Modifications to the Core to support Kubernetes Usecases (Kali Release)

Scenario Agnostic

1. --k8s should be used.
2. The testcase name should be explicitly specified – there should be a 'c' in the testcase.
3. conf/12_k8s.conf is correctly configured - especially the **K8S_CONFIG_FILEPATH, NETWORK_ATTACHMENT_FILEPATH & POD_MANIFEST_FILEPATH.**
4. Important files:
    a. core/pod_controller.py - Based on the deployment, this decides how many pods to deploy.
    b. pods/papi.py - Does the actual deployment of Network attachments and pods.
    c. testcases/testcase.py - controls the flow
        i. Variables: *_pod_ctl, _pod_list, _evfctl,*
        ii. *_pod_ctl: Reference to Pod Controller. This is a main handler using which we manage the pods.*
        iii. *_pod_list : list of pods-references.*
        iv. *_evfctl : Use to configure the external vswitch. Currently, only flow-entries and Interface Up/Down is managed*

There was an issue with the way the ViNeperf python scripts launch the ovs-vswitchd process. With the current code it seem we can only reserve ports once or twice before Trafficgen-pod fails to work. The issue is caused because the ViNeperf process stops reading the file descriptor which the ovs-vswitchd process is using for stdout after around ~30 sec. This causes the ovs-vswitchd stdout buffer to get full and hang any ovs-vswitchd threads which write to stdout.

The reason this happens was because the ovs-vswitchd process is launched using pexpect and a thread is spawned to read the process stdout using the tools.tasks.Process.relinquish() function. The thread which is reading the stdout calls pexpect.pty_spawn.read_nonblocking() with default arguments which by default uses the timeout used for spawn which by default is 30 sec.

https://github.com/opnfv/vswitchperf/blob/master/tools/tasks.py#L394

https://pexpect.readthedocs.io/en/stable/_modules/pexpect/pty_spawn.html#spawn.read_nonblocking

The modification is made to the vineperf/tools/task.py file to call to read_nonblocking to pass timeout=None and it fixed the issue. With this, the trafficgen pod can reserve/release the ports multiple times.

```
[opnfv@worker vswitchperf]$ git diff tools/tasks.py

diff --git a/tools/tasks.py b/tools/tasks.py

index 4e03f85..06408ab 100644

--- a/tools/tasks.py

+++ b/tools/tasks.py

@@ -391,7 +391,7 @@ class Process(object):

        def run(self):

            while True:

                try:

-                   self.child.read_nonblocking()

+                   self.child.read_nonblocking(timeout=None)

                except (pexpect.EOF, pexpect.TIMEOUT):

                    break
```

Scenario-Specific:

1. Vswitch is managed by ViNePerf - North-South
    a. Single-Pod
    b. Multi-Pod*
2. Vswitch is NOT managed by ViNePerf - North-South
    a. Single-Pod
    b. Multi-Pod
3. Using with Non-vSwitch-based-CNI - North-South.
4. Vswitch is manage

Scenario 1a:

Only OVSDPDK is supported. To support VPP, we need to modify how VPP is deployed. Current VPP deployment is NOT compatible with Userspace CNI.

For this case, ensure the following configuration:

1. The 'deployment' of testcase starts with pc (For KaLi release - as only North-South is supported for now).
2. The EXT_VSWITCH in conf/12_k8s.conf is False.
3. VSWITCH is set to ovsdpdk

Scenarios 1b:

This is WIP - Userspace CNI with ovsdpdk has issues with multiple pods deployment.

Scenario 2a:

Only VPP is used, as OVSDPDK is well-managed by ViNePerf.

Changes to support this scenario needs more explanation:

ViNePerf decides on how switch is managed based on (a) 'deployment-scenario' and (b) VSWITCH type.

In ViNePerf vswitch type can be set in 4 ways:

1. in the arguments, --vswitch
2. In testcase definition, under 'vSwitch'
3. In conf/02_vswitch.conf modify the variable VSWITCH (default)
4. Separate configuration file the variable 'VSWITCH'

If, vSwitch is set to 'none', then ViNePerf will treat it as "pkt-forwarding' case and does not deploy any vSwitch, else it creates vswitch based on 'deployment-scenario'. Here, 'deployment-scenario' mainly control what kind of flow and interface configuration should be done on the managed vSwitch.

As this is an external case, we should set the VSWITCH to 'none', which will make ViNePerf to consider it as a 'Packet-Forwarding' scenario. We do not want to use the packet-forwarding option too - hence, a 'Dummy' packet-forwarder (named as DummyFWD) is created. Using vSwitch as 'none' and PKTFWD as DummyFWD, we will ensure the ViNePerf will not bother about vSwitch, and just focuses on Pods and Traffic-management.

For this case, ensure the following configuration:

1. set vswitch to none - use any one of the above 4 approaches.
2. PKTFWD is set to DummyFWD
3. The 'deployment' of testcase starts with pc (For KaLi release - as only North-South is supported for now).
4. The EXT_VSWITCH in conf/12_k8s.conf is set to True.
5. EXT_VSWITCH_TYPE is set to 'vpp'

Scenario 2b:

Just ensure that the the length of these two variables **NETWORK_ATTACHMENT_FILEPATH & POD_MANIFEST_FILEPATH** is 2**.**