# 代码逻辑

接口类：***cinder.scheduler.host\_manager.HostManager***

接口方法：***get\_weighed\_backends***

def get\_weighed\_backends(self, backends, weight\_properties,  
 weigher\_class\_names=None):  
 """Weigh the backends."""  
 """加载所有weighter类"""  
 weigher\_classes = self.\_choose\_backend\_weighers(weigher\_class\_names)  
 """计算过滤器的权"""  
 return self.weight\_handler.get\_weighed\_objects(weigher\_classes,backends,weight\_properties)

上述代码中，\_choose\_backend\_weighers()负责加载配置项scheduler\_default\_weighers中的所有配置的weigher。c-sch在启动时，通过stevedore包以插件的方式导入cinder.scheduler.weights包路径下的所有class到HostManager的weight\_classes对象中，

scheduler\_default\_weighers中的所有配置的weigher，必须存在于HostManager.weight\_classes中才能正常使用。

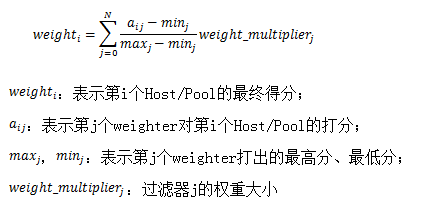
weight\_handler对象实际上是***cinder.scheduler.weights.OrderedHostWeightHandler类***（通过CONF.scheduler\_weight\_handler可以配置），他负责依次调用过滤器的weigh\_objects方法，对HOST/Pool进行排序。

在OrderedHostWeightHandler中每个Host/Pool（PoolState类）通过一个WeighedObject表示，该对象中self.obj成员存储PoolState类型，self.weight记录得分，得分最高的Host/Pool会被选中。

OrderedHostWeightHandler类的**get\_weighed\_objects方法**

def get\_weighed\_objects(self, weigher\_classes, obj\_list,  
 weighing\_properties):  
 """Return a sorted (descending), normalized list of WeighedObjects."""  
 if not obj\_list:  
 return []  
 weighed\_objs = [self.object\_class(obj, 0.0) for obj in obj\_list]  
 for weigher\_cls in weigher\_classes:  
 weigher = weigher\_cls()  
 weights = weigher.weigh\_objects(weighed\_objs, weighing\_properties)  
 # Normalize the weights  
 weights = normalize(weights,  
 minval=weigher.minval,  
 maxval=weigher.maxval)  
 for i, weight in enumerate(weights):  
 obj = weighed\_objs[i]  
 obj.weight += weigher.weight\_multiplier() \* weight  
  
 return sorted(weighed\_objs, key=lambda x: x.weight, reverse=True)

分数的计算规则：



OrderedHostWeightHandler会将打分的结果从大到小顺序输出，而StochasticHostWeightHandler（通过CONF.scheduler\_weight\_handler可以配置）和他的功能类似，但是会将排序结果打乱，随机输出。

# 自定义weighter

派生***BaseHostWeigher***类型，实现以下方法：

1. ***\_weigh\_object***方法：对单个Backend打分；
2. ***weigh\_objects***方法：对多个Backend打分（这个方法可以不实现）；
3. ***weight\_multiplier***方法：返回权；

将py文件打包到cinder.scheduler.weights，并在setup.cfg或者cinder.egg-info/entry\_points.txt添加entry\_point入口。