

runjob 使用文档

版本: v1.3.17

作者: 邓勇

日期: 2019. 11. 25

runjob 是一款高效的用于大量任务投递管理与实时监控的软件，针对任务投递做了定制化的处理，可以用于集群环境和本地服务器任务管理和灵活监控，亦用于研发或测试人员对命令行任务的分批量投递和管理，可有效解决测试任务排队过多影响集群任务投递效率的问题，软件说明如下：

1. 依赖模块：

runjob 采用 python 语言编写，依赖于以下环境：

- (1) python 2.7
- (2) psutil (python 模块包)

2. 安装或更新

安装方式为通用的 python 第三方软件包安装方法，可直接通过 pip 安装，

```
pip install runjob==1.3.17
```

或者通过 git 站点 <https://github.com/yodeng/runjob.git> 安装，

```
pip install git+https://github.com/yodeng/runjob.git
```

pip 安装方式会自动下载和安装依赖包 psutil

3. 使用说明

runjob 一共包含 3 个子命令，runjob、runstate 和 runsge，各命令使用如下：

3.1 runjob

runjob 针对 job 形式的任务进行本地或集群环境的管理和投递，投递前会检查 job 文件中的依赖和指定任务是否存在错误，投递后主程序在集群或本地环境下均几乎不占用 cpu 或内存资源。

3.1.1 参数介绍

选项	说明
-h	打印帮助并退出程序
-n	同时投递和运行的最大任务数，默认为 1000，当指定值小于 100 时，开启任务限制
-j	输入的 job 任务文件
-s	指定 job 文件任务开始的行号，默认第一行
-e	指定 job 文件任务结束的行号，默认最后一行
-r	重投次数设置，某个任务运行失败后进行重新投递，默认重新投递 3 次
-ivs	任务失败后重投的时间间隔，默认 2 秒
-m	任务运行的模式，可选 sge 或 localhost，默认 sge，同时会检测当前环境，若没有 sge 环境，则全部任务强制使用本地 localhost 环境，若有 sge 环境，会根据 job 中的 host 指定环境投递任务，host 默认为 sge
-nc	runjob 主进程终止后，不继续投递任务，同时清空由该进程投递的正在运行的任务，默认开启，如不想清空，可添加该选项
--strict	该参数表示一旦有任务失败，且重投仍然失败，则停止继续投递，终止 runjob 主进程。默认不终止主进程，继续投递任务。
-l	任务状态的实时输出日志，默认标准输出
-i	指定投递的任务名，可接多个参数，支持正则指定
-v	打印软件版本并退出

3.1.2 任务运行状态说明

(1) 本地或集群任务运行过程中会实时（可能会有 1 秒的延迟）输出每个任务的运行状态，任务状态包括 run, submit, resubmit, error, success, wait, killed, already success 这几种，一旦任务状态发生改变时会实时记录下来，各状态意义如下：

- a) run 表示任务正在运行中
- b) submit 表示任务已经投递，还没有开始运行（排队状态）
- c) resubmit 表示任务失败之后，正在重新投递，还没有开始运行
- d) error 表示任务运行完成，且任务运行错误
- e) success 表示任务运行完成，且任务运行成功，运行成功的任务则不

在运行

- f) `wait` 表示任务已经记录，正在等待投递
- g) `killed` 表示任务正在运行的时候，由于 `runjob` 主进程的终止，并被杀死的任务
- h) `already success` 每次投递任务前会检查当前已经运行成功的任务，如果运行成功，则不再运行该任务，此状态表示投递前已经运行成功的任务

3.1.3 任务重投说明

如若发生第一次 `job` 文件的任务运行完成，并且有任务重投运行之后还是失败，需要重新投递的情况，则需要重新使用 `runjob` 运行该 `job` 文件，命令同第一次运行的命令一样即可，已经运行成功的任务不会继续运行，运行失败的任务会根据依赖关系自动投递

3.1.4 任务 log 说明和报错查看

每个 `runjob` 通过 `-l` 参数指定 log 输出，默认标准输出，实时记录当前 `runjob` 的每个任务状态。同时，每个任务的实时日志输出存放在 `$log_dir` 下，以任务名+“`.log`”命名，记录了任务投递时间、排队时间、运行时间、终止（成功或失败）时间以及该任务的所有标准输出和标准错误输出，可以从该文件中查看任务的输出日志，用于报错解决。

3.1.5 任务文件 `job` 和 `order` 的说明

- a) 任务 `job` 的定义方式不变，支持多行 `cmd`

- b) order 定义方式可以用任务名的 after 或者 before 来指定，支持一行给定多个 order 依赖，如“order jobA after jobB jobC jobD”表示任务 A 同时依赖任务 B、C、D，“order jobA before job jobC jobD”表示任务 B、C、D 都依赖任务 A

3.2 runstate

runstate 用于实时统计当前集群任务状态和某个 runjob/runsge 进程的任务状态，若没有集群环境，则可统计当前服务器的所有用户进程和 runjob/runsge 进程的任务状态，用法如下：

3.2.1 runstate + logfile

logfile 即为 runjob 的-l 参数指定的输出 log 文件，可快速统计集群任务状态以及某个 runjob 的任务运行情况，并显示任务名字，可用于实时查看 job 文件中运行任务的情况，便于知道任务进度

3.2.2 runstate + log_dir

log_dir 即为 jobfile 中指定的 log_dir，可统计该 log_dir 中所有已经投递的任务的运行状态

3.2.3 runstate + jobfile

jobfile 即为运行的 job 文件，可统计该 job 文件中所有的任务状态信息。

3.3 runsge

runsge 是针对研发或测试人员大量任务投递或测试的程序，通过任务池队列动态投递任务，可高效管理任务的批量投递，不至于影响集群效率。使用说明如下：

3.3.1 参数介绍

选项	说明
<shellfile>	位置参数，直接传入 shell 文件即可，shell 文件中的每一行当做一个任务投递到集群。
-h	打印帮助信息并退出
-q	任务投递的队列名，默认 all.q
-m	任务投递使用的内存，默认 1G
-c	任务投递使用的 cpu 核数，默认 1 个
-wd	任务投递的目录，默认为当前目录
-N	任务名，每行任务的名默认为任务名+行号后缀，任务名默认为 shell 文件名
-o	所有任务的 log 输出目录，记录每个任务的日志输出，默认为“qsub.out.”+shell 文件名
-n	同时运行和排队的最大任务数限制，默认为所有任务个数
-s	从 shell 文件的第几行开始投递，默认为第一行
-e	从 shell 文件的第几行结束投递，默认为最后一行
-b	批量阻塞任务，等待任务池中所有任务运行完成之后再进行投递，而不是一旦有一个任务运行完成立即投递下一个任务

3.3.2 使用场景

(1) 对于大量的 shell 命令任务，可将所有需要运行的命令写在一个 shell 文件中，通过 runsge 运行 shell 文件，会分批次投递任务，同时投递和运行的任务总数由 -n 参数给定，一旦有一个任务运行完成，立即投递下一个任务，保证任务池中的个数为 -n 指定的任务数，还可以指定任务阻塞参数 -b，表示当任务池中所有任务完成之后再开始投递下一批任务。

(2) 可以在 shell 文件中写入一行 wait，此处会用于 runsge 阻塞信号，即等待任务池中所有任务完成之后再进行下一批次任务的投递。

(3) 任务投递过程中，已经运行成功的任务，再次投递会自动跳过，不会运行，因此如果第一次 runsge 运行有任务失败，下次运行还是按第一次运行的命令即可，会自动跳过已经成功的任务。

(4) runsge 投递的任务，同样支持 runstate 查看任务的状态，使用方式为 runstate + logdir，其中 logdir 即为 -o 参数的文件夹。

4. 说明

- a) runjob 采用 python 编写，源码已放在 <https://github.com/yodeng/runjob>，安装包也上传到 pypi 仓库源里，感兴趣可以自行修改其中的功能，欢迎大家共同开发，进行 git 交流。
- b) 欢迎使用和建议，也欢迎提供新的功能的需求，任何有必要的需求都可以考虑添加到 runjob 软件中，使其更加完善。
- c) 任何疑问或问题，请随时与我联系。