An introduction to SJTU π supercomputer

SJTU HPC Center
hpc@sjtu.edu.cn

Center for High Performance Computing, SJTU
http://hpc.sjtu.edu.cn

Apr 24h, 2017

Part I: Overview

SJTU π's Capability

- Theoretical performance: 385 TFlops
- Peak performance: 231 TFlopos
- Storage Bandwidth: 13GByte/s (100MByte/s per thread)
- Storage Capacity: 3PB
- 56Gbps Infiniband netowrk with $< 2\mu s$ end-to-end delay
- More than 50% of compute capability comes from GPUs.

NO.1 Supercomputer among China universities in 2013. Still the biggest GPU cluster among China universities.

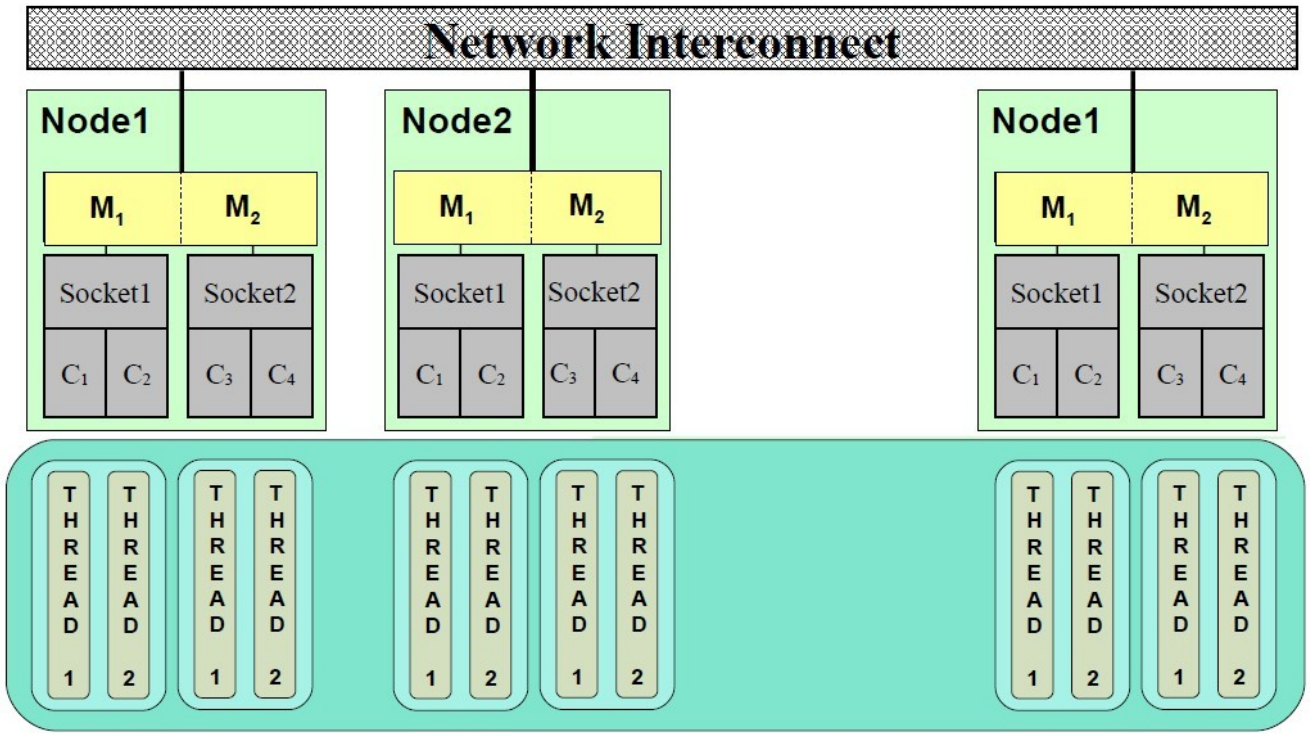
Who is using π?

- π servers more than 140 research groups, covering all STEM shcools in SJTU.
- π provides more than 20 million corehours per years (500x bigger than a workstation).
- Highlight applications: airplane noise analysis, material genomes, deeplearning-based speech recognition, rice sequencing, plasma physics and etc al.
- Rich opensource software available: GCC, OpenMP, MPI, BLAS, CUDA, CUDNN, OpenFOAM, Groamcs, NAMD and etc al.

Average utilization is above 75%.

π is a computer cluster

- Multiple nodes connected by ultra high speed networks
- A virtual computer under programming abstraction (OpenMP, MPI)
- CPUs with low clock frequency, high parallelism, high aggregated computer power



Different Compute Partitions on π

Partition	Num	CPU	Mem	GPUs
cpu	332	2xE52670 16c	64GB	
fat	20	2xE52670 16c	256GB	
gpu	50	2xE52670 16c	64GB	2xK20m
k40	5	2xE52670 16c	64GB	2xK40
k80	11	2xE52670v3 24c	96GB	2xK80
p100	4	2xE52680v3 24c	96GB	2xP100

Documents for new π users

- SSH login: https://pi.sjtu.edu.cn/doc/ssh
- SLURM job scheduling system: https://pi.sjtu.edu.cn/doc/slurm
- Software Modules:https://pi.sjtu.edu.cn/doc/modules
- Ganglia Monitoring: https://pi.sjtu.edu.cn/ganglia
- Accounting System: https://acct.hpc.sjtu.edu.cn

Part II: Job Management via SLURM scheduling system

Why SLURM, not LSF or other choices?

SLURM just works:

- Free and opensource
- Proven scalability and reliability
- Out-of-box fair-sharing job scheduling
- Debug friendly: SSH-login to compute hosts at running

Limitations of SLURM on π

- NO QoS or job quotas yet
- A max walltime of 7 days
- Job privacy is NOT enabled yet.
- Workdir is not recorded in the accounting system.

SLURM Overview

SLURM	Function
sinfo	Cluster status
squeue	Job status
sbatch [JOB_SCRIPT]	Job submission
scancle [JOB_ID]	Job deletion

`sinfo`: check cluster status

Host state: `drain`(something wrong), `alloc`(in full use), `mix`(in partial use), `idle`, `down`.

```
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
cpu*         up 7-00:00:00      1  drain node001
cpu*         up 7-00:00:00     31  alloc node[002-032]
gpu          up 7-00:00:00      4  alloc gpu[47-50]
fat          up 7-00:00:00      2  alloc fat[19-20]
k40          up 7-00:00:00      2  alloc mic[01-02]
k40          up 7-00:00:00      2   idle mic[03-04]
```

squeue: check job status

Job status: R(running), PD (Resources)(Pending).

```
$ squeue
JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REA
 2402       fat   add_upc  hpctheo PD       0:00      2 (Resources)
 2313       cpu    hbn310    physh  R   23:49:00      2 node[003,008]
```

Prepare to submit a job

- Workdir?
- Which partition or queue to use?
- How many CPU cores or nodes to use?
- How many CPU cores on each host?
- Whether GPUs are required?
- Max walltime to run?

`sbatch` usage

SYNOPSIS

```
sbatch jobsript.slurm
```

NO redirection symbol < is reuired!

`sbatch` options

SLURM	Meaning
`-n [count]`	Total processes
`--ntasks-per-node=[count]`	Processes per host
`-p [partition]`	Job queue/partition
`--job-name=[name]`	Job name
`--output=[file_name]`	Standard output file
`--error=[file_name]`	Standard error file
`--time=[dd-hh:mm:ss]`	Max walltime

`sbatch` options (continued)

SLURM	Meaning
`--exclusive`	Use the hosts exclusively
`-mail-type=[type]`	Notification type
`--mail-user=[mail_address]`	Email for notification
`--nodelists=[nodes]`	Job host preference
`--exclude=[nodes]`	Job host to avoid
`--depend=[state:job_id]`	Job dependency

A `sbatch` example (CPU)

```
#SBATCH --job-name=LINPACK
#SBATCH --partition=cpu
#SBATCH -n 64
#SBATCH --ntasks-per-node=16
#SBATCH --mail-type=end
#SBATCH --mail-user=YOU@EMAIL.COM
#SBATCH --output=%j.out
#SBATCH --error=%j.err
#SBATCH --time=00:20:00
```

A `sbatch` sample (GPU)

```
#SBATCH --job-name=GPU_HPL
#SBATCH --partition=gpu
#SBATCH -n 1
#SBATCH --gres=gpu:1
#SBATCH --exclusive
#SBATCH --mail-type=end
#SBATCH --mail-user=YOU@MAIL.COM
#SBATCH --output=%j.out
#SBATCH --error=%j.err
#SBATCH --time=00:30:00
```

Part III: Software modules

Fresh new modules for SLURM: `/lustre/usr/modulefiles/pi`

- Smart enough to derive the combination of compilers and MPI libraries.
- The number of modules is still growing.
- Please refer to `/lustre/usr/samples` for job submission.

Simplified module loading

```
$ module load gcc/4.9 openmpi/gcc49/1.8 fftw3/gcc49/openmpi8/3.3
```

v.s.

```
$ module load gcc/4.9 openmpi/1.8 fftw3/3.3
```

v.s.

```
$ module load gcc openmpi fftw3
```

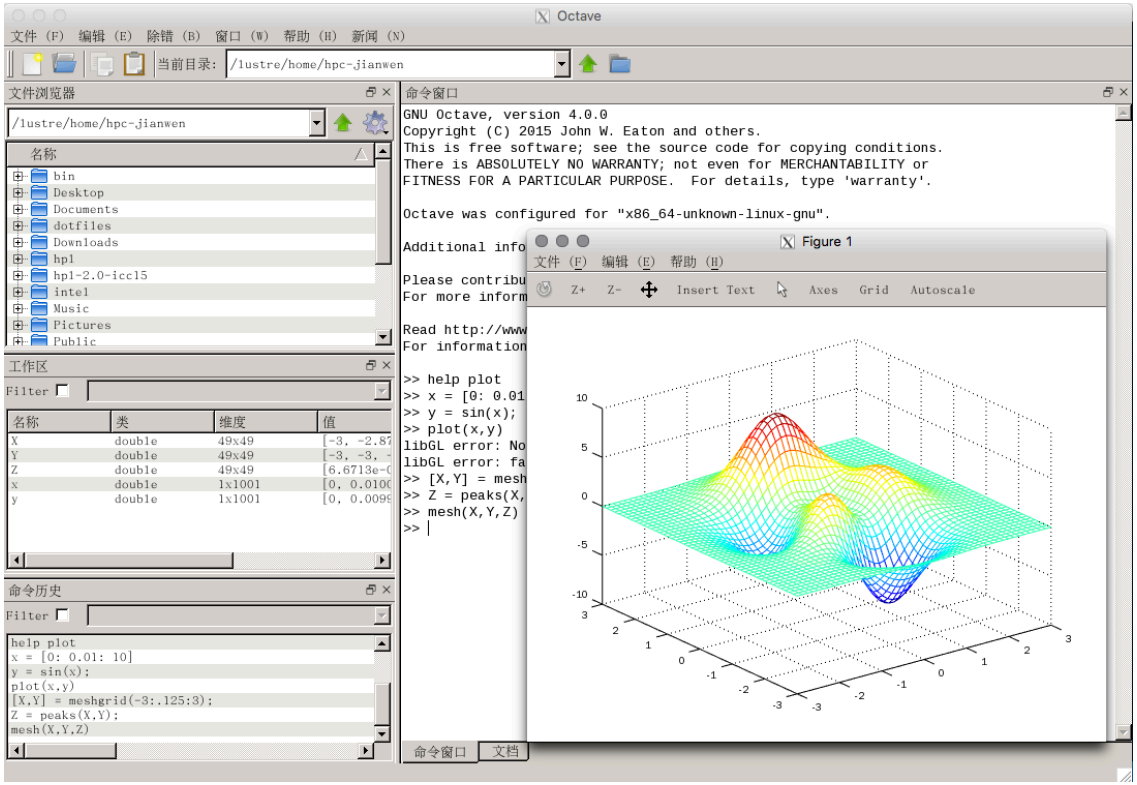
Libraries and software optimized for π supercomputer

```
gcc icc jdk perl python R pgi
impi openmpi mvapich2 mpich
mkl atlas lapack openblas mpc gmp mpfr gsl eigen
abysss samtools smufin gatk maq bwa bowtie
openfoam cgal gromacs
hdf5 netcdf scotch ffmpeg szip
cuda(cublas included) cudnn caffe mxnet cntk
```

Opensource alternatives of MATLAB and IDL

- GNU Octave https://www.gnu.org/software/octave/
- GNU Data Language (GDL)

Please refer http://hpc.sjtu.edu.cn/info/1027/1276.htm on using Octave and MATLAB usage.



Part IV: Tips for monitoring your jobs

Wait, is my application running well?

You can confirm your application's state by:

- Comparing performance between π and your laptop.
- Comparing performance between π and existing traces or benchmarks.
- Monitor the application, compute nodes more exactly, via http://pi.sjtu.edu.cn/ganglia.
- Asking administrators support@lists.hpc.sjtu.edu.cn for help.

Communicate with HPC administrators

Yelling "XXX is slow" doesn't help. Please report the following information:

- Job ID;
- Workdir;
- Job script;
- Website of your application;
- Expected results and what actually happened;

Email support@lists.hpc.sjtu.edu.cn is always preferable over wechat.

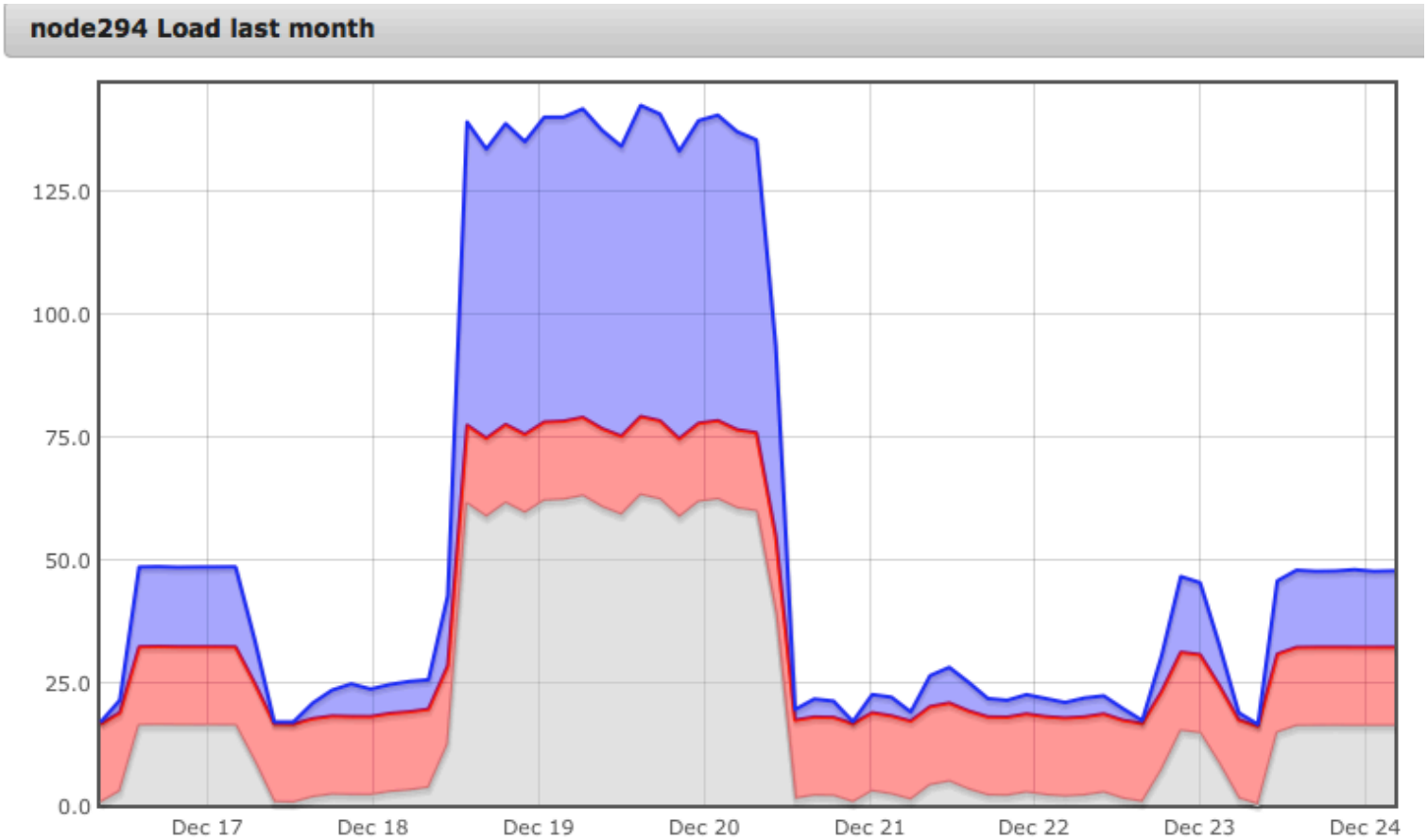
Monitor what? Load, CPU, Mem, Network

Please check http://pi.sjtu.edu.cn/ganglia:

- Load: The number of "threads", should be approximately 16 – the number of CPU cores
  - Below 16: starving
  - Above 16: overload
- CPU report: Charts in yellow color are good
  - `sys` and `wait` should be less than 5%.
- Mem: Do NOT exceed the physical capapcity
- Network: Ethernet traffic should be less than 1MB/s.

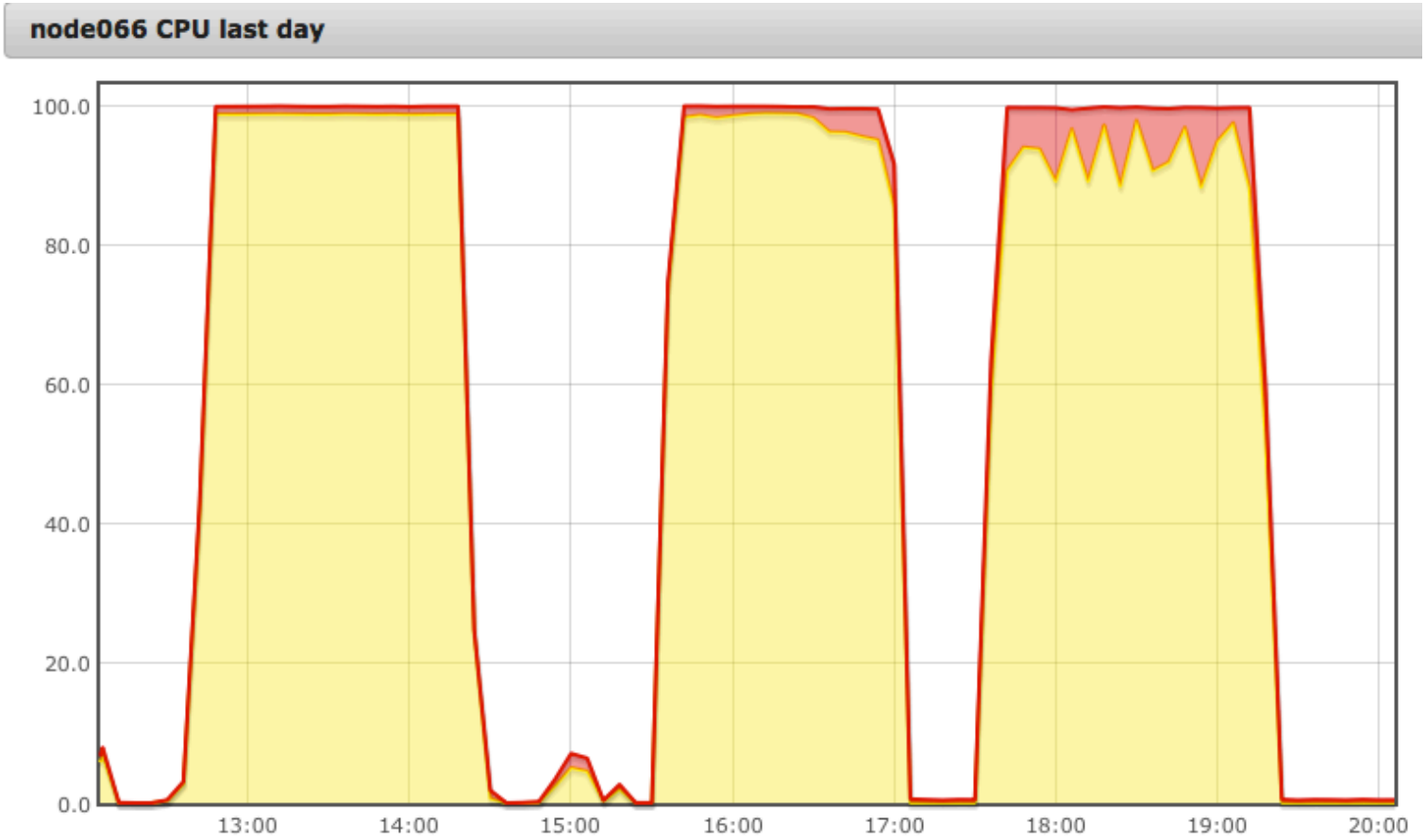
Case 1: Overload due to too many processes

Caused by incorrect setting of NO. of cores, or inbalanced load between nodes.



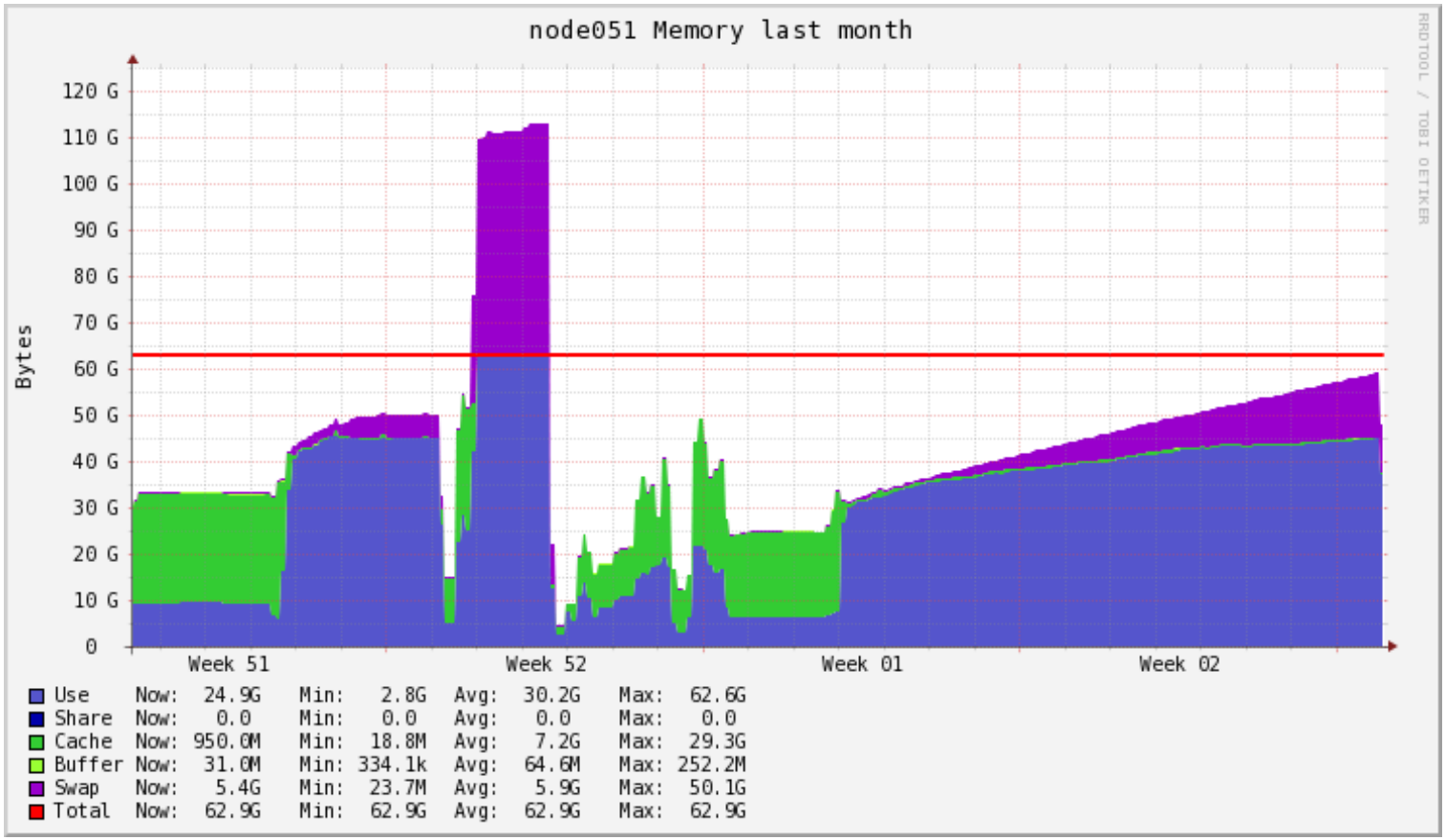
Case 2: Too high sys utilization

Caused by linking or loading incorrect MPI libraries, or hardware issue.



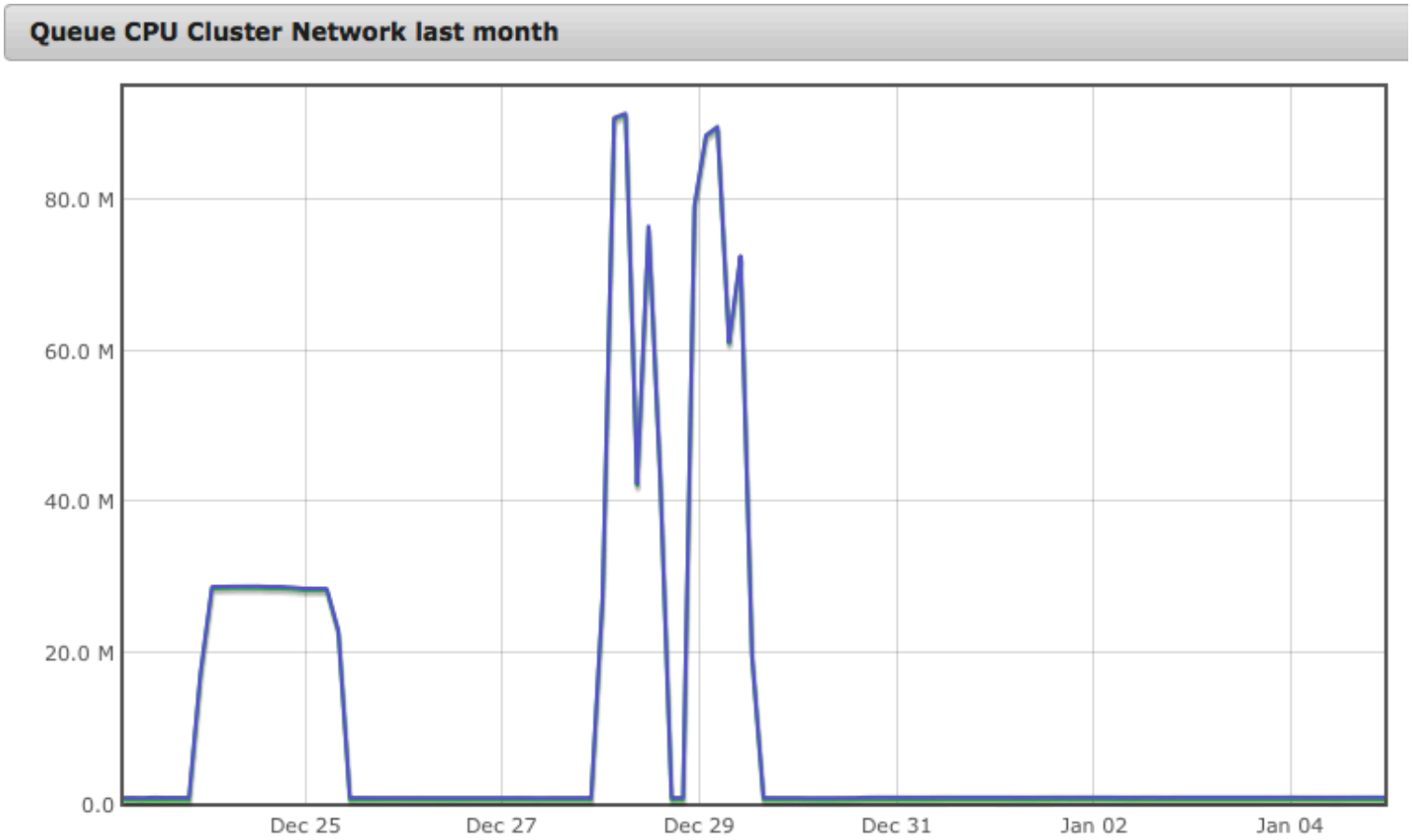
Case 3: Memory Usage Exceeding

The data is just too *fat*. Try the `fat` queue please.



Case 4: Inefficient Use of Ethernet

Caused by linking or loading incorrect MPI libraries, or Infiniband driver issue.



DO's

- SSH login to compute nodes during job execution.
- Use http://pi.sjtu.edu.cn/ganglia to monitor your jobs.
- Attach your username, jobid, workdir and error messages when reaching help from support@lists.hpc.sjtu.edu.cn.

DONT's

- NO `du` !!!
- NO parallel jobs on login nodes.

Reference

- *SJTU π documents* http://pi.sjtu.edu.cn/doc
- *ACCRE's SLURM Documentation* http://www.accre.vanderbilt.edu/?page_id=2154
- *Job samples for Pi supercomputer* http://pi.sjtu.edu.cn/doc/samples/
- *Remote Desktop via NoMachine* http://pi.sjtu.edu.cn/doc/rdp/
- *Environment Module on Pi* http://pi.sjtu.edu.cn/doc/modules/