The Parallel File System and I/O

SNUG TechTalk

SciNet, Toronto

File system recap

- 1,790 1TB SATA disk drives, for a total of 1.4PB
- Two DCS9900 couplets, each delivering:
  - 4-5 GB/s read/write access (bandwidth)
  - 30,000 IOP/s max (open, close, seek, . . .)
- Single *GPFS* file system on TCS and GPC
- I/O goes over Gb ethernet network on GPC (infiniband on TCS)
- File system is *parallel!*

SciNet

File system recap

location	quota	block-size	time-limit	backup	devel	comp
/home	10GB	256kB	unlimited	yes	rw	ro
/scratch	X TB	4MB	3 months	no	rw	rw

- There are quotas
- Home read-only from compute nodes!
- Big block sizes: *small files waste space*
- Issues are common to parallel file systems (Lustre, etc.) present in most modern supercomputers.
- Scratch quota per user oversubscribes disk space, so only for when you *temporarily* really needs a lot of disk space.
- Most users will need much less.

SciNet

File system recap

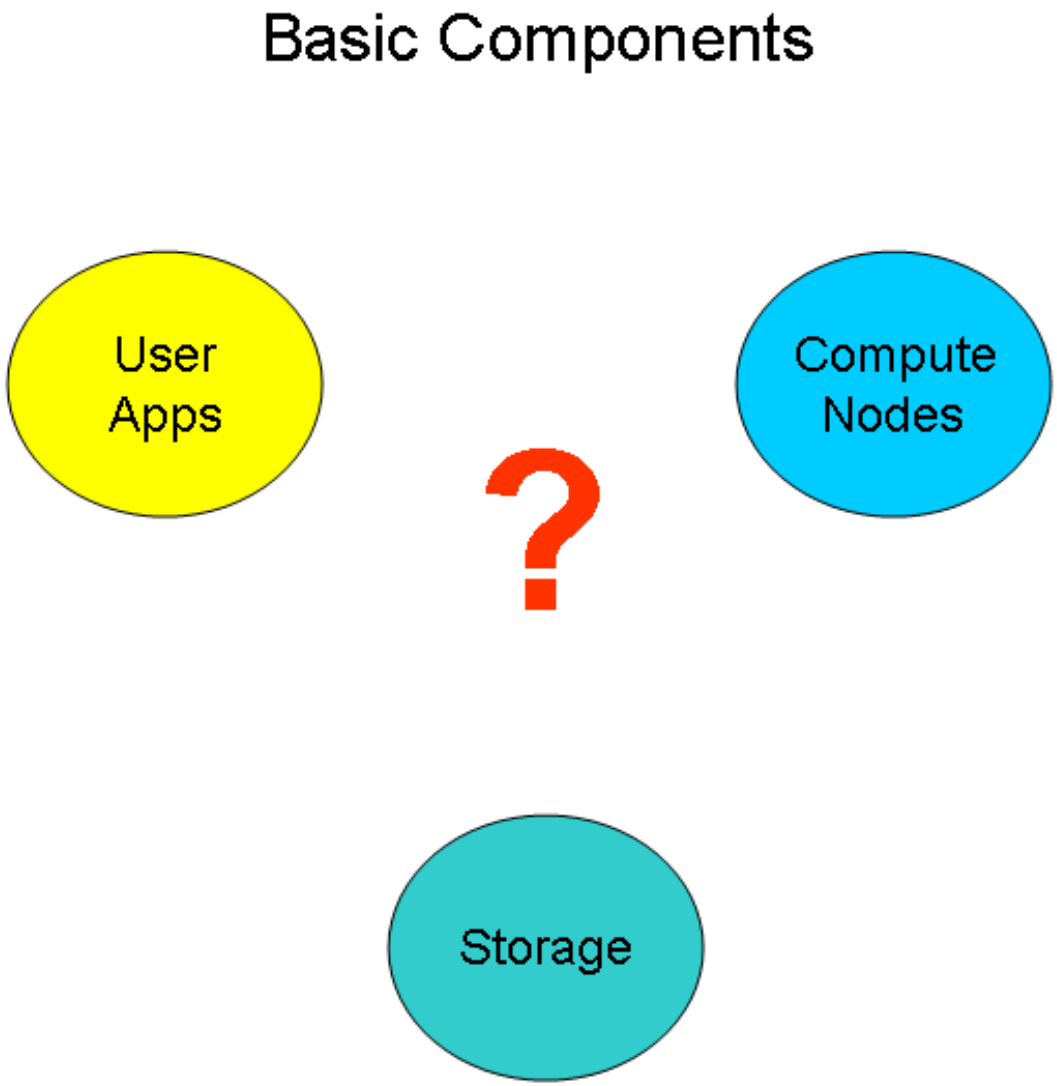
Scratch Policies

- Scratch is intended for active jobs
  (e.g. writing checkpoints and data during a run).
- Files are purged after 3 months
  (may need to reduce this to 2 months soon).
- Quotas on space and number of files will be tightened after this week's shutdown.

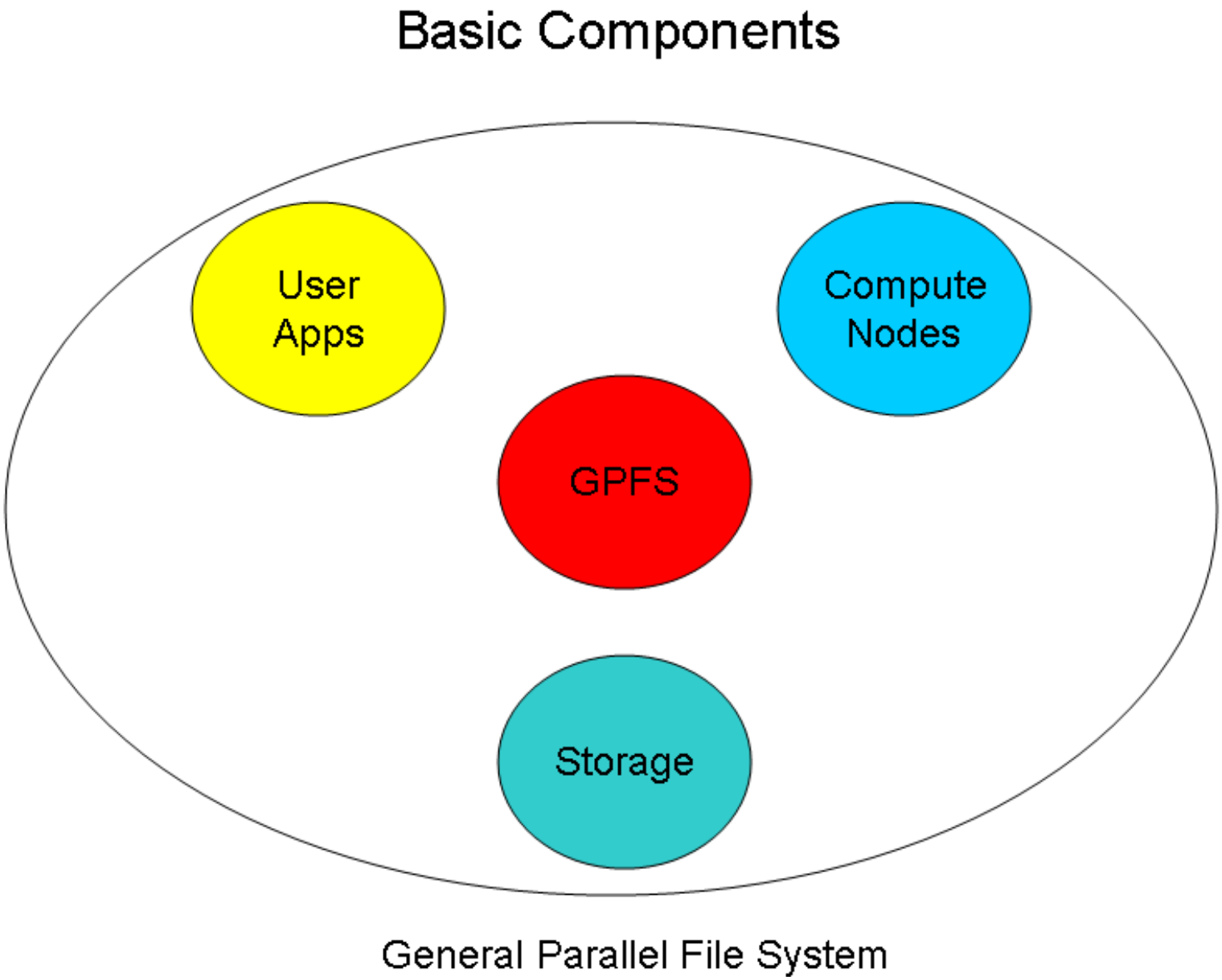
The file system is parallel, what does that mean?

SciNet

Shared file system



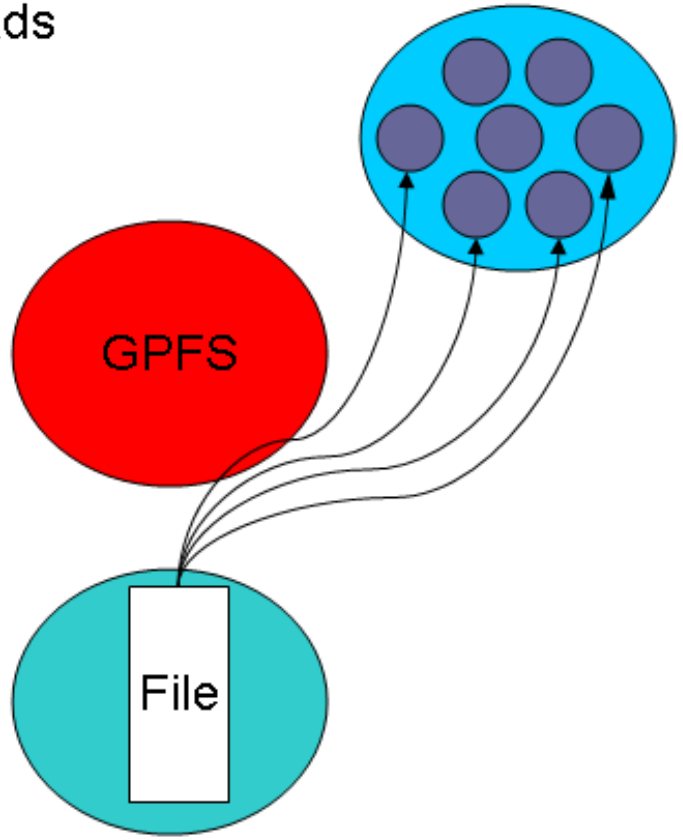
Shared file system



Shared file system

Basic Components

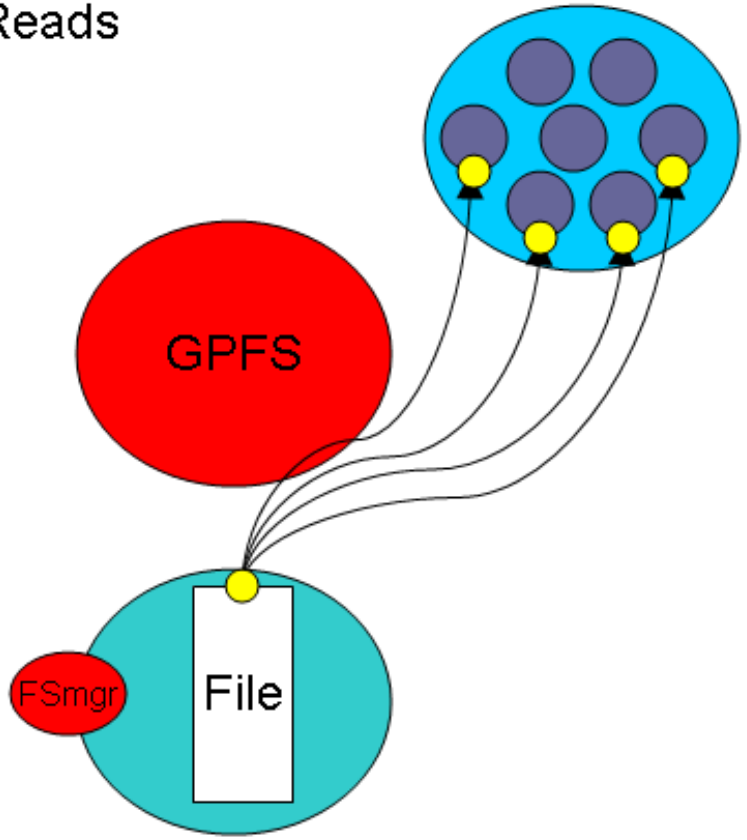
Parallel Reads



Shared file system

Basic Components

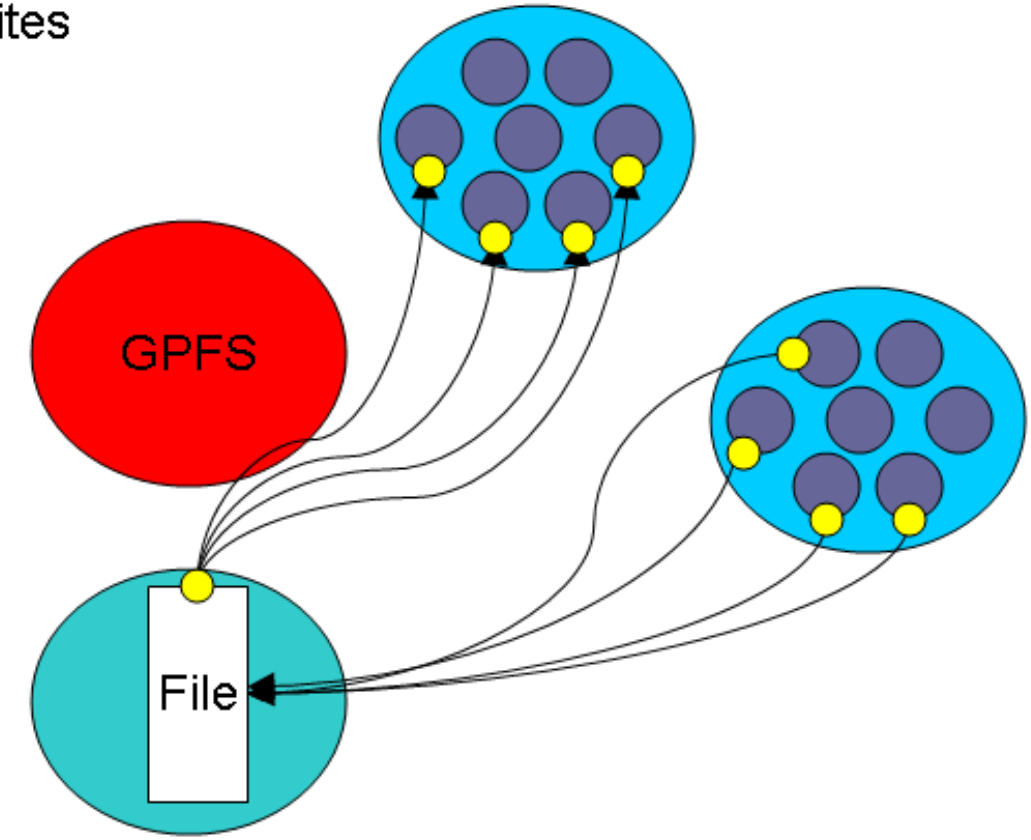
Parallel Reads



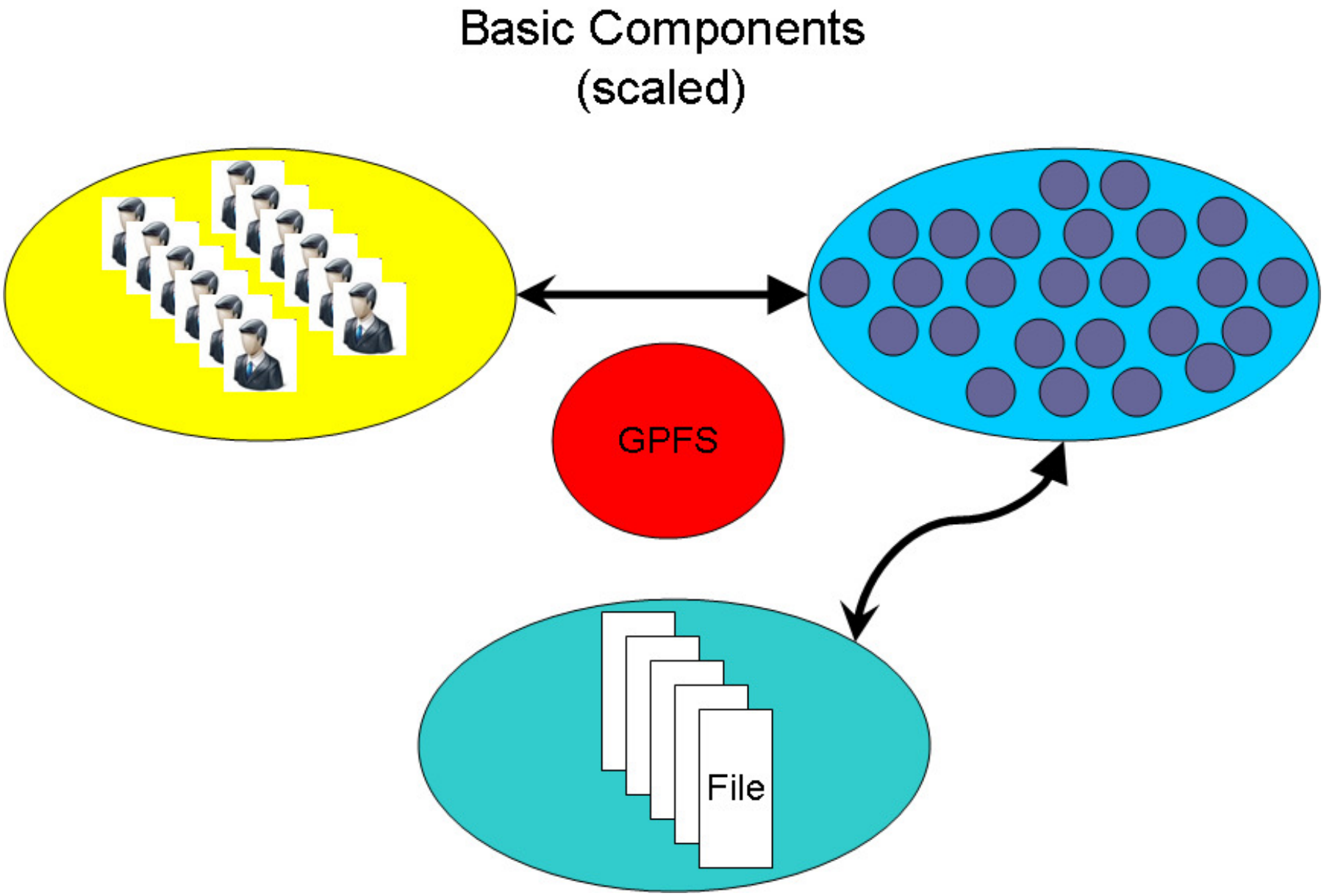
Shared file system

Basic Components

Parallel Writes

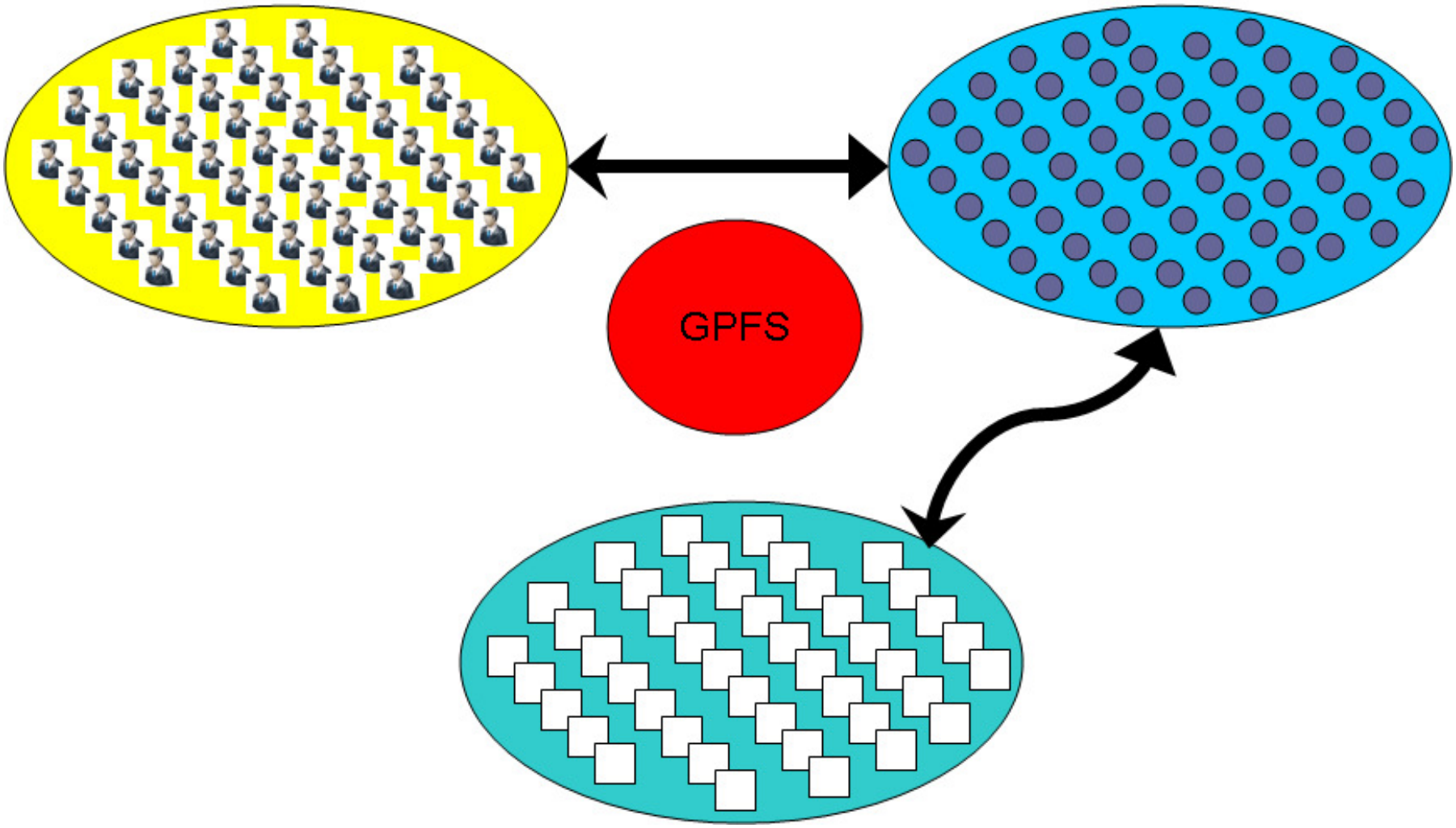


Shared file system



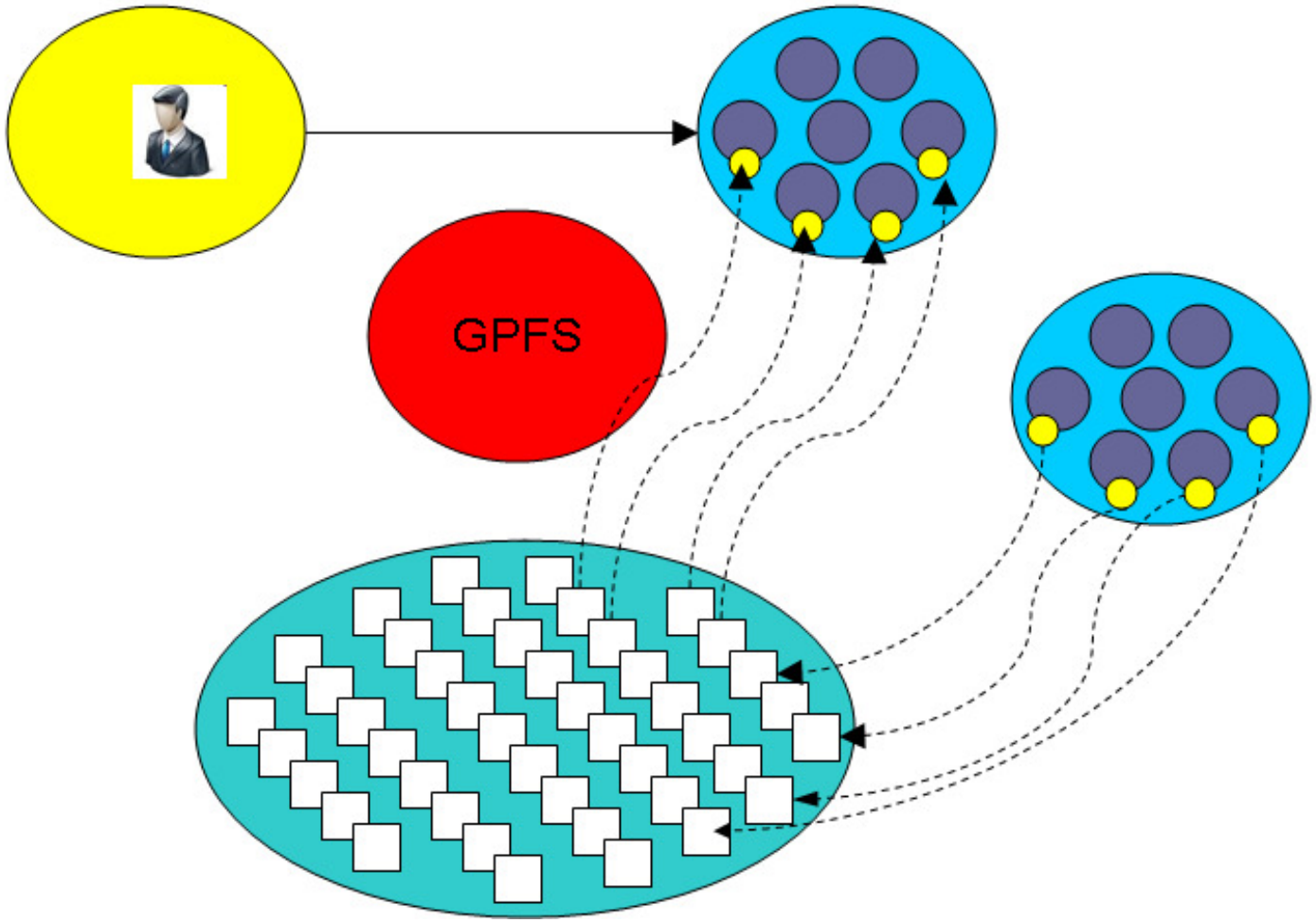
Shared file system

How can we push the limit?



Shared file system

How can we BREAK the limit?



Shared file system

- Optimal for large shared files.
- Behaves poorly under many small reads and writes.
- Your use of it affects everybody!
  (Different from case with CPU and RAM which are not shared.)
- How you read and write, your file format, the number of files in a directory, and how often you `ls`, can all affect every other user!
- The file system is shared over the ethernet network on GPC:
  Hammering the file system can hurt process communications.
- File systems are not infinite!
  Bandwidth, metadata, IOPS, number of files, space, . . .

SciNet

Shared file system

- Think of your laptop/desktop with several people simultaneously doing I/O, doing `ls` on directories with thousands of files . . .
- 2 jobs doing simultaneous I/O can take *much* longer than twice a single job duration due to disk *contention* and directory *locking*.
- SciNet: 500 users doing I/O from 4000 nodes.
  That's a lot of sharing and contention!

SciNet

Some Numbers

- 466 TB on scratch
- Over 500 users - you do the math!
- Want $>$25% free at any given time
  (systems can write 0.5 PB per day!)
- 100 MB/s: maximum possible read/write speed from a node if there is nothing else running on system

When system is fully utilized:

- 1 MB/s: average expected read/write speed from a node
- 10 IOP/s: average expected iops from a node
  So can't open more than 10 files in a second!

How to make the file system work for rather than against you

Make a Plan!

- Make a plan for your data needs:
  - How much will you generate,
  - How much do you need to save,
  - And where will you keep it?
- Note that /scratch is *temporary* storage for 3 months or less.
- Options?
  1. Save on your departmental/local server/workstation (it is possible to transfer TBs per day on a gigabit link);
  2. Apply for a project space allocation at next RAC call (but space is very limited);
  3. Buy tapes through us ($100/TB) and we can archive your data to tape; HSM possibility within next 6 months;
  4. Change storage format.

SciNet

Monitor and control usage

- Minimize use of filesystem commands like `ls` and `du`.
- Regularly check your disk usage using /scinet/gpc/bin/diskUsage.
- Warning signs which should prompt careful consideration:
  - More than 100,000 files in your space
  - Average file size less than 100 MB
- Monitor disk actions with `top` and `strace`

- RAM is always faster than disk; think about using ramdisk.
- Use `gzip` and `tar` to compress files to bundle many files into one
- Try gziping your *data* files. 30% not atypical!
- Delete files that are no longer needed
- Do "housekeeping" (`gzip, tar,` delete) *regularly*.

SciNet

Change storage format

- Write binary format files
  Faster I/O and less space than ASCII files.
- Use parallel I/O if writing from many nodes
  NetCDF, HDF5, MPI-IO
- Maximize size of files. Large block I/O optimal!
- Minimize number of files. Makes filesystem more responsive!
- Attend the parallel I/O course coming soon!
  https://support.scinet.utoronto.ca/courses

Don’ts:

- Don’t write lots of ASCII files. Lazy, slow, and wastes space!
- Don’t write many hundreds of files in a 1 directory.
  Hurts responsiveness!
- Don’t write many small files ($<$ 10MB).
  System is optimized for large-block I/O!

SciNet

Summary

- Make a data plan.
- Regularly check disk usage with /scinet/gpc/bin/diskUsage.
- RAM is always faster than disk: ramdisk.
- Write binary files.
- Use parallel I/O if writing from dozens of nodes.
- Use gzip and tar.
- Delete unneeded files.
- Maximize size of files.
- Do housekeeping regularly.
- Monitor disk actions with `top` and `strace`.
- Visit parallel I/O course coming soon!
- Make an appt to talk with our analysts about your I/O.

Don’ts

- Do not write lots of ASCII files.
- Do not write many hundreds of files in a single directory.
- Do not write many small files.
- Minimize use of file system commands like ls and du.

SciNet

Extras slides: examples

SciNet

Ramdisk example

northrup@aries:pts/11:~

File Edit View Terminal Tabs Help

```
#!/bin/bash
#MOAB/Torque submission script for SciNet GPC
#PBS -l nodes=1:ppn=8,walltime=24:00:00
#PBS -N ramdisk-test

cd $PBS_O_WORKDIR
mpirun -np 8 ./mycode

~
```

northrup@aries:pts/1:~

File Edit View Terminal Tabs Help

```
#!/bin/bash
#MOAB/Torque submission script for SciNet GPC
#PBS -l nodes=1:ppn=8,walltime=24:00:00
#PBS -N ramdisk-test

# stage-in
mkdir -p /dev/shm/$USER
cp -a $PBS_O_WORKDIR/ /dev/shm/$USER
cd /dev/shm/$USER

#run code
mpirun -np 8 ./mycode

# stage-out
tar -czf $PBS_O_WORKDIR/output.tar.gz /dev/shm/$USER
rm -Rf /dev/shm/$USER



:
```

Top example

northrup@gpc-logindm01/~

File Edit View Terminal Tabs Help

```
top - 10:31:47 up 25 days, 20:47, 11 users,  load average: 1.23, 1.54, 1.55
Tasks: 184 total,   1 running, 183 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.7%us,  0.9%sy,  0.0%ni, 98.1%id,  0.0%wa,  0.0%hi,  0.2%si,  0.0%st
Mem:   8174984k total,  6708804k used,  1466180k free,   163188k buffers
Swap:  2096472k total,        0k used,  2096472k free,   732128k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 3470 root       0 -20 2453m 1.1g  21m S  8.3 14.0   2664:48 mmfsd
23606 nolta     18   0  4528 1312  508 D  6.0  0.0   0:02.30 zip
    1 root      15   0 10348  708  592 S  0.0  0.0   0:04.50 init
    2 root      RT  -5     0    0    0 S  0.0  0.0   0:03.60 migration/0
    3 root      34  19     0    0    0 S  0.0  0.0   0:00.30 ksoftirqd/0
    4 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    5 root      RT  -5     0    0    0 S  0.0  0.0   0:07.54 migration/1
    6 root      34  19     0    0    0 S  0.0  0.0   0:00.73 ksoftirqd/1
    7 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/1
    8 root      RT  -5     0    0    0 S  0.0  0.0   0:04.60 migration/2
    9 root      34  19     0    0    0 S  0.0  0.0   0:02.13 ksoftirqd/2
   10 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/2
   11 root      RT  -5     0    0    0 S  0.0  0.0   0:06.93 migration/3
   12 root      34  19     0    0    0 S  0.0  0.0   0:08.96 ksoftirqd/3
   13 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/3
   14 root      RT  -5     0    0    0 S  0.0  0.0   0:04.00 migration/4
   15 root      34  19     0    0    0 S  0.0  0.0   0:00.28 ksoftirqd/4
   16 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/4
   17 root      RT  -5     0    0    0 S  0.0  0.0   0:06.26 migration/5
   18 root      34  19     0    0    0 S  0.0  0.0   0:09.94 ksoftirqd/5
   19 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/5
   20 root      RT  -5     0    0    0 S  0.0  0.0   0:04.22 migration/6
   21 root      34  19     0    0    0 S  0.0  0.0   0:00.42 ksoftirqd/6
   22 root      RT  -5     0    0    0 S  0.0  0.0   0:00.00 watchdog/6
   23 root      RT  -5     0    0    0 S  0.0  0.0   0:07.08 migration/7
```

Tar/gzip example

```
northrup@gpc-f101n084//scratch/northrup/temp/osu_network
File  Edit  View  Terminal  Tabs  Help
[northrup@gpc-f101n084 /scratch/northrup/temp]$ tar -czvf file.tar.gz osu_network/
osu_network/
osu_network/osu.h
osu_network/osu_alltoall.c
osu_network/osu_bcast.c
osu_network/osu_bibw.c
osu_network/osu_bw.c
osu_network/osu_get_bw.c
osu_network/osu_mbw_mr.c
osu_network/osu_multi_lat.c
osu_network/osu_put_bibw.c
osu_network/osu_put_bw.c
[northrup@gpc-f101n084 /scratch/northrup/temp]$ ls
file.tar.gz  osu_network
[northrup@gpc-f101n084 /scratch/northrup/temp]$ tar -tf file.tar.gz
osu_network/
osu_network/osu.h
osu_network/osu_alltoall.c
osu_network/osu_bcast.c
osu_network/osu_bibw.c
osu_network/osu_bw.c
osu_network/osu_get_bw.c
osu_network/osu_mbw_mr.c
osu_network/osu_multi_lat.c
osu_network/osu_put_bibw.c
osu_network/osu_put_bw.c
[northrup@gpc-f101n084 /scratch/northrup/temp]$ tar -xzf file.tar.gz  osu_network/osu_get_bw.c
[northrup@gpc-f101n084 /scratch/northrup/temp]$ ls
file.tar.gz  osu_network
```

I/O speed for ASCII

Writing 128M doubles:

/scratch:

ASCII	173 s
binary	6 s

/dev/shm:

ASCII	174 s
binary	1s (!)

typical work station:

ASCII	260 s
binary	20s

SciNet

File system at a glance