PARAM SANGANAK USAGE POLICY

1. This policy applies exclusively to users of IIT Kanpur. The policy will be periodically reviewed by the "HPC Advisory Committee" based on usage patterns.

2. The computational resources of Param Sanganak will be shared on a fair use basis with non-IIT Kanpur users (maximum of 40% of computing resources) and the rest by IIT Kanpur users.

3. The resource allocations are based on a fair share policy for all the users of IIT Kanpur. The heavier the usage by an individual user, the lower would be the priority of their jobs.

4. A usage-based charging policy was applied to the PARAM Sanganak starting August 1, 2024. Users can now apply for high-priority access in addition to regular access. Charges for high-priority access differ from those for regular access. For information related to charges, please visit the HPC website.

5. The summary of the total compute nodes on Param Sanganak is mentioned in Table-1. For extensive details about the nodes, please refer to the Param Sanganak Manual.

Table 1: Hardware Configuration

	Standard CPU Compute Nodes	High Memory CPU Compute Nodes	GPU Compute Nodes
Number of Nodes	217 (214+3)	78	20
CPU	2* Intel Xeon Platinum 8268	2* Intel Xeon Platinum 8268	2* Intel Xeon G-6248
Cores	48	48	40
Clock	2.9 GHz	2.9 GHz	2.5 GHz
Memory	192 GB, DDR4 2933 MHz	768 GB, DDR4 2933 MHz	192 GB, DDR4 2933 MHz
SSD	480 GB (local scratch) per node	480 GB (local scratch) per node	480 GB (local scratch) per node
Nodes	cn[001-217]	hm[001-078]	gpu[001-020]
			2*NVidia V100 per node
			GPU Cores per node= 2*5120= 10240
			GPU Memory = 16 GB HBM2 per NVidia V100
Note: There are total of 312(214+78+20) nodes in Param Sanganak. 3 Extra Nodes are provided by OEM for hardware replacement.			

6. A user can submit up to 8 jobs in total, but only a maximum of 4 of them can be running at any given time. The system will reject any additional jobs beyond the 8-job limit.

7. Partition Scheme: The current partition scheme is mentioned in Table 2. Users can check the current configuration using the system commands `sinfo` and `scontrol show partition`.

Table 2: Partition Table

Partition Name	Minimum #CPU cores (equivalent nodes)	Maximum #CPU cores (equivalent nodes)	Maximum Time [Days-HH:MM:SS]	Default Time [HH:MM:SS]	Nodes allocated to each partition (some nodes are shared in more than one partition)	Total Nodes
serial	1	24	02-00:00:00	02:00:00	hm[001-002]	2(hm)
small	48 (1 node)	96 (2 nodes)	02-00:00:00	02:00:00	cn[001-050], hm[003-029]	50(cn)+27(hm)
medium	96 (2 nodes)	192 (4 nodes)	02-00:00:00	02:00:00	cn[051-110], hm[030-054]	60(cn)+25(hm)
large	192 (4 nodes)	480 (10 nodes)	03-00:00:00	02:00:00	cn[111-187], hm[055-078]	76(cn)+24(hm)
fat	480 (10 nodes)	____	02-00:00:00	02:00:00	cn[171-217], hm[055-078]	47(cn)+24(hm)
hm	48 (1 node)	288 (6 nodes)	02-00:00:00	02:00:00	hm[003-078]	76(hm)
gpu	cpu=1,gres/gpu=1	cpu=160,gres/gpu=8	02-00:00:00	02:00:00	gpu[001-020]	20(gpu)

* Maximum #CPU Cores has not been set explicitly in the FAT partition, but it can be dictated by the maximum number of nodes in the partition.

8. The default wall time for every partition is 2 hours. The maximum wall time for most partitions is 2 days, except for the large partition, which has a maximum wall time of 3 days.

9. For any queries regarding the policy, please write to <sanganaksupport@iitk.ac.in>. Additionally, you can visit Room No. 212 for related concerns.

10. For any queries regarding the account creation and recharging, please write to <hpcreports@iitk.ac.in>. Additionally, you can visit Room No. 208 for related concerns.

PARAM SANGANAK INITIAL INSTRUCTIONS

1. **Login, password and account usage:**

- How to login: `ssh -p 4422 [username]@paramsanganak.iitk.ac.in`

- Password Change: You can change your password upon login using Linux "`passwd`" command.

- Password Issue: In case you forgot your password you can email us at "sanganaksupport@iitk.ac.in" to reset the password.

- Support: NSM support will be provided by NSM team to Param Sanganak users through the Ticketing tool at https://paramsanganak.iitk.ac.in/support/. Users need to login at this NSM-Portal/Ticketing-Tool using the same <username> and <password> as their Param Sanganak account. Additionally, you can visit Room No. 212 for any concerns or write to <sanganaksupport@iitk.ac.in>.

- Storage Usage: Users can check their /home and /scratch storage usage using "`myquota`" command.

- CPU/GPU Balance: Users can check their CPU Hours and GPU Hours Balance using the command "`IITK_RA_Balance`" and "`IITK_HP_Balance`" respectively. Where command `IITK_HP_Balance` gives usage details about High Priority account and `IITK_RA_Balance` gives usage details about Regular Access account. Users can see details like Total CPU Minutes, Total CPU Hours, Consumed CPU Hours, Balanced CPU Hours, Total GPU Minutes, Total GPU Hours, Consumed GPU Hours, Balanced GPU Hours using the above commands.

2. Sample Code:

```
$cat hello.c                  {sample c code}
#include <stdio.h>
#include "mpi.h"

int main(int argc, char *argv[])
{
        MPI_Init(&argc, &argv);
        int rank , size;
        MPI_Comm_rank(MPI_COMM_WORLD, &rank); //rank of this process
MPI_Comm_size(MPI_COMM_WORLD, &size); //rank of this process
printf("I am %d of %d\n", rank, size);
        MPI_Finalize();
}
```

3. Compilation of Sample Code:

- **May need to load necessary module. To get list of all installed module run command "module available"**
    - o `module load compiler/intel/latest`
        - Above compiler (or compiler module) is given as a test case.

- **Now compile the code using the command below. It will produce "a.out" named file.**
    - o `mpiicc hello.c`

4. Running of Sample Code interactively:

- **For running the executable (a.out in this case) run the below command on login node**
    - o `mpirun -np 4 ./a.out`
    - o User should use login nodes only for testing purposes and file transfers. Submitting a job into a partition as mentioned in Table 2 is recommended for actual use.

- **The above command will produce the below output**
    ```
    I am 0 of 4
    I am 1 of 4
    I am 2 of 4
    I am 3 of 4
    ```

5. Sample Job Script:

- **To run a job on the nodes of any of the partitions mentioned in Table 2.**

```
$ cat myworld.sh
#!/bin/bash
#SBATCH -N 2
#SBATCH --ntasks-per-node=48
#SBATCH --error=err.out
#SBATCH --output=out.out
#SBATCH --time=01:00:00
#SBATCH --partition=small
module load compiler/intel/latest
mpirun -np 96 ./a.out > output_myworld
```

- **sbatch and its option**

  sbatch submits a batch script to Slurm Cluster. The batch script may be given to sbatch through a file name on the command line, or if no file name is specified, sbatch will read in a script from standard input. The batch script may contain options preceded with "#SBATCH" before any executable commands in the script. sbatch will stop processing further #SBATCH directives once the first non-comment non-whitespace line has been reached in the script. For more information click https://slurm.schedmd.com/sbatch.html
  - -N: Request number of nodes be allocated to this job
  - --ntasks-per-node: Number of tasks to be invoked on each node
  - --partition: Request a specific partition for the resource allocation
  - --time: Set a limit on the total run time of the job allocation
  - -A or --account: Charge resources used by a job to specific account; this option is useful when user need to submit in a specific account such high-priority access account
  - --error: error file
  - --output: output file

6. Submit a Sample Job Script:

- **To submit a job-script file use below command:**

  o sbatch myworld.sh

- **Check status of a job:**
  - After submitting a job-script file a user will get a job-id using that the status of the job can be checked later with the help of `squeue` command e.g. "`squeue --job <job-id>`".
    For more information click https://slurm.schedmd.com/squeue.html
  - To check jobs for a user using command "`squeue -u <username>`"

7. **Submit a sample Job in High Priority account:**
  - Users having accounts in both high priority and regular access need to submit jobs with extra options in their submit scripts. For example, for a user with name "bob" the sample scripts for regular and high priority access will be given below. Remember to change bob with your actual username in Param Sanganak cluster.

Regular Access	High Priority Access

```#!/bin/bash```
 ```#SBATCH -N 2```
 ```#SBATCH --ntasks-per-node=48```
 ```#SBATCH --error=err.out```
 ```#SBATCH --output=out.out```
 ```#SBATCH --time=01:00:00```
 ```#SBATCH --partition=small```
 ```#SBATCH --account=paramsanganak```
 ```#SBATCH --qos=bob_ra```
 ```module load compiler/intel/latest```
 ```mpirun -np 96 ./a.out > output_myworld```
	```#!/bin/bash```
	 ```#SBATCH -N 2```
	 ```#SBATCH --ntasks-per-node=48```
	 ```#SBATCH --error=err.out```
	 ```#SBATCH --output=out.out```
	 ```#SBATCH --time=01:00:00```
	 ```#SBATCH --partition=small```
	 ```#SBATCH --account=iitk_hp```
	 ```#SBATCH --qos=bob_hp```
	 ```module load compiler/intel/latest```
	 ```mpirun -np 96 ./a.out > output_myworld```