Neat实验文件目录

home

└─ubuntu

├─ccpe-2014-experiments

│ ├─setup

│ │ ├─create-flavor.sh //创建pico flavor

│ │ ├─deps-arch.sh //pip更新bottle, requests包

│ │ └─deps-centos.sh //安装python-pip及pip安装bottle, requests包

│ ├─full-utilization //用于分发负载，28条满cpu用量

│ ├─planetlab-filtered //33条符合筛选条件的cpu用量

│ ├─planetlab-selected //用于分发负载，33条中随机的28条cpu用量

│ ├─filter-traces.sh //设置traces的筛选条件，对planetlab进行筛选

│ ├─workload-filter.py //根据配置条件对数据文件进行traces筛选

│ ├─workload-selector.py //随机选择指定数量的traces文件

│ │

│ ├─vms-boot-8.sh //创建8个vm

│ ├─vms-boot-8-slow.sh //创建8个vm，间隔5s

│ ├─vms-boot-28-slow.sh //创建28个vm，间隔5s

│ ├─vms-delete-8.sh //删除8个vm

│ ├─vms-delete-28.sh //删除28个vm

│ ├─vms-restart-28.sh //删除并慢创建28个vm

│ │

│ ├─workload-distributor.conf //标识是否分发负载，1分发，0不分发

│ ├─disable-distributor.sh //禁止分发负载（workload-distributor.conf置0）

│ ├─enable-distributor.sh //允许分发负载（workload-distributor.conf置1）

│ ├─workload-distributor.py //分发工作负载

│ ├─workload-start.py //开始产生负载

│ ├─start-distributor-full.sh //开始分发满cpu负载

│ ├─start-distributor-planetlab.sh //开始分发planetlab负载

│ ├─workload-receiver.sh //根据日志显示成功接收负载的文件

│ └─results

│ ├─execution-times.py //计算execution-time

│ ├─do-execution-times.py //计算execution-times

│ ├─energy.py //计算整合/最大/节约能耗

│ ├─idle-time-fraction.py //计算ITF

│ ├─overload-time-fraction.py //计算OTF

│ ├─times-per-host.py //计算AITF与AOTF

│ ├─vm-migrations.py //计算vm迁移数量

│ ├─db.py //数据库操作接口程序

│ ├─db-drop.sh //清空数据库

│ ├─db-dump.sh //导出数据库

│ ├─db-restore.sh //用sql文件恢复数据库

│ ├─

│ ├─do-one-per-host.sh //根据一个宿主机日志计算metrics与execution-times

│ ├─do-one.sh //根据所有宿主机日志计算metrics与execution-times

│ ├─do-all-per-host.sh //根据一个宿主机日志计算AITF与AOTF

│ ├─do-all.sh //产生metrics.csv与execution-times.csv

│ ├─do-metrics.sh //计算ITF, OTF与vm迁移数量

│ ├─

│ ├─parse-execution-times.sh //解析执行时间

│ ├─make-package.sh //将计算节点日志拷贝到控制节点，打包

│ ├─package-start-time.sh //将控制节点日志添加到tar.gz

│ ├─

│ ├─energy.txt //所有算法的能耗结果

│ ├─metrics.csv

│ ├─execution-times.csv

│ ├─metrics-per-host.txt //mhod, lrr算法的结果

│ ├─times-per-host.txt //mhod, otf算法的结果

│ ├─ //不同算法的实验结果压缩文件

│ ├─best-idf-[1, 2, 3].tar.gz

│ ├─lr-1.0-[1, 30].tar.gz

│ ├─lrr-0.8-[0.9, 1.0]-[1, 2, 3].tar.gz

│ ├─lrr-[0.9, 1.0, 1.1]-[1, 2, 3].tar.gz

│ ├─max-idf-[1, 2, 3].tar.gz

│ ├─mhod-0.8-[0.20, 0.30, 0.40, 0.50]-0.8-[1, 2, 3].tar.gz

│ ├─otf-0.8-[0.1, 0.2, 0.3].tar.gz

│ ├─thr-[0.8, 0.9, 1.0]-[1, 2, 3].tar.gz

│ ├─thr-2-0.8-[1, 2, 3].tar.gz

│ └─old

├─cpu-load-generator

│ ├─README.md //安装、使用及样例

│ ├─cpu-load-generator.py //生成cpu负载

│ ├─install-lookbusy.sh //安装lookbusy工具

│ └─test.data //用作测试的cpu负载

└─openstack-neat

├─doc //neat文章

│ └─blueprint

│ ├─src

│ ├─openstack-neat-blueprint.epub

│ ├─openstack-neat-blueprint.html

│ └─openstack-neat-blueprint.pdf

├─init.d //服务程序

│ ├─openstack-neat-global-manager

│ ├─openstack-neat-db-cleaner

│ ├─openstack-neat-local-manager

│ └─openstack-neat-data-collector

├─neat

│ ├─globals

│ │ ├─vm\_placement

│ │ │ ├─\_\_init\_\_.py

│ │ │ └─bin\_packing.py //基于装箱的vm放置算法

│ │ ├─\_\_init\_\_.py

│ │ ├─db\_cleaner.py //数据库清洗模块

│ │ └─manager.py //global-manager模块

│ ├─locals

│ │ ├─overload

│ │ │ ├─mhod //mhod算法

│ │ │ │ ├─\_\_init\_\_.py

│ │ │ │ ├─bruteforce.py //用暴力破解解决NPL问题

│ │ │ │ ├─core.py //MHOD算法主模块

│ │ │ │ ├─l\_2\_states.py //MHOD算法的双状态配置的L函数

│ │ │ │ ├─multisize\_estimation.py //工作负载估计算法

│ │ │ │ └─nlp.py //定义MHOD算法的NPL问题模型

│ │ │ ├─\_\_init\_\_.py

│ │ │ ├─otf.py //基于OTF阈值的算法

│ │ │ ├─statistics.py //基于统计的过载检测算法

│ │ │ └─trivial.py //琐碎的过载检测算法

│ │ ├─underload

│ │ │ ├─\_\_init\_\_.py

│ │ │ └─trivial.py //琐碎的低载检测算法

│ │ ├─vm\_selection

│ │ │ ├─\_\_init\_\_.py

│ │ │ └─algorithms.py //迁移vm选择算法

│ │ ├─\_\_init\_\_.py

│ │ ├─collector.py //数据收集模块

│ │ └─manager.py //local-manager模块

│ ├─\_\_init\_\_.py

│ ├─common.py //组件的共享函数

│ ├─config.py //读取、验证并返回配置文件

│ ├─contracts\_extra.py

│ ├─contracts\_primitive.py

│ ├─db.py //neat使用的对数据库的操作

│ └─db\_utils.py //初始化数据库中neat表

├─setup

│ ├─deps-arch.sh //arch安装依赖包

│ ├─deps-centos.sh //centos安装依赖包

│ ├─update-chkconfig.sh //向系统中增加global/local-manager, db-cleaner, data-collector服务

│ └─README.md //指导如何为neat创建数据库和用户

├─utils

│ ├─db.py //neat使用的对数据库的操作

│ ├─idle-time-fraction.py //计算指定时间内ITF

│ ├─overload-time-fraction.py //计算指定时间内OTF

│ └─vm-migrations.py //计算指定时间内vm迁移数量

├─all-clean-logs.sh //清除计算节点所有日志，再清除控制节点所有日志

├─compute-clean-logs.py //清除计算节点所有日志

├─all-sync-time.sh //重启计算节点ntpdata，再重启控制节点ntpdata

├─compute-sync-time.py //计算节点配置并重启ntpdata服务

├─all-update.sh //计算节点git clone并安装neat，控制节点安装neat

├─compute-update.py //计算节点git clone并安装neat

├─

├─setup.py //安装neat

├─distribute\_setup.py

├─compute-clone-neat.py //让计算节点git clone neat

├─compute-install-deps.py //让计算节点安装依赖包

├─compute-install-neat.py //让计算节点安装neat

├─compute-copy-conf.py //将neat.conf拷贝给计算节点

├─neat.conf //neat的配置文件

├─

├─all-start.sh //开始所有global/local-manager, data-collector, db-cleaner

├─all-stop.sh //停止所有global/local-manager, data-collector, db-cleaner

├─compute-data-collector-start.py //计算节点启动data-collector服务

├─compute-data-collector-status.py //计算节点查看data-collector服务状态

├─compute-data-collector-stop.py //计算节点停止data-collector服务

├─compute-local-manager-start.py //计算节点启动local-manager服务

├─compute-local-manager-status.py //计算节点查看local-manager服务状态

├─compute-local-manager-stop.py //计算节点停止local-manager服务

├─start-data-collector.py //执行data-collector.start()

├─start-global-manager.py //执行global-manager.start()

├─start-local-manager.py //执行local-manager.start()

├─

├─stats.sh //计算指定时间内的OTF, ITF及vm迁移数量

├─vm-placement.py //监控当前vm部署情况

├─

├─install-rpm.sh //安装bdist\_rpm并重装openstack-neat-0.1-1.noarch.rpm

├─build-rpm.sh //安装bdist\_rpm

├─nosetests.sh //测试安装包

├─TODO //创建rpm包的流程

├─MANIFEST.in

├─README.md //Neat开发信息，部署指导

└─tests //测试用的算法模块

├─globals

│ ├─vm\_placement

│ │ ├─\_\_init\_\_.py

│ │ └─test\_bin\_packing.py

│ ├─\_\_init\_\_.py

│ ├─test\_db\_cleaner.py

│ └─test\_manager.py

├─locals

│ ├─overload

│ │ ├─mhod

│ │ │ ├─\_\_init\_\_.py

│ │ │ ├─test\_bruteforce.py

│ │ │ ├─test\_core.py

│ │ │ ├─test\_core\_real\_data.py

│ │ │ ├─test\_l\_2\_states.py

│ │ │ ├─test\_multisize\_estimation.py

│ │ │ └─test\_nlp.py

│ │ ├─\_\_init\_\_.py

│ │ ├─test\_otf.py

│ │ ├─test\_statistics.py

│ │ └─test\_trivial.py

│ ├─underload

│ │ ├─\_\_init\_\_.py

│ │ └─test\_trivial.py

│ ├─vm\_selection

│ │ ├─\_\_init\_\_.py

│ │ └─test\_algorithms.py

│ ├─\_\_init\_\_.py

│ ├─test\_collector.py

│ └─test\_manager.py

├─resources

│ └─vms //空文件

├─\_\_init\_\_.py

├─test\_common.py

├─test\_config.py

├─test\_db.py

└─test\_db\_utils.py