/neat/locals/manager.py

每轮迭代的工作：

1. 读取宿主机上VM的资源用量数据
2. 根据配置，执行低载判断算法
3. 若低载，向global manager发送迁移请求
4. 否则，执行过载判断算法，若过载，执行vm选择算法，向global manager发送迁移请求
5. 否则，仅更新状态

伪代码：

start:开始local-manager的循环

|  |
| --- |
| 1. 读取配置文件 2. 初始化日志，并记录程序开始 3. 使用common提供的接口开始local manager作为返回值 |

init\_state：初始化local-manager的state词典

|  |
| --- |
| 1. 创建libvirt连接 2. 利用libvirt读取cpu核数与主频，计算宿主机总cpu资源 3. 创建词典记录state信息并返回 |

execute:local-manager执行一轮迭代

输入参数：配置文件、state

返回值：更新后的state

|  |
| --- |
| 1. 日志记录开始一轮迭代 2. 获取宿主机上vms存储路径，读取vms数据 3. 获取当前每个vm的uuid及最大RAM 4. 将不再此宿主机上的vm从vm cpu用量列表中移除 5. 若vm cpu用量列表为空，记录空闲，返回state 6. 获取主机cpu用量信息 7. 根据vm cpu用量计算宿主机cpu用量百分比 8. 将信息记入日志中 9. 若没有宿主机cpu用量，则记录日志，返回state 10. 获取数据收集时间间隔，计算每个vm迁移时间 11. 若state中无低载检测 12. 从config中读取低载检测算法 13. 使用common，通过接口获取低载检测函数 14. 更新state词典：低载算法函数、是否低载 15. 从config中读取过载检测算法 16. 使用common，通过接口获取过载检测函数 17. 更新state词典：过载算法函数、是否过载 18. 从config中读取vm选择算法 19. 使用common，通过接口获取vm选择函数 20. 更新state词典：vm选择算法函数、vm选择的状态 21. 否则 22. 从state中读取低/过载检测算法、vm选择算法函数 23. 开始低载检测，使用state中的函数进行检测 24. 记录相应日志 25. 开始过载检测，使用state中的函数进行检测 26. 记录相应日志 27. 若低载 28. 日志记录检测到低载 29. 向global-manager发送请求：reason = 0 30. 等待global-manager的回复，记录日志 31. 若过载 32. 日志记录检测到低载 33. 根据vms的cpu用量与ram，执行vm选择算法 34. 记录相关日志 35. 向global-manager发送请求：reason = 1，迁移决策 36. 等待global-manager的回复，记录日志 37. 否则 38. 记录日志，没有检测到过载或低载 39. 记录日志完成一次迭代 40. 返回状态 |

/neat/common.py

核心工作：

读取宿主机上VM的资源用量数据

伪代码：

Start

输入参数：初始化函数init\_state, 执行函数execute, 配置文件，interval

返回值：state

|  |
| --- |
| 1. 利用config和it\_state对state进行初始化 2. 若未指定迭代次数 3. 死循环 4. 执行函数execute，参数为config与state 5. 睡眠interval时间 6. 否则 7. 有限次迭代 8. 执行函数execute，参数为config与state 9. 睡眠interval时间 10. 返回state |

/neat/globals/manager.py

负责工作：

1. 从local-manager接收请求
2. 通过接口产生vm放置决策，初始化迁移过程，向nova递交请求
3. 将主机切换至睡眠模式，及唤醒主机

* 收到一个local-manager请求后：
  + 解析vm\_uuids参数，转为迁移vm的UUID列表
  + 调用nova api获取当前vm的放置情况
  + 调用vm放置算法，将迁移vm的UUID和当前vm放置情况作为参数
  + 根据vm放置算法的结果，调用nova api进行vm迁移
* 宿主机切换睡眠模式时，global-manager通过配置文件打开ssh连接，执行sleep\_command的命令
* 宿主机需要唤醒时，使用Wake-on-LAN技术，通过ether-wake发送magic packet，mac地址作为参数（mac地址与ip的映射在global-manager初始化时产生）

start

|  |
| --- |
| 1. 验证并读取配置文件 2. 初始化日志 3. 初始化状态列表 4. state列表中休眠宿主机切换到活跃模式switch\_hosts\_on 5. 调用bottle库，开始监听请求 |

service

监听到put方法时执行，且仅允许put方法

|  |
| --- |
| 1. 获取请求的参数、状态 2. 验证参数 3. 记录日志 4. 若参数reason==0，则为低载检测 5. 记录日志 6. 执行低载函数，参数为config, state, host信息 7. 否则，为过载检测 8. 记录日志 9. 执行过载函数，参数为config, state, host, vm\_uuid 10. 若产生错误，则记录日志 |

switch\_hosts\_on

输入参数: db, etherwake接口, 主机mac地址列表, 主机列表

|  |
| --- |
| 1. 对唤醒主机列表中主机 2. 若主机不在mac地址列表中 3. 获取主机mac地址 4. 创建唤醒cmd: etherwake -I interface mac地址 5. 记录日志 6. 进行唤醒 7. 记录日志 8. 将结果放入neat数据库 |

execute\_underload

输入参数: config, state, host(name)

|  |
| --- |
| 1. 记录日志：开始underload请求 2. 读取低载宿主机及其资源用量 3. 通过nova获取该宿主机上vm列表 4. 通过neat\_db获取最后的vm列表其资源用量 5. 从宿主机的vm列表中移除数据仍未被收集的新vm 6. 记录日志 7. 若宿主机上有vms 8. 对于有vm不在neat记录列表中的宿主机 9. 记录日志 10. 将宿主机加入”active”列表 11. 记录宿主机cpu用量 12. 否则 13. 该宿主机是不活跃的，将其排除 14. 记录日志：主机cpu用量，排除低载主机的信息 15. 开始排除低载主机 16. 获取要迁移的vms 17. 对于每个vm 18. 若存在新的vm，其信息未记录在neat中 19. 记录日志 20. 返回state 21. 记录获取该vm的cpu用量 22. 记录这些vms的ram用量 23. 迁移vm列表中移除不在nova提供的ram用量列表中的vm 24. 若迁移vm列表为空 25. 记录日志 26. 返回state 27. 记录时间间隔，计算迁移时间 28. 若state中无vm放置 29. 从config中读取vm放置算法 30. vm放置state为None 31. 用common，通过接口获取vm放置函数 32. 记录在state中 33. 否则 34. 从state中读vm放置函数及放置state 35. 记录日志：开始低载vm放置 36. 通过vm放置算法获取放置策略 37. 获取活跃宿主机、睡眠宿主机、之前不活跃宿主机、切换到睡眠宿主机 38. 若不存在放置策略 39. 记录日志 40. 将宿主机移出要切换到睡眠宿主机列表 41. 否则 42. 记录日志 43. 根据策略开始迁移 44. 若存在要切换到睡眠宿主机 45. 将这些宿主机切换到睡眠模式 46. 记录日志：完成一次低载请求 47. 返回state |

execute\_overload

输入参数: config, state, host(name),vm\_uuids

|  |
| --- |
|  |

migrate\_vms

输入参数：db, nova, vm实例词典，放置策略，是否使用块迁移

|  |
| --- |
| 1. 初始化重试迁移词典 2. 从迁移策略中获取迁移vm 3. 对vm\_pairs列表中每个vm\_pair 4. 开始计时 5. vm\_pair中的每个vm\_uuid 6. 迁移当前vm 7. 进程睡眠10s 8. 死循环 9. 对vm\_pair中每个uuid 10. 根据uuid获取nova的vm实例 11. 记录日志 12. 若迁移决策中的主机名等于vm当前宿主机名，且vm是活跃的 13. neat中记录迁移信息 14. 记录日志 15. 若迁移时间大于300s，且迁移决策中的主机名不等于当前vm宿主机名，且vm是活跃的 16. vm放入迁移重试队列 17. 记录日志 18. 否则跳出死循环 19. 否则跳出死循环 20. 进程睡眠3s 21. 若有策略中有重试vm 22. 记录日志 23. 递归调用自身，进行重试 |

migrate\_vm

输入参数：nova, vm实例列表, 迁移vm的uuid, 目的主机名, 是否使用块迁移

|  |
| --- |
| 将vm实例所在文件夹所有者改为nova  使用novaclient的api迁移vm  记录日志 |