

УТВЕРЖДЕН

RU.БИГЕ.04.13.001 13 01-ЛУ

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
«BMSTU STORAGE»

Описание программы

RU.БИГЕ.04.13.001 13 01

Листов 50

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2021 г.

Аннотация

В настоящем документе содержатся сведения о Программном обеспечении «BMSTU STORAGE» RU БИГЕ.04.13.001-01 (далее — «ПО»), предназначенному для передачи и хранения информации с применением СХД, его функциональном назначении, логической структуре и программно-аппаратных требованиях к рабочим станциям, на которые устанавливаются компоненты изделия.

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

Термины, определения и сокращения

<i>Ethernet</i>	- Семейство технологий пакетной передачи данных для компьютерных сетей.
<i>FC (Fibre Channel)</i>	- Семейство протоколов для высокоскоростной передачи данных.
<i>HDD (Hard Disk Drive)</i>	- Накопитель на жестких магнитных дисках, устройство хранения информации произвольного доступа, основанное на принципе магнитной записи.
<i>HTML (HyperText Markup Language)</i>	- Стандартный язык разметки документов в сети Интернет.
<i>HTTP (HyperText Transfer Protocol)</i>	- Протокол прикладного уровня передачи данных.
<i>IP-адрес (Internet Protocol Address)</i>	- Уникальный сетевой адрес узла в компьютерной сети, построенной по протоколу IP.
<i>iSCSI (Internet Small Computer System Interface)</i>	- Протокол, который базируется на TCP/IP и разработан для установления взаимодействия и управления системами хранения данных, серверами и клиентами.
<i>JavaScript</i>	- Прототипно-ориентированный сценарный язык программирования.
<i>NFS (Network file system)</i>	- Протокол сетевого доступа к файловым системам.
<i>RAID (redundant array of independent disks)</i>	- Избыточный массив независимых дисков— технология виртуализации данных, которая объединяет несколько дисков в логический элемент для избыточности и повышения производительности.
<i>RPC (Remote Procedure Call)</i>	- Класс технологий, позволяющих компьютерным программам вызывать функции или процедуры в другом адресном пространстве (как правило, на удалённых компьютерах).

Изм.	Подп.	Дата

<i>SAS интерфейс (Serial Attached SCSI)</i>	-	Последовательный интерфейс подключения жестких дисков, который сочетает преимущества интерфейсов SCSI (глубокая сортировка очереди команд, хорошая масштабируемость, высокая помехозащищённость, большая максимальная длина кабелей) и Serial ATA (тонкие, гибкие, дешёвые кабели, возможность горячего подключения, топология типа "точка-точка", позволяющая достигать большей производительности в сложных конфигурациях) с новыми уникальными возможностями.
<i>SATA (Serial ATA)</i>	-	Последовательный интерфейс обмена данными с накопителями информации.
<i>SMB (Server Message Block)</i>	-	Сетевой протокол прикладного уровня для удалённого доступа к файлам, принтерам и другим сетевым ресурсам, а также для межпроцессного взаимодействия.
<i>SMTP (Simple Mail Transfer Protocol)</i>	-	Сетевой протокол, предназначенный для передачи электронной почты в сетях TCP/IP.
<i>SSD (Solid-State Drive)</i>	-	Твердотельный накопитель, компьютерное немеханическое запоминающее устройство на основе микросхем памяти.
<i>TCP (Transmission Control Protocol)</i>	-	Один из основных протоколов передачи данных Интернета, предназначенный для управления передачей данных в сетях и подсетях TCP/IP, выполняет функции протокола транспортного уровня в стеке протоколов IP.
<i>TCP/IP</i>	-	Набор сетевых протоколов передачи данных, используемых в сетях, включая сеть Интернет.
<i>URI (Uniform Resource Identifier)</i>	-	Унифицированный (единообразный) идентификатор ресурса.
<i>V8</i>	-	Высокопроизводительный движок JavaScript с открытым программным кодом

Изм.	Подп.	Дата

<i>WWN (World Wide Name)</i>	- Уникальный идентификатор, который определяет конкретное целевое устройство Fibre Channel или Serial Attached SCSI (SAS). Каждый WWN представляет собой 8-байтное число.
<i>XDR (eXternal Data Representation)</i>	- Международный стандарт передачи данных в Интернете, используемый в различных RFC для описания типов.
<i>ОЗУ</i>	- Оперативное запоминающее устройство, энергозависимая часть системы компьютерной памяти.
<i>ОС</i>	- Операционная система.
<i>ПЭВМ</i>	- Персональная электронно-вычислительная машина.

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

Содержание

1	Общие сведения.....	9
1.1	Обозначение и наименование программы	9
1.2	Программное обеспечение, необходимое для функционирования программы	9
1.3	Языки программирования, на которых написана программа	9
2	Функциональное назначение	12
2.1	Назначение программы.....	12
2.2	Возможности программы	12
2.2.1	Модуль «Файловая система»	13
2.2.2	Модуль «Сервер удаленного вызова процедур»	14
2.2.2.1	Блок управления дисковыми накопителями	14
2.2.2.2	Блок мониторинга	14
2.2.2.3	Блок резервного копирования.....	15
2.2.2.4	Блок обработки команд	15
2.2.3	Модуль «Web-сервер»	17
2.2.4	Описания функциональных возможностей, реализуемых компонентами ПО	17
2.3	Ограничения, накладываемые на область применения программы....	20
3	Описание логической структуры.....	21
3.1	Используемые методы	21
3.2	Алгоритм программы	21
3.2.1	Модуль «Web-сервер»	21
3.2.2	Модуль «Сервер удаленного вызова процедур»	22
3.2.3	Модуль «Файловая система»	23
3.2.4	Описание реализации заявленных функциональных возможностей защиты, на уровне файлов и функциональных объектов.....	29
4	Используемые технические средства.....	30

Изм.	Подп.	Дата

4.1	Требования к аппаратным средствам	30
4.2	Требования к программным средствам	30
5	Вызов и загрузка	31
5.1	Способы вызова программы	31
5.2	Входные точки в программу	31
6	Входные данные	32
6.1	Взаимодействие с Администратором ПО	32
6.1.1	Взаимодействие по протоколу HTTP	32
6.1.1.1	Описание протокола	32
6.1.2	Описание взаимодействия	34
6.1.3	Команды управления	34
6.2	Взаимодействие с Пользователем	36
6.2.1	Протокол Fibre Channel	36
6.2.1.1	Описание протокола	36
6.2.1.2	Описание взаимодействия	37
6.2.2	Протокол iSCSI	40
6.2.2.1	Описание протокола	40
6.2.2.2	Описание взаимодействия	41
6.2.3	Протокол NFS	42
6.2.3.1	Описание протокола	42
6.2.3.2	Описание взаимодействия	43
6.2.4	Протокол SMB	43
6.2.4.1	Описание протокола	43
6.2.4.2	Описание взаимодействия	44
7	Выходные данные	45
7.1	Отображение общей информации о системе	45
7.1.1	Состояние дисков	45
7.1.2	Отображение информации о массиве	46

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

7.1.3 Отображение информации со всех служб.....	46
7.1.4 Отображение информации о состоянии контроллеров	47
7.2 Оповещения о статусе работы.....	47
7.3 Системные записи	48
7.4 Команды предоставления доступа к виртуальным дискам для протоколов Fibre Channel, iSCSI, NFS, SMB	49

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

1 Общие сведения

1.1 Обозначение и наименование программы

Наименование программы – «Программное обеспечение управления системой хранения данных» (далее — «ПО»).

Обозначение программы: RU БИГЕ.04.13.001-01.

1.2 Программное обеспечение, необходимое для функционирования программы

Для реализации функциональности отдельных программных модулей, ПО использует в своем составе программное обеспечение с открытым кодом от стороннего производителя:

- 1) SCST - пакет драйверов оптических устройств и таргет система iSCSI.
- 2) kmod-zfs-4.15.3-x-generic_0.8.6-1_amd64.deb – компонент файловой системы ZFS.
- 3) kmod-zfs-devel_0.8.6-1_amd64.deb – компонент файловой системы ZFS.
- 4) kmod-zfs-devel-4.15.3-x-generic_0.8.6-1_amd64.deb – компонент файловой системы ZFS.
- 5) libnvpair1_0.8.6-1_amd64.deb - библиотека api файловой системы ZFS.
- 6) libuutil1_0.8.6-1_amd64.deb 1 - предоставляет функционал совместимости для zfs on Linux.
- 7) libzfs2_0.8.6-1_amd64.deb - библиотека api файловой системы ZFS.
- 8) libzfs2-devel_0.8.6-1_amd64.deb - библиотека файловой системы ZFS.

Изм.	Подп.	Дата

- 9) `libzpool2_0.8.6-1_amd64.deb` - библиотека файловой системы ZFS.
- 10) `lsscsi_master_0.30.deb` - компонент использующийся для получения соответствия SAS адреса и `wwn`.
- 11) `node_v6.11.5.deb` - компонент предоставляющий среду выполнения JavaScript, построенный на JavaScript-движке.
- 12) `python-argcomplete_1.8.1-1.baum.deb` - компонент, использующийся в `baum-cli` и `baum-manager` для автоматического завершения ввода пользователем команды.
- 13) `scst_3.4.0.deb` - компонент отвечающий за предоставление клиентам доступ к томам СХД в виде LUN.
- 14) `smp_utils_0.98.deb` - компонент, служащий для управления Serial Attached SCSI (SMP).

1.3 Языки программирования, на которых написана программа

В качестве среды исполнения программных модулей, компонентов ПО используется 64-битная операционная система Astra Linux SE 1.6 (модуль Linux-платформы – заимствованный модуль). Astra Linux SE 1.6 характеризуются поддержкой широкого спектра аппаратных платформ и соответствующих драйверов различных версий, что позволит в дальнейшем быстро адаптировать ПО к новым функциональным требованиям, возникающим вследствие изменения, модификации аппаратной базы.

Программа имеет модульную структуру и состоит из трех модулей (модуль «Файловая система», модуль «Сервер удаленного вызова процедур» и модуль «Web-сервер»), каждый из которых включает компоненты, выполняющие законченный набор функций.

Модуль «Web-сервер» разработан на базе Node.js. Node.js — программная платформа, основанная на программе управления веб-сервером

Изм.	Подп.	Дата

V8, конвертирующая JavaScript из узкоспециализированного языка в язык общего назначения (машинный код). Модуль реализован средствами языка программирования Си и JavaScript.

Модуль «Сервер удаленного вызова процедур» (BAUM RPC Server (BRS)) - это многопоточный TCP сервер, реализован средствами языка программирования Си.

Модуль «Файловая система» реализован средствами языка программирования Си.

ПО написано на языках Си и JavaScript.

В качестве среды разработки для компонентов ПО использовались текстовые редакторы vim, eclipse, nano и emacs; архиватор – tar.

В качестве компиляторов – компилятор языка Си – gcc v. 6.3, утилиты сборки – GNU Make 4.1.

Утилита сборки и установки файлов программы – dpkg.

Линковщик: GNU ld (GNU Binutils for Ubuntu) 2.28.

Пакет утилит сборки: GNU Binutils 2.28.0.

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

2 Функциональное назначение

2.1 Назначение программы

ПО предназначено для организации хранилищ баз данных, серверов приложений, серверов электронной почты, файловых серверов, веб-служб, распределенных приложений, резервного копирования данных и т.д.

2.2 Возможности программы

Задачи, решаемые ПО, позволяют выделить три взаимосвязанных функциональных модуля:

- 1) Модуль «Файловая система»;
- 2) Модуль «Сервер удаленного вызова процедур»:
 - Блок управления дисковыми накопителями;
 - Блок мониторинга;
 - Блок резервного копирования;
 - Блок обработки команд;
- 3) Модуль «Web-сервер».

Общая схема работы ПО представлена ниже (Рисунок 1).

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

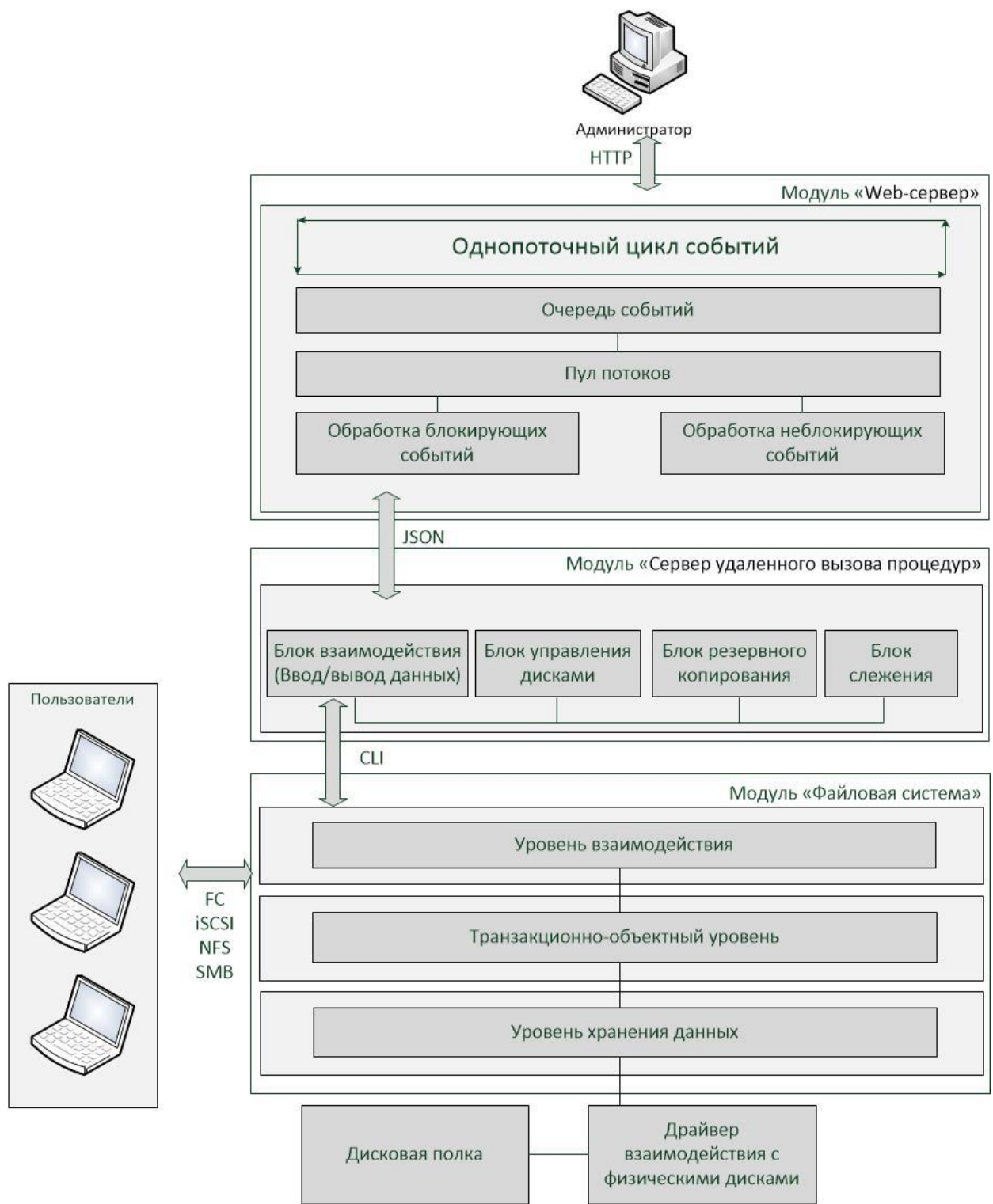


Рисунок 1. Схема работы ПО

2.2.1 Модуль «Файловая система»

Модуль «Файловая система» решает следующие задачи:

Изм.	Подп.	Дата

- возможность «горячей» замены одного или нескольких дисков (последовательно один за другим), входящих в массив, без потери данных и доступа к ним;
- обязательное одноуровневое кэширование операций чтения;
- опциональное двухуровневое кэширование операций чтения данных из массива;
- зеркалирование кэша на запись;
- сквозной контроль целостности данных, хранимых на ресурсах СХД;
- защита данных от повреждений, вызванных нескорректированными ошибками записи на носители информации.

2.2.2 Модуль «Сервер удаленного вызова процедур»

2.2.2.1 Блок управления дисковыми накопителями

Блок управления дисковыми накопителями решает следующие задачи:

- конфигурирование параметров работы устройств ввода/вывода;
- автоматическое определение количества дисков, подключенных к ПО в данный момент времени, а также их типов и характеристик;
- автоматическую блокировку работы ПО с дисками, не прошедшими «контроль»;
- автоматическое, без остановки сервиса, обнаружение и ввод в эксплуатацию новых дисковых устройств;
- совместное использование дисков SSD, HDD с интерфейсом SAS.

2.2.2.2 Блок мониторинга

Блок мониторинга решает следующие задачи:

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

- логирование действий администратора ПО и пользователей ресурсов ПО, предоставление накопленных данных для аудита администратору в сокращенном виде или в виде полного архива;
- отображение текущих параметров состояния ПО на удаленном рабочем месте администратора;
- мониторинг событий в кластере высокой доступности.

2.2.2.3 Блок резервного копирования

- резервное копирование (снэпшоты).

2.2.2.4 Блок обработки команд

Блок обработки команд решает следующие задачи:

- взаимодействие с web-интерфейсом управления;
- отображение общей информации о состоянии ПО;
- отказоустойчивое хранение конфигураций доступа пользователей к ресурсам ПО;
- настройка параметров доставки сообщений электронной почты;
- настройка системного времени;
- управление состоянием всех служб доступа через web-интерфейс управления;
- настройка параметров сетевых подключений контроллеров;
- создание массивов с различными уровнями дисковой отказоустойчивости (RAID): 0, 1, 5, 6;
- автоматическая балансировка нагрузки в кластере высокой доступности при создании массива;

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

- подключение к массивам запасных дисков, которые ПО введет в эксплуатацию автоматически при обнаружении отказа одного или нескольких из основных устройств;
- подключение к массиву высокоскоростных твердотельных дисковых накопителей в качестве кэша 2-го уровня;
- подключение к массиву разделов двух высокоскоростных твердотельных дисковых накопителей (SSD диск) в отказоустойчивой конфигурации RAID 1 в качестве кэша записи;
- дробление кэша на запись;
- включение/отключение подсветки диска по команде администратора через web-интерфейс управления;
- расширение массивов без потери данных и доступа к ним существующих массивов с уровнями дисковой отказоустойчивости RAID 1, RAID 5 и RAID 6 (создание массивов RAID 10, RAID 50 и RAID 60);
- перемещение массивов с одного контроллера на другой с целью проведения сервисных работ на ПО;
- удаление массивов с различными уровнями дисковой отказоустойчивости (RAID): 0, 1, 10, 5, 6, 50, 60;
- создание виртуальных дисков с последующим предоставлением доступа по протоколам Fibre Channel, iSCSI, NFS, SMB;
- настройка и изменение параметров разграничения доступа по протоколу Fibre Channel;
- настройка и изменение параметров разграничения доступа по протоколу iSCSI;
- настройка и изменение параметров разграничения доступа по протоколу NFS;

Изм.	Подп.	Дата

- настройка и изменение параметров разграничения доступа по протоколу SMB;
- настройка и выполнение графика резервного копирования индивидуально для каждого виртуального диска;
- настройка размера блока виртуального диска для блочных протоколов доступа при создании;
- увеличение размера существующего виртуального диска без потери данных или доступа к ним;
- перемещение виртуального диска из массива в массив в пределах одного контроллера с сохранением настроек доступа к нему;
- удаление виртуальных дисков;
- предоставление системных записей;
- выключение/перезагрузка системы;
- обработка событий в кластере высокой доступности.

2.2.3 Модуль «Web-сервер»

Модуль «Web-сервер» решает следующие задачи:

- прием команд от администратора, передача их RPC-серверу, обработка результата выполнения.

2.2.4 Описания функциональных возможностей, реализуемых компонентами ПО

Наименование функциональной возможности	Компоненты, реализующие функциональную возможность
Возможность «горячей» замены одного или нескольких дисков (последовательно один за другим), входящих в массив, без потери данных и доступа к ним	uds-bpool-service, baum-web-ui
Обязательное одноуровневое кэширование операций чтения	uds-bpool-service

Изм.	Подп.	Дата

Наименование функциональной возможности	Компоненты, реализующие функциональную возможность
Опциональное двухуровневое кэширование операций чтения данных из массива	uds-bpool-service
Зеркалирование кэша на запись	uds-bpool-service
Сквозной контроль целостности данных, хранимых на ресурсах СХД	uds-bpool-service
Защита данных от повреждений, вызванных нескорректированными ошибками записи на носителе информации	uds-bpool-service
Конфигурирование параметров работы устройств ввода/вывода	baum-web-ui
Автоматическое определение количества дисков, подключенных к ПО в данный момент времени, а также их типов и характеристик	uds-disk-manager
Автоматическую блокировку работы ПО с дисками, не прошедшими «контроль»	uds-disk-manager
Автоматическое, без остановки сервиса, обнаружение и ввод в эксплуатацию новых дисковых устройств	uds-disk-manager
Совместное использование дисков SSD, HDD с интерфейсом SAS и SATA	uds-disk-manager
Логирование действий администратора ПО и пользователей ресурсов ПО, предоставление накопленных данных для аудита администратору в сокращенном виде или в виде полного архива	uds-log-service, baum-web-ui
Отображение текущих параметров состояния ПО на удаленном рабочем месте администратора	uds-hwmon-service, baum-web-ui
Мониторинг событий в кластере высокой доступности	uds-snmp-service, uds-snmp-ext, baum-web-ui
Резервное копирование (снэпшоты)	uds-bpool-service
Взаимодействие с web-интерфейсом управления	baum-web-ui
Отображение общей информации о состоянии ПО	uds-hwmon-service, baum-web-ui
Отказоустойчивое хранение конфигураций доступа пользователей к ресурсам ПО	uds-ac-service, uds-ck-service
Настройка параметров доставки сообщений электронной почты	baum-web-ui, dma
Настройка системного времени	baum-web-ui, uds-time-service
Управление состоянием всех служб доступа через web-интерфейс управления	baum-web-ui
Настройка параметров сетевых подключений контроллеров	uds-bncd-service, baum-web-ui
Создание массивов с различными уровнями дисковой отказоустойчивости (RAID): 0, 1, 5, 6	uds-bpool-service, uds-pool-ext-service,

Изм.	Подп.	Дата

Наименование функциональной возможности	Компоненты, реализующие функциональную возможность
	baum-web-ui
Автоматическая балансировка нагрузки в кластере высокой доступности при создании массива	uds-bpool-service
Подключение к массивам запасных дисков, которые ПО введет в эксплуатацию автоматически при обнаружении отказа одного или нескольких из основных устройств	uds-bpool-service
Подключение к массиву высокоскоростных твердотельных дисковых накопителей в качестве кэша 2-го уровня	uds-bpool-service
Подключение к массиву разделов двух высокоскоростных твердотельных дисковых накопителей (SSD диск) в отказоустойчивой конфигурации RAID 1 в качестве кэша записи	uds-bpool-service
Дробление кэша на запись	uds-bpool-service
Включение/отключение подсветки диска по команде администратора через web-интерфейс управления	uds-disk-manager, baum-web-ui
Расширение массивов без потери данных и доступа к ним существующих массивов с уровнями дисковой отказоустойчивости RAID 1, RAID 5 и RAID 6 (создание массивов RAID 10, RAID 50 и RAID 60)	uds-bpool-service
Перемещение массивов с одного контроллера на другой с целью проведения сервисных работ на ПО	uds-bpool-service, baum-web-ui
Удаление массивов с различными уровнями дисковой отказоустойчивости (RAID): 0, 1, 10, 5, 6, 50, 60	uds-bpool-service
Создание виртуальных дисков с последующим предоставлением доступа по протоколам Fibre Channel, iSCSI, NFS, SMB	uds-bpool-service, baum-web-ui
Настройка и изменение параметров разграничения доступа по протоколу Fibre Channel	uds-fc-service, baum-web-ui
Настройка и изменение параметров разграничения доступа по протоколу iSCSI	uds-fc-service, baum-web-ui
Настройка и изменение параметров разграничения доступа по протоколу NFS	uds-nfs-service, baum-web-ui
Настройка и изменение параметров разграничения доступа по протоколу SMB	uds-smb-service, baum-web-ui
Настройка и выполнение графика резервного копирования индивидуально для каждого виртуального диска	uds-bpool-service, baum-web-ui
Настройка размера блока виртуального диска для блочных протоколов доступа при создании	uds-bpool-service, baum-web-ui
Увеличение размера существующего виртуального диска без потери данных или доступа к ним	uds-bpool-service

Изм.	Подп.	Дата

Наименование функциональной возможности	Компоненты, реализующие функциональную возможность
Перемещение виртуального диска из массива в массив в пределах одного контроллера с сохранением настроек доступа к нему	uds-bpool-service
Удаление виртуальных дисков	uds-bpool-service
Предоставление системных записей	uds-log-service
Выключение/перезагрузка системы	baum-web-ui
Обработка событий в кластере высокой доступности	uds-rpc-router
Прием команд от администратора, передача их RPC-серверу, обработка результата выполнения	baum-web-ui, baum-cli, baum-manager

2.3 Ограничения, накладываемые на область применения программы

ПО функционирует на платформе ОС Astra Linux SE 1.6. Чтобы обеспечить управление техническими средствами СХД (процессом приема, хранения и передачи данных) компоненты параллельной файловой системы хранения данных, должны функционировать на платформе Linux.

Изм.	Подп.	Дата

3 Описание логической структуры

3.1 Используемые методы

Для решения поставленной задачи при разработке ПО были учтены современные тенденции использования расширения команд стандартных серверных процессоров для ускорения обработки хранимых данных и использования стандартных высокопроизводительных серверных компонентов в качестве технических средств, что обеспечивает автоматическое повышение производительности при выпуске новых поколений технических средств.

Решение задачи сбора событий, для последующего аудита, построено на использовании компонента операционной системы – syslog. Для сбора событий вызванных действиями пользователей ресурсов, дополнительно используется модуль full_audit входящий в состав пакета samba. Для настройки параметров логирования используются компоненты ПО uds-smb-service_4.4.1 и baum-web-ui_4.4.1. Просмотр журнала событий реализован в компоненте baum-web-ui_4.4.1.

3.2 Алгоритм программы

3.2.1 Модуль «Web-сервер»

Модуль «Web-сервер» использует асинхронную событийную модель. В основе асинхронной событийной модели лежат событийный цикл и шаблон «reactor».

Событийный цикл представляет собой бесконечный цикл, который опрашивает «источники событий» (дескрипторы) на предмет появления в них «события». Опрос происходит с помощью библиотеки «синхронного» ввода/вывода, который, при этом будет являться «неблокирующим» (в системную функцию ввода/вывода передаётся флаг O_NONBLOCK).

Изм.	Подп.	Дата

Второй компонент асинхронной событийной модели – это шаблон «reactor». Шаблон состоит из небольших блоков программного кода, каждый из которых вызывается тогда, когда происходит связанное с ним событие/запрос. Сервисный обработчик затем разбирает прибывшие запросы и синхронно перенаправляет их на соответствующие обработчики запросов.

Схема работы Модуля «Web-сервер» представлена ниже (Рисунок 2).

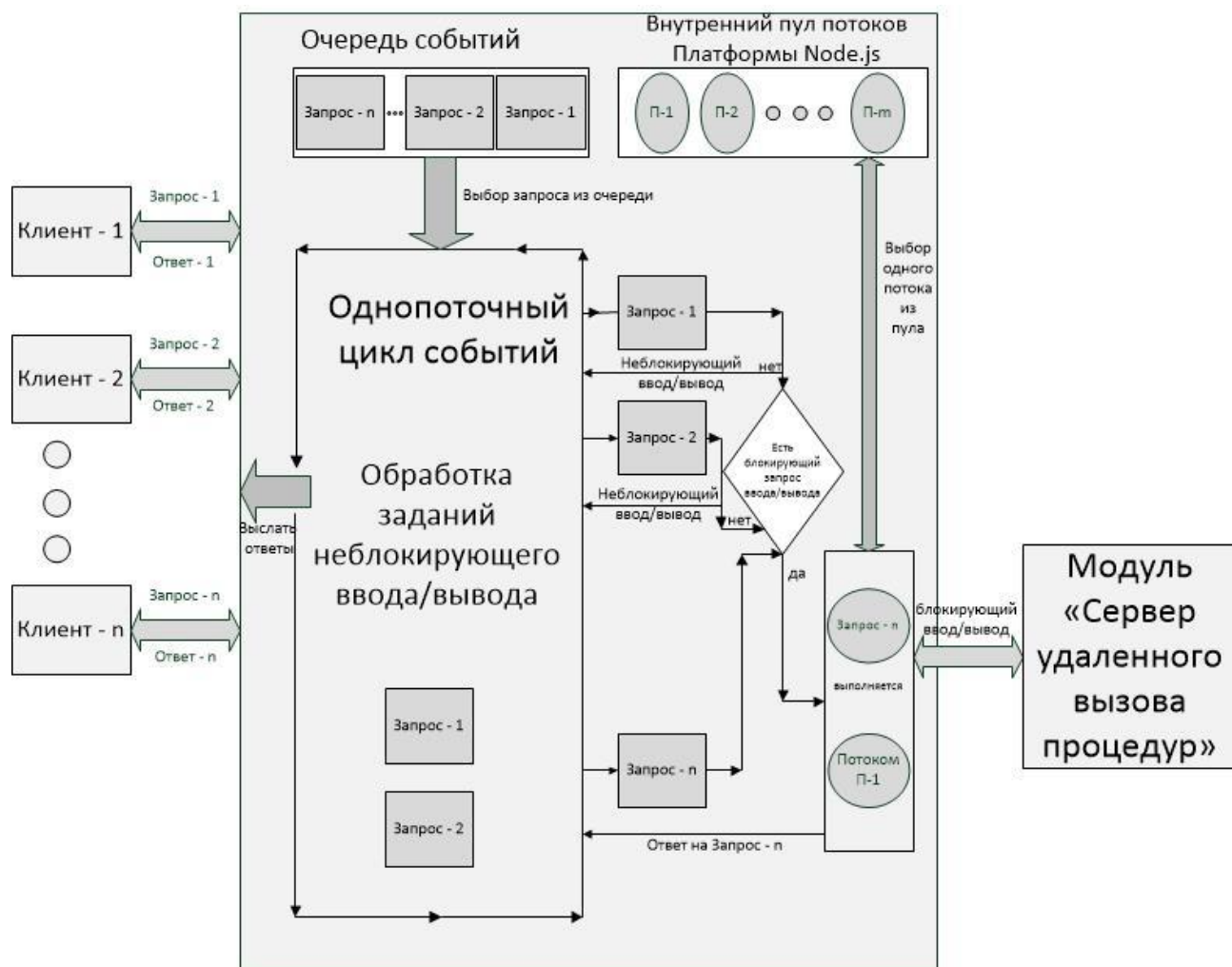


Рисунок 2 -Принцип работы Модуля «Web-сервер»

3.2.2 Модуль «Сервер удаленного вызова процедур»

Модуль «Сервер удаленного вызова процедур» - это многопоточный TCP сервер, написан на С и основывает свой событийный цикл на библиотеке

Изм.	Подп.	Дата

«libev». Библиотека libev предназначена для упрощения асинхронного неблокирующего ввода/вывода.

Событийный цикл представляет собой бесконечное ожидание появления каких-либо данных от «источника событий» (файловый дескриптор TCP соединения с Web-сервером).

Принцип работы Модуля «Сервер удаленного вызова процедур» представлен ниже (Рисунок 3).

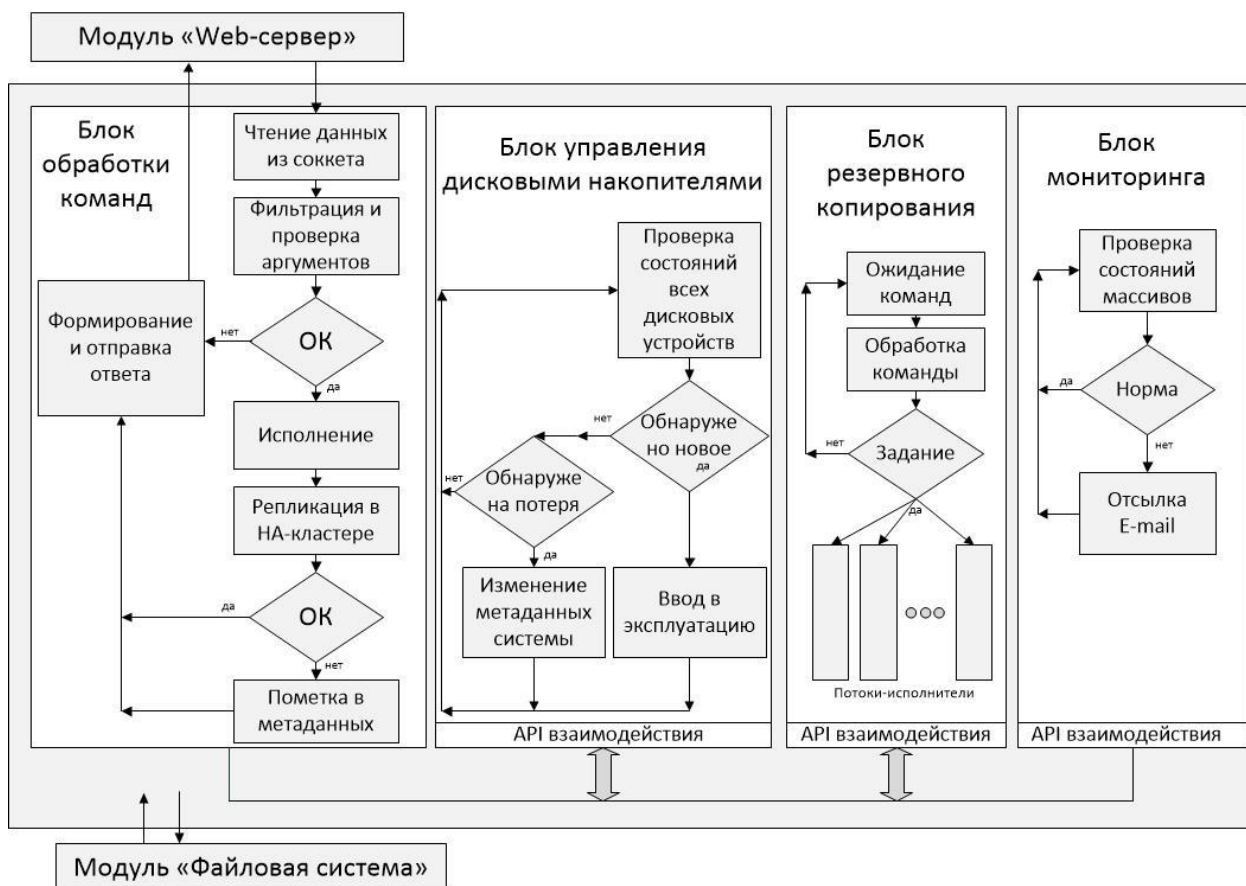


Рисунок 3 - Принцип работы Модуля «Сервер удаленного вызова процедур»

3.2.3 Модуль «Файловая система»

Модуль «Файловая система» делится на три основных уровня:

- Уровень хранения данных;

Изм.	Подп.	Дата

- Транзакционно-объектный уровень;
- Уровень взаимодействия.

Для взаимодействия Модуля «Файловая система» и физического диска используется Драйвер взаимодействия с физическими дисками.

Уровень хранения данных

Данные передаются между диском и Модулем «Файловая система» в элементах, называемых блоками. Указатель на блок, 128 байтная структура, используется для описания координат, проверки и определения свойств блоков данных на диске.

Блок виртуальных устройств отвечает за представление и создание иерархии логических устройств. Пулы Модуля «Файловая система» выполнены в виде совокупности виртуальных устройств двух типов: физических и логических. Физические - это записываемые блочные устройства (например, диски), логические — концептуальные группы физических.

Логические устройства выстраиваются в дерево, листьями которого являются физические устройства. Все пулы имеют специальный логический диск - корень этого дерева. Все прямые потомки корневого логического диска (физические или логические) называются виртуальными устройствами верхнего уровня.

Блок ввода/вывода представляет собой механизм, через который проходит все данные от физического диска и к физическому диску. Блок отвечает за перевод виртуального адреса данных в логический адрес на виртуальном устройстве, а также контрольных сумм и сжатия данных, когда это необходимо. Виртуальный адрес данных - это комбинация идентификатора виртуального устройства и номера блока на нем. Модуль «Файловая система» обеспечивает возможность хранения до 3х копий данных,

Изм.	Подп.	Дата

адресуемых этим указателем, причем каждая будет располагаться по своему виртуальному адресу данных. Данные в копиях полностью идентичны. Эта возможность введена для предохранения наиболее критичных метаданных от разрушений диска.

Блок кэширования используется, чтобы обеспечить первоочередные потребности кэширования данных. Данный блок функционирует между Блоком управления данными и Блоком виртуальных устройств, что позволяет ему оперировать блоками данных. Это в свою очередь дает возможность Модулю «Файловая система» делиться кэшированными данными между снапшотами.

Блок конфигурирования обеспечивает консолидацию Блоков Уровня хранения данных. Данный Блок включает в себя функции создания/удаления пула, конфигурационную информацию, а также осуществляет синхронизацию метаданных Уровня хранения данных.

Транзакционно-объектный уровень

Блок управления данными объединяет блоки в логические элементы, называемые объектами, которые впоследствии могут группироваться Блоком управления данными в наборы объектов. Наборы объектов используется в Модуле «Файловая система» для группировки связанных объектов в файловой системе, снапшоте или томе.

Блок наборов объектов обеспечивает механизмы описания взаимоотношений и свойств наборов объектов. Модулем «Файловая система» предоставляет возможность создавать 3 вида наборов данных:

- Файловая система: хранит и организует объекты в доступном, совместимом с POSIX формате.

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

- Снапшот: доступный только для чтения срез файловой системы или тома в некий момент времени. Файловая система или том не могут быть удалены, пока не удалены все их снапшоты.

- Том: логический том, экспортируемый файловой системой в виде блочного устройства.

Модуль «Файловая система» поддерживает несколько операций и/или конфигураций, которые вызывают сложные зависимости и взаимоотношения между наборами объектов. Задачей Блока наборов данных является управление этими связями.

Блок управления атрибутами - это блок, расположенный над Блоком управления данными. Объекты Блока управления атрибутами используются для хранения свойств набора объектов, навигации по файловой системе, хранения параметров пула и других целей.

Блок учета транзакций сохраняет информацию о транзакциях, изменяющих Модуль «Файловая система» в объеме, достаточном для их завершения или атомарного отката в случае сбоя. Они накапливаются в памяти до тех пор, пока группа транзакций Блока управления данными не зафиксирует их в стабильном пуле, или пока они не будут отправлены в журнал. Существует один Блок учета транзакций на файловую систему. На диске он состоит из трех частей: заголовка, записей логирования и блоков логирования.

Блоки хранят множество записей и объединяются в цепочки. Каждый блок логирования содержит указатель на следующий блок в цепи. Блоки могут быть разного размера. Заголовок указывает на первый блок в цепи.

Блок контроля обеспечивает безопасный, эффективный и перезапускаемый метод перемещения всех данных по Транзакционно-объектному уровню. Данный Блок сопровождает все метаданные, отслеживает

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

модифицированные блоки в течение определенного периода времени. Основной функционал Блока контроля сосредоточен на Уровне хранения данных, но при этом Блок содержит информацию о некоторых структурах Блока управления данными, чтобы обслуживать задачи для снапшотов, клонов и некоторые других наборов объектов.

Уровень взаимодействия

Блок представления файловой системы POSIX, позволяет объектам Блока управления данными выглядеть как файловая система POSIX. Блок представления файловой системы POSIX представляет файловую систему как набор объектов определенного типа, используется хорошо известный формат для объединения объектов в наборы. Все снапшоты, клоны и файловые системы реализованы как наборы данного типа.

Блок логических томов обеспечивает механизм для создания логических томов. Тома Блока логических томов экспортируются как блочные устройства. Внутри Блока логических томов тома представляются набором объектов, имеют очень простой формат — состоят всего лишь из двух объектов: свойства тома и данные.

Блок основного управления - это основной интерфейс для приложений управления, чтобы взаимодействовать с Модулем «Файловая система». Данный Блок представляет единый, объектно-ориентированный механизм для доступа и управления пулами хранения данных и файловыми системами.

Принцип работы Модуля «Файловая система» представлен ниже (Рисунок 4).

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

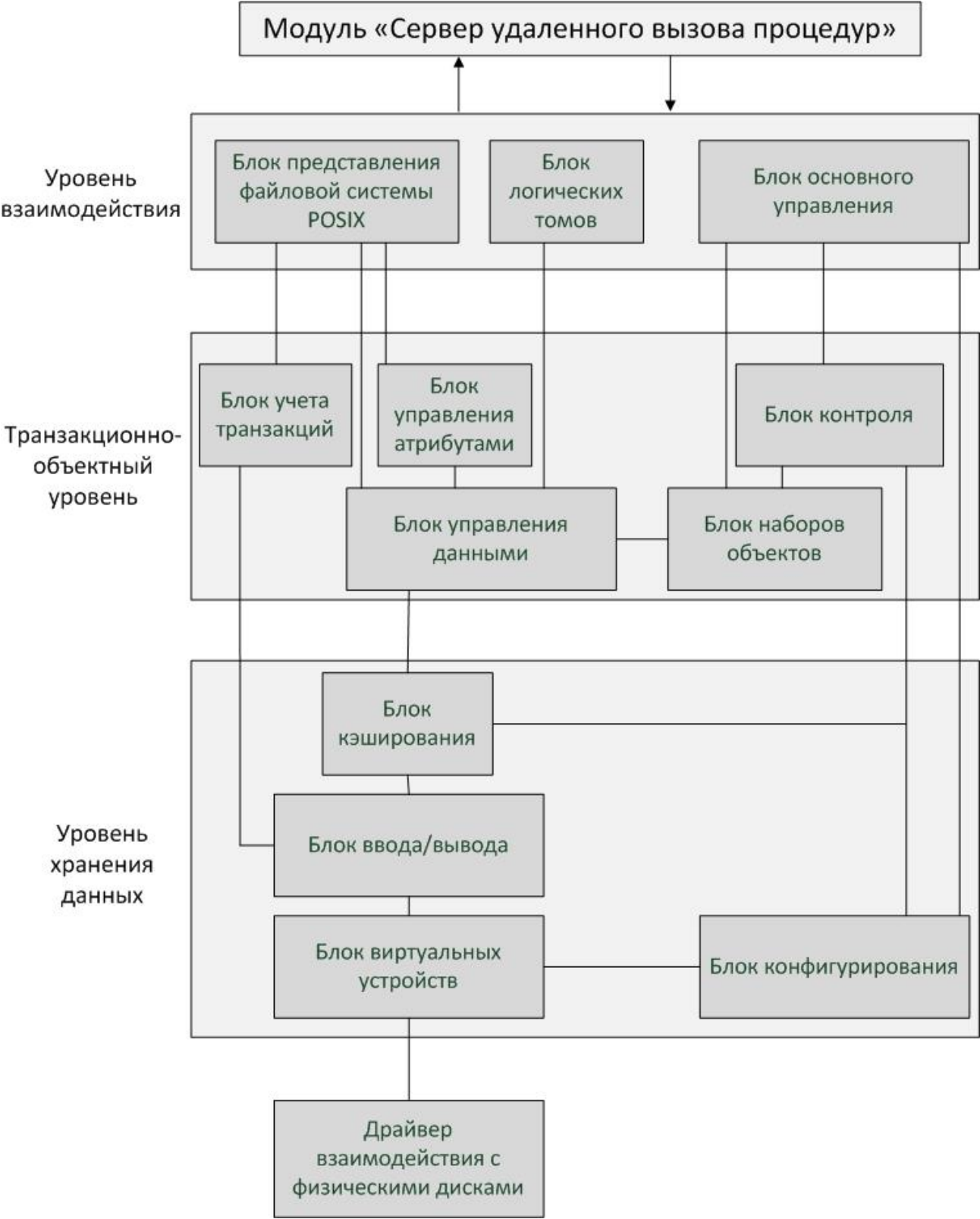


Рисунок 4 - Принцип работы Модуля «Файловая система»

Изм.	Подп.	Дата

3.2.4 Описание реализации заявленных функциональных возможностей защиты, на уровне файлов и функциональных объектов.

Наименование функциональной возможности	Реализация на уровне файлов	Реализация на уровне функциональных объектов
Логирование действий администратора ПО и пользователей ресурсов ПО, предоставление накопленных данных для аудита администратору в сокращенном виде или в виде полного архива	uds_bls_control.h uds_bls_control.c	uds_bls_check_loglevel uds_bls_register_serv_in_rsyslog uds_bls_ask_rsyslog_for_restart uds_bls_ask_for_logrotate uds_bls_arc_create
	uds_bls_rpc_mngr.h uds_bls_rpc_mngr.c uds_bls_rpc_sm_methods.h uds_bls_rpc_sm_methods.c	uds_get_log uds_bls_set_default_log_level uds_bls_check_default_log_level_in_params uds_refresh_cfg_from_params uds_update_log_level_in_mem uds_update_log_level_in_rsyslog uds_bls_update_config_in_ck uds_bls_send_cfg_changed_notify uds_bls_remote_logging_cfg_to_answ uds_bls_make_log_archive uds_bls_archive_status uds_bls_log_search uds_bls_log_list_to_response
	uds_bls_settings.h, uds_bls_settings.c	uds_bls_parse_args
	uds_bls_utils.h, uds_bls_utils.c	uds_bls_add_status_by_usage uds_bls_log_fs_filled uds_bls_tmp_fs_filled

Изм.	Подп.	Дата

4 Используемые технические средства

4.1 Требования к аппаратным средствам

- аппаратная платформа x86_64;
- количество ядер процессора не менее 4 (с частотой 2 ГГц или выше);
- не менее 16 Гб ОЗУ;
- для установки ПО необходимо не менее 1 Гб свободного места на используемом ПЭВМ носителе машинной памяти.

4.2 Требования к программным средствам

Требования к операционной системе: 64-битная операционная система Astra Linux SE 1.6.

Требования к интернет-браузеру:

- Internet Explorer версии не ниже 10;
- Mozilla Firefox версии не ниже 36;
- Google Chrome версии не ниже 41;
- Opera версии не ниже 12.16.

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

5 Вызов и загрузка

5.1 Способы вызова программы

Установка и загрузка ПО осуществляется с помощью Оптического диска (с записанным на него инсталляционным образом ПО).

Вызов программы ПО производит Администратор системы в веб-интерфейсе управления (посредством ввода IP адреса системы, на которой установлена ПО, в адресной строке веб-браузера).

Детальное описание процедур загрузки и вызова программы представлено в документе «RU БИГЕ.04.13.001 32-01. Руководство системного программиста».

5.2 Входные точки в программу

Модуль «Сервер удаленного вызова процедур» производит инициализацию компонентов системы и обеспечивает запуск всех вспомогательных нитей и процессов. Точка входа: функция **main()**, находится в файле **brs.c**.

У модуля «Файловая система» отсутствует единая точка входа. Модуль ядра загружается, выгружается как совокупность драйверов составляющих его механизмов посредством команд от компонента модуля управления каждый механизм загружается отдельно.

У модуля «Web-сервер», также отсутствует единая точка входа. Так как используется скриптовый язык, обработка которого осуществляется построчно.

Изм.	Подп.	Дата

6 Входные данные

ПО использует следующий перечень входных данных:

- 1) От Администратора ПО – по протоколу http иницируются соединение и посылаются запросы на Web-сервер node.js., для ввода команд управления.
- 2) От Пользователя ПО – команды доступа к виртуальным дискам для протоколов Fibre Channel, iSCSI, NFS, SMB.

6.1 Взаимодействие с Администратором ПО

6.1.1 Взаимодействие по протоколу HTTP

6.1.1.1 Описание протокола

Протокол передачи гипертекста – это протокол прикладного уровня передачи данных (изначально — в виде гипертекстовых документов в формате HTML, в настоящий момент используется для передачи произвольных данных). Основой HTTP является технология «клиент-сервер», то есть предполагается существование потребителей (клиентов), которые иницируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом. HTTP в настоящее время повсеместно используется во в сетях связи общего пользования и локальных вычислительных сетях.

Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (Uniform Resource Identifier) в запросе клиента. Обычно такими ресурсами являются хранящиеся на сервере файлы, но ими могут быть логические объекты или что-то абстрактное. Особенностью протокола HTTP является возможность указать в запросе и ответе способ представления одного и того же ресурса по различным параметрам: формату, кодировке, языку и т.д.

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

(В частности для этого используется HTTP-заголовок.) Именно благодаря возможности указания способа кодирования сообщения клиент и сервер могут обмениваться двоичными данными, хотя данный протокол является текстовым.

Обмен сообщениями идёт по обыкновенной схеме «запрос-ответ». Для идентификации ресурсов HTTP использует глобальные URI. В отличие от многих других протоколов, HTTP не сохраняет своего состояния. Это означает отсутствие сохранения промежуточного состояния между парами «запрос-ответ».

Каждое HTTP-сообщение состоит из трёх частей, которые передаются в указанном порядке:

- Стартовая строка — определяет тип сообщения;
 - Заголовки — характеризуют тело сообщения, параметры передачи и прочие сведения;
 - Тело сообщения — непосредственно данные сообщения.
- Обязательно должно отделяться от заголовков пустой строкой.

Заголовки и тело сообщения могут отсутствовать, но стартовая строка является обязательным элементом, так как указывает на тип запроса/ответа.

Большинство протоколов предусматривают установление TCP-сессии, в ходе которой один раз происходит авторизация, и дальнейшие действия выполняются в контексте этой авторизации. HTTP же устанавливает отдельную TCP-сессию на каждый запрос. Для поддержки авторизованного (неанонимного) доступа в HTTP используются cookies; причём такой способ авторизации позволяет сохранить сессию даже после перезагрузки клиента и сервера. HTTP перед тем, как передать сами данные, передаёт заголовок «Content-Type: тип/подтип», позволяющую клиенту однозначно определить, каким образом обрабатывать присланные данные.

Изм.	Подп.	Дата

6.1.2 Описание взаимодействия

Администратор ПО через браузер по протоколу http иницируют соединение и посылают запросы на Web-сервер node.js.

Основным объектом манипуляции в HTTP является ресурс, на который указывает URI (Uniform Resource Identifier) в запросе Администратора. По запросу Администратор получает HTML и CSS файлы, а также файлы, написанные на языке JavaScript (Рисунок 5).



Рисунок 5 - Общая схема взаимодействия с веб-сервером node.js

6.1.3 Команды управления

Администратор ПО с помощью команд управления получает доступ к основным функциям системы (перечислены в разделе 2.2 «Возможности программы»), а также возможность настройки и конфигурирования ПО.

В ПО предусмотрен ввод команд управления посредством веб-браузера (список всех доступных браузеров приведен в разделе 4.2 «Требования к программным средствам»).

Администратор вводит команды в определенном формате в веб-интерфейс управления. Обработка команд в данном случае осуществляется по технологии «клиент-сервер».

Изм.	Подп.	Дата

Основные команды управления ПО по функциональному назначению подразделяются на следующие категории:

- 1) Базовая настройка системы:
 - первичный вход в систему,
 - изменение Логина/Пароля,
 - настройка оповещений по почте,
 - настройка системного времени,
 - включение и выключение служб,
 - конфигурирование параметров работы устройств ввода/вывода,
 - изменение сетевых настроек.
- 2) Работа с массивами:
 - создание RAID-массива,
 - изменение RAID-массива,
 - добавление диска в RAID-массив (расширение),
 - миграция RAID-массива,
 - удаление RAID-массива.
- 3) Работа с виртуальными дисками:
 - создание виртуального диска,
 - модификация виртуального диска,
 - удаление виртуального диска.
- 4) Управление резервным копированием.
- 5) Управление системными записями.
- 6) Проверка конфигурации системы:
 - Проверка физических дисков,
 - Просмотр информации о массиве,

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

- Проверка сервисов,
- Проверка контроллеров.

Веб-интерфейс управления представляет собой иерархию страниц, организованных с определенным уровнем вложенности. Главная страница веб-интерфейса управления (Рисунок 6).

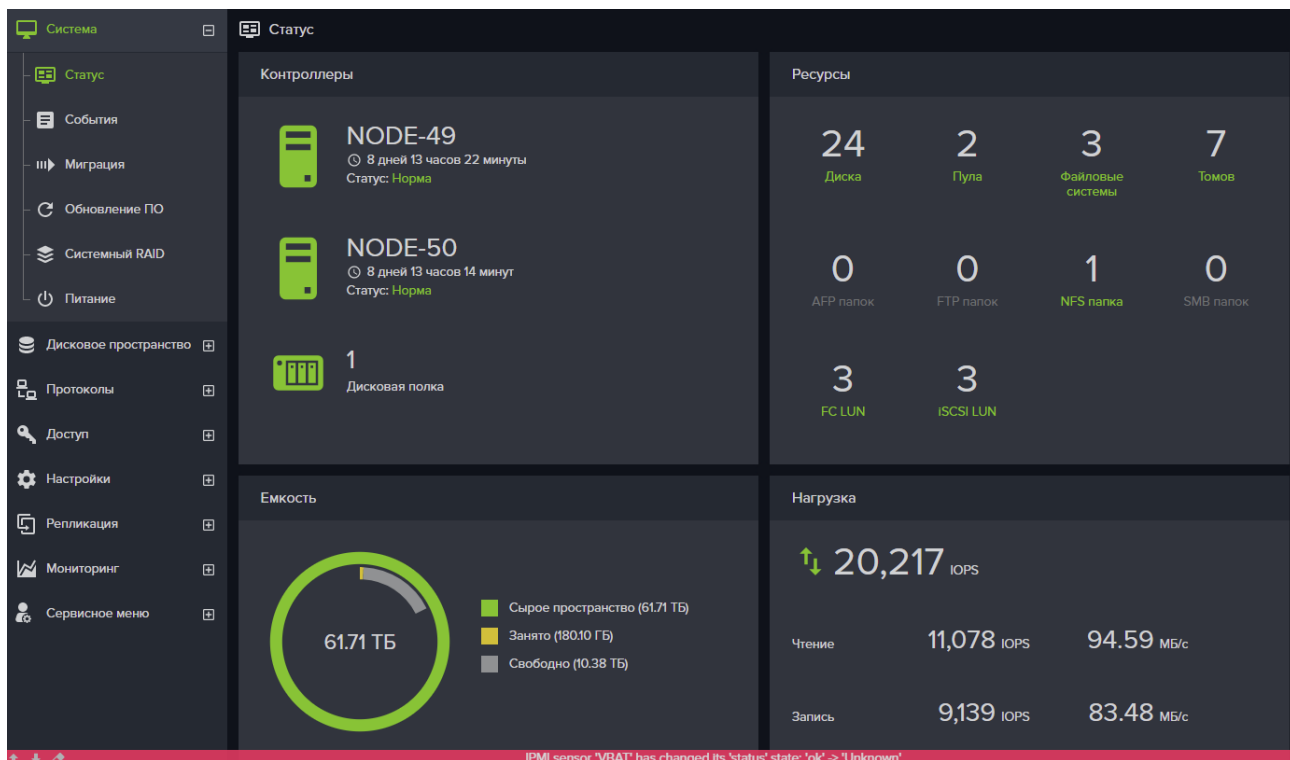


Рисунок 6. Главная страница интерфейса управления

6.2 Взаимодействие с Пользователем

6.2.1 Протокол Fibre Channel

6.2.1.1 Описание протокола

Интерфейс Fibre Channel — это технология межсистемного взаимодействия, которая объединяет в себе возможности высокоскоростного ввода-вывода и сетевого обмена данными.

В терминологии Fibre Channel устройства называются узлами (nodes). Это весьма напоминает узлы в терминологии сетей IP. Узел Fibre Channel

Изм.	Подп.	Дата

может иметь несколько портов, как и узел IP, который зачастую получает несколько адресов IP. Разница между ними в том, что порт Fibre Channel представляет собой физический элемент, а порт IP — логический. Каждый узел Fibre Channel имеет уникальное 64-разрядное имя WWN (World Wide Name), которое назначается производителем. Это напоминает уникальные адреса MAC, которые назначаются производителями сетевым адаптерам Ethernet.

Каждый порт сети хранения данных на базе кольца с разделением доступа Fibre Channel имеет 8-битовый адрес, а порт в коммутируемой связной архитектуре — 24-битовый. При подключении кольца с разделяемым доступом (arbitrated loop) к коммутатору связной архитектуры (fabric switch), коммутатор представляет 8-битовый адрес в виде 24-битового. Оба идентификатора порта назначаются динамически.

6.2.1.2 Описание взаимодействия

Все данные, передаваемые в среде Fibre Channel разбиваются на фреймы (кадры). Структура фрейма следующая:

- SoF — 4 байта — идентификатор начала фрейма.
- Header — 24 байта — заголовок. Содержит такую информацию как адрес источника и приёмника, тип фрейма (FT-0 — управляющий или FT-1 — данные), номер последовательности и порядковый номер фрейма в ней и прочая служебно-контрольная информация.
- Data — 0-2112 байт — непосредственно данные (например, SCSI-команды).
- CRC — 4 байта — контрольная сумма.
- EoF — 4 байта — идентификатор конца фрейма.

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

Промежутки между фреймами заполняются специальными «заполняющими словами».

6.2.1.3 Последовательности

Чаще всего источник стремится передать приёмнику гораздо больше информации, чем 2112 байт (максимальный объём данных одного фрейма). В этом случае информация разбивается на несколько фреймов, а набор этих фреймов называется последовательностью. Чтобы в логическую последовательность фреймов не вклинилось что-то лишнее при параллельной передаче, заголовок каждого фрейма имеет поля SEQ_ID (идентификатор последовательности) и SEQ_CNT (номер фрейма в последовательности).

6.2.1.4 Обмен

Одна или несколько последовательностей, отвечающих за какую-то одиночную операцию, называется обменом. Источник и приёмник могут иметь несколько параллельных обменов, но каждый обмен в единицу времени может содержать только одну последовательность. Пример обмена:

- инициатор запрашивает данные (последовательность 1),
- таргет возвращает данные инициатору (последовательность 2),
- затем сообщает статус (последовательность 3).

В этот набор последовательностей не может вклиниться какой-то посторонний набор фреймов. Для контроля этого процесса заголовок каждого фрейма включает поля OX_ID (Originator Exchange ID — заполняется инициатором обмена) и RX_ID (Responder Exchange ID — заполняется получателем в ответных фреймах, путём копирования значения OX_ID).

6.2.1.5 Классы обслуживания

Различные приложения предъявляют разные требования к уровню сервиса, гарантии доставки, продолжительности соединения и пропускной

Изм.	Подп.	Дата

способности. Некоторым приложениям требуется гарантированная пропускная способность в течение их работы (бэкап). Другие имеют переменную активность и не требуют постоянной гарантированной пропускной способности канала, но им нужно подтверждение в получении каждого отправленного пакета. Для удовлетворения таких потребностей и обеспечения гибкости, FC определяет 6 классов обслуживания.

6.2.1.6 Flow Control

В целях предотвращения ситуации, когда отправитель перегрузит получателя избыточным количеством фреймов так, что они начнут отбрасываться получателем, FC использует механизмы управления потоком передаваемых данных (Flow Control). Их два — Buffer-to-Buffer flow control и End-to-End flow control. Их использование регламентируется классом обслуживания. Например, класс 1 использует только механизм End-to-End, класс 3 — Buffer-to-Buffer, а класс 2 — оба эти механизма.

6.2.1.7 Buffer-to-Buffer flow control

Принцип технологии — отправка любого фрейма должна быть обеспечена наличием кредита на отправку.

Все поступающие на вход порта фреймы помещаются в специальную очередь — буферы. Количество этих буферов определяется физическими характеристиками порта. Один буфер (место в очереди) соответствует одному кредиту. Каждый порт имеет два счётчика кредитов:

- TX BB_Credit — счётчик кредитов передачи. После отправки каждого фрейма, уменьшается на 1. Если значение счётчика стало равным нулю — передача невозможна. Как только от порта-приёмника получено R_RDY, счётчик увеличивается на 1.

Изм.	Подп.	Дата

- RX BB_Credit — счётчик кредитов приёма. Как только фрейм принят и помещён в буфер, уменьшается на 1. Когда фрейм обрабатывается или пересылается дальше, счётчик увеличивается на 1, а отправителю отправляется R_RDY. Если значение счётчика падает до 0, то в принципе, приём новых фреймов должен быть прекращён.

6.2.1.8 End-to-End flow control

Реализуется счётчиком EE_Credit, который определяет максимум фреймов, которые источник может отправить приёмнику без получения подтверждения. В отличие от BB_Credit распространяется только на фреймы с данными, а обмен/учёт происходит между конечными узлами.

6.2.2 Протокол iSCSI

6.2.2.1 Описание протокола

Протокол интерфейса взаимодействия компонентов небольших компьютерных систем (Internet Small Computer System Interface, iSCSI) используется для установления взаимодействия между компонентами систем хранения данных и управления данными этих систем. Он является протоколом транспортного уровня и базируется на интерфейсе SCSI, разработанном для управления взаимодействием подключенных к общей шине разнородных по своему назначению устройств. Протокол iSCSI основан на использовании протокола SCSI и протокола транспортного уровня IP, применяющегося в корпоративных сетях Ethernet.

Протокол SCSI управляет обменом блоками данных между сервером и хранилищем данных. Протокол iSCSI работает поверх протокола TCP/IP, при этом используется механизм инкапсуляции команд SCSI в пакеты протокола IP. Физически он представлен разъемом RJ-45 для подключения в сеть

Изм.	Подп.	Дата

Ethernet. Протокол предусматривает использование одной и той же сетевой технологии как для локальной сети Ethernet, так и для сетей хранения данных.

Компонентами сети хранения данных, для которых применяется этот протокол, являются серверы, устройства хранения (диски, дисковые массивы), шлюзы, применяемые для передачи данных по этому протоколу. При этом используются обычные сетевые устройства (коммутаторы, маршрутизаторы) высокопроизводительной сети Ethernet, скорости передачи данных в которой составляют от 1 Гбит/с и 10 Гбит/с.

Технология iSCSI позволяет организовывать сети хранения данных на основе протокола IP без ограничения расстояний, благодаря чему устройства могут взаимодействовать в глобальном масштабе.

6.2.2.2 Описание взаимодействия

«Клиент» инициирует запросы на считывание или запись данных с исполнителя - «сервера», в роли которого, выступает СХД. Команды, которые выдает «клиент» и обрабатывает «сервер», помещаются в блок описания команды.

Блок описания команды (Command Descriptor Block) - это структура, с помощью которой приложение-клиент направляет команды устройству-серверу. «Сервер» выполняет команду, а окончание ее выполнения обозначается специальным сигналом. Инкапсуляция и надежная доставка CDB-транзакций между инициаторами и исполнителями через TCP/IP сеть и есть главная задача iSCSI, причем ее приходится осуществлять в нетрадиционной для SCSI, потенциально ненадежной среде IP-сетей.

Протокол iSCSI осуществляет контроль передачи блоков данных и обеспечивает подтверждение достоверности завершения операции ввода/вывода, что в свою очередь обеспечивается через одно или несколько TCP-соединений.

Изм.	Подп.	Дата

6.2.3 Протокол NFS

6.2.3.1 Описание протокола

Протокол NFS — протокол сетевого доступа к файловым системам. Протокол основан на протоколе вызова удалённых процедур и позволяет подключать (монтировать) удалённые файловые системы через сеть.

NFS предоставляет клиентам прозрачный доступ к файлам и файловой системе сервера. Протокол NFS осуществляет доступ только к тем частям файла, к которым обратился процесс, и основное достоинство его в том, что он делает этот доступ прозрачным. Это означает, что любое приложение клиента, которое может работать с локальным файлом, с таким же успехом может работать и с NFS файлом, без каких-либо модификаций самой программы.

Структура NFS включает три компонента разного уровня:

- Прикладной уровень - это вызовы удалённых процедур, которые и выполняют необходимые операции с файлами и каталогами на стороне сервера.

- Функции уровня представления выполняет протокол XDR (eXternal Data Representation), который является межплатформенным стандартом абстракции данных. Протокол XDR описывает унифицированную, каноническую форму представления данных, не зависящую от архитектуры вычислительной системы. При передаче пакетов RPC-клиент переводит локальные данные в каноническую форму, а сервер проделывает обратную операцию.

- Сервис RPC, обеспечивающий запрос удалённых процедур клиентом и их выполнение на сервере, представляет функции сеансового уровня.

Изм.	Подп.	Дата

6.2.3.2 Описание взаимодействия

NFS-клиенты получают доступ к файлам на NFS-сервере путём отправки RPC-запросов на сервер. Это может быть реализовано с использованием обычных пользовательских процессов — а именно, NFS-клиент может быть пользовательским процессом, который осуществляет конкретные RPC-вызовы на сервер, который так же может быть пользовательским процессом.

Процедура подключения сетевого ресурса средствами NFS называется «экспортированием». Клиент может запросить у сервера список представляемых им экспортируемых ресурсов. Сам сервер NFS, в отличие от, например, сервера SMB, не занимается широковещательной рассылкой списка своих экспортируемых ресурсов.

Механизм доступа к файлам через NFS прямолинеен и прозрачен. После монтирования том NFS становится частью системы конечного пользователя.

6.2.4 Протокол SMB

6.2.4.1 Описание протокола

Протокол SMB — это сетевой протокол для общего доступа к файлам, который позволяет приложениям компьютера читать и записывать файлы, а также запрашивать службы серверных программ в компьютерной сети. Протокол SMB может использоваться поверх протокола TCP/IP или других сетевых протоколов. С помощью протокола SMB приложение (или использующий его пользователь) может получать доступ к файлам и другим ресурсам удаленного сервера. Это позволяет приложениям читать, создавать и обновлять файлы на удаленном сервере. Этот протокол может также обмениваться данными с любой серверной программой, которая настроена на получение клиентских запросов SMB.

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

6.2.4.2 Описание взаимодействия

Данный протокол описывает правила взаимодействия рабочей станции с сервером. На основе SMB реализуется файловая служба. Если более подробно, то посредством SMB осуществляется:

1. Управление сессиями связи: инициализация и разрыв канала между рабочей станцией и сетевыми ресурсами файлового сервера.

2. Файловый доступ. Результатом обращения рабочей станции к файловому серверу может стать создание и удаление каталогов, создание, открытие и закрытие файлов, чтение и запись в файлы, переименование и удаление файлов, поиск файлов, получение и установка файловых атрибутов, блокировка записей.

3. Сервис сообщений. SMB поддерживает простую передачу сообщений. Можно послать простое сообщение, широковещательное сообщение, послать начало блока сообщений, переслать имя пользователя или получить имя машины.

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

7 Выходные данные

ПО в своей работе создает следующие выходные данные:

- Отображение общей информации о системе.
- Оповещения о статусе работы.
- Системные записи.
- Команды предоставления доступа к виртуальным дискам для протоколов Fibre Channel, iSCSI, NFS, SMB.

7.1 Отображение общей информации о системе

Проверка состояния системы позволяет Администратору получить общую информацию о компонентах системы, а именно:

- Состояние дисков;
- Состояние всех служб;
- Время работы контроллеров.

7.2 Состояние дисков

В области «Диски» отображается количество дисков и информация о каждом диске (Рисунок 7).

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

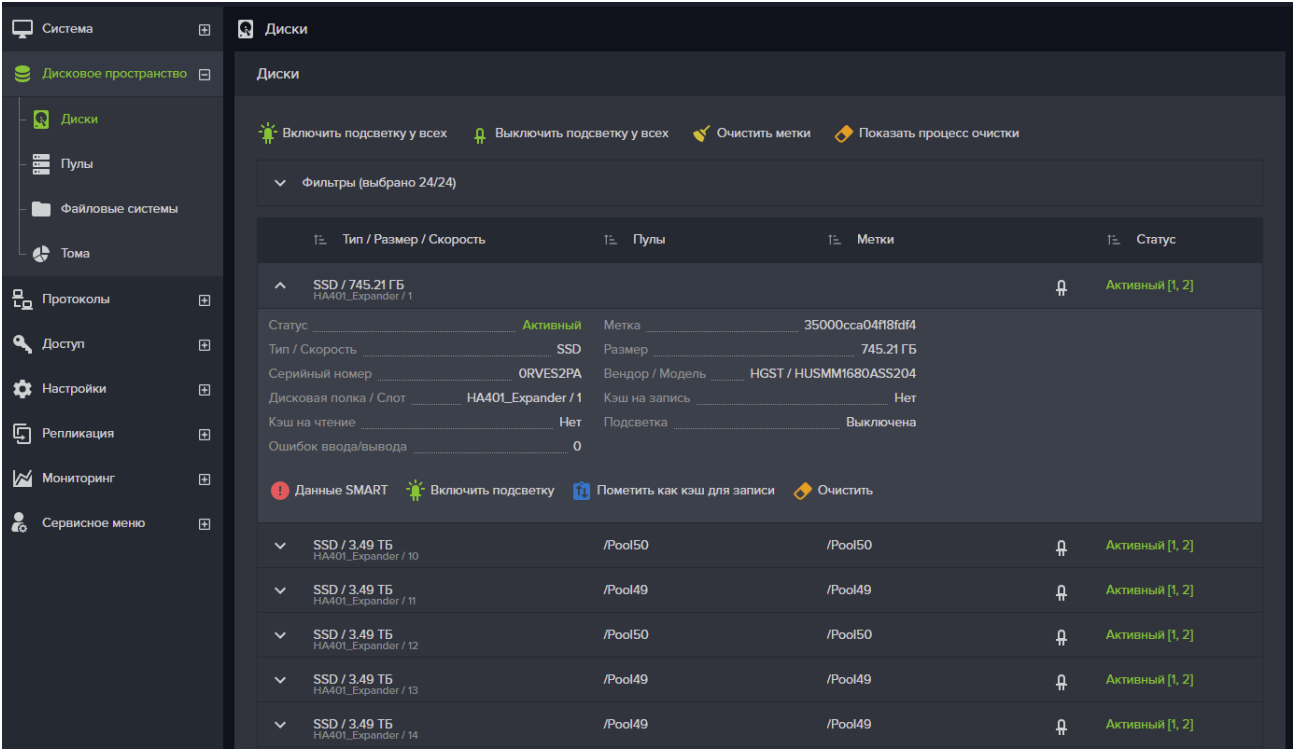


Рисунок 7 - Информация о состоянии дисков

7.2.1 Отображение информации о массиве

Информация о массиве содержит в себе:

- общая информация о массиве (его имя, размер, статус, тип RAID и т.д.),
- таблица физических дисков, которые включены в данный массив (указывается их идентификатор, размер, тип (SSD/HDD), статус),
- таблицы виртуальных ресурсов, которые выделены на данном массиве.

7.2.2 Отображение информации со всех служб

В правой части основной области расположена таблица с наименованием всех служб, доступных в системе, и их статусом и количеством ресурсов по каждой службе (Рисунок 6).

Изм.	Подп.	Дата

7.2.3 Отображение информации о состоянии контроллеров

В области «Контроллеры» расположена таблица, в которой указаны название каждого контроллера и время его работы (Рисунок 6).

7.3 Оповещения о статусе работы

В ПО реализовано оповещение при помощи отправки письма посредством электронной почты Администратору ПО.

Для настройки сервиса оповещения посредством электронной почты в веб-интерфейсе доступны следующие параметры (Рисунок 8):

- адрес получателя сообщений;
- параметры почтового сервера:
 - адрес отправителя сообщений;
 - адрес SMTP сервера;
 - порт SMTP сервера;
 - шифрование сообщения;
 - SMTP аутентификация;
 - имя пользователя;
 - пароль пользователя.

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

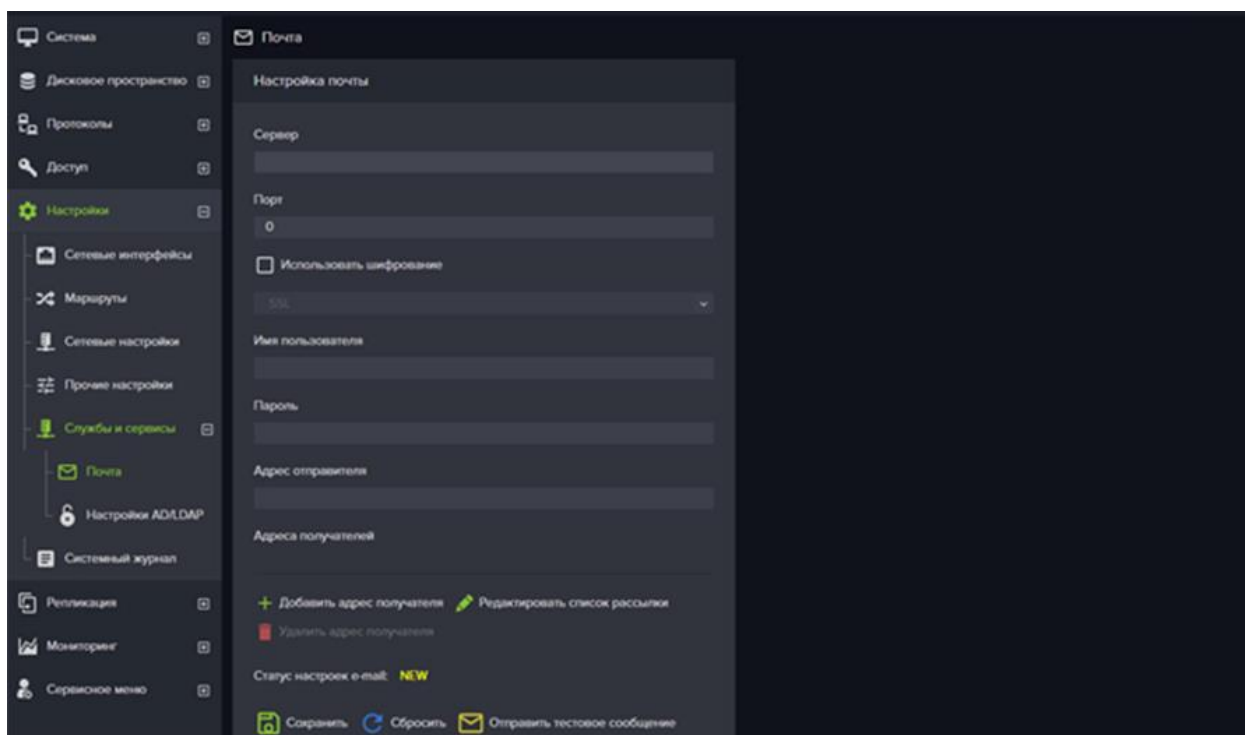


Рисунок 8 - Параметры настройки оповещений

Администратор управляет сервисами оповещения посредством веб-интерфейса. Все действия администратора и конфигурация СХД (подключенные /отключенные диски) имеют отражение в метаданных системы хранения. Часть метаданных заполняется самим сервером в отдельных потоках в процессе периодического исполнения вызовов инструментов операционной системы. Часть метаданных заполняется на основе данных файловой системы. Остальное – результат выполнения команд администратора СХД.

7.4 Системные записи

Системные записи (журнал событий) используются для диагностики состояния ПО. Системные записи формируются программным компонентом мониторинга и отображаются в веб-интерфейсе управления или выгружаются в отдельный файл (Рисунок 9).

Изм.	Подп.	Дата

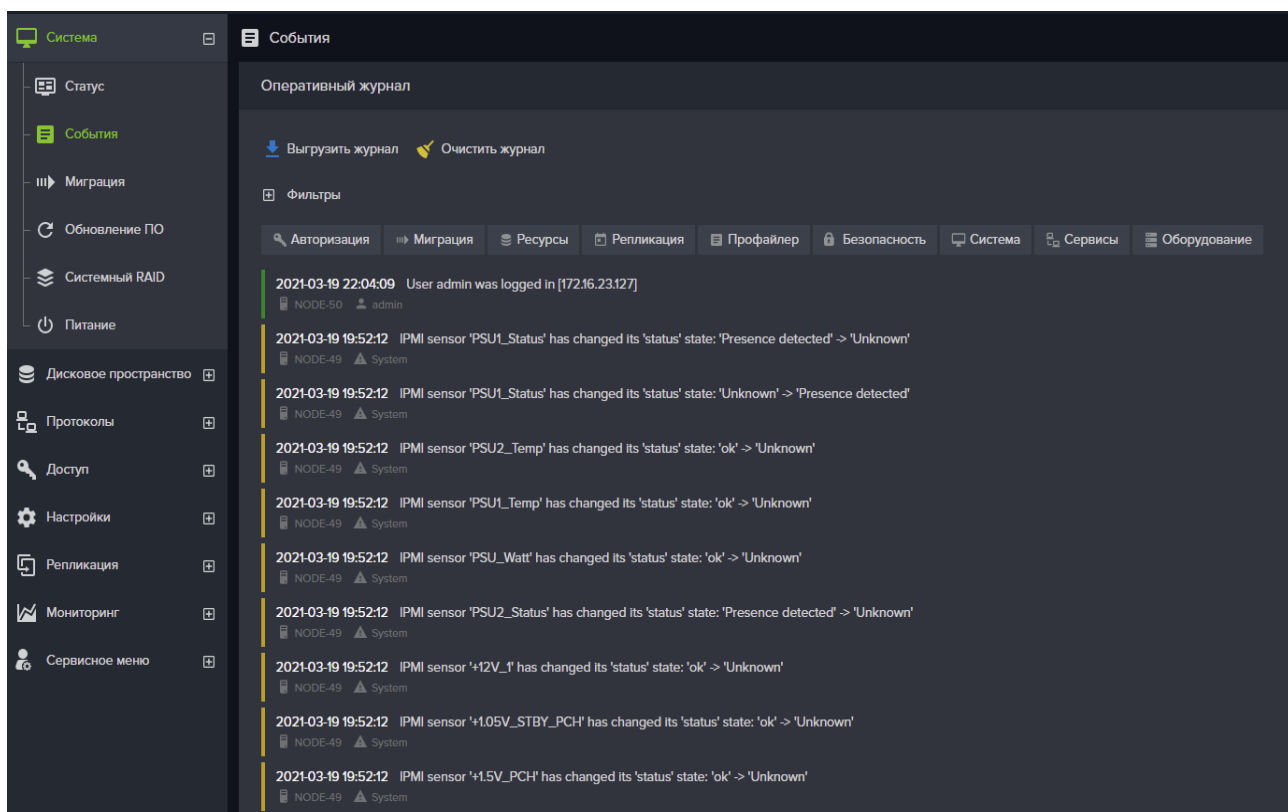


Рисунок 9 - Параметры отображения системных записей

7.5 Команды предоставления доступа к виртуальным дискам для протоколов Fibre Channel, iSCSI, NFS, SMB

Подробное описание представлено в разделе 6.2 «Взаимодействие с Пользователем».

Изм.	Подп.	Дата

[illegible]

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>

--	--	--	--	--	--	--	--	--	--

<i>Изм.</i>	<i>Подп.</i>	<i>Дата</i>