Openstack

**Troubleshooting:**

1. In the command line terminal, if the “openstack” command is not recognized, you need to change the language setting of the terminal.

unsupported locale setting

Traceback (most recent call last):

File "/usr/lib/python3/dist-packages/CommandNotFound/util.py", line 24, in crash\_guard

callback()

File "/usr/lib/command-not-found", line 69, in main

enable\_i18n()

File "/usr/lib/command-not-found", line 40, in enable\_i18n

locale.setlocale(locale.LC\_ALL, '')

File "/usr/lib/python3.2/locale.py", line 541, in setlocale

return \_setlocale(category, locale)

locale.Error: unsupported locale setting

Solution:

export LANGUAGE=en\_US.UTF-8

export LANG=en\_US.UTF-8

export LC\_ALL=en\_US.UTF-8

locale-gen en\_US.UTF-8

sudo dpkg-reconfigure locales

1. When we create instances, we might accidentally create “zombie instances” that are in ERROR or BUILD status, mostly because of the timeout issue for connections to RabbitMQ service. The “zombie instances” are unsuccessfully launched and not running, and they are just recorded in data base, which will make us unable to delete these instances with “nova delete” command.

Solution: Force delete the instances. First, Using “nova force-delete instance-id”, if it works, then problem solved. If not, you will need to go into the database and delete the corresponding data.

**ls /var/lib/nova/instances/**

**rm -rf /var/lib/nova/instances/instance-00000010**

Then create your bash file:

**nano deletevm.sh**

#!/bin/bash

mysql -uroot << EOF

use nova;

DELETE a FROM nova.security\_group\_instance\_association

AS a INNER JOIN nova.instances AS b

ON a.instance\_id=b.id where b.uuid='$1';

DELETE FROM nova.instance\_info\_caches WHERE instance\_id='$1';

DELETE FROM nova.instances WHERE uuid='$1';

EOF

**chmod +x deletevm.sh**

Then run the script, for example, if you want to delete instance with id of efc6e9c7-4ef8-4350-9451-83bcfcafe101, run commands as follows:

**./deletevm.sh efc6e9c7-4ef8-4350-9451-83bcfcafe101**

1. How to delete an instance with a volume still attached

The instances get stuck in a “deleting” state and the instances and the volumes aren’t able to be cleaned up. The compute node has actually released the iscsi target cinder presented to it:

[root@host11 ~]# iscsiadm -m session  
iscsiadm: No active sessions.

though, tgtd hasn’t released the lvm device for some reason, so the device can’t be deleted:

[root@host2 ~]# lvremove cinder-volumes/volume-b9869d42-418f-4d7c-b4bf-951b035d1817  
Do you really want to remove active logical volume volume-b9869d42-418f-4d7c-b4bf-951b035d1817? [y/n]: y  
device-mapper: remove ioctl on failed: Device or resource busy  
Unable to deactivate cinder–volumes-volume–b9869d42–418f–4d7c–b4bf–951b035d1817 (253:59)  
Unable to deactivate logical volume “volume-b9869d42-418f-4d7c-b4bf-951b035d1817”  
[root@host2 ~]# lsof /dev/cinder-volumes/volume-b9869d42-418f-4d7c-b4bf-951b035d1817  
COMMAND PID USER FD TYPE DEVICE SIZE/OFF NODE NAME  
tgtd 10812 root 74u BLK 253,59 0t0 45476393 /dev/cinder-volumes/../dm-59

To fix this use tgt-admin to delete the target, there by relinquishing tgtd’s use of the volume and then clean up the cinder entry in the database so that OpenStack think’s it can now do the needful. (and actually can now!)

[root@host2 ~]# tgt-admin -s | grep b98  
Target 61: iqn.2010-10.org.openstack:volume-b9869d42-418f-4d7c-b4bf-951b035d1817  
Backing store path: /dev/cinder-volumes/volume-b9869d42-418f-4d7c-b4bf-951b035d1817  
[root@host2 ~]# tgt-admin –delete iqn.2010-10.org.openstack:volume-b9869d42-418f-4d7c-b4bf-951b035d1817

mysql> use cinder;  
mysql> update volumes set status = ‘error’, attach\_status = ‘detached’ where id = ‘b9869d42-418f-4d7c-b4bf-951b035d1817’;  
Query OK, 1 row affected (0.04 sec)  
Rows matched: 1 Changed: 1 Warnings: 0

[root@host2 ~]# cinder delete b9869d42-418f-4d7c-b4bf-951b035d1817

Now that the volume is cleaned up the instance needs to be massaged a bit too so that it can be torn down as well:

[root@host2 ~]# nova reset-state 4365e90f-b7cf-4253-9ded-1844df1c786b  
[root@host2 ~]# nova delete 4365e90f-b7cf-4253-9ded-1844df1c786b

And if the instance still doesn’t want to delete set it to deleted in the db:

mysql> UPDATE instances SET  vm\_state=’deleted’,task\_state=NULL,deleted=1,deleted\_at=now()  WHERE uuid=’4365e90f-b7cf-4253-9ded-1844df1c786b’;

If you had to edit the DB like this then go make sure that the instance is actually undefined on the compute node. For me I was able to look at the dashboard’s admin panel to see how many instances were on the compute node and make sure that virsh list reported the same number, and it did.

1. LXC issue. For GPU, you will need to change the configuration from “qemu” to “lxc”, this won’t be hard. However, once you do this, you will fail to launch a instance. Since OpenStack does not support LXC in the first place. However, it is still included into nova and working via the libvirt driver. So I edited /etc/nova/nova.confand enabled lxc for libvirt.

[libvirt]

vif\_driver=nova\_contrail\_vif.contrailvif.VRouterVIFDriver

virt\_type=lxc

First I just tried to boot a usual kvm-Image, and it turned out, that it almost works. For lxc the image is mounted via qemu-nbdsomewhere to /var/lib/nova/instances/7849061d-740f-4727-9fa8-eca84bb3d77b/rootfs, but nova does not take care about partitions. So the mount failed:

Failed to mount filesystem: Unexpected error while running command.

Command: sudo nova-rootwrap /etc/nova/rootwrap.conf mount /dev/nbd8 /var/lib/nova/instances/77d6a580-eaba-4e8f-99e2-f9810df74f24/rootfs

Exit code: 32

Stdout: ''

Stderr: 'mount: block device /dev/nbd8 is write-protected, mounting read-only\nmount: you must specify the filesystem type\n'

Not much to do to fix this. Edit /usr/lib/python2.7/dist-packages/nova/virt/disk/api.pyline 380 and add partition=1:

img = \_DiskImage(image=image, use\_cow=use\_cow, mount\_dir=container\_dir, partition=1)

Restarting nova-compute. The image can be mounted, but the instance will not start because of the network anyway.

**TIPS:**

1. When error occurs, first, check which part of the Openstack it is related to, is it a nova (compute) error, an image (glance) error, or a network (legacy) error. Then you might need to go into the /var/log directory and check the corresponding log file to find out the errors. Most of time you will able to find the errors and work it around by checking the error message on the Internet.
2. The network topology is slightly different from the one in ASCC lab. It is connected as follows: both of em1(network interface) of compute 1 and em1(network interface) of compute2 are connected to the LANs ports of the szar\_lab2 router. Em2 of controller, compute1 and compute2 are all connected to the NETGEAR96 router, which will be providing communication in management network. Last, the controller node must also connect to the szar\_lab2 router.

**Future work:**

GPU: To add GPU to the current Openstack cloud, we can use the help of the following links.

https://wiki.openstack.org/wiki/Pci\_passthrough

<http://www.linux-kvm.org/page/How_to_assign_devices_with_VT-d_in_KVM>

Commands shortcuts:

To save some trouble for maintainance, I wrote a couple of commands shortcuts so that we don’t have to go back to the documents again and again for those instructions.

Xunfei

1. ShareSDK.

Weibo share :

The whole idea of Weibo sharing is that when user say “发微博”, the Xunfei will recognize the “发微博” characters and stop current speechRecognizer and start another speechRecognizer to recognize the content of user’s following speech, which will be the content of Weibo.

Weixin share (Weixin SDK and shareSDK)

Both of the Weixin SDK and shareSDK provides functions to send Weixin message to friends and to Moments, however, we are not currently using it because even though it can translate the speech into text and prepare the content to send for you, you will still have to press the “send” button to send out the message, which makes it not 100 percent automate.

Weixin share (currently in use)

To solve the problem mentioned above, we combine the wxBot in this context. When the user say “发微信”, it will start a recognizer to get the content of the speech and then send the username and content to the wxBot. Then wxBot will send the content to the user.

Troubleshooting:

1. Sometimes when you modify the code and reinstall the app on Android, you will get an error when you say “发微博”, because you are not authorized yet. To initialize the sharing system, you will need to say “启动分享”, which will bring up the onekeyshare function. Then you will need to click the “微博” and log in to your account. After the initialization, you will able to do Weibo sharing.
2. When sending username and content to the wxBot, remember to make sure they are connected to the port, for example, 10003, for the socket communication to work.
3. Set the default username and default content to some value to avoid the situation when the speechRecognizer fail to get the content of your speech and leading to void of username and content.
4. In the hearClass, remember to change the ip address of the socket in uploadToServer class. Change the ip address to the machine that is running wxBot. And set the port to the same
5. Remember to put ShareSDK.xml in the src/asset folder..
6. Xunfei with wxBot

To use Weixin voice message to control the wxBot. We need to implement part of Xunfei’s function on Ubuntu.

Since wxBot is running under Ubuntu, we need to use the Xunfei SDK of Linux.

There is a certain requirements of the voice file for Xunfei Linux SDK:

1. For speech to text:

16k sample rate or 8k sample rate, little endian, wav formatted, mono, 16 bit

Since the Weixin’s message is 8k sample rate and meet the rest of above requirements, by using the following command:

ffmpeg -i "+ voice +" -acodec pcm\_s16le -ac 1 -ar 16000 weixinvoice.wav

we can convert “voice” file into the acceptable .wav file that can be processed by Xunfei.

1. For awaken(唤醒)

ffmpeg -i "+ voice +" -acodec pcm\_s16le -ac 1 -ar 16000 weixinvoice.pcm

We need to convert it into pcm file.

After translating the Weixin voice file, we can send the translated text to the Turling robot’s cloud, and it will give us the responses.

1. Facial Expression

To control the facial expression of the Xiaobao robot, there are following steps:

1. Wake up Xiaobao
2. Listen to the user and check if the content contains some commands to control facial expression, for example “卖个萌”
3. Call the corresponding facial expression function
4. Xunfei’s Q&A database.

We can write our own questions in the Excel just like the template. Then we save it as csv file and upload it to the Q&A database.

具体见（讯飞云问答库模板）

4 channel mics in ROS

1. Mc.py
   1. This python file is used to control the servo by using the message from topic /sound\_src, which will give us the angle information of the sound source. Then will use serial port to send the message to Arduino.

Troubleshooting:

Remember to use a class to do the job instead a straightforward controlling node. The reason to that is you need to initialize your serial port class as a member of your class, so that you can use thie self.ser (serial port communication class) anywhere in your code without error.

try:

self.ser = serial.Serial('/dev/ttyUSB0', 9600)

time.sleep(2)

except Exception, e:

traceback.print\_exc()

print 'open serial failed.'

exit(1)

1. Speed control

To control the speed. We need to modify the firmware of the Arduino nano. The basic concept is to divide a large angle into smaller angles and add delay between this small angles. Also, we need to keep record of the previous angle so we can correct turn from the current one to the desired one.

1. Transfer functions
2. Put the robot\_sound\_loc.zip in to the config folder.
3. Change the LOC variable to robot\_sound\_zip in the .sh file.
4. Troubleshooting:
5. Cannot find the microphone.

Solution: run “cat /proc/asound/cards” to find the correct device number.

1. Cannot find file “ttyUSB0”

Solution: change the “ttyUSB0” in the bot.py and mc.py to “ttUSB1”

1. Permission deny to serial port when run the mc.py

Solution: 1. Su to the root user 2. Run the “ source /opt/ros/indigo/setup.bash” 3. Run “source /catkin\_ws/devel/setup.bash”.

Weixin Utility Development Flow

1. wxBot movement control

To control the movement of Wechat (“左转” and “右转”), the following steps are required.

1. Convert the Wechat’s voice message to the right format
2. Translate the formatted voice file into text.
3. Check if the text contains “左转” or “右转” command.
4. If so, we parse the Chinese character and leave out only numbers, which will be the degree to turn.
5. Weixin picture ability

Same idea as above, first we convert the voice message into the text. Then we call the photo shooting function.

(具体见 command description)