

自动任务监控程序 PyMonitor

Li Miao

泛亚太医学研究部

- 不常驻内存，不占用登陆节点资源。
- 直接使用DRMAA API与SGE底层通信，不需要调用qsub, qstat等命令。
- 所有任务以项目为单位统一管理，可对项目进行添加、删除、暂停、继续、设置优先级。每日自动发送邮件报告各项目进展，并支持图形化展示。
- 自动异常处理（对于Eqw, dr, dT, cpu不动, 节点死机等情况自动重投，对于超内存任务自动用更大内存重投、硬盘空间不足时自动挂起任务、脚本未正常退出时自动重投）。
- 独有DirectRun功能，不用编写脚本即可使程序运行在计算节点上，且外在表现与本地运行相同（支持管道、返回值判断）。
- 兼容task_monitor和qsub-sge的输入格式，旧有项目可直接导入本程序管理，自动识别之前的进度。
- 增量运行模式。脚本修改（如数据加测）后自动重新运行所有受影响的脚本，未受影响的保持不变。

- 深圳集群：
 - /ifs5/PC_HUMAN_AP/USER/limiao/bin/monitor
- 香港集群：
 - /ifshk5/PC_HUMAN_AP/USER/limiao/bin/monitor
- 程序运行形式为：
 - monitor <command> [arguments]
 - 根据command决定要执行的动作，根据arguments决定具体参数。

Command列表如下

taskmonitor	导入task_monitor.py格式的任务列表
qsubsg	导入qsub-sge.pl格式的任务列表
directrun	直接在计算节点上运行程序
stat	获取项目进度统计
setdefault	设置默认参数
pauseproject	暂停指定项目的任务
resumeproject	恢复指定项目的任务执行
removeproject	删除指定项目的任务
updateproject	更新指定项目的状态
setmaxjobs	设置项目的最大运行任务数量
logdump	显示项目日志
help	显示帮助信息

以下为各个命令的使用方法



taskmonitor

taskmonitor命令用于导入task_monitor.py格式的任务列表。接受的参数如下：

-i		要导入的文件（通常名为config.txt）
-p		项目名称（后续操作均使用此名称）
-P		投任务所需的-P参数
-q		投任务所需的-q参数
-n		项目的最大运行任务数量
-f		任务的正常结束标志。取值如下：
	0	若.e文件末尾出现特定字符串（由-s参数指定）则认为任务成功完成
	1	若生成了.sign文件则认为任务成功完成
	2	不进行判断，默认任务成功完成
-s		-f 参数所需的字符串（默认Still_water_runs_deep）
-m		任务添加模式。取值如下：
	0	默认模式。自动识别列表中未完成任务并运行。
	1	自动识别列表中未完成或完成后又经过修改的任务并运行。
	2	重新运行列表中的所有任务。

格式说明

- config.txt是李祥春开发的monitor.py监控程序输入文件，DNA流程生成此格式的文件。其格式说明如下：

/.../a.sh:2G	/.../b.sh:3G
/.../a.sh:2G	/.../c.sh:2G
/.../b.sh:3G	/.../d.sh:2G
/.../c.sh:2G	/.../e.sh:1G
/.../d.sh:2G	/.../e.sh:1G

- 其中第一列为第二列运行的前提条件，冒号之后为申请内存大小。
- 例如以上示例表示：先投a.sh，待其运行完毕后投b.sh和c.sh，待b.sh运行完毕后投d.sh，待c.sh和d.sh都运行完毕后投e.sh

- 情景1：默认情况下新增项目
 - `monitor taskmonitor -i config.txt -p HUMxucX`
 - （此处的-P -q 等参数使用了默认值，见后文setdefault命令）
- 情景2：流程已经运行完成，但其中部分脚本运行出错，希望重新运行这些脚本
 - `monitor taskmonitor -i config.txt -p HUMxucX`
- 情景3：流程已经运行完成，但其中一个样品的脚本进行了修改，希望更新这个样品相关的数据
 - `monitor taskmonitor -i config.txt -p HUMxucX -m 1`
- 情景4：流程已经运行完成，但希望全部重新运行。
 - `monitor taskmonitor -i config.txt -p HUMxucX -m 2`

qsubsgsge命令导入qsub-sgsge.pl格式的任务列表。接受的参数如下：

-i		要导入的文件（通常名为work.sh）
-p		项目名称（后续操作均使用此名称）
-P		投任务所需的-P参数
-q		投任务所需的-q参数
-n		项目的最大运行任务数量
-l		投任务所需的-l参数
-L		每个子任务分割行数
-f		任务的正常结束标志。取值如下：
	0	若.e文件末尾出现特定字符串（由-s参数指定）则认为任务成功完成
	1	若生成了.sign文件则认为任务成功完成
	2	不进行判断，默认任务成功完成
-s		-f 参数所需的字符串（默认Still_water_runs_deep）
-m		任务添加模式。取值如下：
	0	重投未完成任务
	1	重新产生脚本并重投所有任务
	2	添加任务，与该项目已有任务并行运行
	3	添加任务，待该项目已有任务全部运行完毕后再运行

格式说明

- work.sh是樊伟开发的qsub-sge.pl监控程序输入文件，RNA流程生成此格式的文件。其格式说明如下：

a.sh

b.sh

c.sh

- 由-L参数控制分割行数，每L行作为一个子单元，各子单元并行投任务。
- 例如以上示例表示：a.sh b.sh c.sh并行投任务

- 情景1：默认情况下新增项目，子任务按照每1行分割。
 - `monitor qsubsgc -i jobs.txt -p HUMxucX -l vf=4g`
 - （此处的-P -q 等参数使用了默认值，见后文setdefault命令）
- 情景2：新的任务列表依赖于旧有任务
 - `monitor qsubsgc -i jobs.txt -p HUMxucX -m 3`
 - （此处使用了默认的-l参数（vf=4g））

stat

stat命令显示当前项目的运行情况。接受的参数如下：

-p		项目名称（若不提供项目名称则显示所有项目）
-m		显示模式。取值如下：
	0	显示项目中各脚本执行情况汇总
	1	生成任务关系网络图
	2	显示项目名称和项目路径
	3	详细显示每个脚本的状态
-e		e-mail地址。若指定此参数，相应信息将直接发至邮箱（只适用于-m 0或-m 1模式）



汇总表格介绍

汇总表格示例:

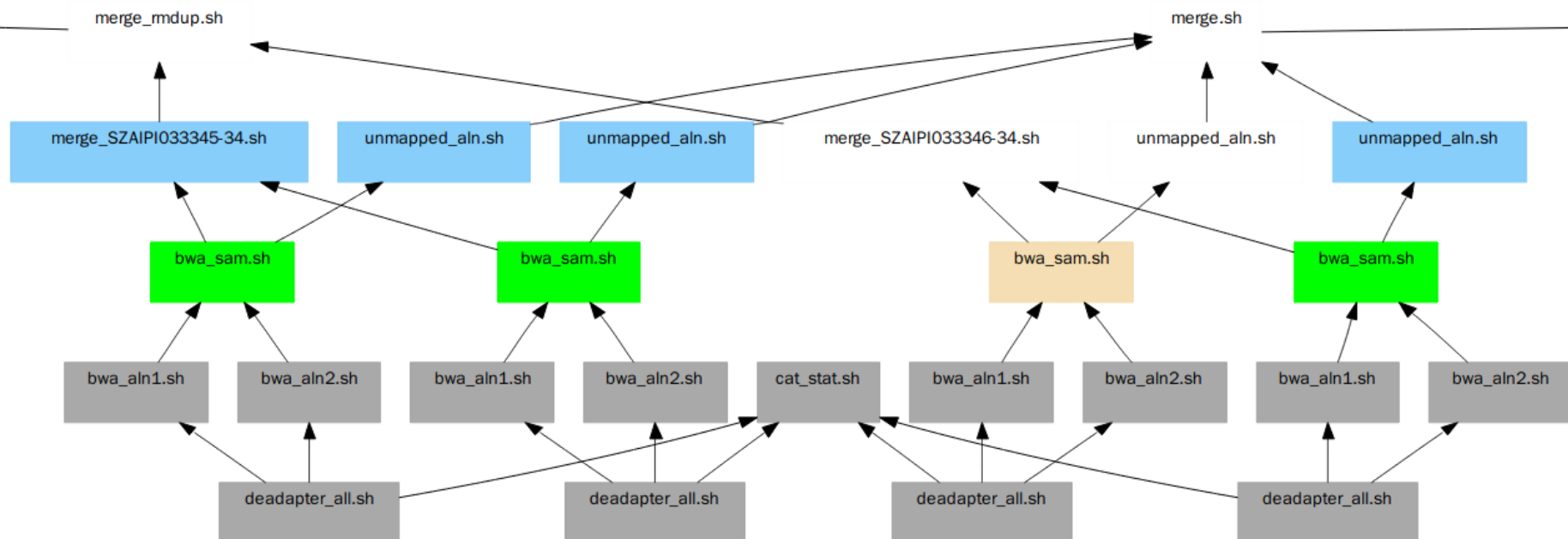
Name	Pending	Queued	Hold	Running	Done	Error	Total	Max
test	0	0	0	0	3	0	3	10
HUMqrfR	1918	3	66	45	156	0	2188	200

汇总表格说明:

Name	项目名称
Pending	尚未运行的任务数量
Queued	正在排队的任务数量
Hold	暂停状态的任务数量
Running	正在运行的任务数量
Done	已完成的任务数量
Error	出现错误的任务数量
Total	项目中的任务总数
Max	项目可同时运行的最大任务数

网络图介绍

- 网络图通过前述stat命令生成。图形为svg.gz格式，解压后可通过IE10/Firefox/Chrome/Safari/Opera等浏览器直接打开
- 鼠标指向各节点可显示相应脚本的完整路径。



	任务未运行
	任务已提交
	任务已进入队列，等待前置任务完成(hqw)
	任务已进入队列，被用户暂停(hqw)
	任务已进入队列，等待运行(qw)
	任务正在运行(r)
	任务已开始运行，被用户暂停(s)
	任务已开始运行，节点异常(T)
	任务完成
	任务失败

- 情景1：显示所有项目的进度。
 - `monitor stat`
- 情景2：绘制HUMxucX项目的执行情况图并使用邮件发送
 - `monitor stat -p HUMxucX -m 1 -e xxx@genomics.cn`
- 情景3：列出HUMxucX项目中所有运行失败的脚本
 - `monitor stat -p HUMxucX -m 3 | grep failed`

setdefault

setdefault命令设置程序的默认参数，这样不必每次投任务时指定。接受的参数如下：

-P		投任务所需的-P参数
-q		投任务所需的-q参数
-m		每日邮件发送模式。取值如下：
	0	以表格形式发送项目状态。
	1	以图形形式发送项目状态。
	2	不发送邮件
-e		e-mail地址。
-n		计划任务的时间间隔（分钟）。默认为20，表示每20分钟运行计划任务一次。此数值需为60的约数。



- 情景1：设定默认值。
 - `monitor setdefault -P tumortest -q bc.q`
- 情景2：设置每日以表格汇总形式发送邮件。
 - `monitor setdefault -m 0 -e xxx@genomics.cn`



暂停项目执行。接受的参数如下：

-p		需暂停的项目名称
-m		操作模式。取值如下：
	0	等待当前已在运行的任务完成，但不再投新任务。
	1	立即挂起所有正在运行的任务。



恢复暂停的项目。接受的参数如下：

-p	需恢复的项目名称
----	----------



removeproject

删除项目。此命令只删除监控数据库，项目数据不受影响。项目完成后默认保留一周，一周后若未更改则自动删除监控数据。接受的参数如下：

-p		需删除的项目名称
-m		操作模式。取值如下：
	0	等待当前已在运行的任务完成，之后删除项目。
	1	立即删除所有属于该项目的任务。
-d		删除所有已完成项目。（不需要指定-p参数）
-b		强制删除项目。（用于删除出错的数据库，需要指定-p参数）



updateproject

更新项目状态。通常情况下项目状态会自动更新，若出现特殊情况无法更新，或项目紧急需要立即更新，可使用此命令手动更新。接受的参数如下：

-p	需恢复的项目名称
----	----------



- 情景1：立即暂停HUMxucX项目的所有任务。
 - `monitor pauseproject -p HUMxucX -m 1`
- 情景2：删除正在运行的项目，保留已投上的任务。
 - `monitor removeproject -p HUMxucX -m 0`
- 情景3：删除所有已完成项目的监控数据（已完成项目默认保留1天，若无变化第二天自动删除）
 - `monitor removeproject -d`

setmaxjobs

设置项目同时运行的最大任务数。可通过此属性设定项目的优先级。接受的参数如下：

-p	项目名称
-n	同时运行的最大任务数。若不设置则自动为平均任务数。



logdump

显示项目日志。此命令只显示当天的项目日志，之前的日志会自动归档到项目所在目录的.log文件中。接受的参数如下：

-p	项目名称
-d	删除已有的项目日志



directrun

directrun命令可直接在计算节点上运行程序，表现形式与本地运行一致。接受的参数如下：

-i	需运行的程序名称。若带参数必须用单引号引用。
-b	使用此参数表示运行的是一个二进制程序。不使用此参数表示运行的是一个shell脚本
-d	若程序需要从管道中读取输入则加此参数
-P	投任务所需的-P参数
-q	投任务所需的-q参数
-l	投任务所需的-l参数



- 情景1：运行某R脚本分析程序。
 - `monitor directrun -i 'Rscript dnacopy.R' -b -l vf=3g`
 - （此处使用了默认的-P -q参数）
- 情景2：通过管道处理数据。
 - `samtools mpileup -r chr10:1-1000000 a.bam | monitor directrun -i 'gzip -9' -b -d -l vf=4g > chr10.pileup.gz`

cron

cron是内部命令，用于管理计划任务。通常不需要手工操作，只在特殊情况下使用。接受的参数如下：

-m		操作模式。取值如下：
		...
	5	更改计划任务运行的节点。计划任务只会在一个登陆节点运行，通常是首次使用所在的节点。如果该节点死机，一种做法是等待节点重启后自动恢复，另一种做法是在其他节点上使用此命令，将计划任务切换至当前节点。



- 日志文件

- 程序会在home目录生成一个名为 .pymonitor.log 的日志文件。其中记录了程序的错误信息。正常情况下该文件为空白，表示没有错误。

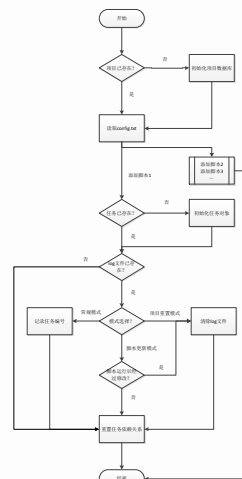
- 配置文件

- 程序会在home目录生成一个名为 .pymonitor.conf 的配置文件，其中包含各种高级参数。通常情况下默认参数即可满足需求，如有特殊需要可修改。
- 常用参数修改可通过setdefault命令实现，非常用参数可直接修改配置文件。
- 随着程序的不断升级，默认参数也可能更改。以下是当前的默认参数及其含义。

配置文件格式

[project]	项目参数
ProjectName = ProjectLocation.db	每行表示一个项目的名称及位置
[base]	全局参数
suspendinterval = 1	节点死机判定周期。默认2次检查节点均挂起则判定为节点死机
memoryexceedlimit = 1.5	内存限制。默认为实际内存用量超过设定的1.5倍时自动重投
rescheckinterval = 2	节点状态检查周期。默认每3次计划任务中有一次检查节点状态
defaultmemg = 4	投任务时若未设置内存用量，默认vf=4g
jobmaxretries = 10	任务重试次数，默认超过10次仍失败则不再重投
defaultfinishmark = Still_waters_run_deep	投任务时若未指定-s参数，则使用此字符串作为正常结束标志
diskminspaceg = 500	磁盘空间限制。默认空间小于500G时自动暂停所有任务，待空间足够再恢复
cputimeratio = 0.01	Cpu时间限制，默认为1%，即真实时间经过100分钟后cpu时间应大于1分钟，小于此数值则认为节点已死机
globalmaxjobs = 600	项目默认最大任务数。投任务时若未指定-n参数，则使用此数值

项目导入流程图



- 这个程序...怎么用？
 - 以DNA流程CSAP为例，生成脚本和config.txt文件后，执行monitor taskmonitor -i config.txt -p projectname -P tumortest -q bc.q 即可。之后一切由监控程序接手，不需要人工干预。
- 如何查看项目的执行进度？
 - 通过stat功能完成。执行monitor stat默认可列出所有项目的进度统计。更多选项见前文解释。
- 为何新投的任务为queued状态的很少，更多是hold的状态？
 - 这是正常现象。详细解释见后文。
- 为何有些样品已经跑到了somatic阶段，有些样品还没完成比对？
 - 程序是根据config.txt文件中的顺序确定任务优先级。
- error状态的任务应如何处理？
 - 可能是脚本有问题，建议检查脚本及log文件，修正后使用taskmonitor功能重新导入任务列表，重置失败任务。

- 程序运行后自动退出，用top命令也看不到？
 - 这是正常现象。本监控程序采用定时运行的模式，不常驻内存。
- 投任务后用qstat看不到任务？
 - 这是正常现象。所有项目的任务会统筹规划，定时投入队列。
 - 使用监控程序后，所有任务均由监控程序管理，可通过stat功能查看进度，不建议再使用qstat命令。
- 使用stat出现一段错误信息？
 - 这表示监控程序正在投任务，请稍后再试。
- 使用directrun时无法用Ctrl-C终止程序？
 - 由于程序已投到了队列上，无法用常规方法停止。可用qdel删除相应的编号的任务，程序会自然终止。

- 程序是否会消耗系统资源？
 - 任何监控程序都会消耗系统资源。本程序采用了一系列方法降低资源消耗：
 1. 不常驻内存，也不需要频繁调用。
 2. 无论管理多少项目，每个用户只需运行一个实例。
 3. 采用DRMAA API与SGE通信，开销远低于qsub, qstat等。
 4. 对于必须调用qstat -j获取信息的情况，尽量降低其使用频率（默认每小时不超过1次）
 5. 用户查询任务状态时不必与服务器通信。
 - 可以认为，本程序的资源消耗低于现有的各种监控程序，甚至低于人工监控的开销。
- 既然程序不需频繁调用，如何能保证任务及时运行？
 - 程序使用了SGE的hold功能，对于依赖关系未满足的任务以hold的形式投上队列，这样当前置任务完成后后续任务会自动运行，不需要监控程序的干预。
- 我想研究程序的源代码...
 - 本程序完全开源，任何人均可直接查看源代码并进行修改。唯一的要求是修改后的版本也必须公开，让其他人也可以借鉴。

Special thanks to:

- Fan Wei
- Li Xiangchun
- He Jun
- Li Lin
- ...

Cancer Group

thanks

