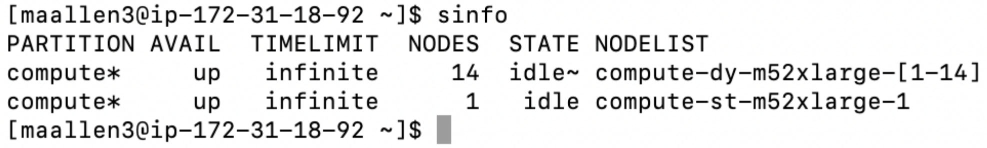
How do I choose my slurm parameters?

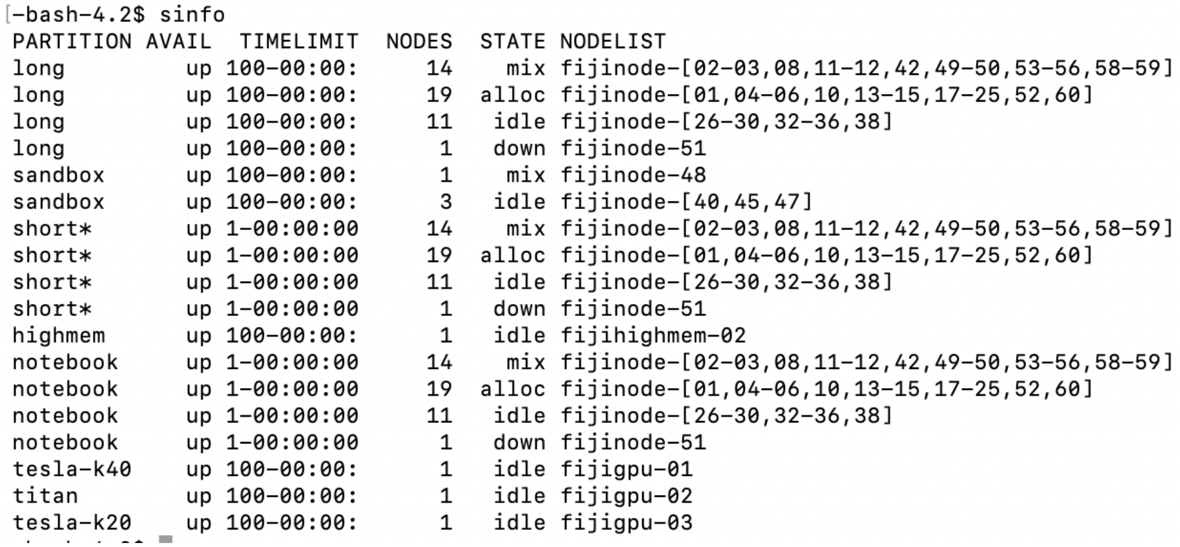
Author: Mary Allen, 2022

1. Talk to your coworkers. Have they run this before. What resources did it take?
2. What if no one has run this before? Guess and Share.
    a. Look at what's on your supercomputer so you can play nice. Never take over the whole super computer. Half of the CPUs and Memory is about as high as you should go!
        i. I want to know what this supercomputer has so I can be nice and share.
            1. First I run sinfo (this tells you the partitions and the node names on the computer)
                a. This computer has a partition named compute and has a node named compute-st-m52xlarge-1.



```
[maallen3@ip-172-31-18-92 ~]$ sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
compute*     up   infinite     14  idle~ compute-dy-m52xlarge-[1-14]
compute*     up   infinite      1   idle compute-st-m52xlarge-1
[maallen3@ip-172-31-18-92 ~]$ 
```

                b. The below super computer has many partitions (long, short, sandbox, highmem). It also says the longest you are allowed to use that partition. (timelime limit)
                    i. timelimit= The stuff before the dash is days, after the dash is time.
                    ii. nodes= The number of nodes on your computer that look alike
                    iii. The default is the short partition. It will run for no more than 1 day.



```
[-bash-4.2$ sinfo
 PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
 long         up 100-00:00:     14    mix fijinode-[02-03,08,11-12,42,49-50,53-56,58-59]
 long         up 100-00:00:     19  alloc fijinode-[01,04-06,10,13-15,17-25,52,60]
 long         up 100-00:00:     11   idle fijinode-[26-30,32-36,38]
 long         up 100-00:00:      1   down fijinode-51
 sandbox      up 100-00:00:      1    mix fijinode-48
 sandbox      up 100-00:00:      3   idle fijinode-[40,45,47]
 short*       up 1-00:00:00     14    mix fijinode-[02-03,08,11-12,42,49-50,53-56,58-59]
 short*       up 1-00:00:00     19  alloc fijinode-[01,04-06,10,13-15,17-25,52,60]
 short*       up 1-00:00:00     11   idle fijinode-[26-30,32-36,38]
 short*       up 1-00:00:00      1   down fijinode-51
 highmem      up 100-00:00:      1   idle fijihighmem-02
 notebook     up 1-00:00:00     14    mix fijinode-[02-03,08,11-12,42,49-50,53-56,58-59]
 notebook     up 1-00:00:00     19  alloc fijinode-[01,04-06,10,13-15,17-25,52,60]
 notebook     up 1-00:00:00     11   idle fijinode-[26-30,32-36,38]
 notebook     up 1-00:00:00      1   down fijinode-51
 tesla-k40    up 100-00:00:      1   idle fijigpu-01
 titan        up 100-00:00:      1   idle fijigpu-02
 tesla-k20    up 100-00:00:      1   idle fijigpu-03
```

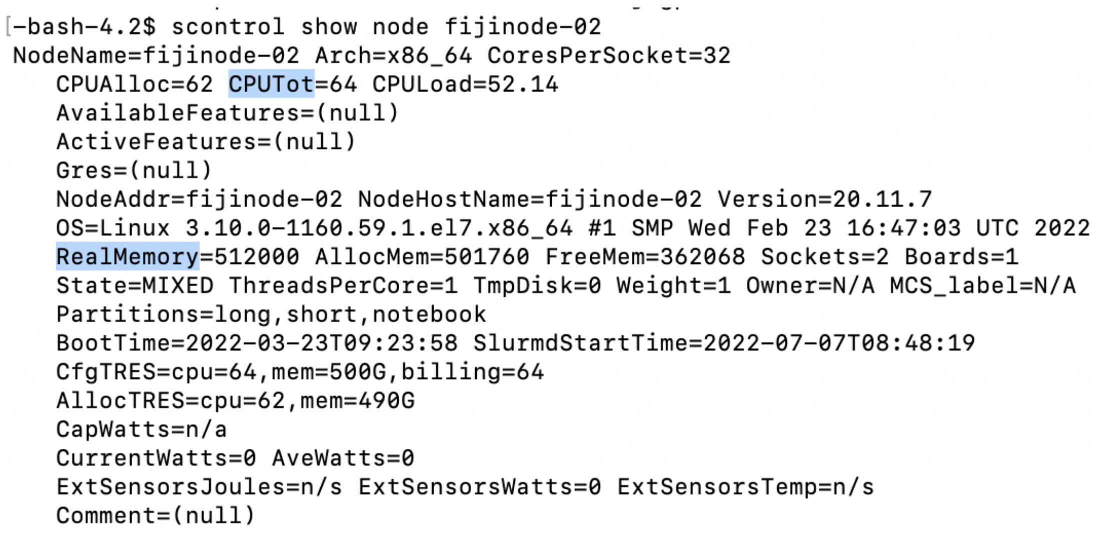
b. Now that I know the partitions and their timelimits— that narrows which partition I might need. I tend to start with the smallest time limit. If the job fails due to lack of time- the err file or the email will say that. Then I move to a node with a longer time limit by changing the partition.

Test it yourself.

What is the timelimit on the compute partition?
How many CPUs and how much memory is on one of the nodes.

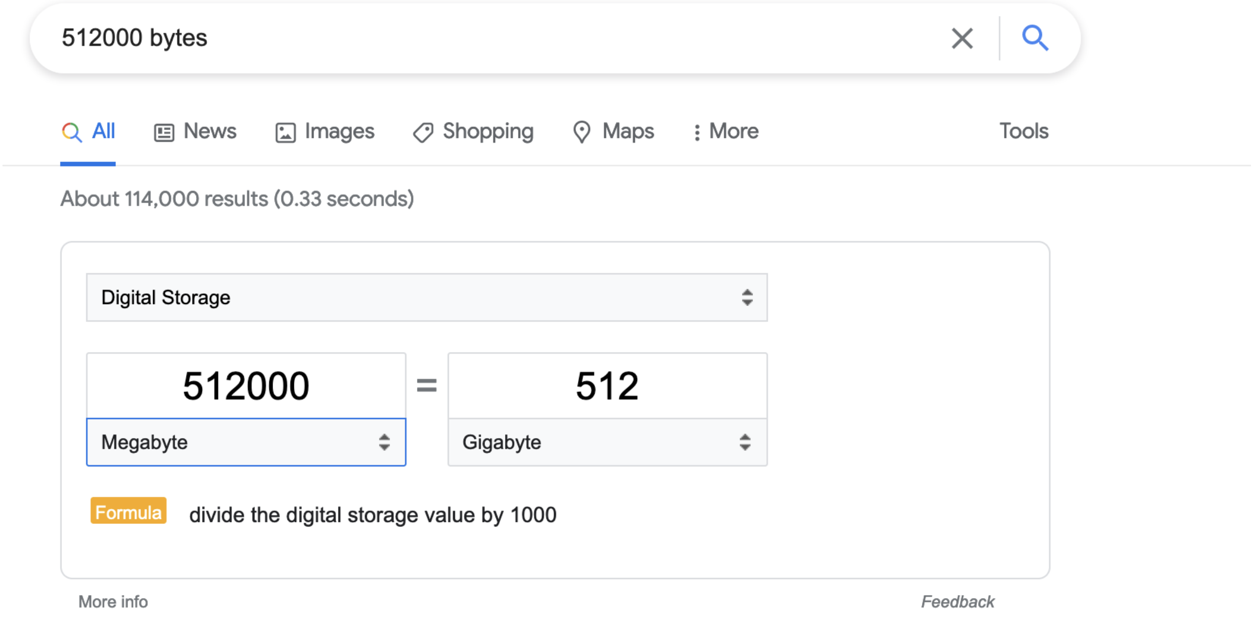
Create a script that runs the command sleep for 100 seconds. Set the slurm time of that script to 2 seconds. What error message do you get?

c. Next— I need to check what the nodes in the partition I'm thinking about have for CPUs and for memory.
   i. I look at some nodes I might want to use.
      1. scontrol show node <nodename>



```
[-bash-4.2$ scontrol show node fijinode-02
 NodeName=fijinode-02 Arch=x86_64 CoresPerSocket=32
    CPUAlloc=62 CPUTot=64 CPULoad=52.14
    AvailableFeatures=(null)
    ActiveFeatures=(null)
    Gres=(null)
    NodeAddr=fijinode-02 NodeHostName=fijinode-02 Version=20.11.7
    OS=Linux 3.10.0-1160.59.1.el7.x86_64 #1 SMP Wed Feb 23 16:47:03 UTC 2022
    RealMemory=512000 AllocMem=501760 FreeMem=362068 Sockets=2 Boards=1
    State=MIXED ThreadsPerCore=1 TmpDisk=0 Weight=1 Owner=N/A MCS_label=N/A
    Partitions=long,short,notebook
    BootTime=2022-03-23T09:23:58 SlurmdStartTime=2022-07-07T08:48:19
    CfgTRES=cpu=64,mem=500G,billing=64
    AllocTRES=cpu=62,mem=490G
    CapWatts=n/a
    CurrentWatts=0 AveWatts=0
    ExtSensorsJoules=n/s ExtSensorsWatts=0 ExtSensorsTemp=n/s
    Comment=(null)
```

This node has 64 CPUs (62 we can use) and 512000 Megabytes of memory.



512000 bytes

About 114,000 results (0.33 seconds)

Digital Storage

512000 Megabyte = 512 Gigabyte

Formula divide the digital storage value by 1000

More info

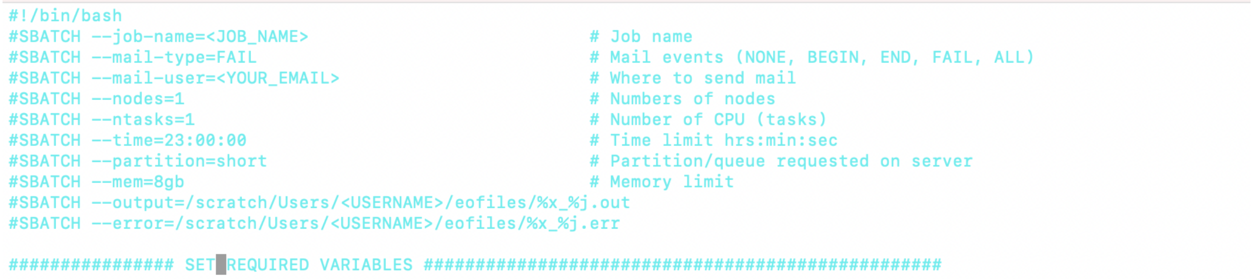
Feedback

512000 megabytes is 512 gigabytes. Because 512/64 is 8, there are 8 Gigabytes of memory per CPU.

3) Design a test job.

Pick your most large input file to do a test run on.

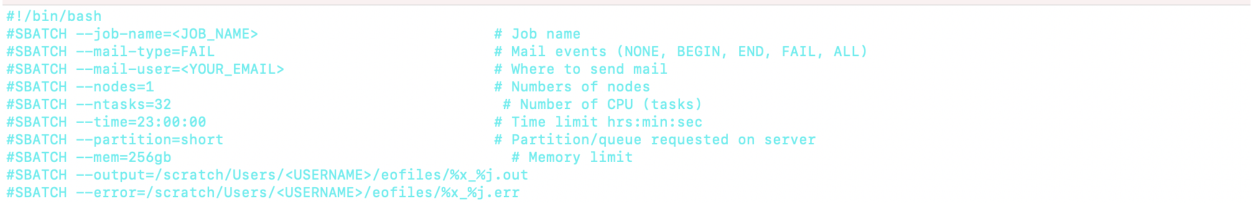
If I have a program without paralyzation, based on what I learned above, I start with 1CPU and the amount of memory that should belong to this CPU. In my case that is 8 because 512/64 is 8.



```
#!/bin/bash
#SBATCH --job-name=<JOB_NAME>                           # Job name
#SBATCH --mail-type=FAIL                                # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=<YOUR_EMAIL>                        # Where to send mail
#SBATCH --nodes=1                                       # Numbers of nodes
#SBATCH --ntasks=1                                      # Number of CPU (tasks)
#SBATCH --time=23:00:00                                 # Time limit hrs:min:sec
#SBATCH --partition=short                               # Partition/queue requested on server
#SBATCH --mem=8gb                                       # Memory limit
#SBATCH --output=/scratch/Users/<USERNAME>/eofiles/%x_%j.out
#SBATCH --error=/scratch/Users/<USERNAME>/eofiles/%x_%j.err

################ SET REQUIRED VARIABLES ##################################################
```

So— if I have a program with paralyzation I generally use half of the CPUs on a node and the number of gigabytes of memory each CPU would get if it was fairly shared. Why half? Then people using 1 CPU can hope on the same node as me. I can set it to all the CPUS, then I get the node to myself-but I often wait in the queue a long time for that.



```
#!/bin/bash
#SBATCH --job-name=<JOB_NAME>                           # Job name
#SBATCH --mail-type=FAIL                                # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=<YOUR_EMAIL>                        # Where to send mail
#SBATCH --nodes=1                                       # Numbers of nodes
#SBATCH --ntasks=32                                      # Number of CPU (tasks)
#SBATCH --time=23:00:00                                 # Time limit hrs:min:sec
#SBATCH --partition=short                               # Partition/queue requested on server
#SBATCH --mem=256gb                                       # Memory limit
#SBATCH --output=/scratch/Users/<USERNAME>/eofiles/%x_%j.out
#SBATCH --error=/scratch/Users/<USERNAME>/eofiles/%x_%j.err
```

Test it yourself.

Look at what partitions, timelimits, and CPUS are on your super computer.

If your script ran a program that was not parallelized what is fair to take? What if your script ran a program that was parallelized? What is fair to take?

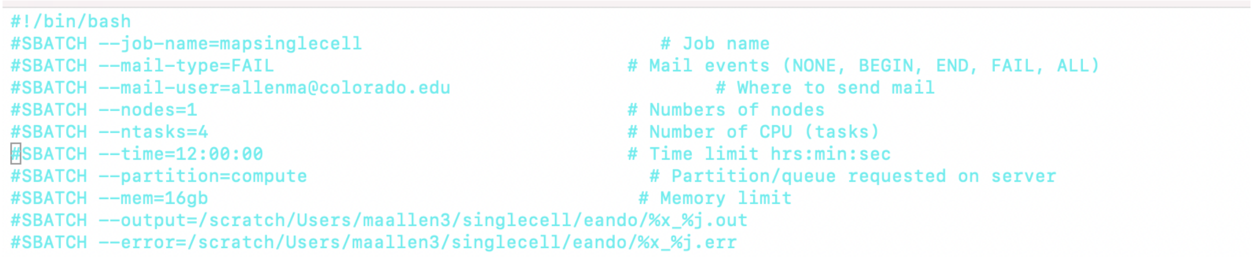
4) Optimize your Guess/a.k.a. “Are you playing nice?”

a. To check how much time/memory/CPU you are really using:
   i. While the job is running:
      1. sstat --format=AveCPU,AvePages,AveRSS,AveVMSize,JobID -j 2050904.batch #where 2050904 is the jobid #**NOTE**: The **sstat**

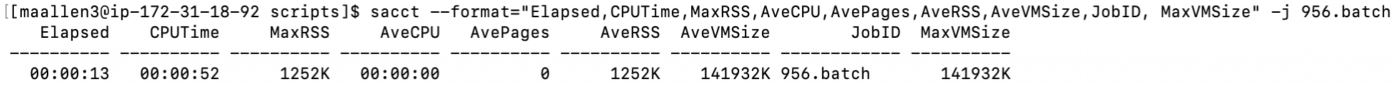
        command requires that the **jobacct_gather** plugin be installed and operational.
    2. scontrol show jobid -dd 2050904 #where 2050904 is the jobid

ii. Or for completed jobs

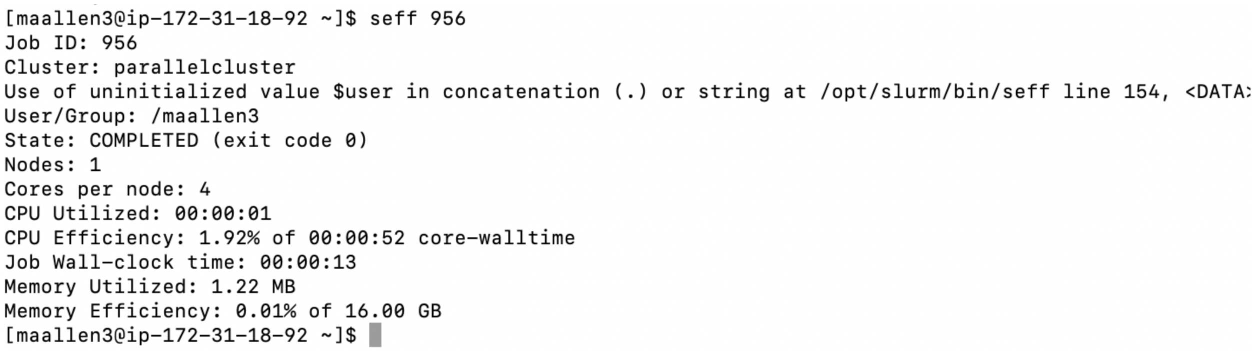
1. sacct --format="Elapsed,CPUTime,MaxRSS,AveCPU,AvePages,AveRSS,AveVMSize,JobID, MaxVMSize" -j 2050904.batch #where 2050904 is the jobid
2. seff jobid



```
#!/bin/bash
#SBATCH --job-name=mapsinglecell                           # Job name
#SBATCH --mail-type=FAIL                                # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=allenma@colorado.edu                        # Where to send mail
#SBATCH --nodes=1                                       # Numbers of nodes
#SBATCH --ntasks=4                                      # Number of CPU (tasks)
#SBATCH --time=12:00:00                                 # Time limit hrs:min:sec
#SBATCH --partition=compute                               # Partition/queue requested on server
#SBATCH --mem=16gb                                       # Memory limit
#SBATCH --output=/scratch/Users/maallen3/singlecell/eando/%x_%j.out
#SBATCH --error=/scratch/Users/maallen3/singlecell/eando/%x_%j.err
```



```
[[maallen3@ip-172-31-18-92 scripts]$ sacct --format="Elapsed,CPUTime,MaxRSS,AveCPU,AvePages,AveRSS,AveVMSize,JobID, MaxVMSize" -j 956.batch
   Elapsed    CPUTime     MaxRSS     AveCPU   AvePages     AveRSS  AveVMSize        JobID  MaxVMSize
---------- ---------- ---------- ---------- ---------- ---------- ---------- ------------ ----------
  00:00:13   00:00:52      1252K   00:00:00          0      1252K    141932K 956.batch       141932K
```

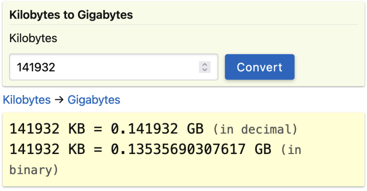


```
[maallen3@ip-172-31-18-92 ~]$ seff 956
Job ID: 956
Cluster: parallelcluster
Use of uninitialized value $user in concatenation (.) or string at /opt/slurm/bin/seff line 154, <DATA>
User/Group: /maallen3
State: COMPLETED (exit code 0)
Nodes: 1
Cores per node: 4
CPU Utilized: 00:00:01
CPU Efficiency: 1.92% of 00:00:52 core-walltime
Job Wall-clock time: 00:00:13
Memory Utilized: 1.22 MB
Memory Efficiency: 0.01% of 16.00 GB
[maallen3@ip-172-31-18-92 ~]$
```

Did using 4 CPUS work well for this script? How do you know?

How much memory did I ask for? How much did I use?

hint:



Kilobytes to Gigabytes

Kilobytes

141932

```
141932 KB = 0.141932 GB (in decimal)
141932 KB = 0.13535690307617 GB (in
binary)
```

5)Run on all your files in individual sbatch scripts. For the slurm command, add 20% more memory than running on the largest file took.