GPFS-FPO

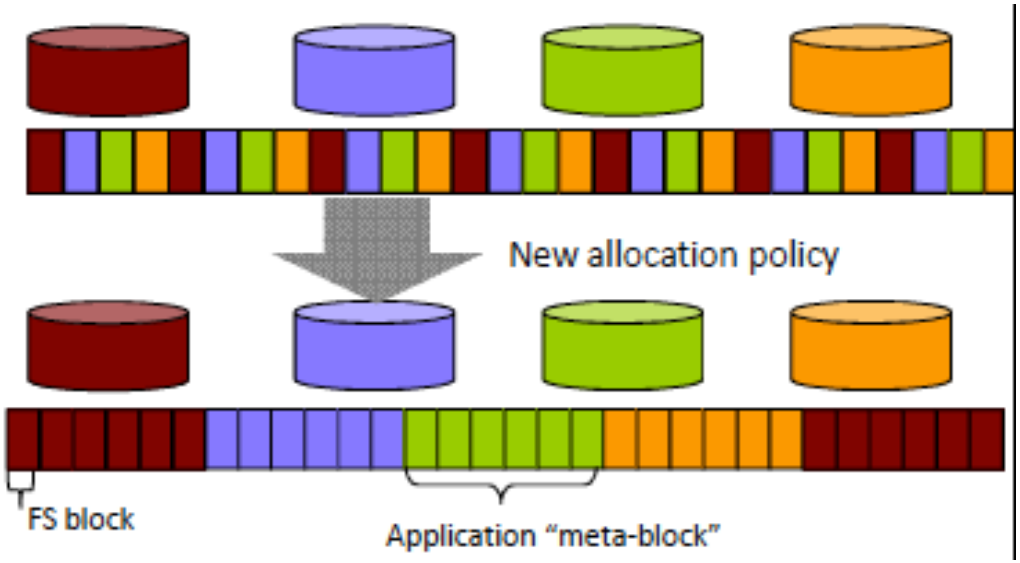
Bringing Hadoop to enterprise level

A Scalable File-system for Shared Nothing Architectures

- Advantages of using GPFS:
  - High scale (thousands of nodes, petabytes of storage), high performance, high availability, data integrity,
  - POSIX semantics, workload isolation,
  - enterprise features (security, snapshots, backup/ restore, archive, asynchronous caching and replication)
- Challenges:
  - Adapt GPFS to shared nothing clusters
  - Maximize application performance relative to cost
  - Failure is common, network is a bottleneck

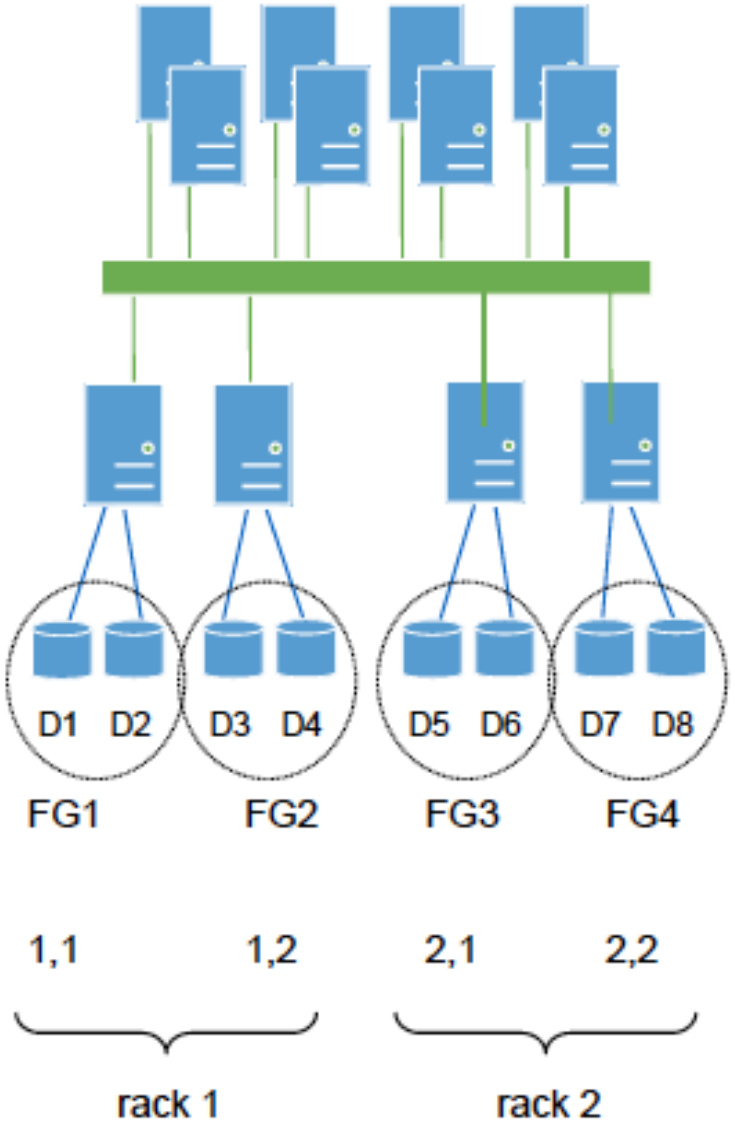
Metablocks

- Multiple block sizes in the same file system
- New allocation scheme
- Block group factor for block size
- Effective block size = block size * block group factor
- Every FPO storage pool can have its own block size



Extended Failure group

- Failure Group: collection of disks that could become unavailable simultaneously, e.g.,
  - Disks attached to the same storage controller
  - Disks served by the same NSD server
- Used for two purposes:
  - Replication: replicas of the same block must be on disks two different failure groups
  - Striping: stripe across failure groups, then across disks within failure group: D1, D3, D5, D7, D2, D4, D6, D8
- Reason: common point of failure = common resource that requires load balancing
- GPFS-FPO: "extended failure group" (additional location information)
  - Example: r,n = rack, node within rack with replication 3:
  - second copy placed in a different rack
  - third copy: same rack, but different node

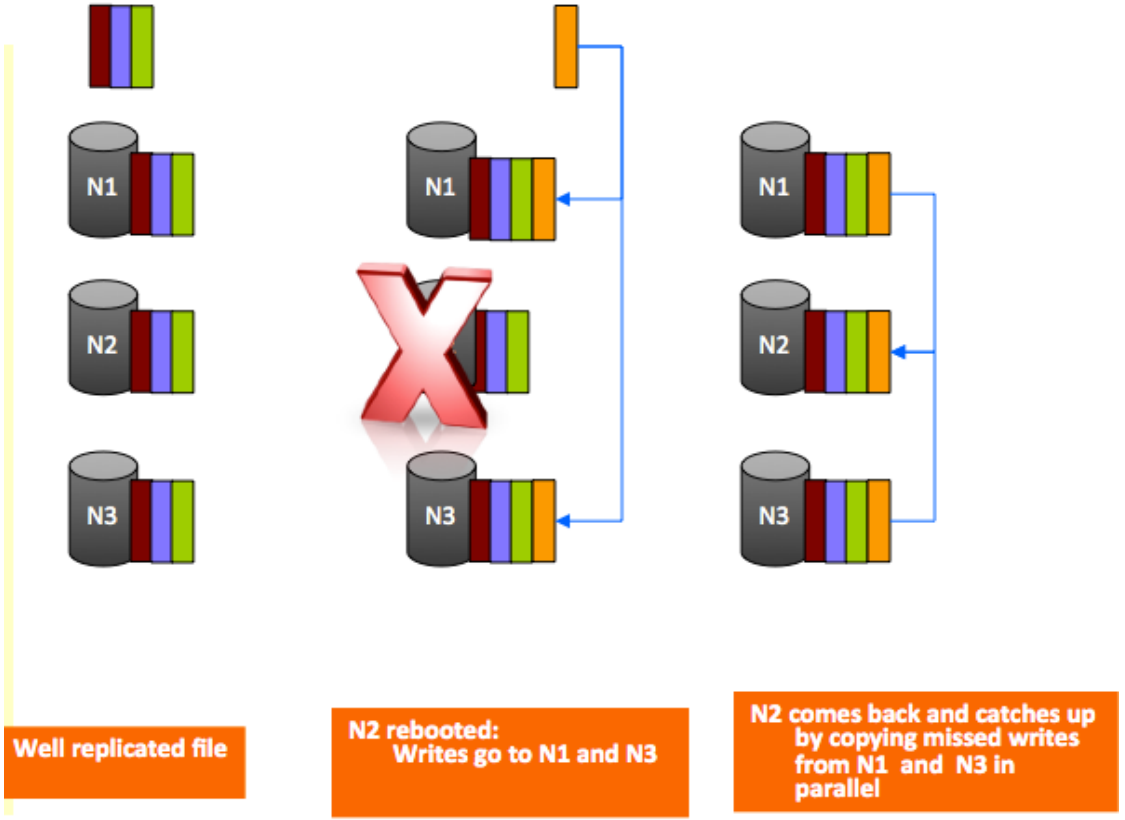


Handling failures

- policy driven
- Incorporated node failure and disk failure triggers
- Policy based recovery
  - Restripe on a node failure or disk failure
  - Alternatively, rebuild the disk when the disk is replaced

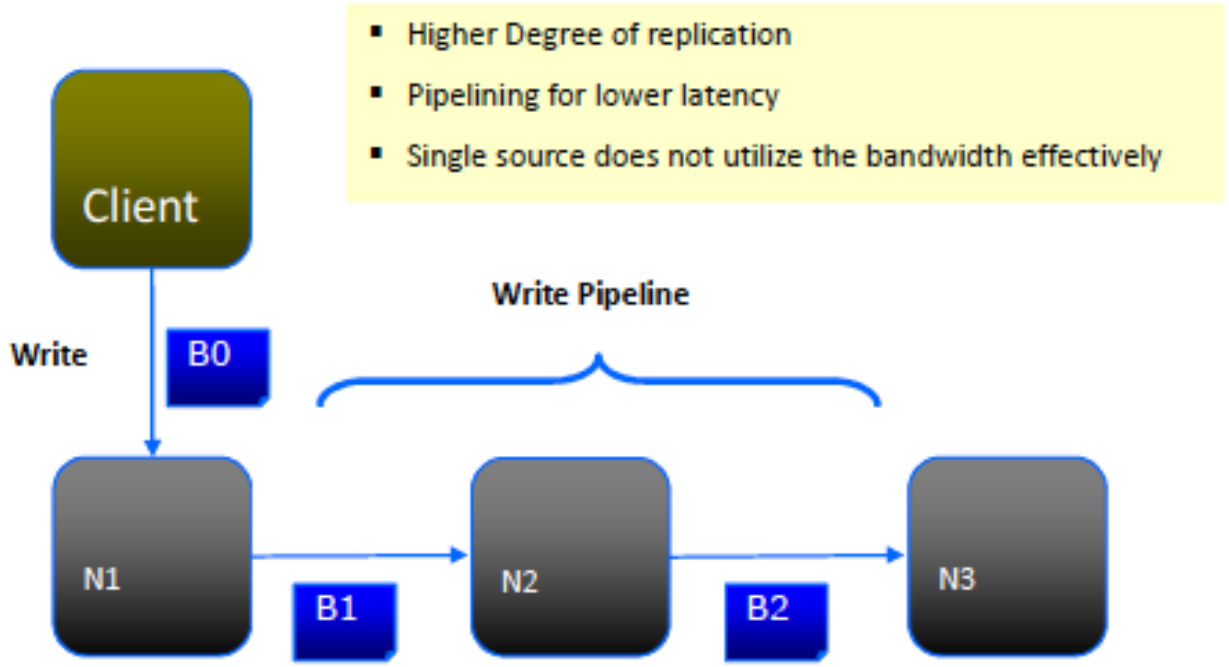
Fast recovery

- Incremental recovery
  - Keep track of changes during failure and
  - recover what is needed
- Distributed restripe
  - Restripe load is spread out over all the nodes
  - Quickly figure out what blocks are needed to be recovered when a node fails



Pipelined replication

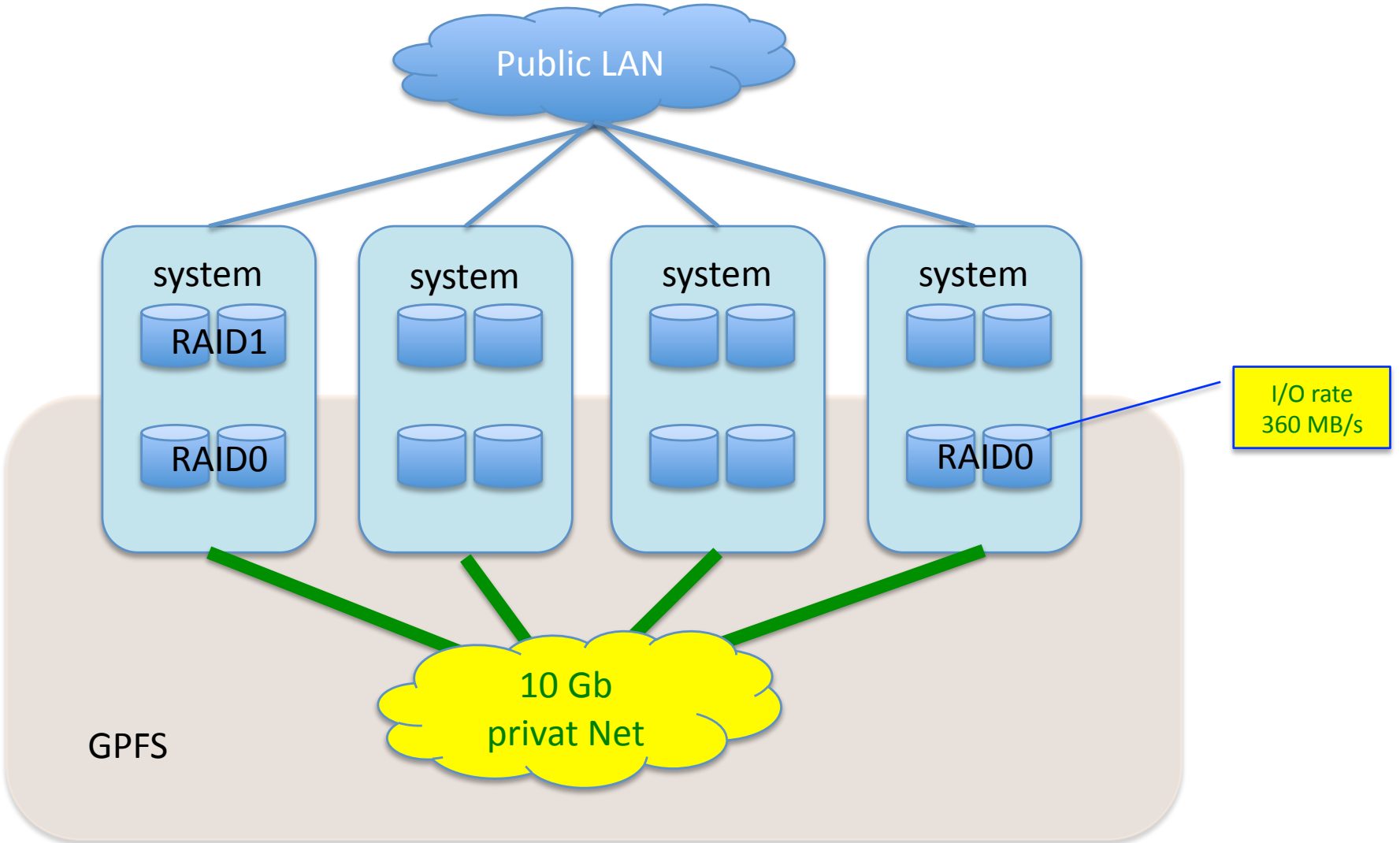
- Higher Degree of replication
- Pipelining for lower latency



Hadoop vs. GPFS-FPO

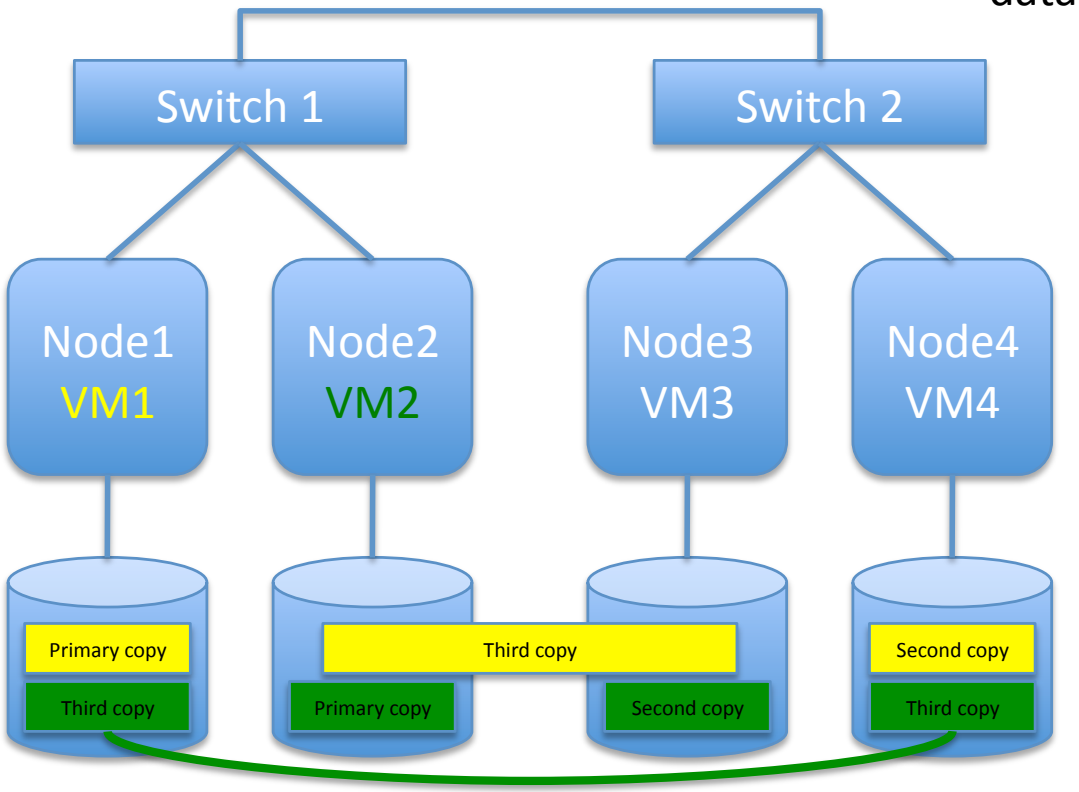
Hadoop HDFS	IBM GPFS-FPO Advantages
HDFS NameNode is a single point of failure	**No single point of failure**, distributed metadata
Large block-sizes – poor support for small files	**Variable block sizes** – suited to multiple types of data and data access patterns
Non-POSIX file system – obscure commands	**POSIX file system** – easy to use and manage
Difficulty to ingest data – special tools required	**Policy based** data ingest
Single-purpose, Hadoop MapReduce only	**Versatile, Multi-purpose**
Not recommended for critical data	**Enterprise Class** advanced storage features

GPFS Cluster as BE for VM



FPO GPFS file system

readReplicaPolicy local
restripeOnDiskFailure yes
metadataDiskWaitTimeForRecovery 60
dataDiskWaitTimeForRecovery 120



Replication
data and metadata = 3

Nsd1:failure group 1,0
Nsd2:failure group 1,1
Nsd3:failure group 2,0
Nsd4:failure group 2,1

Handling of a Node failure

```
#
# Simulating node failure
#mmshutdown storm-4-priv:
#
13:44:59.980: Recovering nodes: 192.168.1.4
13:45:00.040: Recovery: gpfs_fpo, delay 45 sec. for safe recovery.
13:45:45.042: Recovered 1 nodes for file system gpfs_fpo.
13:45:45.055: Disk failure. Volume gpfs_fpo. rc = 5. Physical volume storm4_sdb.
13:45:45.056: Calling User Exit Script gpfsRecoverFailedDisk: event diskFailure, Async command /usr/lpp/mmfs/bin/mmcommon.
13:45:45 CET: mmcommon recoverFailedDisk invoked. Parameters: gpfs_fpo storm4_sdb
recoverFailedDisk: Waiting for 120 seconds before taking any action.
#
# timeout 120 sec - configurable via "dataDiskWaitTimeForRecovery"
#
# Checking status of all disks
#
13:47:45.393: Command: tschdisk gpfs_fpo start -a
13:47:45.394: Starting gpfs_fpo disk storm1_sdb InUse/OK
13:47:45.393: Starting gpfs_fpo disk storm2_sdb InUse/OK
13:47:45.394: Starting gpfs_fpo disk storm3_sdb InUse/OK
13:47:45.393: Starting gpfs_fpo disk storm4_sdb InUse/Unavailable
13:47:45.438: Command: err 19: tschdisk gpfs_fpo start -a
13:47:45.439: No such device
#
# Attempting recovery of missing disk
#
13:47:45.480: Command: tschdisk gpfs_fpo start -F /var/mmfs/tmp/cmdTmpDir.mmcommon.45879/downDisksFile
13:47:45.481: Starting gpfs_fpo disk storm4_sdb InUse/Unavailable
13:47:45.521: Command: err 19: tschdisk gpfs_fpo start -F /var/mmfs/tmp/cmdTmpDir.mmcommon.45879/downDisksFile
13:47:45.522: No such device
#
# Suspend missing disk
#
13:47:45.711: Command: tschdisk gpfs_fpo suspend -F /var/mmfs/tmp/cmdTmpDir.mmcommon.45879/downDisksFile
13:47:45.712: Suspending gpfs_fpo disk storm4_sdb InUse/Unavailable
13:47:45.714: Command: err 0: tschdisk gpfs_fpo suspend -F /var/mmfs/tmp/cmdTmpDir.mmcommon.45879/downDisksFile
#
# automatic restripe (restore missing replica)
#
13:47:45.732: Command: tsrestripefs gpfs_fpo -r
13:52:02.714: Command: err 0: tsrestripefs gpfs_fpo -r
#
# recovery complited
#
# Re-Starting missing node:
# mmstartup storm-4-priv:
#
14:01:13.698: Command: mmdf /dev/gpfs_fpo
14:01:14.019: Command: err 0: mmdf /dev/gpfs_fpo
14:01:56.168: Accepted and connected to 192.168.1.4 storm-4-priv <c0n3>
14:01:56.205: Calling User Exit Script gpfsRestartDownDisks: event nodeJoin, Async command /usr/lpp/mmfs/bin/mmcommon.
14:01:56 CET: mmcommon restartDownDisks invoked. Parameters: storm-1-priv storm-2-priv storm-4-priv
14:01:56.600: Command: tschdisk gpfs_fpo start -a
14:01:56.601: Starting gpfs_fpo disk storm1_sdb InUse/OK
14:01:56.600: Starting gpfs_fpo disk storm2_sdb InUse/OK
14:01:56.601: Starting gpfs_fpo disk storm3_sdb InUse/OK
14:01:56.600: Starting gpfs_fpo disk storm4_sdb Suspended/Unavailable
14:01:58.033: Command: err 0: tschdisk gpfs_fpo start –a
```