Cloudified IP Multimedia Subsystem (IMS) for Network Function Virtualization (NFV)-based architectures

Giuseppe Carella[1], Marius Corici[2], Paolo Crosta[3], Paolo Comi[3], Thomas Michael Bohnert[4], Andreea Ancuta Corici[2], Dragos Vingarzan[2], Thomas Magedanz[1]

[1]: Technische Universität Berlin, Germany, {giuseppe.a.carella, thomas.magedanz}@tu-berlin.de
[2]: Fraunhofer FOKUS, Germany, {marius-iulian.corici, andreea.ancuta.corici, dragos.vingarzan}@fokus.fraunhofer.de
[3]: Italtel, Italy, {paolosecondo.crosta, paolomaria.comi}@italtel.com
[4]: Zurich University for Applied Sciences, Switzerland, thomas.bohnert@zhaw.ch

Abstract—**The maturity reached by virtualisation technology enabled great innovation for efficient applications and services development and delivery, independent of the underlying hardware equipment, especially with the large deployment of off-the-shelf hardware based cloud infrastructures. In order to take advantage of this technology, the existing network functions have to be developed and adapted to the new paradigm. However, traditional telecom services are still implemented on dedicated hardware resulting in high deployment and maintenance costs compared to the other already cloudified services. ETSI Network Functions Virtualisation (NFV) aims to fill this gap by applying to telecom the virtualisation technologies.**

This paper introduces a set of three software architectures for efficient virtualisation of IP Multimedia Subsystem (IMS) in different operator environments responding to the high level requirements of the ETSI NFV use case for virtualizing operator core network functions. Additionally, a management architecture for simplifying the deployment and runtime orchestration of such a virtual service on top of a cloud infrastructure is presented. Furthermore, one of the IMS software architectures was implemented based on the Fraunhofer FOKUS Open IMS Core, measured and evaluated on top of an OpenStack cloud.

Keywords - Mobile Cloud Networking, Network Functions Virtualisation, IP Multimedia Subsystem, Cloudification

I. INTRODUCTION

Cloud computing mechanisms have already gained broad attention attracting a steadily increasing number of service providers by enabling service realization with optimized resource consumption, and by enabling the outsourcing of infrastructure and service management costs.

To guarantee high availability and performance level that are typical for telecom services, both hardware and software components must be extremely reliable, well tested and proven in their capabilities. This type of systems are usually called Carrier Grade Systems, tested and engineered to meet or exceed "five-nines" high availability standards [1], and provide very fast fault recovery through redundancy. To reach these results, carrier solutions were until now based on components developed on dedicated hardware and interconnected through vendor specific synchronization and load balancing mechanisms. To cope with high peak situations and high availability requirements, telecom service infrastructures are usually over-provisioned, sometimes by an order of magnitude, leading to low resource utilization rate during typical times and overall highly increasing the deployment – number of hardware components required to be integrated - and operational costs – energy consumption, management, etc.

Additionally, hardware resource utilization significantly varies with current user load during the day or during the week. That holds true even more for special peak times in the year, such as during special sport and cultural events and during specific periods, like New Year's Eve. Moreover, the network shall guarantee service continuity even in case of very critical environmental situations such as fire, flood or earthquake. To satisfy the availability requirement many network elements shall be duplicated and deployed in different geographical locations [2]. Hence, elastic resource allocation is the basis for best-effort service delivery models, elastic scalability is definitely crucial for QoS sensitive services, such as the case of the multimedia services [3]. With the advent of virtualisation, telco operators started to realize how they could gain tremendous opportunities affecting costs and revenues of traditional business models, while improving the quality of user experience to their customers.

The IP Multimedia Subsystem (IMS) [4] is the overlay-architecture for session control in all-IP Next Generation Networks (NGNs) aiming at openness and interoperability by adopting a separated application-layer approach based on the Session Initiation Protocol (SIP) signaling [5]. Common hardware architectures currently address mainly applications and are not yet suitable for complex service control platforms, such as IMS. In order to level this limitation, a larger number of entities have to be deployed for the same number of subscribers, thus resulting in a higher management complexity doubled by the automatic scaling procedures. This results in a critical level of instability of the overall system that should be mitigated through a service orchestration function. The deployment of IMS "as a Service" (IMSaaS) is expected to reduce the overall deployment and management costs through the acquisition of software components running on top of common hardware architecture and through elastic deployments. The system might become less stable and highly distributed, requiring novel paradigms of management and orchestration.

This paper presents the results obtained through the virtualisation of IMS on top of a cloud based infrastructure based on OpenStack [6], a best practice for its management and orchestration as well as a set of evaluation measurements proving the feasibility of deploying IMS “as a Service” on top of Mobile Cloud Networking infrastructures.

The remainder of this paper is structured as follows. Section II provides background information, Section III describes the functional design of the IMSaaS architecture solution and the software realization solutions while Section IV describes the evaluation of a selected IMSaaS architecture and Section V concludes the paper providing an outlook on next steps.

II. BACKGROUND AND RELATED WORK

IMS [4] is a signalling platform aiming at providing session control for multimedia services. It includes functionality for End User authentication and authorization, call control and charging for multimedia sessions, as well as QoS decision and notifications at data path level through the integration with core network platforms such as the 3GPP Evolved Packet Core (EPC) [7]. Fig.1 show the typical IMS architecture involving only the signalling nodes which are the main target of the virtualisation solution presented in this paper.

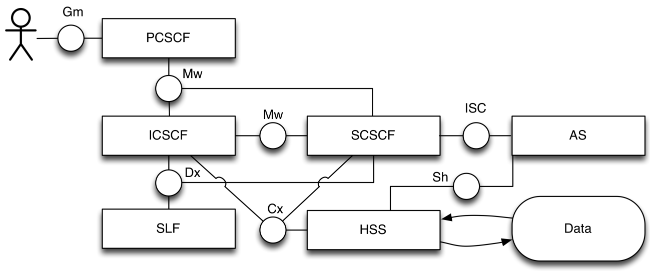

Figure 1. Typical IMS Architecture

In particular those are the main functional entities:

- **Proxy Call Session Control Function (P-CSCF) -** is the first point of contact between the IMS terminal generically named IMS User Equipment (IMS UE) and the IMS network. The IMS UE does a DNS query for resolving the P-CSCF Fully Qualified Domain Name (FQDN). In case of multiple P-CSCF there will be a DNS round-robin mechanism that will be used by the IMS terminal for selecting the entry P-CSCF. The P-CSCF is allocated to the IMS terminal and does not change for the duration of the REGISTRATION (typically 1hour for fixed networks). For enabling elasticity the user should de-register from one instance and register to a new one.
- **Serving Call Session Control Function (S-CSCF)** – is the central node of the signalling plane. It is basically a SIP server and it performs also some session control. It acts also as SIP registrar maintaining a binding between the IP address of the user’s IMS UE and the user’s SIP address of record (Public User Identity). It has also a Diameter interface with the HSS for downloading the user profile containing a set of triggers and initial filter criteria for redirecting SIP messages to application servers. In case of existing triggers for the incoming SIP requests, it will redirect those requests to the selected application servers.
- **Interrogating Call Session Control Function (I-CSCF)** is a SIP proxy located at the edge of an administrative domain. Its address is listed in the DNS records of the domain. It has a Diameter interface to the Subscriber Location Function (SLF) (introduced underneath) for retrieving the HSS associated with the users. It routes SIP requests to the appropriate S-CSCF based on the user profile information received from the HSS.
- **Home Subscriber Server (HSS)** – represents the central repository for end user subscription profiles and network connectivity dynamic information. In particular, it contains security information, authentication and authorization, user profile information, trigger points, initial filter criteria and S-CSCF association points. In one domain there could be more than one HSS. But all the data belonging to a user need to be stored in a single HSS.

The following additional functions are extending the IMS architecture:

- **Subscriber Location Function (SLF) –** is used by the other components for retrieving the HSS,. It is usually used in case of multiple HSS instances in a single domain.
- **Application Server (AS)** – is a SIP entity that hosts and executes services. It has a critical role in the IMS infrastructure, because most of the time it is the component that implements the logic of services like messaging, presence, etc. Customers of Next-Generation-Services require high-performance and reliable application servers in their core networks.
- **User Equipment (UE) –** is a device at customer premises or under customer control able to act as originating or terminating endpoint of multimedia sessions such as a SIP phone.

Several works in the field of cloud computing have already been presented, but only very limited part of them focused on cloud-based IMS/telecommunication infrastructures and services. Many virtualization techniques were introduced in the IT world and Telco’s increased their interest of the potentialities of pure software implementations over virtualized platforms. After this first phase, many players continue evolving IMS functionalities towards the cloud or cloud-compatible model. One of the most relevant initiatives is the open-source project called Clearwater [8]. Clearwater is an open source implementation of IMS embracing a cloud-oriented design making it extremely well suited for deployment in a Network Functions Virtualization (NFV) [9] environment. The developers started to rethink the structure of the software itself, because cloud-based virtual components cannot be made reliable by conventional means. This is an interesting study of some of the issues in cloud-based virtualization of network features.

Virtualization and cloud-based architectures are actually appealing for Telcos. However, many challenges remain. New redundancy models for guaranteeing carrier-grade availability level, automatic resource scaling for reducing the need of over-provisioning the network equipment, adequate performance level for effective real-time media processing are some of the issues to analyze and to address in this context. Collectively known as NFV, the evolution to pure software in cloud generates great interest to Telcos but requires a lot of study and experimentation to understand and amend its limitations and to exploit its value.

Authors of [10] proposed novel solutions for enhancing the intradomain scalability problem. Whereas authors of [11] focus on a specific service, a cloud-based IMS presence service, in [12] are focusing on Web Services having a similar approach but with a limited number of analyzed monitored data and using non-weighted round robin load balancing algorithm. In [13] a "profile-based" solution is being described, which only takes into account the CPU utilization of a given Virtual Machine (VM).

III. IMS As A Service architecture

In order to obtain an efficient IMS-as-a-Service architecture, the following research steps were considered.

First, virtualization presumes the logical separation between functional entity and the underlying host. A function is realized as software on top of a container, that is, a Virtual Machine (VM), running on top of a generic host. The hosts and the containers are considered rather uniform, due to the underlying infrastructure and the convergence of cloud control infrastructures. In order to allow this transition from having dedicated hardware for network components, to deploy them on top of virtual machines, the most important requirement is to have network entities realized in software. Indeed it is not only possible to deploy them on physical machines with common hardware, but also on VMs. Once a network function is virtualized, meaning it has the possibility to be deployed on a VM, its lifecycle needs to be orchestrated.

Virtualization of network elements and their implementation as software components is only an aspect of the cloudification process. Once the components are virtualized, they can reside on top of virtual machines, and all the cloud principles can be applied also to them. An elastic infrastructure is able to automatically increase or reduce the number of resources potentially without limits. For automatically enabling elasticity of network elements, a specific scale-in/scale-out procedure has to be decided and executed, thus the inclusion of a service orchestrator as part of the service management functionality.

Scalability of the IMS components was already addressed by 3GPP during standardization [4]. For each component some of the procedures on how it can be distributed are described, and then how every component can be discovered, shortly introduced here:

- **P-CSCF distribution -** The UE queries the DNS for discovering the P-CSCF where it should send the subsequent messages. If one domain has more than one P-CSCF instances, meaning more than one host associated with that P-CSCF, then the DNS name server should select one of them. Once a P-CSCF has been selected the UE sends its subsequent messages to that entity. While serving UEs requests, the P-CSCF saves information locally. This means that the P-CSCF is a stateful component, so subsequent requests of the UEs need to be delivered to the component which should retrieve the state.
- **I-CSCF distribution -** The I-CSCF is a stateless component, its main task being to retrieve which S-CSCF should be assigned to the incoming request. When more than one I-CSCF are present in one domain, then it is possible to dynamically select one of them using DNS NAPTR.
- **S-CSCF distribution** – The S-CSCF selection is done by the I-CSCF. In particular, the I-CSCF acts as a load balancer, having all the algorithms and mechanisms for assigning a particular S-CSCF instance to a EU. It is called "S-CSCF selection procedure", the I-CSCF sends a Diameter request to the HSS, and it returns the capabilities in its response. EU. It is then the task of the I-CSCF to select a specific S-CSCF that satisfies the capabilities required for that particular EU.
- **HSS distribution –** The HSS offers a Diameter interface. For this reason, having multiple HSS in one domain requires the presence of the SLF component.
- **SLF distribution -** As the I-CSCF, the SLF is a stateless component. Scaling a stateless entity requires only DNS load balancing mechanism.

Depending on the software of the IMS instance, the different functions may be split and grouped as required by the specific instance. Three deployment options are presented next, following these principles:

- Virtualized-IMS (vIMS) – an architecture option in which each 3GPP IMS functional entity is mapped 1:1 with the virtual network function units;
- Split-IMS – an architecture option in which each 3GPP IMS component is split into multiple sub-components in order to be deployed on top of multiple hosts and containers;
- Merge-IMS – an architecture option in which components are merged into less components enabling a low delay and functional reduced processing for external requests by a single same virtual network function unit.

Other architecture options, which combine the Split-IMS with the Merge-IMS, are possible for example merging all the IMS components into a single one and then splitting it according to the underneath description.

1) Virtualized-IMS

The first architecture option requires the implementation of each 3GPP IMS functional entity on a single virtual machine. In particular this is the main architectural approach followed while virtualizing network components. In this approach the interfaces with the external components are not changed and are the ones standardized by 3GPP [4]. Elasticity is realized using the procedures already standardized by 3GPP. Those are already described in the previous section.

2) Split-IMS

Figure **2** shows a possible architecture for the virtualization and distribution of network functions. A Network Function Balancer (NFBalancer) is the entry point for the incoming requests. The NFBalancer distributes the requests to multiple stateless NFWorkers. The state of the subscriber is maintained in an external functional entity named SharedMemory.

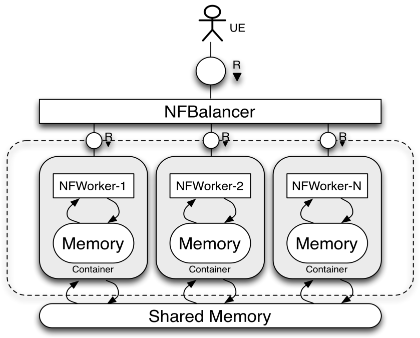

Figure 2. Split-IMS Architecture Concept

Only the interface between the user and the NFBalancer is standardized while the other interfaces are specific for each practical realization and does not affect interoperability. Between the NFBalancer and the NFWorker and between the NFWorker and the SharedMemory any kind of interface can be used. Having this three levels separation, the following advantages are foreseen: reduced complexity of load balancers; large number of stateless components (NFBalancers and NFWorker) enabling horizontal stability; move the state to the external Shared Memory enables a high elasticity of the procedure executing functions; isolation of the NFWorkers into a private, dedicated pool.

This kind of configuration can be used for deploying a single network entity in a single datacenter. In particular, there is no need to have a balancing in front of the I-CSCF, because the I-CSCF is stateless and its operations have a very short duration. So from this perspective, the I-CSCF component is not statically allocated to the subscribers and a next free one may be selected from the pool whenever needed. The selection from the P-CSCF or the S-CSCF acts as a Balancer. For the CSCFs functions it is needed a specific Balancer component in order to distribute the requests. Indeed the Mw interface is implemented using the SIP protocol, so the Balancer should be able to offer the SIP interface and distribute the SIP requests to the available Workers from the pool.

The UE discovers the PCSCF-Balancer via DNS queries. The PCSCF-Balancer will then redirect the requests to one of the PCSCF-Workers.

3) Merge-IMS

The last architectural model describes the possibility of merging together different IMS functional entities in order to reduce the complexity, making it a very simple architecture. In particular this virtual machine is called IMS VM, as depicted in Figure 3, and contains the four main entities: P-CSCF, S-CSCF, I-CSCF and HSS.

Having each subscriber on a dedicated instance allows reducing the number of exchanged messages among those components. Figure 3 shows the full architecture for the merge-IMS scenario.

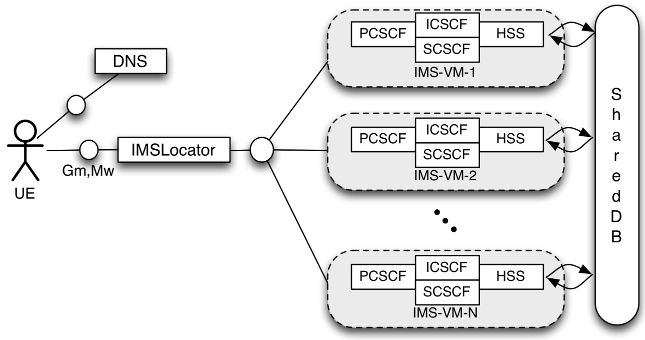

Figure 3. Merge-IMS architecture

It is possible to notice that there are three main entities involved:

- IMSLocator: this is an entity required for assigning the subscribers to a specific IMS-VM instance during the registration procedure, and localizing the IMS-VM instance during a phase of discovery. This entity should expose the Gm and Mw interface as standardized by 3GPP [4];
- IMS-VM pool: this is a pool of IMS-VM instances. It is managed by the service orchestrator, which decides whether to instantiate new IMS-VMs based on some conditions defined by the EEU;
- SharedDB: a shared database among the different IMS VM instances. It is required for storing information about the subscribers.

In this architecture, the IMSLocator is acting as a simple proxy, without changing the content of the message, only adding a SIP Via [5] to add itself on the SIP routing path of the answers between the IMS-VM and the UE. The IMSLocator exposes also the Mw for allowing the P-CSCF from the originating domain to contact the I-CSCF of the terminating network, and interoperating with legacy IMS networks Developing such architecture allows achieving the highest possible optimization as a software realization of IMS. However, it has to be considered that it reduces the flexibility in particular for the runtime and management phase.

For this reason it has been proposed also an architecture in the context of the Mobile Cloud Networking (MCN) project [14] for managing the lifecycle of those deployment models, reducing the complexity in the provisioning and runtime phases. The architecture proposed in Fig. 4.

In order to be able to provide cloudified IMS infrastructures as a service, a virtual service provider deploys in its network a Service Catalog describing the services offered as well as a tenant credential management system and an AAA server for the virtual IMS administration. These components along with the Service Manager through which requests for new virtualized service requests are transmitted form the front-end of the virtualisation service enabler.

The Cloud Controller is the main broker between the cloud infrastructure and the service provider similar to OpenStack Heat [15], enabling the realisation of virtual infrastructures including the deployment of the virtual networks, the connection to external entities such as UEs and Application Servers, the virtual machines and the dependencies between them.

In order to be able to flexibly configure the vIMS components a novel Service Orchestrator (SO) is proposed, separated into an Decision entity (SO-D) executing a policy based, trigger-condition-action matching, and an Execution entity (SO-E) orchestrating the service deployment and runtime management through communication with CC and the Element Management System (EMS). The triggers of the Service Instance (SI) are received from the Support Services such as monitoring and analytics, and used by the SO-D module for selecting and performing the proper actions for solving the related issue.

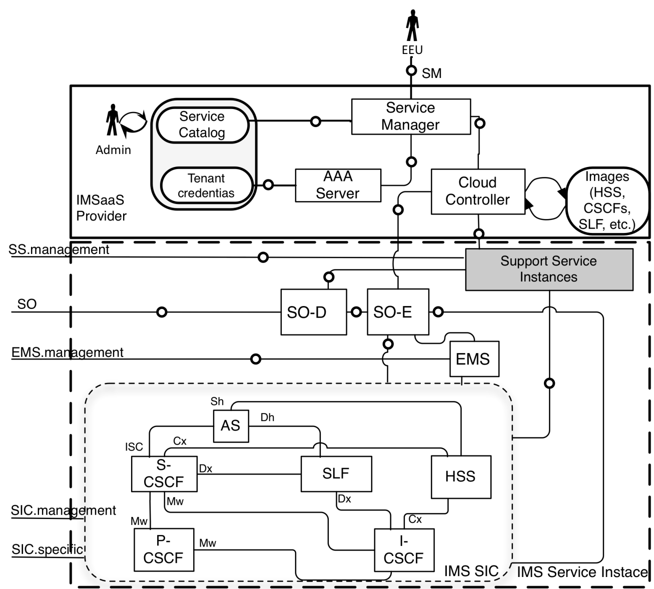

Figure 4. Mobile Cloud Networking vIMS Architecture

IV. EVALUATION OF VIRTUALIZED-IMS ARCHITECTURE

For evaluating the feasibility of the proposed architecture, we have used OpenStack [6] and the Fraunhofer FOKUS OpenSDNCore Orchestrator [16] for the instantiation of the different components, while as IMS infrastructure we have used the open-source Fraunhofer FOKUS OpenIMSCore [17]. In order to evaluate the performance of the virtualized-IMS we employed the IMSBench tool [17].

Fig. 5 shows the architecture of the testbed. In particular all the network functions of the vIMS are deployed and managed by the OpenSDNCore Orchestrator. The vIMS entities are deployed with a small flavour, 1vCPU and 2GB of RAM.

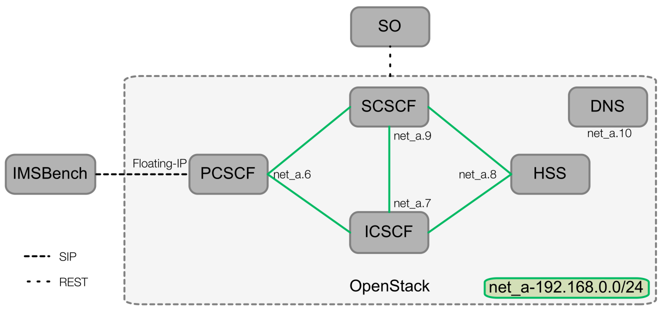

Figure 5. Testbed Archtiecture

The test has been divided in three main phases: the first one regards the registration of a pool of subscribers, the second is just a stir phase for preparing the system before the third one which is performing the real benchmark of the SUT. In particular during the third step a mix of scenarios are evulated: registration, de-registration, call setup, messaging and presence. Table 1 details some values regarding the different steps.

Phase	Initial CPS	Number of Steps	Increase per step	Duration	IHS
Pre-reg	40	1	0	20000 max requests	10
Stir	5	3	5	100 s/steps	10
Bench	10	6	10	300 s/steps	5

Table 1. Details about the different phases

Fig. 6 shows a graph with the Scenario Attempts per Seconds (SAPS) as described in the previous table and the memory usage over time of the different components part of the vIMS.

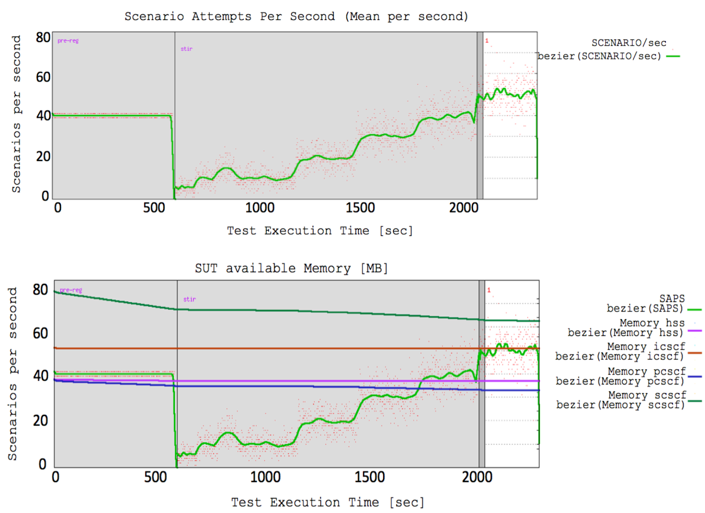

Figure 6. Scenario Attempts per Second (SAPS) and Memory Usage over time

Fig. 7 shows the CPU usage over time of the different components: during the pre-registration phase the most overloaded component is the HSS. It is mainly due to the high calculations required during the execution of the authorization and authentication phases.

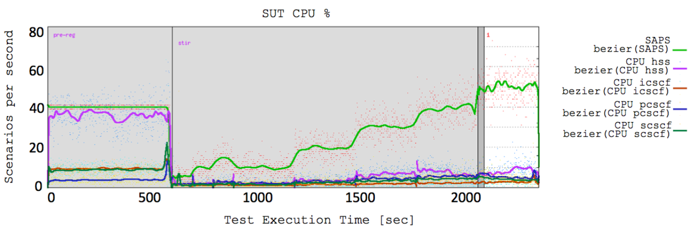

Figure 7. CPU usage over time

Fig. 8 shows the percentage of Inadequately Handled Scenario (IHS). It clearly depicts that this value increases when the SAPS reaches around 45 call per second.

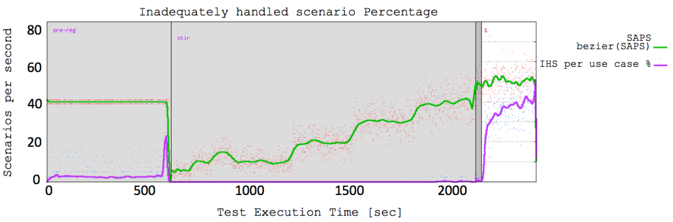

Figure 8. Inadequately Handled Scenario (IHS) percentage over time

This benchmark evaluation gives some performance indication about running a vIMS. In particular the test was failing around 50 Calls-Per-Second (CPS) because of the memory usage limitation of the SCSCF component. The CPS level achieved satisfies the complete IMS support for a current medium-size fixed or mobile operator need at average loads. However, for larger operators and for peak times the vIMS has to be further scaled.

Considering that the flavour used for this experiment is a small one, this value can be easily increased scaling vertically each of the components. Furthermore having a physical limit on scaling vertically, the best approach would be to use the Service Orchestrator for horizontally scaling the proposed vIMS, allowing also saving energy consumption and paying only for the used resources, as well as for offering geographically localized services with low delay for the subscribers.

V. Conclusion and Future Work

A possible architecture for offering IMS-as-a-Service has been proposed. It gives the possibility to deploy on-demand an instance of the IMS platform. In particular, following cloud principles, three different deployment models have been proposed for the implementation of a cloud-based IMS platform. The first model is the Virtualized-IMS that is an implementation 1:1 of the IMS platform as standardized by 3GPP. The Split-IMS architecture, in which each network function has been implemented using a three-tier model typical for web services. The Merge-IMS is the last architectural model proposed, in which the network functions are grouped together and offered on a subscriber-based way. In all of those models the scalability problem of the IMS infrastructure has been addressed. For instance, for the HSS has been proposed a separation between the front-end and the database, using cloud storage services, like the one provided by DBaaS [14].

Furthermore the vIMS platform has been integrated into the MCN platform, in order to interoperate with the SO for being offered as a service. This integration allows the combination among IMSaaS and other MCN services like for instance EPCaaS. A further step represents the development and the deployment of a merge-IMS infrastructure. As seen from the presented evaluation results, with a very limited number of entities a large number of subscribers can be served. We expect that a merge-IMS maintain this property while being able to serve the subscribers in a low delay localized fashion.

Acknowledgment

The authors are very grateful to Andy Edmonds (ZHAW) and Thijs Metsch (Intel) for providing their contribution on the MCN platform. The research leading to these results was partially funded by EU FP7 Mobile Cloud Networking project (FP7-ICT-318109).

References

[1] Continuity Central, “Five nines: chasing the dream?”, Nov. 2005, http://www.continuitycentral.com/feature0267.htm .

[2] P. Bellavista, K. Campowsky , G. Carella, L. Foschini, T. Magedanz, F. Schreiner, "QoS-aware elastic cloud brokering for IMS infrastructures", The Seventeenth IEEE Symposium on Computers and Communications (ISCC’12). July 1 - 4, 2012, Cappadocia, Turkey.

[3] K. Campowsky , G. Carella, T. Magedanz, F. Schreiner, "Optimization of Elastic Cloud Brokerage Mechanisms for Future Telecommunication Service Environments", Praxis der Informationsverarbeitung und Kommunikation. Volume 0, Issue 0, ISSN (Online) 1865-8342, ISSN (Print) 0930-5157, DOI: 10.1515/pik-2012-0036, June 2012

[4] 3GPP. TS 23.228. IP Multimedia Subsystem (IMS) .

[5] J. Rosenberg et al., “SIP: Session Initiation Protocol,” IETF RFC 3261

[6] Open source software for building private and public clouds: http://www.openstack.org/

[7] 3GPP TS 23.401, “General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access”, www.3gpp.org;

[8] Project Clearwater – IMS in the Cloud, http://www.projectclearwater.org/

[9] ETSI NFV Whitepaper Update, October 2013, http://portal.etsi.org/NFV/NFV_White_Paper2.pdf

[10] P. Bellavista, A. Corradi, L. Foschini, “Enhancing Intradomain Scalability of IMS-Based Services”, Parallel and Distributed Systems, IEEE Transactions on (Volume:24 , Issue: 12)

[11] T. Louvain-la-Neuve, “Migration of Mobicents Sip Servlets on a cloud platform.” Master Thesis, Universite catholique de Louvain, Louvain, School of Engineering, academic year 2010 – 2011, online: http://thibault.leruitte.name/tmp/cc04980d.pdf

[12] W. Iqbal, M. Dailey, and D. Carrera, “Sla-driven adaptive resource management for web applications on a heterogeneous compute cloud,” in Cloud Computing, ser. Lecture Notes in Computer Science, M. Jaatun, G. Zhao, and C. Rong, Eds. Springer Berlin / Heidelberg, 2009, vol. 5931, pp. 243–253.

[13] Y. Jie, Q. Jie, and L. Ying, “A profile-based approach to justin-time scalability for cloud applications,” sep. 2009, pp. 9–16

[14] T. Bohnert and A. Edmonds (eds.), *Overall Architecture Definition*, deliverable D2.2, MobileCloud Networking Project, Oct. 2013

[15] OpenStack Heat, http://wiki.openstack.org/wiki/Heat

[16] Fraunhofer FOKUS OpenSDNCore, http://www.opensdncore.org

[17] Fraunhofer FOKUS OpenIMSCore, www.openimscore.org

[18] IMS Bench SIPp, Open Source IMS benchmarking tool, http://www.sipp.sourceforge.net/ims_bench