APPSAWG
Internet-Draft
Intended status: Informational
Expires: September 13, 2012

K. Li
Huawei Technologies

March 12, 2012

Virtualized Application: Problem Statement
draft-li-appsawg-virtualized-application-protocol-01

Abstract

Virtualized Application aims to enable the user device to remotely consume various applications on the network. This involves having all the virtualized applications hosted in the network and from there providing them to the users using cloud computing technologies like virtualization. This document tries to explain the problems to achieve the virtualized applications.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The basic idea of the Virtualized Application is that, the applicaion is running on the network server, the network server sends screen streams to the user device, the user can use a client on the device to view the screen streams. And the client can capture the user's interaction with the application and send the user interactivity to the network server. The network server renders user interactivity on the hosted application. In this way, the user can remotely consume various applications without installing the applciations on the user device.

Virtualized Application provides some application consumption models:

- OS-Independent applications: This enables users to use applications irrespective of the Operating System they are using. This will increase the application availability for users and vice versa.

- On-line applications: This allows users to use applications online instead of downloading them on to their devices. This will allow users to overcome terminal barriers (e.g memory) and accessing virtualized applications from any terminal at any time.

- No-loss reconnection: This enables users to reconnect to an abnormally suspended virtualized application session without application data being lost.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

1.2. Terminology and Abbreviation

- OS: Operation System

- VA: Virtual Application

- VAC: Virtual Application Client

- VM: Virtual Machine

2. Problem Statement

With the advancement of high profile applications (e.g games) and various available platforms, the service consumption is becoming complex and difficult. The number of devices available with different hardware and software specification is making things worse. Applications are being developed for a particular platform and with strict hardware and software requirements. These constraints mostly proved to be a hurdle for the complete value chain:

- Users can only use applications which are compliant to their device hardware and software platform specification.

- Content providers have to create multiple versions of the application depending on which hardware and software platform they want them to execute on.

- Despite of knowing this inconvenience of their users and partnered Content Provider, service providers can't do much to help their subscribers.

In an attempt to solve the problem, it is possible to optimize the current application usage model by providing a unified platform (cloud computing platform) which can host various applications, enabling different content and services, remotely in the cloud and provide them to the user using virtualization techniques (cloud computing).

This will aid end-user to use virtualized applications irrespective of the platform they are using with consistent user experience as compared to using applications hosted locally on the device. The user only needs to install Virtualized Application Client on his/her device. In addition to that, developers don't have to create a different version (for each mobile OS) of a single application, and service providers can offer a larger selection of applications and services to their end-users reducing costs.

Currently, most of existing Virtualized Application systems are based on proprietary implementation, and targeting different market with different features. Each of the current implementations provides bundle of components based on proprietary implementation, it's difficult to interwork between different vendors. Since virtualized application technology is believed that it will become a mainstream application delivery method, so it's important to make the virtualized application access protocol open and standardized.

3. Use Cases

3.1. Online Application Trials

This use case describes how a user can try-out the application remotely on the server-side without first downloading them on his/her device.

Note: There might be another use case when a user want to suspend and resume an application session voluntarily.

3.1.1. Short Description

Alice goes to a SP (App Store) searching for an application related with racing games. She found one application which cost about $20. Considering the cost Alice went into dilemma whether to buy that application or not. She found a button saying "On-line Trial", she clicked that button and a window popped-up on her mobile device in which the games got initialized instantly. She tried that game, played for about 10-15 minutes and decided to go for that game. She then went for the download/buy process.

3.1.2. Market benefits

Alice doesn't have to first choose, buy, download, install, run, play the game and then in case she doesn't like it uninstall, probably ask for refund and then again start from choosing another game. Instead, she can go for online trial, make up her mind and then buy and download the game.

3.2. OS-independent application

This use case describe how a user can use any application irrespective of the device platform he/she is using and the device platform the application is built for. This is achieved by hosting applications on the network-side and allowing users to use that application remotely from their terminal.

3.2.1. Short Description

Alice is having a device with Symbian OS. She goes to an App Store with an intention to buy an application. She liked an application but found that the application is for Andriod device only. She was also provided with an option to use that application from any OS i.e "Online Subscription". Alice found it feasible to go for this option and successfully subscribed for the "Online Subscription" for that application. Now Alice can use this application online as and when required irrespective of the OS that she is using.

3.2.2. Market benefits

User doesn't have to take their device platform in consideration while choosing a suitable application of their needs. Developers don't have to create different version (for each mobile OS) of a single application.

3.3. Application session suspend resume

This use case describes how a user can reconnect to the UVE enabled services in case of an unexpected connection failure. This also explains what user can get (or expect) at the reconnection.

3.3.1. Short Description

Alice is using a UVE enabled application. Alice is allowed to save the application state at any time.

At some point of time after getting disconnected, due to any reason (e.g bad network), Alice tries to reconnect with the application. At the reconnection, Alice is provided with an option to reload the application from one of the saved states. Alice then continues using the application from that state.

It is possible that Alice did not save the application. In this case, at reconnection, Alice continues using the application from the point where she was disconnected.

3.3.2. Market benefits

This will enable service continuity for users.

4. Architecture Overview

4.1. Architecture Diagram

The architecture diagram is shown below.

```
            -----------------                  --------------------
            |               |                  |                  |
            |  Virtualized  |    Virtualized   |                  |
            |  Application  | <--------------> | Virtual Machine  |
            |    Client     |    Application   |                  |
            |               |     Protocol     |                  |
            -----------------                  --------------------
```

Figure 1: Architecture Diagram

4.2. Virtualized Application Client

Virtualized Application Client is a device side component residing in the terminals enabling virtualized applications and utilizing virtualization technology to enable underlying operating system agnostic applications. It is mainly responsible for:

- Output rendering: On receiving Output Stream form Virtual Machine, Virtualized Application Client renders Output Stream to the user.

- Interaction provisioning: Virtualized Application Client is also responsible to transfer interactions commands to the Virtual Machine, where they get executed on the Virtualized Application Client.

- Local Resource Provisioning: Virtualized Application Client is responsible for providing local resource to Virtual Machine as per the client requirements. Client may use local device APIs to get the Local Resource data.

4.3. Virtual Machine

Virtual Machine is a server side component emulating different operating systems and responsible for::

- Application Hosting: The basic responsibility of Virrual Machine is to host the Virtualized Applications. The Virtualized Application can be hosted on Virtual Machine dynamically at runtime or they can be pre-configured.

- Output generation: Virtual Machine is the entity which will actually interact with Virtualized Application Client in the Virtualized Application session. One of the main responsibilities of Virtual Machine is to capture the application display periodically and creating a single video stream (Output Stream) to be transferred to Virtualized Application Client. Where, that will be rendered to the user by Virtualized Application.

- Local Resource Rendering: Virtual Machine is responsible to provide Local Resource data (received from Virtualized Application Client) to the Virtualized Application.

- Interaction Execution: Virtual Machine is responsible to execute interaction commands on the Virtualized Application.

5. Possible Standard Opportunities

A standardized protocol is needed, to exchange request/response between Virtualized Application Client and Virtual Machine. The message exchanges may include:

- Application Session Request: Client can request an application session with the Virtual Machine. The parameters may include: screen coordinate, device capability, mouse action (left click, right click, scroll down, scroll up), keyboad keys, compass.

- Output Stream Response: Virtual Machine sends application output streams to the Virtualized Application Client. It can be based on the stream transmission protocols, for example, RTSP.

- Sesssion Suspend Request/Response: Client can request to suspend an application session, and Virtual Machine should keep the application state and send response to Client.

- Session Resume Request/Response: Client can request to rwsume an application session, and Virtual Machine should keep the application state and send response to Client.

6. Difference With Virtual Desktop Infrastructure Proposal

Virtualized Desktop Infrastructure [[I-D.wang-clouds-vdi-problem-statement]] aims to provide capability for the client to access a remote virtual desktop using virtualization technologies.

The differences between Virtualized Application and Virtualized Desktop Infrastructure are:

- VDI focuses on the whole desktop, but virtualized application focuses on one specific application, instead of the whole desktop.

- VDI requires the server to send the whole desktop stream to the client, but virtualized application just needs to show the screen for one specific application to the client.

- VDI requires the client to send the user interactivity to the desktop to the Virtual Machine, but virtualized application just needs to send user interactivity related to one specific application.

- VDI will not use the local resource of the user device, but Virtualized Application can use the local resource on the user

device.

- VDI does not require session suspend/resume, but this is required by Virtualized Application.

- VDI does not cover OS independent use case, usually the VDI Client uses the same OS with Virtual Machine. But for Virtualized Application, one main use case is to support OS-independent use case, that means, client can use a different OS with Virtual Machine.

7. IANA Considerations

This memo currently includes no request to IANA.

8. Security Considerations

User should be authenticated and authorized to consume the Virtualized Applications hosted on the Virtual Machine.

9. Acknowledgements

Thanks to Deepanshu Gautam for the discussion and some initial texts.

10. Normative References

[I-D.wang-clouds-vdi-problem-statement]
Wang, J., Ma, S., and L. Liang, "Virtual Desktop Infrastructure Problem Statement", draft-wang-clouds-vdi-problem-statement-00 (work in progress), January 2011.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

Author's Address

Kepeng Li
Huawei Technologies
Huawei Base, Bantian, Longgang District
Shenzhen, Guangdong 518129
P. R. China

Phone: +86-755-28974289
Email: likepeng@huawei.com