StorPool

Distributed Storage Software

Technical Overview

StorPool Overview

StorPool is distributed storage software. It pools the attached storage (hard disks or SSDs) of standard servers to create a single pool of **shared block storage**. The StorPool software is installed on each server in the cluster and combines the performance **and** capacity of all drives attached to the servers into **one global namespace**.

StorPool provides standard block devices. You can create one or more volumes through its sophisticated volume manager. StorPool is compatible with ext4 and XFS file systems and with any system designed to work with a block device, e.g. databases and cluster file systems (like OCFS and GFS). StorPool can also be used with no file system, for example when using volumes to store VM images directly.
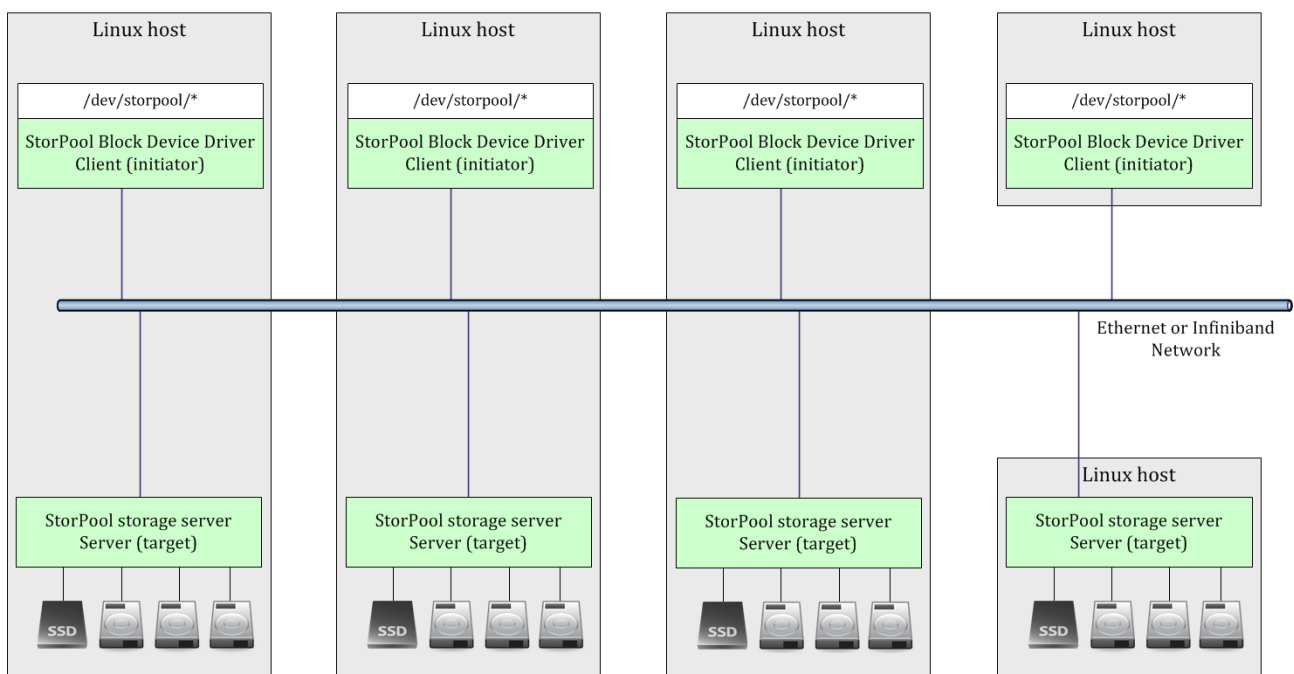
StorPool – Logical diagram

Fig 1. StorPool - Logical diagram

Redundancy is provided by multiple copies (replicas) of the data written synchronously across the cluster. Users set the number of replication copies. We recommend 3 copies as a standard and 2 copies for data that is less critical. This technology is superior to RAID in both reliability and performance:

- Unlike RAID, StorPool replication provides copies on **different servers** (not copies on **one** server). So even in the case of a server or component failure you do not lose the data that sits on this server.

- Rebuilding a failed drive/node is done from many drives (and not from one particular drive as in RAID). This means rebuild times are significantly shorter and they do not impact the overall system performance

StorPool also protects data and guarantees **data integrity** by a 64-bit checksum and a version number for each sector maintained by StorPool.

StorPool provides a very high degree of flexibility in volume management. Unlike other storage technologies, such as RAID or ZFS, StorPool does not rely on device mirroring (pairing drives for redundancy). So every disk that is added to a StorPool cluster adds capacity to the cluster, not just for new data but also for existing data. Provided that there are sufficient copies of blocks, drives can be added or taken away with no impact to the storage service. Unlike rigid systems like RAID, StorPool does not impose any strict hierarchical storage structure that links and reflects onto the underlying disks. StorPool simply creates a single pool of storage that utilizes the full capacity and performance of a set of commodity drives.

Architecture

StorPool works on a cluster of servers in a distributed shared-nothing architecture. All functions are performed by all the servers on an equal peer basis. It works on standard off-the-shelf servers running GNU/Linux or is attached via iSCSI to hosts running other hypervisors/OS.

The software consists of two parts - a storage server and a storage client that are installed on each physical server (host, node). Each host can be a storage server, a storage client, or both (i.e. converged set-up, converged infrastructure). To storage clients, StorPool volumes appear as block devices under /dev/storpool/* and behave identically to the dedicated physical alternatives. Data on volumes can be read and written by all clients simultaneously and consistency is guaranteed through a synchronous replication protocol. Volumes can be used by clients as they would use a local hard drive or disk array.
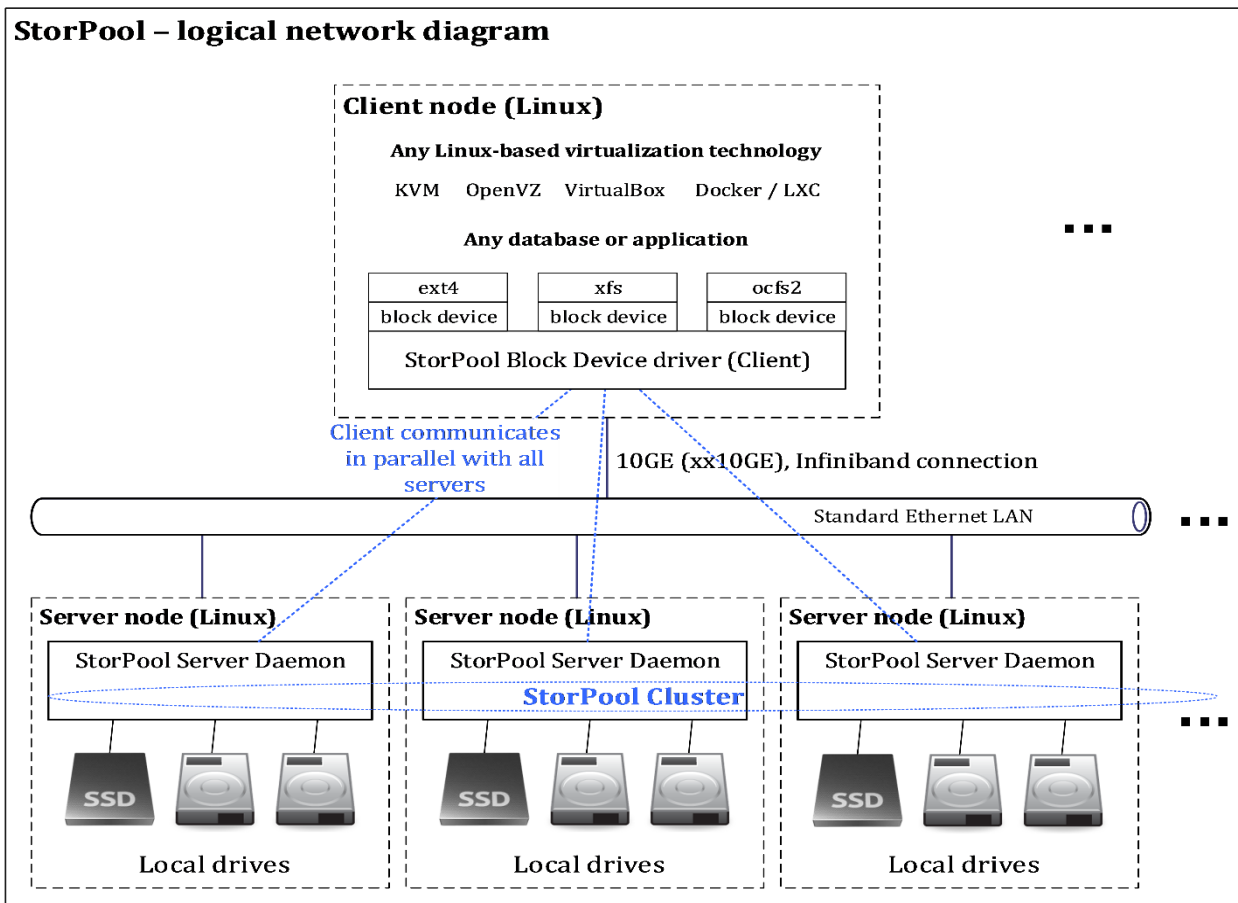

Fig. 2. Logical network diagram, segregated. In practice we recommend converged deployments – a StorPool Client and a StorPool Server on the same physical server.

Each storage node is responsible for data stored on its local drives. Storage nodes collaborate to provide the storage service. StorPool provides a shared storage pool combining all the available storage capacity. StorPool uses synchronous replication across servers. The StorPool client communicates in parallel with all StorPool servers.

In a typical deployment, each virtual machine (or container) is given its own volume in StorPool. The volume is attached to the hypervisor and the virtual machine is configured to use it directly. There is no filesystem, or VM image format (such as QCOW) in between. This gives you the capability to control each VMs storage requirements independently. For example you could have different placement (on SSDs, on HDDs) configuration, replication (2 copies, 3 copies) and different Quality of Service, QoS (IOPS, MB/s) configuration. This architecture also enables you to perform VM snapshots independently for each VM.

Using StorPool

StorPool can be thought of as a SAN replacement or a virtual SAN. StorPool is accessed with the StorPool driver, installed on the client (initiator) machine. The driver provides block-layer volumes which are visible as block devices under /dev/storpool/* on StorPool clients.

Currently, StorPool supports Linux with KVM, Xen, LXC, Docker and any other technology compatible with the Linux storage stack. StorPool is integrated with OpenStack, OpenNebula, OnApp, CloudStack, lipvirt and Proxmox, as well as custom cloud management solutions.

User applications are installed on top of the Linux block device provided by the StorPool storage solution. StorPool is compatible with any system designed to work with a block device.

For example, in a virtualized environment virtual volumes are created from the StorPool device and assigned to a single virtual machine directly mapped upon it. Alternatively, StorPool block devices can be used with shared-disk file systems (like GFS2) or Clustered LVM. It is also possible to format volumes with standard file systems such as ext4 or XFS, which enables you to mount the file system on one Linux host at a time.

StorPool combines the space, bandwidth and IOPS of all your drives and leaves plenty of CPU and RAM for compute loads. This means virtual machines, applications, databases or any other compute load can run on the same server as StorPool. When using StorPool customers can run both storage and compute loads on the same servers to achieve higher levels of utilization.

The StorPool volume management is very flexible and allows users to specify which drives should be used for which copies (e.g. the 1st copy on SSDs and the 2nd and 3rd copies on hard disks). It also allows users to specify which exact drives a specific volume should be stored on. These Data Placement Policies can also be used to ensure high availability and performance, including data locality, e.g. one copy stored on a particular server for faster local access.

StorPool has many advantages over traditional storage architectures:

- It scales in capacity *and* performance with every added drive or server
- It works on standard servers alongside applications and virtual machines
- It delivers very high performance while at the same time it takes minimal CPU and RAM from participating servers
- It provides better reliability and protects against silent data corruption

- It comes at a fraction of the cost of other storage solutions

Our key benefits, compared to other distributed storage systems, are:

- Higher Performance (IOPS, MB/s)

- Higher Efficiency (low CPU usage, low RAM usage)

- Simplicity and Compatibility

Typical Use Cases

Public / Private Clouds

StorPool is designed to combine storage and compute on the same node to deliver a high reliability, high performance cloud infrastructure while dramatically reducing the Total Cost of Ownership. StorPool can be used on the same servers as virtual machines, in conjunction with the hypervisor (e.g. KVM). StorPool can also be used without a hypervisor, in the so called "bare-metal" cloud architecture, for example to provide resilience and high availability for a stand-alone application. This can be achieved by keeping all the data in the StorPool cluster, so if the application server dies it is easy to spin it up on another machine.

Virtual Desktops (VDI)

StorPool is well suited for virtual desktop environments. StorPool's clustered and non-RAID approach leverages the throughput of the entire network and the available performance of the underlying hardware. This means it delivers incredible IOPS performance. In addition the fine-grained volume control and thin provisioning makes StorPool an ideal storage solution for VDI.

Dev Ops

As with the VDI use case StorPool's fine-grained volume control makes it easy and quick for new instances to be assigned storage from the pool. StorPool simply creates a standard storage (block) device from the cluster of servers, so no specialist skills are required to manage it. A junior System Administrator will be able to create (resize and delete) volumes and assign them to virtual machines.

How is StorPool Better?

Higher Performance

StorPool is a genuine scale-out storage system that leverages the entire cluster of servers to give massively superior IOPS performance compared to other software storage solutions (e.g. ZFS, Ceph, etc.). StorPool works much closer to the hardware, so it is much closer to the raw IO speeds of the underlying hardware. StorPool also achieves unmatched performance through its superior network protocol and its copy-on-write on-disk format, while also accelerating random writes and mitigating the "I/O blender effect".

Scalability & Flexibility

Individual drives in a StorPool cluster can be added and taken away with no impact on the service/data availability, provided sufficient copies have been made. The StorPool cluster is a flat hierarchy of drives and new drives can be added or removed at any time. When new drives are added to a StorPool cluster the capacity of these drives is simply added to the pool. StorPool

automatically ensures that copies are placed in separate fault domains (typically separate servers), so there is no need to plan and configure RAIDs.

StorPool volume management also offers a higher flexibility than most storage systems. StorPool allows for the creation of multiple volumes each with a different placement policy out of any subset of drives from the pool. For instance you might want some volumes on SATA drives, while other volumes on SSDs. Each volume can also be configured for data locality, keeping data close to where it is used.

Simplicity & Ease of Use

StorPool creates block storage volumes from a cluster of servers. Volumes can be created and assigned using the StorPool CLI. After creation, volumes are regular block devices and can be monitored with standard tools such as iostat. Unlike many other storage solutions StorPool does not require special skills to operate (i.e. it is not a new file system). A junior Sys Admin is usually able to operate StorPool.

Lower Cost

Unlike big SAN boxes from major storage vendors, StorPool uses commodity hardware. This is why StorPool could cost up to 10 times less than a comparable dedicated storage hardware solution. In addition it is cheaper to set-up, maintain and support. It is much easier to procure as well.

Feature Highlights

Scale-out, not Scale-Up

The StorPool solution is fundamentally about scaling-out (scaling by adding more drives or nodes) rather than scaling-up (adding capacity by replacing a storage box with a larger storage box). This means StorPool can scale independently by IOPS, storage space and bandwidth. There is no bottleneck or single point of failure. StorPool can grow without interruption and in small steps - one hard drive, one server and one network interface at a time.

High Performance

StorPool combines the IOPS performance of all drives in the cluster and optimizes drive access patterns to provide low latency and handling of storage traffic bursts. Load is distributed equally between all servers through striping and sharding.

High Availability and Reliability

StorPool uses a replication mechanism that slices and stores copies of the data on different servers. For primary, high performance storage this solution has many advantages compared to RAID systems and provides considerably higher levels of reliability and availability. In case of drive, server or other component failure, StorPool uses another copy of the data located on another server (or rack) and none of your data is lost or even temporarily unavailable.

Commodity Hardware

StorPool supports drives and servers in a vendor-agnostic manner, allowing you to avoid vendor lock-in. This allows the use of commodity hardware, while preserving reliability and performance requirements. Moreover, unlike RAID, StorPool is drive agnostic - you can mix drives of various types, make, speed or size in a StorPool cluster.

Shared Block Device

StorPool provides shared block devices with semantics identical to a shared iSCSI or FC disk array.

Co-existence with hypervisor software

StorPool can utilize re-purposed existing servers and can co-exist with hypervisor software on the same server. This means that there is no dedicated hardware for storage, and growing an IaaS cloud solution is achieved by simply adding more servers to the cluster.

Compatibility

StorPool is compatible with 64-bit Intel and AMD based servers. We support all the Linux-based hypervisors and hypervisor management software. Any Linux software designed to work on a shared storage solution such as an iSCSI or FC disk array will work on StorPool. StorPool guarantees the functionality and availability of the storage solution at the Linux block device interface.

CLI interface and API

StorPool provides an easy yet powerful Command Line Interface (CLI) for administration of the data storage solution. It is simple and user-friendly making configuration, provisioning and monitoring fast and efficient.

StorPool also provides a RESTful JSON API, exposing all the available functionality, so you can integrate it with any existing management system. We are also working on adding a web-based user interface (GUI).

Reliable Support

StorPool comes with reliable dedicated support:

- Remote installation and initial configuration by StorPool's specialists
- 24x7 support
- Software updates

Compatibility List & Features

Software:

- We support all the Linux distributions, typical ones are CentOS, RHEL, Debian, Ubuntu LTS
- KVM, Xen, LXC and other Linux-based virtualization technologies
- Other operating systems / hypervisors such as Microsoft Windows Server, Hyper-V, VMware vSphere/ESX/ESXi - through highly available, scale-out iSCSI
- All major file systems and logical volume managers – ext2/3/4, XFS, GFS2, LVM, LIO, etc.

Servers:

- Servers with 64-bit x86 CPUs from Intel and AMD
- Recommended minimal configuration – 3 cores, 32 GB RAM
- Recommended Sandy Bridge microarchitecture or newer CPU (e.g. Xeon E5)

- StorPool server resource utilization – 1 core, 3 GB RAM
- StorPool client resource utilization – 1 core, 1 GB RAM

Network:

- 10-Gigabit Ethernet
- Infiniband QDR/FDR

Enterprise-level Feature Set:

- **Distributed, scale-out, shared-nothing cluster architecture** – high performance, scalability, availability, reliability

- **End-to-end data integrity** – our protection mechanisms prevent silent data corruption, phantom/partial/misplaced writes, common for systems handling large amounts of data

- **Online configuration changes** – volumes are not brought down during reconfiguration.

- **In-service software upgrades** – the storage service continues operating while the cluster is being upgraded on a rolling basis.

- **Snapshots and clones** - instantaneous, copy-on-write (CoW) snapshots and clones

- **Storage Quality of Service (QoS) per volume** – fairness between volumes and user-configurable per-volume limit of IOPS and MB/s

- **Thin provisioning and thin reclaim (TRIM/DISCARD)**

- **Write-back cache (WBC)**

- **Converged Infrastructure** - Storage+Compute – use the same servers for storage and applications / virtual machines. StorPool uses only 5-15% of CPU and RAM.

- **Storage pools** – each pool is a set of drives. There may be a single pool or multiple pools per cluster. Supports hard-drive pools for high capacity use-cases and hybrid SSD+HDD pools for high performance use-cases.

- **Data locality** – each volume can be configured independently for high performance access from a particular node

- **Data tiering** – data on HDD pool or on SSD pool. Live migration between pools.

- **Synchronous replication** - replication level set per volume

- **RESTful JSON API with API failover**

- **Self-healing & automatic data balancing**

- **Shared block devices** – available on all the client nodes, simultaneously accessible and fully consistent.

Contact us to learn more or schedule a free test

StorPool Storage
info@storpool.com
+1 415 670 9320
www.storpool.com