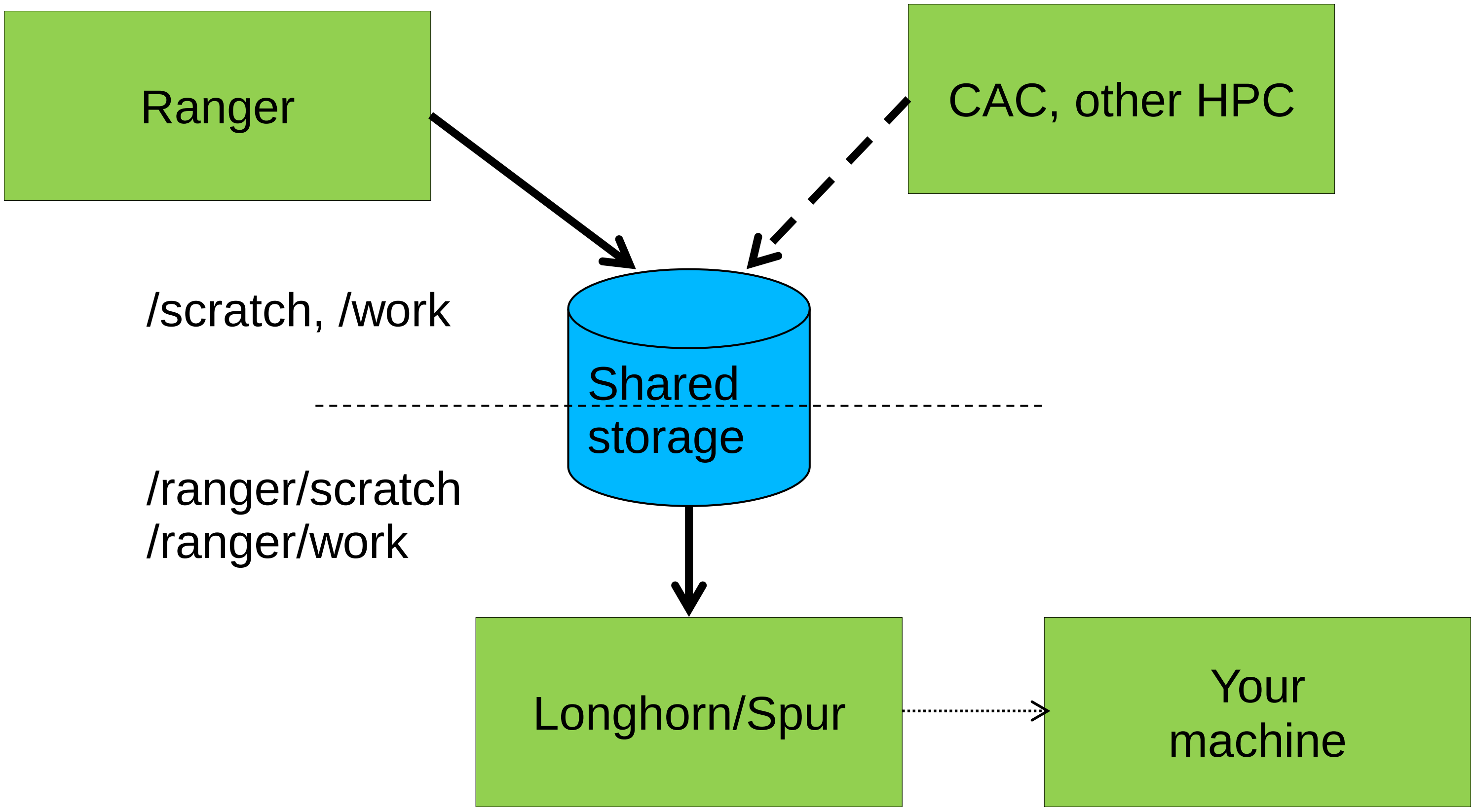
Cornell University
Center for Advanced Computing

Remote and Collaborative Visualization

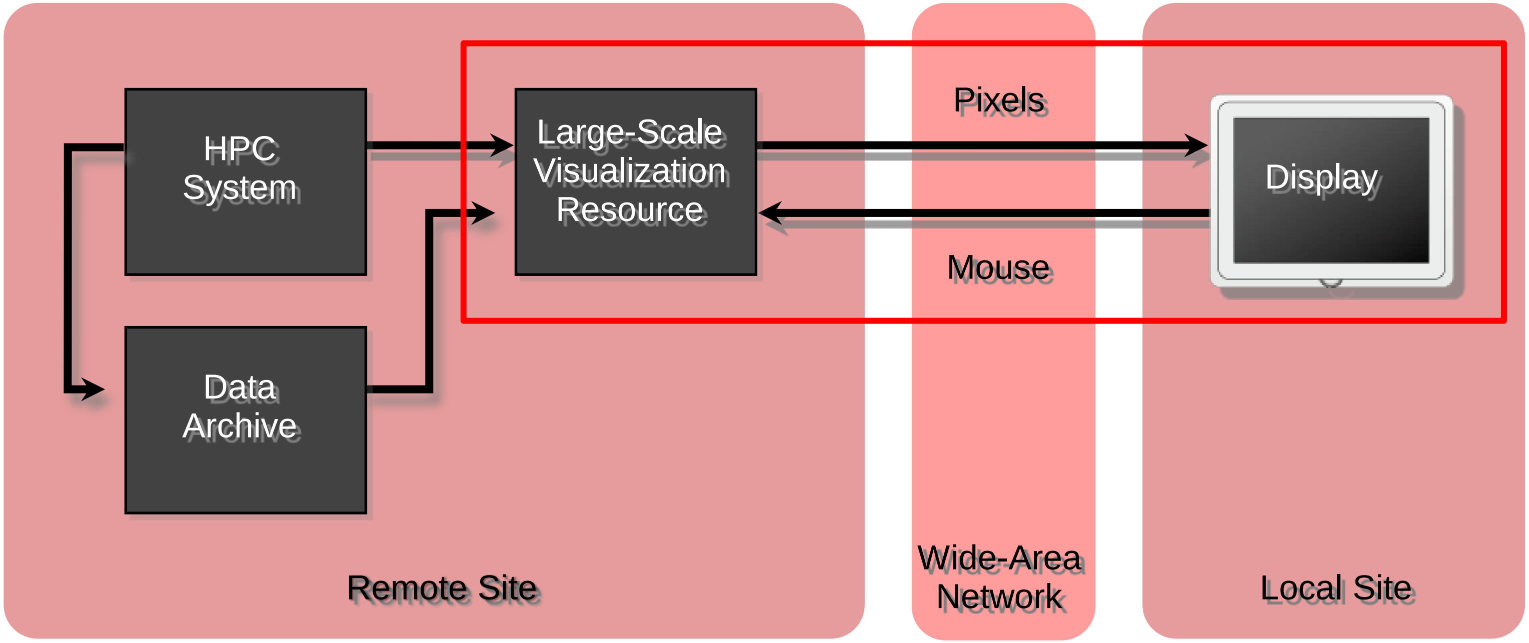
Aaron Birkland
Cornell Center for Advanced Computing

Data Analysis on Ranger
January 2012

Large Data, Remote Systems

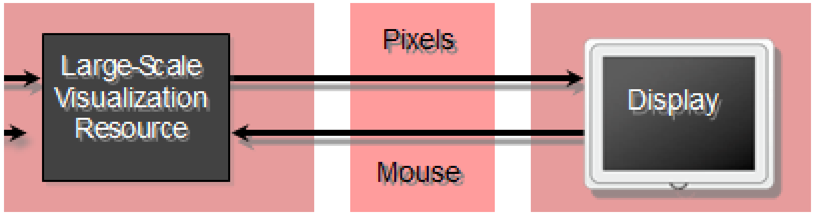


Remote Visualization Model



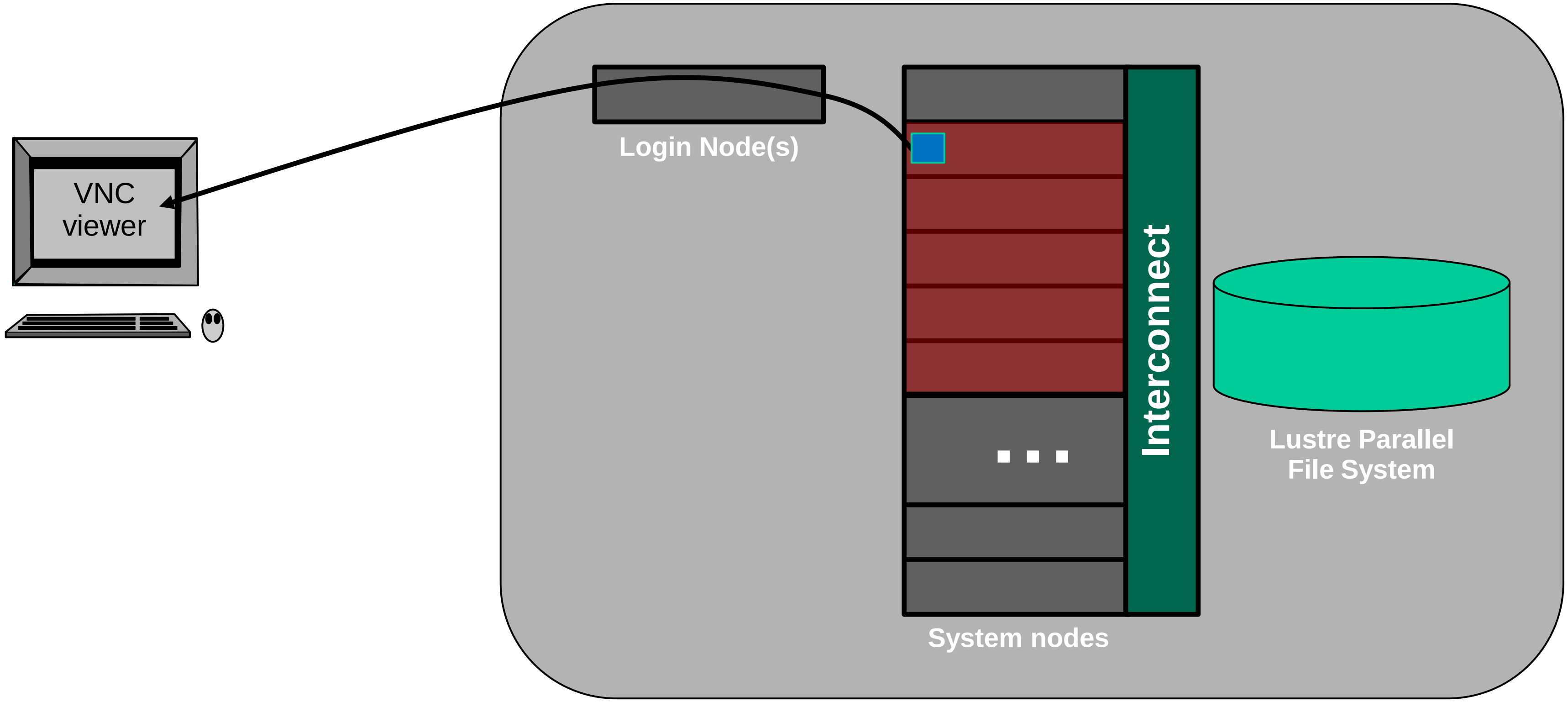
VNC

- Desktop process runs on remote *server*: vnc session
  - Windows, applications, mouse position
- Rendering occurs on server
  - Render on remote GPU. Send pixels to client
- Collaboration
  - Many can join vnc session, share control of mouse.
- VNC password to protect *session*
  - Share passwd with collaborators! Don’t use login passwd!!



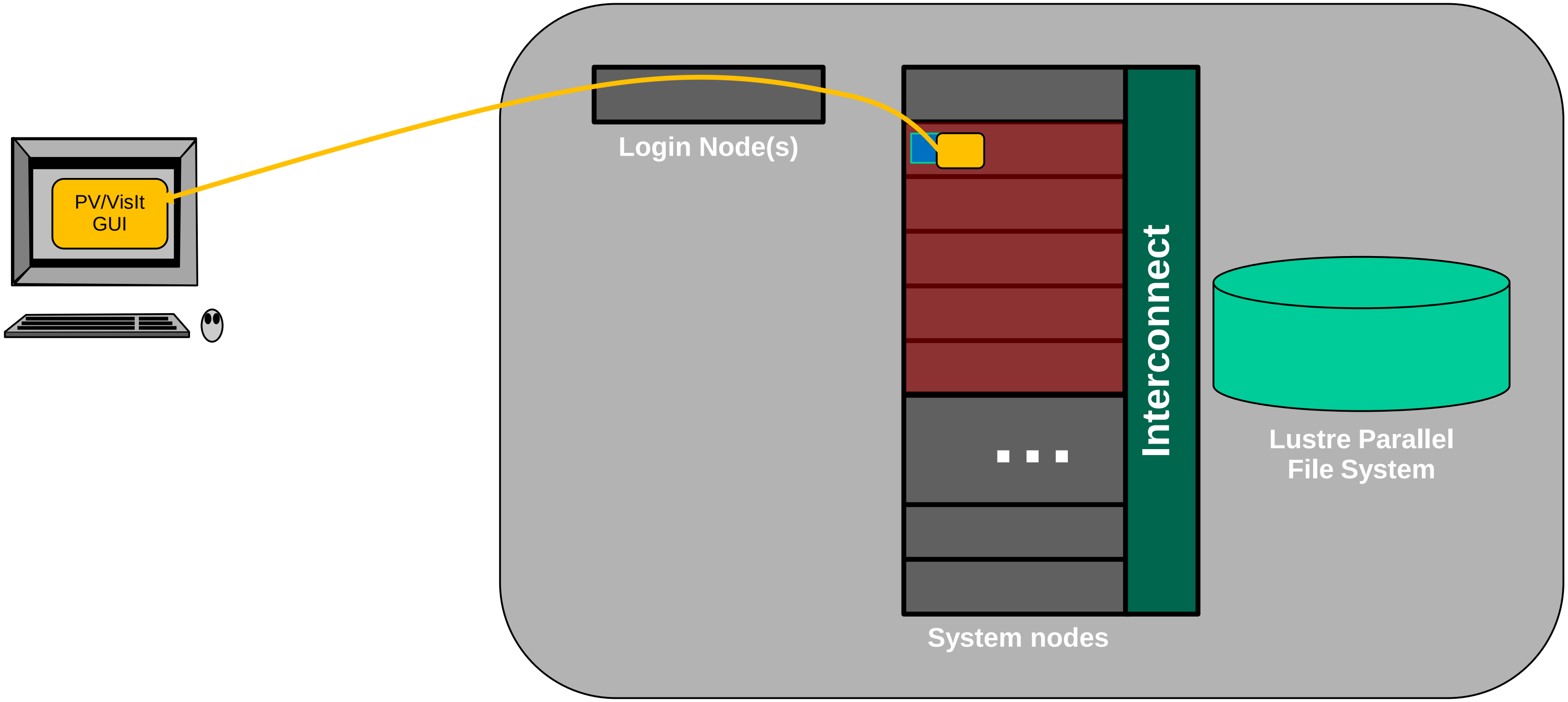
Visualization session

1. Allocate set of nodes on visualization system. This will start a VNC server one one node, which you will connect



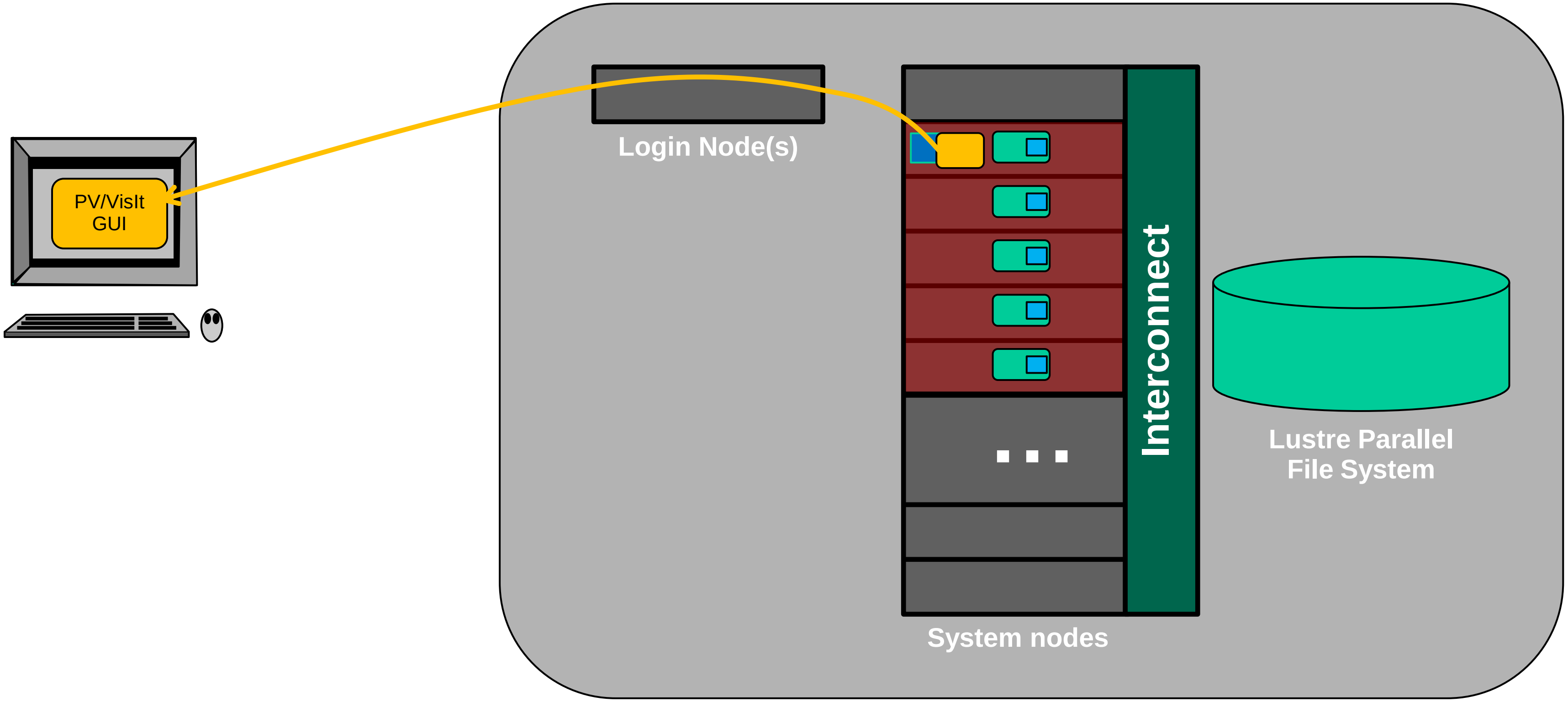
Visualization session

2. From that desktop, launch the PV or VisIt Client App



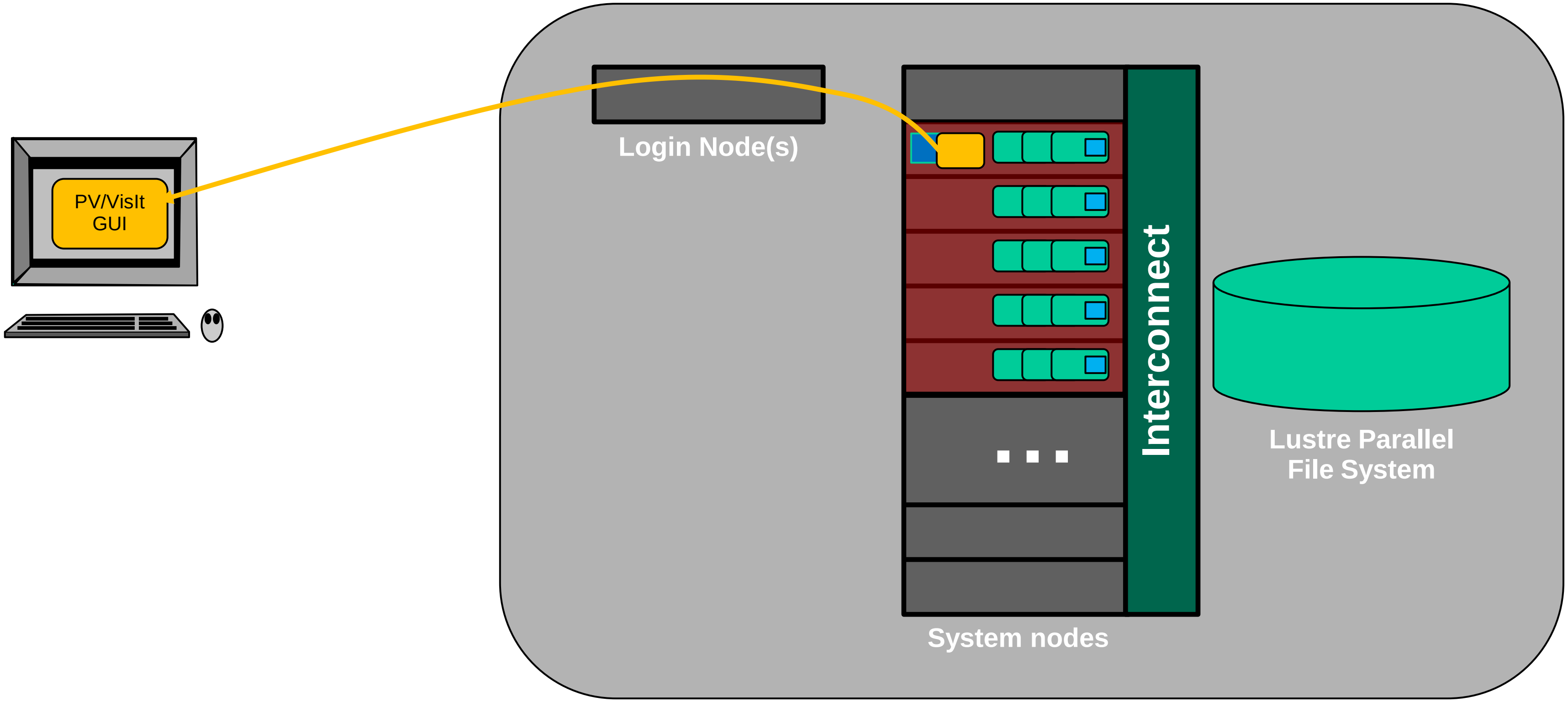
Visualization session

3. Start Paraview or VisIt Server Processes
   - PV, from command line; VisIt through GUI



Visualization session

4. Multiple processes/node to take advantage of multiple cores/node -- wayness



Wayness: processes per node

Process memory is limited to 1/n of node memory for each of n processes on that node If you need large memory per process, use fewer processes per node

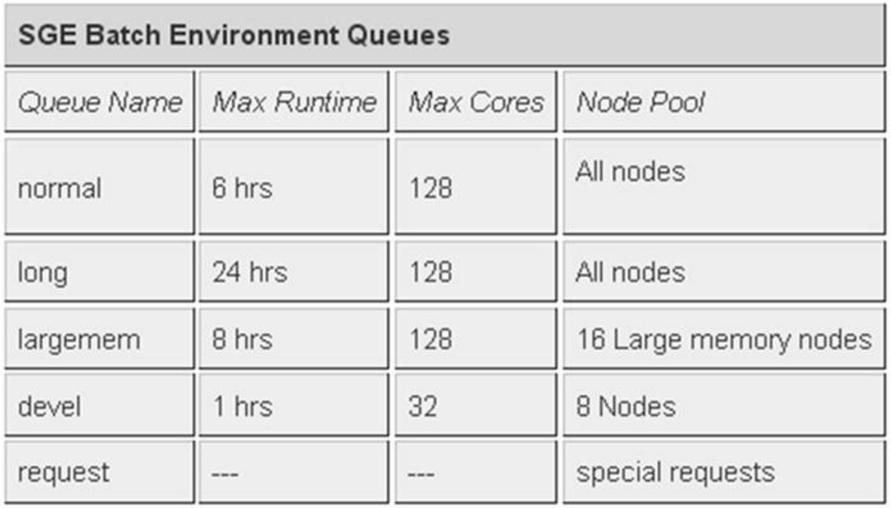
Why would you need to?

- Data is partitioned in large chunks
- Visualization algorithms can expand the data

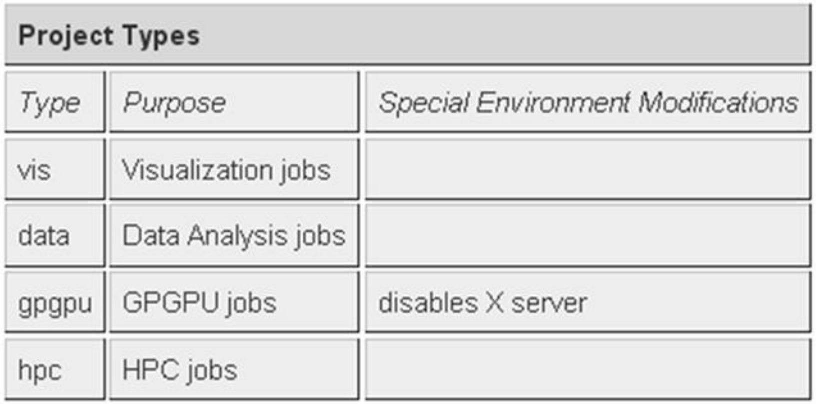
Way-ness is set up at *allocation time*

- Parallel jobs launched from VNC desktop will adopt the way-ness specified when the VNC server job is launched

Longhorn Queues



SGE Batch Environment Queues			
Queue Name	*Max Runtime*	*Max Cores*	*Node Pool*
normal	6 hrs	128	All nodes
long	24 hrs	128	All nodes
largemem	8 hrs	128	16 Large memory nodes
devel	1 hrs	32	8 Nodes
request	---	---	special requests



Project Types		
Type	*Purpose*	*Special Environment Modifications*
vis	Visualization jobs	
data	Data Analysis jobs	
gpgpu	GPGPU jobs	disables X server
hpc	HPC jobs	

qsub –q normal –P vis –pe 4way 16 /share/doc/sge/job.vnc

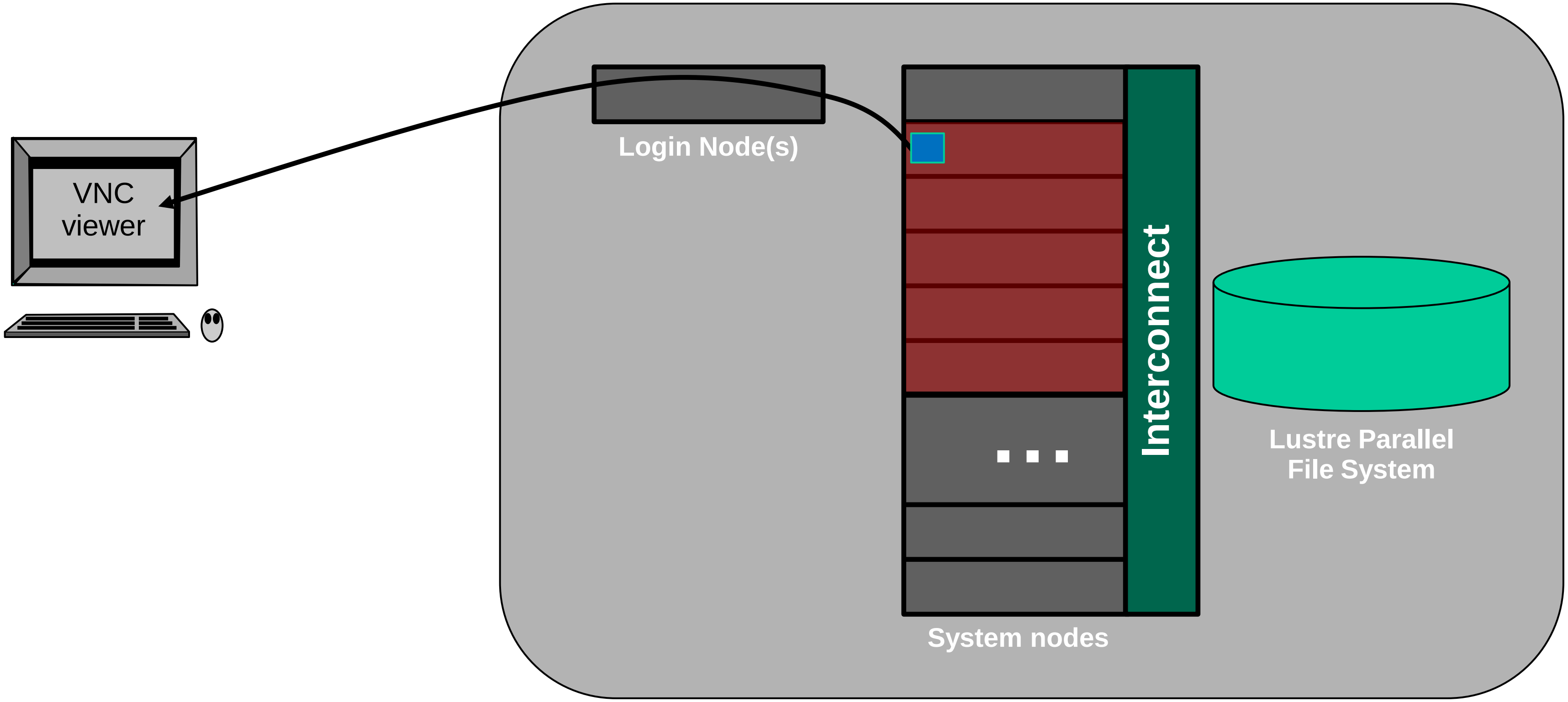
Longhorn Allocations

```
qsub –q normal –P vis –pe 4way 40
/share/doc/sge/job.vnc
```

- Wayness and number of nodes matter most
- Cores must be allocated in blocks of 8
  - 8 cores = 1 node
  - In our example above, we ask for **5 nodes** (5*8=40 cores)
- Number of MPI processes (i.e. parallel visualization backends) will be automatically limited by wayness parameter
  - 4 processes per node, **20 total parallel processes**

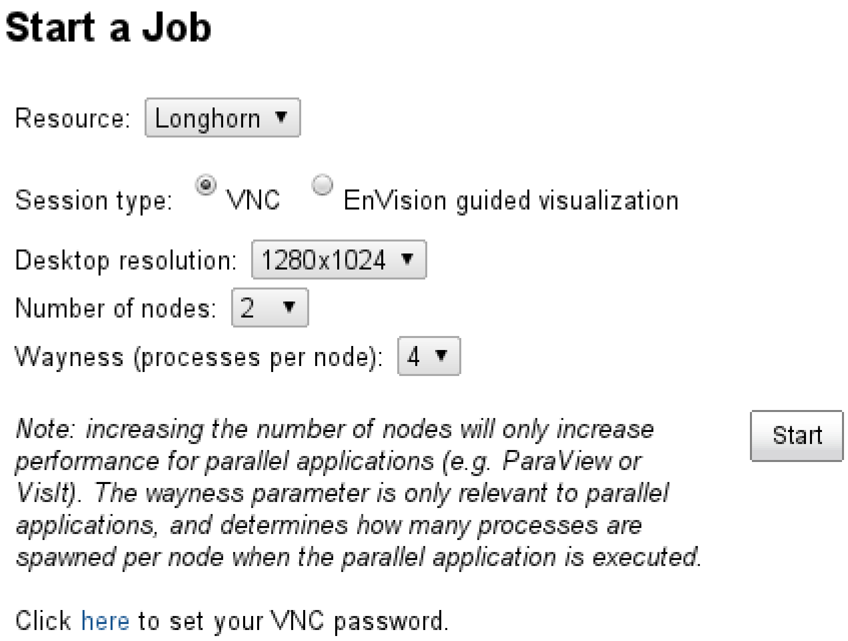
Longhorn Allocations

qsub –q normal –P vis –pe 4way 40 /share/doc/sge/job.vnc



Longhorn Allocations: Portal

- **Very** easy to use.
- Built-in VNC viewer
- Set/change vnc session password
- Shows vnc server address to connect remote clients



Start a Job

Resource: Longhorn ▾

Session type: ● VNC ○ EnVision guided visualization

Desktop resolution: 1280x1024 ▾

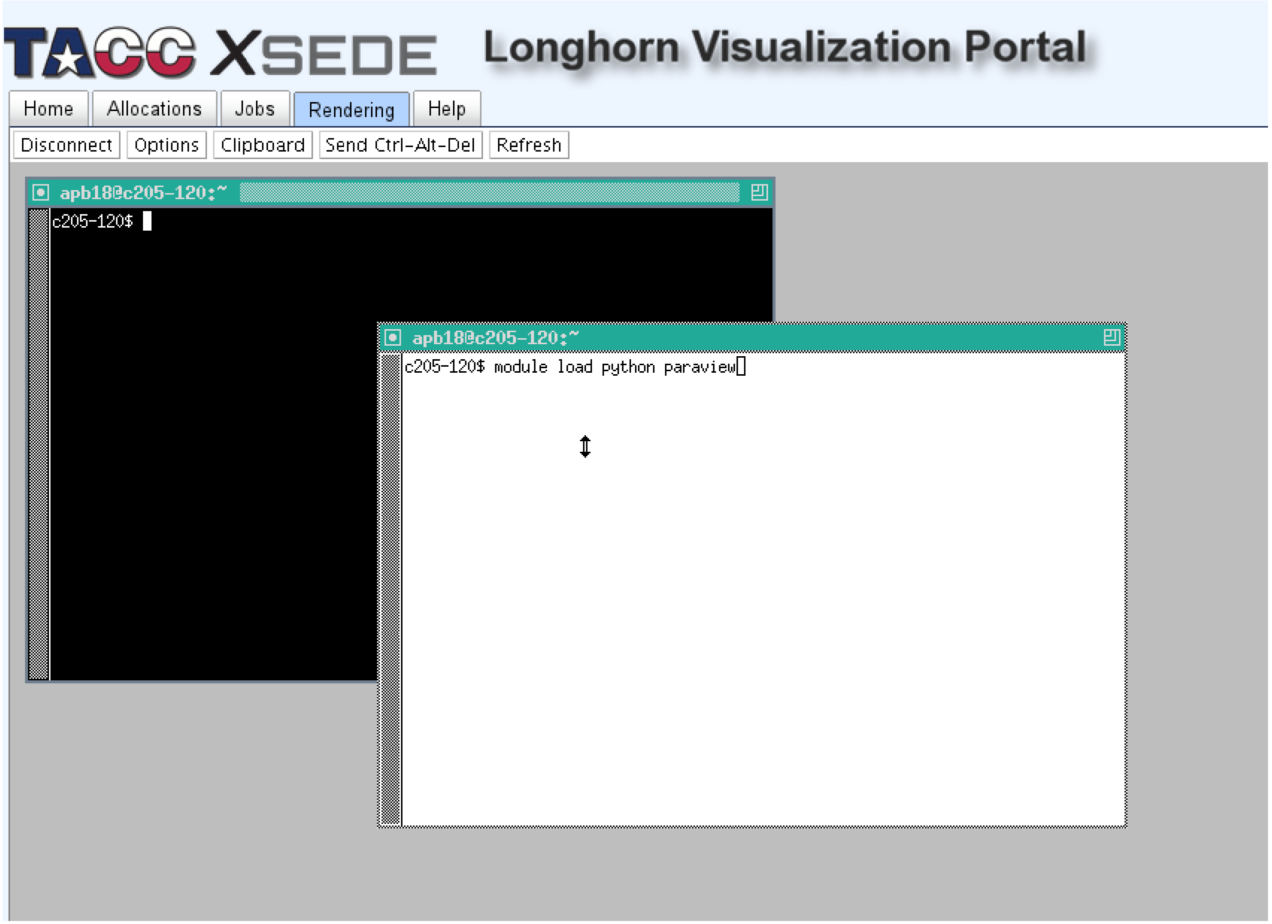
Number of nodes: 2 ▾

Wayness (processes per node): 4 ▾

Note: increasing the number of nodes will only increase performance for parallel applications (e.g. ParaView or VisIt). The wayness parameter is only relevant to parallel applications, and determines how many processes are spawned per node when the parallel application is executed.

Click here to set your VNC password.

VNC Desktop



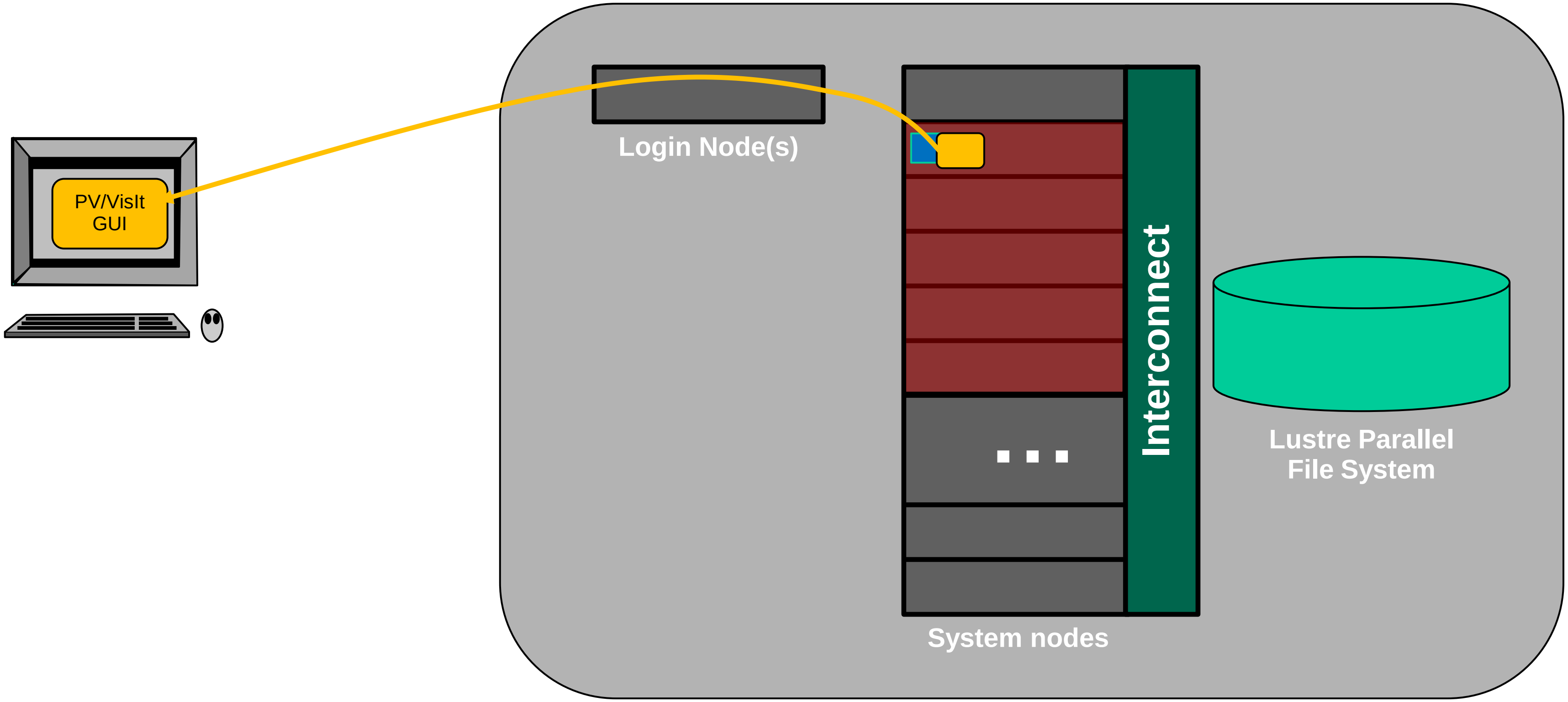
```
c205-120$
```

```
c205-120$ module load python paraview
```

Launching vis applications

- Applications using openGL (i.e. all visualization apps) need to be wrapped with `vglrun <app>`
  - This is a workaround for the fact that vnc servers do not support openGL natively
- This starts the visualization GUI only.
  - Parallel backends are launched on-demand by visualization app, or manually by user
  - If not using parallel mode, then you’re done!
- Visit simply asks if you want parallel or serial mode
  - Params automatically determined by session params
- Paraview needs explicit command or manual intervention

Launching vis applications

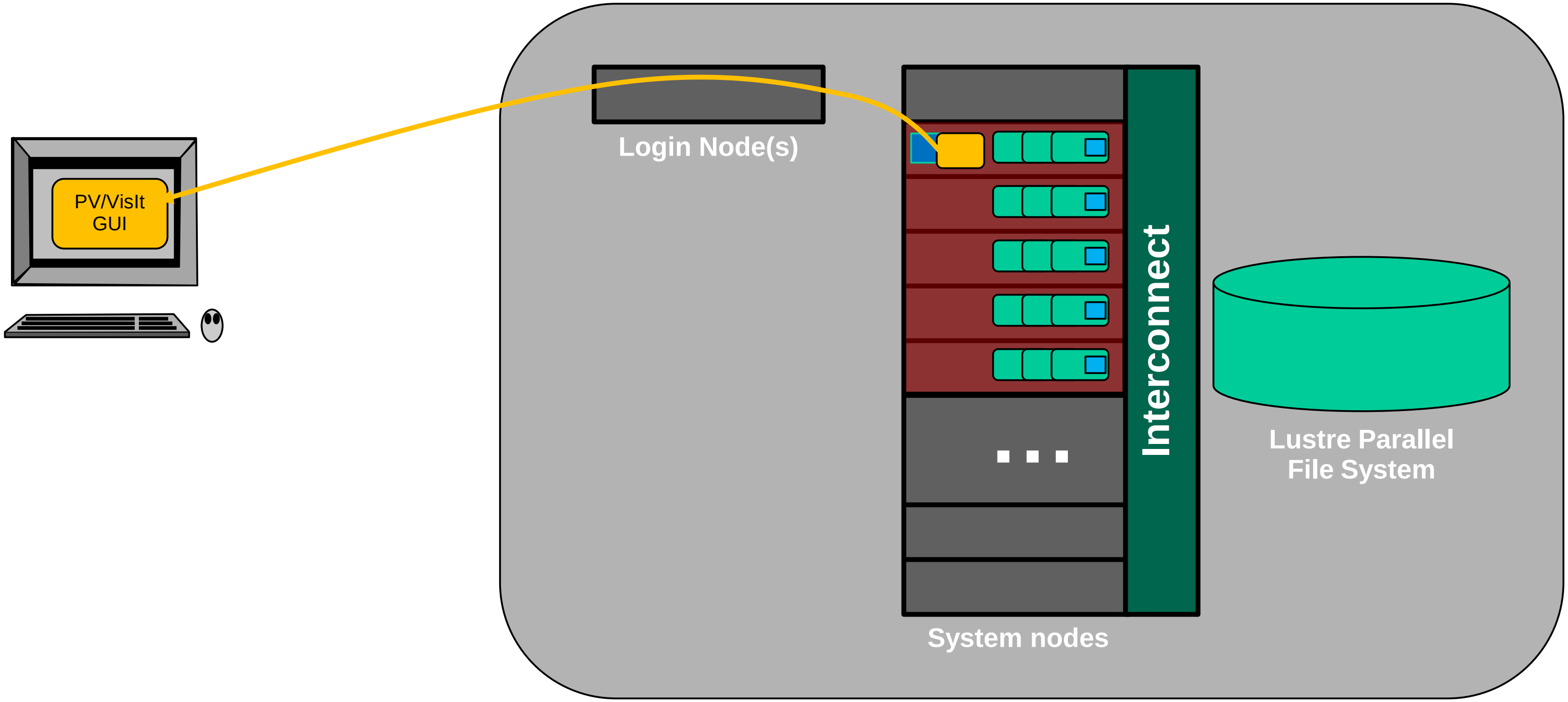


Launching ParaView backends

```
(csh)    env NO_HOSTSORT=1 ibrun tacc_xrun pvserver
```

- `pvserver`: This is the paraView backend
- `tacc_xrun`: wrapper to allow access to the GPU, but prevent from requiring own window on desktop
- `ibrun`: wrapper to execute MPI programs on Longhorn
- `env NO_HOSTSORT=1`: tells ibrun not to optimize placement of jobs
  - Otherwise, root backend process can be on different machine than GUI. GUI won’t be able to connect!

Remote, Parallel Visualization


PV/VisIt
GUI
Login Node(s)
Interconnect
. . .
System nodes
Lustre Parallel
File System