Understanding Resource-aware Control of Media-centric Service Composition over Future Internet Testbeds

Sep. 12, 2012
Sang Woo Han, JongWon Kim
{swhan, jongwon}@nm.gist.ac.kr

Networked Computing Systems Lab.,
School of Information and Communications,
GIST (Gwangju Institute of Science and Technology), KOREA
Resource-aware media-centric service composition over Future Internet

- Resource-aware control of each step in the service composition process in interaction with resources
- Monitoring and management services that assist in handling of versatile (i.e., virtualized and programmable) computing/networking resources
Media-centric service composition over Future Internet testbeds

Local area network

- $S_1$: Live content streaming service
- $S_2$: On-demand content service
- $S_3$: Video composition service
- $S_4$: Caching service
- $S_5$: Multicast service
- $S_6$: Transcoding service
- $S_7$: Display service
- $S_8$: Video tiling service
- $S_9$: Tiled display service

Resource:
- Storage capability
- Networking capability
- Computing capability

Media stream
Motivation

- Experimenters desire to **create an assemble of composable services**, optimized in efficiently utilizing (virtualized/programmable) computing/networking resources, where multiple media-centric (i.e., retrieval, transcoding, display services) services are stitched via control and data dependencies.
- One of the major problems with current media-centric service composition is that one has to plan the relationship among media services **without sufficient consideration on underlying infrastructure**.
- This problem makes it difficult to maintain media-centric service composition within an acceptable quality range, especially given fluctuations in resource availability and load.
- Existing Web service composition solutions provide little help in this regards.
- We need a better approach for **RESOURCE-AWARE** service composition, which gives knowledge about the relationship between the amount of resources and the quality of service composition and then guidance of adaptation behaviors in intelligent ways.
NGSON (c.f., Smart Network)

- NGSON provides service composition on overlay networks
  - End users and service nodes join and leave SON via service routers
  - Creating network paths are independent the type of networks (e.g., IMS, 3G, WiMAX)
  - Service-aware network paths are established with considering physical network status
  - P4P, IETF ALTO (Application-Layer Traffic Optimization)
X. Huang, S. Shanbhag, and T. Wolf, "Automated Service Composition and Routing in Networks with Data-Path Services," in Proc. of ICCCN 2010

- In-network processing
- I/O parameter based service composition
- Performance indices
  - Link delay
  - Service processing delay
Z. Zhao, P. Grosso, R. Koning, J. van der Ham, C. de Laat, "Network resource selection for data transfer processes in scientific workflows," in Proc. of WS. on Workflows in Support of Large-Scale Science 2010

- Semantic description
  - NDL (network description language) + CDL (CineGrid description language)
- Finds the optimal paths for digital media streaming
Proposed approach

• Describe **key elements** to clearly represent the resource-aware and media-centric service composition

• Define **a workflow-style service composition process** for maintaining resource-aware media-centric service composition within an acceptable operating range

• Has not practically verified and validated yet
Key element: Resource, Service, Content

• Resource
  – Node is a resource container that supports computation and networking resource capabilities via virtualization
  – Resource capability provides specific resource capabilities, characterized by its type and amount, that is a set of resources virtualized from dispersed nodes. It is exposed as an interface that is understandable by a service

• Service
  – has input and output interfaces and functions
  – needs resource primitive to obtain the desired resources

• Content
  – pre-stored or live media
  – has an unique name and a format
Key element: Service dependency graph

- The service dependency graph describes service composition (i.e., a chain of service instances) by representing the relationship between services.
Key element: Process

- **Service matchmaking** $P_{mm}$
  - looking for candidate services that meet the requirements of services included in a given service dependency graph

- **Service placement** $P_{pl}$
  - loading code of the candidate services, stored in repositories, on the chosen set of resources and creating service instances via execution

- **Service stitching** $P_{st}$
  - actually binding service instances according to the service dependency graph

- **Service monitoring** $P_{mon}$
  - observing measurement values from service instances and checking whether service composition is keep going well or not

- **Service tuning** $P_{tune}$
  - individually adjusting preset control parameters in order to remove performance bottlenecks of individual services and resources
An entity relationship diagram of key elements for resource-aware service composition

Diagram details:
- **P** (Process)
- **G** (Service dependency graph)
- **C** (Content)
- **S** (Service)
- **R** (Resource)
- Timely
- Resource efficiency
- Quality
- Compute
- Storage
- Bandwidth
- Programmable
- Virtualized
- Functional
- Stationary
A conceptual illustration on resource-aware service composition experiments

**Concept**
- Experiment description
  - Service descriptions
  - Content descriptions
  - Resource descriptions
  - Service dependency graphs
  - Process description
- Pool of contents
  - MediaX RA
  - NetOpen RA
  - FIRST Cloud RA
- Pool of services
  - Pool of resources

**Physical actions**
- Experiment input
  - Experimenter’s intension
  - $P_d$
- Experimenters
  - $R_d$
- Experiment output
  - Composite service by $G$
  - $\triangledown$ SP
  - $\triangledown$ TP
  - $\triangledown$ MP
- Experimenters
  - $S_{mp}$ $R_{mp}$

**Tools**
- Experiment description tool
  - $S$, $C$
- Resource provisioning tool
  - $G_d$, $P_d$, $S$, $C$
- Workflow engine
  - Interpretation based on $P_d$
  - $G_d$, $S$, $R$
  - $S_c$
  - $S_{mp}$, $R_{mp}$
- Experimenter’s status viewer
  - $S_{mp}$
- Mediator
  - Mediation tasks
  - $S$, $R$, $G_d$, $I$
  - $R_c$, $I_c$
### Design phase

e.g., An experiment description for HD video delivery & presence

<table>
<thead>
<tr>
<th>Content Set $C^{(des)}$</th>
<th>Full HD live videos $c_1, c_2, c_3$</th>
</tr>
</thead>
</table>
| Service Set $S^{(des)}$ | DXT producing services $s_{vp}^{c_1}, s_{vp}^{c_2}, s_{vp}^{c_3}$  
Networking service $s_{net}$  
Networked display service $s_{dp}$ |
| Resource Set $R^{(des)}$ | Media servers $dxt01.gist, dxt02.gist, dxt01.cnu, dxt01.khu$  
Networking resource $of.gist$  
Networked display resource $disp.gist$ |

**Service dependency Graph $G$**

**Process $P$**

![Graph showing service dependencies and process phases](image)

- DXT producing service
- Networking service
- Networked display service

### Table 1: Resource pool

<table>
<thead>
<tr>
<th>Number</th>
<th>Type</th>
<th>Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DXT</td>
<td>dxt01.gist</td>
</tr>
<tr>
<td>1</td>
<td>net</td>
<td>dxt01.khu</td>
</tr>
<tr>
<td>2</td>
<td>vp</td>
<td>dxt02.gist</td>
</tr>
<tr>
<td>3</td>
<td>disp</td>
<td>disp.gist</td>
</tr>
</tbody>
</table>

- $P_{mm}$
- $P_{pl}$
- $P_{st}$
- $P_{mon}$

[condition $c$ is true]
Provisioning phase

• For given $C_d$, $S_d$, $R_d$, an experimenter gets the authority to legally use $C$, $S$, $R$
  – Set of desired contents $C_d$, Set of desired services $S_d$, Set of desired resources $R_d$
  – Pool of contents $C$, Pool of services $S$, Pool of resources $R$

• Based on the resource description $R_d$ and the network topology, a network-stitched resource pool $R$ is provided by establishing physical network paths over the authorized resources.
Execution phase

- Service composition conducted by interpretation based on $P_d$
  - Input: $C, S, R, G_d$
    - Desired service dependency Graph $G_d$, Desired process $P_d$
  - Process: conduct service composition following given Process $P_d$
  - Output: Composite service CS
Resource-efficient service composition by mediator

- The mediator assists the interpretation box in conducting resource-efficient service composition
  - For given $S$, $R$, $G_d$, the mediator recommends a matching, named as candidate Service Instance $I_c$, that matches services and resources
  - At this stage, we define a tentative problem on resource-efficient service composition problem, but remain any mediation algorithms to solve the problem as future work

![Diagram](image)
A tentative resource-efficient media-centric service composition problem

• Assumption
  – Experimenters do not know the exact amount and type of resources services need

• Problem
  – For given $S$, $R$
  – Find an efficient matching between $S$ and $R$ so that $q(S, R)$, objective function to represent resource efficiency, comes close to 1
    • If $q(S, R)$ is 1, than it is ideal case! (the most efficient utilization)
    • If $q(S, R)$ is 0, than resources are overprovisioned

  $$q(S, R) = \prod_{s_i \in S} q(s_i, R)$$

  $$= \prod_{s_i \in S} \left( \alpha \frac{r^{(com)}(s_i)}{R^{(com)}_i} + (1 - \alpha) \frac{r^{(bw)}(s_i)}{R^{(net)}_i} \right)$$

• Notation
  – $r^{(com)}(S_i)$, $r^{(bw)}(S_i)$: the amount of computation and bandwidth resources consumed by $S_i$
  – $R^{(com)}_i$, $R^{(net)}_i$: the amount of computation and network resources allocated for $S_i$
  – $\alpha$: a real value [0, 1]
Service Matchmaking

\[ S_c = P_{mm}(G_d, S, R) \]

- Task \( P_{mm} \) looks for a syntactically compatible Set \( S_c \) of candidate services among services of Set \( S \).
- Syntactical compatibility is established when the name of service is the same as the service name specified in \( G_d \).
- It also checks whether a subset of given Set \( R \) can support resource types that Set \( S_c \) of candidate services demands.
Service Placement

\[ I = P_{pl}(S_c, R_c) \]

- Based on Set \( S_c \) of candidate services as the service matchmaking result, Task \( P_{pl} \) places Set \( S_c \) of candidate services on the chosen Set \( R_c \) of physical resources.
- Task \( P_{pl} \) launches the candidate services and creates service instances.
- At this stage, Set \( I \) of service instances are ready to run individually without interacting with the other service instances.
Mediation for Service Placement

\[ R_c = M_{pl}(S_c, R, R_{mp}) \]

• Choosing Set \( R_c \) is important because the matching between candidate service Set \( S_c \) and candidate resource Set \( R_c \) becomes crucial on the resource efficiency of service composition.

• So the experimenters can get any help with the mediator that automatically selects Set \( R_c \) (if possible, by considering resource availability \( R_{mp} \)).
Service Stitching

\[ G = P_{st}(G_d, I_c) \]

- Task \( P_{st} \) actually binds the stitching points of the chosen Set \( I_c \) of service instances to ones of their neighboring service instances.
  - Flows gets generated by service stitching
- Both service instances to be stitched know the addresses of their counter service instances and establish direct connections between them.
- \( G = \{(i_a, i_b) | i_a \in I_c, i_b \in I_c\} \) is a graph representing the relation on service instances as the result of service composition.
Mediation for Service Stitching

\[ I_c = M_{st}(G_d, I, I_{mp}, R_{mp}) \]

- To choose appropriate Set \( I_c \) regarding current status of service composition, the mediator executes Task \( M_{st} \).
- Task \( M_{st} \) changes links by considering Set \( I_{mp} \) of measurement values originated from Set \( I_c \) and Set \( R_{mp} \) of measurement values on resource availability.
Service Monitoring

\[ q = P_{\text{mon}}(G, l_{mp}, R_{mp}) \]

- Task \( P_{\text{mon}} \) periodically gathers the monitoring results \( l_{mp} \) and \( R_{mp} \), and calculate performance index \( q \).
  - Set \( l_{mp} \): measurement values originated from Set \( l_c \)
  - Set \( R_{mp} \): measurement values on resource availability

- Performance index \( q \) exerts a influence on decision whether to keep current service composition or to take appropriate actions to remedy problematic symptoms.

- This decision is depending on the performance index \( q \) reflecting resource efficiency of target service composition.
Future work

• To verify this conceptual approach, we need validating its feasibility (i.e., resource-efficient service composition) over Future Internet testbeds.

• In doing so, we will continue appropriate refinements of the proposed process, tailoring to the target experiment scenario.
BACKUP SLIDES
A brief review on service composition

• Web Service composition
  – A service composition is comprised of services that have been assembled to provide the functionality required to automate a specific task or process

• Multimedia service composition (via Service Overlay Network)
  – Controlling a composed flow for continuous, timely, quality-aware and synchronized media delivery
  – Coordinating different adaptive techniques in end-to-end composed services over heterogeneous devices and networks
Resource-aware media-centric service composition process

Control of service composition process

Resource awareness

Provisioning phase

Execution phase

Experiment description

Content descriptions

Service descriptions

Resource descriptions

Service composition description
(service dependency graphs, process)

Service matchmaking

Service placement

Service stitching

Service monitoring

Service tuning

Content selection

Service discovery

Node selection

Topology decision

Content inventory

Service inventory

Resource inventory

Authorized content pool

Authorized service pool

Authorized resource pool

Service matchmaking

Service placement

Service stitching

Service monitoring

Service tuning

Flows

Primitives

Slice

Host

Flow

Service

Content
Multiple options for media-centric service composition process

(a) Service matchmaking — Service placement — Service stitching — Service monitoring — Service tuning

(b) Changing the process — Enabling service migration

(c) Changing the service dependency graph

(d) Service discovery — Service matchmaking — Service placement — Service stitching — Service monitoring — Service tuning

- The timescale of monitoring
- The intensity level of tuning

Update the available service list