

DOI: 10.15514/ISPRAS-2025-37(6)-34



## Типовая архитектура высокопроизводительной вычислительной системы для решения задач численного моделирования

*С.Ю. Мокшин, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>*

*А.О. Игнатьев, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>*

*А.И. Мельников, ORCID: 0009-0008-1982-9501 <a.i.melnikov@vniitf.ru>*

*Д.В. Иванков, ORCID: 0000-0003-4254-0104 <d.v.ivankov@yandex.ru>*

*Российский Федеральный Ядерный Центр – Всероссийский научно-исследовательский институт технической физики имени академика Е.И. Забабахина,  
Россия, 456770, г. Снежинск, Челябинская область, ул. Васильева, 13.*

**Аннотация.** Работа посвящена тематике высокопроизводительных вычислительных систем (ВВС), предназначенных для решения задач численного моделирования. Приводится подробное описание архитектуры, функционального состава, а также используемого системного программного обеспечения и особенностей обработки информации в ВВС. Авторы работы опираются на собственный многолетний опыт создания ВВС для предприятий России. Данная работа может быть полезна специалистам, занимающимся разработкой и эксплуатацией современных вычислительных систем высокой производительности, предназначенных для проведения научных исследований.

**Ключевые слова:** архитектура вычислительной системы; высокопроизводительные вычисления; обработка данных математического моделирования; системное программное обеспечение.

**Для цитирования:** Мокшин С.Ю., Игнатьев А.О., Мельников А.И., Иванков Д.В. Типовая архитектура высокопроизводительной вычислительной системы для решения задач численного моделирования. Труды ИСП РАН, том 37, вып. 6, часть 3, 2025 г., стр. 19–44. DOI: 10.15514/ISPRAS–2025–37(6)–34

## Typical HPC System Architecture for Numerical Simulation

*S.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <s.yu.mokshin@vniitf.ru>*

*A.O. Ignatyev, ORCID: 0000-0003-4902-2123 <a.o.ignatyev@vniitf.ru>*

*A.I. Melnikov, ORCID: 0009-0008-1982-9501 <a.i.melnikov@vniitf.ru>*

*D.V. Ivankov, ORCID: 0000-0003-4254-0104 <d.v.ivankov@yandex.ru>*

*Russian Federal Nuclear Center –*

*Zababakhin All-Russian Research Institute of Technical Physics,  
13, Vasilieva street, Snezhinsk, Chelyabinsk region, 456770, Russia.*

**Abstract.** The paper discusses issues related to the topic of high-performance computing (HPC) systems designed to solve numerical simulation problems. A detailed description of the architecture, functional composition, as well as the system software used and information processing features in the HPC systems is provided. The authors of the work rely on their own long-term experience in creating HPC systems for Russian scientific and industrial centers. This work may be useful to specialists involved in the development and operation of modern high-performance computing systems designed for scientific research.

**Keywords:** computing system; calculations; high-performance computing simulation; mathematical simulation.

**For citation:** Ignatyev A.O., Mokshin S.Yu., Melnikov A.I., Ivankov D.V. Typical HPC system architecture for numerical simulation. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 6, part 3, 2025. pp. 19-44 (in Russian). DOI: 10.15514/ISPRAS-2025-37(6)-34

### 1. Введение

Понятие высокопроизводительной вычислительной системы (суперкомпьютера) появилось в конце 60-х годов и применялось в то время к уникальным по производительности вычислительным системам Сеймура Крэя [1]. Под суперкомпьютером понималась вычислительная система, создаваемая, как правило, в единственном экземпляре или в небольшом количестве, значительно превосходящая по производительности массово выпускаемые образцы и ориентированная в первую очередь на вычисления, связанные с обработкой большого количества данных за приемлемое время (это, как правило, задачи численного моделирования физических процессов).

Первоначально в суперкомпьютерах применялись специально разработанные дорогие быстродействующие процессоры с уникальными архитектурами. Суперкомпьютер тех времен состоял из одного или нескольких (до 10) центральных процессоров (как правило, с векторной архитектурой), взаимодействующих с общей оперативной памятью. Каждый суперкомпьютер работал под управлением собственной операционной системы, на нем использовались специально написанные под него компиляторы (как правило, для языков Алгол или Фортран).

Однако, появление в конце 90-х годов прошлого столетия серийных универсальных процессоров с производительностью, сравнимой с производительностью ранее выпускаемых суперкомпьютеров, а также коммуникационного оборудования, способного передавать информацию между ними с приемлемо высокой скоростью, привело к революции в суперкомпьютеростроении. С тех пор и по сегодняшний день, все суперкомпьютеры имеют кластерную архитектуру, то есть состоят из большого числа как правило однотипных серверов, объединенных высокоскоростным коммуникационным оборудованием.

Первая серьезная публикация о высокопроизводительной вычислительной системе (ВВС) такого типа на русском языке появилась в 2003 году [2]. Чуть позже вышла докторская диссертация А.О. Лациса на эту же тему, с которой авторы имели возможность ознакомиться. Многие последующие публикации, как в журналах, так и в Интернете, посвящены, как правило, либо обоснованиям необходимости применения ВВС, либо отдельным архитектурным аспектам конкретного экземпляра или семейства. Лишь в 2020 году усилиями

коллег из ФГУП «РФЯЦ-ВНИИЭФ» были разработаны ГОСТ на тематику ВВС [3, 4], в которых вводятся общие понятия о структуре ВВС и порядке проведения их испытаний. И хотя за прошедшие 20 лет общая архитектура ВВС, используемых для решения задач численного моделирования, практически устоялась, но до сих пор не существует доступного и полного ее описания.

В настоящей работе дано подробное описание архитектуры ВВС, предназначенной для решения задач численного моделирования, включающее в себя описание основных функциональных компонент ВВС, используемое системное и прикладное программное обеспечение, а также рассматриваются особенности обработки информации в ВВС такого типа. Авторы работы опираются на многолетний опыт участия сотрудников РФЯЦ-ВНИИЭФ в создании ВВС и их компонент для предприятий России, а также на опыт подобных разработок за рубежом [5-10].

## **2. Основные компоненты ВВС**

Архитектура современной ВВС, предназначенной для решения задач численного моделирования, строится исходя из основной цели создания такой ВВС – обеспечения выполнения массово-параллельных задач численного моделирования на всех доступных пользователю вычислительных ресурсах ВВС. Под архитектурой ВВС здесь понимается общее описание структуры и функций ВВС, связей между ее компонентами на уровне, достаточном для понимания принципов работы, не включающее деталей технического и физического характера. Исходя из этого, ВВС должна содержать очень большое количество вычислительных узлов, объединенных высокопроизводительной вычислительной сетью передачи сообщений, подключенных к доступным с каждого вычислительного узла файловым ресурсам и оснащенным системным и прикладным программным обеспечением, обеспечивающим заданную функциональность.

На практике вычислительным узлом ВВС является материнская плата, на которой установлены одно или несколько вычислительных устройств (процессоров), общая для них оперативная память, контроллеры ввода/вывода и управления, сетевые адаптеры и, при необходимости, дисковые накопители. Каждый узел объединяется с другими узлами ВВС посредством коммуникационного (сетевого) оборудования. В то же время, с точки зрения пользователей, ВВС является единым цельным объектом, подключенным к сети предприятия.

В зависимости от назначения, узлы могут быть разного типа и комплектации. Каждый узел работает под управлением собственного экземпляра операционной системы (ОС узла). Согласованную работу всех узлов обеспечивает системное программное обеспечение ВВС (СПО ВВС).

ВВС состоит из нескольких систем и подсистем, выполняющих различные функции:

- вычислительное поле (ВП) – основная из подсистем ВВС (с точки зрения реализации её функций), предназначена для проведения вычислений;
- система хранения (СХ) – предназначена для хранения исходных данных, промежуточных и финальных результатов вычислений;
- подсистема доступа (ПД) – обеспечивает непосредственный доступ пользователей к ВВС, подготовку начальных данных для расчетов, обработку результатов расчетов и запуск задач на ВП;
- подсистема управления и мониторинга (ПСУМ) – реализует функции общего управления работой всех узлов ВВС, включая запуск и прохождение задач, мониторинг и диагностику оборудования и системного программного обеспечения;
- сервисная подсистема (СП) – обеспечивает организацию единого пространства пользователей (подсистему учета и аутентификации пользователей), доменных

имен, информационных и некоторых других вспомогательных для ВВС сервисов;

- коммуникационная подсистема (КПС) – обеспечивает сетевое взаимодействие всех узлов ВВС между собой.

Таким образом, функциональную схему ВВС можно представить следующим образом (см. рис. 1).

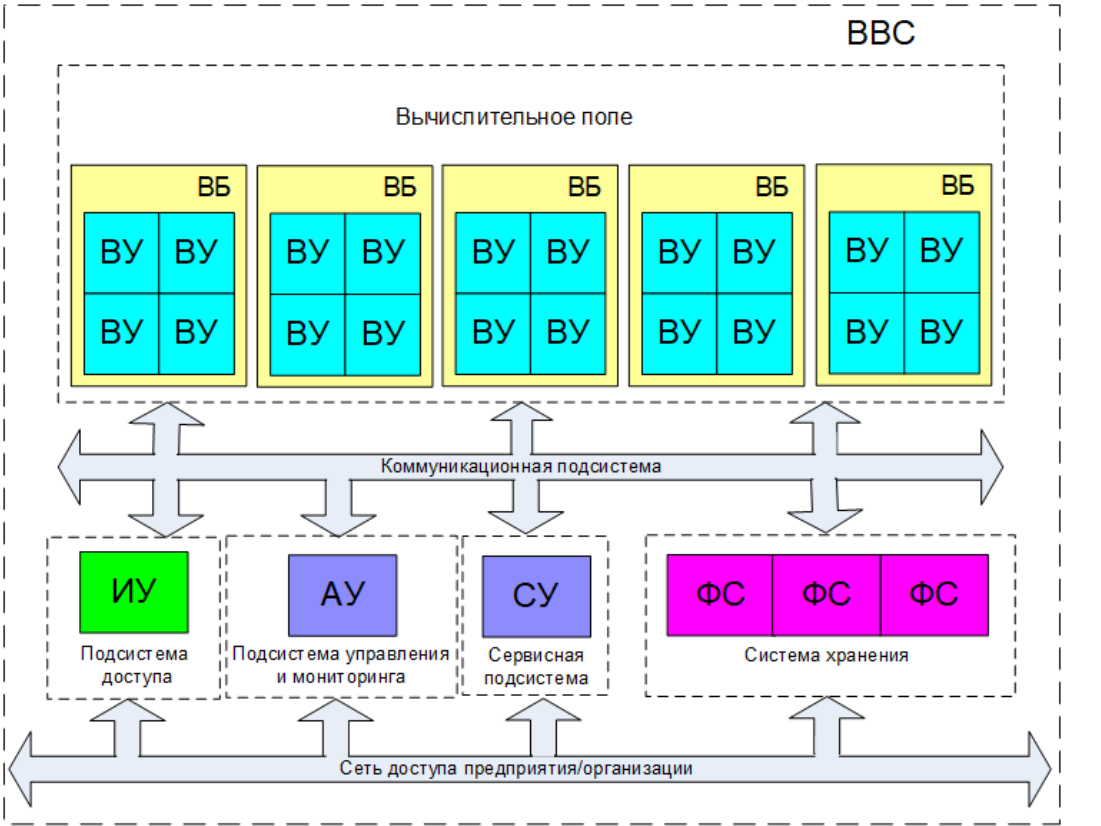


Рис. 1. Функциональная схема ВВС.  
Fig.1. HPC system functional diagram.

На этом рисунке приняты следующие обозначения:

- ВУ – узлы ВП (вычислительные узлы);
- ВБ – вычислительные блоки ВП;
- ИУ – узлы подсистемы доступа (инструментальные узлы);
- СУ – узлы сервисной подсистемы (сервисные узлы);
- АУ – узлы подсистемы управления и мониторинга (административные узлы);
- ФС – узлы системы хранения (файловые серверы).

Рассмотрим каждую из этих подсистем подробнее.

### 2.1 Вычислительное поле

Вычислительное поле – подсистема ВВС, предназначенная для проведения расчетов, содержит в себе множество вычислительных узлов (ВУ), объединенных в вычислительные блоки (ВБ). Фактически реализует суммарную вычислительную мощность ВВС для расчета задач.

Каждый ВУ содержит один или несколько процессоров, общую для них оперативную память и устройства или каналы передачи информации (сетевые адаптеры, интегрированные порты), подключенные к коммуникационной подсистеме, также может содержать сопроцессоры или различные ускорители вычислений, которые, в некоторых случаях, могут вполне успешно использоваться для решения задач численного моделирования.

Суммарная пиковая производительность вычислительного поля  $R_{peak}$  вычисляется по формуле:

$$R_{peak} = N_{CPU} * F * N_c * N_{CORE} * M$$

где:

$N_{CPU}$  – количество процессоров в одном вычислительном узле;

$F$  – частота процессора, ГГц;

$N_c$  – количество конвейеров с плавающей запятой в одном ядре процессора;

$N_{CORE}$  – количество ядер в процессоре;

$M$  – количество вычислительных узлов в вычислительном поле.

Загрузка операционной системы на ВУ осуществляется либо с локального жесткого диска, либо по сети с системного узла. Как правило, на ВВС отдается предпочтение сетевому способу загрузки операционной системы (ОС) на ВУ, обеспечивающему высокую эффективность администрирования, гибкость управления и некоторую экономию денежных средств за счет отсутствия необходимости покупки жестких дисков для ВУ (особенно при большой размерности вычислительного поля).

При большом количестве ВУ (а на реальных ВВС в списке TOP-500 [11] это количество составляет от нескольких тысяч до десятков тысяч) вычислительные узлы целесообразно объединять в вычислительные блоки (ВБ). Такое деление на логические единицы – ВБ реализует возможность управления ими индивидуально, обеспечивая возможность проведения работ с необходимыми и конкретными ВБ в ходе эксплуатации ВВС: проведения профилактических работ, аварийного поблочного отключения ВВС при отказе систем жизнеобеспечения ВВС (энергоснабжения, охлаждения) и т.п. Кроме того, очевидно, что управлять очень большим вычислительным полем с одного системного узла в случае сетевой загрузки – невозможно. Поэтому такое разделение структуры вычислительного поля на ВБ позволяет объединять от 80-ти до 120 ВУ в один вычислительный блок, при этом загрузка ОС на ВУ данного ВБ обеспечивается с отдельного системного узла, входящего в состав этого ВБ. Таким образом, в составе вычислительного поля ВВС может быть некоторое множество ВБ и каждый ВБ содержит в себе системный узел ВБ, обеспечивающий:

- бездискковую загрузку каждого ВУ блока;
- диагностику всех вычислительных узлов блока.

Важно отметить, что такая организация структуры вычислительного поля позволяет обеспечивать и его отказоустойчивость, так как, фактически, каждый системный узел представляет собой резервную копию любого другого системного узла в составе вычислительного поля (при необходимости с минимальными отличиями в настройках системных компонентов).

Необходимо учитывать, что при сетевой загрузке, в случае использования механизма монтирования файловой системы ВУ с системного узла по протоколу NFS, количество эффективно обслуживаемых системным узлом ВУ не превышает 100-120 узлов (особенности протокола NFS, являющегося «золотым стандартом» для сетевой загрузки). Безусловно, это количество можно увеличить, используя на вычислительном поле узлы с микроядром ОС без прямого монтирования всей структуры файловой системы ВУ, но в этом случае многие оперативные изменения в настройках ОС ВУ потребуют перезапуска ВУ. В то же время необходимость такого оперативного внесения изменений в ОС ВУ достаточно очевидна и

обусловлена, в свою очередь, необходимостью адаптации системного окружения и конфигурации системного программного обеспечения (ПО) под задачи пользователя, либо оптимизацией системных настроек при изменении состава оборудования, масштабировании ВВС. Кроме того, каждый ВУ является сетевым устройством в общей сети ВВС и администратору ВВС важно обеспечить такой IP-адрес ВУ, который будет соответствовать некоторой логической структуре ВП (блок-узел), при этом не превышающий общую размерность IP-подсети, соответствующей вычислительному блоку, поэтому превышать вышеуказанное количество ВУ в ВБ нецелесообразно.

На каждом ВУ запускается агент системы управления и клиенты сетевых файловых систем СХ, а также с помощью специального ПО обеспечивается исполнение различных процедур библиотек передачи сообщений (как правило, MPI), используемых в параллельных прикладных программах.

## 2.2 Система хранения

Система хранения – подсистема ВВС, обеспечивающая хранение программ, начальных данных, промежуточных и финальных результатов расчетов задач численного моделирования. СХ состоит из множества устройств хранения, коммуникационного и серверного оборудования. Входящие в состав СХ программные компоненты обеспечивают как файловый, так и объектные сервисы хранения различного назначения.

Главным элементом СХ типовой ВВС является параллельная файловая система, обладающая высокой производительностью ввода-вывода и способная оперативно обслуживать большое количество запросов, поступающих от процессов счетных задач, программ постобработки данных и других пользовательских приложений ВВС. Ресурсы хранения параллельной файловой системы представлены группой файловых серверов, оснащенных высокопроизводительными дисковыми массивами. Используемое здесь оборудование обладает большой степенью аппаратной избыточности, что обычно определяет высокую удельную стоимость хранения данных на параллельной файловой системе и, как следствие, – ограниченный объем ее ресурсов. Клиентская часть параллельной файловой системы устанавливается на всех узлах ВП и инструментальных узлах системы доступа. Такой широкий охват и высокая зависимость вычислительного техпроцесса от наличия и работоспособности оперативных файловых ресурсов делают параллельную файловую систему одной из важнейших подсистем ВВС. Примерами параллельных файловых систем, широко используемых в настоящее время в данной предметной области, являются Lustre [12], BeeGFS [13], Spectrum Scale [14].

Ввиду ограниченного объема ресурсов, а также характерного для параллельных файловых систем приоритета производительности ввода-вывода над надежностью хранения, эти файловые системы востребованы, прежде всего, на этапе расчета задач численного моделирования, когда генерируется и модифицируется основной объем расчетных данных. Поэтому параллельные файловые системы, входящие в ВВС, часто называют «рабочими», «оперативными» или даже «черновыми» (от англ. *scratch* - черновик). Период нахождения данных на файловой системе обычно ограничен временем выполнения расчетов соответствующей задачи численного моделирования и может составлять от суток до нескольких недель. Несмотря на столь ограниченный срок, параллельная файловая система решает практически все поставленные перед СХ задачи по обеспечению процесса вычислений ресурсами хранения данных и, тем самым, удовлетворяет большинство потребностей многих заказчиков.

Для более длительного и более надежного хранения сгенерированных задач расчетных данных, а также для обеспечения возможности их постобработки и визуализации различными программами в структуру СХ вводят дополнительные компоненты, обладающие, помимо указанных свойств, встроенной информационной избыточностью, масштабируемостью и

способностью предоставлять прямой доступ к данным посредством стандартных сетевых протоколов не только для структурных компонентов ВВС, но и со стороны смежных с ней информационных систем. Такие ресурсы хранения часто называют «проектными», так как административно они выделяются для отдельных научных коллективов (проектов). Отдельным их подвидом можно считать домашние директории пользователей ВВС. Время жизни данных на проектных ресурсах обычно измеряется месяцами.

С технической точки зрения проектные ресурсы хранения ВВС реализуется либо с помощью специализированных файловых серверов (так называемых «NAS-файлеров»), либо на основе распределенных масштабируемых программно-определяемых «облачных» систем хранения с файловым или объектным интерфейсом. Примерами подобных систем, кроме множества коммерческих продуктов разных производителей, можно назвать Ceph [15], Openstack Swift [16], отечественная облачная система хранения данных (ОСХД) «Стриж» [17].

В зависимости от потребностей заказчика система хранения ВВС может включать в себя дополнительную подсистему, предназначенную для долговременного хранения некоторого подмножества сгенерированных задачей расчетных данных. Ввиду длительности их хранения, требуемых на протяжении всего этого времени гарантий их неизменности, а также их постоянно возрастающего объема, эта (архивная) подсистема строится на основе технологии магнито-ленточной памяти, которая обладает минимальной по индустрии удельной стоимостью гигабайта. Наличие системы такого класса позволяет эксплуатирующей ВВС организации увеличить объем данных, находящихся в СХ ВВС, и уменьшить стоимость владения ими. К примерам таких подсистем можно отнести HPSS [18], Enstore [19], и отечественную разработку – архивную систему хранения данных (АСХД) [20]. Таким образом, полную архитектуру системы хранения любой современной ВВС можно с уверенностью назвать многоуровневой/многослойной. Выбор слоя для размещения/перемещения очередного набора данных расчетной задачи определяется предъявляемыми к СХ требованиями, которые отражают текущий этап жизненного цикла этих данных. Ввиду уникальности процессов расчета задач численного моделирования, сложившихся у разных заказчиков, а также из-за различия их организационных, технических и финансовых возможностей, система хранения любой типовой ВВС является уникальной.

## 2.3 Подсистема доступа

Подсистема доступа – это подсистема ВВС, предназначенная для организации доступа пользователей к ресурсам ВВС, включает в себя множество инструментальных узлов (ИУ). Каждый ИУ подключен посредством коммуникационной системы к остальным узлам ВВС, а с помощью сети доступа – к сети предприятия/организации, откуда пользователи ВВС могут к ним обращаться.

На ИУ выполняются следующие виды работ:

- организация полноценного графического или консольного сеанса пользователя;
- создание и редактирование файлов программ и данных;
- компиляция и сборка программ;
- проведение расчетов на ИУ;
- запуск задач на ВП;
- обработка результатов счета.

ИУ похож на ВУ, за исключением возможно большего числа процессоров и большего объема оперативной памяти, количество которых варьируется в зависимости от количества пользователей на ИУ, а также дополнительного сетевого интерфейса для подключения к сети доступа. Кроме того, для выполнения задач визуализации результатов расчетов, на ИУ может быть установлен специализированный графический процессор (видеокарта).

При большом количестве ИУ для удобства их администрирования в состав подсистемы доступа может входить системный узел, по функциям аналогичный системному узлу вычислительного блока.

На узлах подсистемы доступа реализуются сервисы доступа, на этих узлах доступны средства разработки прикладного программного обеспечения и клиентские компоненты системы управления задачами и ресурсами, а также другие системные сервисы.

## 2.4 Подсистема управления и мониторинга

Подсистема управления и мониторинга обеспечивает выполнение следующих функций:

- запуск и прохождение задач на ВП (подсистема управления задачами и ресурсами);
- ведение статистики по расчетам;
- мониторинг и диагностику оборудования и системных сервисов;
- ведение статистики по состоянию оборудования;
- конфигурирование и управление (менеджмент) КПС ВВС.

В состав ПСУМ входят несколько административных узлов, оснащенных необходимым СПО. На узлах подсистемы управления и мониторинга запускаются:

- серверные компоненты подсистемы управления задачами и ресурсами;
- ПО менеджмента сетей КПС;
- серверные и клиентские компоненты подсистем мониторинга и диагностики всех узлов и функциональных подсистем ВВС.

## 2.5 Сервисная подсистема

В состав сервисной подсистемы входят вспомогательные, но важные для работы ВВС сервисы:

- сервис учета и аутентификации пользователей;
- сервис доменных имен;
- сервис точного времени;
- WEB-сервис;
- сервис баз данных;
- сервис ведения проектов;
- информационные сервисы: электронная почта и сервис мгновенных сообщений;
- сервис маршрутизации, реализующий функции экспорта данных из СХ на автоматизированные рабочие места (АРМ) пользователей в сети предприятия.

На узлах сервисной подсистемы запускаются серверные компоненты перечисленных системных сервисов. Все указанные сервисы могут быть виртуализованы, либо могут использовать систему контейнеризации для обеспечения удобства миграции и надежности функционирования.

## 2.6 Коммуникационная подсистема

Коммуникационная подсистема ВВС (КПС) – совокупность коммуникационного оборудования, линий связи и специализированного ПО, обеспечивающая передачу данных между компонентами ВВС. КПС состоит из нескольких типов функциональных сетей: сети передачи сообщений, сети данных, сети управления, сети мониторинга и т.д. Может включать различные типы сетевого оборудования для выполнения разных целей (Ethernet, InfiniBand и пр.).



Сеть передачи сообщений и сеть данных, как правило, создаются на базе высокопроизводительных решений, наподобие InfiniBand. С точки зрения обеспечения высоких показателей надежности и готовности вычислительной системы КПС может содержать дублирующие сети, например, сеть данных подсистемы доступа, построенную на оборудовании Ethernet и дублирующую сеть данных на оборудовании InfiniBand.

Сети управления и мониторинга строятся на базе оборудования Ethernet и могут быть совмещены на уровне физической сети. Данные сети подключены ко всем узлам BBC, контроллерам управления (BMC, Baseboard Management Controller) узлов, контроллерам управления коммутаторов КПС, а также контроллерам управления устройств хранения информации CX.

### **3. Обработка информации в BBC**

Поскольку BBC является сложным по своей структуре и назначению объектом, обработка информации на ней существенно отличается от обработки информации на персональных компьютерах или традиционных электронных вычислительных машинах прошлого века. Основные причины этих отличий следующие:

- BBC является распределенной системой, поэтому разные этапы обработки информации могут выполняться на разных функциональных подсистемах BBC.
- Вычислительное поле BBC состоит из множества вычислительных узлов, поэтому эффективное его использование может быть получено только с использованием параллельных программ. В этом случае пользователь вынужден распараллеливать алгоритмы своей вычислительной задачи таким образом, чтобы обеспечить её запуск на общей памяти и нескольких процессорах ВУ, или на распределенной памяти и нескольких узлах BBC (а может быть и на разных BBC). То есть пользователь BBC должен предусмотреть и реализовать возможные уровни распараллеливания для своей задачи. Более того, пользователь вынужден думать о том, чем именно будет заниматься тот или иной процесс вычислительной задачи на ВУ или даже на конкретном ядре процессора, сопроцессора, ускорителя.
- Архитектура BBC оптимизируется для получения сверхвысокой производительности, но, как это всегда бывает, в ущерб универсализму. Типичные BBC для задач численного моделирования поддерживают, как правило, только одну многоуровневую модель параллельного программирования: на верхнем уровне процессы параллельной программы обмениваются сообщениями с использованием библиотеки MPI [21], каждый процесс, в свою очередь, распараллеливается на общей памяти средствами OpenMP [22]. Кроме того, процессы могут обращаться к различным арифметическим ускорителям.
- Так как BBC является распределенной системой, файловые ресурсы, доступные пользователю, размещаются на специальных сетевых файловых системах CX, при этом CX может быть многоуровневой. Такая сложная структура CX BBC обязывает пользователя BBC учитывать эту особенность в процессе своей работы, заставляет планировать размещение своих файлов на различных уровнях CX, реализовывать схемы их миграции между уровнями CX, а также планировать файловый вывод расчетной задачи, например, выбирая схему параллельного вывода с каждого процесса, либо сбор данных и вывод с одного процесса задачи.
- BBC является системой коллективного доступа. Как правило, BBC создаются для нужд целой организации, научного центра, фирмы-разработчика и т.п. Количество пользователей BBC может исчисляться сотнями и тысячами. Поэтому при создании BBC применяются технологии работы пользователей, учитывающие организацию системы разграничения доступа. С одной стороны, пользователи могут работать над

одними и теми же данными общего доступа, с другой стороны они должны иметь доступ только к своим, совершенно конкретным данным.

В то же время, если разделить работу пользователя на основные функциональные этапы, то работа пользователя на ВВС во многом похожа на его работу на ПЭВМ.

На рис. 2 показаны основные этапы обработки информации на ВВС:

- вход в систему;
- разработка прикладного программного обеспечения;
- отладка прикладного программного обеспечения;
- подготовка начальных данных;
- запуск и контроль за выполнением вычислительных задач.

Рассмотрим эти этапы подробнее.

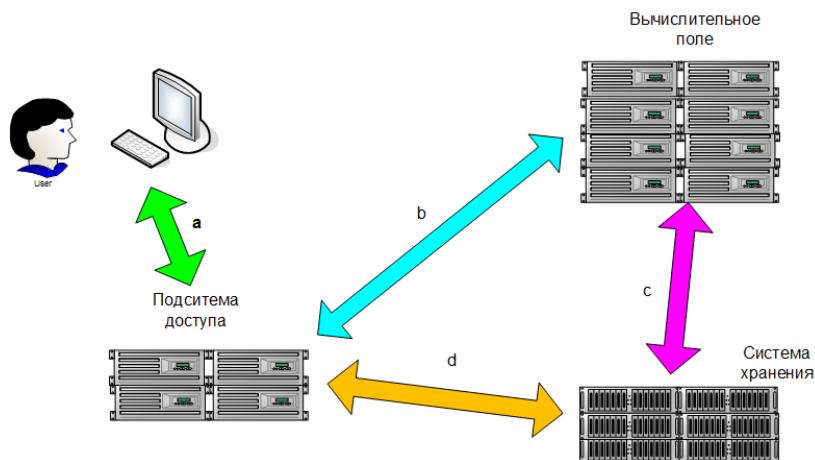


Рис. 2. Основные этапы обработки информации в ВВС.

*a* – вход в систему и обмен информацией между пользователем и системой;

*b* – запуск задач и проведение расчетов; *c* – обмен данными с системой хранения в процессе счета;

*d* – обмен данными с системой хранения в процессе разработки, отладки программ, подготовки начальных данных и обработки полученных результатов.

Fig.2. Key stages of information processing in the HPC system.

*a* – user login and information exchange between user and system;

*b* – tasks launching and performing calculations; *c* – data exchange with the storage system during the calculations; *d* – data exchange with the storage system during the development, debugging, preparation of initial data and post-processing.

### 3.1 Вход в систему

Пользователи получают доступ к ВВС, как правило, со своих АРМ, расположенных в сети предприятия. Существует несколько видов доступа:

- Терминальный (символьный или графический). При этом виде доступа организуется терминальный сеанс, в котором пользователь может использовать приложения из состава СПО ВВС и исполнять прикладные программы. Сеанс организуется на ИУ подсистемы доступа, как правило, с использованием протоколов SSH, XDMCP, VNC (RDP) и т.д. Перед созданием сеанса пользователь должен авторизоваться (т.е. предъявить свое учетное имя) и пройти обязательную аутентификацию (т.е. подтвердить свое имя, например, паролем или ключом доступа). За авторизацию и аутентификацию отвечает сервис учета и аутентификации пользователей (как

правило, создаваемый на основе OpenLDAP [23]). Сеанс организует один из сервисов доступа, о которых мы расскажем чуть позже в данной публикации.

- **Обмен данными.** Существует два вида обмена данными: передача файлов (с использованием сервисов доступа, например, на основе протоколов FTP, SFTP) или экспорт файлового пространства CX BBC в качестве сетевого диска на АРМ пользователя (например, на базе протоколов CIFS/SMB с использованием маршрутизаторов сервисной подсистемы).
- **WEB-доступ.** WEB-сервер предоставляет пользователям доступ к общедоступной информации (руководства, инструкции и другая справочная информация), а после авторизации и обязательной аутентификации – к WEB-приложениям, реализованным в BBC (с использованием протоколов HTTP, HTTPS).
- **Доступ к информационным и пользовательским сервисам.** При установке соответствующих клиентских приложений на своей АРМ пользователь может подключиться к информационным и пользовательским сервисам BBC – электронной почте, сервису мгновенных сообщений, сервису ведения проектов и т.д., для организации обмена информации (с использованием протоколов SVN, HTTPS, HTTP, SMTP, IMAP, POP3, XMPP и др.).

## 3.2 Разработка прикладного программного обеспечения

Разработка прикладного программного обеспечения (ППО) заключается в создании и редактировании текстов программ, компиляции и сборки программ из исходных текстов и необходимых библиотек. Разработка ППО выполняется на ИУ в терминальном режиме доступа. При разработке ППО используются текстовые редакторы и средства разработчика, компиляторы и библиотеки из состава СПО BBC и сторонних производителей, установленные на ИУ.

Кроме ППО собственной разработки на BBC может использоваться ППО сторонних производителей, доступное в текстах или в бинарном виде.

ППО сторонних производителей, доступное в исходных текстах, собирается аналогично собственному ПО предприятия. ППО, доступное в бинарном виде, должно быть совместимо с СПО BBC.

## 3.3 Отладка прикладного программного обеспечения

Отладка прикладного программного обеспечения заключается в устранении ошибок, допущенных разработчиком ППО при создании и редактировании текстов программ, компиляции и сборки программ из исходных текстов и необходимых библиотек. Отладка ППО выполняется на ИУ в терминальном режиме доступа, либо на ВП в пакетном или интерактивном режиме запуска отлаживаемой программы через подсистему управления задачами и ресурсами. При отладке ППО используются штатные отладчики из состава СПО BBC (например, GDB), а также коммерческое ПО сторонних производителей (например, Linaro DDT [24], Perforce Totalview [25]).

## 3.4 Подготовка начальных данных

Подготовка начальных данных для проведения расчетов производится на ИУ подсистемы доступа с использованием текстового редактора, либо специализированных для ППО программ (например, ПО Salome [26]). Также подготовка начальных данных может проводиться на АРМ пользователей, при условии, что формируемые данные будут располагаться на экспортируемых на АРМ пользователя ресурсах CX BBC.

### 3.5 Запуск вычислительных задач

Запуск вычислительных задач, использующих ВП для организации расчетов, производится с ИУ подсистемы доступа с использованием клиентских компонент подсистемы управления задачами и ресурсами. Подсистема управления задачами и ресурсами предоставляет пользователю информацию о состоянии запущенных и выполняемых задач, а также о доступных для его задач ресурсах ВП. Кроме того, она выполняет функции управления расчетами в части задания приоритетности той или иной задачи (группы задач) [8].

### 3.6 Обработка полученных результатов

Обработка полученных в ходе выполнения расчетов данных является заключительным этапом проведения расчетов. Обработка результатов расчетов может проводиться как на ИУ, так и на АРМ пользователя (при наличии возможности получения обрабатываемой информации непосредственно на АРМ). В ряде случаев, при обработке большого объема информации могут потребоваться специальные программы обработки, использующие для своей работы ресурсы вычислительного поля. В частности, при проведении графической обработки (процесса визуализации) результатов могут использоваться программы визуализации, в том числе параллельной, разработанные в российских федеральных ядерных центрах для BBC – VIZI, ЛОГОС Scientific View, а также за рубежом – VisIt [27], ParaView [28].

### 3.7 Организация доступа к данным

Каждый из этапов обработки информации предполагает доступ к данным, хранимым на ресурсах BBC. При этом, безусловно, работа на BBC связана с некоторыми особенностями при доступе к данным, которые мы и рассмотрим далее.

Вход на BBC предполагает создание сеанса пользователя на подсистеме доступа BBC, при этом обрабатываются конфигурационные файлы сеанса, расположенные в домашнем каталоге пользователя на ИУ BBC, с помощью которых формируется системное окружение пользователя на BBC. Одной из особенностей организации доступа к данным на BBC является то, что системное программное обеспечение BBC конфигурируется таким образом, что данное окружение пользователя обеспечивается на всех подсистемах BBC, задействованных в обработке его информации.

Работы по созданию прикладного ПО и подготовке начальных данных выполняются на ИУ как в домашнем каталоге пользователя, так и на файловых ресурсах оперативного уровня хранения. Учитывая очень большие объемы оперативного уровня CX, и, как следствие, отсутствие репликации данных на этом уровне, пользователю следует хранить критичную к потере информацию либо в домашнем каталоге, либо перемещать ее на второй уровень CX.

Данные, используемые в процессе счета задач, хранятся на оперативном уровне CX. К ним также должно применяться следующее правило – критические важные данные должны копироваться на второй уровень CX и резервироваться на архивном уровне CX.

Обработка результатов счета производится на основе данных с оперативного уровня. Полученные результаты могут быть выведены на АРМ пользователя, помещены на второй уровень CX либо на архивный уровень (для долговременного хранения).

Другой особенностью организации доступа к данным на BBC является то, что на BBC пользователи могут совместно работать над общими для них данными. В этом случае в рамках BBC такие пользователи объединяются в тематические группы. На оперативном уровне хранения (а также в домашних каталогах) файловые объекты приписываются к этим тематическим группам и, соответственно, все пользователи тематической группы могут полноценно работать с таким образом организованным набором файлов. В то же время, для сторонних пользователей, не входящих в данную тематическую группу, эти данные недоступны.

#### **4. Программное обеспечение ВВС**

Системное программное обеспечение ВВС основано на дистрибутивах Linux семейства Red Hat. В основном используются дистрибутивы, доступные в исходных кодах, такие как CentOS или Rocky Linux [29,30], дополненные необходимыми компонентами из открытых источников.

Можно выделить следующие программные средства, используемые в ВВС:

- ОС узла – совокупность СПО нижнего уровня, обеспечивающая работу аппаратных составляющих узла (сервера). Для ВВС типично использование дистрибутивов на основе Linux, например, разработки РФЯЦ-ВНИИТФ – СПО Супер-ЭВМ [31], или разработки РФЯЦ-ВНИИЭФ – ЗОС Арамид [32];
- ПО, используемое для разработки прикладных программ – компиляторы с языков программирования C, C++, Fortran и др., системы разработки, прикладные, научные и графические библиотеки и пр.;
- ПО коммуникационной подсистемы – системное ПО, обеспечивающее передачу данных по каналам связи с использованием специфичных для конкретного вида сетевого оборудования протоколов (включая библиотеки, реализующие интерфейс передачи сообщений MPI);
- ПО системы хранения – системное ПО, используемое для организации доступа и хранения данных в системе хранения. Специфично для каждого уровня системы хранения;
- сервис учета и аутентификации пользователей – использует СПО, предназначенное для хранения учётных записей пользователей и реализующее механизмы, применяемые для аутентификации;
- сервис точного времени – использует СПО, обеспечивающее синхронизацию точного времени на всех подсистемах ВВС;
- сервис ведения проектов – сервис, реализующий функции структурированного хранилища программных проектов и ведения версий исходных текстов прикладного и системного программного обеспечения;
- подсистема управления задачами и ресурсами – программная подсистема, основанная на СПО, предназначенном для организации многозадачного и многопользовательского режима выполнения задач на вычислительном поле;
- средства управления и мониторинга компонент ВВС – системное ПО, используемое в процессе эксплуатации ВВС для управления узлами, диагностики работы оборудования;
- сервисы доступа к узлам ВВС – системные компоненты, установленные на узлах ВВС и обеспечивающие подключение пользователей локально или удаленно со своих АРМ, входящих в состав сопряженных автоматизированных систем, к узлам ВВС;
- WEB сервис – сервис, предоставляющий доступ к информационным ресурсам по протоколам HTTP/HTTPS, а также к компонентам подсистемы управления задачами и ресурсами;
- информационные сервисы – сервисы, предназначенные для передачи сообщений между пользователями, а также между компонентами СПО и администраторами на основе электронной почты и ПО мгновенных сообщений;
- сервис баз данных (БД) – сервис, предназначенный для хранения и обработки различной информации в базах данных;
- прикладные программы – программы, предназначенные для решения различных задач численного моделирования.

Вышеперечисленные компоненты, за исключением прикладных программ, составляют системное программное обеспечение BBC.

Приведем краткое описание вышеперечисленных компонент СПО.

## 4.1 ОС узла

Операционная система узлов BBC относится к семейству ОС Linux.

Загрузка ОС на любом узле BBC начинается с загрузки ядра Linux. В зависимости от наличия или отсутствия загрузочного диска выделяется два типа загрузки: дисковая или бездисковая. В случае дисковой загрузки BIOS (Basic Input Output System) узла определяет устройство, содержащее образ загрузчика, считывает данные загрузчика в память и передает ему управление. Загрузчик находит образ ядра на устройстве, загружает его в память, разархивирует и передает ему управление.

В случае сетевой загрузки первичный загрузчик находится в энергонезависимой памяти сетевого адаптера. Начальная загрузка осуществляется в соответствии со спецификацией PXE (Preboot Execution Environment). Для получения параметров загрузки (сетевой адрес сервера загрузки, имя файла вторичного загрузчика и др.) используется протокол DHCP (Dynamic Host Configuration Protocol). Для получения файлов по сети (вторичный загрузчик, ядро, начальная файловая система initramfs) используется протокол TFTP (Trivial File Transfer Protocol).

После загрузки ядро ОС производит тестирование обнаруженного оборудования, инициирует подсистемы управления памятью и процессами, виртуальную файловую подсистему, гипервизор виртуальных машин, а также запускает подсистему инициализации, которая на основании своих конфигурационных файлов запускает остальные системные процессы.

С другими программами ядро взаимодействует путем предоставления им системных вызовов, выполняющих определенные запросы от программ.

Ядро ОС на узле BBC обеспечивает функции создания, управления и идентификации процессов, организацию виртуальной памяти, выделяемой процессам, обслуживания устройств, сетевые коммуникации, доступ к файловым объектам и прочие базовые функции, необходимые для работы вычислительной системы. Ядро представляет собой загружаемый бинарный файл, а также набор динамически подгружаемых модулей, хранящихся в сжатом виде на системных узлах BBC и локальных дисках узлов BBC.

## 4.2 ПО для разработки программ

При разработке прикладных программ на BBC могут использоваться текстовые редакторы VI, Emacs, Gedit, mcedit из состава СПО BBC, а также среды разработки Eclipse [33] и NetBeans [34], Qt Creator [35], компиляторы с языков высокого уровня C, C++, Fortran семейств GNU, от производителей процессоров Intel и AMD (Intel One API и AMD ROCm) [36,37], библиотеки математических функций BLAS [38], LAPACK [39], FFTW [40], и т.д., графические библиотеки GTK, Qt, OpenGL, отладчик GDB из состава СПО BBC, и множество других открытых и проприетарных продуктов.

## 4.3 ПО коммуникационной подсистемы

Коммуникационное ПО включает программное обеспечение, предназначенное для реализации высокоскоростной передачи данных между абонентами в сетях InfiniBand. Данное ПО содержит средства диагностики, мониторинга, конфигурирования сетей и обеспечивает, в том числе, доступ к различным транспортным и управляющим протоколам посредством коммуникационных библиотек и модулей ядра ОС Linux, а также предоставляет возможности и инструментарий для разработки приложений с использованием коммуникационной среды InfiniBand.

Для доступа к коммуникационной подсистеме из прикладных программ используется, как правило, библиотека, реализующая стандарт передачи сообщений MPI (например, MPICH [41], MVAPICH [42], OpenMPI [43], S-MPI [44]), предоставляющая параллельным задачам возможность выполнять передачу массивов данных между процессами задачи в режиме точка-точка или в режиме выполнения коллективных операций (выполняемых одновременно на всех или группе процессов задачи).

## 4.4 Программное обеспечение системы хранения

Опишем программное обеспечение, которое может использоваться на разных уровнях хранения данных, более подробно.

### 4.4.1 NFS

NFS является сетевой файловой системой общего назначения.

NFS предоставляет клиентам прозрачный POSIX-совместимый доступ к файлам и файловой системе сервера. NFS осуществляет доступ только к тем частям файла, к которым обратился процесс, и основное достоинство NFS состоит в том, что она делает этот доступ прозрачным. Это означает, что любое приложение, которое может работать с локальным файлом, с таким же успехом может работать и с файлом NFS, без каких-либо модификаций самой программы. Однако, существующие реализации NFS имеют ограничение на количество одновременных подключений и скорость доступа со стороны параллельных задач, поскольку за обработку запросов отвечает серверная компонента на одном файловом сервере (узле хранения). Поэтому в BBC NFS имеет ограниченное использование:

- в BBC с числом узлов менее 100 – для домашних каталогов и на оперативном уровне;
- в BBC с числом узлов от 100 и более – только для организации бездисковой загрузки ВУ в вычислительном блоке (для монтирования корневой файловой системы ВУ), при этом в качестве NFS-сервера выступает системный узел ВБ.

### 4.4.2 Файловый сервер Lustre

Lustre – это кластерная файловая система объектного типа, способная обеспечивать высокопроизводительный ввод-вывод для множества процессов как в режиме «file-per-process», так и в режиме параллельного доступа «single-shared-file».

Программное обеспечение файловой системы Lustre состоит из трех серверных компонентов, одного клиентского компонента, пользовательских и административных утилит, а также программных средств разработки. Функциональность серверных и клиентской компонентов файловой системы Lustre реализована в виде модуля ядра операционной системы Linux.

Серверные компоненты размещаются на выделенных для соответствующих их целям узлах CX, которые оснащены локальными дисковыми ресурсами либо подключены к внешним дисковым подсистемам.

Сервер объектного хранения OSS (object storage server) – выделенный узел, выполняющий функции хранения файлового содержимого в виде уникально идентифицируемых объектов. Дисковые ресурсы, которыми оснащен каждый OSS, сгруппированы в тома файловой системы Lustre (OST – object storage targets).

Сервер хранения метаданных MDS (meta data server) – выделенный узел, выполняющий функции хранения иерархии пространства пользовательских данных ФС и метаданных. Дисковые ресурсы, которыми оснащен каждый MDS, сгруппированы в тома метаданных файловой системы Lustre (MDT – meta data targets).

Сервер хранения конфигурации ФС (management server) – выделенный или совмещенный с другой ролью узел, выполняющий функции хранения текущей конфигурации файловой системы, охватывающей все множество ее программных и аппаратных компонентов. Конфигурационные данные файловой системы Lustre размещаются на локальных для MGS блочных устройствах (MGT – management targets).

Большинство серверных узлов BBC (вычислительные, административные, инструментальные, сервисные) являются клиентами файловой системы Lustre, которая подключается к ним путем статического монтирования с использованием высокоскоростной коммуникационной сети данных BBC.

Размещение пользовательских файлов в файловой системе Lustre производится согласно встроенному алгоритму распределения, который «старается» равномерно заполнять тома OST. По умолчанию каждый создаваемый в файловой системе Lustre файл целиком размещается на динамически назначенном ему OST. Такой способ размещения соответствует режиму индивидуального ввода-вывода для процессов задачи («file-per-process»). Для включения режима параллельного ввода-вывода нескольких процессов задачи в один файл («single-shared-file») параметры размещения этого файла, определяемые в этом случае самим пользователем, задаются с помощью пользовательской утилиты lctl или посредством интерфейсных функций, предоставляемых программной библиотекой liblustre.so. Назначение этих параметров для отдельной директории позволяет автоматически их наследовать для всех создаваемых файловых объектов в нижележащих уровнях иерархии.

Объединение в единую систему множества томов OST позволяет не только сформировать значительный по объему ресурс хранения данных, но и суммировать пропускную способность задействованных сетевых каналов к соответствующим серверам хранения OSS. Получившаяся в результате такого объединения высокая интегральная производительность файловой системы позволяет задачам численного моделирования минимизировать длительность периодов регулярной записи своего промежуточного состояния на внешней памяти, что, в свою очередь, сокращает астрономическое время проведения расчетов и, как следствие, повышает эффективность эксплуатации BBC.

#### 4.4.3 ОСХД «Стриж»

«Стриж» – это объектная распределенная система хранения данных, обеспечивающая их доступность и целостность, а также способная масштабироваться как по объему, так и по пропускной способности. Основным программным способом взаимодействия с системой является объектный интерфейс доступа к неструктурированным данным и две его реализации SwiftAPI [45] и S3API [46].

ОСХД «Стриж» базируется на двух компонентах облачной платформы OpenStack [16] – сервисе аутентификации Keystone и сервисе объектного хранения Swift, а также включает в себя:

- Gate – многофункциональный пользовательский web-интерфейс управления данными, разработанный в РФЯЦ-ВНИИТФ;
- Elasticsearch [47] – документная СУБД;
- Zabbix [48] – служба сетевого мониторинга;
- HSDS [49] – сервис хранения структурированных данных с HDF5 API [50];
- и другие компоненты.

Благодаря своим характеристикам ОСХД «Стриж» способна выступать в роли площадки для организации проектных ресурсов хранения для BBC и смежных с ней информационных систем.



#### 4.4.4 ПО АСХД

Ключевые особенности технического решения вопроса долговременного, целостного и масштабируемого хранения «холодных» данных можно рассмотреть на примере разработанной во РФЯЦ-ВНИИТФ архивной системы хранения данных (АСХД) [20, 51].

АСХД представляет собой систему массового обслуживания, выполняющую загрузку, хранение, поиск и восстановление цифровой информации на основании пользовательских заявок. Высокая степень сохранности обрабатываемых данных обеспечивается благодаря широкому применению средств контроля целостности, встроенной репликации, использованию технологии магнитных лент, изоляции заполненных архивных носителей и самих архивных объектов. Способность хранения значительного объема информации в течение длительного времени обеспечивается как за счет применения автоматизированных ленточных библиотек различных технологических поколений, так и благодаря разработанному регламенту обмена архивными носителями между библиотеками и стеллажным хранилищем. Источниками входных информационных потоков для АСХД выступают ресурсы хранения данных с файловым и объектным интерфейсом.

В состав программного обеспечения АСХД входят: метасервер, медиасервер, транспортный агент, консольный и веб клиенты, программное обеспечение оператора хранилища архивных носителей. Технический фундамент площадки АСХД составляют группа медиасерверов и ленточных библиотек, объединенных сетью хранения SAN, а также один или несколько служебных узлов. Транспортные агенты размещаются вне контура АСХД — на серверах системы доступа и выделенных серверах, подключенных к ресурсам-источникам данных (файловые системы и ОСХД).

#### 4.5 Сервис учета и аутентификации пользователей

Важно отметить, что в рамках ВВС организовано единое пространство пользователей и единое пространство хранения. Это означает, что на каждом узле пользователь ВВС имеет одинаковый идентификатор, обладает возможностью доступа к данным, расположенным на системе хранения ВВС.

Сервис учета и аутентификации пользователей предназначен для обеспечения идентификации и аутентификации пользователя при входе в систему, идентификации процессов пользователя, запущенных на узлах ВВС, а также для хранения информации об учетных записях пользователей ВВС.

Сервис учета и аутентификации пользователей базируется на Едином Пространстве Пользователей (ЕПП), реализованном с использованием централизованной базы данных пользователей, хранящейся в службе каталогов OpenLDAP [23] (в составе сервисной подсистемы), динамического модуля аутентификации Linux PAM (на каждом узле ВВС) и механизмов идентификации пользователей и процессов, реализованных в ядре ОС узла. Для автономных АРМ и отдельных учетных записей на узлах ВВС допускается использовать локальные учетные записи, расположенные в файле /etc/passwd.

Идентификация пользователя производится при его входе в систему (точнее, при подключении пользователя к одному из системных сервисов, размещенных в системе доступа ВВС и предоставляющих услуги по организации доступа – терминальный и графический доступ, режим передачи файлов, удаленное монтирование файловых ресурсов ВВС и пр.)

Функция идентификации и аутентификации пользователей основывается на использовании механизма PAM и наличии ЕПП на основе LDAP.

После успешной регистрации пользователя на одном из сервисов доступа, как правило, создается первичный процесс пользователя. В дальнейшем, процессы пользователя порождаются средствами ОС узла или с помощью подсистемы управления задачами и ресурсами. Основными атрибутами, идентифицирующими пользователя в процессе,

являются уникальный идентификатор пользователя (uid) и уникальный идентификатор группы (gid). Эти атрибуты используются в подсистеме разграничения доступа при организации дискреционной модели доступа.

## 4.6 Сервис точного времени

Сервис точного времени на основе NTP (Network Time Protocol) входит в сервисную подсистему BBC и обеспечивает синхронизацию системного времени на всех узлах BBC по отношению к эталонному узлу. Как правило, эталонным узлом является один из административных узлов системы управления и мониторинга, либо специально выделяемый сервер точного времени, использующий оборудование GPS или GLONASS [52,53], либо высокоточные часы с цифровым радиоприемником для получения сигнала точного времени.

## 4.7 Сервис ведения проектов

Сервис ведения проектов входит в сервисную подсистему и предназначен для ведения версий исходных текстов прикладного и системного программного обеспечения. Сервис ведения проектов может использовать ПО Subversion (SVN) [54] или GIT [55], доступное в исходных кодах и обладающее широким функционалом.

SVN и Git предназначены для помощи в разработке программных проектов: позволяют управлять файлами и каталогами, а также сделанными в них изменениями. Это дает возможность отслеживать изменения в исходном коде программ при разработке программных проектов, восстановить более ранние версии исходных текстов, изучить историю всех изменений.

Указанное ПО представляет собой ПО общего назначения, которое можно использовать для управления любыми типами файлов, в частности для ведения версий разрабатываемого на BBC прикладного и системного программного обеспечения.

Основным объектом является проект – совокупность данных, описывающих структуру хранимой в проекте информации и ее версии. К каждому проекту назначается список доступных к нему пользователей.

Пользователь может экспортировать свои локальные данные в проект, импортировать данные из проекта (любой версии) в свое локальное пространство, внести изменение в проект под новой версией.

## 4.8 Подсистема управления задачами и ресурсами

Подсистема управления задачами и ресурсами предназначена для запуска и выполнения задач пользователей (прикладных программ) в пакетном и интерактивном режиме с использованием ресурсов вычислительного поля BBC.

На BBC для организации данной подсистемы в основном используется ПО Slurm (Simple Linux Utility for Resource Management) [56], являющееся высокомасштабируемой программной системой с открытым исходным кодом, предназначенной для управления ресурсами BBC, планирования и запуска вычислительных задач. Slurm выполняет три ключевые функции:

- захватывает ресурсы (вычислительные узлы) для пользователей в необходимом количестве на определенное время;
- предоставляет средства для запуска и мониторинга задач на выделенных узлах;
- организует очередь задач, выполняет планирование запуска задач согласно настроенным правилам, предотвращает конфликты при захвате ресурсов.

Slurm состоит из следующих основных компонент:

- Центральный демон slurmctld (контроллер) – управляет работой всей подсистемы и

обрабатывает запросы пользовательских команд. Дополнительно может функционировать резервный контроллер, запущенный на другом сервере. Резервный контроллер получает управление в случае отказа основного сервера.

- Узловые демоны `slurmd`, работающие на каждом вычислительном узле. Все узловые демоны взаимодействуют друг с другом и другими компонентами через сеть, образуя отказоустойчивую иерархическую структуру.
- Утилиты (команды), предназначенные для взаимодействия пользователя (или администратора) с `Slurm`;
- Опциональный демон взаимодействия с базой данных `slurmdbd`, который может использоваться для сохранения учётной информации по задачам и ресурсам в базу данных. Один демон `slurmdbd` может обслуживать несколько ВВС.

Логические объекты, управляемые `Slurm`, включают следующие сущности:

- узлы (`nodes`) – вычислительные ресурсы;
- разделы (`partitions`) – узлы, логически объединённые в одно множество;
- задачи (`jobs`) – выделенные ресурсы для конкретного пользователя на определённое время;
- шаги задачи (`job steps`) – множества процессов (экземпляров программы) внутри задачи.

Разделы можно рассматривать как очереди задач, каждая из которых имеет набор ограничений, таких как: размер задачи, пользователи, для которых разрешен на данный момент запуск задач и т.д.

Узлы внутри раздела выделяются для задач до тех пор, пока достаточно ресурсов (узлов, процессоров, памяти и т.д.). После того, как под задачу выделен набор узлов, пользователь может инициировать шаги задачи (запускать параллельные программы) в любой конфигурации на выделенных ресурсах. Например, один шаг может использовать сразу все узлы, выделенные для задачи, или несколько шагов могут независимо использовать только часть узлов.

При постановке задачи пользователя в очередь на исполнение `Slurm` запоминает идентификатор пользователя, поставившего задачу и связанный с ним идентификатор группы. В дальнейшем, после выделения вычислительных ресурсов, `Slurm` выполняет запуск процессов задачи и назначает им ранее сохранённые идентификаторы пользователя и группы. Таким образом, процессы на выделенных задаче узлах имеют те же права доступа к данным, что и пользователь, запустивший задачу.

## 4.9 Средства управления и мониторинга компонент ВВС

Средства управления и мониторинга компонент ВВС представляют собой ПО, являющееся частью подсистемы управления задачами и ресурсами ВВС и предназначенное для выполнения следующих функций:

- анализа состояния узлов и другого оборудования ВВС;
- управления (включения, выключения, конфигурации) оборудованием ВВС;
- диагностики состояния оборудования ВВС;
- диагностики и управления системных сервисов ВВС.

Средства управления и мониторинга компонент ВВС состоят из системных компонент, взаимодействующих с контроллерами оборудования и системными журналами, и получающих от них данные диагностики. Кроме того, в состав этих средств входят

консольные утилиты, взаимодействующие с контроллерами оборудования с целью управления и конфигурации.

Для этих целей используется обычно ПО Ganglia [57], Zabbix [48], Nagios [58], а также ПО нижнего уровня для работы с интерфейсом IPMI (Intelligent Platform Management Interface), например, ipmitool [59].

#### 4.10 Сервисы доступа

Сервисы доступа обеспечивают подключение пользователей к ВВС. Функционально данные сервисы реализуются на наборе следующих инструментов и компонентов ОС:

- `login` – вход на узел (АРМ) с непосредственно подключенного к узлу (АРМ) терминала. После авторизации и аутентификации на узле образуется сессия (`shell`), позволяющая выполнять на узле различные команды от имени пользователя.
- `ssh` – удаленный вход на узел с АРМ пользователя или другого узла. После авторизации и аутентификации на узле образуется сессия (`shell`), позволяющая выполнять на узле различные команды от имени пользователя. Для входа могут быть использованы либо встроенные средства ОС Linux, либо различные `ssh`-клиенты, например, PuTTY [60].
- `xdm` – удаленный вход на ИУ подсистемы доступа в графическом режиме с АРМ пользователя. После авторизации и аутентификации на ИУ создается графическая сессия, позволяющая пользователю запускать различные графические приложения. Для входа могут быть использованы либо встроенные средства ОС Linux, либо различные пакеты, эмулирующие X-сервер на АРМ под управлением ОС Windows – Xming [61], OpenText Exceed [62] и т.д.
- `lightdm` – вход на АРМ в графическом режиме. После авторизации и аутентификации на АРМ создается графическая сессия, позволяющая пользователю запускать различные графические приложения.
- `vnc` – удаленный вход на ИУ подсистемы доступа в графическом режиме с АРМ пользователя. Предполагает запуск на ИУ из ранее созданной терминальной сессии программы `vncserver`, после этого на АРМ пользователя запускается утилита `vncviewer`. После авторизации и аутентификации на ИУ создается графическая сессия, позволяющая пользователю запускать различные графические приложения, в том числе, ориентированные на обработку 3D-графики. Для входа могут быть использовано различное специализированное ПО, например, TurboVNC [63].
- `ftp` – удаленный вход с АРМ пользователя или других узлов ВВС на узлы с установленным `ftp`-сервером для передачи файлов. Для входа могут быть использован любой `ftp`-клиент.
- `samba` – подключение к узлам маршрутизации сервисной подсистемы для подключения ресурсов СХ в качестве сетевых дисков к АРМ пользователя. Требуется авторизация и аутентификация. Для входа могут быть использованы встроенные средства ОС Linux и ОС Windows.

#### 4.11 WEB-сервис

WEB-сервис может быть реализован на основе ПО Apache [64], входящего в состав большинства дистрибутивов ОС Linux, в том числе дистрибутива СПО Супер-ЭВМ. WEB-сервер Apache в составе сервисной подсистемы обеспечивает как авторизованный, так и неавторизованный (анонимный) доступ пользователей, администраторов и ремонтно-обслуживающего персонала к информации и системным сервисам по протоколам HTTP/HTTPS.

В роли клиента может выступать любой WEB-браузер (например, Яндекс-браузер [65]), а также любая утилита, передающая серверу запросы по протоколам HTTP/HTTPS.

При анонимном доступе пользователю предоставляется только информация открытого доступа (информация справочного характера).

При авторизованном доступе Apache запрашивает имя и пароль пользователя, затем выполняет аутентификацию с использованием PAM. Далее пользователю может быть предоставлена информация в соответствии с правилами разграничения доступа, реализованными в Apache.

На BBC в режиме авторизованного доступа могут быть реализованы и доступны пользователям многие информационные и функциональные сервисы, например:

- WEB-интерфейс подсистемы управления задачами и ресурсами;
- WEB-интерфейс системы архивного хранения;
- WEB-интерфейс контроля ресурсов на CX;
- портал поддержки, содержащий список вопросов и ответов по интересующим пользователей вопросам и т.д.

## 4.12 Информационные сервисы

Данные сервисы используются для передачи диагностических сообщений от системных сервисов администраторам и обслуживающему персоналу и реализуют функции электронной почты и сервиса мгновенных сообщений.

### 4.12.1 Электронная почта

Электронная почта может использовать следующие компоненты: агент передачи сообщений Postfix [66] и агент доставки сообщений Dovecot [67]. Почтовый сервер Postfix отсылает и принимает почту, а также осуществляет локальный доступ к почтовым ящикам пользователей узла. Dovecot осуществляет удалённый доступ пользователей к их почтовым ящикам по протоколам POP3 и IMAP.

Postfix – это агент передачи сообщений (MTA, Message Transport Agent), который занимается пересылкой по протоколу SMTP сообщений от пользовательского почтового агента (MUA, Mail User Agent), называемого также почтовым клиентом, к удалённому почтовому серверу. MTA также принимает сообщения от удалённых почтовых серверов и пересылает их другим MTA или доставляет в локальные почтовые ящики. Переслав или доставив сообщение, Postfix заканчивает свою работу. За доставку сообщения конечному пользователю отвечают другие серверы, например, сервер Dovecot, который передает по протоколам POP3 и IMAP сообщения различным почтовым клиентам, с помощью которых пользователь BBC может прочитать их.

### 4.12.2 Сервис мгновенных сообщений

Информационный сервис Jabber обеспечивает передачу информации по протоколу XMPP. Для BBC может быть использовано ПО Jabberd2 [68], реализующее модульный сервер XMPP, написанный на языке C.

Поддержка многопользовательских конференций в Jabberd2 реализована в виде отдельного проекта MU-Conference. Хотя XMPP не привязан к какой-то определенной сетевой архитектуре, реализация сессии осуществляется по схеме клиент-сервер, где клиент реализует подключение к серверу с помощью TCP-транспорта и сами серверы взаимодействуют друг с другом, используя протокол TCP.

Сервер ответственен за:

- установление и поддержание соединения или сессии с другими объектами, в виде

XML-потоков к или от авторизованных клиентов, серверов и прочих объектов;

- маршрутизацию корректно адресованных атомарных запросов.

Большинство клиентов подключается непосредственно к серверу посредством TCP и используют XMPP для получения полной функциональности, доступной на сервере.

### 4.13 Сервис баз данных

Сервис баз данных (системы управления базами данных – СУБД) является неотъемлемой частью ВВС. В ВВС данный сервис используется для хранения и обработки различной информации в базах данных. Это и статистические данные по расчетам на ВВС, и сами расчетные данные, накапливаемые при решении прикладных задач численного моделирования.

На ВВС могут применяться СУБД MariaDB [69] и PostgreSQL [70], которые являются реляционными СУБД с открытым исходным кодом.

### 4.14 Прикладные программы

Прикладное ПО является неотъемлемой, и даже подчас определяющей, частью ВВС. Это объясняется тем, что зачастую прикладное ПО разрабатывается отдельными специализированными научными организациями, фирмами, на протяжении многих десятков лет и зачастую никоим образом не зависит от процесса создания ВВС для нужд той или иной организации. Созданное таким образом прикладное ПО является уникальной разработкой и стоимость его иногда сравнима со стоимостью небольшой вычислительной системы. Поэтому часто потребность использования того или иного прикладного ПО определяет структуру и состав как функциональных подсистем ВВС, так и ВВС в целом. В то же время в некоторых крупных научно-исследовательских организациях, принят несколько иной подход к созданию ВВС и разработке прикладного ПО. Прикладное ПО в таких организациях разрабатывается для решения конкретных, исторически определенных узких задач самостоятельно и безусловно учитывает архитектуру и особенности ВВС, специально создаваемых для этих организаций.

Как правило, на ВВС используется ПО ведущих производителей инженерного и научного ПО: Ansys [71], Siemens [72], Dassault Systemes [73], MathWorks [74], ЛОГОС [75], OpenFOAM [76], LAMMPS [77] и многое другое. Данное ПО включает в себя множество модулей для решения задач расчета прочности конструкций, аэродинамики, газодинамики, гидродинамики, теплопроводности, молекулярной динамики, различных физико-химических процессов и т.д. Это ПО позволяет проводить массово-параллельные расчеты фактически по любой научной проблематике.

Все прикладное ПО тестируется на совместимость с системным ПО ВВС, изучаются его возможности по использованию ускорителей вычислений, различных оптимизирующих библиотек и, в соответствии с результатами, может меняться либо состав оборудования, системного программного обеспечения ВВС, либо, в свою очередь, это прикладное ПО со временем дорабатывается и оптимизируется разработчиками для эффективного использования на ВВС.

## 5. Заключение

Разработка архитектуры ВВС является сложной многоэтапной задачей, заключающейся в многофакторном анализе множества исходных данных: комплекса решаемых ВВС задач, наличия или отсутствия прикладного ПО для их решения, финансовых, инженерных возможностей предприятия/организации, наличия или отсутствия квалифицированного персонала по разработке ПО и эксплуатации ВВС. В настоящей работе предложено описание общей архитектуры ВВС: функциональный состав ВВС, используемое системное

программное обеспечение и особенности обработки информации в ВВС, на основе анализа информации о вычислительных системах, создаваемых и эксплуатируемых как за рубежом, так и на предприятиях России. Результаты данной работы носят, безусловно, достаточно общий и субъективный характер и отражают один из возможных подходов к построению ВВС. Тем не менее, авторы надеются, что опубликованная ими работа будет полезна специалистам, занимающимся разработкой и эксплуатацией современных вычислительных систем высокой производительности, предназначенных для проведения научных исследований. Авторы намерены продолжить публикации по данной тематике, с целью представления более развернутой информации по каждой из функциональных подсистем ВВС, основным аспектам и особенностям администрирования ВВС, особенностям разработки прикладного и системного ПО для ВВС, предназначенных для решения задач численного моделирования.

## Список литературы/References

- [1]. J.E. Thornton, Design of a Computer – The Control Data 6600. In the editorials series of Malcolm C Harrison, Courant Institute of Mathematical Sciences, New York University, 1970
- [2]. А. О. Лацис. Как построить и использовать суперкомпьютер. - М.: Бестселлер, 2003. -240 с. 3000 экз. ISBN 5-98158-003-8.
- [3]. ГОСТ Р 57700.27-2020. ВВС. Термины и определения. 2020.
- [4]. ГОСТ Р 57700.26-2020 Высокопроизводительные вычислительные системы. Требования приемочных испытаний.
- [5]. Глазырин А.И., Мокшин С.Ю., Доклад «Суперкомпьютер «Зубр» средней производительности» // Всероссийская Конференция «Информационные технологии в оборонно-промышленном комплексе» (17-20 мая 2016 г), г. Челябинск.
- [6]. Цифровые продукты РФЯЦ-ВНИИТФ. Available at: <https://vniitf.ru/rubric/tsod-i-svem>, accessed 10.05.2025.
- [7]. Мокшин С.Ю. Общие подходы к проектированию подсистемы доступа высокопроизводительных вычислительных систем. Труды ИСП РАН, том 32 вып. 4, 2020, стр. 41-52, DOI: 10.15514/ISPRAS-2020-32(4)-3 / Mokshin S.Yu. General ways to design the access subsystem of high performance supercomputing systems. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 4, 2020, pp. 41-52 (in Russia), DOI: 10.15514/ISPRAS-2020-32(4)-3.
- [8]. Игнатьев А.О., Калинин А.А., Мокшин С.Ю. Реализация функций управления задачами и ресурсами высокопроизводительной вычислительной системы в «СПО Супер-ЭВМ». Труды ИСП РАН, том 34 вып. 2, 2022, стр. 159-178, DOI: 10.15514/ISPRAS-2022-34(2)-13 / Ignatyev A.O., Kalinin A.A., Mokshin S.Yu. Task and resources management function in HPC operation system «SPO Super-EVM». Trudy ISP RAN/Proc. ISP RAS, vol. 34, issue 2, 2022, pp. 159-178 (in Russia), DOI: 10.15514/ISPRAS-2022-34(2)-13.
- [9]. Игнатьев А.О., Мокшин С.Ю., Иванков Д.В., Бекетов Е.А. Пути организации параллельного доступа к структурированным данным. Труды ИСП РАН, том 35 вып. 2, 2023, стр. 111-126, DOI: 10.15514/ISPRAS-2023-35(2)-8 / Ignatyev A.O., Mokshin S.Yu., Ivankov D.V., Beketov E.A. The parallel access to structured data organization. Trudy ISP RAN/Proc. ISP RAS, vol. 35, issue 2, 2023, pp. 111-126 (in Russia), DOI: 10.15514/ISPRAS-2023-35(2)-8.
- [10]. Игнатьев А.О., Мокшин С.Ю. Типовая архитектура высокопроизводительной вычислительной системы для решения задач численного моделирования, Препринт РФЯЦ-ВНИИТФ № 265, Снежинск, 2020, 21 с. / Ignatyev A.O., Mokshin S.Yu. Base architecture of the mathematical modelling HPC system, Preprint FSUE «RFNC-VNIITF named after Academ. E.I. Zababakhin» № 265, Snezhinsk, 2020, 21 p. (in Russian).
- [11]. TOP500. Available at: <https://www.top500.org/>, accessed 01.05.2025.
- [12]. Understanding Lustre Filesystem Internals, Tech Report: ORNL/TM-2009/117, Available at: <http://wiki.lustre.org/lid/index.html>, accessed 10.05.2025.
- [13]. BeeGFS, Available at: <https://www.beegfs.io/>, accessed 10.05.2025.
- [14]. IBM Spectrum Scale. Available at: <https://www.ibm.com/support/pages/ibm-spectrum-scale/>, accessed 10.05.2025.
- [15]. Ceph, Available at: <https://ceph.io/>, accessed 10.05.2025.

- [16]. Open standard cloud computing platform. Available at: <https://www.openstack.org/>, accessed 10.04.2025.
- [17]. ОСХД Стриж. Available at: <https://vniitf.ru/data/marketing/ZOS/33.01-rukovodstvo%20programmista.pdf>, accessed 10.04.2025.
- [18]. High Performance Storage System. Available at: <https://computing.llnl.gov/projects/hpss>, accessed 10.04.2025
- [19]. Enstore. Available at: <https://github.com/Enstore-org/enstore/>, accessed 10.05.2025.
- [20]. АСХД. Программа для ЭВМ № 2018610434. Available at: <https://onlinepatent.ru/software/2018610434/>, accessed 10.04.2025.
- [21]. MPI: The Message Passing Interface. Available at: [http://parallel.ru/tech/tech\\_dev/mpi.html](http://parallel.ru/tech/tech_dev/mpi.html), accessed 01.05.2025.
- [22]. The OpenMP API specification for parallel programming. Available at: <https://www.openmp.org/>, accessed 01.05.2025.
- [23]. OpenLDAP. Available at: <https://www.openldap.org/>, accessed 01.05.2025.
- [24]. Linaro DDT Debugger. Available at: <https://www.linaroforge.com/linaro-ddt/>, accessed 01.05.2025.
- [25]. Perforce Totalview Debugger. Available at: <https://totalview.io/>, accessed 01.05.2025.
- [26]. Salome. Available at: <https://www.salome-platform.org/>, accessed 01.05.2025.
- [27]. VizIt. Available at: <https://sd.llnl.gov/simulation/computer-codes/visit/>, accessed 01.05.2025.
- [28]. Paraview. Available at: <https://www.paraview.org/>, accessed 01.05.2025.
- [29]. CentOS. Available at: <https://www.centos.org/download/>, accessed 01.05.2025.
- [30]. Rocky Linux. Available at: <https://rockylinux.org/>, accessed 01.05.2025.
- [31]. СПО Супер-ЭВМ. Available at: <https://vniitf.ru/article/spo-super-evm>, accessed 01.05.2025.
- [32]. ЗОС Арамид. Available at: <https://vniief.ru/researchdirections/civildevelopment/Aramid/>, accessed 01.05.2025.
- [33]. Eclipse. Available at: <https://www.eclipse.org/>, accessed 01.05.2025.
- [34]. NetBeans. Available at: <https://netbeans.apache.org/front/main/>, accessed 01.05.2025.
- [35]. Qt Creator. Available at: <https://www.qt.io/product/development-tools>, accessed 01.05.2025.
- [36]. The Intel® oneAPI HPC Toolkit. Available at: <https://www.intel.com/content/www/us/en/developer/tools/oneapi/hpc-toolkit.html>, accessed 10.02.2022.
- [37]. AMD ROCm. Available at: <https://www.amd.com/en/products/software/rocm.html>, accessed 10.04.2025.
- [38]. Basic Linear Algebra Subprograms. Available at: <https://netlib.sandia.gov/blas/>, accessed 10.04.2025.
- [39]. Linear Algebra PACKage. Available at: <https://netlib.sandia.gov/lapack/>, accessed 10.04.2025.
- [40]. The Fastest Fourier Transform in the West (FFTW). Available at: <https://fftw.org/>, accessed 10.04.2025.
- [41]. MPICH. Available at: <https://www.mpih.org>, accessed 30.05.2025.
- [42]. MAVAPICH. Available at: <https://mvapich.cse.ohio-state.edu/>, accessed 10.04.2025.
- [43]. OpenMPI. Available at: <https://www.open-mpi.org/>, accessed 10.04.2025.
- [44]. Библиотека S-MPI. Available at: <https://agora.guru.ru/abrau2013/pdf/338.pdf>, accessed 10.04.2025.
- [45]. Object Storage API. Swift documentation. Available at: <https://docs.openstack.org/api-ref/object-store/>, accessed 10.05.2025.
- [46]. Amazon S3 REST API Introduction – Amazon Simple Storage Service. Available at: <https://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html>, accessed 10.05.2025.
- [47]. Elasticsearch: The Official Distributed Search & Analytics Engine. Available at: <https://elastic.co/elasticsearch/>, accessed 10.05.2025.
- [48]. Zabbix – open source distributed monitoring solution. Available at: <https://zabbix.com/ru/>, accessed 10.05.2025.
- [49]. Highly Scalable Data Service (HSDS) – The HDF Group. Available at: <https://hdfgroup.org/solutions/highly-scalable-data-service-hds/>, accessed 10.05.2025.
- [50]. The HDF5 Library & File Format – The HDF Group. Available at: <https://hdfgroup.org/solutions/hdf5/>, accessed 10.05.2025.
- [51]. Иванков Д.В. Реализация функции долговременного хранения научных данных большого объема в вычислительном центре. Труды ИСП РАН, том 34 вып. 4, 2022, стр. 117-134. DOI: 10.15514/ISPRAS-2022-34(4)-9 / Ivankov D.V. Large-scale scientific data and long-term data storage function in a computing datacenter. *Trudy ISP RAN/Proc. ISP RAS*, vol. 34, issue 4, 2022. pp. 117-134 (in Russia), DOI: 10.15514/ISPRAS-2022-34(4)-9.
- [52]. GPS. Available at: <https://www.gps.gov/>, accessed 10.05.2025.



- [53]. ГЛОНАСС. Available at: [https://glonass-iac.ru/en/about\\_glonass/](https://glonass-iac.ru/en/about_glonass/), accessed 10.05.2025.
- [54]. SVN. Available at: <https://subversion.apache.org/>, accessed 10.05.2025.
- [55]. Git. Available at: <https://git.kernel.org/pub/scm/git/git.git>, accessed 10.05.2025.
- [56]. Slurm Workload Manager. Available at: <https://slurm.schedmd.com/>, accessed 10.05.2025.
- [57]. Ganglia. Available at: <https://github.com/ganglia/>, accessed 10.05.2025.
- [58]. Nagios. Available at: <https://www.nagios.org/>, accessed 10.05.2025.
- [59]. IPMI tool. Available at: <https://github.com/ipmitool/ipmitool>, accessed 10.05.2025.
- [60]. PuTTY. Available at: <https://www.putty.org/>, accessed 10.05.2025.
- [61]. Xming X Server. Available at: <https://www.straightrunning.com/XmingNotes/>, accessed 10.05.2025.
- [62]. OpenText Exceed. Available at: [www.opentext.com/products-and-solutions/products/specialty-technologies/connectivity/exceed/](http://www.opentext.com/products-and-solutions/products/specialty-technologies/connectivity/exceed/), accessed 11.12.2024.
- [63]. TurboVNC. Available at: <https://www.turbovnc.org/>, accessed 10.05.2025.
- [64]. Apache Software Foundation. Available at: <https://www.apache.org/>, accessed 10.05.2025.
- [65]. Яндекс Браузер. Available at: <https://browser.yandex.ru/>, accessed 10.05.2025.
- [66]. Postfix. Available at: <https://www.postfix.org/>, accessed 11.07.2023.
- [67]. Dovecot. Available at: <https://www.dovecot.org/>, accessed 10.05.2025.
- [68]. Jabberd2. Available at: <https://jabberd2.org/>, accessed 10.05.2025.
- [69]. MariaDB Foundation. Available at: <https://mariadb.org/>, accessed 10.05.2025.
- [70]. PostgreSQL. Available at: <https://www.postgresql.org/>, accessed 10.05.2025.
- [71]. ANSYS. Available at: <https://www.ansys.com/>, accessed 19.02.2022.
- [72]. SIEMENS. Available at: <https://www.siemens.com/global/en/products/software.html>, accessed 10.05.2025.
- [73]. Dassault Systemes. Available at: <https://dassault.fr/en/subsidiary/dassault-systemes>, accessed 10.05.2025.
- [74]. MathWorks. Available at: <https://www.mathworks.com/>, accessed 10.05.2025.
- [75]. ЛОГОС. Available at: <https://www.logos.vniief.ru/>, accessed 10.05.2025.
- [76]. OpenFOAM. Available at: <https://www.openfoam.com/>, accessed 10.05.2025.
- [77]. LAMMPS. Available at: <https://www.lammps.org/>, accessed 18.02.2022.

## **Информация об авторах / Information about authors**

Сергей Юрьевич МОКШИН – начальник отдела ВНИИ технической физики имени академика Е.И. Забабахина с 2016 года. Сфера научных интересов: проектирование вычислительных систем, разработка функциональных подсистем для высокопроизводительных вычислительных систем, разработка операционных систем, методы и средства защиты информации.

Sergey Yurievich MOKSHIN – Head of the Department of Zababakhin All-Russian Scientific Research Institute of Technical Physics since 2016. Research interests: design of supercomputer systems, development of functional subsystems for high performance supercomputing systems, operating systems development, methods and means for protecting information.

Алексей Олегович ИГНАТЬЕВ – начальник лаборатории ВНИИ технической физики имени академика Е.И. Забабахина с 1998 года. Сфера научных интересов: проектирование вычислительных систем, разработка параллельных программ численного моделирования, разработка операционных систем, методы и средства защиты информации.

Alexey Olegovich IGNATYEV – Head of the Laboratory of Zababakhin All-Russian Scientific Research Institute of Technical Physics since 1998. Research interests: design of supercomputer systems, parallel numerical simulation programs development, operating systems development, methods and means of information security.

Александр Иосифович МЕЛЬНИКОВ – ведущий научный сотрудник ВНИИ технической физики имени академика Е.И. Забабахина с 2015 года. Сфера научных интересов: высокопроизводительные вычислительные системы, разработка ПО для Linux.

Alexander Iosifovich MELNIKOV – Leader scientist of Zababakhin All-Russian Scientific Research Institute of Technical Physics since 2015. Research interests: height performance computing, Linux development.

Дмитрий Владимирович ИВАНКОВ – начальник лаборатории ВНИИ технической физики имени академика Е.И. Забабахина с 2016 года. Сфера научных интересов: проектирование многоуровневых систем хранения данных, разработка высокопроизводительных систем хранения данных, исследования методов управления данными.

Dmitry Vladimirovich IVANKOV – Head of the Laboratory of Zababakhin All-Russian Scientific Research Institute of Technical Physics since 2016. Research interests: design of tiered data storage systems, development of high performance storage systems, research in data management methods.