Secure Software-Defined Storage

Sergei A. Petrenko[1,2], Diana E. Vorobieva[1] and Alexei S. Petrenko[1]

[1] *Saint-Petersburg Electrotechnical University «lETI», ul. Professora Popova, 5, St Petersburg, 197022, Russia*
[2] *Innopolis University, Universitetskaya St, 1, Innopolis, Republic of Tatarstan, 420500, 420500, Russia*

Abstract

Relevance of the development of the special software - defined data storage is due to the need to ensure the required security and stability of digital platforms and the imperfection of known models, methods, tools, server virtualization and distributed storage to work in conditions of growing security threats. Presented are the main results of solving the above problem based on software-defined approach (Software-Defined Storage), as well as author's models and methods of similarity of cloud computing in the framework of the federal project "Information Security" of the national program "Digital Economy of the Russian Federation". It is important to note that this made it possible to develop and offer a special hypervisor for solving problems of dynamic control of the semantics of digital platforms functioning based on similarity invariants. To set up an optimal algorithm for the behavior of the program-defined repository of similarity and dimensional invariants, we have proposed the well-known methods of machine learning and depth learning.

Keywords

Software-Defined Storage, cybersecurity, artificial intelligence, artificial neural network, machine learning, deep learning, big data

1. Introduction

There are two main classes of data storage systems (*SDS)* - traditional and *Software-Defined Storage (SDS)* systems. Both are high-performance software and hardware systems designed for data storage, characterized by high complexity of structure and behavior.

As a rule, the traditional storage systems are universal and initially designed to solve a certain class of functional tasks in standard operating conditions. At the same time, they have good performance characteristics, including high values of performance and fault tolerance. Traditional storage is divided into network storage devices (*Network Attached Storage, NAS*) and storage networks (*Storage Area Networks, SANs*) [1-5,10-18]. The former are systems of many individual devices to work with files connected to each other by the local Ethernet network. The second form systems from disk arrays with the block access method and communicate with each other using a high-performance fiber-optic communication network, such as *InfiniBand*. Well-known traditional storage solutions include products from *Dell EMC, IBM, NetApp* and others.

The basic functionality for data storage in software-defined *SDS*-systems is implemented by software, and the necessary hardware is selected from the list of compatible solutions. Among the main reasons for the emergence (approximately since 2016) and development of *SDS*-systems:

- Ability to get rid of the hardware dependence of one or more manufacturers;
- Flexibility to increase (or vice versa reduce) the computing resources used;
- Ability to solve new functional problems;
- Significant reduction of operating costs for operation and maintenance of these systems.

In the conditions of digitalization and implementation of federal projects of the national program "Digital Economy of the Russian Federation", such factors as the need to work with large data (Big

---

BIT-2021: XI International Scientific and Technical Conference on Secure Information Technologies, April 6-7, 2021, Moscow, Russia
EMAIL: s.petrenko@rambler.ru (A. 1); dinvor@mail.ru (A. 2); A.Petrenko1999@rambler.ru (A. 3)
ORCID: 0000-0003-0644-1731 (A. 1); : 0000-0002-3181-5961 (A. 2); 0000-0002-9954-4643 (A. 3)

CEUR Workshop Proceedings (CEUR-WS.org)

Data), the increasing volume of cloud computing, the implementation of the object model of data storage and, of course, the rapid growth of security threats have greatly influenced the development of storage. One of the first solutions to meet the new storage requirements is *Amazon Web Services (AWS)*, a cloud service based on the public cloud computing platform of the same name, with *Amazon Simple Storage Service (Amazon S3)* as its object storage. The first solution was followed by a number of similar ones, including *Microsoft, Google, IBM* and others. In 2015 IBM purchased *Cleversafe* startup with data storage object model, and then released on the storage market a corresponding solution called *IBM Cloud Object Storage*. Also known solutions are *Hitachi Content Platform (HDS), Elastic Cloud Storage (Dell EMC)* and *Nautilus (Dell EMC)* and others. For example, the solution *Nautilus (Dell EMC)* was one of the first to work with Internet streaming data of things (IoT/IIoT). According to experts, these solutions are best suited to work with poorly structured and unstructured data [6-9, 19-24]. According to the estimates of analytical companies Gartner and IDC, the three leaders of SDS-systems include solutions *Dell EMC, IBM* and *VMware*.

Also, according to analysts, the market for *SDS* solutions will evolve towards improving the three main models of access and data storage, namely, file, block and object. Their average annual growth rates for the period from 2017 to 2020 were 10.5%, 7.5% and 16.2% respectively [23-25]. At the same time, hyper-converged SDS solutions, which are understood as solutions based on hyper-converged infrastructure (*Hyper-converged infrastructure, HCI*) - a highly integrated platform that accumulates all the necessary structures, resources and tools - computing, network and data storage proper - to solve the problems were in greater demand. High performance of hyper-converged *SDS solutions* is ensured by using flash arrays, hybrid storage model implementation, as well as integration with cloud computing orchestration systems.

The known *HCI* solutions include: *Nutanix, SimpliVity (*part of *HPE), ScaleIO* from *Dell EMC, VMware (vSphere* - for server virtualization*; vSAN* - for creating high-performance hyper-converged storage for virtual machines on flash arrays and *vCenter* - for managing *vSphere* environments*), NetApp* and *Cisco (FlexPod* - for creating hyper-converged storage on *Cisco* equipment and *NetApp SolidFire* flash arrays*)* and others. Let us briefly consider the features of the above mentioned HCI solutions:

VMwarev Sphere - is a platform for virtualizing information infrastructure of a typical digital enterprise (previously - *VMware Infrastructure*). The solution implies simultaneous use of *ESXi-host (x86)* and *vCenter Server* for their centralized management. The features of the solution include: high initial cost (expensive licenses), limited support for guest operating systems, dependence on external storage for fault-tolerance scenarios, expensive implementation of distributed storage - *VMware VSAN,* etc.

Nutanix is a hyper-converged platform that supports *VMware API* for integration with data warehouses (*VAAI*). The features of this solution include high initial cost, limited set of server options and others.

Simplivity - is a platform that is based on *x86 servers, PCIe cards* and proprietary *FPGA hardware*. Devices of this platform are delivered under the brand *name OmniCube* ™ and include computing tools, data storage and Ethernet hardware with *VMware ESXi hypervisor*. Features of the solution include high dependence on proprietary *FPGA hardware*, a limited set of supported server options and others.

Rosplatform - one of the first domestic hyper-converged products that allows you to build appropriate platforms based on conventional (commodity) and relatively inexpensive servers with drives, greatly increasing the degree of useful use of equipment and the level of manageability of the platform as a whole. The features of this solution include high performance and scalability of distributed storage, support for virtualization in system containers, compatibility with *OpenStack*, compatibility with hardware (*x86*) of well-known manufacturers, a wide range of supported guest operating systems.

2. Self-healing SDS solutions

It was required to transform the observed data models into a special kind of model, which allows controlling the semantics of digital platforms under real operating conditions in order to solve the problem. For this purpose, the author's models and methods of similarity and dimensions were used [25-32, 42-44]. This allowed us to propose and implement the following prospective concept for storing similarity and dimensional invariants ("three -in-one"):

- placement of data processing and storage models, in terms of likeness and dimensional invariants, on the same server nodes of Linux system containers;
- use of hypervisor virtual machines for dynamic control of semantics of digital platforms functioning based on likeness and dimensional invariants;
- accumulation and use of reference instances of similarity and dimensional invariants for prompt self-recovery of computations and prevention of transitions of digital platforms to irreversible catastrophic states under conditions of heterogeneous mass cyber-attacks by cybercriminals, including those previously unknown.

Here the main idea is to build a system of relationships between the dimensions of processed and stored data as follows.

Let each operator of some digital platform be represented as a sum of functions φ:

$$f_u(x_1, x_2, \dots, x_n) = 0, \quad \text{and} = 1, 2, \dots, r, \tag{1}$$

where

$$f_u(x_1, x_2, \dots, x_n) = \sum_{s=1}^{q} \varphi_{us}(x_1, x_2, \dots, x_n) \tag{2}$$

and

$$\varphi_{us}(x_1, x_2, \dots, x_n) = \prod_{j=1}^{n} x_j^{\alpha_{jus}} \tag{3}$$

In this case, the provisions of the theory of dimensions and similarity [30-39] allow creating a system of requirements to the dimensions of xj, resulting from the following considerations (the record *[X]* stands for "dimensions of X"):

$$[\varphi_{us}(x_1, x_2, \dots, x_n)] = [\varphi_{uq}(x_1, x_2, \dots, x_n)], \tag{4}$$

$$\left[\prod_{j=1}^{n} x_j^{\alpha_{jus}}\right] = \left[\prod_{j=1}^{n} x_j^{\alpha_{juq}}\right], \tag{5}$$

$$\prod_{j=1}^{n} [x_j]^{\alpha_{jus}} = \prod_{j=1}^{n} [x_j]^{\alpha_{juq}}, \tag{6}$$

$$\prod_{j=1}^{n} [x_j]^{\alpha_{jus} - \alpha_{juq}} = 1, \tag{7}$$

and after logarithmization:

$$\sum_{j=1}^{n} (\alpha_{jus} - \alpha_{juq}) \cdot \ln[x_j] = 0, \tag{8}$$

$$u = 1, 2, \dots, r; \quad s = 1, 2, \dots, (q-1).$$

Then the necessary criterion of semantic correctness of the observed digital platform is the existence of a solution in which none of the variables (*ln[xj]*) is turned to zero. Here, to solve this problem one can use trivial equivalent equation transformations of the system recorded in the matrix form [42-44].

Let us now perform a critical analysis of possible variants of constructing the required SDS-system and propose a number of architectural solutions suitable for the task of storing likeness and dimensional invariants.

At rare access to data archives in the form of likeness and dimension invariants for the purpose of dynamic control of semantics of functioning of digital platforms, these data can be stored on file servers. For example, in autonomous dual-controller storage systems or on local disks of distributed storage systems with multiple redundancy. However, it is not enough to work with the mentioned data in real time. It requires large capacity "active data warehouses" with high performance and continuous retrieval and storage requirements for reference likeness and dimensional invariants. Indeed, single-controller solutions can lead to downtime risks, and dedicated hardware solutions (based on traditional *NAS and SAN*) will require significant support and maintenance resources. In addition, the operation of distributed storage systems will lead to long time delays and increase overhead due to the need to place multiple copies of data on the network nodes.

In practice, the organization of similarity storage and dimensional invariants for dynamic control of the semantics of digital platforms has been compared with the tasks of organizing storage of virtual machines with a high transactional load (*Online Transaction Processing, OLTP*) and cloud hosting, as well as the organization of high-performance computing (*HPC*) and streaming video processing (*Media & Entertainment, M&E*). Here it was necessary to provide an active traffic to reference and observed similarity and dimensional invariants, and also to provide hundreds of terabytes of memory on disks for storage of "passports" of functioning of observed digital platforms and corresponding "memory snapshots". The *I/O* load each time differed greatly: by the volume of transmitted data, by the type of addresses (random/threaded), by read/write proportions, by transfer protocols, etc. Accordingly, it was necessary to have a flexible enough organization of the storage system of similarity and dimensional invariants, which differed both in terms of media sets and *RAID* algorithms and *I/O* interfaces.

It should be noted that solving the task "on the forehead" by selecting special hardware data storage (based on traditional *NAS* and *SANs*) that meets the requirements of performance, reliability and fault tolerance will cost quite a lot (thousands and even millions of dollars). Therefore, it was decided to implement a suitable software model of data management [40, 41], the cost of which is an order of magnitude lower than that of traditional storage. At the same time, it became possible to make a free choice of data carriers, as well as ways to access them and storage scaling scenarios. In addition, it is possible to flexibly adjust performance and fault tolerance parameters, select service services, provide the required level of security and stability, etc. For example, a suitable alternative to hardware dual-controller storage of similarity and dimensional invariants is a cluster of two storage servers with shared access to a single disk space. In this case, the container with disks (enclosure, in fact - *JBOD*) can be connected to the *SAS HBA* management servers via block direct access protocol (low latency, high bandwidth). In this case, the server software is responsible for working with logical data volumes, their backup, information recovery in case of disk failures, switching between cluster nodes and related services.

Let us consider in detail possible variants of organizing software-defined storages of similarity and dimensional invariants for dynamic control of semantics of digital platforms functioning in conditions of growing security threats.

Windows Server solution 2016 (2012)

The peculiarities of such a solution include:

- *RAID* - with Storage Spaces policy technology (*2-way* or *3-way mirror*) provides performance at the hardware *RAID 10* level;
- *Spaces* - virtual disks collected from *SSD/HDD* logical pools provide high-capacity *HDD* for "cold" data, and high-performance *SSD* for "hot" data. Dynamic capacity allocation is supported;
- *Automatic Tiering* - In a two-level *SSD/HDD* storage scheme, the file system in the background tracks access to blocks of data and on a set schedule (for example, once a day) moves popular blocks to a fast layer (*SSD)*, with a granularity of *1 MB*;
- Write-back cache - smoothes write peaks to the virtual disk by SSDs from the pool, increasing *IOPS* performance;
- *SMB 3.0* - a network protocol that provides applications with access to third-party server data: shared files are presented to all nodes of the *Scale-Out File Server (SOFS) cluster*, and in case of failures, the client application is automatically serviced by the working nodes. (Microsoft recommends using direct *RDMA memory* access network adapters to offload server processors and reduce data access delays);
- *SOFS* - provides data availability and continuity of file services: a cluster of servers applies for data in shared containers (*Shared SAS JBOD*);
- *Shared SAS JBOD* - shared storage is used for server cluster on *SSD/HDD* disks. In this case, the capacity is increased by adding ordinary *NL SAS* disks to *JBOD*, as well as new *JBOD*s with the whole disks (it is possible to use relatively inexpensive *SAS*-switches); in dedicated industrial storage, even the disks themselves will cost more: *HDD* - in times, *SSD* - by an order of magnitude.

Windows Server 2016 has the functionality of synchronous replication and distributed storage on the local disks of the Storage Spaces Direct server cluster.

Jovian DSS based solution.

The solution for storing likeness and dimensional invariants based on Jovian *DSS* is a *Linux* software (and *ZFS file* system). Here, the file system with built-in support for hybrid *RAM/SSD/HDD* pools provides high performance and scalability of the solution. At the same time, repositories of similarity and dimensional invariants are built into *NAS* and *SAN* environments and provide services related to volumetric data: dynamic capacity allocation, snapshots, compression, deduplication.

Two servers on *Intel Xeon E5 26xx* processors and *JBOD* shared access are required to build a cluster of high availability data with *NFS*- and *iSCSI*-connection in the minimum configuration. The features of such a solution include:

Scalability - *128-bit ZFS* file system does not limit storage capacity with volumes up to a zetabyte on any number of disks (in *JBOD* storage clusters with a large number of capacitive *disks 6-12 Gbit SAS* is connected to management servers);

Data security - *RAID* arrays (activated remotely via command line) handle failures of up to three disks at a time; an unlimited number of snapshots are supported, which is useful for disaster data recovery;

Multi-layer caching - along with the file system, caching algorithms are inherited, and popular files are sent to one of the categories "frequently used" and "recently accessed" - separate caching areas in the RAM of the server nodes and to the *SSD*;

Hybrid storage pools - utilize *SSD I/O* performance and high *HDD* capacity in a single management logic;

On-the-fly data compression and deduplication - this is how to save disk space and reduce storage overhead (deduplication ratio can reach *3:1*, when, for example, for *3TB* of data recording *1TB* of physical disk space is enough);

Thin provisioning - virtual allocation of disk space allows you to increase storage capacity without reformatting, eliminates overspending of disks (they can be put into operation as needed);

Environmental optimization - servers can be easily adapted to the external load and set of services: selection of processors, *RAM* capacity, *SSD* pools, network interfaces. *10-40 GB Ethernet* allows coping with the “heaviest” requests and provides access to similarity variants and dimensions in the broadband range with minimal delays.

OC RAIDIX based solutions

The horizontally scalable data storage of *NetApp FAS or EMC* Isilon level could be used to solve the task. *NetApp* internal file storage system with recording everywhere (*Write Anywhere File Layout, WAFL*) is characterized by high performance - both for files and block access data (*SAN)*. This file system is deeply integrated with the *RAID manager*. For example, *RAID-DP* writes data in full stripes ("random" writes are "sequential"), which provides *"fast" RAID* in striping mode with double parity (protection against simultaneous failure of *two disks* as in *RAID 6*). And with Flash Pool and Flash Cache technologies, an optimal balance of performance and capacity is achieved in hybrid systems with an *SSD* layer above the main *HDD* capacitive array. However, test results have shown that when an array is filled and data is highly fragmented in the form of similarity and dimensional invariants, there is some *WAFL* performance loss. Despite the operation of the background defragmentator ("garbage collector") under the *OS*, 10-30% of the space had to be left free for predictable performance of intensive recording. It was found that if reading and writing have similar organization, the performance drop is not noticeable, but in case of heterogeneity of data location in stream reading there were problems.

Therefore, we decided to use the native *RAIDIX* operating system to organize a software-defined storage of similarity and dimensional invariants. The mentioned *OS* was created on the basis of the classic *RAID (read-modify-write)* approach and is characterized by high-speed algorithms of data storage and acceptable performance. For example, it provides performance of a *RAID* group with simultaneous failure of up to three disks (*RAID 7.3*) and even *32* simultaneously (*RAID N+M*) without hardware *RAID* controllers. The *RAIDIX* operating system demonstrates high speed of checksum calculation, high reliability and efficient use of useful disk space. At the same time, it allows storing and processing similarity and dimensional invariants on standard server hardware, using well-known block (*FC, iSCSI, SAS, InfiniBand*) and file (*SMB, NFS, AFP*) access protocols. And to increase the productivity of transactional operations, *SSD*-caching is provided.

3. Self-healing SDS clusters

Figure 1 shows a variant of *SDS* solution based on *Intel Xeon E5 16xx*, which flexibly increase the required amount of *RAM* and connect the necessary peripherals. *FC HBA (8-16Gb) (or 10-40 Gb Ethernet NIC)* was used to integrate the *SDS* storage system of similarity and dimensional invariants into the network environment.

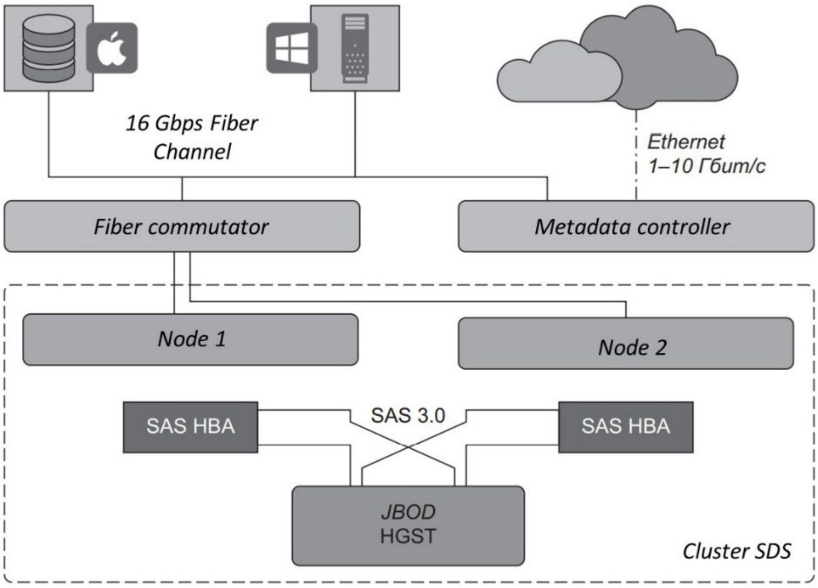


Figure 1: An example of building a FC cluster to store likeness and dimensional invariants

Since the volume of images of similarity invariants and "passports" of semantics of behavior of digital platforms can reach hundreds of terabytes (which are dozens of *HDDs*), we used SATA drives of corporate series (or related to them *NL SAS*). At the same time, the disks for storing the invariants of similarity were taken to an external *JBOD* - a dense container with duplicated *I/O*, power and ventilation modules. Here, multi-channel connection of *JBOD* to the head server ("controller") via *SAS 6-12Gb* guarantees minimal delays and wide access bandwidth to similarity invariants and dimensions stored on disks.

In the presented variant of *SDS* solution, continuity of operation is ensured by *RAIDIX Failover Cluster (FC or 40 GbE)* - a high performance platform with high data availability (without a single point of failure). The "dual-controller" software-defined storage of similarity and dimensional invariants consists of two servers, to which *JBOD* of shared access was connected. Each controller can serve a different *RAID* group. In the Active-Active cluster, the nodes are connected by an interface with low latency *FC, SAS 12 Gb* or *InfiniBand* (the cache of both controllers is always synchronized and in a coherent state). If one of the controllers is lost, it takes a few seconds to restore the *SDS* system.

JBOD has two independent *I/O* modules with expanders-duplicators. Due to the dual connection of *NL SAS* drives, data on them is available when any *I/O* module is lost (as opposed to *SATA* on the same platform). In addition, *NL SAS* serve a greater depth of the queue than *SATA*, which gives an array performance gain with the same mechanics of hard drives (in terms of cost *NL SAS* drives practically do not differ from *SATA* of the same capacity). *SAS* protocol also includes integrity control of *T10 CRC* along the whole way of extraction of reference similarity and dimensional invariants, from disk to control unit (comparison and response to security incidents).

Thus, *FC 8-16Gb/s* infrastructure is responsible for delivery and extraction of hybrid multithreaded similarity and dimensional invariants at consistently high speed (without failures). Embedding *FC Storage Cluster RAIDIX* in the existing environment significantly increased the storage volume of similarity and dimensional invariants, and improved their overall processing performance. *Dual-channel FC HBA 8 or 16Gb/s* were supplied to the cluster nodes, the array as a block access device (*LUN*) was introduced into the *SAN* and automatically configured to solve the task of dynamic control of semantics of digital platforms based on similarity and dimensional invariants. The metadata controller allowed assigning access rights to groups of administrators of the considered solution.

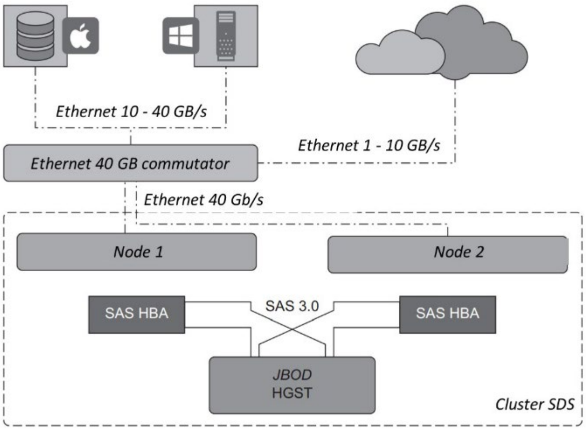


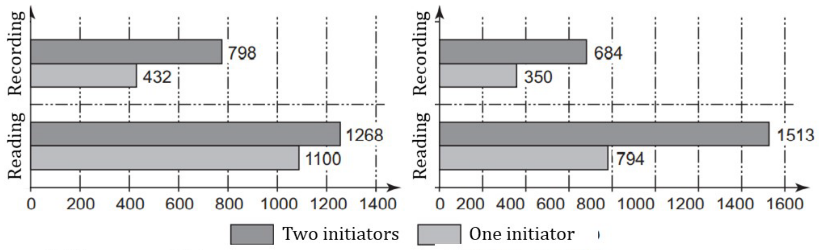
Figure 2: An example of building a NAS cluster to store similarity and dimensional invariants

A variant of the data storage solution for similarity and dimensional invariants based on the *NAS* cluster is shown in Figure 2. This solution used relatively inexpensive computing and network devices *10-40 Gb/s* (with the prospect of replacement by devices up to *100 Gb/s*). From the previous version of the storage solution based on *FC-cluster* this solution differs in external interfaces (put *10-40 Gb Ethernet NICs*) and file exchange protocols (*SMB, NFS, AFP).* The server nodes of the two solutions considered are identical: *Shared SAS JBOD* is connected to the cluster nodes (Table 1).

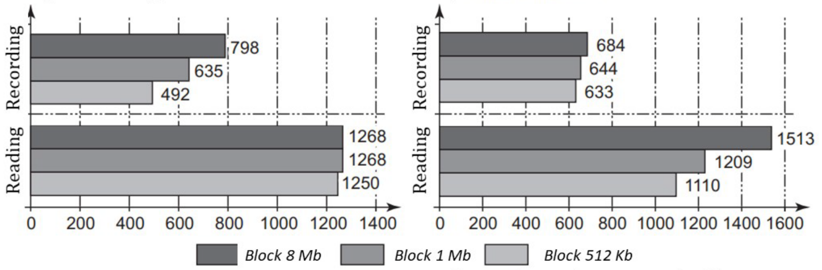
Table 1
Adaptation of storage clusters of similarity invariants for block and file access

Scecification	FC cluster	NAS cluster (Ethernet 40 GB)
Cluster node		
Central processor	1 × Intel Xeon E5-1620 v3 (4 × 3,5 Ghz)	
Core memory	4 × 16 GB DDR4-2133 reg	
Hard disk connection interface	SAS	
SAS-controller	LSI SAS HBA 9302-16e	
Network controller	ATTO 16 Gbit/s Dual Channell FC HBA	Mellanox ConnectX-3 Pro EN NIC, Dual 40/56 Gbit Ethernet
Entry SAS JBOD 60		
JBOD	HGST 4U60 Storage Enclosure 60×80 Tbytes	
Hardware capacity	480 Tbytes	

Performance test results of two designed and built clusters (*FC and NAS*) are shown in Figure 3 (*AJA System Test 2.1* and *IO Meter 2008.06.18RC2* tests were used to simulate single and multithreaded load). The second group of tests measured performance with two *512K/1M/8M* block size initiators.



a) FC-кластер, Mb/s *b) NAS-cluster, Mb/s*



c) FC- cluster (two initiators) *d) NAS-cluster, (two initiators) Mb/s*

Figure 3: Cluster performance test results (FC and NAS) for storing likeness and dimensional invariants

A solution based on clusters of several nodes

The functionality of traditional file systems was not enough to accomplish this task. Here are the known limitations of classic file systems:

- metadata and data are stored in the same partitions;
- files are "smeared" into partitions, and access delays occur;
- mechanism to prevent defragmentation is absent;
- lack of scalability by size, performance, number of files, folder nesting, etc.;
- "non-native" cross-platform.

It was necessary to use cluster file systems, including *Hyper FS* from Scale Logic, which provided high scalability and simultaneous access to data from different operating systems (in particular, through file gateways). As a result, a technical solution (Figure 3) was designed and implemented for storing similarity and dimensional invariants based on *RAIDIX* software and the Hyper FS cluster file system, which allowed organizing a single address space for block and file access.

The advantages of the obtained solution are as follows:

- up to *4 billion files* in one directory;
- up to *4096 partitions*, which can be combined into one *FS*;
- lack of a single point of failure;
- dynamic file system extension in terms of capacity and performance without downtime;
- support for the latest versions of popular operating systems - *Mac/Windows/Linux*.

It is important to note that *Hyper FS for SAN* has allowed transforming multiple file systems or iSCSI disk arrays into a storage cluster that supports simultaneous editing and playback of data from multiple client machines, provides high performance and shared access within a single namespace. The system has an optional metadata controller (*MDC*) with redundancy structure, full redundancy *SAN* structure with metadata mirroring and supports multiple path configuration in *Fibre Channel* and *iSCSI* environments. At the same time, it does not have a single point of failure and provides high stability of similarity and dimensional invariants storage.

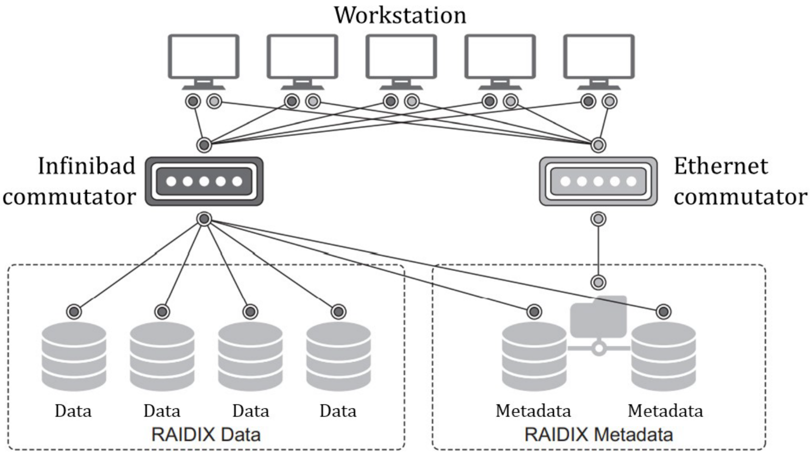


Figure 4: Example of a SDS system based on RAIDIX software and a cluster file system Hyper FS

The use of Scale-Out NAS systems for dynamic control of digital platforms semantics (Figure 4) allowed to create consolidation up to 64 nodes in a cluster with simultaneous access via different protocols (*SMB v2/v3, NFS v3/v4, FTP/FTPS, HTTP/HTTPS/WebDAV*) and load balancing between nodes *(Round-Robin, Connection Count, Load Node*), as well as support for *Active Directory*.

Essentially, it became possible to extend the functions of the *SDS*-system, namely, to offer the following additional services:

- optimization of the system for large and small files;
- support for user and folder quotas;
- *SNMP* monitoring over *SNMP* for *SONG* and *MDC;*
- *LDAP/Active Directory* support - the ability to use the local user base or integrate with *Active Director*y;
- possibility to use *ACL* on all supported operating systems.

Thus, the solution based on *RAIDIX* and *HyperFS* is characterized by high performance, single address space, simultaneous access via different protocols, low latency, high extensibility, file and block access to similarity invariants.

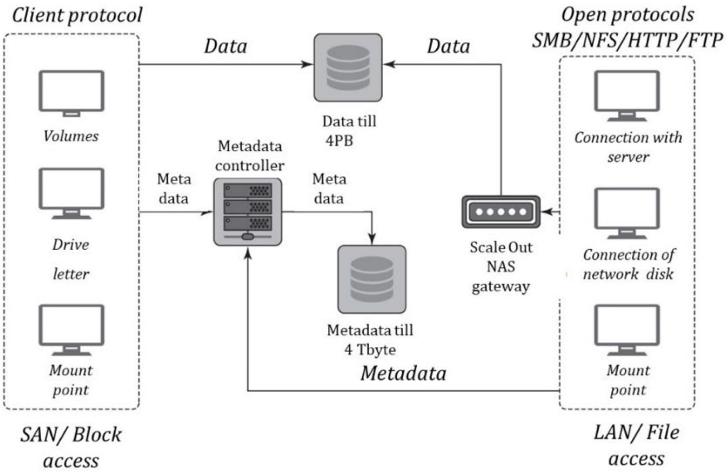


Figure 5: Example of a SDS system based on RAIDIX software and Scale-Out NAS

The proposed approach uses the multiple storage nodes (*Data storage*), dynamically allocating information between them and balancing the load; architecture - to add to the system new storage nodes on demand, without the need to transfer data and change the configuration of the system. A clear advantage of this solution is the ability to simultaneously handle data stored on one or more storage devices and from a large number of workstations at the block level and with high performance, which is impossible in a classic *SAN* architecture. On the whole, the *RAIDIX* software solution in combination with the *Hyper FS file* system meets the requirements in terms of speed and fault tolerance, and provides simultaneous parallel operation with hybrid similarity invariants and dimensions. The solution also minimizes the cost of hardware upgrades when creating storage clusters, expanding the existing infrastructure horizontally without downtime or performance degradation.

Solution based on virtualization cluster on VMware

Today server virtualization is one of the most effective ways to deploy most private and public clouds, development and testing environments, and enterprise applications. It reduces the cost of ownership of the system by saving on power and space occupied, eliminate dependence on specific branded hardware and increase uptime.

Let us list the following features of the proposed solution (Figure 5):

- Various connection protocols are used to connect *VMware ESXi* and data storage: *FCP, iSCSI, NFS*. *Virtual machines (VM)* can use the corresponding files (configuration and *vDISK*s). VMware functions related to data storage (*VMotion, VMware DRS, VMware HA* and *VMware Storage VMotion*) can be used;
- Achieved performance depends on the server used for data storage (*RAID* controller and disk functions). The maximum possible hardware bandwidth is supported. Scalability elasticity is achieved without loss of speed as the number of virtual machines and parallel highly loaded data streams increases;
- Good compatibility: *VMware ESX 5.0/5.1/5.5/6.0* and higher virtualization platforms are supported; *KVM (Kernel-based Virtual Machine); RHEV (Red Hat Enterprise Virtualization), Microsoft Hyper-V Server, XenServer*.

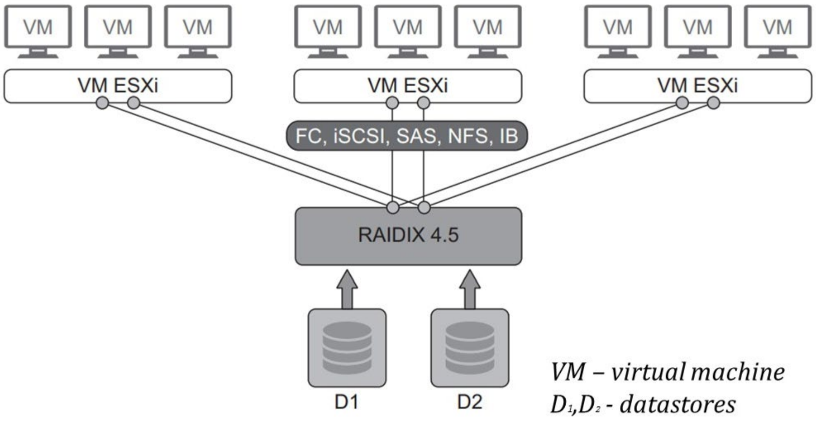


Figure 6: SDS system for storage of similarity and dimensional invariants based on VMware and RAIDIX software

Here the solution hardware infrastructure includes *10 Supermicro* servers with *Broadcom HBA* cards and *Mellanox InfiniBand* adapters. *ISCSI* over *InfiniBand* is chosen as the fastest way to synchronize in this configuration. *ISCSI* over *Ethernet* was used for automatic provisioning.

The proposed solution uses three *RAID 6i* partitions and on average three *LUNs* per partition on each server. All servers have *VMware ESXi 5.1* and *VCenter 5.1* with virtual machines (*VM*). The *VM*s serve as data storage for special applications, backup servers, file servers and more.

The selected configuration ensures efficient processing of random data and high reliability. In general, the solution is characterized:

- fail-safe storage of reference likeness and dimensional invariants;
- flexible virtualization of existing information infrastructure;
- high performance of transactional applications;
- high availability of data - "three nines" ($P = 0.999$).

4. Conclusion

The development of a software-defined data warehouse was carried out under the federal project "Information Security" of the national program "Digital Economy of the Russian Federation". In the course of the work, the possible variants of SDS-solutions for storing similarity and dimensional invariants were designed and implemented in order to introduce the semantics dynamic control of typical digital platforms functioning of the Russian Federation digital economy. The proposed options of SDS-solutions flexibly and more efficiently use servers of different types in the following main modes: hyper-convergence, computing virtualization, data storage.

Hyper-convergence. The servers simultaneously install components of computing virtualization, storage, local disks and others. Servers are assembled into local clusters with the ability to access the cloud. A special client refers to the storage of similarity and dimensional invariants using internal protocols, eliminating the need to create classic iSCSI-targeting.

Computing Virtualization. Diskless servers deliver their computing power using the cloud as a virtual machine environment. This scheme maintains the required level of computing power, and if necessary, adds the storage capacity of similarity and dimensional invariants.

Storage of data. Local hard drives are used to increase total cloud storage capacity. This scheme is necessary if you want to increase storage capacity at the expense of relatively inexpensive low-power servers filled with physical disks.

It is important that this approach, in contrast to other well-known approaches to organizing software-defined data storage, ensures the required security and stability of the information infrastructure of modern digital enterprises in conditions of growing security threats, including organizing work on a level above the computers, network equipment, storage network and means of cybersecurity and fault tolerance - the above devices and means have become software-defined components. Such software configuration of management (based on the methods of *Machine Learning* and *Deep Learning*) itself decides on which nodes to physically place the software-defined components, monitors the "health" of components of the information infrastructure in a heterogeneous mass cyber attacks of attackers (including previously unknown), decommissions unusable and connects new components of the said infrastructure. At the same time, security administrators only set basic configuration parameters, and the system independently determines on which physical nodes to place the necessary resources (computing, network and data storage) and how to manage them automatically.

Further research areas should be included:

- Development of trusted *SMART hypervisors (Storage Hypervisor*), which can be run and fine-tuned based on the methods of *Machine Learning (ML)* and *Deep Learning (DL)* - to solve the task in a controlled critical infrastructure on servers, virtual machines, within classic hypervisors and in the storage network;
- Creation of special system software on open source, *Storage Virtual Software*, eliminating dependence on specific manufacturers and providing open, secure and scalable data management to ensure the required security and stability;
- Development of application software, *Control Planes*, responsible for creation, configuration, maintenance of storage policies and broadcasting them to lower levels of resources and services to solve the problem of dynamic control of the semantics of typical digital platforms of the Digital Economy of the Russian Federation;
- Creation of additional services of safe and efficient use of similarity and dimensional invariants, *Data Services,* to ensure the required level of information security and cyber resilience.

5. Acknowledgements

The publication was carried out with the financial support of Russian Foundation for Basic Research (RFBR) in the framework of the scientific project No. 20-04-60080.

6. References

[1] H. Smith, Data center storage: cost-effective strategies implementation and management, Auerbach Publications, 2016.

[2] D. Reinsel, J. Gantz and J. Rydning, "Data age 2025: The evolution of data to life-critical. Don't focus on big data; focus on the data that's big", International Data Corporation (IDC) White Paper, 2017.

[3] R. Macedo, J. Paulo, J. Pereira and A. B. Bessani, "A Survey and Classification of Software-Defined Storage Systems", ACM Computing Surveys, May 2020.

[4] A. Verbitski et al., "Amazon Aurora: Design Considerations for High Throughput Cloud-Native Relational Databases", ACM SIGMOD, 2017.

[5] I. Canadi, S. Dong et al., RocksDBCloud:A Key-Value Store for Cloud Applications, 2017.

[6] K. Belgaied and D. Paulsen, "Improving Cassandra Latency and Resiliency with NVMe over Fabrics", NVMe Developer Days, 2018.

[7] S. W. Fong, C. M. Neumann and H.-S. P. Wong, "Phase-Change Memory-Towards a Storage-Class Memory", IEEE Tran. Electron Devices, 2017.

[8] J. Al-Badarneh et al., "Software Defined Storage for cooperative Mobile Edge Computing systems", Proc. IEEE SDS, 2017.

[9] P. X. Gao, A. Narayan, S. Karandikar, J. Carreira and S. Han, "Network Requirements for Resource Disaggregation", Proc. OSDI, 2016.

[10] Y. Shan, Y. Huang, Y. Chen and Y. Zhang, "LegoOS: A Disseminated Distributed OS for Hardware Resource Disaggregation", OSDI, 2018.

[11] M. Hilmi et al., "Analysis of Network Capacity Effect on Ceph Based Cloud Storage Performance", IEEE TSSA, 2019.

[12] V. Shankar and R. Lin, "Performance Study of Ceph Storage with Intel Cache Acceleration Software: Decoupling Hadoop MapReduce and HDFS over Ceph Storage", IEEE CSCloud, 2017.

[13] I. Adams, J. Keys and M. Mesnier, "Respecting the block interface - computational storage using virtual objects", USENIX HotStorage, 2019.

[14] D. Patterson and I. Stoica, "Cloud Programming Simplified: A Berkeley View on Serverless Computing", Berkeley Tech. Report EECS-2019–3.

[15] S. Just, "Crimson: A New Ceph OSD for the Age of Persistent Memory and Fast NVMe Storage", USENIX Vault, 2020.

[16] L. Teng, "Erasure Coding in Object Stores: Challenges and Opportunities", ACM PODC, 2018.

[17] Y. Zou, A. Raghunath, A. Chagam, S. Sen and T. Gohad, "Rethinking Software Defined Cloud Storage for Disaggregation", Proc. IEEE Service and Cloud Computing, 2019.

[18] B. Schroeder, R. Lagisetty and A. Merchant, "Flash Reliability in Production: The Expected and the Unexpected", USENIX FAST, 2016.

[19] Klimovic et al., "Pocket: Elastic ephemeral storage for serverless analytics", USENIX OSDI, 2018.

[20] Xu Kim et al., "Finding and Fixing Performance Pathologies in Persistent Memory Software Stacks", ASPLOS, 2019.

[21] S. Fedorova, "Getting storage engines ready for fast storage devices", MongoDB Engineering Journal, March 2020.

[22] R. Meredith, All-NVMe Performance Deep Dive Into Ceph SNIA Flash Memory Summit, 2019.

[23] Zhang et al., High Performance Ceph All Flash Array Software Defined Storage Solutions with NVM Technologies Flash Memory Summit, 2018.

[24] S. Sen and M. Kumar, "Distributed Block Storage using NVMe-oF", SNIA Storage Developer Conference, 2018.

[25] L. Baumann, S. B. Abraxas, L. Militano and T. M. Bohnert, "Monitoring Resilience in a Rook-managed Containerized Cloud Storage System", IEEE European Conf. on Networks and Comm. (EuCNC), 2019.

[26] X. Espinal et al., "CERN data services for LHC computing", Journal of Physics: Conference Series, vol. 898, no. 062028, pp. 8, 2017.

[27] G. Bitzes, E. A. Sindrilaru and A. J. Peters, "Scaling the EOS namespace - new developments and performance optimization", EPJ Web Conferences, vol. 214, no. 04019, pp. 8, 2018, 2019.

[28] H. Gonzalez Labrador et al., "CERNBox: the CERN cloud storage hub", EPJ Web of Conferences, vol. 214, no. 04038, pp. 8, 2018, 2019.

[29] R. P. Taylor et al., "Consolidation of cloud computing in ATLAS", Journal of Physics: Conferences Series, vol. 898, no. 052008, pp. 8, 2017.

[30] L. Bauerdick et al., "Experience in using commercial clouds in CMS", IOP Conf. Series: Journal of Physics: Conferences Series, vol. 898, no. 052019, pp. 8, 2017.

[31] O. Gutsche et al., "CMS Analysis and Data Reduction with Apache Spark", Journal of Physics: Conference Series, vol. 1085, no. 042030, pp. 6, 2018.

[32] Markov A., Barabanov A., Tsirlov V. Models for Testing Modifiable Systems. In Book: Probabilistic Modeling in System Engineering, by ed. A.Kostogryzov. IntechOpen, 2018, Chapter 7, pp. 147-168. DOI: 10.5772/intechopen.75126.

[33] Markov A., Barabanov A., Tsirlov V. Periodic Monitoring and Recovery of Resources in Information Systems. In Book: Probabilistic Modeling in System Engineering, by ed. A.Kostogryzov. IntechOpen, 2018, Chapter 10, pp. 213-231. DOI: 10.5772/intechopen.75232.

[34] Martinez Pedreira, C. Grigoras and V. Yurchenko, "JAliEn: the new ALICE high-performance and high-scalability Grid framework", EPJ Web of Conferences, vol. 214, no. 03037, pp. 8, 2018, 2019.

[35] S. Al-Kiswany and M. Ripeanu, "A Software-Defined Storage for Workflow Applications", 2016 IEEE International Conference on Cluster Computing (CLUSTER), pp. 350-353, 2016.

[36] R. Gracia-Tinedo et al., "IOStack: Software-Defined Object Storage", IEEE Internet Computing, vol. 20, no. 3, pp. 10-18, 2016.

[37] R. Gracia-Tinedo et al., "Crystal: Software-Defined Storage for Multi-tenant Object Stores", Proceedings of the 15th USENIX Conference on File and Storage Technologies (FAST ’17), pp. 243-256, 2017.

[38] R. Macedo, J. Paulo, J. Pereira and A. Bessani, "A Survey and Classification of Software-Defined Storage Systems", ACM Computing Surveys, vol. 53, no. 48, pp. 38, 2020.

[39] H. Rousseau et al., "Providing large-scale disk storage at CERN", European Physics Journal Conferences, vol. 214, no. 04033, pp. 7, 2018, 2019.

[40] Barabanov A., Markov A., Tsirlov V. Procedure for Substantiated Development of Measures to Design Secure Software for Automated Process Control Systems. In Proceedings of the 12th International Siberian Conference on Control and Communications (Moscow, Russia, May 12-14, 2016). SIBCON 2016. IEEE, 7491660, 1-4. DOI: 10.1109/SIBCON.2016.7491660.

[41] Barabanov A., Grishin M., Markov A., Tsirlov V. Current Taxonomy of Information Security Threats in Software Development Life Cycle. In: 2018 IEEE 12th International Conference Application of Information and Communication Technologies (AICT). IEEE (17-19 Oct 2018, Almaty, Kazakhstan). 2018, pp. 356-361. DOI: 10.1109/ICAICT.2018.8747065

[42] Sergei Petrenko, Developing a Cybersecurity Immune System for Industry 4.0, ©2020 River Publishers, River Publishers Series in Security and Digital Forensics. ISBN: 9788770221887, e-ISBN: 9788770221870, 386 p., https://www.riverpublishers.com/book_details.php?book_id=764 (Scopus).

[43] Sergei Petrenko. Cyber Resilience, ISBN: 978-87-7022-11-60 (Hardback) and 877-022-11-62 (Ebook) © 2019 River Publishers, River Publishers Series in Security and Digital Forensics, 1st ed. 2019, 492 p. 207 illus.

[44] Petrenko S. Cyber resilient platform for internet of things (IIoT/IoT)ed systems: survey of architecture patterns. Voprosy kiberbezopasnosti. 2021. N 2 (42). P. 81-91. DOI: 10.21681/2311-3456-2021-2-81-91.