FIRST@PC MediaX: A Service-Oriented Testbed for Realistic Media Networking Experiments

Sang Woo Han, Bum Hyun Baek, and JongWon Kim

Networked Media Lab.,
Dept. of Information and Communications,
Gwangju Institute of Science and Technology (GIST),
261 Cheomdan-gwagiro, Buk-gu, Gwangju, 500-712, Korea
{swhan,bhbaek,jongwon}@gist.ac.kr

Abstract. As an effort to devise and experiment diverse types of media-oriented service compositions supported by Future Internet infrastructure, this paper introduces an attempt to build a service-oriented testbed named as FIRST@PC (Future Internet Research on Sustainable Testbed based on PC) MediaX (Media eXperiment). Following the SOA (service oriented architecture) paradigm, FIRST@PC MediaX targets a flexible and cost-effective testing environment where media-oriented service compositions are flexibly realized on top of virtualized computing/networking resources. In this paper we will discuss on-going efforts on designing and building this testbed with several PC-based devices for media acquisition, media processing, display (networked tiled display), and networking. Specially, the preliminary implementation of agent-based software toolkit called as OMX (Open Media eXperiment for service composition) is explained and verified by testing a HD-media service scenario that combines multiple HD videos.

Keywords: Service-oriented testbed, service composition tool, media services, service composition experiment, and Future Internet.

1 Introduction

Ever-increasing demand for immersive media contents has raised flexible multimedia systems that understand widely dispersed media contents/tools and dynamically compose media-centric services with diverse computing/networking environments [1]. To realize the vision, several work accommodated the fundamental basis of SOA (service oriented architecture) [2] in building large-scale multimedia applications based on media-oriented service composition [3]. This paradigm made a broad impact to multimedia communities, which led to move from monolithic multimedia applications to more flexible component-based ones. Note that the service composition in SOA decomposes complex tasks into smaller independent entities (e.g., Web services), and then supports a flexible service composition in a variety of ways. Unlike the Web service composition, a

T. Magedanz et al. (Eds.): TridentCom 2010, LNICST 46, pp. 176–185, 2011.

composed media service should handle the support challenge of networking continuous data flows (e.g., audio/video streams) that have strict restrictions on timing and resources [3].

Testbeds are considered as a key tool to test semantical correctness and basic functioning of a new technological idea. These testbeds are running with a control framework that controls and manages testbed resources for user experiments. Testbeds additionally provide experimenters with tools and methods to build and execute their own experiments. For example, GENI Future Internet testbed effort [4] develops a suite of network research infrastructure with virtualized resource substrate, which encourages experimenters to easily develop new protocols and services using open, large-scale, and realistic experimental facility. PlanetLab, ProtoGENI, ORCA, and OMF (cOntrol and Management Framework) are current candidates of GENI control framework.

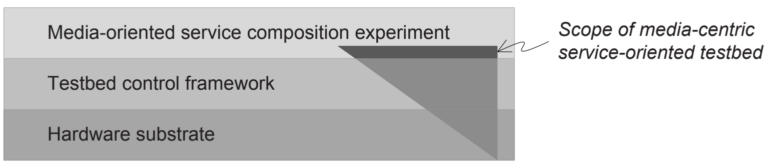


Fig. 1. Technical focus of service-oriented testbed

To facilitate the diverse experimental needs for media-centric service composition, as shown in Fig. 1, in this paper, we focus on a media-centric service-oriented testbed that carries out media-centric service composition experiments, by leveraging the support of testbed control framework (e.g., from GENI) and by specializing on tools and hardware for massive media processing and delivery. Like this, in order to flexibly devise and build new media-centric services, we attempt to build a service-oriented testbed named as FIRST@PC (Future Internet Research on Sustainable Testbed based on PC) MediaX (Media eXperiment)[1]. With this experimental testbed, we hope to incubate innovative and creative ideas for futuristic media-centric services. Key challenges are in extending existing testbeds further to support complicated service composition and in stably operating the developed testbed through new control and management tools for media-centric service composition, albeit with following features:

- *Flexible service composition:* Experimenters should be able to flexibly compose services based on the functional service dependency relationship among component services;
- *Dynamic service adaptation combined with monitoring:* The composed service should be able to adapt itself according to the monitored service status and resource utilization.

[1] In [5], a very early design for FIRST@PC testbed is presented by combining hardware-accelerated programmable networking [6] and virtualization [7] to support the experiments on media-oriented service composition.

The rest of this paper is organized as follows. Section 2 introduces basic concept of FIRST@PC testbed and key building blocks. Section 3 describes the building progress about FIRST@PC MediaX testbed with special attention to OMX (Open Media eXperiment for service composition) toolkit. After explaining the HD-media service composition experiment for verification in Sec. 4, we conclude this paper in Sec. 5.

2 FIRST@PC: Media-Centric Service-Oriented Testbed

In this section, we explain the basic idea about FIRST@PC testbed from conceptual illustration to key building blocks.

2.1 Testbed Conceptual Design

A conceptual illustration of FIRST@PC testbed (in short TB) is depicted in Fig. 2. Administrators and experimenters are accessing the computing/networking resource substrates in an aggregated form called *resource aggregation (RA)*. The controlled access to resource substrates are managed by *slice*

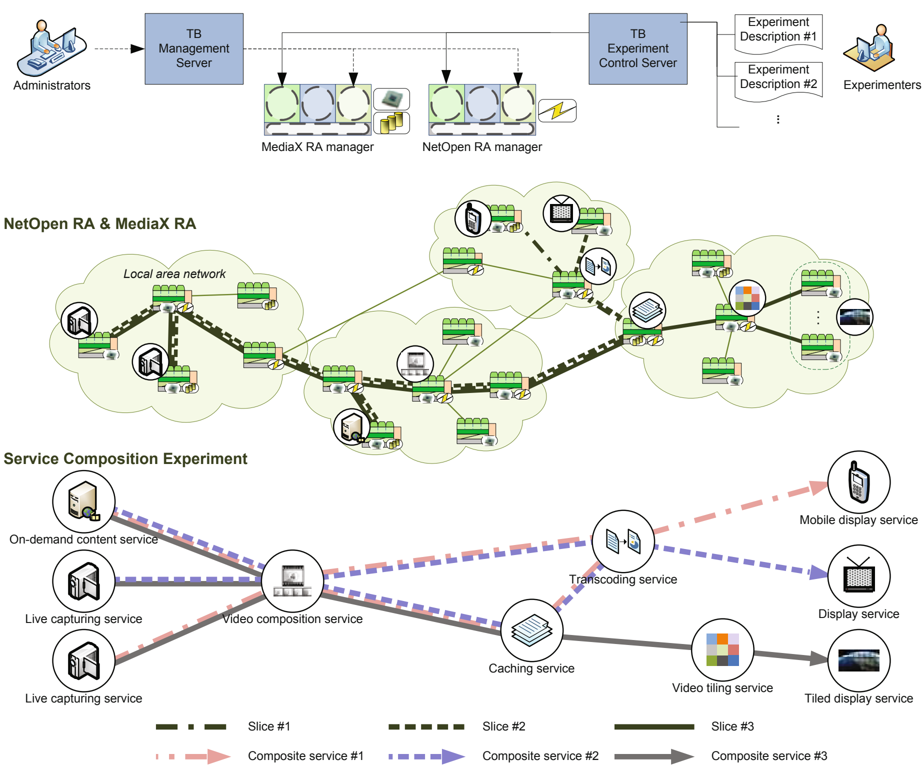


Fig. 2. A conceptual illustration of FIRST@PC testbed

control, which is actually coordinated among the corresponding RA managers, TB management server, and TB experiment control server. Two kinds of RAs are assumed: MediaX (for computing and storage) RA represented by two small CPU and storage icons and NetOpen (for networking) RA represented by communication link icon. *MediaX RA* prepares high-performance computing and GPU-based media processing acceleration capability with high-volume storage for diverse media-centric service composition experiment. *NetOpen RA* supports OpenFlow-compatible [8] hardware-accelerated (e.g., NetFPGA [6]) programable and virtualized networking capability. To make a service composition, an end-to-end slice is created where multiple nodes in both MediaX and NetOpen RAs are connected in a service path. Note that, in Fig. 2, three slices #1, #2, and #3 are illustrated to depict three connected composite services in a Internet-based broadcasting scenario.

2.2 Testbed Key Building Blocks

FIRST@PC testbed mainly comprises of two key building blocks: RAs, and coordination servers (RA managers, TB management server, and TB experiment control server).

Each *RA (resource aggregation)* represents a group of PC-based computing and/or networking resource substrates. Typical resource substrates include computing (e.g., CPU and GPU power), storage (e.g., memory and disk array), and networking (e.g., NetFPGA and wireless network interfaces) resources. A node should support *virtual nodes* that can take utilize selected portion of virtualized (hopefully isolated) resources. Note that virtual nodes are can be associated either with slice control for experiments or resource management for TB operation & management.

Among the coordination server, first, each RA manager manages his RA by allocating resources on behalf of individual nodes and helping the configuration of RA nodes. It also supports open but authorized access for resources according to the privilege of experimenters and administrators. The TB management server is responsible for operating and managing TB by involving with slice control, resource management, and resource monitoring. Finally, TB experiment control server enables experimenters to describe service composition experiments and supports service control and its status monitoring.

3 Preparing FIRST@PC MediaX Testbed for Experiments

In this paper, we focus on running experiments on media-centric service composition only with MediaX RA, temporarily setting aside the programmable and virtualized networking capability of NetOpen RA. After explaining how to conduct the media-centric service composition experiments, we discuss on-going realization of FIRST@PC MediaX testbed with special emphasis on the agent-based OMX toolkit for service composition experiments.

3.1 Target Service Composition Experiments

With the FIRST@PC MediaX testbed, we are currently working to realize the experiment on service mapping coordination based on a template-matching approach, as investigated in SpiderNet [9] and SeSCo [10]. Generally, for the media-centric service composition, the input media sources that end users want to receive are going through composition processes to be displayed in various user-defined presentation formats (e.g., layout including size and resolution, and visual effects). The service composition experiment matches component services with available resources of appropriate QoS characteristics (e.g., delay, jitter, loss, and playout continuity). This process is conducted by connecting component services into a composite service according to the service dependency graph and composition algorithm. The connection is actually made by binding the interfaces of component services together, which is equal to making an end-to-end slice for service composition.

As the first target experiment of FIRST@PC, we are considering a personalized and interactive broadcasting scenario. As depicted in Fig. 2, we define two underlying networks specialized for media production and distribution. Future media production enables real-time and online distributed video editing, where numerous media sources (e.g., live 4K video and panoramic multiple HD video) are dynamically integrated together. These integrated media streams are converted (with the aid of transcoding services and media upscaling/downscaling services) and delivered (e.g., multicasting services) to match the target end systems. Our target experiment assume that such heterogenous display devices receive common contents from live capturing service(s) and/or an on-demand content service. Three composite services are defined for mobile display, TV, and tiled display, respectively. Especially, for the composite service #3, the most powerful tiled display can service multiple media contents by selectively binding on-demand content service, live capturing service, video editing service, caching service, video tiling (partitioning into several tiles) service, and networked tiled display service.

These composite services, as a connected set of composable component services, are represented by a directed acyclic service dependency graph to express the functional dependencies. To run the service composition experiments, we first need to describe the targeted service composition with the service dependency graph. For this, it is required to clarify the relationship between the component services and used resources. However, since this clarification is not an easy task, the experimenter may rely on his interpretation and interactively guide the service composition. To avoid the manual interactive composition, we can upgrade the experiment to adopt a composition algorithm to automatically coordinate the service mapping between the service dependency graphs and physical distributed component services. We may further extend to the dynamic composition experiments, where the dynamic change in the connection path is automatically negotiated to overcome the unexpected congestion of underlying network links. We may also experiment the impact of load balancing method that evenly distributes total resources for dynamic composition of multiple media distribution.

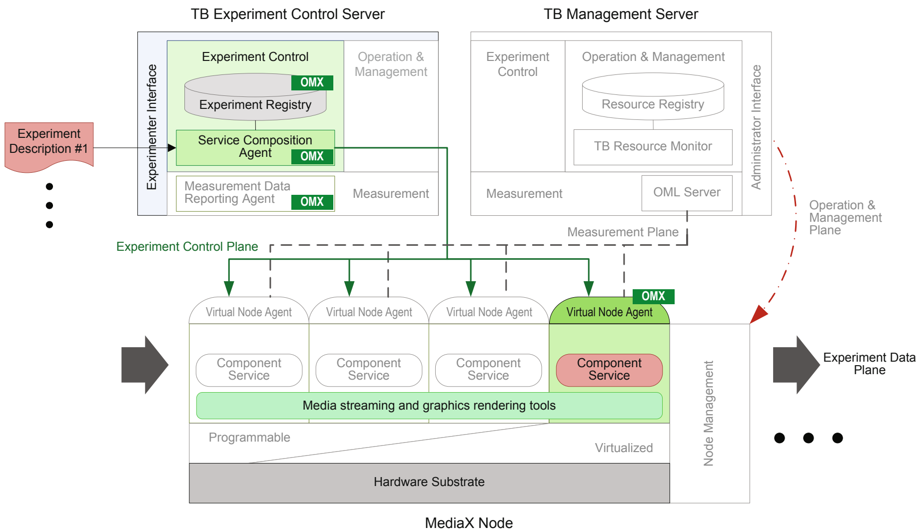


Fig. 3. FIRST@PC MediaX architecture (tentative)

3.2 FIRST@PC MediaX Architecture

Fig. 3 illustrates FIRST@PC MediaX architecture[2]. Experimenters formally specify the targeted service composition with quantitative requirements such as required component services and their configuration parameters (with numerical permissible ranges). They also describe adaptation rules to dynamically change composite services (e.g., service migration) according to events. The TB experiment control server assists the experimenters by feeding registry information about the capabilities of MediaX RA nodes to estimate the capabilities of instantiated component services. Measurement reports collected from real-time computing/networking performance data are used to configure component services and later enforce adaptation rules. On the other side, the TB management server enables administrators to register MediaX RA nodes and periodically monitor their status. We currently use OMF [11] for TB management and manage resources (e.g., resetting nodes, retrieving node status information, and installing new OS images). We also use the companion OML (Orbit Measurement Library) [12] to monitor TB resources (i.e., gather CPU/memory usage and the IO traffic amount).

To efficiently support multiple concurrent experiments, MediaX RA nodes need to provide virtualization of computing/networking resources. We are working on OpenVZ container-based virtualization that can virtualize resources of a common operating system for multiple virtual nodes. In a node, the node management governs the allocation of resources to virtual nodes. In each virtual

[2] Blocks in gray text in white background are not yet fully implemented. Also, blocks tagged with OMX belong to OMX toolkit, explained in Section 3.3.

node, an agent starts and stops component services and monitors its current status (e.g., inactivated, activated, or defected). Utilizing the allocated (virtualized) resources, component services provide elementary media functions and they are connected by taking the order from the TB experiment control server.

3.3 OMX Toolkit for Service Composition

To assist the composition experiment, the OMX (Open Media eXperiment for service composition) toolkit provides software agents (i.e., including experiment control agent, node agent, and service agent) with experimenter interfaces. Currently, the OMX toolkit (version 0.1) manually connects component services according to the service path[3] drawn by experimenters. To implement OMX toolkit, we use JADE (Java Agent DEvelopment framework), a FIPA-compliant multi-agent middleware.

The service composition description, written in XML, includes <name> representing the service identifier, <max_instances> representing the permissible number of the service instances to be executed, and <control_interfaces> specifying control interfaces (e.g., shell command) and their input parameters. For each virtual node, experimenters write Java code to describe their experiments by linking with the OMX agent that implements callback methods such as `start`, `stop`, `serviceLinkAdded` (when the service to be connected is determined), and `serviceLinkRemoved` (when the service to be disconnected is determined). Also, the OMX toolkit provide an experimenter UI (user interface) to facilitate the service composition experiment. The experimenter UI shows all available nodes and component services supported in each node. To draw a service dependency graph, an experimenter selects nodes by drag-dropping it from the node list, chooses a component service to be run, and connects a node with another.

4 Service Composition Experiment with FIRST@PC MediaX Testbed

4.1 Testbed Setup

FIRST@PC MediaX testbed, depicted in Fig. 4, includes MediaX nodes categorized into three different purposes: media servers, adaptors, and a networked tiled display. Media servers feed MPEG2-encoded live video. Adaptors do real-time media processing with GPU-based computing assist. The networked tiled display realizes ultra-high resolution by tiling multiple LCD displays. Most MediaX nodes are connected via 1Gbps LAN while some are connected via 10Gbps to handle multiple uncompressed HD videos.

4.2 Service Composition Experiment

A service composition experiment for sharing HD-media and desktop screen is specified by a service dependency graph depicted in Fig. 5(a). Live capturing

[3] The service path, a special case of a service dependency graph, shows how all component services are connected into a single successive end-to-end chain.

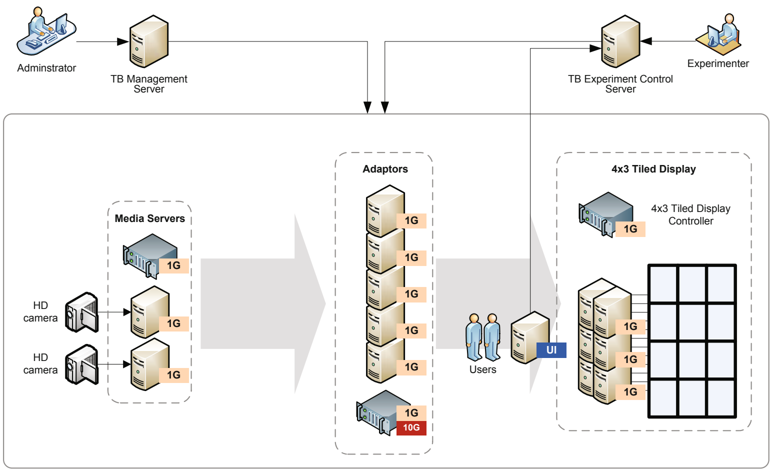
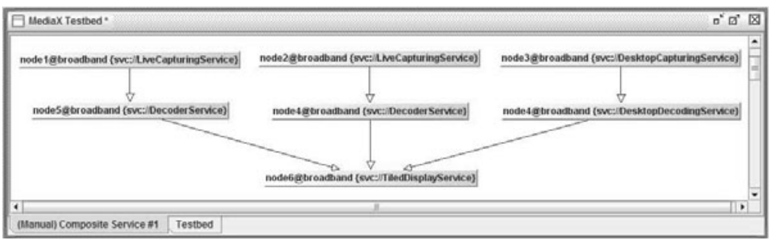
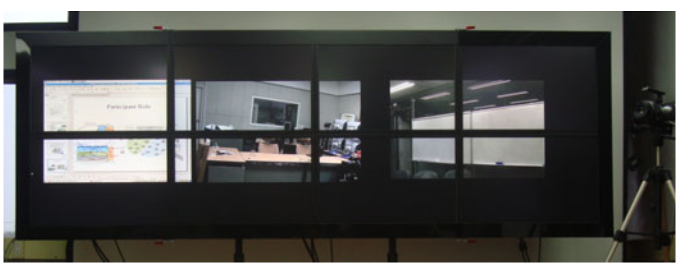


Fig. 4. Deployment diagram of FIRST@PC MediaX testbed



(a) Service graph drawn in the OMX toolkit interface.



(b) HD videos and shared desktop screen rendered.

Fig. 5. Experimental results of multiple HD video service composition

services acquire HD live videos and delivers the MPEG2-TS video stream to corresponding decoding services (realized by a VLC media player). Decoded uncompressed video are then sent to the networked tiled display. Also, an interactive graphics producer service captures desktop screen and transports graphic

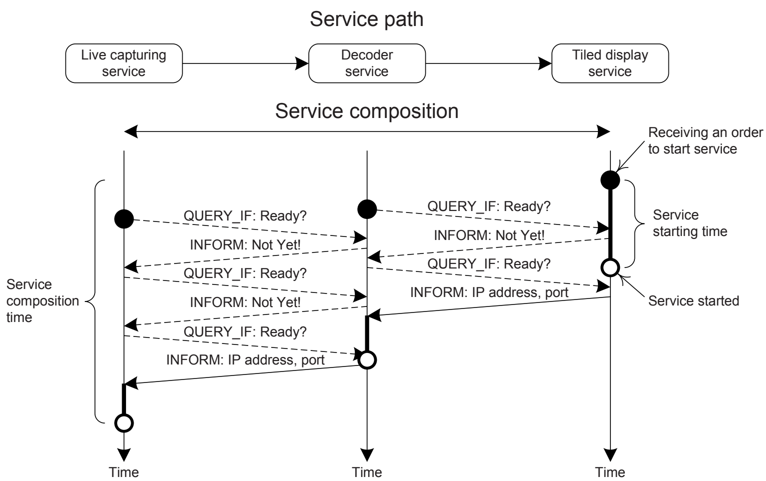


Fig. 6. Agent communication messages exchanged between service agents

streams (using TightVNC). An interactive graphics consumer service receives graphic streams and converts them for networked tiled display. All these composite service results are depicted in Fig. 5(b).

Fig. 6 illustrates the communication messages exchanged between the agents for involved component services. Composition is performed in backward sequential order from destination to source services along the service path. When a component service is asked to start, it consumes time to initialize its function. While the tiled display service prepares the service execution, the decoder service periodically sends QUERY_IF messages to inquire about the service status. The tiled display returns INFORM message with its IP address and port number when it is started. The decoder service prepares streaming to the IP address and port number. Similar procedures are repeated between the live capturing service and the decoder service.

5 Conclusion

An on-going effort to design media-centric service-oriented testbed called FIRST@PC MediaX was described where media-oriented service compositions are flexibly realized on top of virtualized computing/networking resources. Current design and realization on top of several PC-based devices for media acquisition, media processing, display (networked tiled display), and networking were extensively explained with the preliminary experimental testing based on the OMX toolkit.

Acknowledgments. This paper is one of results from the project (2009-F-050-01), "Development of the core technology and virtualized programmable platform for Future Internet" that is sponsored by MKE and KCC.

References

1. Daras, P., et al.: Why do we need a content-centric Future Internet? Proposals towards content-centric internet architectures. European Commission, Networked Media Unit, Information Society and Media (May 2009)
2. Huhns, M., Singh, M.: Service-oriented computing: key concepts and principles. IEEE Internet Computing 9(1), 75–81 (2005)
3. Nahrstedt, K., Balke, W.T.: Towards building large scale multimedia systems and applications: challenges and status. In: Proc. of Int. Workshop on Multimedia Service Composition (November 2005)
4. GENI Project Office: GENI System Overview. GPO Documents, GENI-SE-SY-SO-02.0 (September 2008)
5. Han, S.W., Kim, N., Kim, J.: Designing a virtualized testbed for dynamic multimedia service composition. In: Proc. Int. Conf. on Future Internet Technologies (CFI 2009), Seoul, Korea (June 2009)
6. Gibb, G., Lockwood, J., Naous, J., Hartke, P., McKeown, N.: NetFPGA–An open platform for teaching how to build Gigabit-rate network switches and routers. IEEE Trans. on Education 51(3), 364–369 (2008)
7. Bhatia, S., Motiwala, M., Muhlbauer, W., Mundada, Y., Valancius, V., Bavier, A., Feamster, N., Peterson, L., Rexford, J.: Trellis: A platform for building flexible, fast virtual networks on commodity hardware. In: Proc. ROADS 2008 (December 2008)
8. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: OpenFlow: enabling innovation in campus networks. ACM Computer Communication Review 38(2), 69–74 (2008)
9. Gu, X., Nahrstedt, K.: Distributed multimedia service composition with statistical QoS assurances. IEEE Trans. on Multimedia 8(1), 141–151 (2006)
10. Kalasapur, S., Kumar, M., Shirazi, B.: Dynamic service composition in pervasive computing. IEEE Trans. on Parallel and Distributed Systems 18(7), 907–918 (2007)
11. Rakotoarivelo, T., Ott, M., Jourjon, G., Seskar, I.: OMF: a control and management framework for networking testbeds. In: Proc. of Int. Workshop on Real Overlays and Distributed Systems (ROADS 2009), Big Sky, MT (October 2009)
12. Singh, M., Ott, M., Seskar, I., Kamat, P.: ORBIT Measurements framework and library (OML): motivations, implementation and features. In: Prof. Tridentcom 2005 (February 2005)