

cinder-volume多进程

背景：

在安装了cinder-volume的环境中，查看cinder-volume进程，可以发现多个！！

分析：

启动volume时，主进程会为每个backend创建一个子进程，配置了三个后端的环境进程信息如下。

cinder.conf配置：

```
default_volume_type = NetAppIscsiBackend
enabled_backends = NetAppIscsiBackend,NetAppIscsiBackend2,NetAppIscsiBackend3
```

```
[stack@localhost logs]$ cinder service-list
```

Binary	Host	Zone	Status	State	Updated_at	Disabled Reason	
cinder-scheduler	localhost.localdomain	linqingTEST	uuuuu	enabled	up	2017-05-16T04:34:02.000000	-
cinder-volume	localhost.localdomain@NetAppIscsiBackend	linqingTEST	uuuuu	enabled	up	2017-05-16T04:34:00.000000	-
cinder-volume	localhost.localdomain@NetAppIscsiBackend2	linqingTEST	uuuuu	enabled	up	2017-05-16T04:34:00.000000	-
cinder-volume	localhost.localdomain@NetAppIscsiBackend3	linqingTEST	uuuuu	enabled	up	2017-05-16T04:34:01.000000	-

```
[stack@localhost logs]$ ps aux|grep cinder-volume
stack 29403 1.8 1.3 496668 106220 pts/22 S+ 12:29 0:09 /usr/bin/python /usr/bin/cinder-volume --config-file /etc/cinder/cinder.conf
stack 29422 0.3 1.2 496324 99092 pts/22 S+ 12:29 0:01 /usr/bin/python /usr/bin/cinder-volume --config-file /etc/cinder/cinder.conf
stack 29427 0.3 1.2 498100 100812 pts/22 S+ 12:29 0:01 /usr/bin/python /usr/bin/cinder-volume --config-file /etc/cinder/cinder.conf
stack 29433 0.3 1.2 498348 101176 pts/22 S+ 12:29 0:01 /usr/bin/python /usr/bin/cinder-volume --config-file /etc/cinder/cinder.conf
stack 29630 0.0 0.0 112648 924 pts/24 R+ 12:38 0:00 grep --color=auto cinder-volume

[stack@localhost logs]$ pstree 29403 -p
cinder-volume(29403)
├── cinder-volume(29422)
├── cinder-volume(29427)
└── cinder-volume(29433)
```

结论：

在cinder-volume中每个backends对应一个进程，这些进程是启动进程的子进程。

Cinder-API的线程数可以通过osapi_volume_workers配置

Service对象的职责：

- 1.A service takes a manager and enables rpc by listening to queues based on topic.
- 2.periodically runs tasks on the manager
- 3.reports it state to the database services table.

启动代码分析

程序启动入口：cinder.cmd.volume.main

1.cinder会依次读取配置项enabled_backends中的后端名，根据该名称读取配置文件cinder.conf中的配置组，依次为每个back_end创建Service。

2.cinder.service.Service是oslo_service.Service的子类，运行在独立的子线程，每个后端配置对应一个Service。

3.Service的职责：

```
1      1).为每个服务创建cinder.volume.manager.VolumeManager
2          (VolumeManager包含了相关的cinder.volume.driver.BaseVD的派生类，而Bas
3      2).创建每个服务的RPC接口；
4      3).定义每个服务定时任务，包括上报数据库消息时间，以及定时任务时间。
5          涉及两个配置项：
6              report_interval
7              periodic_interval
```

4.Service对象在main方法中创建好后，会被传递到oslo_service的ProcessLauncher中，ProcessLauncher在原进程的基础上fork出子进程，并回调Service的start（）方法进行下一步工作！