

Huawei Storage实现了自身的migrate方法，不依赖于Cinder本身的_migrate_volume_generic进行迁移。

HuaweiBaseDriver的migrate_volume方法实现以下逻辑：

1. 校验迁移卷是否存在于当前Backend; ----- migrate_volume
2. 检查迁移卷的卷类型是否包含replication_enabled=true, 如果是那么返回, 并且调用_migrate_volume_generic进行迁移; ----- migrate_volume
3. 检查设备一致性, 通过location_info, 该参数时华为存储的物理设备参数。换句话说, 迁移卷迁移前后, 任然必须在同一个物理设备上; ----- _check_migration_valid
4. 后端创建一个临时的LUN, 该LUN为位于target_pool, 并且名字是源卷的hash码, 如: '-8007425411870660189'; ----- _migrate_volume
5. 发送迁移的Resful_API: ----- create_lun_migration

```
def create_lun_migration(self, src_id, dst_id, speed=2):
    url = "/LUN_MIGRATION"
    data = {"TYPE": '253',
            "PARENTID": src_id,
            "TARGETLUNID": dst_id,
            "SPEED": speed,
            "WORKMODE": 0}

    result = self.call(url, data, "POST")
    msg = _('Create lun migration error.')
```

6. 通过get_lun_migration_task()接口轮询迁移是否完成 ----- _is_lun_migration_complete

7. 删除临时LUN (推测这里storage内部将src_lun和des_lun的ID进行了互换, 因此最后完成迁移时删除的是临时卷而不是源卷)。

整个迁移过程, Cinder不感知volume的任何变化, 迁移前后只有host字段发生了改变。

PS: 关于该迁移方式下卷的迁移速度

迁移速度相当慢, 3GB的卷迁移了一分多钟, 报文中SPEED参数似乎是速度限制, 但是这个参数不可调整, 并且修改代码调整后报错!