HPC: PARAM Utkarsh Supercomputing Facility @ C-DAC Bangalore

Enhance your business to next level

Knowledge Park

Electronic City

CPSF,Team
CDAC-Bangalore
C-DAC,Bengaluru

C-DAC PARAM Supercomputing Facility (CPSF) Data Centre

Agenda

- Data Centre
- PARAMUtkarsh Architecture
- Applications
- Scheduler
- Sample Script
- Conclusion

C-DAC PARAM Supercomputing Facility (CPSF) Data Centre

DC Area 1800 SqFt (Total Area 7625 SqFt)

Three High density DLC racks cooled with Adiabatic dry-cooler

Two service node racks

Server Racks and Storage racks

Staging area, User terminal Area and Conference room

BMS – Building Management System

- **Integrated systems in BMS**
- **Generator (1+1 redundant)**
- **UPS (n+1 redundant)**
- **Precision Air Conditioning – PAC (n+1 redundant)**
- **Fire Alarm System - FAS (VESDA and NOVEC)**

  Very Early Smoke Detection Apparatus – VESDA

  NOVEC- Fire suppresser
- **Dry cooler (Adiabatic)**
- **Temperature and humidity sensors**

PARAM Utkarsh Security

Perimeter Firewall (UTM)

Cluster Firewall (HA load balancer and IPS)

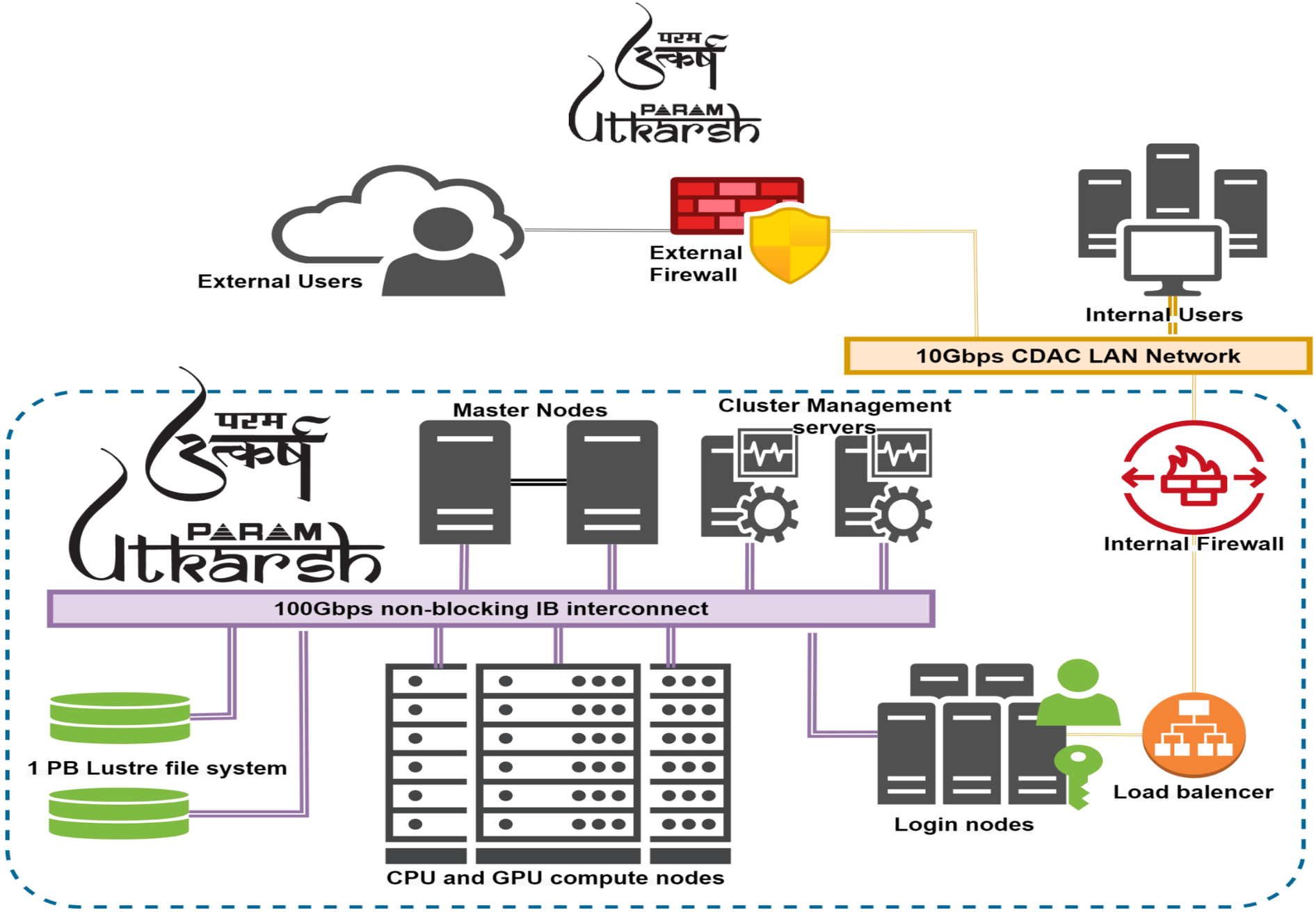
Geographical filtering

Isolated Network

Secured authentication

Physical security with ACS(access control system)

CCTV surveillance(24/7)



System Specifications

Theoretical Peak Floating-point Performance Total (Rpeak)	838 TFLOPS
Base Specifications (Compute Nodes)	2 X Intel Xeon Cascadelake 8268, 24 Cores, 2.9 GHz, Processors per node, 192 GB Memory, 480 GB SSD
Master/Service/Login Nodes	10 nos.
CPU only Compute Nodes (Memory)	107 nos. (192GB)
GPU Compute Nodes (Memory)	10 (192 GB)
High Memory Compute Nodes	39 nos. (768GB)
Total Memory	52.416 TB
Interconnect	Primary: 100Gbps Mellanox Infiniband Interconnect network 100% non blocking, fat tree topology Secondary: 10G/1G Ethernet Network Management network: 1G Ethernet
Storage	1PiB PFS based storage

CPU Only Compute Nodes

- 107 Nodes
- 5136 Cores
- Compute power of Rpeak 476.6 TFLOPS
- Each Node with
  - 2 X Intel Xeon Cascadelake 8268, 24 cores, 2.9 GHz, processors
  - 192 GB memory
  - 480 GB SSD

GPU Compute Nodes

- 10 Nodes
- 400 CPU Cores
- 102400 CUDA Cores
- Rpeak CPU 32 TFLOPS + GPU 156 TF
- Each Node with
  - 2 X Intel Xeon Skylake 6248, 20 cores, 2.5 GHz, processors
  - 192 GB Memory
  - 2 x NVIDIA V100 SXM2 GPU Cards
  - 480 GB SSD

High Memory Compute Nodes

- 39 Nodes
- 1872 Cores
- Compute power of Rpeak 173.7 TFLOPS
- Each Node with
  - 2 X Intel Xeon Cascadelake 8268, 24 cores, 2.9 GHz, processors
  - 768 GB Memory
  - 480 GB SSD

System Details

SN	Server	Number
01	Master Node	02
02	Login Nodes	04
03	Management Nodes	03
04	Firewall	01
05	CPU only nodes	75
06	GPU Nodes	10
07	GPU Ready Nodes	32
08	High Memory Nodes	39
	Total Nodes	166

System Details

Parameter	CPU only(75)	GPU Nodes(10)	GPU Ready(32)	HM Nodes(39)
Processor	2 x Xeon platinum 8268	2 x Xeon G-6248	2 x Xeon platinum 8268	2 x Xeon platinum 8268
Cores	48	40	48	48
Speed	2.9 GHz	2.5 GHz	2.9 GHz	2.9 GHz
Memory	192 GB	192 GB	192 GB	**768 GB**
HDD	480GB SSD	480GB SSD	480GB SSD	480GB SSD
Total cores	**3600**	**400**	**1536**	**1872**
Total Memory	14400 GB	1920 GB	6144 GB	29952 GB
	-	2 x NVIDIA V100	-	-

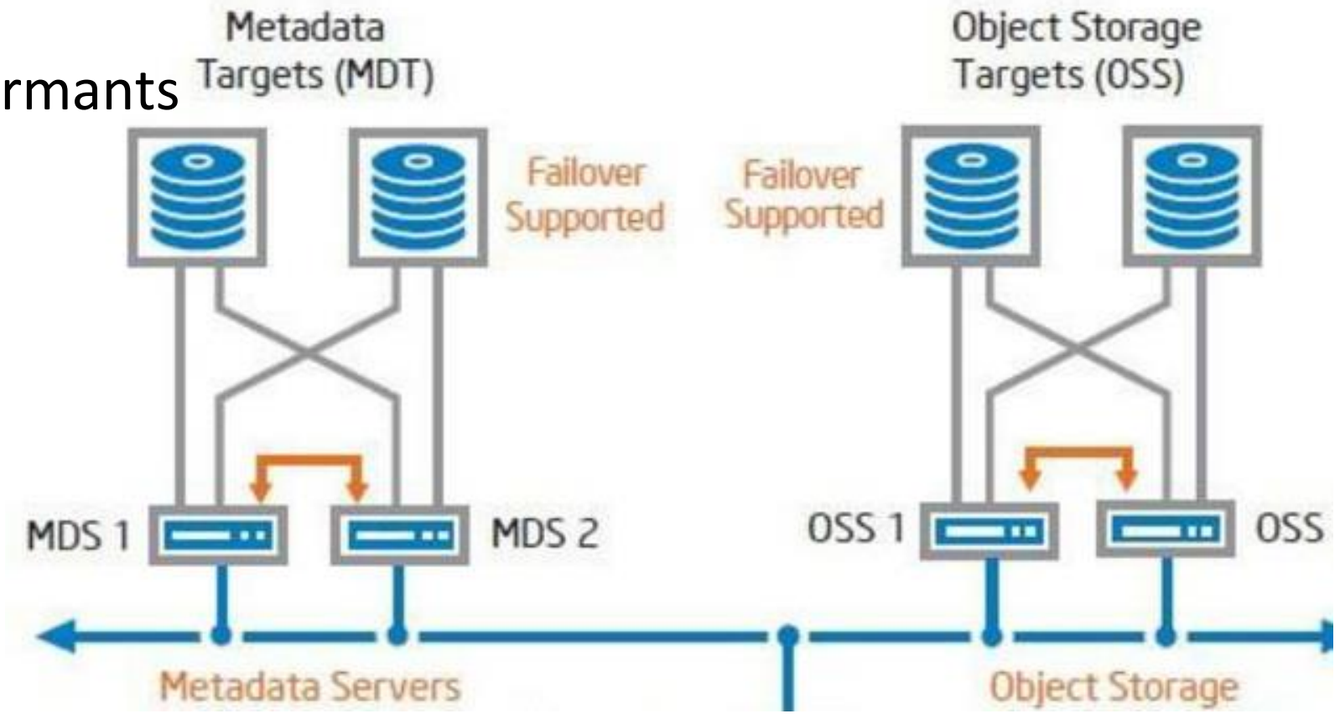
Storage Details

Storage Size: 1 PiB Usable capacity

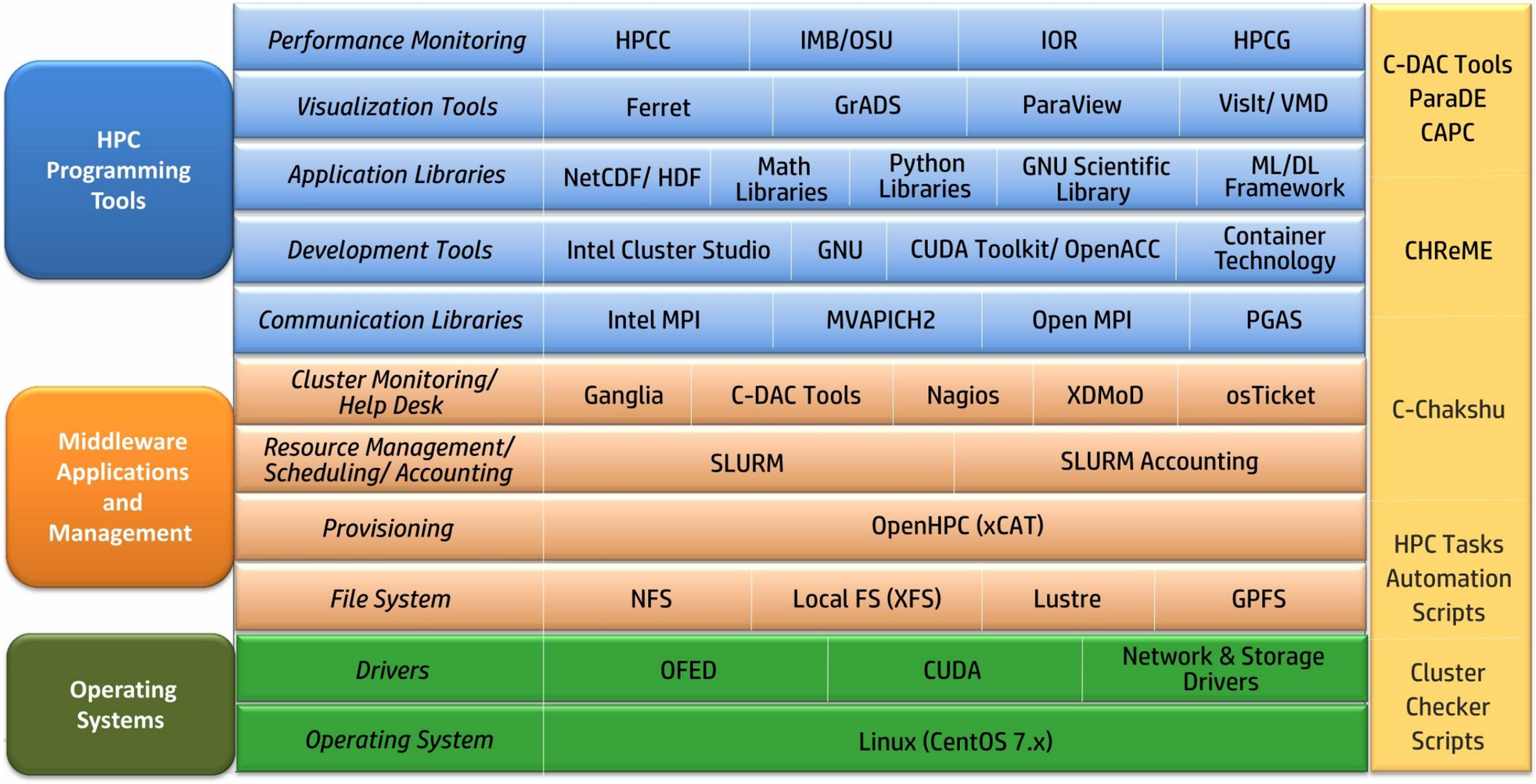
2 Embedded Lustre Parallel File System storage appliance with redundant controller

25 GB/s sustained read and write performants

8 x 100Gbps InfiniBand interconnect.



Software Stack



NSM Clusters – Applications, Tools, Programming Models- AI & HPC

HPC Applications	**Bio-informatics**	MUMmer, HMMER, MEME, PHYLIP, mpiBLAST, ClustalW	**Visualization Programs**	GrADS, ParaView, VisIt, VMD
	Molecular Dynamics	NAMD (CPU & GPU), LAMMPS(CPU & GPU), GROMACS	**Dependency Libraries**	NetCDF, PNETCDF, Jasper, HDF5, Tcl, Boost, FFTW
	Material Modeling, Quantum Chemistry	Quantum-Espresso, Abinit, CP2K, NWChem,	**Programming Models**	MPI, OpenMP, OpenACC, CUDA, PGAS, Pthreads
	CFD	OpenFOAM, FDS, SU2	**Installed additional applications, libraries, tools as per requirements from users**	
	Weather, Ocean, Climate	WRF, RegCM, MOM, ROMS		
	Disaster Management	ANUGA Hydro		
AI/ ML/ DL Tools/ Technologies	**DL Frame work:** TensorFlow , keras, theano, pytorch, scikit-learn,scipy, cuDNN			
	Data Science: Numpy , RAPIDS			
	Distributed DL Framework: TensorFlow with Horovod			
	Container Technology: enroot			
	JupyterHub: DL application development platforms and web based IDE			

Job Scheduler - SLURM

- When you login to HPC cluster, you land on **Login Nodes**
  - Login nodes are not meant to run jobs
  - These are used to submit jobs to **Compute Nodes.** You can setup your files and data on Login node. If compilation/installation of an application is required, user must do it on login node.

- To submit job on the cluster, you need to write a scheduler job script

- SLURM - Simple Linux Utility for Resource Management

- It is a workload manager that provides a framework for job queues, allocation of compute nodes, and the start and execution of jobs.

SLURM

How to set Environment ?

- By default no application is set in your environment. User must explicitly set required ones

- **module** is the utility (also command name) to enable use of applications / libraries / compilers available on the HPC cluster.

- Module structure on Cluster
  - apps/<application name>/version : Applications available on the cluste
  - compiler/<compiler name>/version : Compilers available on the cluster
  - lib/<library name>/version : Available libraries
  - ….

SLURM

➢ Some Important commands :

- **module avail** To see the available software installed on HPC system
  - list of precompiled applications
  - different compilers and libraries (compilers include GNU, Intel, PGI)
- **module list** Shows the currently loaded modules in your shell
- **module load <Name of the module>**
  - module load compiler/intel/2018.2.199 (to set Intel compilers version 2018 in your environment)
  - module load apps/namd/2.12/impi2018/cpu (to set NAMD app version 2.12 in your environment)

SLURM

➢ Some Important commands :

- **module unload <Name of the module>**
- **module purge** To clear all the loaded modules

Note : If you want the corresponding environment to be loaded into your shell by default, then you can set the environment via .bashrc file.

Caution : Try to avoid loading of too many modules or setup of unwanted variables via .bashrc.

SLURM

- **sbatch <script>** To submit the job on HPC cluster

- **squeue** To see the status of all jobs submitted on the cluster

  squeue -u <user name> To see status of user's jobs only. Also shows job-id.

SLURM: Useful Commands

SLURM

- **sinfo** Provides the basic information about the resources on HPC cluster such as
  - Partitions/queue such as for cpu / gpu / high memory nodes
  - Number of nodes for each type and their numbering/names
  - State of the nodes

scancel <job ID> To delete the submitted jobs

- **scontrol hold <Job ID>**
- **scontrol release <Job ID>**
- **scontrol show job <Job ID>**
- **srun** To get resources in interactive mode
  - `srun --nodes=1 --ntasks-per- node=1 --time=00:05:00 --pty bash -i`

SLURM: Sample Job Script for Serial Jobs

```
#!/bin/sh
#SBATCH -N 1                          # specify number of nodes
#SBATCH --ntasks-per-node=1           # specify number of CPU cores per node
#SBATCH --time=00:10:00               # specify maximum d    on of run
                                      in hours:minutes:se    s format
#SBATCH --job-name=openfoam           # specify job name
#SBATCH --error=job.%J.err            # specify error file na
#SBATCH --output=job.%J. out          # specify output file n
#SBATCH --partition=cpu               # specify type of resource such as
CPU/GPU/High Memory etc.
cd $SLURM_SUBMIT_DIR                  # change to directory from where job is submitted

#     Set your environment (e.g. load required compiler, application)
#     module load apps/mpiblast/1.6.0/intel

### Your command for Serial Execution (i.e. execution on one CPU core)

time <executable name with required parameters OR application command>
```

Resource specifications

(user must choose values appropriate for her/his job)

All necessary settings for your applications

SLURM: Sample Job Script for Parallel Jobs on CPU

```
#!/bin/sh
#SBATCH -N 1                          # specify number of nodes
#SBATCH --ntasks-per-node=8           # specify number of CPU cores per node
#SBATCH --time=01:00:00               # specify maximum duration of run
                                        in hours:minutes:seconds format
#SBATCH --job-name=gromacs            # specify job name
#SBATCH --error=job.%J.err            # specify error file name
#SBATCH --output=job.%J. out          # specify output file name
#SBATCH --partition=standard                  # specify type of resource such as
                                                                    CPU/GPU/High
cd  $SLURM_SUBMIT_DIR                 # change to directory from where job is submitted

### Set your environment (e.g. load required compiler,
application) module load apps/gromacs/15.2.2019/intel

### Your command for Parallel Execution (e.g. with MPI)

time mpirun -np $SLURM_NTASKS <executable name or application
```

SLURM: Job Script for Parallel Jobs on GPUs

```
#!/bin/sh
#SBATCH -N 1                                  # specify number of nodes requested
#SBATCH --ntasks-per-node=20  # specify number of CPU cores per node
#SBATCH --gres=gpu:1                          # specify no. of GPU devices per node
#SBATCH --time=01:0:00                        # specify maximum duration of run
                                                in hours:minutes:seconds format
#SBATCH --job-name=namdgpu                    # specify job name
#SBATCH --error=job.%J.err                    # specify error file name
#SBATCH --output=job.%J. out                  # specify output file name
#SBATCH --partition=gpu                       # specify type of resource i.e. GPU

cd $SLURM_SUBMIT_DIR                          # change to directory from where job is submitted

### Set your environment (e.g. load required compiler,
application) module load apps/namd/2.12/impi2018/cpu

### Your command for Parallel Execution with GPUs (e.g. with MPI)
time  mpirun  -np  $SLURM_NTASKS  <executable name or application command>
```

Access Methods

Command line interface through SSH

Hostname: ***paramutkarsh.cdac.in***

Tools for accessing:

Putty, WinSCP, MobaXterm

PARAMUtkarsh Website (paramutkarsh.cdac.in)

paramutkarsh.cdac.in

NATIONAL SUPERCOMPUTING MISSION
INFRASTRUCTURE | APPLICATIONS | R&D | HRD

Search ...

HOME | ABOUT US | INFRASTRUCTURE | FOR USERS | ACCESS TO CLUSTER | GALLERY | CONTACT US

Enhance your Business to next level using PARAM Utkarsh

Micro, Small and Medium Enterprises (MSME) are the backbone of the Indian economy, and many could use high performance computing (HPC) to enhance their business. However, it can be challenging for MSMEs to adopt HPC. They may have little in-house expertise, limited access to hardware, or be unable to commit resources to a potentially risky endeavour. This is where National Supercomputing Mission (NSM) project comes in, by making it easier for MSMEs to try out their ideas for utilising HPC to enhance their business, for example to improve product quality, reduce time to delivery, or create innovative new services.

Helpdesk: PARAM Utkarsh

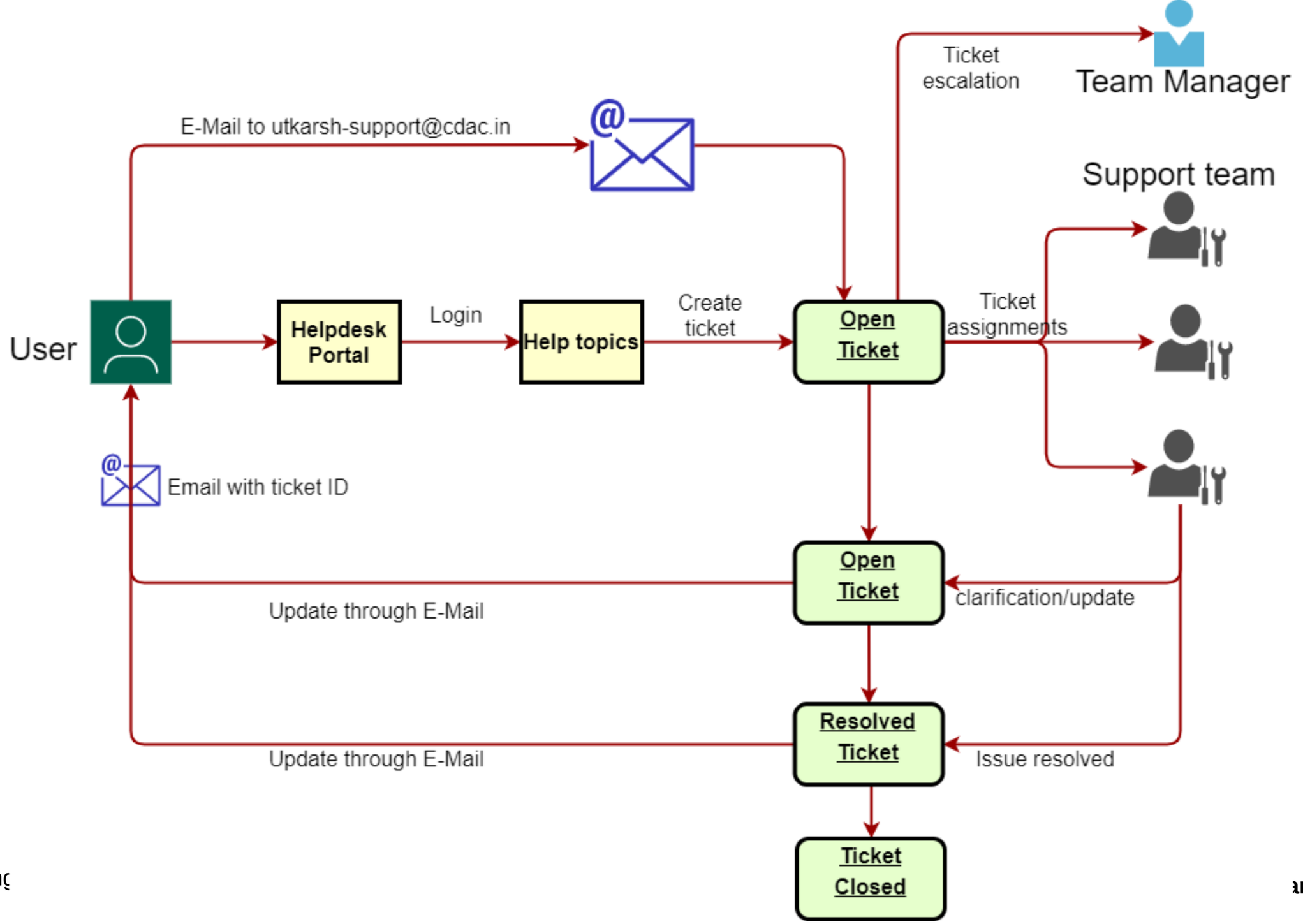
One-stop solution for reporting all of your problems.

A dedicated team is designated to address all the issues of PARAM Utkarsh.

Ticket is assigned to team not individual.

Ticket creation.

- **Through ticketing system (PARAM Utkarsh account required)**
- **Through email (No need of an account)**


Ticket escalation
Team Manager
E-Mail to utkarsh-support@cdac.in
Support team
User
Helpdesk Portal
Login
Help topics
Create ticket
Open Ticket
Ticket assignments
Email with ticket ID
Open Ticket
Update through E-Mail
clarification/update
Resolved Ticket
Update through E-Mail
Issue resolved
Ticket Closed

paramutkarsh.cdacb.in/support/

Support Center

Guest User | Sign In

Support Center Home | Open a New Ticket | Check Ticket Status

Welcome to the Support Center

Open a New Ticket

Check Ticket Status

In order to streamline support requests and better serve you, we utilize a support ticket system. Every support request is assigned a unique ticket number which you can use to track the progress and responses online. For your reference we provide complete archives and history of all your support requests. A valid email address is required to submit a ticket.

paramutkarsh.cdacb.in/support/open.php

Support Center

Guest User | Sign In

Support Center Home | Open a New Ticket | Check Ticket Status

Sign in to Utkarsh Support

To better serve you, we encourage our Clients to register for an account.

Email or Username

Password

Sign In

Not yet registered? Create an account

I'm an agent — sign in here

If this is your first time contacting us or you've lost the ticket number, please open a new ticket

Contact Details

E-Mail: utkarsh-support@cdac.in

Phone: 080-25093400

Facebook: https://www.facebook.com/CDACBengaluru

Twitter: https://twitter.com/CDACBengaluru

YouTube: https://www.youtube.com/channel/UCh-fxwvQbIeAxgE7sallBvg

Thank you