DOI: 10.15514/ISPRAS–2020–32(4)–3

Общие подходы к проектированию подсистемы доступа высокопроизводительных вычислительных систем

С.Ю. Мокшин, ORCID: 0000-0002-7454-6597 <sumo@rambler.ru>
Всероссийский НИИ технической физики имени академика Е.И. Забабахина,
456770, Россия, г. Снежинск, Челябинская область, ул. Васильева, 13

Аннотация. Создание высокопроизводительной вычислительной системы, предназначенной для решения задач численного моделирования различных физических процессов, является сложной и трудоемкой задачей. В статье рассматриваются основные подходы по проектированию подсистемы доступа такой высокопроизводительной вычислительной системы, позволяющие систематизировать и упростить процесс ее разработки. Проводится анализ основных факторов, оказывающих влияние на структуру и состав подсистемы доступа. Приводится пример методики расчета размерности подсистемы доступа.

Ключевые слова: высокопроизводительная вычислительная система; кластер; подсистема доступа; высокопроизводительные вычисления; моделирование.

Для цитирования: Мокшин С.Ю. Общие подходы к проектированию подсистемы доступа высокопроизводительных вычислительных систем. Труды ИСП РАН, том 32, вып. 4, 2020 г., стр. 41–52. DOI: 10.15514/ISPRAS–2020–32(4)–3

General approaches to the design of the access subsystem of high-performance computing systems

S.Yu. Mokshin, ORCID: 0000-0002-7454-6597 <sumo@rambler.ru>
E. I. Zababakhin All-Russian Scientific Research Institute of Technical Physics,
13, Vasilieva street, Chelyabinsk region, Snezhinsk, 456770, Russia

Abstract. Mathematical modeling allows to create a digital model of a product, so-called digital twin. Such digital twins are quite complex mathematical models, the description of which as well as the calculation of their parameters and characteristics is a very serious computational problem that can be solved only on high-performance computing systems. However, building of a high-performance supercomputing system designed for solving problems of computer simulation of various physical processes is a difficult time-consuming task. The paper discusses basic approaches to design a login nodes subsystem for such high-performance supercomputing system allowing to systematize and simplify the process of its development. An analysis of main factors affecting a structure and a composition of login nodes subsystem is carried out. These factors include a number of potential users of a high-performance computing system; availability of categories of users of the system and their percentage; characteristics of computing tasks solved on the system; hardware platform chosen for system building. An example of a methodology for calculating of the login nodes subsystem size is given.

Keywords: high-performance computing system; cluster; computer simulation; login nodes subsystem; HPC modeling; HPC.

For citation: Mokshin S.Yu. General approaches to the design of the access subsystem of high-performance computing systems. Trudy ISP RAN/Proc. ISP RAS, vol. 32, issue 4, 2020. pp. 41–52 (in Russian). DOI: 10.15514/ISPRAS–2020–32(4)–3

1.Введение

Любое научное или производственное предприятие в своей деятельности рано или поздно сталкивается с тем, что создание опытных образцов результатов своей деятельности, тестовых или конечных продуктов – это достаточно дорогой и затратный по времени процесс. Причем неважно, какую именно область деятельности затрагивает производственный или научный процесс – авиастроение ли это, машиностроение или атомная энергетика, медицина или фармакология и т.д. И тогда на помощь ученым и производственникам приходит математическое моделирование, которое позволяет создать цифровую модель продукта, так называемый цифровой двойник. Естественно, такие цифровые двойники представляют собой достаточно сложные математические модели, описание которых, а также расчет их параметров и характеристик, представляет собой очень серьезную вычислительную задачу, решаемую на высокопроизводительных вычислительных системах (ВВС). В данной статье мы сделаем попытку раскрыть основные подходы к проектированию одной из функциональных подсистем ВВС – подсистемы доступа, учитывая опыт создания ВВС для открытых исследований в Российском федеральном ядерном центре – Всероссийском научно-исследовательском институте технической физики имени академика Е.И. Забабахина (РФЯЦ-ВНИИТФ).

Любая высокопроизводительная вычислительная система, предназначенная для решения задач численного моделирования, состоит из множества функциональных подсистем, среди которых, как правило, присутствует подсистема доступа ВВС. На протяжении последних 20-25 лет рабочее место инженера или научного сотрудника на любом предприятии представляет собой обычный персональный компьютер (ПЭВМ) с установленной на него операционной системой Microsoft Windows какого-либо поколения. По данным Statcounter GlobalStats [1] в апреле 2020 года 76.52% персональных компьютеров использовали операционные системы (ОС) от компании Microsoft и только 1.61% – операционные системы семейства Linux. В тоже время практически на всех ВВС в качестве операционных систем используются различные сборки ОС Linux [2].

Отсюда и возникает потребность в создании своего рода промежуточного слоя между ВВС и ПЭВМ пользователя ВВС – подсистемы доступа (ПСД), реализующей функции полноценно однородного с ВВС рабочего места, оснащенном средствами разработки, компиляции программ, а также всем необходимым системным и прикладным программным обеспечением, которое обеспечивает функциональное использование ВВС. Фактически ПСД в составе ВВС выполняет функции некоего «мощной» ПЭВМ, на которой ведется разработка и отладка программ, выполняется расчет начальных данных (РНД), проводятся процедуры постобработки и визуализации расчетных данных.

Термин «подсистема доступа» возник в конце 1990-ых годов при разработке ВВС в ядерных центрах РФ. В зарубежных публикациях по соответствующей тематике обычно речь идет об отдельных вычислительных узлах, выполняющих функции доступа к ВВС, и не объединяемых в отдельную подсистему (так называемые login nodes).

В данной статье мы рассмотрим ключевые аспекты, учитываемые при создании подсистемы доступа ВВС на примере исследования, которое было проведено для ВВС «Зубр», созданной в РФЯЦ-ВНИИТФ. [3].

2. Выбор ключевых характеристик ПСД ВВС

Очевидно, что структура и состав ПСД ВВС будут зависеть от нескольких факторов:

- количества потенциальных пользователей ВВС;
- наличия категорий пользователей ВВС и их процентного соотношения;
- характеристик вычислительных задач, решаемых на ВВС;
- аппаратной платформы, выбранной для ВВС.

Рассмотрим подробнее каждый из этих факторов и их влияние на структуру и состав ПСД ВВС.

Естественно, если в организации имеется некоторое количество потенциальных пользователей ВВС, то возникает первое желание создать ПСД, используя количество узлов (серверов) ПСД равное количеству пользователей ВВС. Такой подход даже в случае исключительно хорошего финансового состояния организации будет, безусловно, экономически неэффективен и никоим образом не окупит себя, так как стоимость серверного оборудования всегда намного выше стоимости обычных персональных компьютеров. Поэтому количество узлов ПСД конечно будет пропорционально зависеть от количества пользователей ВВС, но с какими-то поправочными коэффициентами, определяющими эту зависимость. Формулу этой зависимости мы постараемся определить позднее.

Режим работы с ПСД ВВС, как правило, удаленный и подразумевает, что пользователь может обращаться к узлу ПСД по протоколам SSH [3], X [4], RFB (VNC) [5], FTP [6], используя различные программные инструменты, такие как PuTTY [7], OpenText Exceed [8], Xming [9], X2Go [10], TurboVNC [11], TightVNC [12] и многие другие. В данной статье мы не будем останавливаться на некоторых других протоколах, используемых для удаленного доступа, которые разработаны для использования с коммерческим ПО наподобие VMware [13] (протоколы Blast, PCoIP, Microsoft RDP). Подобные продукты обладают рядом очень полезных характеристик, но их исходные коды являются закрытыми, а стоимость самого ПО достаточно велика.

Подсистема доступа ВВС должна быть подключена к некоторой локальной вычислительной сети (ЛВС) предприятия. Параметры такого подключения тоже определяются исходя из количества потенциальных пользователей ВВС, но, естественно, учитывают финансовые возможности предприятия. Как правило, для подключения ПСД ВВС к ЛВС предприятия используются сети на базе Ethernet со скоростями от 1 Гбит/с до 10 Гбит/с. Стоимость других решений пока еще слишком велика для массового использования и может быть сравнима со стоимостью самой ПСД ВВС. Имеющийся у автора статьи многолетний опыт использования ПСД ВВС показывает, что на данный момент один пользователь ВВС в среднем утилизирует канал примерно в 50 Мбит/с, причем за 10 лет использования ВВС [14] ширина этого канала выросла примерно в 5 раз. Данный рост обуславливается спецификой использования удаленного графического стола, ростом графических возможностей программных оболочек ОС, программ предварительной и постобработки результатов расчетов, количеством расчетов и объемом насчитанных данных. Поэтому это значение в 50 Мбит/с можно принять в данное время в качестве минимальной базовой пропускной способности канала сети доступа на одного пользователя ВВС – E_{min}, которое попробуем использовать в дальнейшем при расчетах размерности ПСД.

Опыт использования ВВС показывает, что пользователь ВВС – это такое достаточно общее понятие, объединяющее в себе и высококвалифицированного разработчика прикладных программ, программиста, математика, и потребителя результатов расчетов – физика, инженера, экономиста и др. Конечно, потребности в составе ПО, в вычислительных ресурсах, а также интенсивность их работы с ВВС у всех этих категорий пользователей абсолютно разные. Кто-то из пользователей ВВС умеет и может заниматься созданием математических моделей исследуемых объектов, генерацией расчетных сеток для трехмерных объектов и их декомпозицией, распараллеливанием алгоритмов физико-математического моделирования различных процессов, написанием программ и их адаптацией под оборудование ВВС и т.п.

(назовем такую категорию пользователей категорией А). А кто-то использует готовое прикладное программное обеспечение для своих расчетов, не вдаваясь в тонкости создания программ и написания алгоритмов (категория Б). При проектировании ПСД важно знать, хотя бы приблизительно процентное соотношение между категориями А и Б, либо их абсолютные значения, которые мы обозначим как n_A и $n_Б$ соответственно.

Как правило, потребности в вычислительных ресурсах, как-то количество ядер CPU – N_{cpu}, количество оперативной памяти, объем необходимых файловых ресурсов для категории А в 2-3 раза больше, чем для категории Б. Такое соотношение обусловлено прежде всего необходимостью компиляции (часто параллельной), отладки программ (как правило тоже параллельной), визуализации насчитанных данных, наличии процедур расчета начальных данных. Напоминаем, что речь идет именно о подсистеме доступа, на которой, как правило, не производятся сами расчеты; на вычислительном поле ВВС это соотношение потребностей в вычислительных ресурсах может быть совсем иным.

Важно отметить еще один момент: как правило, ВВС – это многопользовательские системы. И часто, помимо указанных категорий пользователей А и Б, в них формируются и устойчиво существуют так называемые тематические группы пользователей. Объединение пользователей в такие группы подразумевает решение ими общих для конкретной тематической группы вычислительных задач и, следовательно, общей потребности в определенном количестве вычислительных ресурсов и составе прикладного и системного ПО. В том числе, введение таких тематических групп в ВВС проводится с целью разграничения доступа к вычислительным и файловым ресурсам ВВС.

Безусловно, на структуру ПСД ВВС влияют и характеристики вычислительных задач, предполагаемых к решению на ВВС. Не секрет, что большинство потребителей ВВС старается использовать для решения своих конкретных математических, конструкторских или иных задач готовое прикладное программное обеспечение, в которое заложены существующие и известные десятилетиями математические алгоритмы. Это вполне понятно и оправдано, так как придумывать самому алгоритмы и писать программы достаточно сложно, требует наличия высокой квалификации у разработчика; фактически это малоэффективно и дорого.

Исходя из этого, при проектировании ПСД ВВС и ВВС, требуется провести анализ необходимого к использованию на ВВС прикладного и системного ПО, определить все зависимости такого ПО от сторонних оптимизирующих библиотек (BLAS [15], SCALAPACK [16], PLASMA [17], MKL [18] и др.), его возможности в плане использования тех или иных средств распараллеливания (pthreads [19], OpenMP [20], MPI [21], Linda [22] и т.д.), возможности в плане использования на специализированных ускорителях вычислений, например программируемых логических матрицах, либо графических ускорителях и т.п.

Кроме того, важно учитывать характеристики ввода-вывода прикладного ПО, касающиеся его работы с системой хранения ВВС (формат выводимых данных, возможность параллельного чтения и записи данных, количество операций ввода-вывода в секунду, объемы формируемых файлов, частота обращений к ним, наличие контрольных операций с файлами и т.п.).

Только на основе такого достаточно глубокого и многофакторного анализа, делается вывод о ключевых параметрах ВВС – архитектуре процессоров ВВС, необходимости в ускорителях, архитектуре высокопроизводительной коммуникационной среды (Ethernet, Infiniband, Omni-Path и т.д.), количестве и типе необходимой оперативной памяти, объемах и типах системы хранения ВВС, операционной системе ВВС и составе системного и прикладного ПО. И уже после этого, на основе разработанной общей архитектуры ВВС, выбранной для ВВС аппаратной платформы, формируется вывод и об общей архитектуре ПСД ВВС. В рамках данной статьи мы не будем описывать процедуру выбора аппаратной платформы для ВВС,

отметим лишь что она, как правило, строится из расчета требуемой реальной или пиковой производительности ВВС.

Учитывая все вышеперечисленные факторы, можно сделать вывод, что ПСД ВВС должна полноценно реализовывать все необходимые для её пользователя функции – разработку приложений, их компиляцию, отладку, постобработку результатов расчетов, но с обязательным учетом всех заложенных в общую архитектуру ВВС особенностей – её аппаратной конфигурации, типа и версии операционной системы, коммуникационного ПО. Такая жесткая взаимозависимость ПСД от ВВС особенно справедлива для ВВС, где присутствует пользователь-разработчик прикладного ПО (категория А) и в целом обусловлена тем, что любое прикладное ПО, во-первых, имеет определенные программные зависимости, и просто может не работать, например, на ОС, отличной от типа и версии ОС, на которой было собрано это ПО (даже в независимости от вида сборки ПО – динамической или статической). Те же проблемы с запуском и корректной работой прикладного ПО наблюдаются так же при отсутствии другого необходимого системного окружения, различных математических и оптимизирующих библиотек, компиляторов, библиотек для задания формата данных и т.д.

Во-вторых, для того, чтобы максимально эффективно использовать вычислительные ресурсы ВВС, прикладное ПО должно быть оптимизировано под архитектуру ВВС (процессоров, ускорителей, коммуникационной среды и т.д.) По опыту автора публикации, оптимизация прикладного ПО – процесс трудоемкий и многоэтапный, и подразумевает прямой и интерактивный режим работы с оборудованием ВВС. На первом этапе производится исследование архитектуры процессора или ускорителя, изучения свойств и характеристик оперативной памяти и памяти устройств. На этом этапе пользователь ВВС должен иметь возможность взаимодействия с оборудованием ВВС минуя промежуточный программный слой в виде набора ПО системы управления ВВС или коммуникационных библиотек, которые всегда присутствуют на вычислительном поле ВВС. Именно на этом этапе функцию рабочего места должна выполнять ПСД в ВВС, реализуя функции ПЭВМ с прямым доступом к оборудованию.

На втором этапе обычно исследуются механизмы распараллеливания ПО, прикладное ПО переписывается при необходимости и с учетом тех возможностей, которые дают коммуникационная среда и система управления ВВС. На третьем этапе, когда программа каким-то образом уже оптимизирована, её дальнейшая отладка и адаптация непосредственно происходит на вычислительном поле ВВС.

Таким образом, можно сформулировать ключевое правило построения ПСД ВВС – архитектура процессоров и выбранные типы ускорителей, а также состав системного и прикладного ПО должны быть, как минимум, аналогичными используемым в ВВС в целом. Многолетний опыт вычислительных расчетов в РФЯЦ-ВНИИТФ показывает, что остальные характеристики ПСД достаточно независимы от ВВС и определяются ранее названными факторами (количества пользователей, наличия их категорий, характеристик прикладного ПО для ПСД).

3. Общая структура ПСД ВВС

Поскольку подсистема доступа является неотъемлемой частью ВВС, для понимания того, каким образом должна выглядеть общая структура ПСД, надо рассмотреть структуру типовой ВВС (рисунок 1).

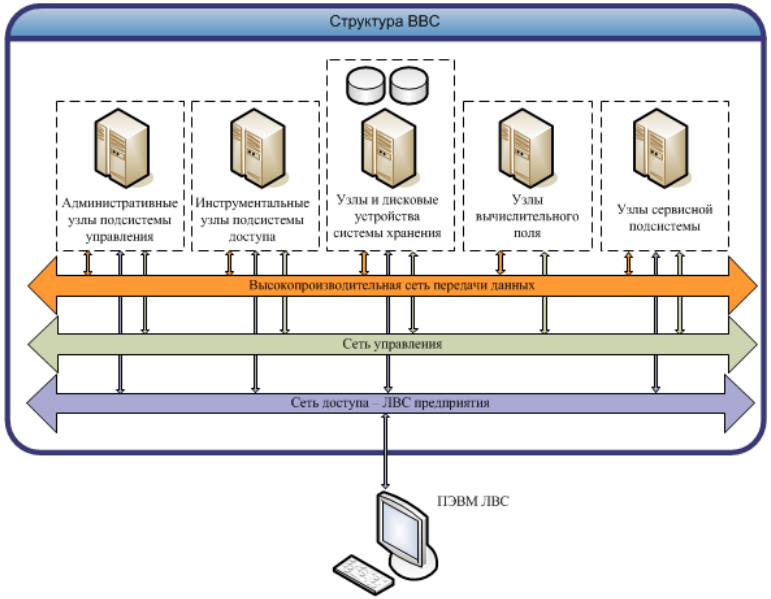

Рис. 1. Общая структура ВВС
Fig.1. High-performance supercomputer structure

С учетом сложившейся общемировой практики построения ВВС, структура ВВС может выглядеть так, как представлено на рис. 1. Она состоит из подсистемы доступа, системы хранения, подсистемы управления, сервисной подсистемы и вычислительного поля, объединенных высокопроизводительной коммуникационной средой (высокопроизводительной сетью передачи данных) и сетью управления. Кроме того, обычно подсистема управления, сервисная подсистема и подсистема доступа объединены с ЛВС предприятия (организации) через сеть доступа.

Предлагаемая к рассмотрению общая структура ПСД ВВС представлена на рис. 2. ПСД ВВС состоит из некоторого количества инструментальных узлов (ИУ) F1…FN, объединенных сетью управления (СУ), высокопроизводительной коммуникационной средой (ВКС) и сетью доступа, которая в свою очередь должна быть сопряжена с ЛВС предприятия (организации). Термин «узел», в данном контексте используется для единицы вычислительного оборудования и отличается от термина «сервер», так как физически современный сервер может состоять из нескольких вычислительных модулей – узлов, объединенных, в одном корпусе с одним, например, блоком питания. Таких модулей, в серверах на данный момент времени, например, может быть от двух до восьми в корпусах размерности 1U и 2U соответственно (Twin server, Twin^2 server, Twin^3 server).

В зависимости от требуемого количества инструментальных серверов, потребность в которых мы попробуем рассчитать чуть позже, ПСД может быть построена с использованием механизмов бездисковой или дисковой загрузки ОС. Бездисковая загрузка используется для удобства администрирования ПСД при большом количестве инструментальных узлов (как правило, от 10 и более), и позволяет хранить образ системы на одном системном узле (СУ) ПСД, загружая его на инструментальные узлы с помощью протоколов и инструментов PXE, TFTP, DHCP, NFS.

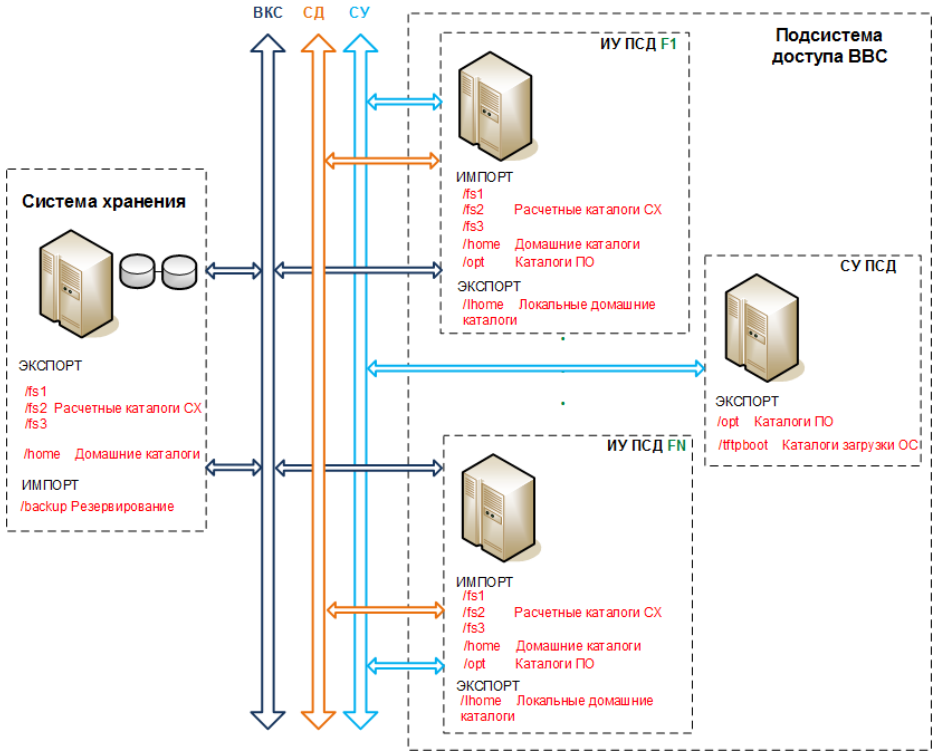

Рис. 2. Общая структура ПСД ВВС

Fig.2. High performance supercomputer access subsystem structure

На инструментальных узлах создаются каталоги/точки монтирования сетевых файловых систем – каталоги расчетных данных и домашние каталоги. Домашние каталоги реализуют функцию личного хранилища данных, а также точки входа на ВВС, с помощью которой можно задавать системное окружение, глобальные настройки для конкретного пользователя на всем пространстве ВВС. Как правило, домашний каталог содержит настройки окружения в конфигурационных файлах командной оболочки (например, для широко используемой Bourne-Again Shell (BASH) это файлы .bashrc и .bash_profile), настройки рабочего стола, файловых менеджеров и редакторов, компиляторов и отладчиков, исходные тексты программ в случае их разработки на ПСД. Существует два подхода к организации домашних каталогов:

- их можно размещать на локальных дисках каждого инструментального узла (например, используя массивы RAID 1, RAID 10, RAID 5, RAID 6),
- их можно размещать на файловой системе системы хранения ВВС (СХ) и экспортировать по протоколам сетевой файловой системы.

Первый подход можно использовать для небольших ВВС, так как необходимо помнить, что точка входа пользователя на любой узел ВВС должна быть одинаковой на всем пространстве ВВС (для единообразия настроек окружения). В этом случае обычно на каждом инструментальном узле запускается NFS сервер, обеспечивающий экспорт домашних каталогов на вычислительное поле. В случае если ВВС содержит в своем составе множество узлов (клиентов) вычислительного поля (как правило, более 200), такая схема становится

неудобной и неспособной обеспечить требуемую функциональность из-за слишком большого количества клиентских запросов. Собственно, эти ограничения накладываются самим протоколом NFS и не зависят от версии этого протокола.

Гораздо более удобной является схема размещения домашних каталогов на параллельной файловой системе, способной обеспечивать множественные клиентские запросы (например, Lustre). Кроме того, домашние каталоги могут быть построены не на обычных серверных RAID системах (с использованием дисковых серверов), а на дисковых подсистемах профессионального уровня, что обеспечивает их сохранность и доступность, а также упрощает функции масштабирования и резервирования с использованием достаточно больших объемов дисковых подсистем.

Аналогичный подход используется и при организации расчетных каталогов; для больших ВВС целесообразно использовать несколько файловых систем в качестве каталогов расчетов, это облегчает их обслуживание, снижает загрузку на конкретную файловую систему при интенсивном ее использовании расчетными программами. В этом случае на инструментальных узлах ПСД создается несколько точек монтирования для сетевых файловых систем.

Использование сетевых файловых систем для домашних каталогов, несомненно, имеет огромные плюсы в плане унификации, администрирования, обеспечения сохранности данных и т.д. Но есть и некоторые минусы. Таковым, например, является неудобства, возникающие у разработчиков программ при проведении процедуры параллельной компиляции программ. Такая потребность возникает часто для больших программных продуктов, где количество файлов исходных текстов может достигать нескольких тысяч и более. В этом случае скорость работы с файловой системой, на которой одновременно несколько пользователей может совершать подобную процедуру, становится неприемлемой.

Кроме того, при параллельной компиляции сетевая файловая система должна обеспечивать функцию временной блокировки файла (flock); не каждая файловая система может обеспечить такой режим без существенного замедления работы. В этом случае, помимо обычных домашних каталогов, обеспечивающих функцию точки входа, автором статьи была предложена идея использовать специализированные локальные домашние каталоги, обеспечивающие максимальную скорость работы с локальными файловыми системами.

Современные технологии позволяют реализовывать локальные RAID массивы из быстрых SSD дисков (различного типа и форм-фактора: SATA, M2, U2, Intel Optane и т.д.), обеспечивающие скорости чтения до 5000 Мбит/с и записи до 3500 Мбит/с, объемы до 10 ТБ и время наработки на отказ до 2 млн. часов. Конечно, в этом случае должно быть предусмотрено резервирование всей ценной информации, хранимой в локальных домашних каталогах. Такой локальный домашний каталог доступен только в пределах каждого инструментального узла ПСД, экспорт данных с его файловой системы на другие узлы ВВС обычно не предусматривается.

Для унификации общедоступного системного и прикладного программного обеспечения целесообразно на системном узле ПСД создавать каталог, который будет содержать весь набор неизменяемого прикладного и системного ПО: расчетных программ, библиотек оптимизации, компиляторов, отладчиков и т.д. Данный каталог обычно экспортируется по протоколу NFS на все инструментальные узлы ПСД, его содержимое синхронизируется встроенными в ОС инструментами типа Rsync на системные узлы вычислительного поля, экспортирующие в свою очередь по протоколу NFS эти данные на вычислительные узлы (пример – каталог /opt на рис. 2).

Таким образом, для создания ПСД в качестве вычислительного оборудования должно выбираться оборудование, учитывающее особенности организации технологии работы

пользователей на ПСД. Как минимум, требования к вычислительному оборудованию могут быть такими:

- серверная платформа должна позволять использовать процессоры однородной архитектуры с процессорами, используемыми на вычислительном поле;
- серверная платформа должна иметь не менее двух портов Ethernet для подключения к сети управления и сети доступа;
- серверная платформа должна позволять устанавливать адаптеры высокопроизводительных сетей (например, Infiniband), либо иметь такие интегрированные порты;
- центральный процессор, выбираемый для использования на инструментальном узле ПСД, должен иметь достаточно высокую тактовую частоту ядра (более высокую или сравнимую с частотой ядра процессора обычного ПЭВМ в ЛВС);
- количество памяти на инструментальном узле должно выбираться из соотношения:

$$M = M_0 * (n_А + kn_Б),$$

где

$n_А$ – количество пользователей категории А в одной тематической группе;

$n_Б$ – количество пользователей категории Б в одной тематической группе;

k – условный коэффициент сложности работ, определяемый необходимостью решения на ПСД некоторого количества N дополнительных задач (РНД, задач визуализации и т.д., $k = 1..N$);

M_0 – величина, соотносимая или равная размеру памяти на используемых в ЛВС предприятия (организации) ПЭВМ.

В зависимости от наличия в ВВС пользователей-разработчиков, их потребностей, требования к оборудованию, используемому в качестве инструментальных узлов ПСД, могут быть расширены до следующих:

- серверная платформа должна иметь возможность установки полноразмерного графического ускорителя или видеокарты (либо нескольких единиц этого оборудования) для работы с мощными графическими приложениями;
- серверная платформа должна иметь возможность установки ускорителей той же архитектуры, которые используются (если используются) на вычислительном поле ВВС для возможности разработки и интерактивной отладки прикладных программ, адаптированных под архитектуру ускорителей;
- серверная платформа должна иметь возможность установки RAID контроллера, поддерживающего RAID 1, RAID 10, RAID 5, RAID 6 для организации локального файлового пространства, в том числе, если требуется и на SSD накопителях разного типа и форм-фактора.

Из всего вышеуказанного набора требований можно сделать вывод, что серверная платформа, выбираемая для использования в качестве инструментальных узлов ПСД, в некоторых случаях будет иметь достаточно высокую стоимость, поэтому очень важно на этапе проектирования примерно определить количество инструментальных узлов, удовлетворяющее требованиям заказчика ВВС.

4. Примерный расчет размерности ПСД ВВС

При проектировании ВВС в стоимость проекта закладывается стоимость всех создаваемых подсистем ВВС, поэтому естественно важно каким-то образом определить количество единиц оборудования, необходимых для создания ПСД ВВС. Расчет размерности (количества инструментальных узлов) ПСД должен строиться на базе уже выбранной для ВВС аппаратной платформы (модели процессора, модели серверов, типа коммуникационной

среды). Исходя из всего вышеизложенного, можно предложить несколько простейших зависимостей для расчета размерности ПСД ВВС.

Во-первых, учитывая наличие нескольких категорий пользователей, примерное количество инструментальных узлов может быть рассчитано по формуле:

$$F_1 \approx \sum_{i=1}^{G} \frac{(n_А + kn_Б)_i}{N_{fcpu}} N_{cpu},$$

где

F_1 – минимальное количество инструментальных узлов в составе ПСД;

G – общее количество тематических групп пользователей;

$n_А$ – количество пользователей категории А в одной тематической группе;

$n_Б$ – количество пользователей категории Б в одной тематической группе;

k – условный коэффициент сложности работ, определяемый необходимостью решения на ПСД некоторого количества N дополнительных задач (РНД, задач визуализации и т.д., $k = 1..N$);

N_{cpu} – минимально достаточное количество вычислительных ядер на инструментальном узле для сеанса одного пользователя ПСД;

N_{fcpu} – максимально доступное количество вычислительных ядер на инструментальном узле ПСД.

Значение N_{fcpu} рассчитывается как произведение физически возможного количества процессоров в вычислительном узле (сервере) и количества ядер одного процессора. Значение N_{cpu} выбирается эмпирическим путем. Например, его можно определить равным количеству ядер процессора на ПЭВМ в составе сети предприятия. Фактически значение N_{cpu} определяет комфортные условия работы пользователя в сеансе на ПСД.

Во-вторых, мы должны выбрать такое количество инструментальных узлов в составе ПСД, которое бы позволяло обеспечить минимальную базовую пропускную способность канала сети доступа для одного пользователя ВВС:

$$F_2 \approx \sum_{i=1}^{G} \frac{(n_A + kn_Б)_i}{E} E_{min},$$

где

F_2– минимальное количество инструментальных узлов в составе ПСД;

G – общее количество тематических групп пользователей;

$n_А$ – количество пользователей категории А в одной тематической группе;

$n_Б$ – количество пользователей категории Б в одной тематической группе;

k – условный коэффициент сложности работ, определяемый необходимостью решения на ПСД некоторого количества N дополнительных задач (РНД, задач визуализации и т.д., $k = 1..N$);

E_{min} – минимальная базовая пропускная способность канала сети доступа для одного пользователя ВВС;

E – пропускная способность канала сети доступа одного инструментального узла ПСД (физическая скорость канала).

Таким образом, количество необходимых для ПСД узлов можно определить из этих двух приблизительных значений, выбрав максимальное:

$$F \approx \max\{F_1, F_2\}.$$

Безусловно, такая методика расчета является приблизительной и служит для определения стоимости оборудования ПСД при создании проекта ВВС.

5. Заключение

Рассмотренные в данной статье подходы по построению подсистемы доступа высокопроизводительных вычислительных систем были использованы автором публикации

при проектировании нескольких ВВС в РФЯЦ-ВНИИТФ, в том числе по ряду контрактных работ по созданию ВВС для научных организаций РФ. Расчет размерности ПСД по приведенным в статье формулам позволил РФЯЦ-ВНИИТФ упростить проектирование ВВС для нужд своих заказчиков. Автор выражает надежду, что изложенные в статье подходы могут быть полезны и другим специалистам, занимающимся созданием ВВС.

Список литературы / References

[1]. GlobalStats. Available at: https://gs.statcounter.com/os-market-share/desktop/worldwide, accessed 01.06.2020.
[2]. Top500 Supercomputers. Available at: https://www.top500.org/, accessed 01.06.2020.
[3]. Глазырин А.И., Мокшин С.Ю. Суперкомпьютер «Зубр» средней производительности. Труды Всероссийской конференции «Информационные технологии в оборонно-промышленном комплексе, 2016 г. / Glazyrin A.I., Mokshin S.Yu. Supercomputer «Zubr» of average performance. In Proc. of the All-Russian conference "Information technologies in the military-industrial complex, 2016 (in Russian).
[4]. The Secure Shell (SSH) Protocol Architecture. Available at: https://tools.ietf.org/html/rfc4251, accessed 01.06.2020.
[5]. XWindow System Protocol. Available at: ftp://ftp.x.org/pub/X11R7.0/doc/PDF/proto.pdf, accessed 01.06.2020.
[6]. The Remote Framebuffer Protocol. Available at: http://www.realvnc.com/docs/rfbproto.pdf, accessed 01.06.2020.
[7]. File Transfer Protocol. Available at: https://tools.ietf.org/html/rfc959, accessed 01.06.2020.
[8]. PuTTY. Available at: https://www.putty.org/, accessed 01.06.2020.
[9]. OpenText Exceed. Available at: https://www.opentext.com/products-and-solutions/products/specialty-technologies/connectivity/exceed, accessed 01.06.2020.
[10]. Xming X Server. Available at: http://www.straightrunning.com/XmingNotes/, accessed 01.06.2020.
[11]. X2Go. Available at: https://wiki.x2go.org/doku.php, accessed 19.05.2020.
[12]. TurboVNC. Available at: https://www.turbovnc.org/, accessed 01.06.2020.
[13]. TightVNC. Available at: https://www.tightvnc.com/, accessed 01.06.2020.
[14]. VMware Product. Available at: https://www.vmware.com/, accessed 24.05.2020.
[15]. Basic Linear Algebra Subprograms. Available at: https://www.netlib.org/blas/, accessed 24.05.2020.
[16]. Scalable Linear Algebra Package. Available at: https://www.netlib.org/scalapack/, accessed 01.06.2020.
[17]. Parallel Linear Algebra Software for Multicore Architectures. Available at: https://bitbucket.org/icl/plasma/src/default/, accessed 01.06.2020.
[18]. Intel® Math Kernel Library. Available at: https://software.intel.com/content/www/us/en/develop/tools/math-kernel-library.html, accessed 01.06.2020.
[19]. The Open Group Base Specifications Issue 6, pthread.h. Available at: http://www.opengroup.org/onlinepubs/007904975/basedefs/pthread.h.html, accessed 01.06.2020.
[20]. The OpenMP API specification for parallel programming. Available at: https://www.openmp.org/, accessed 01.06.2020.
[21]. MPI: The Message Passing Interface. Available at: http://parallel.ru/tech/tech_dev/mpi.html, accessed 01.06.2020.
[22]. Tutorial on Parallel Programming with Linda. Available at: http://www.cse.chalmers.se/edu/course/TDA384_LP1/files/lectures/semantic-linda-biglecture.pdf, accessed 01.06.2020.

Информация об авторах / Information about authors

Сергей Юрьевич МОКШИН – начальник отдела. Сфера научных интересов: проектирование вычислительных систем, разработка функциональных подсистем для высокопроизводительных вычислительных систем, разработка операционных систем, методы и средства защиты информации.

Sergey Yureivich MOKSHIN – Head of the Department. Research interests: design of supercomputer systems, development of functional subsystems for high performance supercomputing systems, operating systems development, methods and means for protecting information.