

Debian Stretch Functional Tests Using Alpha8 Build Process

General Findings

hkb_stretch_20161227_1000

Test Version: hkb_stretch_20161227_1000
hbrown

Forward or post as needed to resolve these issues!

1) Keywords:

Debian : Stretch : Virtual Box : 3D Acceleration : gdm3 : lightdm

2) Appreciation

Looking past the problems encountered here, it is clear that a great deal of fabulous work has been done by all the contributors and Debian Stretch will provide a great operating environment. This note is to provide quality feedback to help improve the user experience.

3) Purpose

The purpose of this effort was to test the alpha8 Debian “stretch” installer within a virtual machine context using the VirtualBox hyper-visor. Since an installer is the provisioning to conduct fine grain testing for other matters, all technical and perceptual issues are disclosed related to the initial installation. Speculative and perceptual comments are provided since the installation process helps build the first impression.

4) Summary

The core issues during this test are: (1) Virtual Box 3D Acceleration (enabled) does not seem to work well with OpenGL applications; (2) gdm3 Desktop manager does not start and seems to hang the system on xorg start; (3) When the default desktop selection is made in tasksel there is no default operational networking within “/etc/network/interfaces/” which makes the networking nonoperative (by default) during a boot in the recovery mode.

Wild Guess (Last Note Before Posting): From the 3D tests, there are three facts. (1) One is that there seems to be a 15 sec delay (Odd Number) before updates are noticed when glxgears starts. (2) The traceback in blender crash with lib6 in the traceback; and (3) On the host a 64 bit for loop counter takes 11 Seconds to complete. Thus! ... There could be a call back or pointer reference or integer via libc6 that causes a loop to mis-count jumping its termination. The loop continues, potentially corrupting random data and syncs once it has rolled around its full 64 bit range. Maybe a mismatched header file. Could the same mismatch be causing KDevelop to fail on all cases? Hmmm!

Note that a small test program was constructed and tested on the host machine. The full range loop took around 11 seconds with only an “i++” in the loop. This is too close in time to ignore. Libc6 seems to be involved due to the blender crash report. See below. This might be causing the delay at the user interface. Also in some cases, the delays are half as much indicating a “signed” roll around which takes about half the time. **Note ... This is all a wild guess.**

Also please excuse the typos... I have run out of time that I can spend on this post. It's been fun!!!

5) Projected Initial Impression

Overall, the installation process seemed to work, but a few critical bugs made a bad first impression.

Even some temporary warnings might be helpful at this time. The test bed context in this case is: (1) The user has been using Debian Stretch with some success; (2) The user makes use of virtual machine technology; (3) The user plans to tryout the default (GNOME) and KDE Plasma desktops to evaluate the new default installation; (4) The user uses Debian Jessie as the host system.

The user downloads the alpha8 build disk and creates a virtual machine using Debian Stretch alpha8 process using Virtual Box (VBOX). After three attempts, the user finds little that is working, which is an incorrect conclusion. Actually, a great deal is working, but a few critical bugs shadowing the functional elements. The three attempts fall along these lines. (1) gdm3 does not start, (2) KDE Does not work with Virtual Box with 3D Acceleration (Guest Additions was loaded), and three (3) The network does not work in boot recovery mode.

6) General Recommendations

(1) Consider minimizing the use of hardware dependent OpenGL and other GPU technologies in system management functions. This includes boot, display manager, system configuration, basic system diagnostics and development tools. The goal is to be able to have debug and other diagnostics tools operational even when there are hardware issues with hardware dependent OpenGL and GPU drivers so that fine grain testing can continue when there is a problem on the same system. Under these test condition, at this time, GNOME is able to operate to a useful degree to conduct some fine grain testing, but KDE Plasma fails in a catastrophic manner requiring a power-off-cycle within a VM. Even if a desktop system heavenly uses OpenGL / GPU technology, its own stability should minimize the use of the higher risk technologies. It is believed that this approach will help speed up the insertion and test of new technologies in the long run.

(2) Consider breaking out the display manager in the selection process at install time. This should be independent of the desktop being selected. In this test case “lightdm” would have been a better choice as a default.

(3) Consider making sure that when boot recovery mode is selected, the networking that was used during the installation process remains operational in all installation contexts. This is not the case if desk top install is selected.

(4) Consider making provision so that the user is left at an operational console in the event xorg fails to configure. It is important that the network be operational at this time based on the installation process as the default. Note that within a VM context switching to a virtual console within the VM may not work, although there may be a simple work around. The host may intercept this key and switch to a host virtual console. Thus, virtual consoles may not be a viable fall back to a casual user. This may be a feature that, in the long run, could be added to a VM manager, but not clear of the current status or work arounds at this time. Needs more study.

7) Test Findings

(P001) Note that the VBOX Enable Flag within the VBOM Manager toggles this problem on and off. It must be toggled prior to VM start.

There is an over arching problem with VirtualBox(VBOX) if “3D Acceleration” is enabled (Setting/Display). The cause of the problem seems to be at some intersection of OpenGL related calls common to glxgears, QT5, blender, VBOX with 3D enabled, and potentially the host device video driver (NVIDIA in this case); although the host did not encounter any problems during the test.

However, not all applications have problems, but in the limited testing, OpenGL intensive applications do seem to have consistent issues. Most non-OpenGL applications did not have problems.

Detailed observations are provided in later section.

(P002) The default alaph8 install process led to a blinking screen when gdm3 was to start. For some reason, the xorg(X11) process to resolve the default settings did not work. The blinking was a result of various tries to resolve the settings. After a while, the attempts stopped, but there was no fall-back to a console. Thus, the system appeared to freeze.

(P003) Within tasksel, when the graphical installation process is selected, the system is left with no working network in the event there is a need to enter boot recovery mode. This creates an additional obstacle to resolving a problem when there is a need to access the Debian mirrors by-way-of the networks. In this case, the hotplug entry within /etc/interfaces has been removed as compared to that when no graphical installation is selected, which does include the hot-plug entry. It is understood there may be a preference to relying on the desktop network management support, but this is not performed consistently enough throughout all use cases to be relied upon. Consider providing default networking during a recovery mode boot or when xorg fails to start.

(P004) The KDE desk top does not function with 3D Acceleration enabled. Note that 3D Acceleration will not attempt to run until VBOX Guest Additions is installed. Even when 3D is selected, no 3D Acceleration is provided. As such, if one has VBOX 3D Acceleration enabled and installs the KDE desktop, without GNOME, then everything seems to work fine. However, when VBOX Guest Additions is installed, KDE desktop becomes unusable. At first impression, there seems to be a catastrophic failure, but in reality much is working even with 3D Acceleration enabled.

(P005) Bad user impression. At this point, from a casual user perspective, (a) “gdm3” blocks the default GNOME install and one concludes that GNOME is dysfunctional, which is not the case; (b) One retries KDE desktop followed by the VBOX Guest Additions with 3D enabled leading also to a dysfunctional system, which is not really the case; and (c) When one tries to fix the issues, when entering boot recovery mode, one finds that the network used for the installation no longer works. The over all impression is that the system is not ready for user based testing, which is not the case.

(P007) This is a speculative comment. In these tests, NVIDIA Titan Black (Kepler) and GTX-1060 (Pascal) were being used. Both seemed to behaved the same, but full testing was not conducted with both, just the NVIDIA Titan Black (Kepler). The failures where initially observed with the GTX-1060 and was switched out with the Titan Black. However, no change was observed. Note that it is recognized that some of the VBOX 3D issues may be an artifact of the NVIDIA drivers. However, consider treating the issue as an NVIDIA, QT5 and VBOX integration issue; or potentially a minor mismatch between lib versions / header files in the build process. When 3D Acceleration is disabled, the QT5 based applications seems to work well. Note that in this configuration, gdm3 did not work in all cases. Note that throughout all the testing, the host was Debian Jessie with the latest updates and it never crashed or hung once throughout all the testing!!! Also, the VBOX manager seemed to be stable and when one VM had problems, it did not impact the operations of the other VMs that was running at the same time. As many as 4 VMs would be in use at any one time. Also, the latest QT5 development tools were loaded to a user directory within a VM. Qt5 Designer, for example, would not work with VBOX 3D acceleration enabled, but seemed to work find when VBOX 3D acceleration was disabled.

8) Installation Process that Worked

The following is a workaround to the above problems and highlight what is working as of this writing. The test set used is as follows:

- a) Debian Jessie was the host and is fully upgraded. There were no issues with the host.
- b) <http://download.virtualbox.org/virtualbox/debian> was used, but only after the Debian VBOX packages showed the same problems. The latest versions were used to seek a fix, but none was found.
- c) In these tests, NVIDIA Titan Black (Kepler) and GTX-1060 (Pascal) video cards were used. The following commercial driver was used: NVIDIA-Linux-x86_64-375.26.run. Note that host system has always used the latest commercial NVIDIA drivers with no significant issues to date on this hardware. The host was stable during all tests.

The alpha8 installer use process is as follows:

- 1) Start VBOX making sure to disable 3D Acceleration.
- 2) Install Debian stretch using alpha8 installer.
- 3) Use the relevant defaults.
- 4) When tasksel starts, uncheck
“[] Debian desktop environment”
- 5) Complete the boot process using the defaults.
- 6) Login as root using console prompt. “//...” are comments and instructions; and should not be typed. “#” implies console with root privileges in some manner. “\$” implies console with user privileges in some manner. Do the following in the MV:

```
# nano /etc/apt/sources.list
// and add contrib and non-free to all entries.
# apt-get update
# apt-get -u -o APT::Force-LoopBreak=1 dist-upgrade
// This will make sure the kernel is updated. It was needed in this case.
# dpkg --configure -a
# apt-get -f install
// These will help fix errors, just in case. It was not needed in this case. Note this apt-get sequence
works much of the time, does not always provide proper sync for some operations. “aptitude” seems to
work better. Needs further study.
# apt-get autoremove
// Cleanup
# reboot
// New login – Note network is working at this point.
# tasksel
// [*] Debian desktop environment
// [*] ... GNOME
// TAB and Press ok to install selection
// Note: That a number of superbloc probes / errors are
// displayed with respect to sda2 which is a logical
// extended partition with no file system. It was ignored.
# apt-get install lighdm
// Select lightdm as the desktop manager.
// This will fix the gdm3 problem.
# tasksel
// [*] Debian desktop environment
// [*] ... GNOME
// [*] ... IDE
# dpkg-reconfigure lightdm
// Maybe install lightdm last.
```

```

#reboot
// Test and shutdown. At this point, most everything seems to work.
// As a side note, if kdevelop is installed, it crashes at this point. Many of the QT5 Tools seem
// to work although no comprehensive test was conducted.

Boot image and enter into recovery mode.
Login as root into recover mode.
// The following is what is needed to fix the network while in recover mode.
# nano /etc/network/interfaces
// Place the following at the bottom of the file:
//---> allow-hotplug enp0s3
//---> iface enp0s3 inet dhcp
// Save
// Best to logout, shutdown all Vms, and restart the VM manager.
// Some minor error have been noticed during the module build process which this may help.
// On host start VBOX
Within VBox Manager, place the VBOX Guest Additions v5.1.12 iso into the virtual CDROM.
// Start VM
# apt-get install dkms
# apt-get install aptitude
# aptitude
// press uUg and be sure everything is in sync. Unsure why, but it
// seems that attitude helps keep everything in sync. This should not be the case,
// but have not tried to reproduce the problem. However, cycling through with
// aptitude has not caused any problems when not needed. When needed, it highlights
// errors and fixes the problem, as in this case. Then VBox build process works better in such cases.
// I think the root cause was the the kernel was updated since the alpha8 build, and
// the header files need to be sync'ed properly.
// If someone can provide a apt-get method to get things sync'ed for sure, that would be great.
# cd /media
# mount cdrom
# cd cdrom
# sh VBoxLinuxAdditions.run
# usermod -a -G vboxsf user
// user is the account name that you plan to use.
# shutdown now
// DONE
// The installation is now complete and more fine grained testing can be conducted.

```

9) Some Additional Testing Follows

Below is things to try with additional observations.

a) VBOX 3D Disabled – OpenGL Based Application Testing

Applications glxgears, blender and plasma will be the test cases. First, let us create a baseline with VBOX 3D Acceleration disabled. First GNOME, then KDE Plasma.

GNOME Without VBOX 3D Enabled Test: In VBox manager setup the shared folder, for reference it will be called vb_share. Select Auto mount. Start image. Select GNOME as the desktop. At this

point, desktop resizes, share works, copy paste and many applications work. Superficially, most things worked by entering applications and trying a few things and quitting. All the base applications work.

KDE Plasma Without VBOX 3D Enabled Test: Logoff, select plasma as desktop and login again. It was noticed during this test, that under devices, the guest “/” directory was mounted and not vb_share. In GNOME vb_share was mounted correctly. There are some things about Dolphin that does not seem correct. Needs full testing. Moving on.

From a console, type:

```
$ glxgears
```

Behavior is the same in both KDE and GNOME. The host shows that all 8 cores are at about 50% load. Both show about 600 frames per second.

b) VBOX 3D Enabled (**3DA**) – OpenGL Based Application Testing

VBox Guest Additions 3D Enabled Test: Shutdown VM, within VBox / Display; check 3D. Restart VM image. **Be sure to set GNOME as the desktop!**

3DA_Test_01: This is where some of the problems start: The GNOME desktop came up fine. The gnome console was started followed by glxgears. The host CPU usage was minimal as expected since the GPU is being used. At first, the borders around the glxgears did not appear. It took a good 15 seconds for them to display. Also the desktop is almost non-responsive.

It should be noted that when 3D was not selected it indicated that around 600 frames a second was being processed. The speed reduction was due to monitoring syncing. Now, it is just under 30 frames per second. If nvidia-setting is run, and display is at 60Hz which is driver by a display port interface.

When the console window is moved it takes about 15 seconds if responsive. Most of the time it does not respond. Glxgears continue to run, the mouse moves but clicks do now work for the most part. VBox manager window containing the VM works fine, other VMs continue to work and the host works without any issues. When the VBox VM is moved around on the host desktop, compositing works as one can see the underside of the host desktop while the window is being dragged. All this time the glxgearx is working. It almost seems glxgears is in model mode and blocking VM IO, but some IO does get through if one hovers the mouse for around 10 seconds or more. Clicking on the console followed by a ^c kills glxgears, but may to be tried several times with the pointer over the entry area.

3DA_Test_02: When the screen is moved around tearing is observed. LibreOffice is started and the experiment is repeated. Glsgears is started again. On a 4K screen, the VM now takes up more than 50% of the desktop. It take about 15 seconds for the frame to appear. Screen size makes no difference in this time. LibreOffice window will not move when glxgears is running. Tried nice, but no difference. It is noticed that the ^c on the console does not work until the frame displays and there is a console update. Again, this is about 15 seconds after the application starts. Konsole work much the same way from within GNOME console.

3DA_Test_03 - glxgears: On the host, it should be noted that one screen is running at 60Hz using a display port interface and one at 30Hz using a HDMI interface. Setting were changed to a single monitor, most of the nvidia-setting were toggled, the tests repeated on both displays. The output on glxgears seemed to track the frame rate of the screen it was on. When moving things around, for some reason, now and then, a change in size of VM window caused the screen to jump to the other screen. Not sure of the cause. Needs more study. Moving on.

3DA_Test_04 - blender: When blender is started from a command console and then the command console is moved, it seems to move behind the main window, but in front of the title. Other than the title bar, there is no frame showing. It is if the z-ordering is being improperly processed. It is possible to move command console behind the blender main window and then grab it again. The arrow image changes as if the command console is on top, but as being painted, it is behind the main window. The QT5 tools have the same or similar problem.

Both the command console and the blender window can be moved. However, the repaint is delayed in the blender window. As such there is a great deal of tearing between the title bar / frame and the blender application window. The black in the frame area is about 1in when the window is moved. Resizing of the blender window works. Using the menus worked to some degree, but with a great deal of flicker and the 3D box did update. However, as soon as one clicks on the main screen the application crashes.

The following was cut and pasted from the command console VM window when blender was start. The cut and paste features do work well so far.

Cannot connect to server socket err = No such file or directory

Cannot connect to server request channel

jack server is not running or cannot be started

JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for 4294967295, skipping unlock

JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for 4294967295, skipping unlock

OpenGL Warning: crPixelCopy3D: simply crMemcpy'ing from srcPtr to dstPtr

OpenGL Warning: No pincher, please call crStateSetCurrentPointers() in your SPU

Writing: /tmp/blender.crash.txt

Segmentation fault

bpy.ops.transform.translate(value=(-70.0071, 33.9113, 156.106), constraint_axis=(False, False, False), constraint_orientation='GLOBAL', mirror=False, proportional='DISABLED',

proportional_edit_falloff='SMOOTH', proportional_size=1) # Operator

bpy.ops.transform.rotate(value=-0.0152106, axis=(-0.818828, 0.361665, -0.445779),

constraint_axis=(False, False, False), constraint_orientation='GLOBAL', mirror=False,

proportional='DISABLED', proportional_edit_falloff='SMOOTH', proportional_size=1) # Operator

bpy.ops.transform.mirror(constraint_axis=(False, False, False), constraint_orientation='GLOBAL',

proportional='DISABLED', proportional_edit_falloff='SMOOTH', proportional_size=1) # Operator

backtrace

blender(BLI_system_backtrace+0x30) [0x560440659ec0]

blender(+0x1023101) [0x56043fc02101]

/lib/x86_64-linux-gnu/libc.so.6(+0x33040) [0x7fb1a65d3040]

(END)

!!! Maybe a “/lib/x86_64-linux-gnu/libc.so.6(+0x33040) [0x7fb1a65d3040]” Problem.

DA_Test_06 - Plasma: Logout and select Plasma as the desk top. Little or nothing works. The inial paint looks correct. However any ckick causes things to disapear. It is possible that the background is being placed over everything else in the same manner as in blender's main window. Solving the bender main window problem in GNOME may solve part of plasma's problem. “glxgears” has different

symptoms and may lead to a differing problem. This would explain much of the behavior. However, plasma is not functional enough run any applications for fine grain testing.

DA_Test_06 – KDevelop: KDevelop crashes in all contexts.

DA_Test_06 – QT5 Designer: Unstable with 3D Acceleration enabled.

10) System Test Set

a) Host: Jessie with deb <http://download.virtualbox.org/virtualbox/debian> jessie contrib

NVIDIA-Linux-x86_64-375.26.run

Linux Kernel: 3.16.0-4-amd64

AMD FX(tm)-8150 Eight-Core Processor

ASUS SABERTOOTH 990FX R1 Motherboard

NVIDIA Titan Black ... but note same results for the most part with the ASUS GTX-1060 Pascal

2x - 4K Monitors @ 60Hz

b) Host: Virtual Box Version 5.1.12 r112440 (Qt5.3.2)

c) Host: 32GB Real RAM

d) Guest: 3D Disabled or Enabled as stated

Base Memory: 4096MB

VDI

Pointing Device: PS/2

Processor(): 8

Network Adapter Type: virtio-net

Video Memory: 128 MB

Debian Installer Stretch Alpha 8 Installer

ISO Used: debian-stretch-DI-alpha8-amd64-netinst.iso