on a pay-as-you-go basis has not been investigated and formalized yet.

References

Armbrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A. Stoica, I. and Zaharia, M. (2010). A view of cloud computing. Commun. ACM 53, 4 (April 2010), 50-58.

Baranski, B. (2008): Grid Computing Enabled Web Processing Service. In Mohamed Bishr, Edzer Pebesma, & Thomas Bartoschek (Eds.), Proceedings of the 6th Geographic Information Days, Ifgi Prints (Bd. 32, S. 243-256). Held at GI-days 2008, Muenster, Germany

Baranski, B., Schäffer, B. and Redweik, R. (2009). Geoprocessing in the Clouds. Held at Free and Open Source Software for Geospatial (FOSS4G) Conference, Sydney, Australia

Baranski, B., Deelmann, T. and Schäffer, B. (2010a). Pay-per-Use Revenue Models for Geoprocessing Services in the Cloud. 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services (WebMGS 2010). Como, Italy

Bernard, L., Craglia, M., Gould, M. and Kuhn, W. (2005). Towards an SDI research agenda. Proceedings of the 11th EC GI & GIS Workshop - ESDI: Setting the Framework, Alghero, France. 147-151.

Bernard, L., Kanellopoulos, I., Annoni, A., and Smits, P. (2005). The european geoportal - one step towards the establishment of a european spatial data infrastructure. Computers, Environment and Urban Systems, (29), 15-31.

Blower (2010): GIS in the cloud: implementing a web map service on Google App Engine. Com.Geo 2010.

Di, L., Chen, A., Yang, W., and Zhao, P. (2003): The Integration of Grid Technology with OGC Web Services (OWS) in NWGISS for NASA EOS Data (pp. 24-27). Presented at the GGF8 & HPDC12 2003, Seattle, WA, USA: Science Press.

EC. (2007). Directive 2007/2/EC of the european parliament and of the council of 14 march 2007 establishing an infrastructure for spatial information in the european community (INSPIRE)

Fleuren, T., and Muller, P. (2008). BPEL workflows combining standard OGC web services and grid-enabled OGC web services. Proceedings of the Software Engineering and Advanced Applications Conference. SEAA'08. Parma, Italy. 337-344.

Foerster, T., Baranski, B., Schäffer, B., and Lange, K. (2010). Geoprocessing in Hybrid Clouds. In A. Zipf, K. Behncke, F. Hillen, & J. Schaefermeyer (Eds.), Die Welt im Netz (pp. 13-19). Presented at Geoinformatik 2010, Kiel, Germany: Akademische Verlagsgesellschaft.

FOSS4G (2010). WMS Benchmarking. Organized by Jeff McKenna and Andrea Aime. [Online] Available: http://2010.foss4g.org/wms_benchmarking.php

Foster, I., Zhao, Y., Raicu, I and Lu, S. (2008) Cloud computing and grid computing 360-degree compared. [Online]. Available: http://arxiv.org/abs/0901.0131

Gartner (2008) Gartner Says Cloud Computing Will Be As Influential As E-business. Gartner Press Release. [Online] Available: http://www.gartner.com/it/page.jsp?id=707508

Gartner (2009) Gartner Says Cloud Application Infrastructure Technologies Need Seven Years to Mature. Gartner Press Release. [Online]. Available: http://www.gartner.com/it/page.jsp?id=871113

Hartig, K. (2008) What is Cloud Computing? The cloud is a virtualization of resources that maintains and manages itself. .In: NET Developers Journal, SYS-CON Media.

Hobona, G., Fairbairn, D., and James, P. (2007). Workflow enactment of grid-enabled geospatial web services. Proceedings of the 2007 UK e-Science all Hands Meeting,

INSPIRE (2009). Draft Implementing Rules for INSPIRE Transformation Services, Version 3.0. Drafting Team "Network Services", INSPIRE

INSPIRE (2010).Technical Guidance to implement INSPIRE View Services, Version 2.12. IOC Task Force "Network Services", INSPIRE

Kovanen, J. and Lehto, L. (2009). INSPIRE Coordinate Transformation Service - the Specification and experiences gained from a pilot implementation. GSDI 11 World Conference. 2009. Rotterdam, The Netherlands

Kubik, T. and Kopanczyk B. (2009). Implementing WPS as an Coordinate Transformation Service. GSDI 11 World Conference. 2009. Rotterdam, The Netherlands

Lanig, S., Schilling, A., Stollberg, B., & Zipf, A. (2008). Towards standards-based processing of digital elevation models for grid computing through web processing service (WPS). In O. Gervasi, B. Murgante, A. Laganà, D. Taniar, Y. Mun & M. Gavrilova (Eds.), Computational science and its applications ICCS, LNCS 5073 (pp. 191-203). Berlin / Heidelberg: Springer-Verlag.

Lehto, L. (2009) Real-Time content transformations in the European Spatail Data Infrastructures.. GSDI 11 World Conference. 2009. Rotterdam, The Netherlands

Ludwig, B. and Coetzee, S. (2010). A comparison of Platforms as a Service (PaaS) aClouds with a detailed reference to security and geoprocessing services.. 1st International Workshop on Pervasive Web Mapping, Geoprocessing and Services (WebMGS 2010). Como, Italy

Masser, I. (2005). GIS worlds : Creating spatial data infrastructures (1st ed.). Redlands, California: ESRI Press.

Maué, P. and Kiehle, C. (2009). Grid Technologies for Geospatial Applications – An Overview. GIS.Science, 3:65–67

OGC (2006). OpenGIS® Web Map Server Implementation Specification. Beaujardiere, J. (Edt.). OGC 06-042

OGC (2007). OpenGIS® Web Processing Service. OpenGIS Standard. Schut, P. (Edt.). OGC 05-007r7

Onsrud, H. J. (2007). Research and theory in advancing spatial data infrastructure concepts. Redlands, CA: ESRI Press.

Padberg, A., and Kiehle, C. (2009). Towards a grid-enabled SDI: Matching the paradigms of OGC web services and grid computing. International Journal of Spatial Data Infrastructures Research, Special Issue GSDI-11, , 2010/31/12.

Schäffer, B., Baranski, B., and Foerster, T. (2010a). Towards Spatial Data Infrastructures in the Clouds. In M. Painho, M. Santos, & H. Pundt (Eds.), Geospatial Thinking, Lecture Notes in Geoinformation and Cartography (pp. 399-418). Held at The 13th AGILE International Conference on Geographic Information Science, Guimarães, Portugal: Springer Verlag

Schaeffer, B., Baranski, B., and Foerster, T. (2010b). Licensing OGC Geoprocessing Services as a Foundation for Commercial Use in SDIs. In Proceedings of the Second International Conference on Advanced Geographic Information Systems, Applications and Services (pp. 111-116). Held at the Geoprocessing 2010, St. Maarten, Netherlands Antilles

Scholten, M., Klamma, R., & Kiehle, C. (2006). Evaluating performance in spatial data infrastructures for geoprocessing. IEEE Internet Computing, 10(5), 34-41.

Sun Microsystems Inc. (2009) Cloud Computing at a higher level. [Online] Available: https://slx.sun.com/files/Cloud_Computing_Brochure_2009.pdf

Tu, S., Flanagin, M., Wu, Y., Abdelguerfi, M., Normand, E., Mahadevan, V. and Shaw, K. (2004). Design strategies to improve performance of GIS web services. Proceedings of the

International Conference on Information Technology: Coding and Computing (ITCC'04), Las Vegas, NV. 444-448.

Woolf, A. and Shaon, A. (2009) An approach to encapsulation of Grid processing within an OGC Web Processing Service. AGILE 2009: Grid Technologies for Geospatial Applications, Hannover, Germany

Yang, P., Cao, Y., and Evans, J. (2007). Web map server performance and client design principles. GIScience} \& Remote Sensing, 44(4), 320-333. doi:10.2747/1548-1603.44.4.320

Zhang, J., Pennington, D. D., and Michener, W. K. (2007). Performance evaluations of geospatial web services composition and invocation. Transactions in GIS, 12(1), 59-73.

Tables

Table 1: The scalability of the Hybrid Cloud could be configured through the configuration parameters of the Cloud Manager component.

Parameter Name	Default Value	Parameter Description
Breach Duration	15	This parameter describes how "fast" VM instances are stopped when the lower threshold is reached.
Period	5	The parameter describes the repeat interval for monitoring the CPU load of each running VM.
Upper Threshold	20	This parameter describes the upper CPU load threshold (relevant for starting new VM instances).
Lower Threshold	10	This parameter describes the lower CPU load threshold (relevant for stopping running VM instances).
Statistics	"average"	This parameter describes how the monitored CPU load history influences the calculation if the upper/lower threshold is reached. Possible values are "minimum", "maximum" and "average".
Maximum Public Cloud Instances	6	This parameter describes how many Amazon EC2 instances the Cloud Manager could start.

Table 2: The tailored benchmark tool supports the configuration of its behaviour through the following parameters.

Parameter Name	Default Value	Parameter Description
MIN	1	The minimum number of requests in the first sequence of a benchmark run.
MAX	40	The maximum number of requests in the middle sequence of a benchmark run.
STP	1	The step size for increasing/decreasing the number of requests from one to the next sequence. The overall
REP	2	The number of times a sequence with a fixed number of requests is repeated.
DUR	10000	The duration of a single sequence with a fixed number of requests (Milliseconds). The requests that are sent to the targeted web service are uniformly distributed over the specified time period.
URL	$url	The URL of the targeted web service.
REQ	$file	A path to a file containing the (XML-based) request that is sent to the targeted web service.
LOG	$directory	A path to a directory where the measured performance indicators are stored after a benchmark run.

Figures

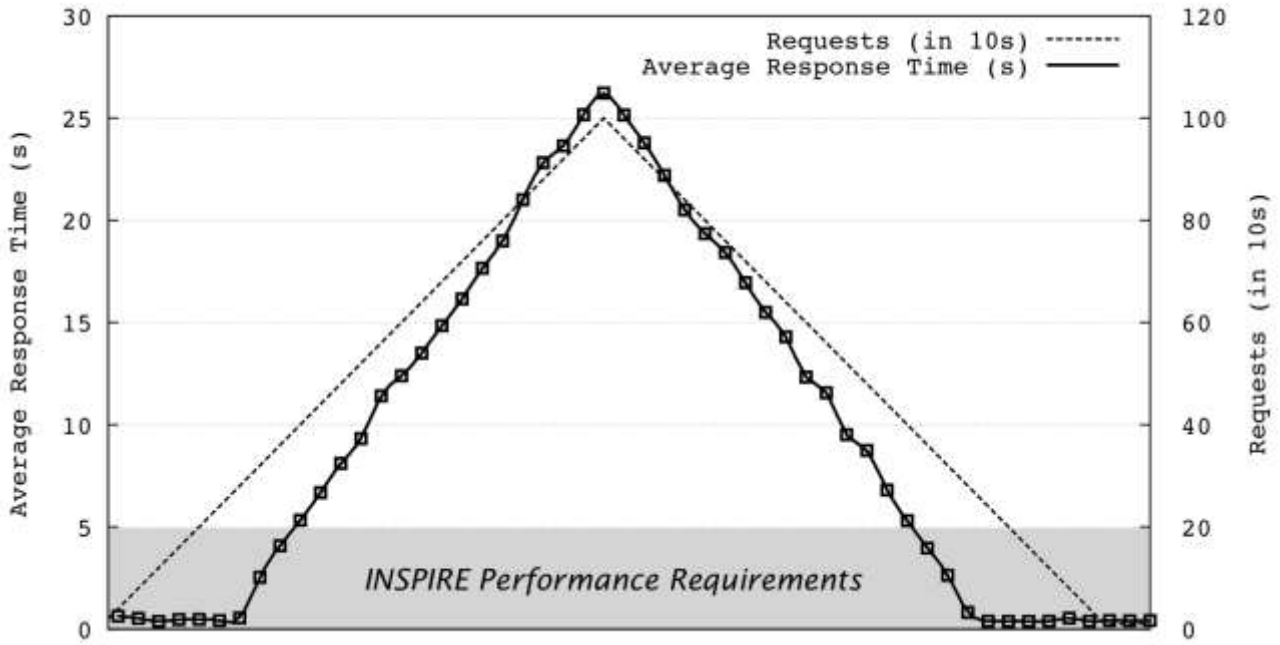

Figure 1: The typical response behaviour of an OGC WMS in case of a high number of served simultaneous requests.

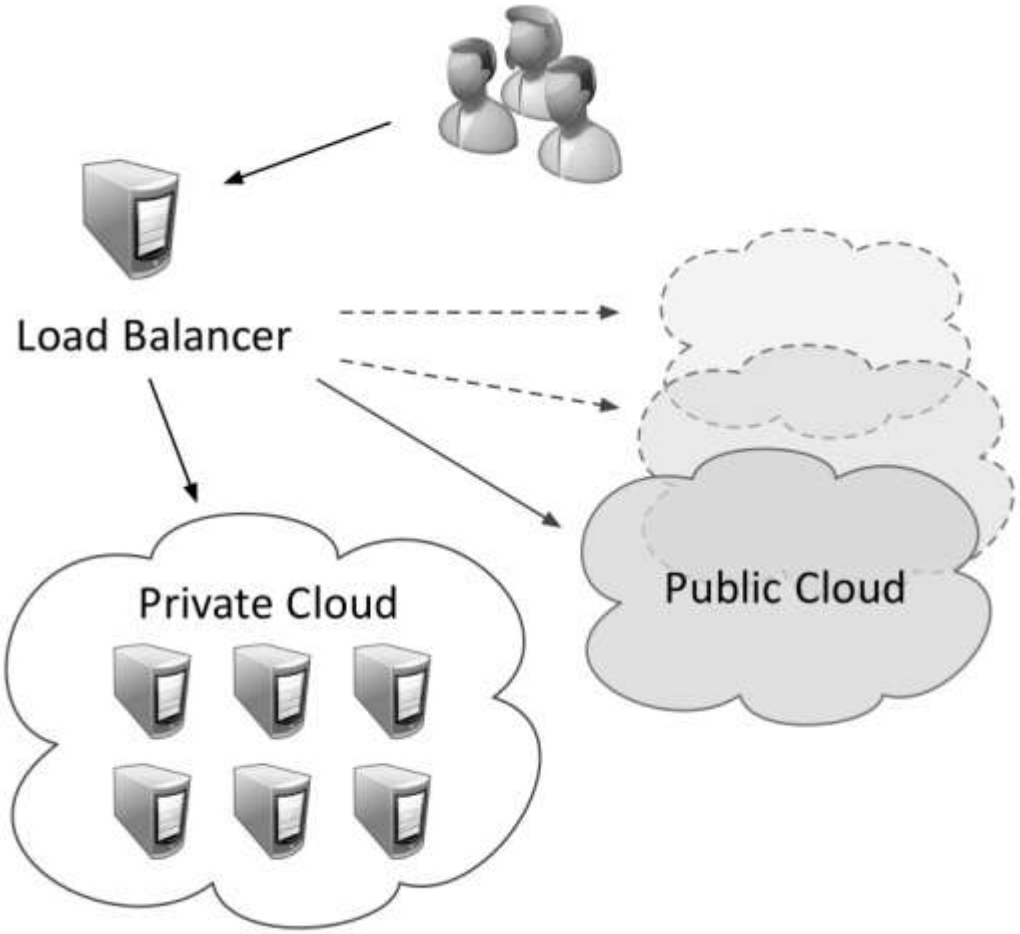

Figure 2: The Hybrid Cloud approach combines the benefits of Cloud-managed local IT infrastructure and potentially unlimited resources of external Cloud providers.

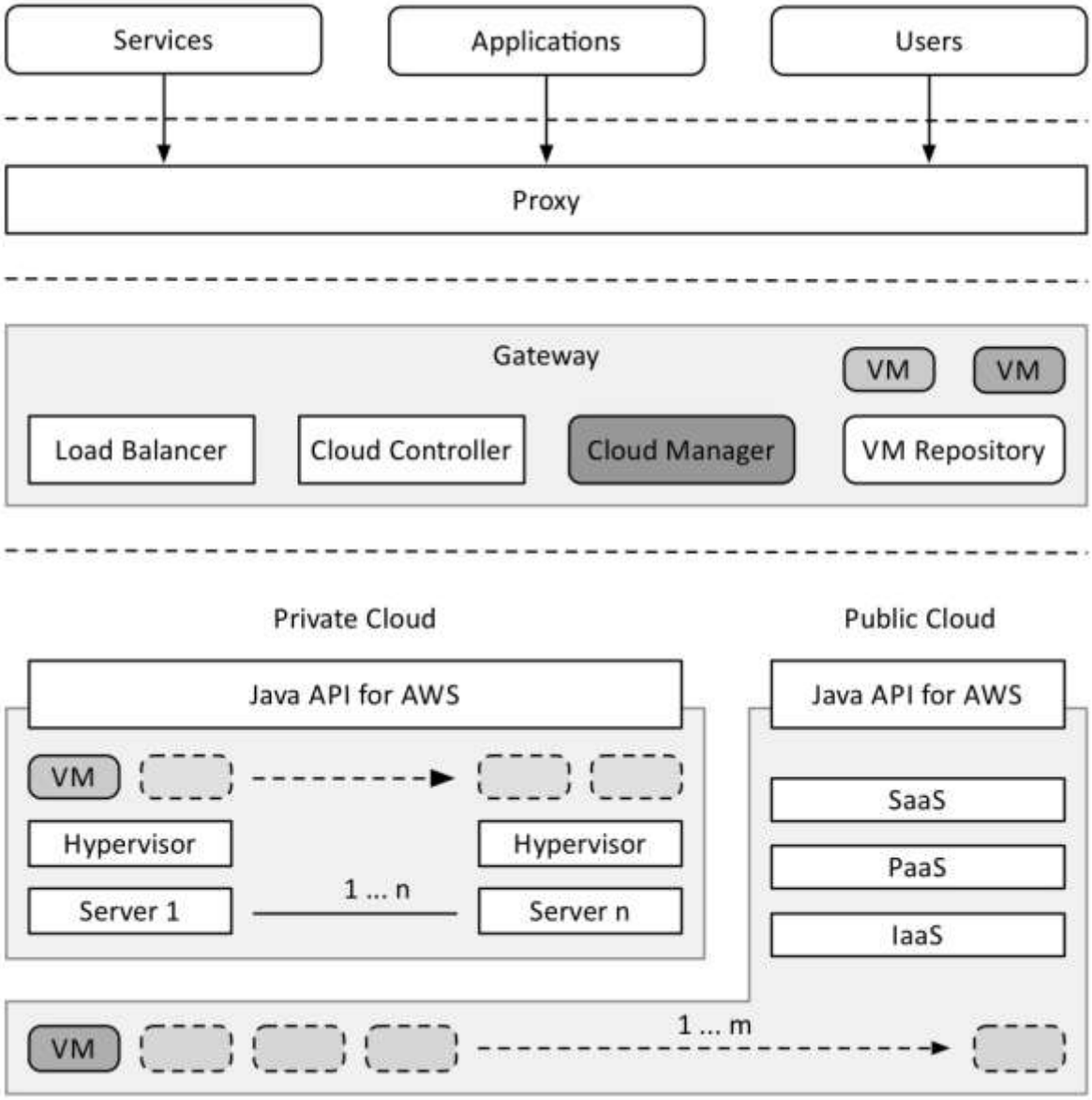

Figure 3: An overview about the Hybrid Cloud architecture.

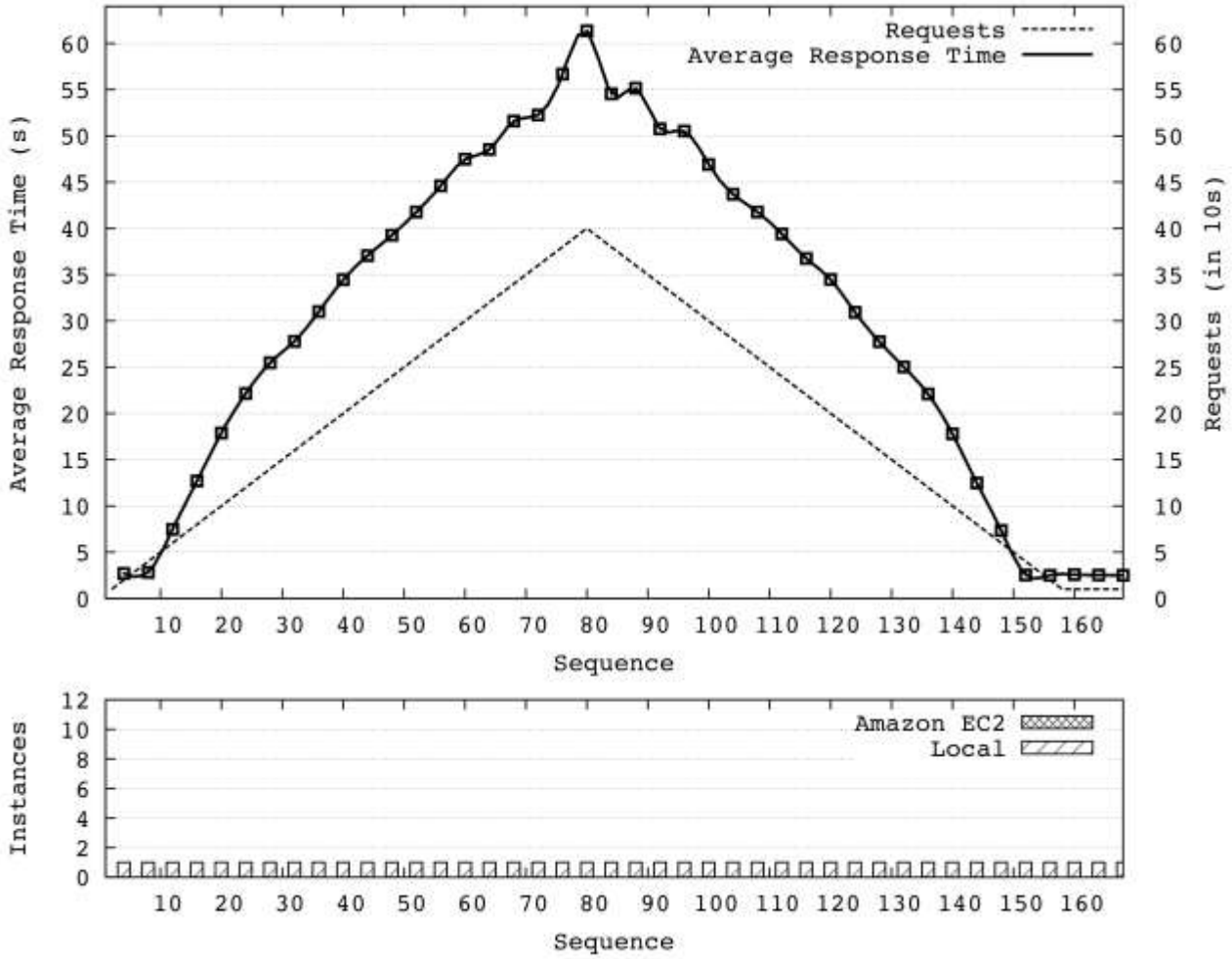

Figure 4: The average response time of an INSPIRE Coordinate Transformation Service according to the number of served simultaneous requests in single server deployment.

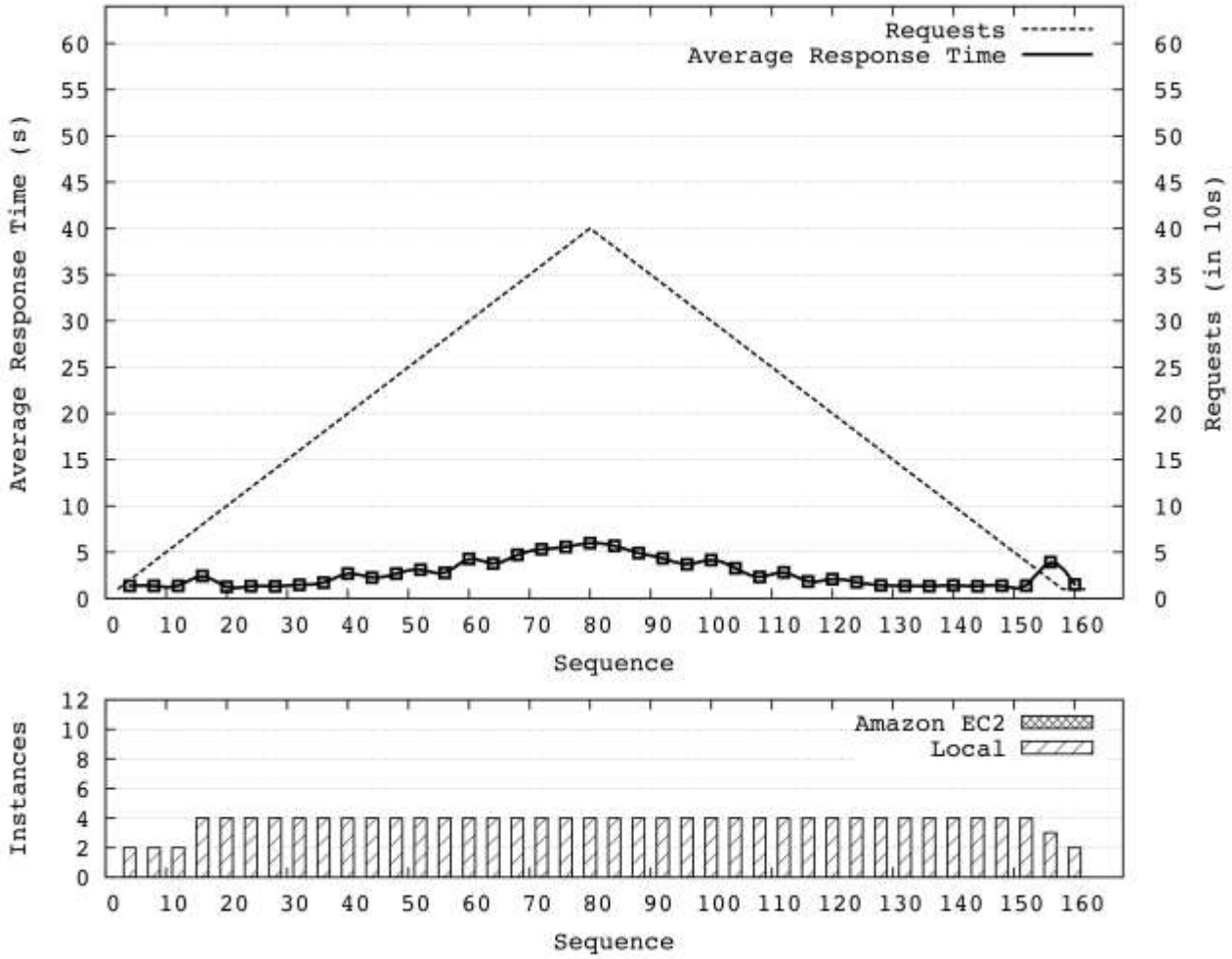

Figure 5: The average response time of an INSPIRE Coordinate Transformation Service according to the number of served simultaneous requests in a Private Cloud.

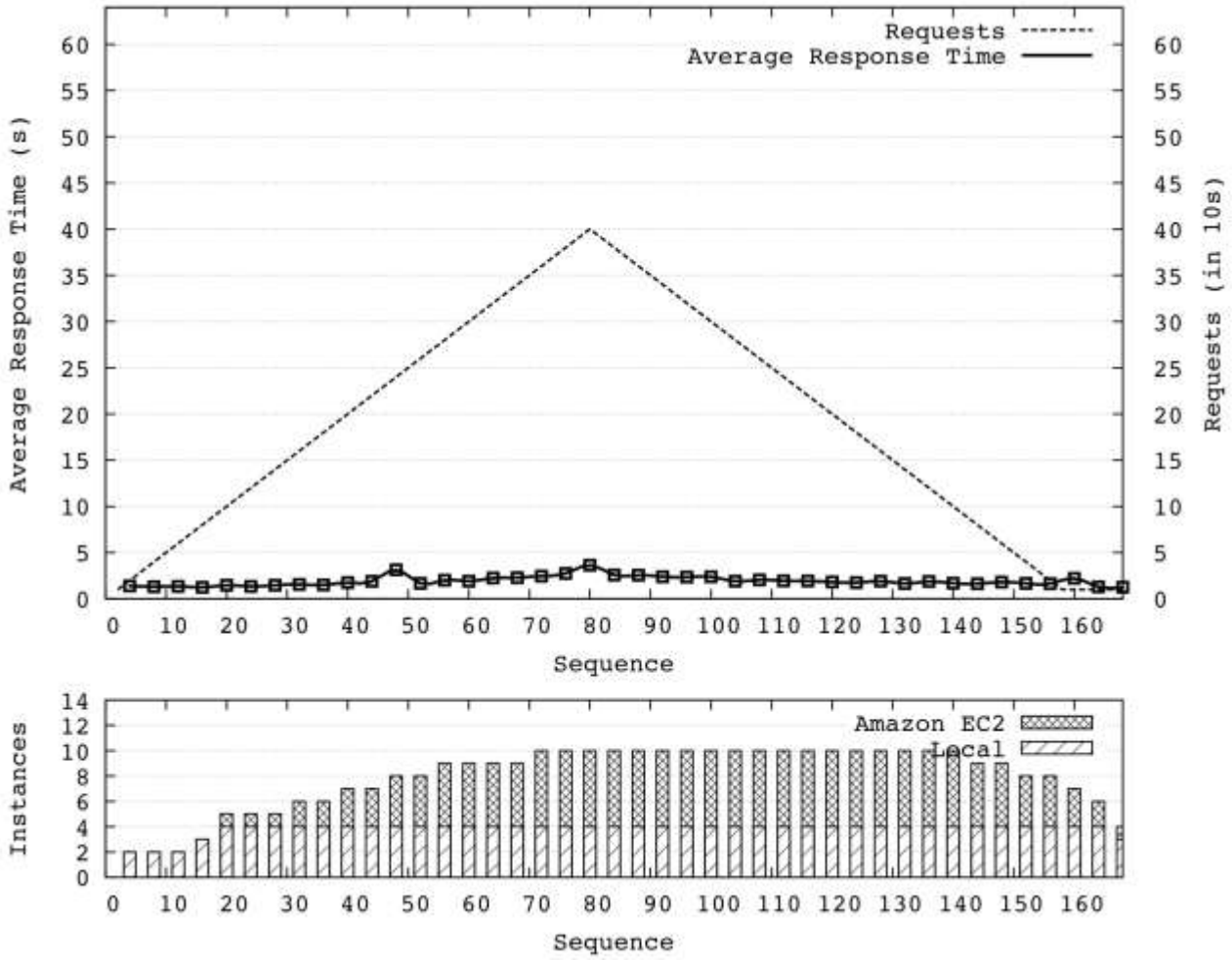

Figure 6: The average response time of an INSPIRE Coordinate Transformation Service according to the number of served simultaneous requests in a Hybrid Cloud.

List of Figures