

Các lưu ý chính về Nova trong Kilo và Liberty

I. Kilo:

1. vCPU(Guest) pinning to pCPU(Host):

- Được hỗ trợ bởi NumaTopologyFilter trong FilterScheduler.
- Sẽ có thêm 2 parameters của flavor được support:
 - hw:cpu_policy=shared|dedicated
 - hw:cpu_threads_policy=avoid|separate|isolate|prefer
- Nếu như cpu_policy được set “dedicated”, các vCPU sẽ được pin xuống pCPU và phụ thuộc vào topology của pCPU (số lượng socket, số lượng core/socket, số lượng thread/socket).
- cpu_threads_policy điều khiển việc guest scheduling vào các host dựa trên CPU threads.
- Trong quá trình migration, cấu hình cho vCPU pinning sẽ không được re-create trên host mới.

2. Huge page

- Flavor hỗ trợ parameter dưới đây cho hugepage:
`hw:mem_page_size=large|small|any|2MB|1GB`
- Nova-scheduler sẽ lấy thông số về kích cỡ của page (page size) trong flavor để chọn những host phù hợp. Quá trình schedule này được hỗ trợ bởi NumaTopologyFilter.
- Host server cần phải được thiết lập các thông số Hugepage để phù hợp với yêu cầu từ scheduler.

3. Hỗ trợ vhost-user vif type

- QEMU hỗ trợ vhost-user trong drivers của libvirt hỗ trợ Virtio-net I/O giữa máy ảo và host.
- Vhost-user có chức năng tương tự như /dev/vhost-net nhưng thay vì giao tiếp dựa trên kernel của vhost-net, vhost-user dựa trên Unix socket.
- Neutron driver cần phải hỗ trợ việc sử dụng vhost-user interface.
- Tạo vif type mới VIF_VHOSTUSER cho Nova trong việc tạo network interface dựa trên vhost-user.

4. I/O based NUMA

- Cho phép allocate memory từ memory bank gần nhất có thể
- Hỗ trợ cho các guests được gắn (attach) với host PCI device sử dụng các ưu điểm của NUMA, đặc biệt đối với NFV.

5. Sự khởi sắc của Hyper-V

- os_type="windows" được hỗ trợ cho ảnh (image) của máy ảo.
- Máy ảo chạy Windows hỗ trợ một vài tính năng của para-virtualization khi chạy trên nền Hyper-V.
- Hỗ trợ cho việc tránh xảy ra BSOD cho máy ảo, tăng hiệu năng của máy ảo.

6. Quiescent guest FS

- Nếu muốn tạo consistent backup của image, ta thường phải thao tác bằng tay việc freeze filesystem. Điều này sẽ làm tăng downtime, tạo ra sự ngắt quãng (disruption) trong quá trình sử dụng của người dùng.
- Tạo consistent snapshot thông qua libvirt bằng cách freeze filesystem.
- Sử dụng khi QEMU Guest Agent được kích hoạt trong instance.
- Yêu cầu:
 - Hypervisor là qemu hoặc kvm.
 - libvirt >= 1.2.5 (bao gồm fsFreeze/fsThaw API)
 - 'hw_qemu_guest_agent=yes', 'hw_require_fsfreeze=yes' được thiết lập trong quá trình tạo image

7. API microversion

- Version của Nova API ngày càng được cải tiến cùng với số lượng extensions tăng rất nhanh dẫn đến kích cỡ của API cũng ngày càng phình to.
- Cho phép DevOp có khả năng điều chỉnh Nova API để phù hợp với các yêu cầu về extension mới nhưng vẫn không loại bỏ các support cho các feature chung, cũ về API.
- Người dùng có thể quyết định sử dụng version nào của REST API phù hợp với yêu cầu về request cho từng extension.

8. Nova Cell

- Khái niệm Cell ra đời thay thế cho “Zone” trong Nova. Điểm khác biệt duy nhất so với Zone chính là Cell sử dụng AMQP để giao tiếp giữa các Cell.
- Trên mỗi API sẽ bao gồm AMQP broker (giao tiếp giữa các cell), database, nova-cells và nova-api.
- Khi các tác vụ Nova (boot, restart, reboot, etc.) được thực hiện, request sẽ được gửi tới Nova-api (trên Cell) và thông qua đó các quá trình tiếp theo của các tác vụ được thi hành trên các Cell.

II. Liberty

1. New VIF type cho phép routing VM data thay vì bridge it:

- 1 dạng mới của VIF là VIF_TYPE_TAP sẽ được libvirt tạo nhằm cho phép dữ liệu tới/từ các Vms cũng như giữa các Vms trên cùng 1 host sẽ không phải qua bridge. Nó hữu dụng trong trường hợp các Vms chỉ yêu cầu dịch vụ tại layer 3.
- Điều này hoàn toàn có thể được áp dụng thậm chí thông qua iptables, filter hoặc các securities policy. Check project: Project Calico (<http://www.projectcalico.org/>)

2. Mapping cell host in db of Nova:

- Table mới trong database of nova-api để lưu trữ mapping của host tới cell.

3. Khởi tạo RequestSpec Object:

- Problem: nova.scheduler.utils.build_request_spec() sẽ tạo ra một request_spec object được dùng cho method select_destinations() trong nova-scheduler interface. Nova-scheduler sẽ sử dụng get_filtered_host() method ở trong nova.scheduler.host_manager để gọi filter object cung cấp filter_properties chứa request_spec trong đó. Các filters sẽ modify request_spec trong quá trình filter các host và cấu trúc của request_spec không được ghi lại khiến cho quá trình scheduling trở nên vô cùng phức tạp và khó hiểu.
- Proposed change: Ta sẽ không dùng method build_request_spec với mục đích làm cho scheduler interface trở nên đơn giản và dễ hiểu hơn. Thay vào đó là refactor scheduling với việc tạo một class mới RequestSpec ở trong nova.objects.request_spec.RequestSpec.

4. Persist RequestSpec object

- Proposed: Sau khi quá trình resize hay migrate, instance sẽ được schedule tương tự như trước khi resize hay migrate. Do đó ta cần persist RequestSpec.

5. Support Cinder Volume Multi-Attach

Hiện nay, Nova chỉ hỗ trợ attach volume tới một instance tại một thời điểm. Làm cách nào để Nova cho phép attach volume tới nhiều instances.

- cinderclient only has volume as a parameter to the detach() call. This makes the assumption that a volume is only attached once.
- nova assumes that if a volume is attached, it can't be attached again. see nova/volume/cinder.py: check_attach()

Proposed change:

- Attach time: Nova phải remove việc chỉ cho phép volume được attach nếu volume ko ở trạng thái "in-use". Cinder sẽ chỉ cho phép volume được attach nhiều hơn 1 lần nếu ta có flag "shareable" được thiết lập tại thời điểm tạo volume.

- Detach time: Nova phải pass 1 parameter mới là attachment_uuid để xác nhận attachment nào được de-attach trong trường hợp multiple attachment. Attachment_id là optional trong cinderclient, nếu giá trị này không được pass trong trường hợp multi-attachment thì cinder sẽ fail vì không xác định được attachment nào được de-attach.

3-7-2015

VietStack Team