ROS
ПЛАТФОРМА

Практические аспекты построения масштабируемых блочных хранилищ данных на базе ПО "Р-Хранилище"

Роман Морев

ведущий эксперт, Росплатформа

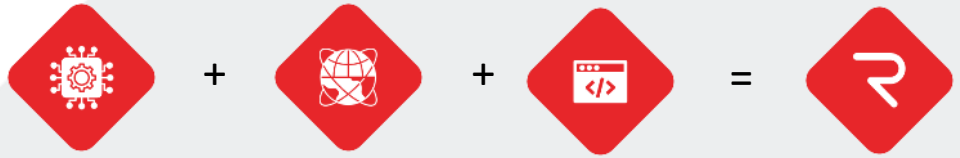
О чем расскажем:

ROS ПЛАТФОРМА

российский разработчик средств серверной виртуализации и распределенного хранения данных – основы для построения программно-определяемых и гиперконвергентных ИТ-инфраструктур, частных и публичных «облаков»

О компании Росплатформа

Продукты компании включены в Реестр российского ПО Минкомсвязи РФ и подходят для внедрения при условии обязательного импортозамещения. В основе наших решений — технологии международного класса, успешно используемые для работы миллионов виртуальных сред и хранения сотен петабайтов данных по всему миру.



Передовые технологии | Мировой опыт | Собственная разработка | Зрелый программный Продукт

Реестр российского ПО и ФСТЭК

Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации

ФСТЭК РФ

Кто использует Росплатформу в России?

Государственные Информационные системы:
ЗАГС, ЕРН, и такие организации, как МинТранс, ГОЗНАК, СургутНефтеГаз, РосГвардия, Дальневосточная Генерирующая компания (ДГК), ФБ МСЭ Минтруда РФ, АО «Прибалтийский судостроительный завод «Янтарь», а также ВУЗы и региональные администрации.

На территории России и на русском языке:

- **Поддержка**
- **Документация**
- **Разработка**

Импортозамещающие «стеки» от российских партнеров

Готовые комплексные российские решения

Российские производители прикладного ПО, СУБД, средств ИБ, бэкапов, операционных систем и т.д.

Российские производители средств виртуализации и хранения данных

Российские производители «железа»

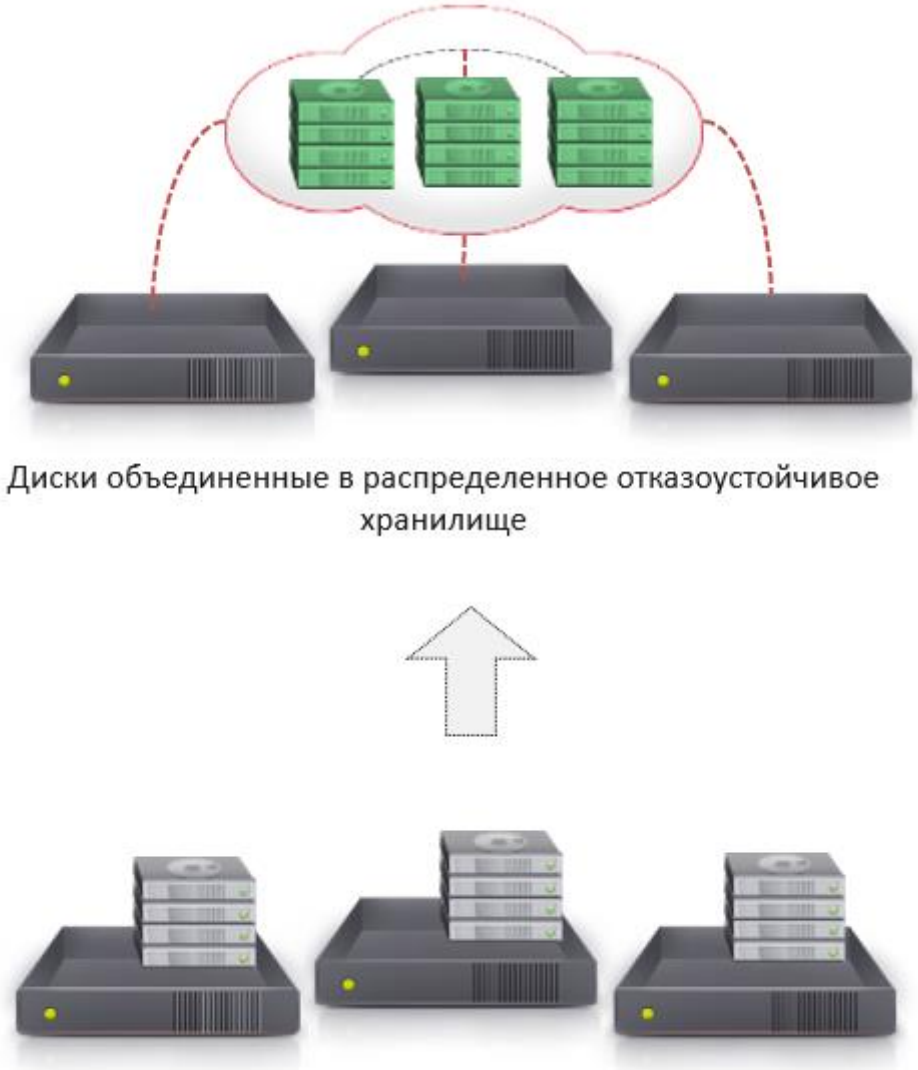
Отгружено лицензий

> 3 800 CPU

> 10 500 ТБ

Р-Хранилище.

- Распределенное программно-определяемое хранилище
- Обеспечивает высокую доступность данных за счет репликации и помехоустойчивого кодирования
- Простая горизонтальная и вертикальная масштабируемость
- Поддержка SSD кэширования для медленных массивов (SSD кэш не является обязательным для работы кластера)
- Отказоустойчивость. Возможно выход из строя до 2-х серверов
- Поддержка уровней хранения – tiering
- Возможность экспортировать хранилище для внутреннего гипервизора, iSCSI и объектное хранилище S3



P-Хранилище. Основные Службы

Служба Метаданных - MDS

- Хранят метаданные кластера
- Контролируют как файлы разделяются на фрагменты и куда они должны быть помещены
- Следят за достаточным числом реплик фрагментов
- Сохраняет журнал событий кластера
- Использует алгоритм PAXOS, для работы требуется большинство.

Служба Хранения – CS

- Хранят все данные кластера.
- Все данные разделены на фрагменты размером от 256 МБ до 2 ГБ
- Все фрагменты данных реплицируются. Реплики хранятся на разных серверах для максимальной отказоустойчивости

Клиент Хранилища

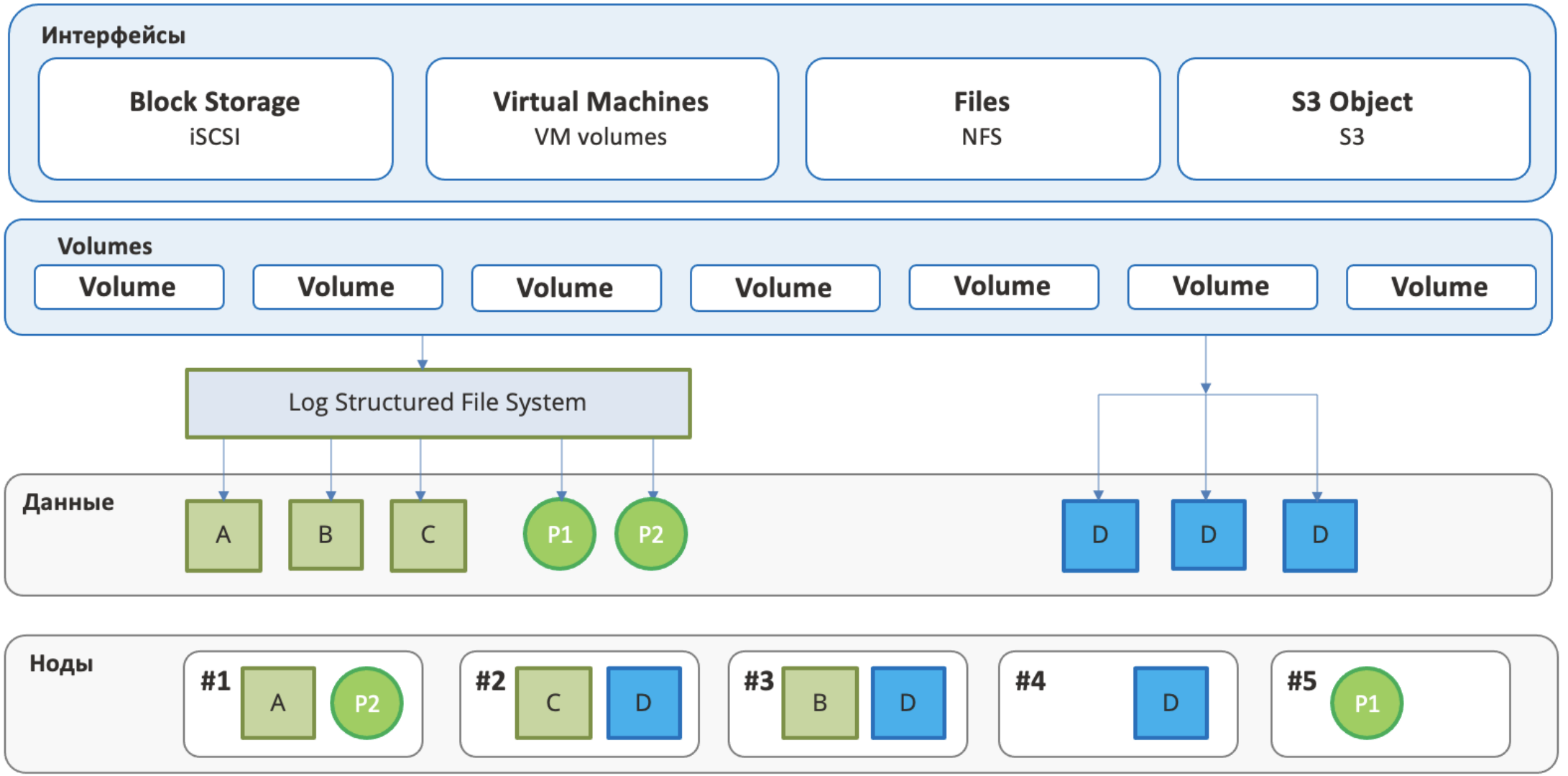
- Монтирует хранилище по заданному пути
- На основе метаданных формирует из фрагментов файловую систему, которую можно использовать для разных задач

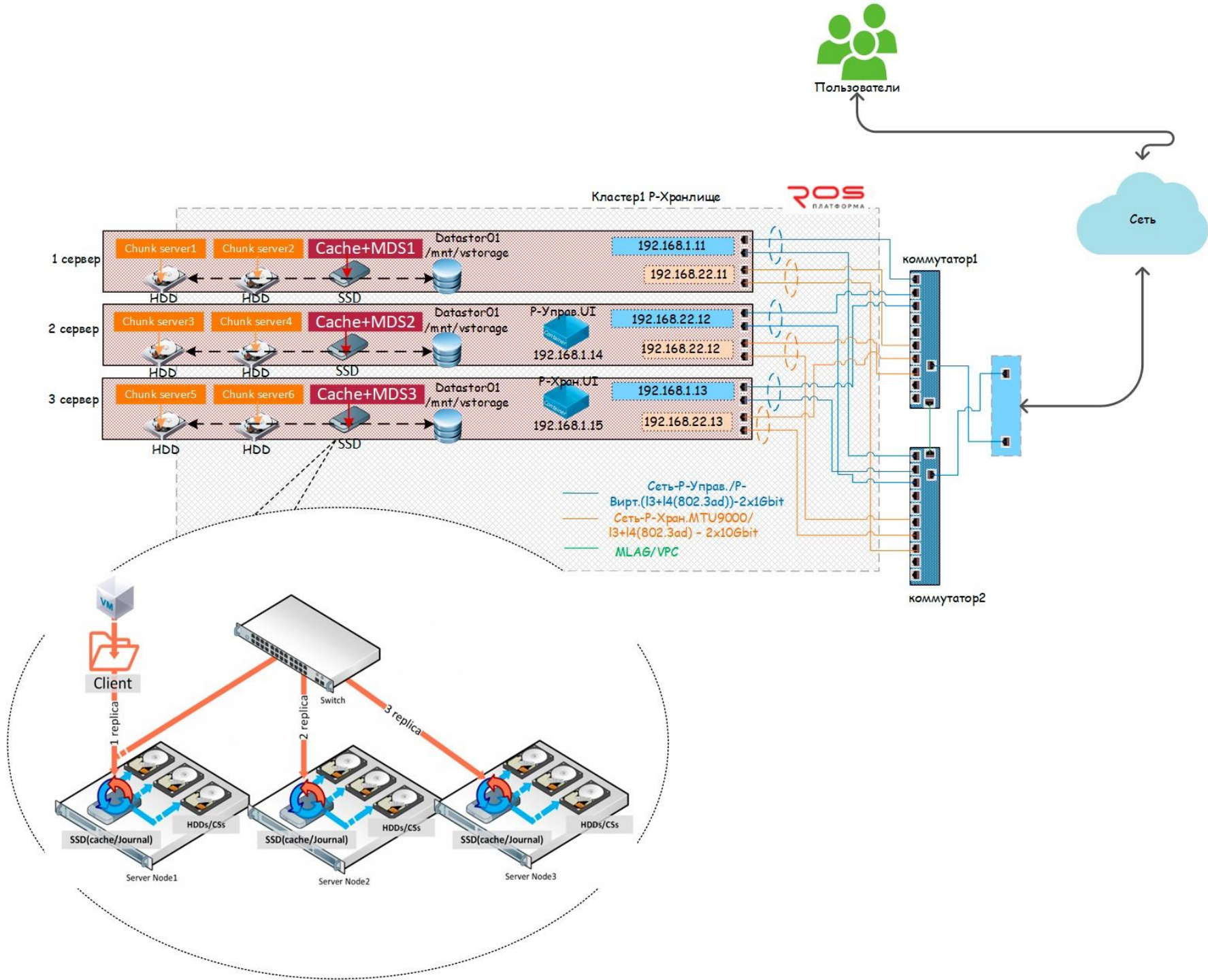
Р-Хранилище. Аппаратные требования

Минимальные рекомендуемые требования для тестирования производительности и промышленного применения:

- Минимум - 3 севера (для S3 хранилища минимум 5)
- 2 x CPU
- RAM 64+ GB
- SAS/SATA HBA-адаптер с поддержкой режима JBOD
- Для загрузки ОС – как минимум один диск объемом не менее 100 ГБ или два диска в RAID1 средствами аппаратного рейда
- 3-4 x HDD (медленный массив), каждый диск объемом не менее 100ГБ презентованные по JBOD или
- 1 x SSD + 3-4 x HDD (быстрый гибридный массив) или
- 2 x SSD (all flash), минимум 2 шт (самый быстрый массив). Все SSD диски с DWPD не менее 1
- 2 x 10 Gbit Ethernet (отдельная сеть для хранилища данных)
- 2 x 1Gbit Ethernet (для сети управления, экспортов и виртуализации)
- Для построения кластера необходимо два коммутатора Ethernet с поддержкой 1/10 Gb портов в агрегации LACP и MLAG или VPC.

P-Хранилище


Интерфейсы
Block Storage
iSCSI
Virtual Machines
VM volumes
Files
NFS
S3 Object
S3
Volumes
Volume
Volume
Volume
Volume
Volume
Volume
Volume
Log Structured File System
Данные
A
B
C
P1
P2
D
D
D
Ноды
#1
A
P2
#2
C
D
#3
B
D
#4
D
#5
P1


Пользователи
Сеть
Кластер1 Р-Хранлище
ROS
ПЛАТФОРМА
1 сервер
2 сервер
3 сервер
Chunk server1
Chunk server2
Cache+MDS1
Chunk server3
Chunk server4
Cache+MDS2
Chunk server5
Chunk server6
Cache+MDS3
HDD
SSD
DatastorO1
/mnt/vstorage
Р-Управ.UI
192.168.1.14
Р-Хран.UI
192.168.1.15
192.168.1.11
192.168.22.11
192.168.22.12
192.168.22.12
192.168.1.13
192.168.22.13
коммутатор1
коммутатор2
Сеть-Р-Управ./Р-Вирт.(l3+l4(802.3ad))-2x1Gbit
Сеть-Р-Хран.MTU9000/ l3+l4(802.3ad) - 2x10Gbit
MLAG/VPC
VM
Client
Switch
1 replica
2 replica
3 replica
SSD(cache/Journal)
HDDs/CSs
Server Node1
Server Node2
Server Node3

Архитектура Объектного Хранилища

Компоненты объектного хранилища

S3 gateway

- Реализует протокол Amazon S3
- Проводит аутентификацию пользователей и проверяет ACL
- Проксирует данные в Object Server

Name Server

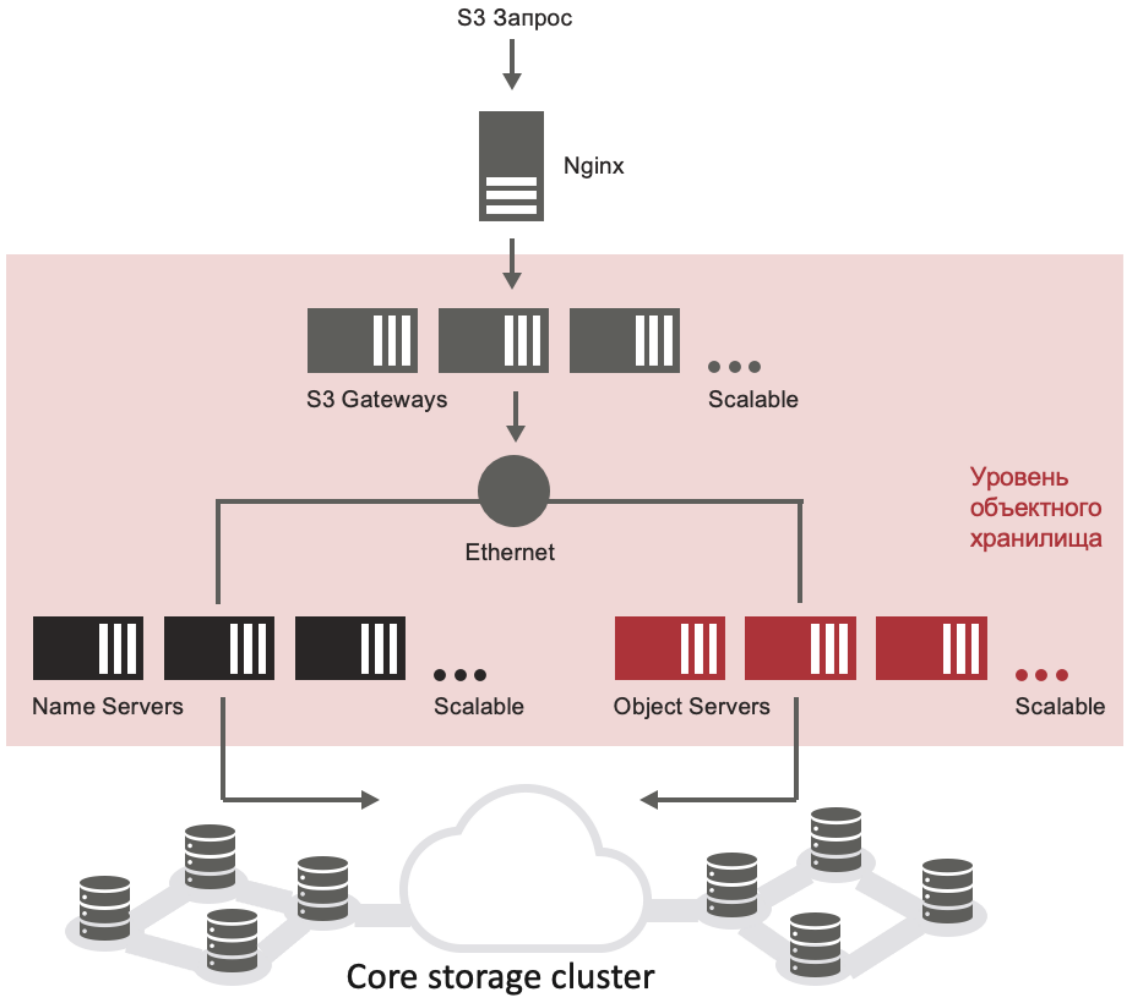
- Хранит информацию об объекте и где он хранится в Object Server
- Проводит действия над метаданными объекта

Object Server

- Хранит непосредственно сами данные
- Проводит действия над данными объекта

Другие Компоненты

- Nginx – балансировщик запросов
- Сервис конфигурации – отказоустойчивая конфигурация кластера
- Репликаторы
- Агенты



S3 Gateway

S3 Gateway – s3gw

- Изолированная реализация Amazon S3 API:
  - Авторизация и аутентификация S3 запросов (ACL)
  - Трансляция модели данных S3 во внутренние ключи для Name Server:
    - Users
    - Buckets
    - Objects
    - Versioning
    - Multipart uploads
- Кэширование метаданных
- Проверка целостности данных
- Контроль трафика и операций
- Реализован в виде однопоточного stateless сервиса
- Обрабатывает запросы сразу же после Nginx через FCGI

Name Server

Name server - NS

- Использует b-tree для внутреннего хранения данных (имен и событий)
- Хранит атрибуты объектов (информация об объекте, ACL, пользовательские метаданные)
- Позволяет сохранять метаданные объекта отдельно от самих данных.
  - Лексикографически индекс
  - Эмулируется иерархический namespace поверх сплошного namespace. Используется в операциях List
  - Быстрый листинг объектов внутри корзины
- Очереди задач и процессинг объектов
  - Гео-репликация между независимыми кластерами
  - Lifecycle правила
- Оптимизация для повышения производительности
  - Block cache
  - «Склейка» последовательных коммитов
- Асинхронных Garbage Collector (GC) для удаленных объектов

Object Server

Object Server - OS

- Использует b-tree для внутреннего хранения данных (UUID, разбиение на блоки)
- Хранит содержимое объектов в гигантских контейнера (пулах)
  - S3 объекты являются неизменяемыми
  - Объект может быть разбит на несколько частей, которые могут быть расположены в разных пулах
  - Пул содержит множество объектов однотипного размера
  - Всего 12 пулов от 4КБ до 8МБ
  - Использование внутренних ссылок на данные позволяют проводить операции COPY над объектом мгновенно без выделения места
- Оптимизация для повышения производительности
  - Block cache
  - «Склейка» последовательных коммитов
- Асинхронный Garbage Collector (GC) для высвобождения блоков, занимаемых удаленными объектами

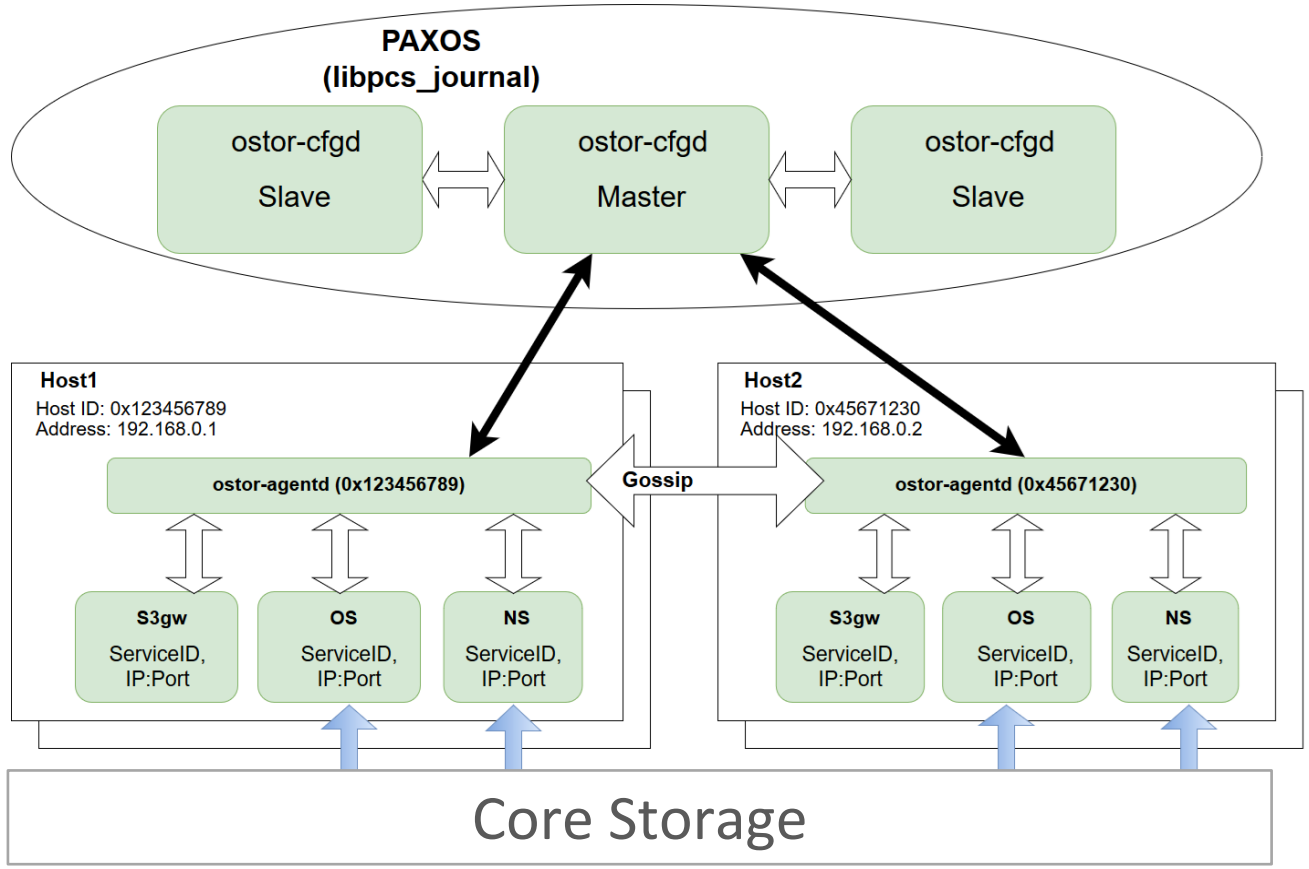
Configuration Service & Agent

Configuration Service - cfgd

- Хранит конфигурацию кластера объектного хранилища
- Использует алгоритм PAXOS (так же как и MDS) для атомарного изменения конфигурации
- Требуется большинство для работы

Агент

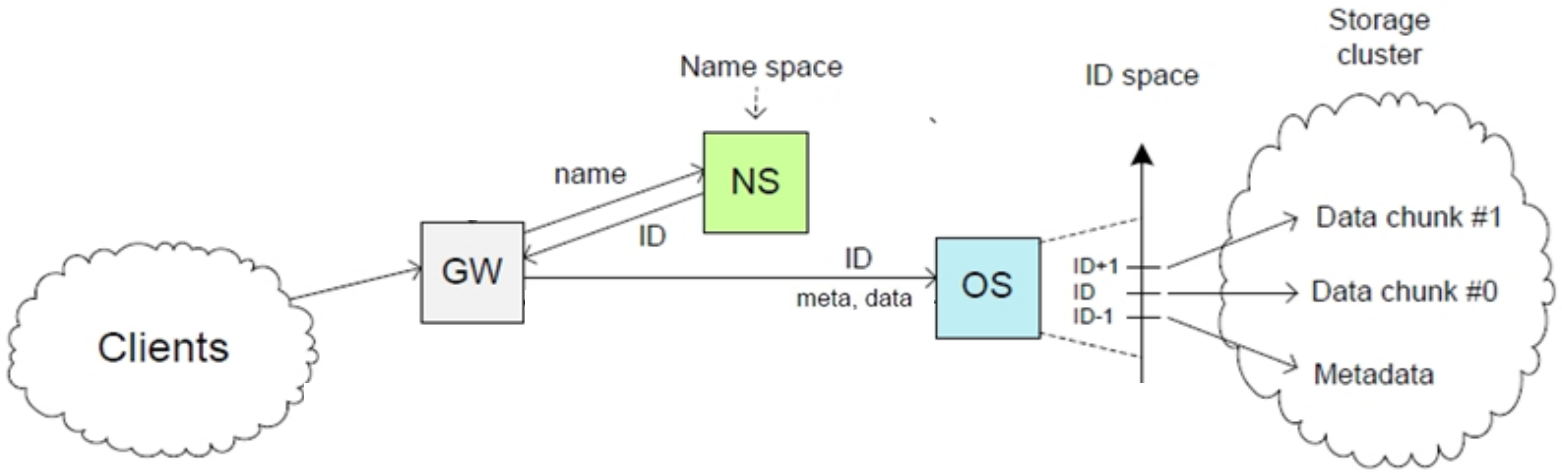
- На каждом сервере кластера отвечает за обнаружение кластера S3, развертывание сервисов S3, согласно конфигурации
- Контролирует сервисы и производит общение со всеми агентами кластера по Gossip protocol



Запись объекта

Стадии записи объекта:

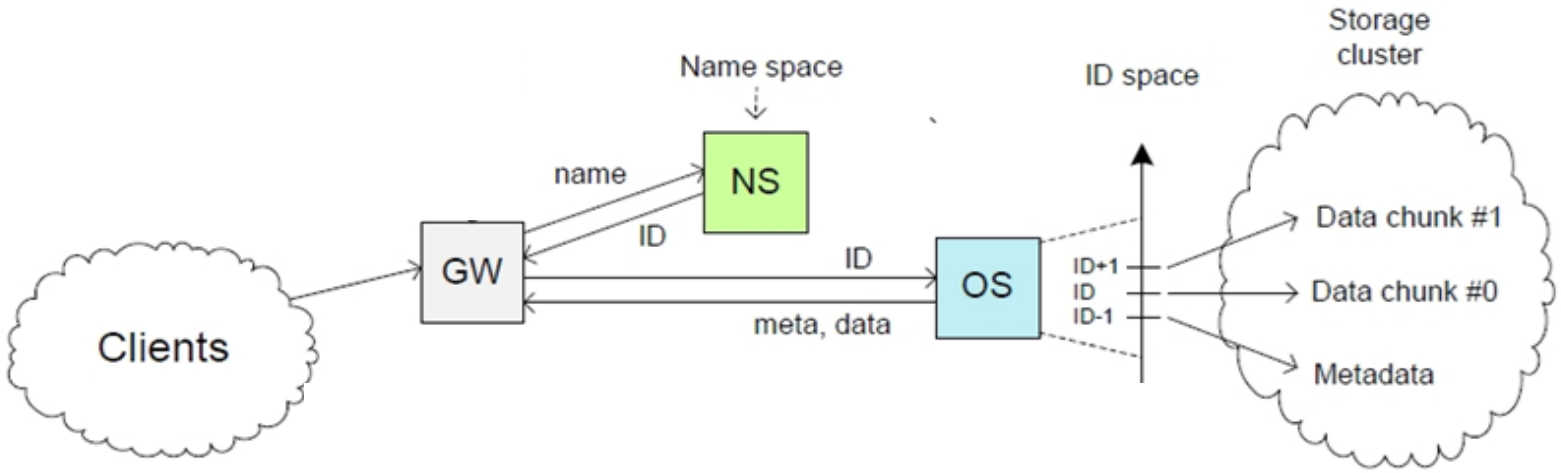
1. S3GW принимает запрос PUT
2. S3GW по началу имени объекта выбирает нужный NS
3. S3GW запрашивает создание записи в NS для объекта под именем NAME и получает в ответ внутренний ID и эпоху (timestamp)
4. S3GW по началу ID выбирает нужный OS
5. S3GW отправляет объект и пользовательские метаданные в OS
6. По завершении записи, S3GW отправляет клиенту ответ на запрос PUT



Чтение объекта

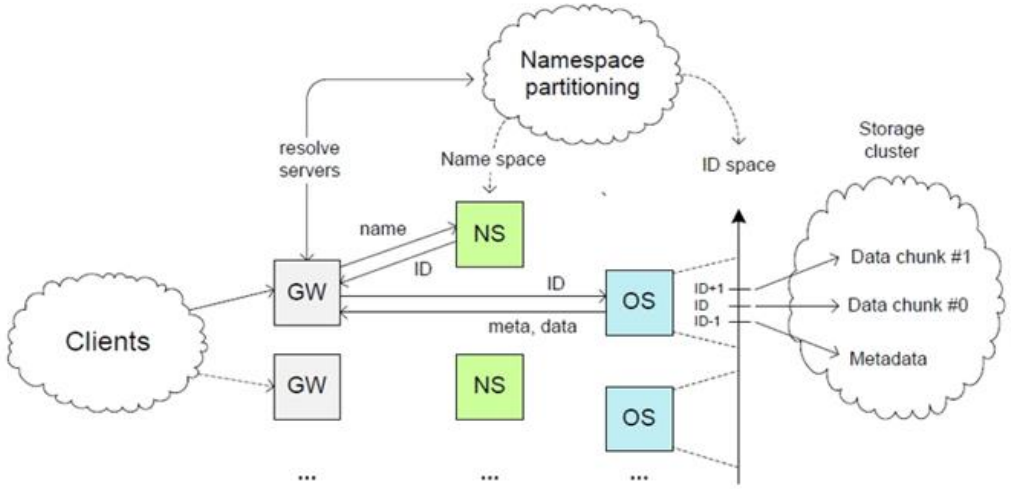
Стадии чтения объекта:

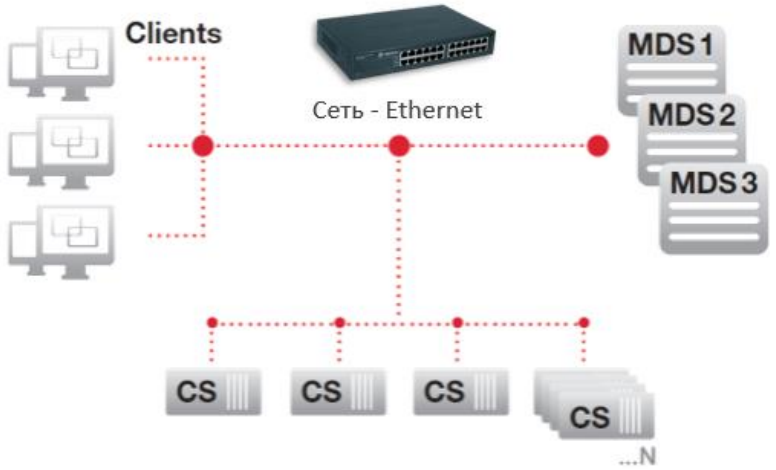
1. S3GW принимает запрос GET
2. S3GW по началу имени объекта выбирает нужный NS
3. S3GW запрашивает об объекте по имени у NS, в том числе и ACL
4. S3GW проверяет ACL
5. S3GW по началу ID объекта выбирает нужный OS и получает от него метаданные и контент
6. OS отправляет объект, пользовательские метаданные на S3GW
7. По завершении записи, S3GW отправляет клиенту ответ на запрос GET

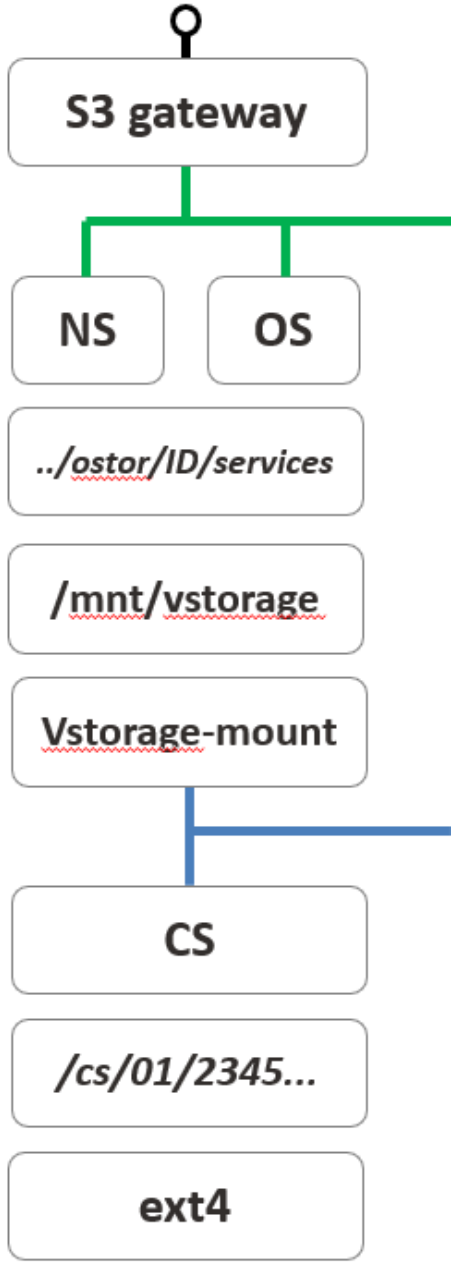


О сетевых нагрузках при выполнении запросов

- Данные от внешнего пользователя поступают по внешней сети S3 внеш.
- Данные между серверами передаются по сети Object storage внутр.
- На блочном уровне передаются данные между серверами по сети Хранения







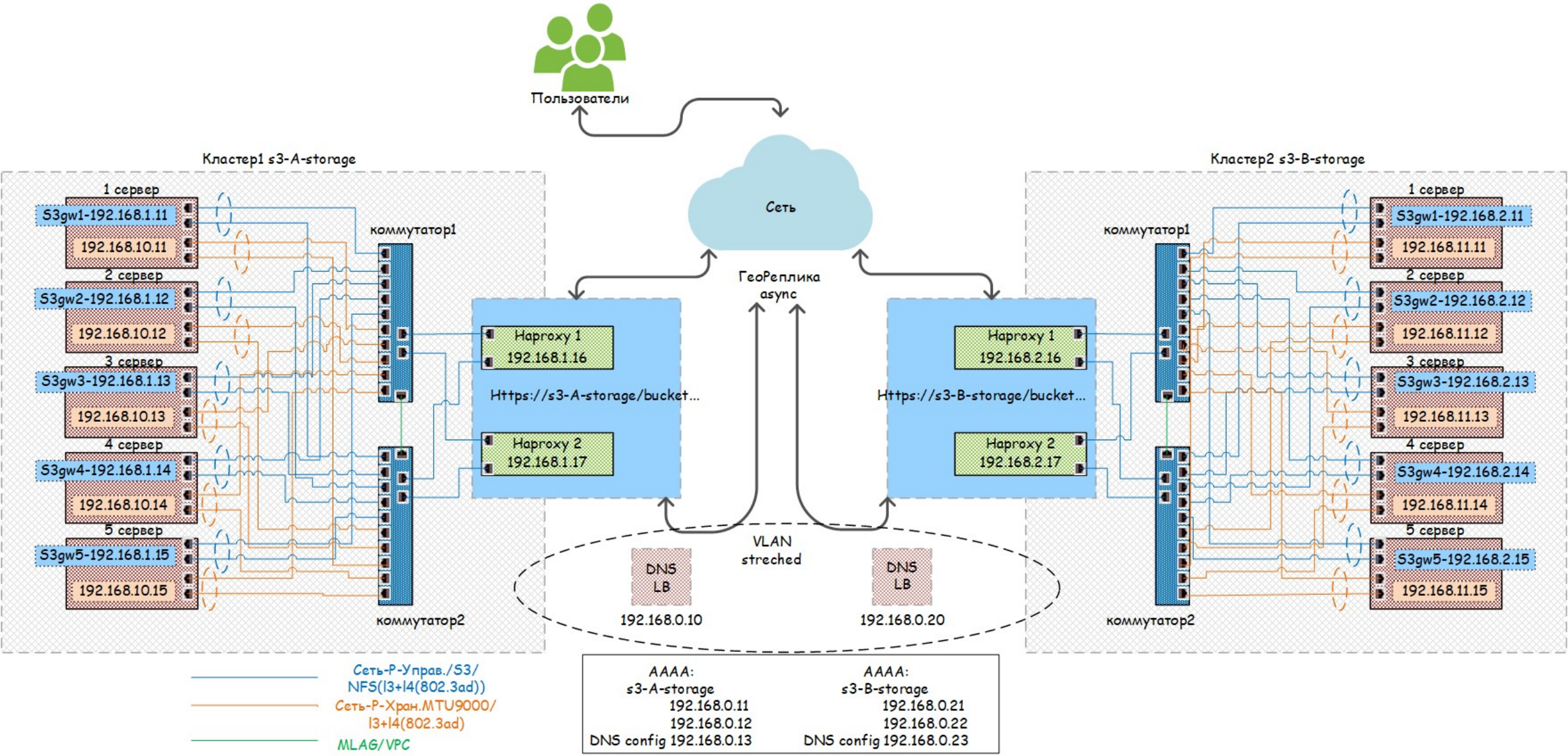
О потреблении ресурсов

Сервис	Системные требования	
	RAM	CPU cores
S3 gateway	0.5 GB	1
Name Server (NS)	1 GB	0.5
Object Server (OS)	0.5 GB	0.2
Config Service	0.25 GB	0.1

Ожидаемое потребление ресурсов на весь кластер службами NS и OS:
<Потребление_NS+OS> = <Количество_сервисов> * <потребление>

Ожидаемое потребление ресурсов на одном сервере:
<S3GW> + <CFGD> + (<Потребление_NS+OS> / min(nodes_online)

nodes_online – минимальное количество серверов онлайн при сбое оборудования.


Пользователи
Сеть
Кластер1 s3-A-storage
Кластер2 s3-B-storage
1 сервер
S3gw1-192.168.1.11
192.168.10.11
2 сервер
S3gw2-192.168.1.12
192.168.10.12
3 сервер
S3gw3-192.168.1.13
192.168.10.13
4 сервер
S3gw4-192.168.1.14
192.168.10.14
5 сервер
S3gw5-192.168.1.15
192.168.10.15
коммутатор1
коммутатор2
Haproxy 1
192.168.1.16
Haproxy 2
192.168.1.17
Https://s3-A-storage/bucket...
ГеоРеплика
async
Haproxy 1
192.168.2.16
Haproxy 2
192.168.2.17
Https://s3-B-storage/bucket...
VLAN
streched
DNS
LB
192.168.0.10
DNS
LB
192.168.0.20
1 сервер
S3gw1-192.168.2.11
192.168.11.11
2 сервер
S3gw2-192.168.2.12
192.168.11.12
3 сервер
S3gw3-192.168.2.13
192.168.11.13
4 сервер
S3gw4-192.168.2.14
192.168.11.14
5 сервер
S3gw5-192.168.2.15
192.168.11.15
Сеть-Р-Управ./S3/
NFS(l3+l4(802.3ad))
Сеть-Р-Хран.MTU9000/
l3+l4(802.3ad)
MLAG/VPC
AAAA:
s3-A-storage
192.168.0.11
192.168.0.12
DNS config 192.168.0.13
AAAA:
s3-B-storage
192.168.0.21
192.168.0.22
DNS config 192.168.0.23

Сравнение Ceph16.2.7 с Р-хранилищем 7.0.13-35
Конфигурация оборудования на тесте

ROS ПЛАТФОРМА

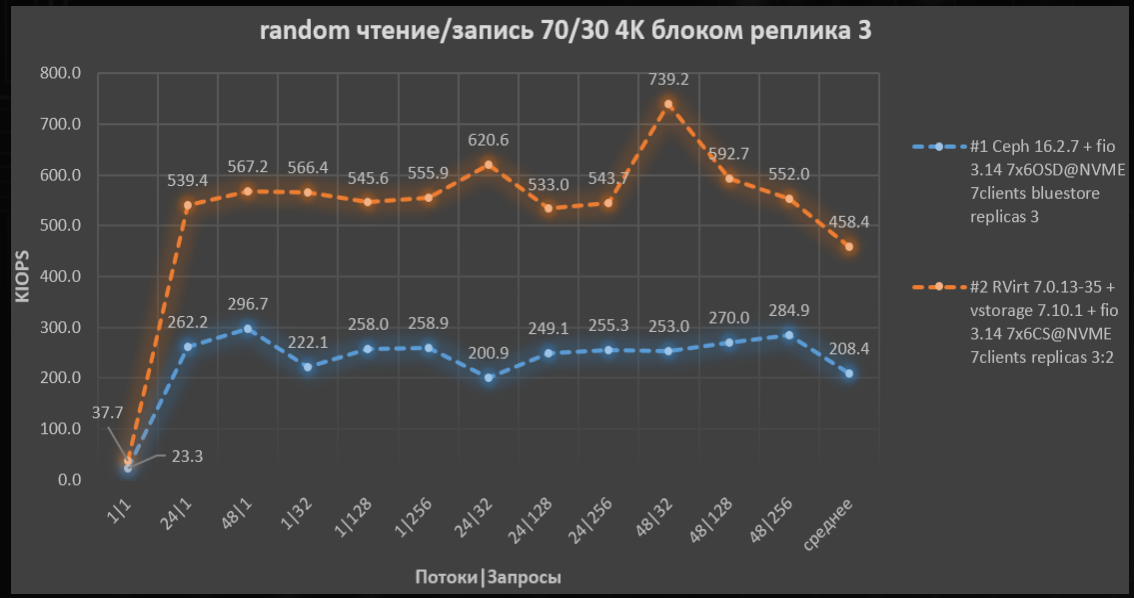
Rosplatforma 7.0.13-35 7.10.1.4-1.rv7.1
Client: 1
MDS: 1
CS: 6
CPU: 48 (2S x 12C x 2T) Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz
Memory: 128695 MB, 521154 MB swap, 2 NUMA node(s)

Products:
- fio: 3.14
- rvirt: 7.0.13-35
- vstorage-chunk-server: 7.10.1.4-1.rv7.1
- vstorage-client: 7.10.1.4-1.rv7.1
- vstorage-metadata-server: 7.10.1.4-1.rv7.1
- vzkernel: 3.10.0-1062.12.1.rv7.131.10.1

CEPH 16.2.7
Monitor: 0
Manager: 0
OSD: 6
CPU: 48 (2S x 12C x 2T) Intel(R) Xeon(R) Gold 6126 CPU @ 2.60GHz
Memory: 128520 MB, 492482 MB swap, 2 NUMA node(s)

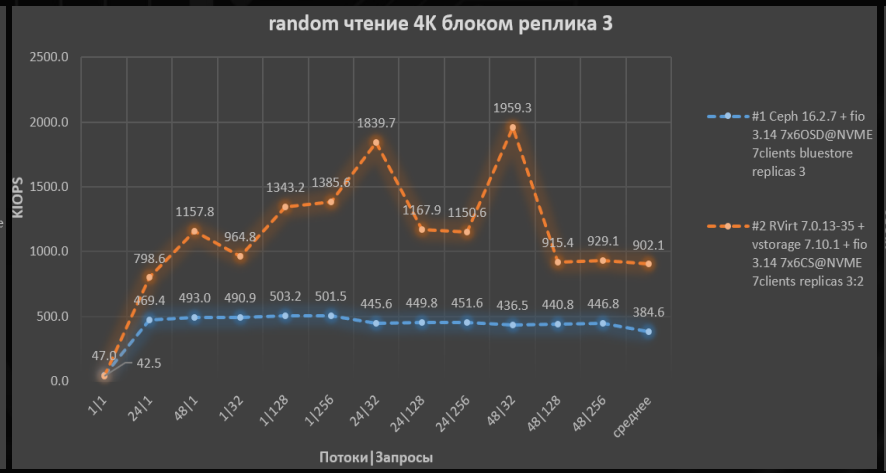
Products:
- ceph: 16.2.7
- fio: 3.14
- kernel: 4.18.0-348.7.1.el8_5.x86_64

7 серверов, по 2 процессора (48 ядер), 128ГБ память,
SSD M.2: 500GB WD Black SN750 (WDS500G3X0C-00SJG0) – система,
SSD U.2: 6 x 960GB Ultrastar DC SN630 (WUS3BA196C7P3E3) – хранилище,
Сеть: 2 x 25Gbps Mellanox MT27710 ConnectX-4 Lx



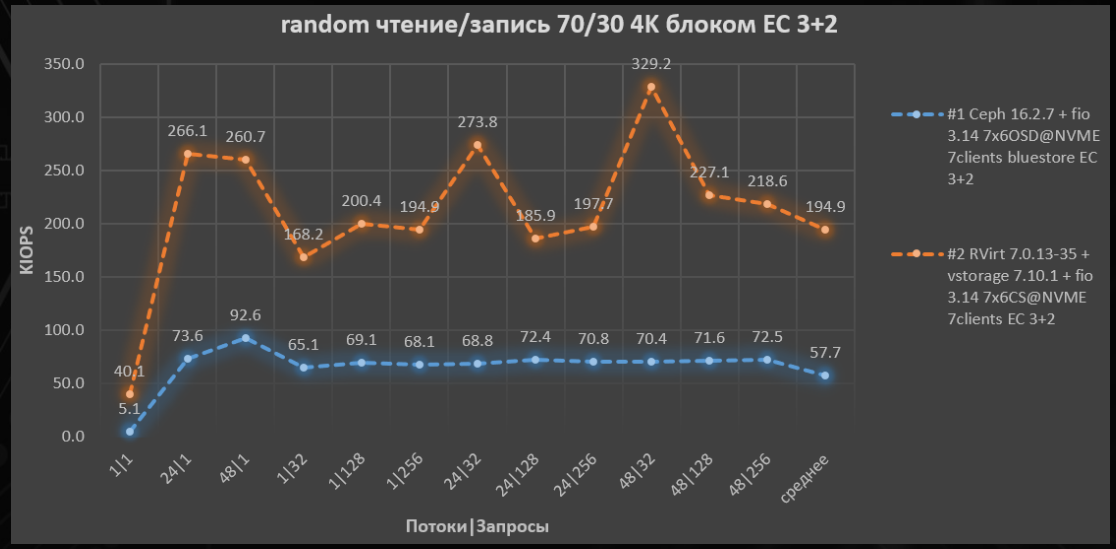
ROS ПЛАТФОРМА

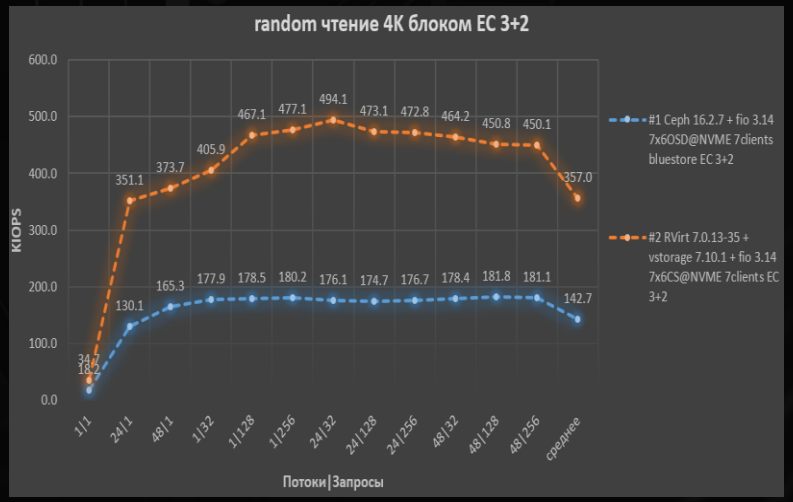
Сравнение Ceph16.2.7 с Р-хранилищем 7.0.13-35
Репликация 3

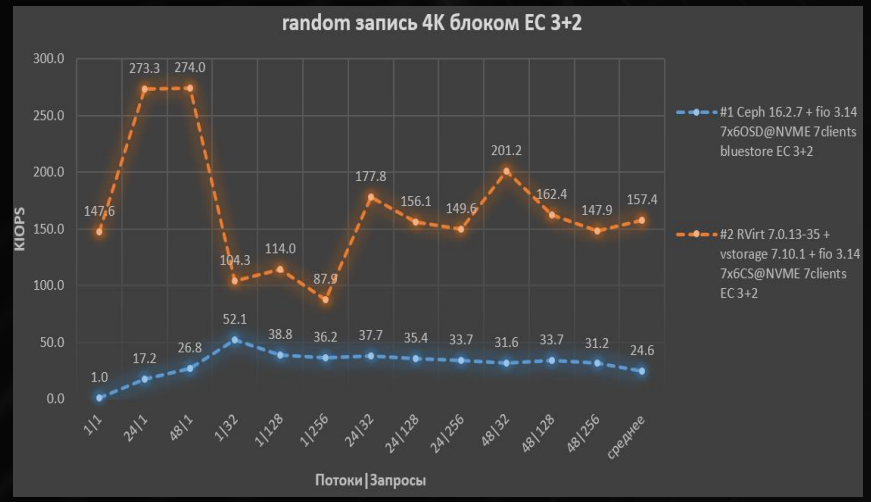


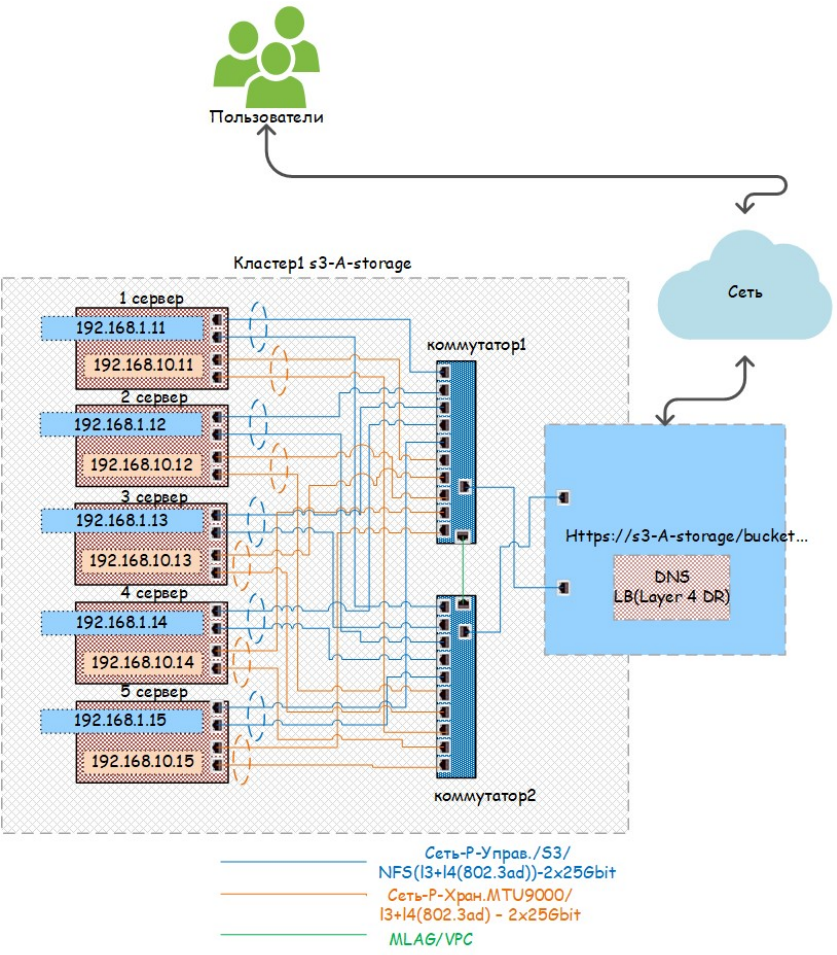
ROS ПЛАТФОРМА

Сравнение Ceph16.2.7 с Р-хранилищем 7.0.13-35
Помехоустойчивое кодирование(EC)3+2

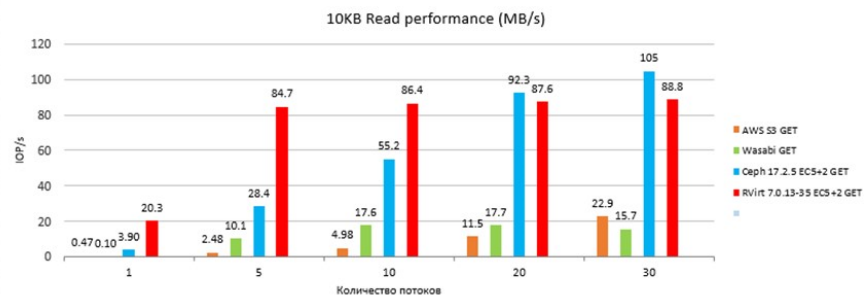


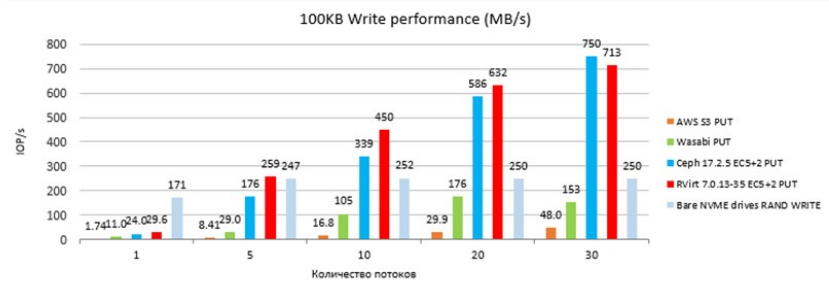


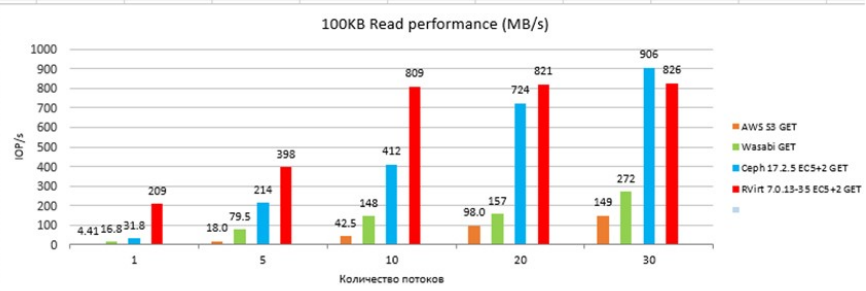


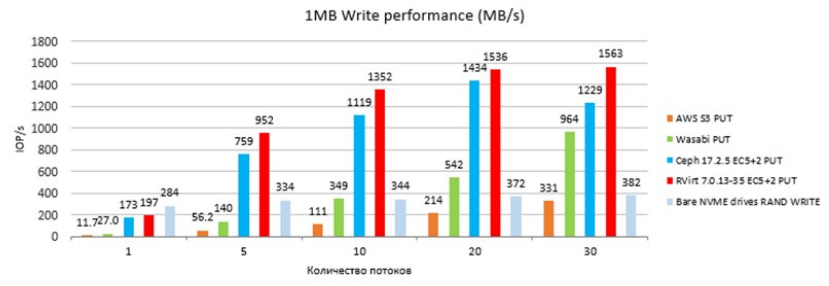


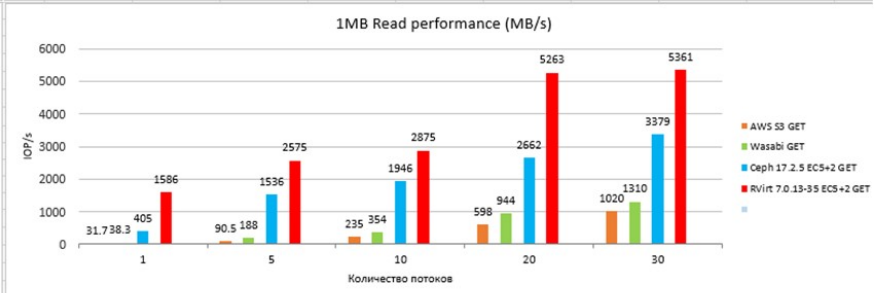
7 серверов, по 2 процессора (48 ядер),
128ГБ память,
SSD M.2: 500GB WD Black SN750 (WDS500G3X0C-00SJG0) - система,
SSD U.2: 6 x 960GB Ultrastar DC SN630 (WUS3BA196C7P3E3) - хранилище,
Сеть: 2 x 25Gbps Mellanox MT27710 ConnectX-4 Lx

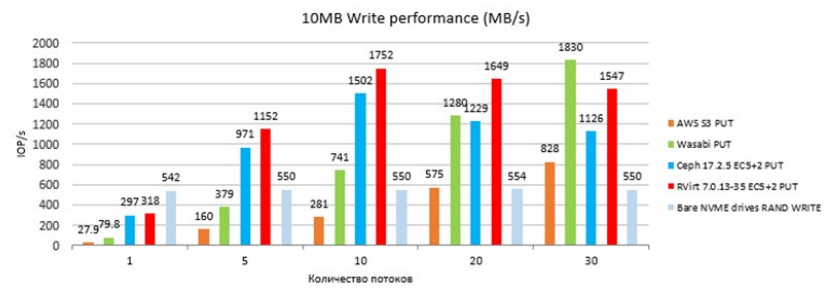


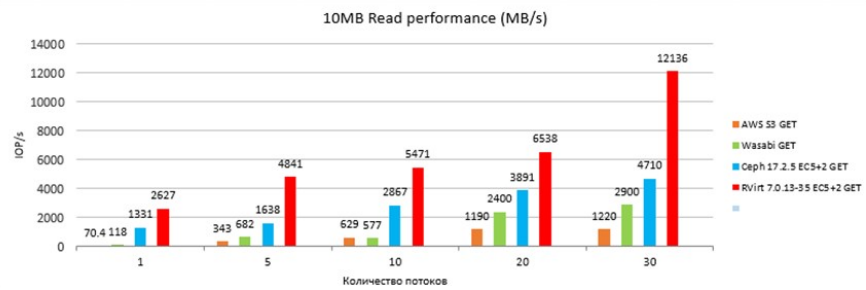












ROS
ПЛАТФОРМА

Росплатформа

- Передовой мировой опыт
- С экономией по стоимости
- С защитой от санкций

ООО «Р-Платформа»

info@rosplatforma.ru

8 (800) 700 74 60

rosplatforma.ru