***Cinder-Scheduler自定义Filter***

1. ***Cinder-Scheduler代码分析***

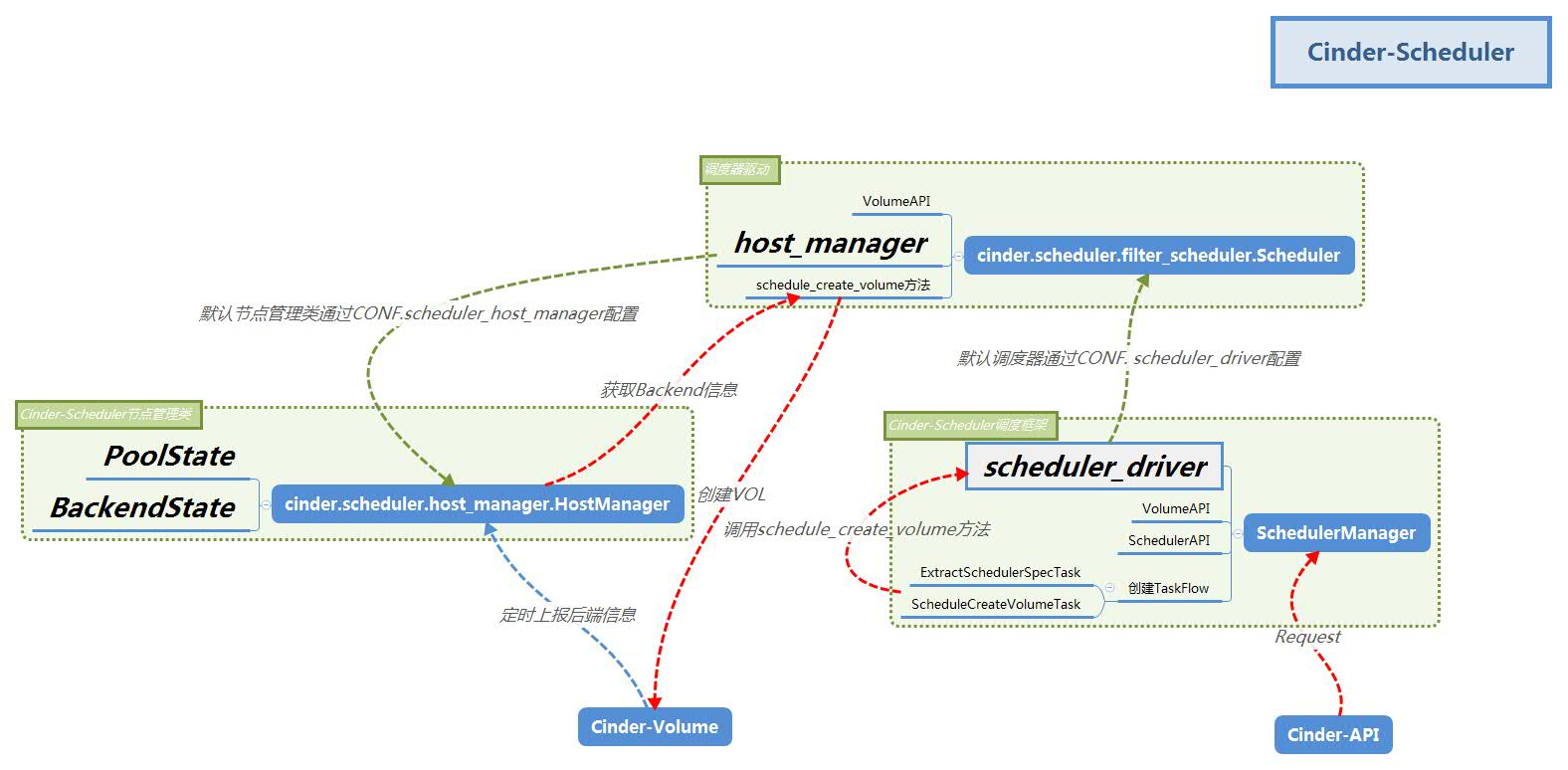
***在创建卷时，c-sch会启动一个名为volume\_create\_scheduler的TaskFlow进行HOST筛选，该流程包括两个Task：ExtractSchedulerSpecTask以及ScheduleCreateVolumeTask。***

***ExtractSchedulerSpecTask主要作用是重新包装从c-api发来的request\_spec，保证各个版本API接口的兼容性。ScheduleCreateVolumeTask是实际进行调度的Task。当创建卷时，ScheduleCreateVolumeTask会调用调度器驱动（scheduler\_driver）的schedule\_create\_volume方法对所有后端进行过滤、赋权最后选出最优的HOST。***

***Cinder-Scheduler是一个调度实现框架，要使其真正的实现调度功能需要提供对应的调度器驱动（CONF.*** ***scheduler\_driver），默认情形下Cinder使用的是官方提供的默认调度器cinder.scheduler.filter\_scheduler.FilterScheduler。***

***FilterScheduler为了对后端存储池进行优选，需要了解后端的存储信息，因此在内部维护了一个HostManager类型，用于记录存储池的使用信息（这些信息通过c-vol定时上报给c-sch保证时效，同样cinder get-pools命令获取的内容同样来自这个变量）。***

***Cinder-Scheduler卷创建调度流程：***



***过滤器的调用过程：***

***-->cinder.scheduler.flows.create\_volume.ScheduleCreateVolumeTask.execute()***

***-->cinder.scheduler.filter\_scheduler.FilterScheduler.schedule\_create\_volume()***

***-->cinder.scheduler.filter\_scheduler.FilterScheduler.\_schedule()***

***#对当前的Backend进行排序，并且选出最优的Backend***

***-->cinder.scheduler.filter\_scheduler.FilterScheduler.\_get\_weighted\_candidates()***

***#1.过滤Backend流程***

***#获取当前的所有Backend***

***-->cinder.scheduler.host\_manager.HostManager.get\_all\_backend\_states()***

***#对获取到Backend进行过滤***

***-->cinder.scheduler.host\_manager.HostManager.get\_filtered\_backends()***

***#热加载当前过滤器***

***-->cinder.scheduler.host\_manager.HostManager.\_choose\_backend\_filters()***

***#对Backend进行过滤***

***-->cinder.scheduler.filters.\_\_init\_\_.BackendFilterHandler.get\_filtered\_objects()***

***#2.排序Backend，且获取权值最大的***

***-->cinder.scheduler.host\_manager.HostManager.get\_weighed\_backends()***

1. ***自定义Filter***

***自定义Filter模块的py文件定义在cinder.scheduler.filters包下，并且继承cinder.scheduler.filters.BaseBackendFilter，需要实现以下方法：***

**def backend\_passes**(self, host\_state, filter\_properties):  
 *"""Return True if the HostState passes the filter, otherwise False.  
  
 Override this in a subclass.  
 """* **raise** NotImplementedError()

***其中，host\_state是一个cinder.scheduler.host\_manager.PoolState对象（在对接netapp iscsi方案下，PoolState的HOST值为XXX@YYYY#ZZZZ），包括所有能从cinder get-pools命令获取的信息。filter\_properties表示当前的过滤参数，包括qos\_specs、request\_spec、volume\_type、scheduler\_hints、context等选项。***

1. ***在生产环境部署自定义过滤器***

***由于过滤器是通过Entry\_Point方式热加载到Cinder-Scheduler中的，因此部署过滤器时除了需要在c-sch节点配置CONF.scheduler\_default\_filters外，还需要添加Entry\_Point。***

***添加Entry Point有两种情形：***

1. ***已经部署完成的节点，在cinder/cinder.egg-info/entry\_points.txt中，添加***

***[cinder.scheduler.filters]***

***...***

***DisableFilter = cinder.scheduler.filters.disable\_filter:DisableFilter***

1. ***对于开发的发布包，需要在cinder/setup.cfg添加对应的过滤器***

***4. 过滤指定的Backend的过滤器方案***

***方案一：通过配置文件的形式实现过滤指定HOST名，PoolState.(已经基本实现提交到GIT multilocation分支)***

***方案二：通过数据库读取字段的方式实现过滤***