

## به نام خدا

### راهنمای ارسال جاب برای نرم افزار Matlab با استفاده از دستور `submit`

قبل از شروع توضیحات به این نکات توجه فرمایید:

- ❖ سیستم عامل لینوکس به حروف بزرگ و کوچک حساس می باشد.
- ❖ در نام گذاری فایل ها و پوشه ها از کاراکترهای خاص مانند `{ } & # $ @` و فاصله استفاده نکنید.
- ❖ فایل های ورودی را در پوشه `wrkdir` قرار داده و دستور فوق را از جایی که فایل ورودی قرار دارد، اجرا کنید.

در صورتیکه این دستور را بدون هیچ گزینه ای اجرا کنید، توضیحات کلی را مشاهده خواهید کرد:

```
u111112@login2:~/Matlab> submit
Create and submit job for Matlab

Usage: submit <INPUT> [OPTION]

-n <n1>          Number of processors.
-m <mem>        Memory required for job (GB).
-p <part>       Partition name to submit the job. (use 'sinfo')
-v <ver>        Matlab Version. Available: 2019b, 2021b, 2022b, 2023a Default: 2019b
-t <time>       run time of the job. Valid format: M, H:M:S, D-H, D-H:M
-so <sopt>      Additional slurm options if needed.
-no             Only write job file.
-h | --help    Print this message and exit.

Example: submit main.m -n 8 -m 10 -v 2022b -t 2-0 -p amd128

u111112@login2:~/Matlab> █
```

توضیحات کامل گزینه های این دستور در جدول زیر آمده است:

-n	تعداد هسته های مورد نیاز برای اجرای برنامه یا تعداد workerها*
-m	مقدار حافظه (RAM) مورد نیاز بر حسب گیگا بایت.
-v	نسخه مورد نظر. همانطور که مشاهده می شود در حال حاضر چهار نسخه موجود بوده و در حالت پیش فرض از نسخه 2019b استفاده می شود.
-p	پارتیشن مورد استفاده. لیست پارتیشن ها را با دستور <code>sinfo</code> می توان مشاهده کرد. در این مورد با ادمین مشورت کنید.
-t	زمان تقریبی اجرای برنامه. در صورت نیاز می توان آنرا تغییر داد. در فرمت های مورد استفاده حروف D، H، M و S به ترتیب بیانگر روز، ساعت، دقیقه و ثانیه می باشند.
-so	گزینه های اضافی مورد نیاز برای دستور <code>sbatch</code> ، به عنوان مثال اگر بخواهید جابی را روی یک نود خاص بفرستید و یا اینکه جاب بعد از اتمام یک جاب دیگر شروع شود، این گزینه استفاده می شود.
-no	در صورت استفاده از این گزینه، تنها فایل جاب ایجاد می شود. که می توانید در صورت نیاز آنرا تغییر داده و سپس با استفاده از دستور <code>sbatch</code> ارسال نمایید.

\* تنها در صورتی که برنامه شما به صورت موازی نوشته شده باشد (به کمک دستوراتی مانند parfor) انتخاب تعداد هسته‌های بیشتر تا حدودی باعث افزایش سرعت اجرای برنامه می‌شود.

در صورت عدم استفاده از گزینه -no، با اجرای صحیح دستور فوق، پیامی مشابه این پیام مشاهده خواهید کرد:

```
u111112@login2:~/Matlab> submit main.m -n 2 -m 4 -v 2022b -t 60 -p amd128
Submitted batch job 28627
u111112@login2:~/Matlab> sq
```

این پیام به این معنی است که جاب شما توسط سیستم پذیرفته شده است و به محض وجود منابع درخواستی، اجرا خواهد شد. عدد نمایش داده شده، شماره جاب یا همان JobID می‌باشد. با اجرای برنامه پیام‌های خروجی برنامه و همچنین پیام‌های slurm در فایل `<JobID>_<jobname>.out` ذخیره می‌شوند. Jobname نام جاب است که در حالت پیش فرض نام فایل ورودی می‌باشد.

با استفاده از دستور `sq` یا `squeue` می‌توانید از وضعیت جاب خود مطلع شوید:

```
u111112@login2:~/Matlab> sq
JOBID PARTITION NAME ST NODE CPUS MEMORY (SUBMIT_)TIME TIME_LEFT NODELIST(REASON)
28627 amd128 main R 1 2 4G 0:09 59:51 en-7-8
u111112@login2:~/Matlab>
```

عبارت نمایش داده شده در ستون چهارم از سمت چپ، بیانگر وضعیت جاب می‌باشد. حرف R به معنی در حال اجرا و عبارت PD به معنی انتظار می‌باشد. در صورتی که جاب در وضعیت انتظار باشد، ستون آخر علت آنرا نشان می‌دهد، عبارت Resource در ستون آخر به معنی فراهم نبودن منابع درخواستی و عبارت AssocGrpBilling به معنی کم بودن مبلغ شارژ برای اتمام جاب با منابع درخواستی می‌باشد. برای جاب‌هایی که در حالت اجرا هستند، ستون هشتم زمانی که جاب در حال اجرا بوده و ستون نهم بیانگر زمان باقیمانده تا اتمام اجرا می‌باشد. نمایش عبارت Invalid در این ستون به این معنی است که زمان اجرای برنامه از زمان پیش‌بینی شده بیشتر شده، و بهتر است قبل از قطع شدن اجرا، با استفاده از دستور `update-job-time` زمان اجرای برنامه را اصلاح کنید.

برای قطع کردن یک یا چند اجرا، از دستور `scancel <JobID>` استفاده می‌شود، به عنوان مثال:

```
scancel 5894,5983
```

اگر بخواهید تمام جاب‌های خود را کنسل کنید از دستور `scancel -u $USER` استفاده کنید.

با توجه به اینکه در این حالت برنامه مطلب بدون محیط گرافیکی اجرا می‌شود، در صورت نیاز به رسم نمودار باید آنرا در یک فایل تصویر ذخیره کنید. به مثال زیر توجه کنید:

```
a = [0:0.5:5];
b = 2*a.^2 + 3*a -5;
h = figure(1);
plot(a,b);
hgexport(h, 'out.jpg', hgexport('factorystyle'), 'format', 'jpeg')
%=====
```

همانطور که گفته شد به کمک دستور sinfo می‌توانید لیست پارتیشن‌ها را مشاهده کنید.

```
u111112@login2:~/wrkdir> sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
short      up          30:00     1     idle en-7-5
gpu        up          10-00:00:0 1     mix  en-7-1
gpu        up          10-00:00:0 1     idle en-7-2
amd128     up          30-00:00:0 1     down* en-7-9
amd128     up          30-00:00:0 2     drain en-7-[6-7]
amd128     up          30-00:00:0 1     mix  en-7-8
amd48*     up          60-00:00:0 9     down* cn-13-[1-9]
amd48*     up          60-00:00:0 10    mix  cn-11-[2-4,6-7],cn-12-[3-4,6-8]
amd48*     up          60-00:00:0 3     alloc cn-11-[1,5],cn-12-5
amd48*     up          60-00:00:0 3     idle  cn-11-8,cn-12-[1-2]
u111112@login2:~/wrkdir> sinfo -s
PARTITION AVAIL  TIMELIMIT  NODES(A/I/O/T) NODELIST
short      up          30:00     0/1/0/1 en-7-5
gpu        up          10-00:00:0 1/1/0/2 en-7-[1-2]
amd128     up          30-00:00:0 1/0/3/4 en-7-[6-9]
amd48*     up          60-00:00:0 13/3/9/25 cn-11-[1-8],cn-12-[1-8],cn-13-[1-9]
u111112@login2:~/wrkdir>
```

با توجه به خروجی فوق، چهار پارتیشن short, gpu, amd128, amd48 در حال حاضر وجود دارند. پارتیشنی که با علامت \* مشخص شده (amd48) پارتیشن پیشفرض می‌باشد. در ستون STATE وضعیت هر پارتیشن مشخص شده است. در این ستون idle به معنی موجود و آماده به کار، عبارت mix به معنی آزاد بودن قسمتی از منابع و عبارت alloc به معنی رزرو شده می‌باشند. همانطور که در شکل فوق مشاهده می‌شود، دستور sinfo -s لیست پارتیشن‌ها را به صورت مختصر نشان می‌دهد.

جزئیات بیشتر هر پارتیشن را می‌توان با دستور زیر مشاهده کرد:

scontrol show partition <partition\_name>

به عنوان مثال:

```
u111112@login2: → scontrol show partition amd48
PartitionName=amd48
  AllowGroups=ALL AllowAccounts=ALL AllowQos=ALL
  AllocNodes=ALL Default=YES QoS=N/A
  DefaultTime=NONE DisableRootJobs=NO ExclusiveUser=NO GraceTime=0 Hidden=NO
  MaxNodes=UNLIMITED MaxTime=90-00:00:00 MinNodes=0 LLN=NO MaxCPUsPerNode=UNLIMITED
  Nodes=cn-12-[1-8],cn-13-[1-9]
  PriorityJobFactor=1 PriorityTier=1 RootOnly=NO ReqResv=NO OverSubscribe=NO
  OverTimeLimit=NONE PreemptMode=OFF
  State=UP TotalCPUs=816 TotalNodes=17 SelectTypeParameters=NONE
  JobDefaults=(null)
  DefMemPerNode=UNLIMITED MaxMemPerNode=90500
  TRES=cpu=816,mem=1643050M,node=17,billing=85782
  TRESBillingWeights=CPU=54,Mem=26G,GRES/gpu=3900
u111112@login2: →
```

عبارت TRESBillingWeights که در سطر آخر نمایش داده شده است، برای محاسبه هزینه اجرا استفاده می‌شوند به عبارتی در پارتیشن فوق هزینه هر اجرا به صورت زیر محاسبه می‌شود:

$$\{(N_{cpu} \times 54) + (Mem(GB) \times 26) + (GPU \times 3900)\} \times Time(Hours)$$

نکته مهم: منابع و زمان درخواستی در هنگام ثبت درخواست سرویس صرفاً برای برآورد هزینه می‌باشند و در هنگام استفاده از کلاستر به عنوان محدودیت اعمال نمی‌شوند. به عنوان مثال اگر شما درخواست یک سرویس ۳۰ روزه داده باشید و اکانت کلاستر در اختیار شما قرار گیرد، تا هنگامی که کل اعتبار شما مصرف شود می‌توانید از کلاستر استفاده نمایید. با استفاده از دستور `my_usage` می‌توانید از کل اعتبار و میزان مصرف خود آگاه شوید.