Typical Administrative Tasks

Ely de Oliveira
ely.oliveira@thinkparq.com

Agenda

- **Installation, upgrade, versioning**
- **Command-line administration tools**
- **Advanced configurations and features**
  - Multiple BeeGFS instances
  - Pin BeeGFS processes to specific CPUs
  - Data Striping
- **Getting statistics**
  - Server
  - Client
  - User
- **Built-in Benchmarks**

Installation

- **Detailed installation guide**
  http://www.beegfs.com/wiki/ManualInstallation
- **Step-by-step reference (Example)**
  http://www.beegfs.com/wiki/ManualInstallWalkThrough
- **Package repositories**

```
# wget -O /etc/yum.repos.d/beegfs-rhel7.repo
      http://www.beegfs.com/release/latest-stable/dists/beegfs-rhel7.repo
```

- **Build instructions for other platforms, e.g.:**

Installation

- **Management service**
  - No special requirements
  - Non performance-critical
  - Typically runs on a cluster master node or on a storage node

```
server01:~ # yum install beegfs-mgmtd
server01:~ # beegfs-setup-mgmtd -p /data/beegfs/beegfs_mgmtd
server01:~ # systemctl start beegfs-mgmtd
```

server01

Installation

Metadata service

- Storage device: SSDs widely used, RAID1 for low latency
- Capacity: 0.5% of total storage space
- Local file system: ext4 (fast handling of small files)
- Processor clock frequency more important than number of CPU cores for low latency
- Number of CPU cores important for systems with lots of clients
- Low latency network important for metadata access
- RAM used for caching – The more the better

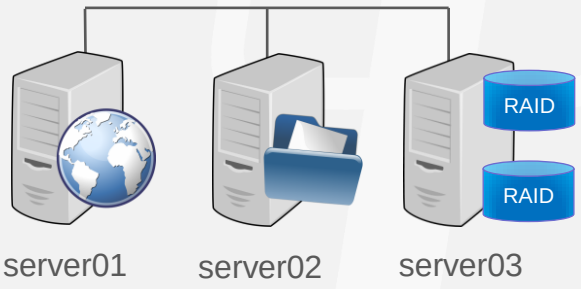
```
server02:~ # yum install beegfs-meta

server02:~ # beegfs-setup-meta -p /mnt/ssd1
             -s 1 -m server01

server02:~ # systemctl start beegfs-meta
```

Installation

Storage service

- Multiple storage targets: RAID6 volumes, 12 disks per volume
- Local file systems: usually XFS, but also others like ZFS and ext4
- Mid-range CPUs are sufficient
- Low latency high throughput network recommended
- RAM used for caching – The more the better
- RAM needed per client: **RDMABufferSize x RDMABufferNum x 2 x MaxInternodeNum = ~12 MB**

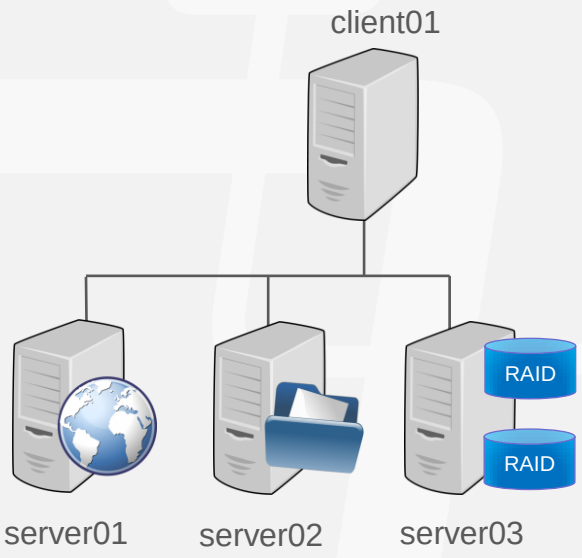
```
server03:~ # yum install beegfs-storage

server03:~ # beegfs-setup-storage -p /mnt/raid1
             -s 3 -i 301 -m server01

server03:~ # beegfs-setup-storage -p /mnt/raid2
             -s 3 -i 302 -m server01

server03:~ # systemctl start beegfs-storage
```



Installation

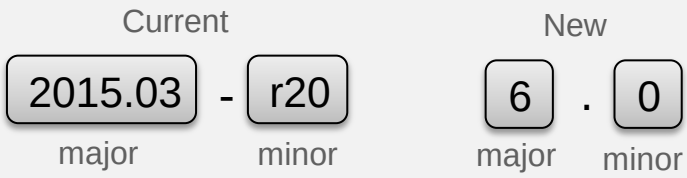
- **Client service**
  - The only kernel module
  - Automatically build for newly installed kernels
  - Low latency high throughput network recommended

```
client01:~ # yum install beegfs-client
client01:~ # beegfs-setup-client -m server01
client01:~ # systemctl start beegfs-client
```



Updating, upgrading, and versioning

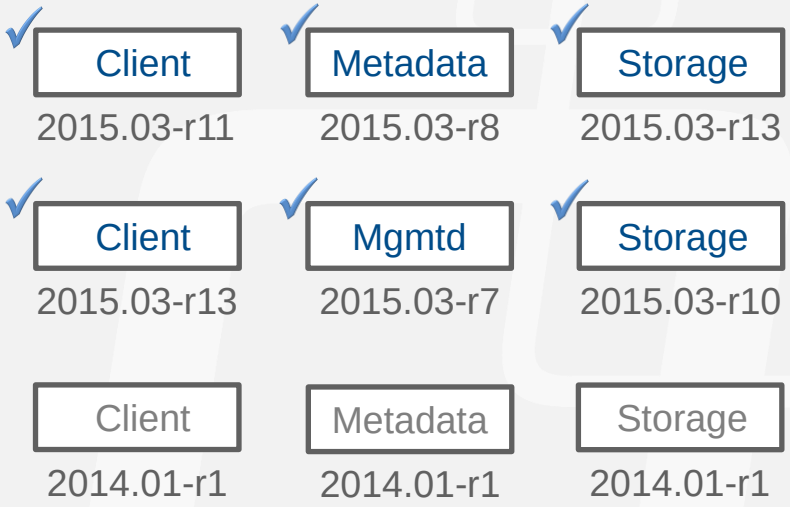
Version scheme




```
# yum update beegfs-mgmtd
# systemctl restart beegfs-mgmtd
```

Minor release update

- Minor releases can be mixed
- Downtime unnecessary
- Service restart required

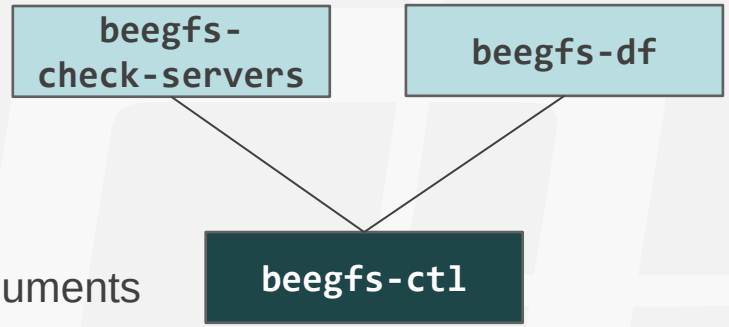


Major release upgrade

- Major releases cannot be mixed
- Downtime needed

Command-line administration

- **Configuration files at /etc/beegfs/*.conf**
  - Reasonable default configuration values
  - All options documented
- **Log files at /var/log/beegfs-*.log**
  - Human-readable messages
- **Command `beegfs-ctl`**
  - General command line administration tool
  - Part of the `beegfs-utils` package
  - Some helpers just call `beegfs-ctl` with special arguments for convenience, e.g. `beegfs-df`, `beegfs-check-servers`



beegfs-ctl has a built-in help with examples

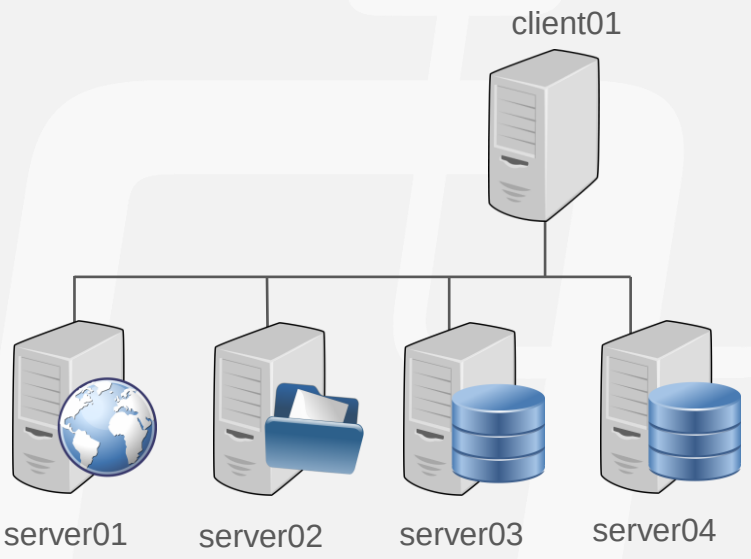
```
client01:~ # beegfs-ctl --help
BeeGFS Command-Line Control Tool (http://www.beegfs.com)

GENERAL USAGE:
  $ beegfs-ctl --<modename> --help
  $ beegfs-ctl --<modename> [mode_arguments] [client_arguments]

MODES:
  --listnodes           => List registered clients and servers.
  --listtargets         => List metadata and storage targets.
  --removenode (*)      => Remove (unregister) a node.
  --removetarget (*)    => Remove (unregister) a storage target.
  --getentryinfo        => Show file system entry details.
  --find                => Find files located on certain servers.
  --migrate             => Migrate files to other storage servers.
  --serverstats         => Show server IO statistics.
  --clientstats         => Show client IO statistics.
  --userstats           => Show user IO statistics.
  --storagebench (*)    => Run a storage targets benchmark.
  --getquota            => Show quota information for users or groups.
  --setquota (*)        => Sets the quota limits for users or groups.
  --listmirrorgroups    => List mirror buddy groups.
  --addmirrorgroup (*) => Add a mirror buddy group.
  ...
```

Each beegfs-ctl mode has its own help with examples

```
client01:~ # beegfs-ctl --listnodes --help
GENERAL USAGE:
  $ beegfs-ctl --<modename> [mode_arguments] [client_arguments]

CLIENT ARGUMENTS:
 Optional:
  --mount=<path>            Use config settings from this BeeGFS mountpoint
                            (instead of "--cfgFile").
  --cfgFile=<path>          Path to BeeGFS client config file.
                            (Default: /etc/beegfs/beegfs-client.conf)
  --logEnabled              Enable detailed logging.
  --<key>=<value>           Any setting from the client config file to override
                            the config file values (e.g. "--logLevel=5").

MODE ARGUMENTS:
 Mandatory:
  --nodetype=<nodetype>  The node type (management, metadata, storage, client).
 Optional:
  --details              Print additional node details, such as network ports
                         and interface order.
  --nicdetails           Print additional network interconnect details, such as
                         IP address of each node interface.
  --showversion          Print node version code.
...
```

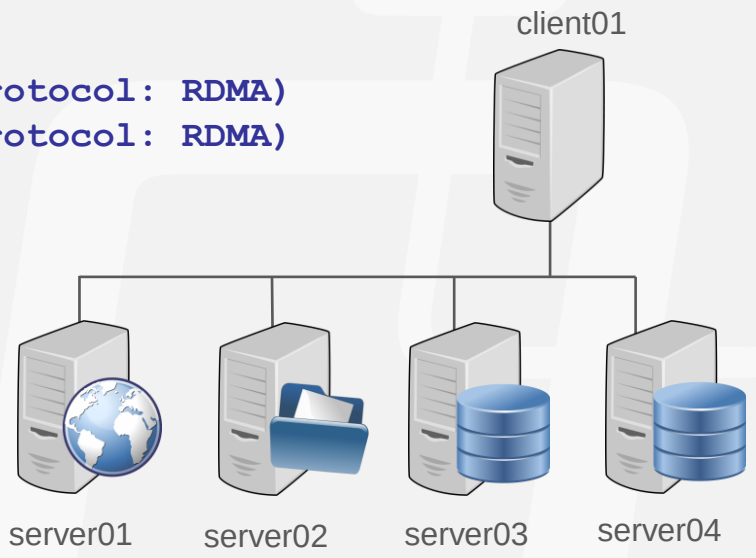
Listing registered nodes and their interfaces

```
client01:~ # beegfs-ctl --listnodes --nodetype=storage

server03 [ID: 3]
server04 [ID: 4]


client01:~ # beegfs-ctl --listnodes --nodetype=storage --nicdetails

server03 [ID: 3]
   Ports: UDP: 8003; TCP: 8003
   Interfaces:
   + ib0[ip addr: 10.12.20.3; type: RDMA]
   + ib0[ip addr: 10.12.20.3; type: TCP]
   + eth0[ip addr: 10.10.20.3; type: TCP]

server04 [ID: 4]
   Ports: UDP: 8003; TCP: 8003
   Interfaces:
   + ib0[ip addr: 10.12.20.4; type: RDMA]
   + ib0[ip addr: 10.12.20.4; type: TCP]
   + eth1[ip addr: 10.10.20.4; type: TCP]
```



Check responsiveness & reachability of servers

```
client01:~ # beegfs-check-servers

Management
==========
server01 [ID: 1]: reachable at 10.12.20.3:8008 (protocol: TCP)

Metadata
==========
server02 [ID: 2]: reachable at 10.12.20.4:8005 (protocol: RDMA)

Storage
==========
server03 [ID: 3]: reachable at 10.12.20.3:8003 (protocol: RDMA)
server04 [ID: 4]: reachable at 10.12.20.4:8003 (protocol: RDMA)
```

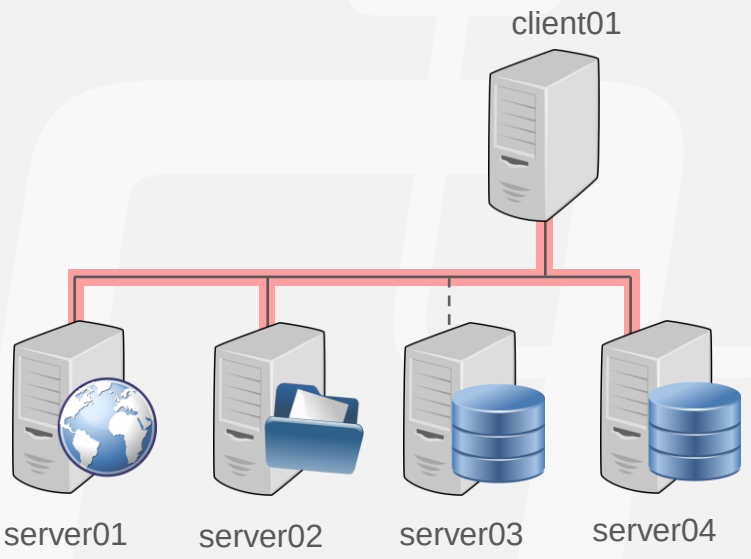


Check responsiveness & reachability of servers

```
client01:~ # beegfs-net

mgmt_nodes
=============
server01 [ID: 1]
   Connections: TCP: 1 (10.12.20.1:8008);

storage_nodes
=============
server03 [ID: 3]
   Connections: <none>
server04 [ID: 4]
   Connections: RDMA: 3 (10.12.20.4:8003);

meta_nodes
=============
server02 [ID: 2]
   Connections: RDMA: 1 (10.12.20.2:8005);
```

TCP is normal for management service (management service is not performance-critical)

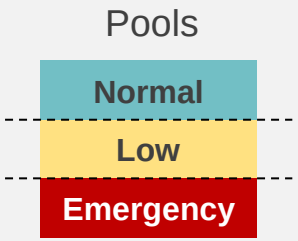
Connections “none” is normal, because connections are only established when needed.

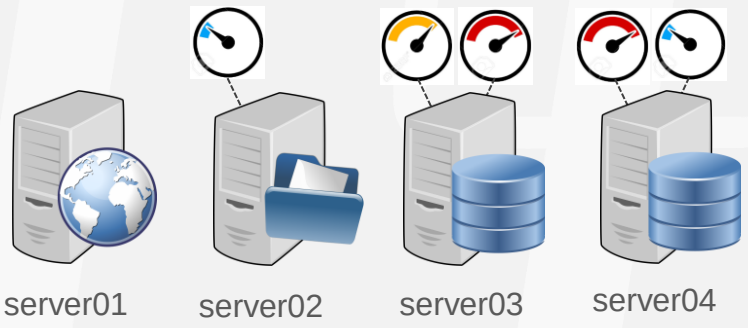


Check free space

```
client01:~ # beegfs-df

METADATA SERVERS:
TargetID        Pool        Total         Free      InodesTotal      InodesFree
========        ====        =====         ====      ===========      ==========
       2    [normal]     240.0GB      230.0GB           158.8M           148.7M

STORAGE TARGETS:
TargetID        Pool        Total         Free      InodesTotal      InodesFree
========  ===========       =====         ====      ===========      ==========
     301  [emergency]    9168.7GB       4.5GB           582.2M           87.2M
     302       [low]     9168.7GB     102.5GB           582.2M           42.0M
     401  [emergency]    9168.7GB       2.5GB           582.2M           75.2M
     402     [normal]    9168.7GB    2112.5GB           582.2M           92.2M
```

tuneStorageSpaceLowLimit = 200G
tuneStorageSpaceEmergencyLimit = 5G





File system consistency check and repair

```
client01:~ # beegfs-fsck --checkfs [--runonline]
--------------------------------------------------------------------
Started BeeGFS fsck in forward check mode [Mon Sep 29 21:47:57 2015]
Log will be written to /var/log/beegfs-fsck.log
Database will be saved as /var/lib/beegfs/beegfs-fsck.db
--------------------------------------------------------------------

Step 1: Check reachability of nodes: Finished
Step 2: Gather data from nodes:

Fetched data > Directory entries: 504598 | Inodes: 504598 | Chunks: 904831

Step 3: Check for errors...

* Target is used, but does not exist... Finished
* File has a missing target in stripe pattern... Finished
* Dentry-by-ID file is present, but no corresponding dentry... Finished
* Dentry-by-ID file is broken or missing... Finished
* Chunk is saved in wrong path... Finished
* Wrong owner node saved in inode... Finished
* Dentry points to inode on wrong node... Finished
...
```

File system consistency check and repair

- **Run while file system is in use**

```
client01:~ # beegfs-fsck --checkfs [--runonline]
```

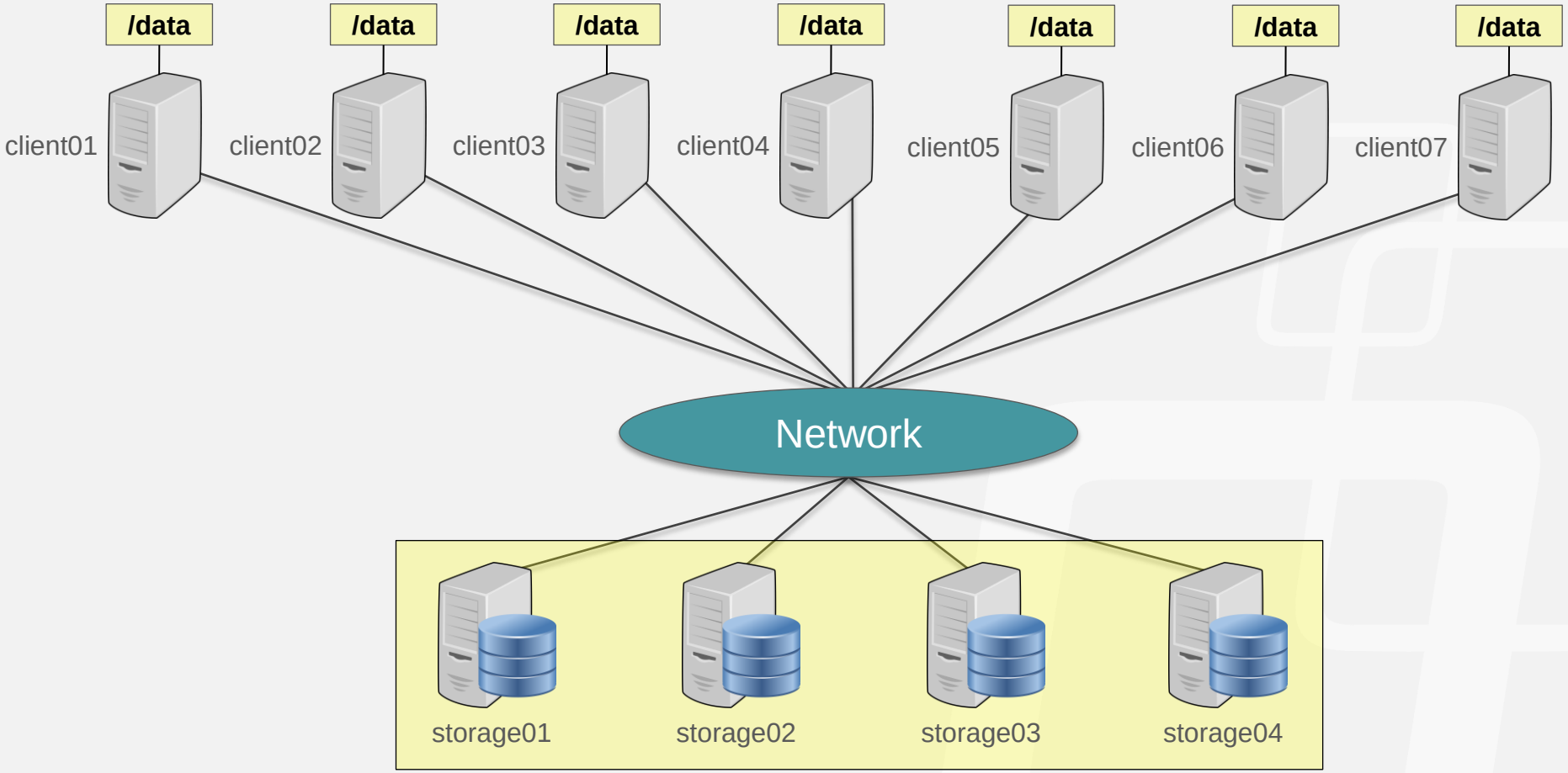
- **Skip repairs**

```
client01:~ # beegfs-fsck --checkfs --readonly
```

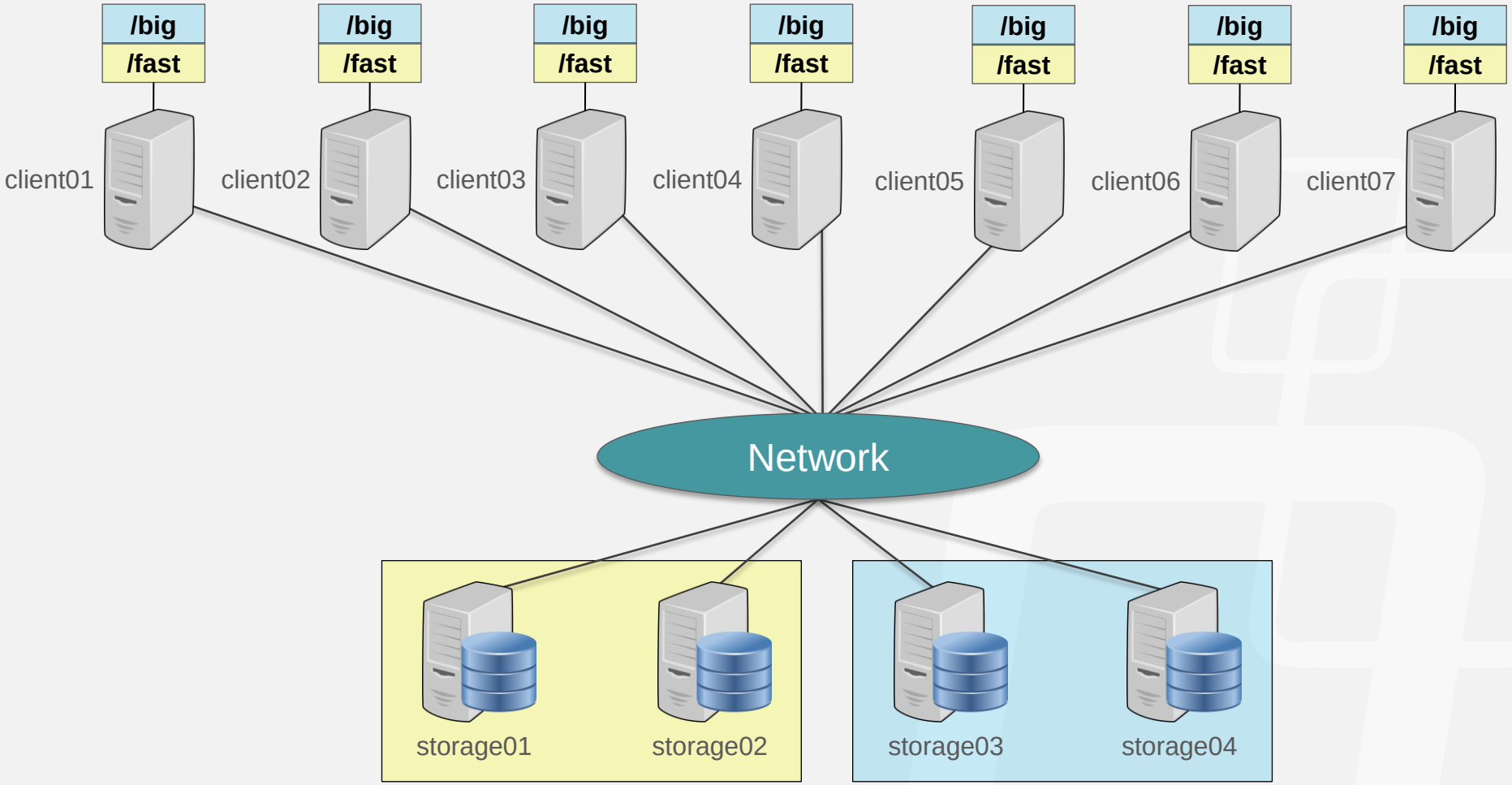
- **Repair errors automatically**

```
client01:~ # beegfs-fsck --checkfs --automatic
```

Mount multiple BeeGFS instances



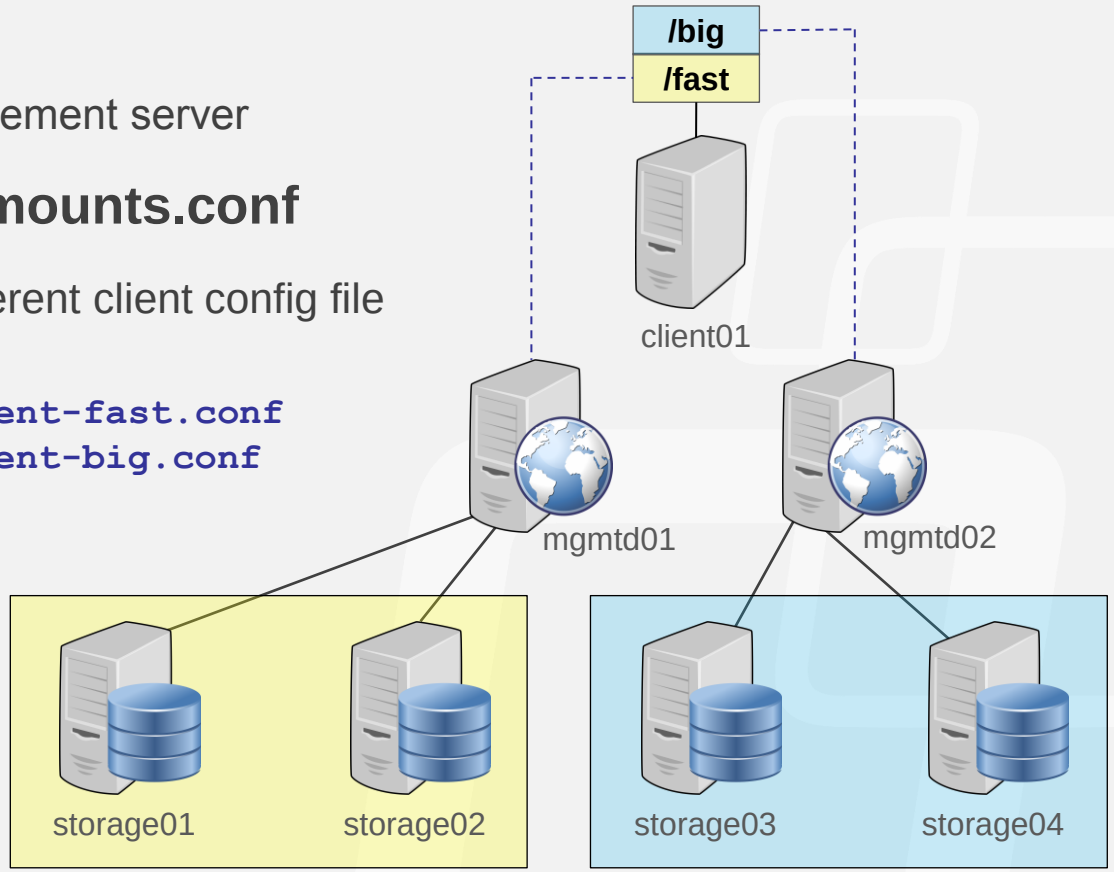
Mount multiple BeeGFS instances



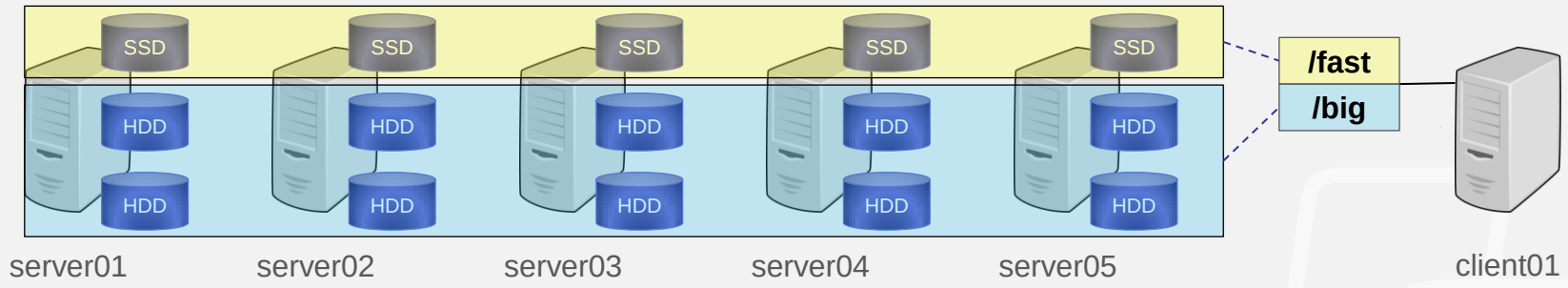
Mount multiple BeeGFS instances

- **Create copies of /etc/beegfs/beegfs-client.conf…**
  - … with new names
  - … pointing to a second management server
- **Edit /etc/beegfs/beegfs-mounts.conf**
  - Map each mountpoint to a different client config file

```
/fast /etc/beegfs/beegfs-client-fast.conf
/big  /etc/beegfs/beegfs-client-big.conf
```



Export multiple BeeGFS instances from the same servers



- **Enable „multi mode“ in /etc/default/beegfs- …**

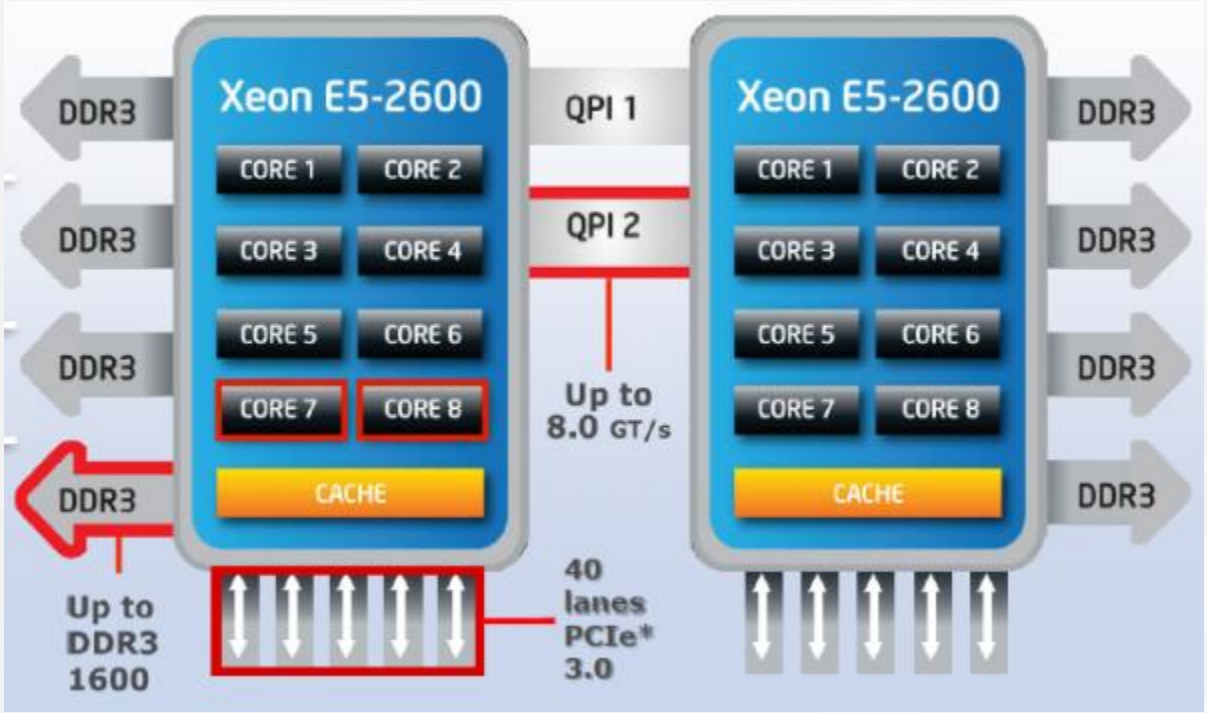
```
server01:~ # cat /etc/default/beegfs-storage
MULTI_MODE="YES"
```

More convenient in future versions

- **Create a separate directory in /etc/beegfs for each instance with config files**
  - /etc/beegfs/fast.d/
  - /etc/beegfs/big.d/

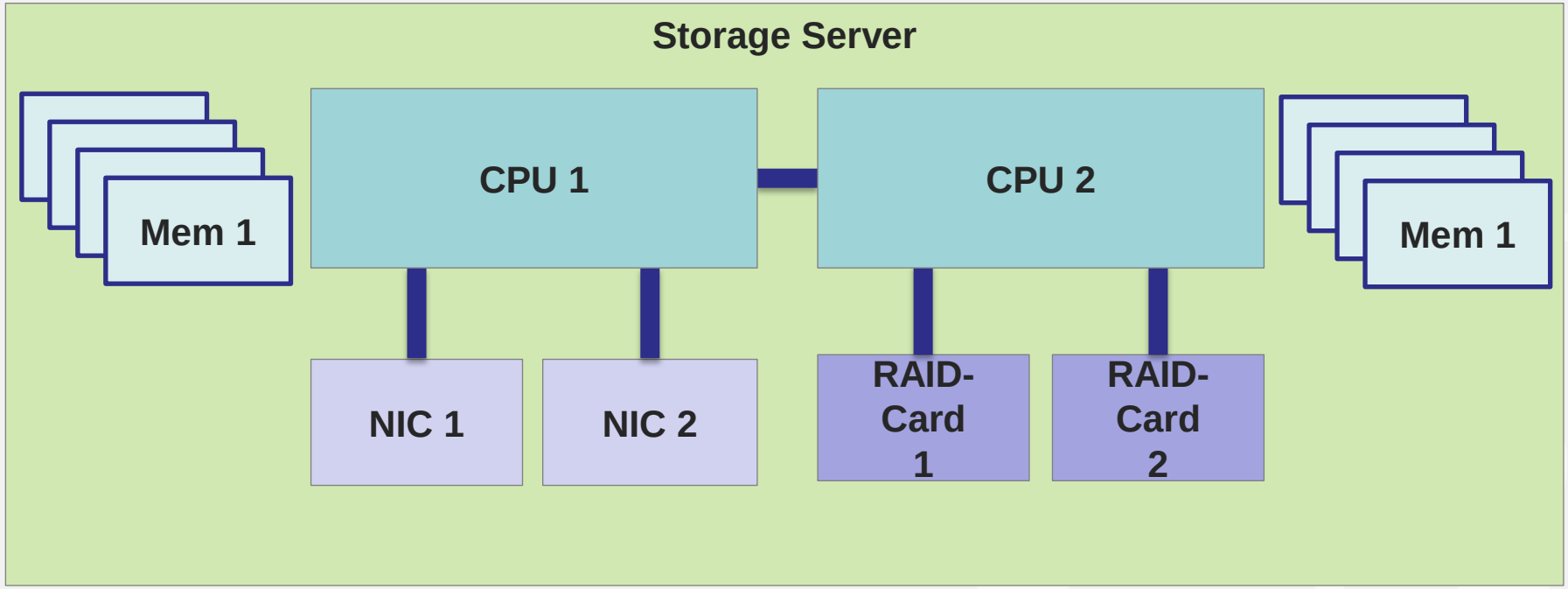
Pin BeeGFS processes

Important for Omni-Path and Mellanox EDR



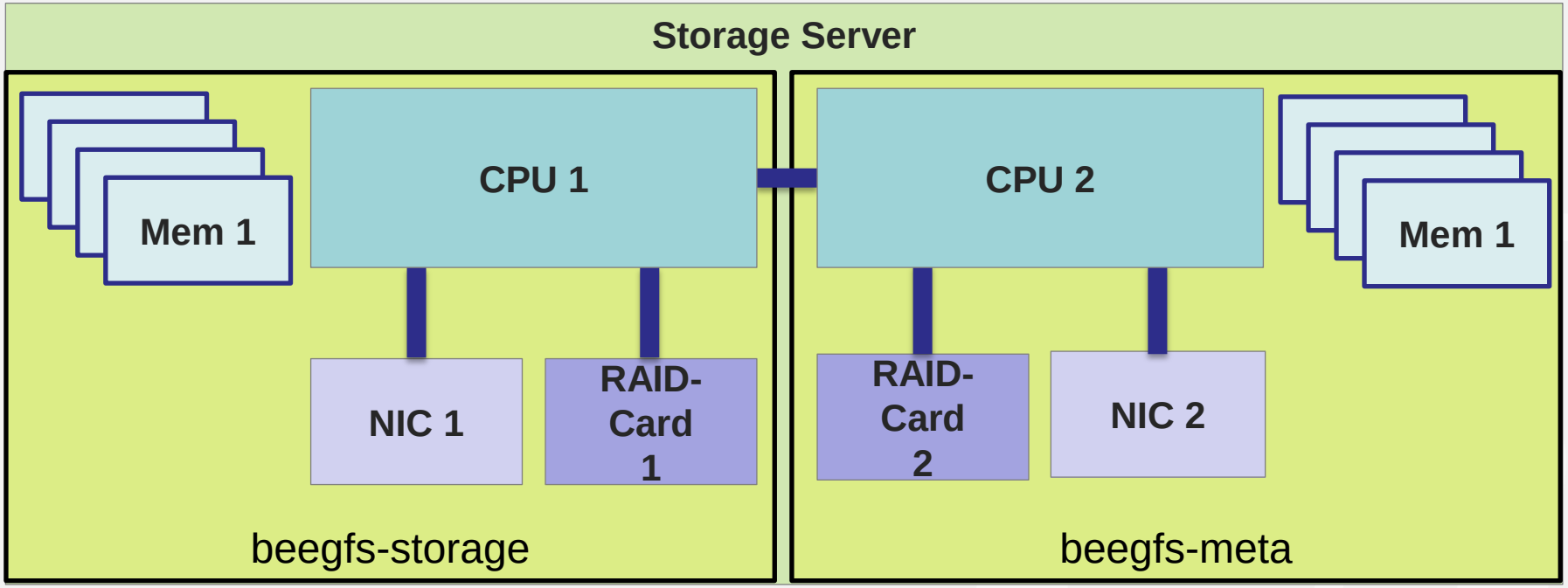
- **Use option `tuneBindToNumaZone` in files beegfs-{meta,storage}.conf**
  - ls –al /sys/devices/system/node/node{0,1}
  - lstopo command shows which devices are attached to which zone

Pin BeeGFS processes



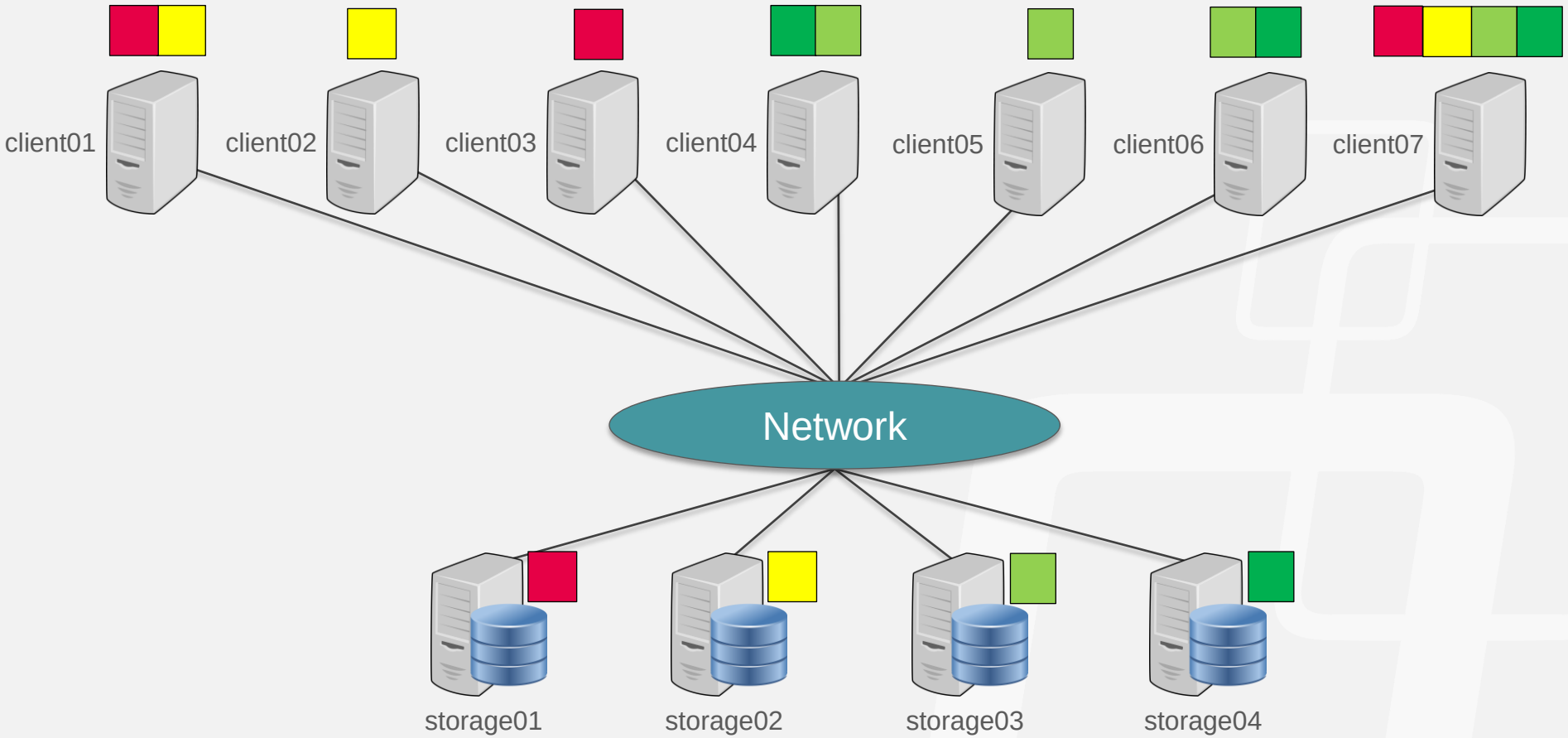
- This is a bad configuration, because data path always involves both sockets (NIC <-> RAID-Card)

Pin BeeGFS processes

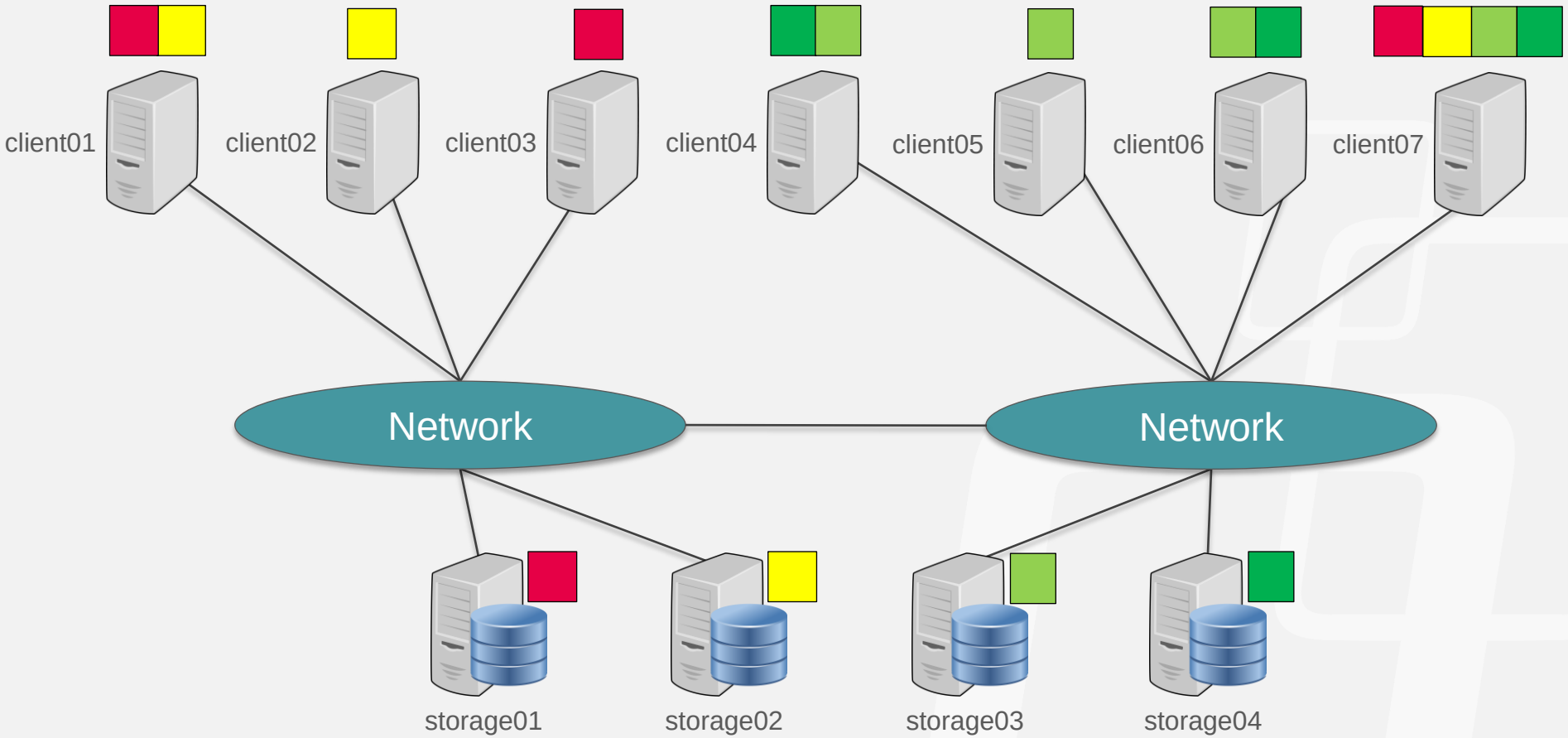


- Each BeeGFS service assigned to a different CPU
- Pair NIC and RAID-card assigned to each CPU

Assign storage servers to clients



Assign storage servers to clients



Assign storage servers to clients

- **Set options in file beegfs-client.conf**
  - `tunePreferredMetaFile`: IDs of preferred metadata nodes
  - `tunePreferredStorageFile`: IDs of preferred storage nodes

```
client01:~ # beegfs-ctl --listtargets --longnodes
TargetID   NodeID
========   ======
     101   storage01 [ID: 1]
     102   storage01 [ID: 1]
     201   storage02 [ID: 2]
     202   storage02 [ID: 2]

client01:~ # beegfs-ctl --listnodes --nodetype=meta
meta01 [ID: 1]
meta02 [ID: 2]
```

Limit traffic to specific network interfaces

- **Use options in files /etc/beegfs/beegfs-*.conf**
  - `connInterfacesFile`: Registered interfaces for incoming connections
  - `connNetFilterFile`: Limit outgoing connections to certain IP address ranges

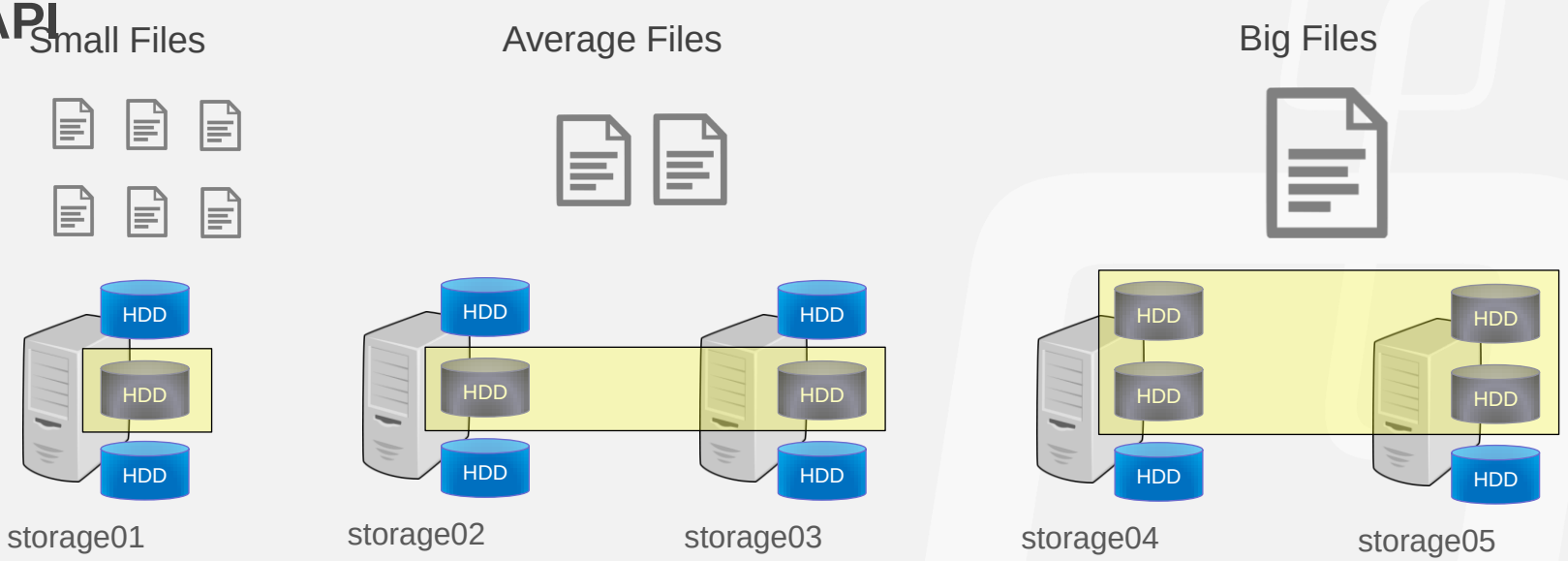
```
client01:~ # beegfs-ctl --listnodes --nodetype=client --nicdetails

14E8-53E4DE10-demo-io1
   Ports: UDP: 18004; TCP: 0
   Interfaces:
   + eth0[ip addr: 10.10.201.203; type: TCP]

FC13-53EB9471-demo-io2
   Ports: UDP: 18004; TCP: 0
   Interfaces:
   + ib0[ip addr: 10.12.201.204; type: RDMA]
   + ib0[ip addr: 10.12.201.204; type: TCP]
   + eth1[ip addr: 10.10.201.204; type: TCP]

Number of nodes: 2
```

Data Striping

- **Setting Data Striping Pattern per Path**

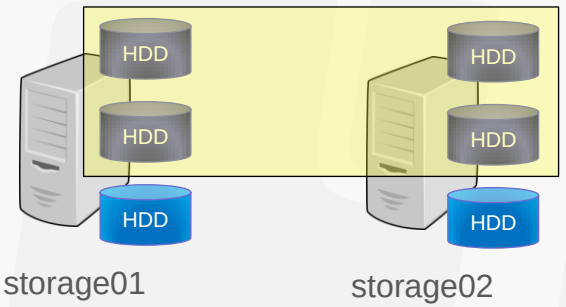
```
client01:~ # beegfs-ctl --setpattern --chunksize=1m --numtargets=4 /data/test
```

- **Application can set pattern via API**



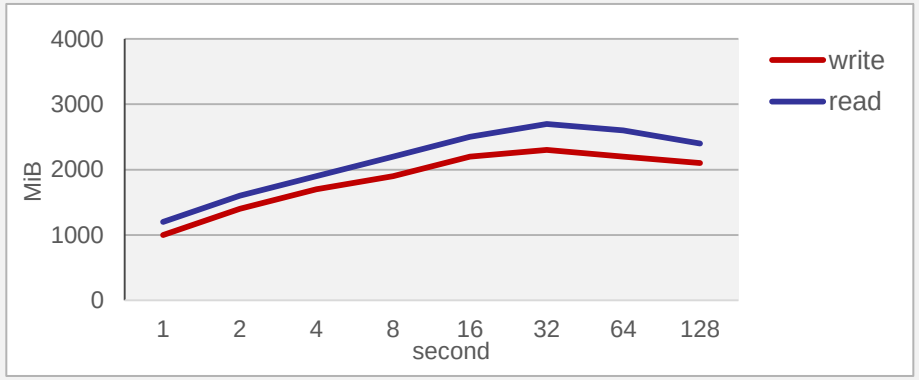
Query properties of files or directories

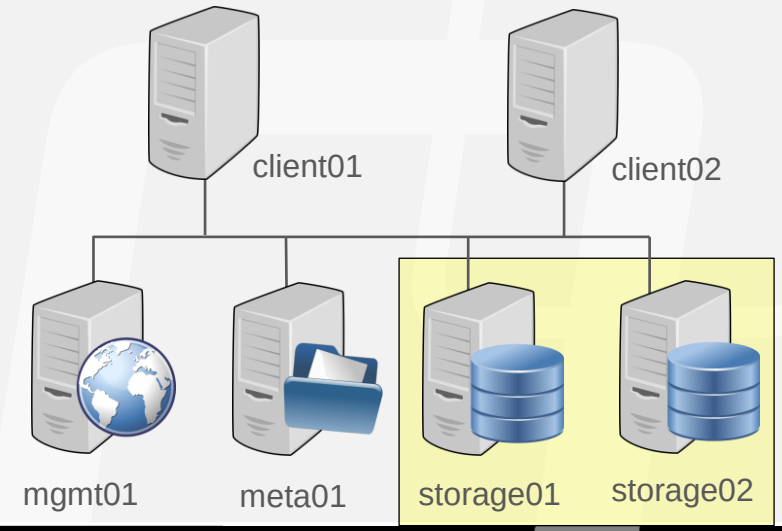
```
client01:~ # beegfs-ctl --getentryinfo /mnt/beegfs/testdir/test.txt

Path: /testdir/test
Mount: /mnt/beegfs
EntryID: 0-5429B29A-AFA6
Metadata node: meta01 [ID: 1]

Stripe pattern details:
+ Type: RAID0
+ Chunksize: 512K
+ Number of storage targets: desired: 4; actual: 4
+ Storage targets:
  + 102 @ storage01 [ID: 1]
  + 101 @ storage01 [ID: 1]
  + 201 @ storage02 [ID: 2]
  + 202 @ storage02 [ID: 2]
```



Getting Server Statistics

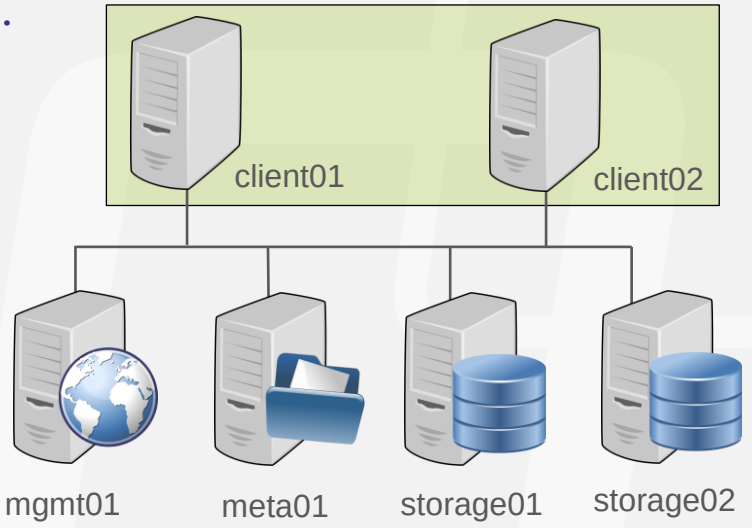
```
client01:~ # beegfs-ctl --serverstats --perserver --interval=1 --nodetype=storage

=== time index: 1412011239 (values show last second only)

nodeID write_KiB  read_KiB   reqs   qlen bsy
     1   1436430      5633  18098      0   0
     2   1536430     14430  18386      0   0

=== time index: 1412011240 (values show last second only)

nodeID write_KiB  read_KiB   reqs   qlen bsy
     1   1404430      9633  15613      0   0
     2   1336330     12403  17274      0   0
...
```





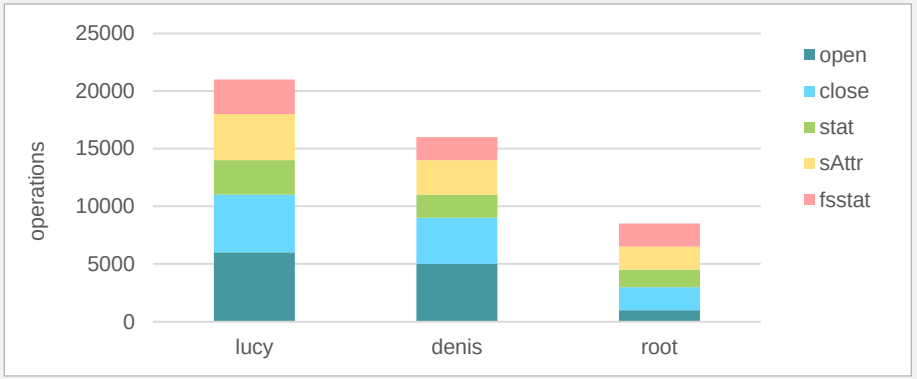
Getting Client Statistics

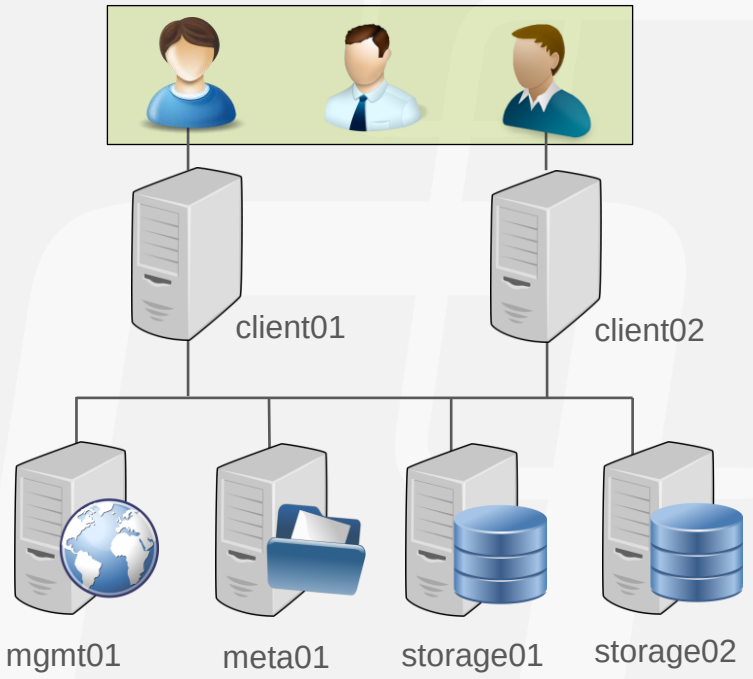
```
client01:~ # beegfs-ctl --clientstats --names --interval=5 nodetype=storage
====== 5 s ======
Sum       22967 [sum] 4720 [close]  65 [sAttr] 5706 [open] 2456 [stat] ...
client01   2847 [sum]  700 [close] 700 [open]  1447 [stat] ...
client02   2847 [sum]  700 [close] 700 [open]  1447 [stat] ...
meta01     1060 [sum] 1060 [close] ...
...
====== 10 s ======
Sum:      28138 [sum] 6511 [close] 1800 [open]  1804 [stat] 4 [sAttr] ...
client01   5803 [sum] 1403 [close] 1400 [open] ...
client02   5803 [sum] 1403 [close] 1400 [open] ...
meta01     2454 [sum] 2454 [close] ...
...
```



Getting User Statistics

```
client01:~ # beegfs-ctl --userstats --names --interval=5 --nodetype=meta

====== 5 s ======
Sum:          60 [sum] 10 [close]  10 [open]   1 [sAttr]  6 [unlnk] ...
lucy          36 [sum]  3 [close]   3 [open]   4 [stat]   1 [unlnk] ...
root          12 [sum]  6 [stat]    4 [unlnk] ...
denis          8 [sum]  1 [sAttr]   3 [stat] ...

====== 10 s ======
Sum:        2314 [sum] 24 [close]   1 [create] ...
lucy        2029 [sum] 23 [close]  23 [open] ...
denis         16 [sum]  1 [close]   1 [create] ...
root           8 [sum]  4 [stat] ...
```

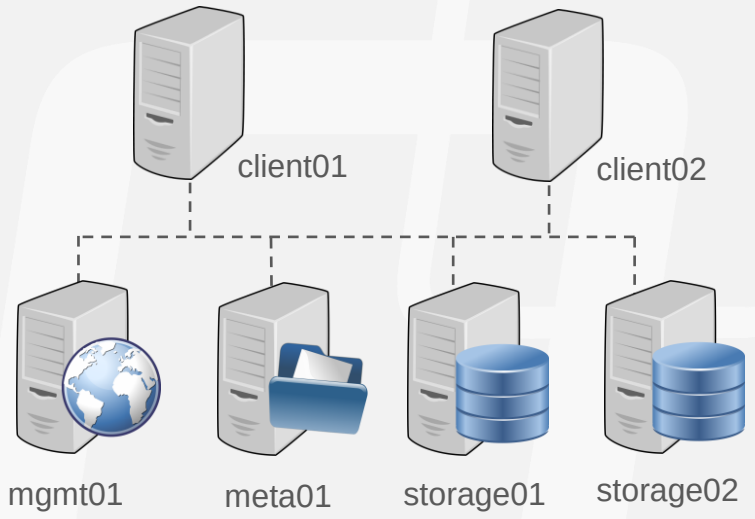




Built-in benchmark tool

- **StorageBench**

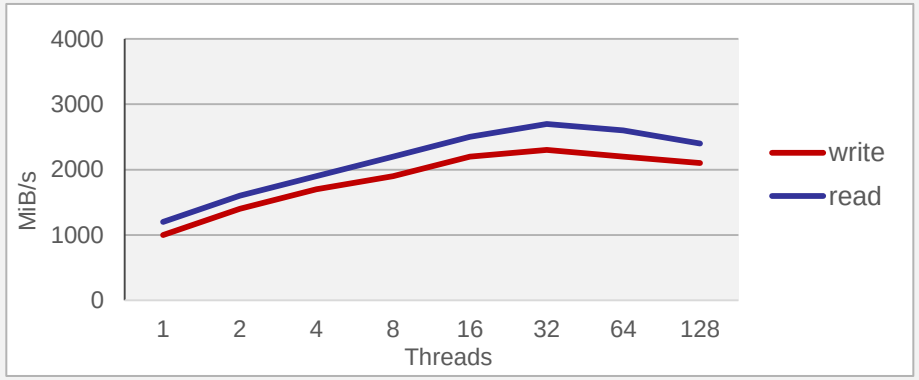
```
client01:~ # beegfs-ctl --storagebench --alltargets --write --blocksize=512K
                        --size=320G --threads=4

client01:~ # beegfs-ctl --storagebench --alltargets --read --blocksize=512K
                        --size=320G --threads=4
```

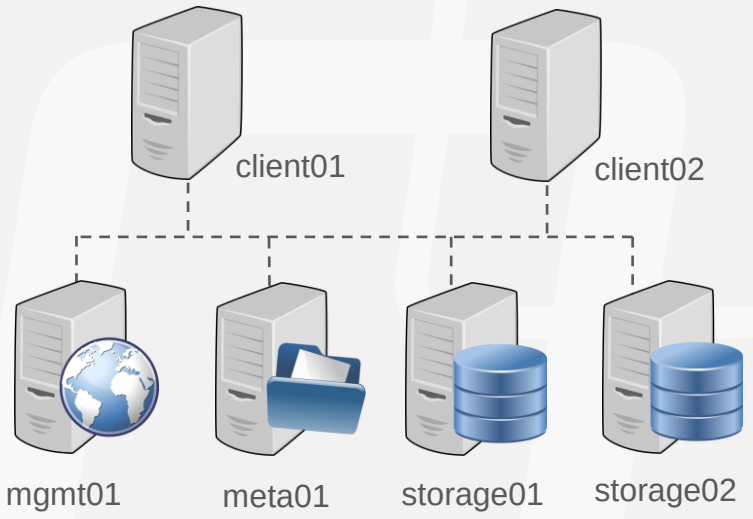


Built-in benchmark tool

StorageBench

```
client01:~ # beegfs-ctl --storagebench --alltargets --status

Server benchmark status:
Running:    4

Write benchmark results:
Min throughput:            1107  MiB/s  nodeID: storage01 [ID: 1], targetID: 101
Max throughput:            1462  MiB/s  nodeID: storage02 [ID: 2], targetID: 201
Avg throughput:            1283  MiB/s
Aggregate throughput:      2590  MiB/s
```

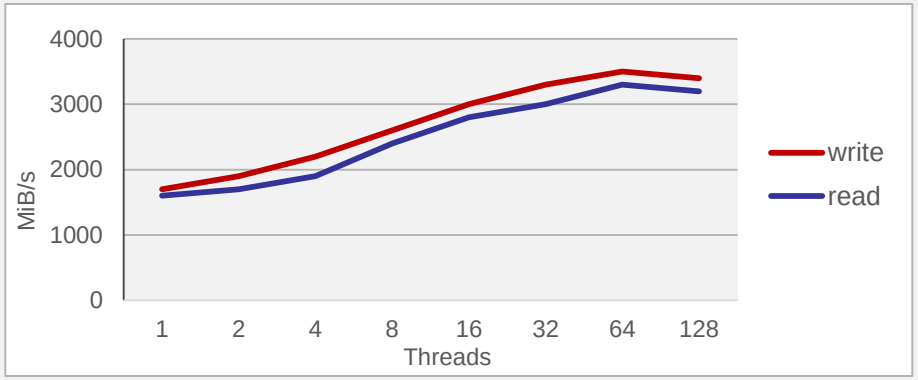


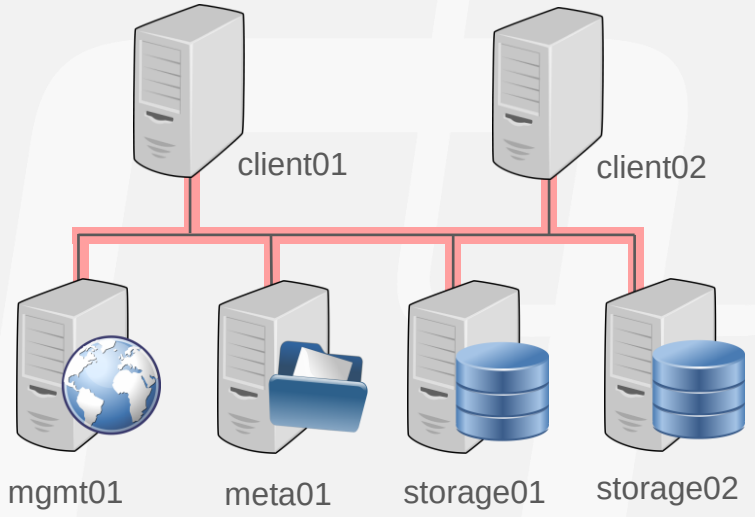


Built-in benchmark tool

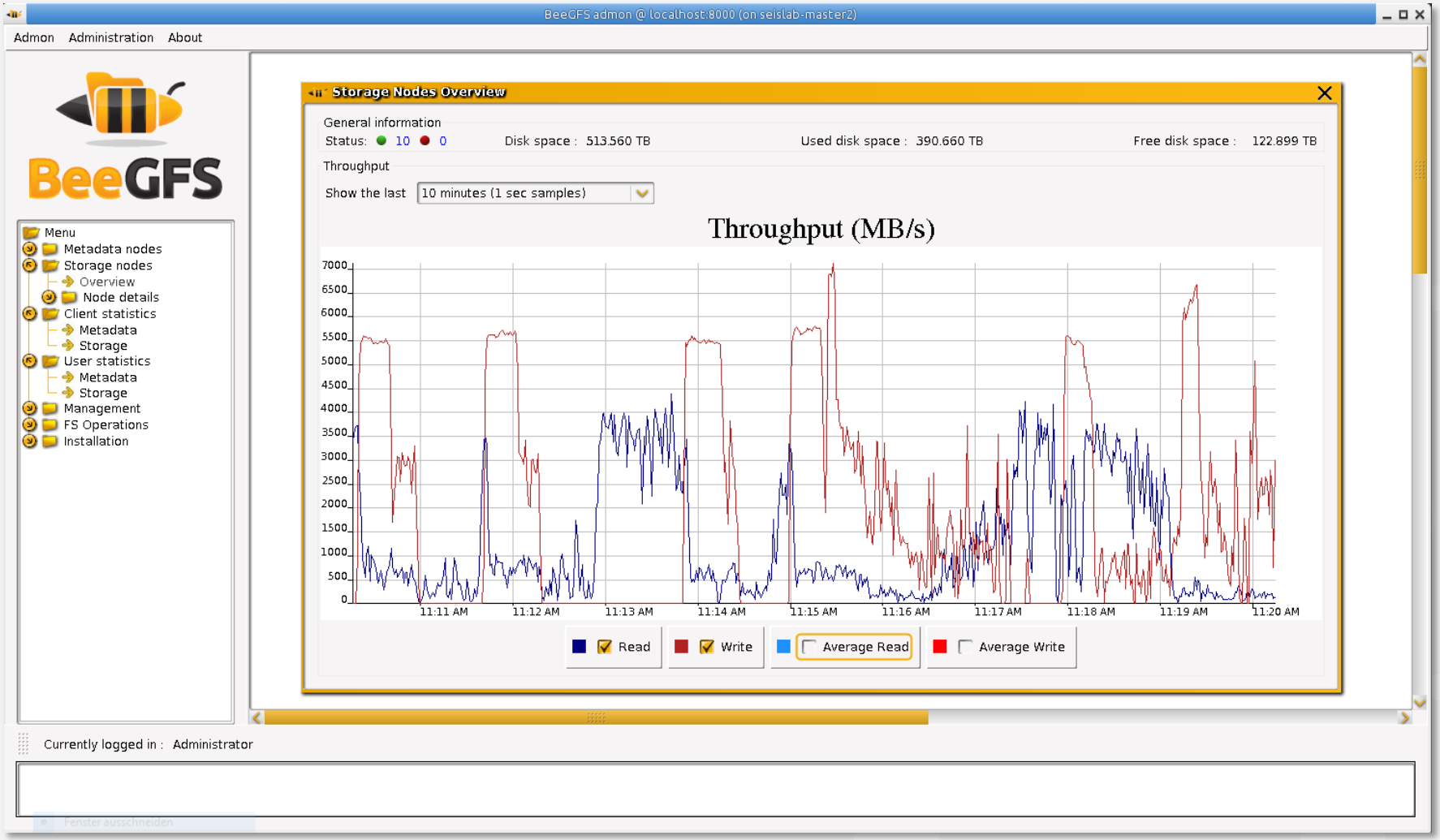
- **NetBench Mode**
  - Network streaming throughput
  - Roughly equivalent to writing to a RAM drive

```
client01:~ # $ echo 1 > /proc/fs/beegfs/<clientID>/netbench_modes
...
client01:~ # $ echo 0 > /proc/fs/beegfs/<clientID>/netbench_modes
```



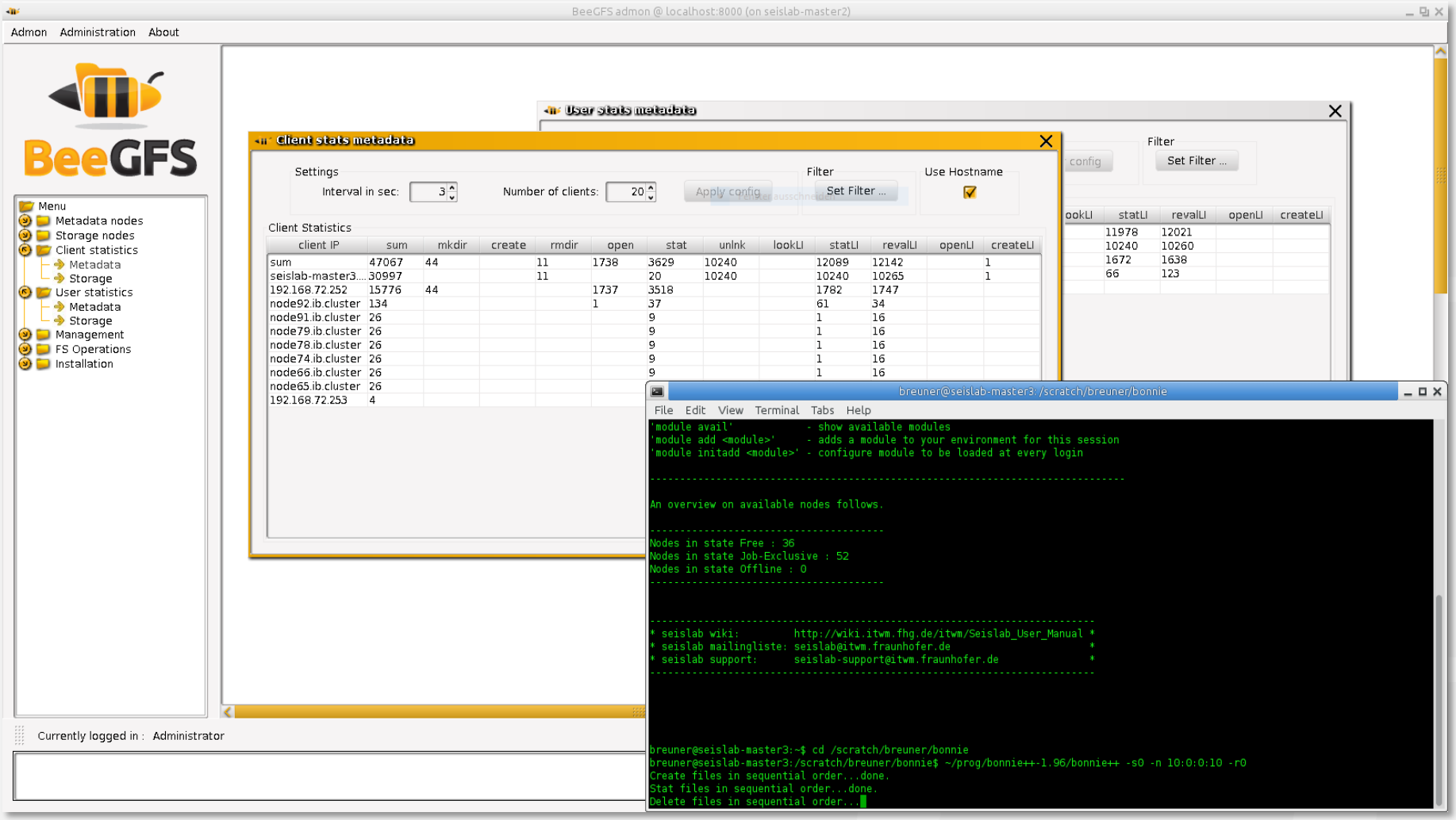


Live Throughput Overview

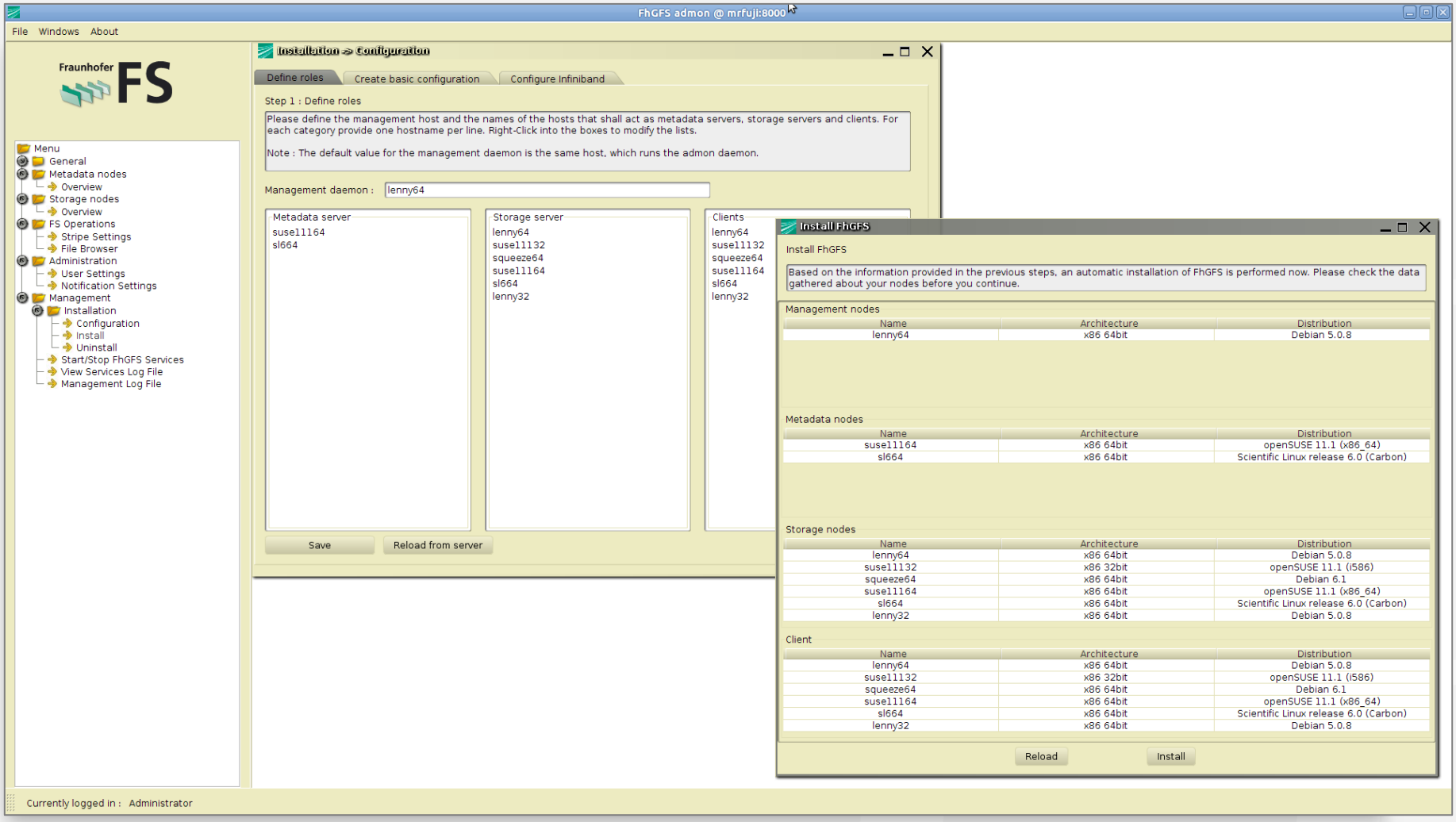


thinkparQ

Live per-Client and per-User Statistics



GUI for Windows-style Installation



Questions?