



Software Defined Storage

LizardFs

Some Linux file system

Ext4: The most popular default filesystem is Ext4, or the fourth version of the extended filesystem.

XFS: XFS specializes in performance and large data files. It formats quickly and has good throughput characteristics when handling large files and when working with large disks. It also has live snapshotting features.

More Scaled Storage

RAID stands for redundant array of independent disks. RAID is a storage management and virtualization technology that allows you to group drives together and manage them as a single unit with additional capabilities.

The characteristics of a RAID array depend on its RAID level, which basically defines how the disks in the array relate to each other.

RAID 0: This level indicates drive striping.

RAID 1: RAID 1 is basically drive mirroring.

RAID 5: RAID 5 stripes data across multiple drives. RAID 5 can withstand with the loss of any 1 drive.

RAID 6: RAID 6 has the same properties as RAID 5, but provides double parity. This means that RAID 6 arrays can withstand the loss of any 2 drives.

Some scaled file system

LVM, or Logical Volume Management, is a storage device management technology that gives users the power to pool and abstract the physical layout of component storage devices for easier and flexible administration.

The main advantages of LVM are increased abstraction, flexibility, and control.

Software Defined Storage

Software-defined storage (SDS) is a computer program that manages data storage resources and functionality and has no dependencies on the underlying physical storage hardware.

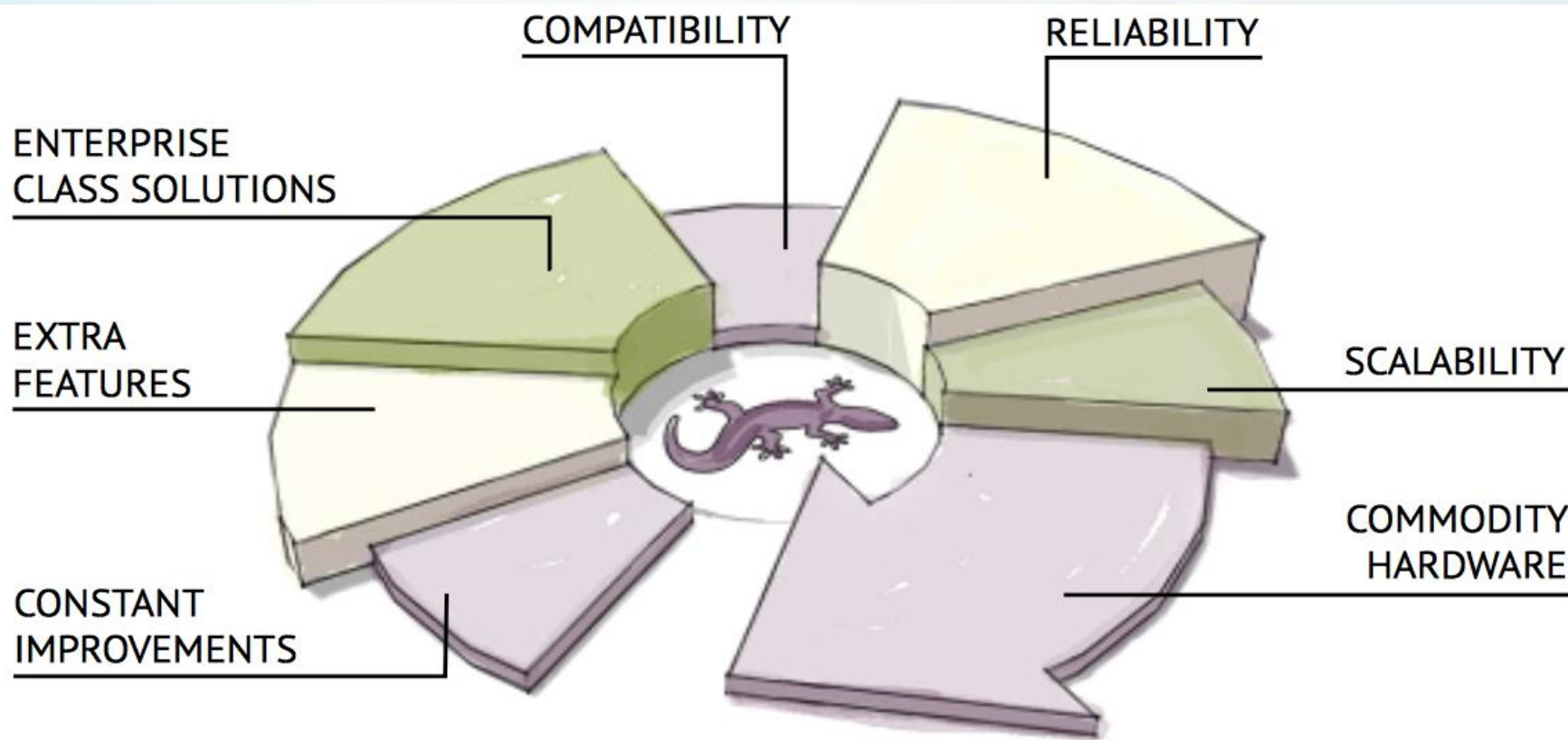
LizardFS is a Software Defined Storage application, which is

- Distributed
- Scalable
- Fault-tolerant
- Highly available
- Parallel
- GEO-REDUNDANT

We can Build our storage from commodity hardware, use your current infrastructure.

- It allows users to combine disk space located on many servers into a single namespace
- LizardFS makes files secure by keeping all the data in multiple replicas spread over the available servers.
- Disk and server failures are handled transparently, without any downtime or loss of data.
- If storage requirements grow, it is straightforward to scale an existing LizardFS installation by simply adding new servers - at any time and without any downtime.

LizardFS as a SDS..Cont.



LizardFS offers unique features such as:

- Support for many data centers and media types
- Fast snapshots
- Transparent trash bin
- QoS mechanisms
- Quotas
- A set of monitoring tools.

- Keeps metadata (e.g. file names, modification timestamps, directory trees) and the data separately.
- Metadata are kept on metadata servers, while data is kept on chunkservers.

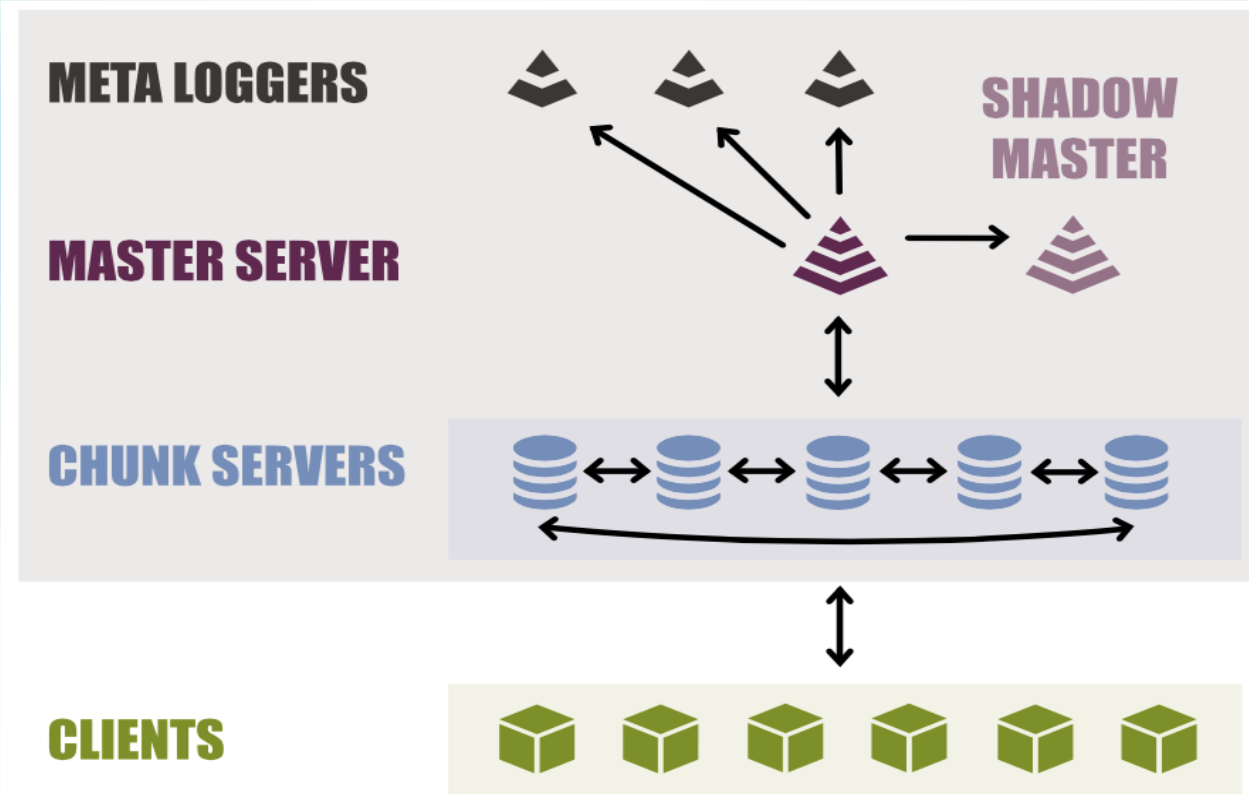
A typical installation consists of:

- At least two metadata servers, which work in the master-slave mode for failure recovery. That is called master and shadow master.
- A suggested configuration of a metadata server is a machine with fast CPU, at least 32 GB of RAM and at least one drive (preferably SSD) to store several GB of metadata.

LizardFs ChunkServer

- A set of chunkservers which store the data.
- Each file is divided into blocks called chunks (each up to 64 MB) which are stored on the chunkservers.
- A suggested configuration of a chunkserver is a machine with large disk space available either in a JBOD or RAID configuration.
- CPU and RAM are not very important.
- You can have as little as 2 chunkservers or as many as hundreds of them.

LizardFS Architecture



LizardFS: Use Cases

- Our customers use LizardFS successfully in the following scenarios:
- Archive - with LTO Library/Drive support
- Storage for virtual machines (as an OpenNebula backend or similar)
- Storage for media / CCTV etc
- Storage for backups
- As a network share drive for Windows servers
- DRC (Disaster Recovery Center)
- HPC (High Performance Computing)

LizardFS: Scalability

- If storage requirements grow, an existing LizardFS installation can be scaled by adding new chunkservers.
- Adding a new server is possible without any downtime and can be performed at any time, i.e. , it is transparent to clients.
- There are no restrictions about disk space of newly added chunkservers
- Performance of the system scales linearly with the number of disks, so adding a new chunkserver will not only increase available storage capacity but also the overall performance of the storage.
- LizardFS automatically arranges data within all chunkservers including the newly added chunkserver, as it balances disk usage across all connected nodes.
- Removing servers (e.g. for longer maintenance) is as easy as adding one.

Lizardfs Scalability

server	total space	used space
1	20 TB	15 TB (75%)
2	16 TB	12 TB (75%)
3	16 TB	12 TB (75%)
4	16 TB	12 TB (75%)
5	8 TB	6 TB (75%)
6	8 TB	6 TB (75%)
TOTAL	84 TB	63 TB (75%)



LizardFs Scalability

server	total space	used space
1	20 TB	15 TB (75%)
2	16 TB	12 TB (75%)
3	16 TB	12 TB (75%)
4	16 TB	12 TB (75%)
5	8 TB	6 TB (75%)
6	8 TB	6 TB (75%)
7	21 TB	0 TB (0%)
TOTAL	105 TB	63 TB (60%)



LizardFS Scalability

server	total space	used space
1	20 TB	12 TB (60%)
2	16 TB	9.6 TB (60%)
3	16 TB	9.6 TB (60%)
4	16 TB	9.6 TB (60%)
5	8 TB	4.8 TB (60%)
6	8 TB	4.8 TB (60%)
7	21 TB	12.6 TB (60%)
TOTAL	105 TB	63 TB (60%)



LizardFs Scalability

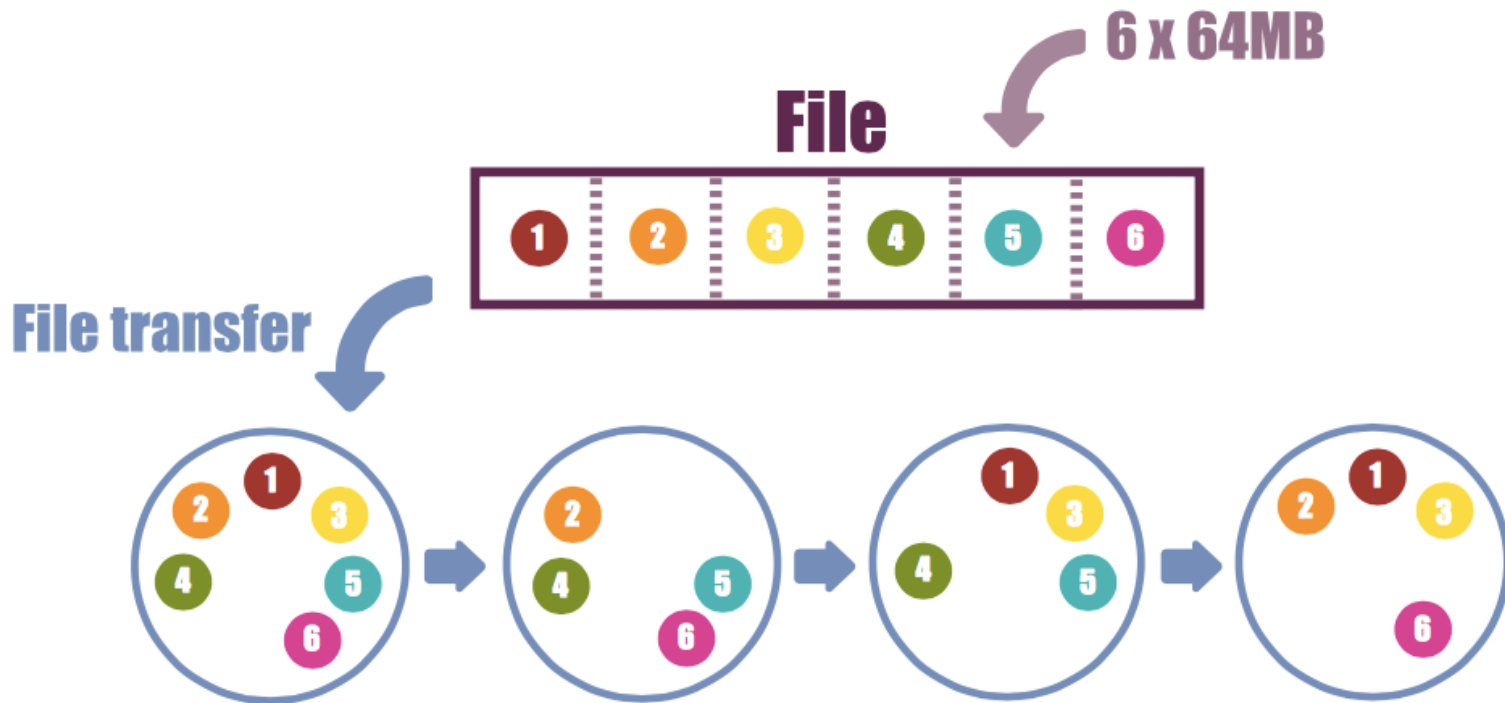
server	total space	used space
1	20 TB	15 TB (75%)
2	16 TB	12 TB (75%)
3	16 TB	12 TB (75%)
4	16 TB	6 TB (75%)
5	8 TB	6 TB (75%)
6	8 TB	6 TB (75%)
7	21 TB	12.6 TB (60%)
TOTAL	105 TB	75.6 TB (72%)



- **Snapshots** : Copying large files and directories (eg. virtual machines) can be done extremely efficiently by using the snapshot feature.
- **QOS**: LizardFS offers mechanisms that allow administrators to set read/write bandwidth limits for all the traffic generated by a given mount point, as well as for a specific group of processes spread over multiple client machines and mountpoints.

- **Data Replication:** Files stored in LizardFS are divided into blocks called chunks, each up to 64 MB in size. Each chunk is kept on chunkservers and administrators can choose how many copies of each file are maintained. For example, choosing to keep 3 copies (configuration goal=3), all the of the data will survive a failure of any two disks or chunkservers, because LizardFS will never keep 2 copies of the same chunk on the same node.

Data Replication



LizardFS is fully hardware agnostic. Commodity hardware can be utilized for cost efficiency.

LizardFS recommend that each node has at least two 1Gbps network interface controllers (NICs).

- Master / Shadow - at least 2 GHz CPU, 64bit
- RAM: depends on the number of files (e.g. 4GB often sufficient for a small installation)
- Disk - 128G, HDD is sufficient, SSD improves performance
- Chunkserver: recommended minimum 2GB RAM
- Metalogger: recommended minimum 2GB RAM

Example 1

- 2 metadata servers, each equipped with 64 GB of RAM and a RAID-1 storage build on top of two 128 GB SSD drives
- 30 chunkservers, each equipped with 12 hard drives (in JBOD configuration)
- 3 TB each
- 1 Gbit network
- replication level 3
- a metalogger running on 3 out of 30 chunkservers
- This configuration offers a possibility to store 360 TB of data in goal=3.

Examples of configuration

No downtime in case of:

failure of any 2 chunkservers or any 2 disks

failure of any single metadata server

No loss of data in case of:

failure of any 2 chunkservers or any 2 disks

failure of both metadata servers



Any Questions ?

Thank you