Media Processing in the Future Internet

Markus Bauer, Stefanie Braun, Peter Domschitz
Bell Laboratories, Service Infrastructure Research Domain
Alcatel-Lucent, Stuttgart, Germany
{ Markus.Bauer, Stefanie.Braun, Peter.Domschitz } @ alcatel-lucent.com

I. INTRODUCTION

Handling the predicted growth of video and media traffic [1] is one of the key challenges future generation networks need to address. Basically this is not a new challenge, as in the past the Internet has already seen and handled enormous growth of traffic. And to a large extend this growth was driven by video centric services. One example for such a service is YouTube, which daily streams out 2 Billion videos to its customers, generating an overall traffic of about 25 PB per day. As a consequence, efficient video distribution is of big importance in today's networks. Up to now cache-assisted delivery schemes [2] enabled the networks to scale with the data traffic imposed by video centric services.

But there is more to video's future than efficient bulk distribution. Future video centric media services will be different. More and more people will not only consume, but also actively produce content. A lot of these services will be used by mobile users and personalization of content will be the common case. Such services ask for intermediate processing of media streams in the network. Interactive/real-time requirements must be met to achieve satisfying QoE for the users of such services.

Today's successful cache-assisted delivery schemes fail to serve personalized services which show such production, processing and individual delivery characteristics. In consequence, it is no longer possible to offload networks by simply caching content at appropriate locations in the network.

Instead, there is a need to acquire computing resources in the network to perform the required processing on the media streams. An approach, which places all required media processing functions in one or a limited number of centralized locations will not scale, as the networks will not be able to sustain all the media traffic that needs to be processed. [3]

The G-Lab project NETCOMP gives an answer to that challenge by offloading the (core) networks by means of localizing traffic. This is achieved by jointly optimizing networking and processing resources.

II. LATE BINDING OF PROCESSING RESOURCES

Future (immersive) media services require processing in the networks to build a personalized user experience from the media streams which compose the basis for the service. To keep traffic in the network local, media processing functions need to be placed on processing resources, which are located close to the media streams, crossing the network from media source or sink to the service consumer's point of attachment. In principle, the more processing resources are distributed across the network, the closer processing functions can be placed to the media streams and the more the network can be offloaded from avoidable traffic. But beneficial utilization of distributed processing resources requires a new approach how resources are selected. Today, placement of processing functions is an administrative task and the decision is performed prior to service runtime. Later, at service runtime any to be processed media stream is sent, independent from origin and destination, to the location where the processing functions have been placed prior to service runtime.

NETCOMP overcomes such inefficient utilization of transport resources by introducing the idea of late binding of processing resources. The location where media processing is executed is no longer decided prior to service runtime. Instead, this placement decision is delayed until educated selection of processing resources can be performed. Such informed selection is possible, when a media flows' minimal footprint (refers to the most efficient path) in the network is known. This is possible only at service runtime, when source and destination of the media flow are known. Having this knowledge, we can derive best fitting close by processing resources for performing required media processing on the media stream.

To implement this approach two main challenges need to be addressed: First, algorithms are needed, which derive best fitting processing resources for executing media processing functions that belong to a specific media stream. Second, as processing resources are dynamically selected at service runtime, instantiation of required media processing functions needs to be performed in way, which preserves satisfying QoE for the user of the service. In the following the second challenges will be discussed in more detail.

III. DYNAMIC INSTANTIATION OF MEDIA FUNCTIONS

Utilizing Cloud Technologies: MediaCloud

The concept of virtualization and its realization by means of virtual machines is the basis how today dynamic allocation of processing resources for execution of user defined processing functionalities is achieved. This section investigates what performance virtual machines can provide for dynamic instantiation of media processing functions on processing resources in the network. These investigations were performed by implementing and evaluating a personalized media centric use case on the MediaCloud framework utilizing the G-Lab experimental facilities.

Use Case: Personal View Generation

Interactive personalized services are composed of multiple media processing components and pose new challenges to today's virtualization technologies and media frameworks. Each media processing step requires decoding and encoding of the media content to guarantee efficient channel utilization. Successive de- and encoding steps impose additional processing overhead and delay on the media processing. And each additional coding step decreases the quality of the media content. An exemplary interactive media use case is built on the G-Lab platform to evaluate if today's virtualization technologies and media frameworks are ready for the challenges imposed by distributed media processing.

The personal view service provides a 360 degree live video, e.g., of a conference room. Each user gets his personal view into the conference. The user can change his view by rotating his mobile device. When the viewing angle of the mobile device changes the user's view into the conference room is adapted synchronously.

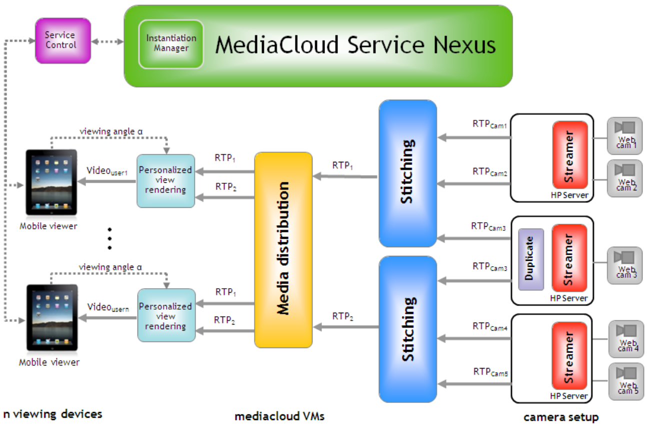

Figure 1. Personal View Generation

In the use case multiple cameras are placed in the centre of the conference room and provide live video content. The stitching component mixes these videos to a 360° video. In a monolithic design implementing that functionality in a single virtual machine the resulting coding delay of the complete 360° video would take a couple of seconds which is not acceptable for this kind of interactive service. That's why multiple stitching components are used to create smaller parts of the 360° video that can be encoded with an acceptable encoding delay. These are sent to the media distribution component. For each connected user a personal view renderer is started which computes the personal view for the user depending on the rotation angle of his mobile device.

The service control triggers the dynamic start of the required components by the MediaCloud service nexus and interconnects them. The service nexus places the components local to the media streams. E.g., the personalized view renderer is placed on a processing resource that is close to the user. The 360° video is multicasted to the access where the user specific renderer components are placed. This offloads the network and the short feedback loop between rendering function and mobile viewer guarantees a fast service response time.

Performance Evaluation

For the evaluation of the use case the MediaCloud was built upon KVM virtualization technology which is part of the Linux kernel. Each media component is started in an own virtual machine (VM) on the experimental facility. The use case shows that today's VMs impose huge overhead on media applications. Each VM contains a complete operating system with many processes and other functions, e.g. printer support, not relevant for media processing. So it takes several minutes to startup a VM. This is not acceptable for real-time services.

Optimization

Tests in the evaluation environment show that the VM startup time is influenced by two factors: copying the disk image to the selected processing resource and the boot time of the operation system (OS). A fast startup approach for VMs was developed and tested that brings down the VM startup time to several seconds instead of minutes. To ensure fast access to the disk images these are distributed to each MediaCloud location by means of a NFS server. Instead of creating a new disk image per VM, MediaCloud takes advantage of template images that are shared between identical components in one location. Only read access is allowed on the disk image to avoid corrupted data in case of concurrent VMs sharing the same disk image. Instead, permanent changes to the image are stored in temporary files local on the physical host on which the VM runs. Hence, disk image copy processes can be completely avoided. Furthermore, the OS boot time is reduced by suspending a VM prior to its distribution. This results in a fast VM start that requires only 4 to 5 seconds until the wake up process is completed and the VM has received a new network address. But real-time interactive media applications require startup times within milliseconds. Even with the optimizations introduced here, this cannot be achieved by means of today's cloud virtualization technologies.

IV. Conclusion And Outlook

Future real-time media services demand distributed execution at disperse cloud processing locations to meet mission critical QoE expectations and limit bandwidth requirements.

Media frameworks for executing flow oriented media services are available today. But those lack support for distributed service deployments and fail to provide required execution performance, efficiency and scalability. To overcome those limitations we have to challenge fundamental pillars of today's (cloud) technologies by inventing systems software specifically designed for extreme efficient execution of interactive media services on network distributed computing resources. Furthermore, we are working on algorithms for the on the fly (re-)placement of fine-grained processing tasks to satisfy expected QoE and offload networks by selecting resources at appropriate locations.

[1] "Cisco Visual Networking Index (VN): Forecast and Methodology, 2010-2015".

[2] "Content Networking: Architecture, Protocols, and Practice", by Markus Hofmann and Leland Beaumont, Morgan Kaufmann, February 2005, ISBN 1-55860-834-6

[3] Peter Domschitz, Markus Bauer, Jürgen Sienel and Marcus Kessler: "Move Apps not Data – A new Paradigm for the Future Internet" Proc. of the 10th Wuerzburg Workshop on IP: 'Visions of Future Generation Networks' (EuroView 2010).

This work is co-funded by the German Federal Ministry of Education and Research (BMBF) within the G-Lab project NETCOMP (FKZ 01BK0940).