

pymonitor

Crack by Zhixin Li

核心类

- `class configFile(object)`
- `class ZDatabase(object)`
- `class cronList(object)`
- `class projectClass`
- `class jobClass`
- `class directExec`
- `class Report`

核心函数

- `def projectstat()`
- `def projectaction()`
- `def projectimport()`
- `def removefinishedprojects()`
- `def removebrokenproject()`
- `def cronjob()`
- `def directexec()`
- `def setdefault()`

现存问题

- 代码模块太老
- Python2
- 代码模块间的独立性不够
- 没有一个检测脚本运行状态的测试脚本

数据存储

- 1.home目录下的配置文件
- 2.ZODB数据库

1.读取或创建配置文件

- 1.找到home下的配置文件"~/.pymonitor.conf"
- 2.判断是否需要更新该配置文件
- 如果文件不存在，那必然要全部创建默认
- 如果文件存在，那就读取，判断，不存在的才do
- 3.我将这个check封装为一个函数
- def addPrj(self, prjname, prjdb)
- def removePrj(self, prjname)
- 先不管

2.读取命令行参数

- Optparse模块
- 将所有命令参数读入optlist对象中，并添加默认参数
- 根据args来决定调用哪一个主函数

3.调用projectimport()主函数

- 1.参数检查
- 2.检查是否有重复的project
- 3.最终得到projectlocation，即数据库地址
- 4.将项目信息写入到配置文件对象addPrj

3.1 调用ConfigFileObj.addPrj

- 1.如果没有test1.db.lock文件，在home目录底下创建一个~/.pymonitor.conf.lock.test1
- 2.并建立软链接
- 3.如果项目里没有任何项目[project]，则创建cronobj对象，并cronobj.addCron()
- 4.将项目信息写入到配置文件
self.cf.set('project', prjname, prjdb)

3.1.1 cronobj.addCron()

- 首先会读取monitor.sh的地址PYMONITOR_SH_PATH为program
- 1.读取之前的节点（可能为空）和现在的
- 2.判断写入配置文件
- 3.读取crontab，判断是否需要添加cron
- 4.添加2个cron周期检查jobs和更新数据库

3.2 开始创建ZODB数据库

- 已经有了DB的位置和名字（下面我会改写）
- 1.查看数据库是否存在，是否损毁
- 2.打开数据库，返回dbroot
- 3.创建一个projectClass(zodbroot)，放到projectobj下
- ...
- 4.lock文件，进程锁

3.2.1 创建projectClass(zodbroot)

- 使用新类class projectClass(Persistent)
- 这是monitor最核心的类
- 1.写入基本的参数
- 2. projectobj.ImportTaskmonitor导入任务

3.2.1.1 def ImportTaskmonitor

3.2.1.2 OOBTree

- Just a dict in ZODB

projectClass类

- 一个项目类，用于组织属性和功能，并存储在ZODB内
- 这不是普通类，它只能用ZODBroot来创建，就是为了存储
- 类属性：
- projectName = “ 项目名
- param_P = “ HuM
- param_q = “ bc.q
- maxJobs = 0 最大任务数
- currentJobs = 0 当前任务数
- finishMark = “ 结束标记
- allFinished = 0 是否全部完成
- DiskWarning = 0 磁盘报警
- submitEnabled = 1 当前是否允许投递

project类的核心方法

- ImportTaskmonitor 解析alldep文件，导入到数据库
- ImportQsubsg
- Update 检查是否完成，磁盘，提交任务
- checkDiskSpace
- getStat
- submitNew
- SuspendProject
- ResumeProject
- DeleteProject
- clearErrorState
- writeLog
- clearLog

ZODB存储

- `zodbobj = ZDatabase()`
- `zodbobj.Init("*.db")`
- `zodbroot = zodbobj.Open()`
- `zodbroot.keys()`
 - `{'errorlog': ['2017-03-03 19:55:45 log_cleared']}`,
 - `'jobs': <BTrees.OOBTree.OOBTree object at 0x7f745566a268>`,
 - `'projectobj': <pymonitor.projectClass object at 0x7f7455679d68>}`
- `zodbroot['jobs']['./MScheckByBam.sh']`
 - Dependent
 - `projectObj`

