Spice User Guide

Lubos Kocman <`lkocman@redhat.com`>
Arnon Giloba <`agiloba@redhat.com`>
Yaniv Kamay <`ykamay@redhat.com`>

Spice User Guide

by Lubos Kocman, Arnon Giloba, and Yaniv Kamay

Draft 54

Copyright © 2009, 2010, 2011 Red Hat, Inc.

Licensed under a Creative Commons Attribution-Share Alike 3.0 United States License (see http://creativecommons.org/licenses/by-sa/3.0/us/legalcode).

Table of Contents

Chapter 1. Introduction

Spice is an open remote computing solution, providing client access to remote displays and devices (e.g. keyboard, mouse, audio). Spice provides a desktop like user experience, while trying to offload most of the intensive CPU and GPU tasks to the client. The basic building blocks of Spice are:

1. Spice Server
2. Spice CLient
3. Spice Protocol

The following sections provide basic iformation on Spice components and features, obtaining, building installing and using Spice.

Spice and Spice-related Compoments

Spice Server

Spice server is implemented in libspice, a VDI pluggable library. Virtual Device Interface (VDI) defines a set of interfaces that provide a standard way to publish virtual devices (e.g. display device, keyboard, mouse) and enables different Spice components to interact with those devices. On one side, the server communicates with the remote client using the Spice protocol and on the other side, it interacts with the VDI host application (e.g QEMU).

Spice Client

Spice cross-platform (Linux and Windows) client is the interface for the end user e.g. spicec or spicy.

QXL Device and Drivers

Spice server supports QXL VDI interface. When libspice is used with QEMU, a specific QEMU QXL PCI device can be used for improving remote display performance and enhancing the graphic capabilities of the guest graphic system. QXL device requires guest QXL drivers for full functionality. However, standard VGA is supported when no driver exists.

VDI Port Device

Spice protocol supports a communication channel between the client and the agent on the server side. When using QEMU, Spice agent resides on the guest. VDI port is a QEMU PCI device used for communication with the agent.

Spice Agent

The Spice agent is an optional component for enhancing user experience and performing guest-oriented management tasks. For example, the agent injects mouse position and state to the guest when using client mouse mode. It also enables you to move cursor freely between guest and client. Other features of agent are shared clipboard and aligning guest resolution with client while entering fullscreen mode.

Features

The server and client communicate via channels. Each channel is dedicated to a specific type of data. The available channels are following.

Multiple Channels

1. **Main** - control and configuration
2. **Display** - graphics commands images and video streams
3. **Inputs** - keyboard and mouse inputs
4. **Cursor** - pointer device position and cursor shape
5. **Playback** - audio received from the server to be played by the client
6. **Record** - audio capture on the client side

Image Compression

Spice offers several image compression algorithms, which can be chosen on server initiation and dynamically at run-time. Quic is a Spice proprietary image compression technology based on the SFALIC algorithm. The Lempel-Ziv (LZ) algorithm is another option. Both Quic and LZ are local algorithms encoding each image separately. Global LZ (GLZ) is another proprietary Spice technology that uses LZ with history based global dictionary. GLZ takes advantage of repeating patterns among images to shrink the traffic and save bandwidth, which is critical in a WAN environment. Spice also offers an automatic mode for compression selection per image, where the choice between LZ/GLZ and Quic is heuristically based on image properties. Conceptually, synthetic images are better compressed with LZ/GLZ and real images are better with Quic.

Video Compression

Spice uses loss-less compression for images sent to the client. However, video streams are handled differently. Spice server heuristically identifies video areas and sends them as a video stream coded using M-JPEG. This handling saves a lot of traffic, improving Spice performance, especially in a WAN environment. However, in some circumstances the heuristic behavior might cause low quality image (e.g. identifying updated text area as a video stream). Video streaming can be chosen on server initiation and dynamically at run-time.

Mouse modes

Spice supports two mouse modes: server and client. The mode can be changed dynamically and is negotiated between the client and the server.

1. **Server mouse** - When a user clicks inside the Spice client window, the client mouse is captured and set invisible. In this mode, the server controls the mouse position on display. However, it might be problematic on WAN or a loaded server, where mouse cursor might have some latency or non-responsiveness.
2. **Client mouse** - Not captured and is used as the effective pointing device. For enabling client mouse, VDI host application must register an absolute pointing device (e.g. USB tablet in QEMU). This mode is appropriate for WAN or loaded server, since cursor has smooth motion and responsiveness. However, the cursor might loose synchronization (position and shape) for a while.

Other Features

1. **Multiple Monitors** - any number of monitors is supported
2. **Bidirectional Audio** - Spice supports audio playback and recording. Playback is compressed using the CELT algorithm
3. **Lip-sync** - between video and audio. Available only when video streaming is enabled.

4. **Migration** - switching channel connectivity for supporting server migration
5. **Pixmap and Pallete caching**

Chapter 2. Installation

Installing Spice in RHEL or Fedora

Be aware that there is no build of spice-server for i386 so you can run only client part on i386. The reason for it is that there is no buld of qemu-kvm for RHEL of Fedora i386 either.

RHEL >=6 and Fedora >=13

```
yum install qemu-kvm spice-client
```

```
yum install spice-gtk  # Only on Fedora >=14
```

Package spice-protocol will be downloaded automatically as a dependency for package kvm.

RHEL 5

```
yum install kvm qspice-client
```

Similarly as in RHEL6 package kvm has qspice-libs as a dependency.

RHEVM Users

RHEVM users could be also interested in spice-xpi package as it's allow you to execute spice-client directly from the RHEVM UserPortal.

```
yum install spice-xpi
```

Building Spice in Generic Linux

This section is for distributions that don't have *spice* packages in their repositories. It will show you step by step how to build required spice components.

Client requirements

1. **autotools**
2. **pixman >= 1.7.1**
3. **celt = 0.5.1.3** The exact version is required due no backwards compatibility of celt.
4. **alsa**
5. **log4cpp**
6. **openssl**
7. **libXrandr**
8. **cegui**
9. **libgcrypt-devel**
10. **pyparsing**
11. **dev86**

12.**iasl**

13.**nss**

Host requirements

1. **KVM supported by kernel** (It should work also without KVM, but it's not being tested as most Linux distrubitions already support KVM.)

Guest requirements

Linux Guest

spice-vdagent requires virtio-serial support should to be enabled. This will be described in chapter Spice basics [SpiceUserManual-Basics.xml#basics]. Guest should have installed qxl driver (xorg-x11-drv-qxl on Fedora and RHEL).

Windows Guest

Drivers for QXL and drivers for virtio-serial assume Win XP SP3 and Win 7.

Setting up build environment

This is list of prerequisities on RHEL or Fedora. Install package equivalents for your distrubition in case that you're not using RHEL or Fedora.

All prerequisities for Windows are available in one big package which is available at http://spice-space.org/download.html.

```
yum install git pixman-devel celt051-devel cegui-devel libjpeg-devel alsa-lib-de
                openssl-devel libXrandr-devel libgcrypt-devel SDL-devel nss-deve
```

Package prerequisities for Ubuntu

```
apt-get install build-essential autoconf git-core libtool liblog4cpp5-dev libave
                libssl-dev xlibmesa-glu-dev libasound-dev libpng12-dev libfreety
                libogg-dev libxrandr-dev kvm libgcrypt-dev libsdl-dev
```

Building libcacard

Fedora >=14 RHEL >=6.1 has libcacard already available. So you can install it directly trough yum.

```
yum install libcacard
```

Otherwise follow these instructions. **Environmental variable $BUILD_ROOT will point to a directory with stored sources and will be used during whole build process. Variable $INST_ROOT will point to a directory in which will be spice installed.**

```
export BUILD_ROOT=/tmp/spice; mkdir $BUILD_ROOT; cd $BUILD_ROOT;
export INST_ROOT="/opt/spice"; mkdir $INST_ROOT
git clone git://anongit.freedesktop.org/~alon/libcacard
cd libcacard
./configure --prefix=/usr --libdir=/usr/lib64 # Ignore --libdir at Ubuntu
make
make install
```

Getting client sources

```
cd $BUILD_ROOT
```

```
git clone git://cgit.freedesktop.org/spice/spice-protocol
git clone git://cgit.freedesktop.org/spice/spice
wget http://downloads.us.xiph.org/releases/celt/celt-0.5.1.3.tar.gz
tar xvzf celt-0.5.1.3.tar.gz
```

Getting client/server sources

Skip this section if you don't want to build server side.

```
cd $BUILD_ROOT
git clone git://cgit.freedesktop.org/spice/qemu
cd qemu; git checkout -b spice.v13 origin/spice.v13; cd ..
git clone git://cgit.freedesktop.org/spice/spice-protocol
git clone git://cgit.freedesktop.org/spice/spice
git clone git://cgit.freedesktop.org/spice/win32/vd_agent
git clone git://cgit.freedesktop.org/spice/win32/qxl
git clone git://cgit.freedesktop.org/spice/slirp
wget http://downloads.us.xiph.org/releases/celt/celt-0.5.1.3.tar.gz
tar xvzf celt-0.5.1.3.tar.gz
```

Building common sources.

This part applies to both server and client build process.

```
cd $BUILD_ROOT/spice-protocol
mkdir m4
./autogen.sh --prefix=$INST_ROOT
sudo make install
cd $BUILD_ROOT/celt-0.5.1.3
./configure --prefix=$INST_ROOT
sudo make install
```

Building client side tools

```
cd $BUILD_ROOT/spice
./autogen.sh --prefix=$INST_ROOT --enable-smartcard
cd client
sudo make install
```

Building server side tools

These instruction contains flags for minimal working build of qemu with spice support enabled. You might want to build qemu with --enable-io-thread options

```
cd $SRC_ROOT/qemu
./configure --prefix=$INST_ROOT --target-list=x86_64-softmmu --enable-spice
make
```

Setting up PATH

Last steps before starting with spice are to set proper PATH variable. For example RHEL is using /usr/libexec as directory for spicec and qemu-kvm binaries. Following setup should be suitable for qemu and spice build according to instructions in this chapter.

```
echo "export PATH=$PATH:$INST_ROOT/bin:$BUILD_ROOT/x86_64-softmmu >> ~/.bashrc
source ~/.bashrc
```

You should now be able to access qemu-system-x86_64 and spicec binaries.

Chapter 3. Spice basics

Basic Definitions

Host

Host is a machine running instance of qemu-kvm.

Guest

Guest is a virtual machine hosted on host which will be accessed by using spice client.

Client

Client is reffering to a system running spice client (e.g. spicec or spicy).

Launching qemu

I'll use qemu-kvm as a name for the executable. If you're using manually built qemu or qemu without kvm then just replace qemu-kvm with your own binary. I'll use host# client# guest# shell prompt notation to distinguish wherever should be the command executed. See section Basic Definitions [definitions] to be sure that you know difference between the host, client and guest. You can ignore notation If guest, client and host are on the same machine.

First important thing to do is to create a guest image. You can use any raw device like clean logical volume, or iSCSI lun. You may also use file as image for guest. I'll use file created by qemu-img as a demonstration.

Following command will allocate 10GB file. See qemu-img manpage for further information.

```
host# qemu-img create /path/to/xp.img 10G
```

So we already have image created and shall now start with image popullation. I assume that you have locally stored ISO of your favourite operating system so you can use it for installation.

```
host# sudo qemu-kvm -boot order=dc -vga qxl -spice port=3001,disable-ticketing
        -cdrom /path/to/your.iso /path/to/your.img
```

Let's take just briefly look trough used qemu options. Option -boot order=dc specifies that guest system should try to boot from first cdrom and then fallback to first disk, -vga qxl specifies that qemu should emulate qxl device adapter.

There will be no running agent [SpiceUserManual-Introduction.xml#vdagent] on the guest system so we don't have to emulate virtio device yet.

Chapter 4. References

1. Spice remote computing protocol definition