УДК 004.07; 004.63; 004.252 doi: 10.15827/0236-235X.148.472-486 2024. Т. 37. № 4. С. 472–486

Ожидания от производительности вычислительного кластера при выборе параллельной файловой системы

О.С. Аладышев [1, 2]✉, А.В. Захарченко [1, 2], В.Ф. Огарышев [1, 2], Б.М. Шабанов [1, 2]

[1] Межведомственный суперкомпьютерный центр РАН, г. Москва, 119334, Россия
[2] Национальный исследовательский центр «Курчатовский институт», г. Москва, 123182, Россия

Ссылка для цитирования
Аладышев О.С., Захарченко А.В., Огарышев В.Ф., Шабанов Б.М. Ожидания от производительности вычислительного кластера при выборе параллельной файловой системы // Программные продукты и системы. 2024. Т. 37. № 4. С. 472–486. doi: 10.15827/0236-235X.148.472-486
Информация о статье
Группа специальностей ВАК: 2.3.5
Поступила в редакцию: 20.06.2024 После доработки: 27.08.2024 Принята к публикации: 30.08.2024

Аннотация. При создании высокопроизводительного вычислительного кластера одной из наиболее актуальных задач является обеспечение производительной внешней системы хранения данных под будущую рабочую нагрузку. В статье исследуются принципы работы параллельных файловых систем, которые могут определять их производительность для различных рабочих нагрузок, и предлагается метод определения пределов производительности внешних систем хранения данных. Основным преимуществом предлагаемого метода является поэтапность выявления пределов производительности. Сначала определяются пределы аппаратной части (инфраструктуры) на основе теоретических расчетов. Потом пределы уточняются с помощью тестов и/или показаний системы мониторинга аппаратной части системы хранения данных. Вместе с выбором файловой системы и конфигурированием аппаратной части формируются программные факторы, которые могут влиять на производительность файловой системы для требуемой рабочей нагрузки. С помощью различных предметных тестов или моделей пределы продолжают уточняться. В конечном итоге выявленные пределы проверяются в предварительно настроенной под требуемую рабочую нагрузку системе хранения данных. Предлагаемый подход к выбору параллельной файловой системы для высокопроизводительного вычислительного кластера и к настройке системы хранения данных для определенного спектра параллельных суперкомпьютерных приложений позволяет обойтись без применения сложных моделей и необходимости анализа больших объемов результатов тестирования. Кроме того, он помогает лучше понять характеристики создаваемой системы хранения данных. Авторский метод поэтапной оценки производительности параллельных файловых систем позволяет упростить и ускорить процесс разработки системы хранения данных с параллельной файловой системой. Метод так же хорошо может работать и с современными специализированными файловыми системами, динамически создаваемыми для суперкомпьютерного приложения.
Ключевые слова: высокопроизводительный вычислительный кластер, параллельные файловые системы, внешние системы хранения данных, суперкомпьютеры
Благодарности. Работа выполнена в МСЦ РАН и НИЦ «Курчатовский институт» по теме FNEF-2024-0016

Введение. С момента появления суперкомпьютеров количество процессоров, вычислительных устройств и узлов постоянно росло. Еще в начале 90-х годов основные позиции занимали суперкомпьютеры, состоящие из единого многопроцессорного узла на общей памяти (*Symmetric Multiprocessing* – SMP), а требования к системам хранения данных основывались на принципе централизации. Сейчас в рейтинге TOP500 не найти SMP-системы, да и требования к системам хранения данных существенно изменились. Количество вычислительных узлов в суперкомпьютере многократно увеличилось, а системы хранения стали внешними по отношению к вычислительным узлам. Появились и специализированные *файловые системы* (ФС) ad-hoc [1], создаваемые только на время работы суперкомпьютерного приложенияю.

Современные *параллельные ФС* (ПФС) состоят из нескольких серверов, обслуживающих единое адресное пространство данных. Каждый сервер может предоставлять доступ одновременно к данным и метаданным, а также только к данным или только к метаданным. Непосредственные носители данных обычно распределяются по серверам данных; программная часть системы хранения разделилась на серверы и клиенты, образуя кластер серверов ФС и вычислительное поле. Требования к таким системам постоянно меняются в зависимости от потребностей приложений, однако основные принципы работы остаются неизменными. Сегодня наиболее актуальным остается принцип масштабируемости. Возможность масштабирования основана в первую очередь на методах разделения данных между параллельными процессами. При увеличении пределов масштабирования увеличиваются и пределы производительности.

В настоящей статье рассматриваются принципы построения ФС для высокопроизводительных вычислительных кластеров, которые

тем или иным способом влияют на время выполнения операций ввода-вывода в приложении, то есть на производительность системы хранения.

Общие требования к ПФС

ПФС в первую очередь должны соответствовать всем основным правилам POSIX. Их нарушение обычно приводит к значительному уменьшению совместимости, и тем самым обрекает ФС на ограниченное использование. Но для больших вычислительных кластеров некоторыми принципами поступаются в угоду производительности [2], например, штампом времени доступа к файлу (atime), который для ПФС не имеет смысла.

На втором месте стоит требование масштабируемости ПФС. Оно относится и к носителям данных, и к механизмам разделения доступа к самим данным. Масштабируемость напрямую влияет на предел производительности.

Следующими требованиями к ПФС являются независимая работа вычислительных узлов над операциями чтения и записи с непересекающимися областями общего файла, а также конкурентные операции записи и чтения различных файлов. Существует множество методов решения этих задач, и выбор наилучшего остается препятствием для команд разработчиков. Основной проблемой является выбор механизма блокировки.

Традиционно суперкомпьютерные приложения, работающие в кластере, требуют параллельного доступа с различных узлов как к общим, так и к отдельным файлам [3]. Другие приложения, например, обработка и поиск информации в больших данных, характеризуются параллельным доступом к отдельным файлам, и преимущественно для чтения. В этом классе приложений общий доступ к отдельным файлам не требуется, но, когда эти файлы расположены в общих каталогах или на общем диске, параллельный доступ к системной информации все равно необходим. Отсюда требование: ПФС должна поддерживать параллельный доступ не только к данным, но и к метаданным. В кластерной системе административная работа, такая как добавление диска в ПФС или оптимизация расположения данных на дисках, является непростой задачей. Эти операции могут влиять на производительность ввода-вывода, даже если они выполняются в фоновом режиме. Увеличение степени их параллелизма может компенсировать накладные расходы в ПФС и тем самым снизить влияние на производительность.

Закладываемые в ПФС правила по выполнению общих требований и обычно не изменяемые после инсталляции, являются неотъемлемой частью системы хранения и имеют основное влияние на пределы производительности. Поэтому можно считать, что при выборе ПФС для указанной аппаратной архитектуры системы хранения данных определяется и уровень снижения пределов производительности.

Типы ПФС

Для лучшего понимания пределов производительности разделим системы хранения данных высокопроизводительного вычислительного кластера на несколько типов. Из внешних систем хранения остановимся на двух основных типах ФС:

– централизованные, в которых все запросы на метаданные проходят через единую точку;

– распределенные, в которых метаданные, как и данные, распределены между множеством точек доступа к данным.

Разнообразие распределенных систем хранения определяется разнообразием типов рабочих нагрузок в вычислительной системе. Под традиционную нагрузку (последовательное чтение и запись большими порциями данных) разработаны GPFS (*General Parallel File System*), Lustre (*Linux Cluster*), CXFS (*Clustered External File System*), GFS (*Global File System*).

ПФС Lustre [4], как и многие другие системы этого класса, появилась в суперкомпьютерах кластерного типа для обеспечения контрольной точки. Разделение функций хранения данных и управления метаданными является основной концепцией этой ПФС. Хотя данные и распределены по серверам данных (*Object Storage Servers* – OSS), метаданные плохо распределяются по разным точкам доступа и в ПФС используется только один сервер метаданных, продублированный для надежности, а не для производительности. ПФС Lustre очень хорошо подходит для процессов массовой параллельной записи/чтения в разные файлы/объекты или в разные области одного файла/объекта.

Сетевые ФС, основанные на принципе pNFS [5], как и ПФС Lustre, распределяют данные по различным серверам. Метаданные, а также структура данных для серверов данных могут кэшироваться на узлах, считывающих

данные. Если метаданные устареют, сервер метаданных попросит клиента перечитать карту распределения данных. Расширение pNFS разделяет метаданные и данные между узлами ввода-вывода. Сервер метаданных централизован, как и в Lustre.

Распределенная ФС HDFS (*Hadoop Distributed File System*) [6] предназначена для обработки больших данных с использованием MapReduce. Технология MapReduce не ориентирована только на HDFS и может использовать в качестве ФС Lustre, SS3, KFS и т.п. Подобно Lustre и pNFS HDFS состоит из двух типов узлов ФС – метаданных и данных. HDFS хранит блоки данных в трех или более копиях на разных узлах данных и позволяет выполнять запись только в конец файла, оставляя исходные данные неизменными. Парадигма обработки больших данных подразумевает перенос выполняющегося программного кода к хранимым данным, что делает такие ФС непригодными для традиционных суперкомпьютерных нагрузок.

В этой работе новые ФС, такие как SuperFS, OceanFS, DAO, ParaStor, flashfs, рассматриваются бегло. Более специализированные ФС, предназначенные для узкого спектра прикладной нагрузки, такие как Ceph, PufferScale, HEPnOS, UnifyFS, Hercules, являются наложенными на базовую архитектуру какой-либо системы хранения, и предел производительности прежде всего зависит от нее. Эти ФС по-прежнему остаются статичными и конфигурируются перед запуском. С появлением более производительных вычислительных систем, которые могут гибко управлять доступными ресурсами для удовлетворения требований различных рабочих нагрузок, системы ad-hoc также должны стать гибкими [1].

Таким образом, каждый тип ПФС имеет свой характер влияния на пределы производительности. Например, ПФС, предназначенные для обработки больших данных, в разы снижают пределы для операций вывода. Это влияние имеет смысл оценивать отдельно от других факторов и после того, как будет оценено влияние основных принципов ПФС.

Работы по оценке производительности

Производительность ПФС можно оценивать исходя из наблюдений за ее поведением при работе различных приложений. Такого рода мониторинг не должен ухудшать ни основную характеристику суперкомпьютера – производительность, ни сбалансированность для традиционных суперкомпьютеров, где параллельные процессы синхронизированы. Так, в работе [3] с помощью системы мониторинга произведен анализ профилей выполнения пользовательских программ и показана степень влияния характеристик внешней системы хранения суперкомпьютера кластерного типа на время выполнения суперкомпьютерных приложений.

Оценку производительности ПФС с учетом рабочей нагрузки или без нее проводят и с помощью различных моделей систем хранения. Так, в работах [7, 8] моделируется среднее время отклика жесткого диска как элемента системы хранения, в [9] с помощью аналитического моделирования прогнозируется пропускная способность дискового массива. Исследование [10] посвящено использованию статистической модели для прогнозирования времени отклика систем хранения как черного ящика. В [11] прогнозируются производительность устройств хранения как черных ящиков с помощью CART-моделей, обученных на время отклика на запрос, и характеристики рабочей нагрузки. В работе [12] исследуется использование табличных моделей для измерения производительности дискового массива в других точках. Рассмотренные в указанных работах подходы применимы и к более сложным системам хранения, где используется иерархия носителей информации.

Оценкой факторов, таких как количество серверов данных, журналирование изменений, наличие кэша, степень гранулирования записываемых данных и других, влияющих на производительность ввода-вывода для конкретной ПФС, занимаются давно. Например, в работе [13] показано, что избыточное чередование данных в ФС может снижать производительность ПФС Lustre. В [14] рассматривается иерархическая конфигурация системы хранения, оценивается производительность отдельных компонентов системы хранения и предлагается улучшение масштабируемости суперкомпьютерного приложения путем эмпирического анализа и изменения схемы распределения файлов. Авторы предлагают объединять объекты (файлы) таким образом, чтобы снизить накладные расходы, связанные с большой шириной полосы пропускания устройств хранения данных. Работа [15] посвящена активным системам хранения, в которых простаивающие ресурсы серверов ПФС используются для некоторых операций с рядом лежащими данными. Тем самым

предлагается сократить объем перемещаемых между хранилищем и вычислительными узлами данных. Авторы [16, 17] воспользовались тестом IOR (от LLNL) для имитации рабочей нагрузки ключевых приложений с интенсивным вводом-выводом, чтобы оценить характеристики ПФС. В [18] описывается механизм группового (секционного) объединения запросов от параллельных процессов суперкомпьютерного приложения для уменьшения накладных расходов на синхронизацию ввода-вывода в ПФС Lustre и для увеличения утилизации пропускной способности системы хранения. Авторы исследуют влияние предлагаемого механизма на производительность системы хранения и показывают существенное уменьшение накладных расходов. В [19] показано, что механизмы разделения доступа к данным между параллельными процессами в ПФС Lustre и ROMIO – реализация функций ввода-вывода MPI для приложений – не согласуются между собой. Также установлено, что максимальная производительность ПФС Lustre достигается при одновременной записи параллельными процессами в независимые файлы.

В работе [20] проведен подробный анализ литературы и подобран внушительный список факторов, влияющих на производительность ПФС Lustre при различных конфигурациях системы хранения данных на ее базе. Рассматриваются следующие факторы (настройки или конфигурации) систем хранения данных: *факторы серверов метаданных* (количество серверов данных, наличие кэша на чтение/запись, тактика распределения блоков по дискам и серверам, количество параллельных нитей на сервере, количество памяти, стратегия управления метаданными, факторы дисковой подсистемы метаданных); *факторы дисковой подсистемы* (журналирование, типы носителей, шаблоны распределения данных по носителям, структура дисковой подсистемы, размер и число индексных дескрипторов файла); *факторы серверов данных* (количество дисков и дисковых подсистем, количество параллельных нитей, обслуживающих потоки данных, пропускная способность шин сервера, кэш); *факторы сетевой среды* (тип сети, количество каналов на сервер данных, пропускная способность); *факторы клиентов файловой системы* (кэш, шаблоны доступа к данным: последовательное или случайное чтение/запись, большие или маленькие порции, разделение или совмещение данных, характер запросов ввода-вывода); *факторы клиентской сетевой среды* (тип сети и пропускная способность, агрегирование или сегментирование каналов, агрегирование или разделение запросов вода-вывода). Авторы исследования разработали модель, с помощью которой можно определить производительность различных систем хранения с ПФС Lustre при разных настройках и понять влияние всевозможных факторов. При этом не предлагается сравнивать различные ПФС с разными механизмами разделения доступа к данным и т.п. Авторы не рассматривают реализацию ввода-вывода в приложении, которая может существенно менять характер рабочей нагрузки. Но сам предложенный подход может быть полезен и в этих случаях. Авторами также отмечено, что настройки системы хранения системно зависят друг от друга и провести чистый эксперимент только с одним фактором не представляется возможным. Таким образом, в качестве теста производительности той или иной конфигурационной настройки модель не годится, к тому же она сложна в применении.

В работе [21] для выяснения характеристик производительности системы хранения применен более прагматичный подход – тестирование производительности на различных схемах ввода-вывода, которые могут быть использованы в суперкомпьютерных приложениях: запись/чтение данных параллельными процессами большими порциями, каждый в свой файл, каждый из своего файла; запись в различные системы хранения или ФС с различными конфигурациями или чтение в этих системах; создание/удаление данных. Авторы разделяют вопросы аппаратных конфигураций и настройки ФС для того, чтобы лучше понимать возможные пределы производительности системы хранения. Подобный подход к выяснению характеристик производительности систем хранения был применен и в работе [22]. Авторы тестировали ФС, глобально распределенные через WAN, на всевозможных операциях ввода-вывода.

Анализируя существующие подходы разных лет, можно выявить тенденцию смещения акцента от аппаратных решений систем хранения к программным реализациям ФС под определенный спектр рабочей нагрузки. В некоторых работах, например в [20], в модель производительности стремятся включить все факторы, влияющие на производительность, – от аппаратных до программных на самом верхнем уровне, прикладном.

В данном исследовании авторы постарались разделить оценку производительности по уров-

ням и использовать принцип, по которому пределы производительности более низкого уровня (аппаратного) на более высоком уровне можно только уменьшить. Хотя не исключается тот факт, что на более высоком уровне допустимо применять механизмы повторного использования данных (например, кэширования), масштабирования и тиражирования данных (удвоение, утроение информации), а также использовать дополнительные вычислительные ресурсы для сокращения издержек на передачу данных (например, дедубликацию), при которых пределы производительности на самом верхнем уровне могут быть увеличены. Такие методы оптимизации потоков информации в представленной работе не рассматриваются, поскольку считаются их частными случаями.

Аппаратная зависимость пределов производительности

Для оценки максимальной производительности ФС прежде всего необходимо понять, какие пределы есть у аппаратной части системы хранения. Предполагается, что эти пределы превысить невозможно.

Предел производительности системы хранения

Рассмотрим общую схему для всех систем хранения данных. В архитектуру системы хранения данных входят следующие компоненты (рис. 1):

• клиенты ФС (узлы, на которых не размещается значительная ФС);

• узлы ФС (узлы, на которых расположена значительная часть ФС, но носители данных не подключены к ним непосредственно);

• узлы ввода-вывода (серверы ФС, на которых работает серверная часть ФС и к которым носители данных подключены напрямую или через сетевую инфраструктуру; узел ввода-вывода может обслуживать только метаданные, только данные или метаданные и данные одновременно);

• носители данных (различные устройства и системы для непосредственного хранения данных; диск, дисковая полка, если она напрямую или через сеть хранения данных подключена к узлам ввода-вывода; хранилище данных с одним, двумя или более серверами резервирования, обеспечивающими эффективное объединение носителей данных более низкого порядка);

• транспортная сеть (сеть доставки данных между клиентами, узлами ФС и узлами ввода-вывода);

• сервисная сеть (сеть для обмена служебной информацией, включая блокировочные токены);

• сеть хранения данных (SAN, сеть обмена данными на уровне блоков).

Именно максимальная пропускная способность сетевой фабрики между параллельными процессами приложения и данными на носителях будет недосягаемым пределом производительности.

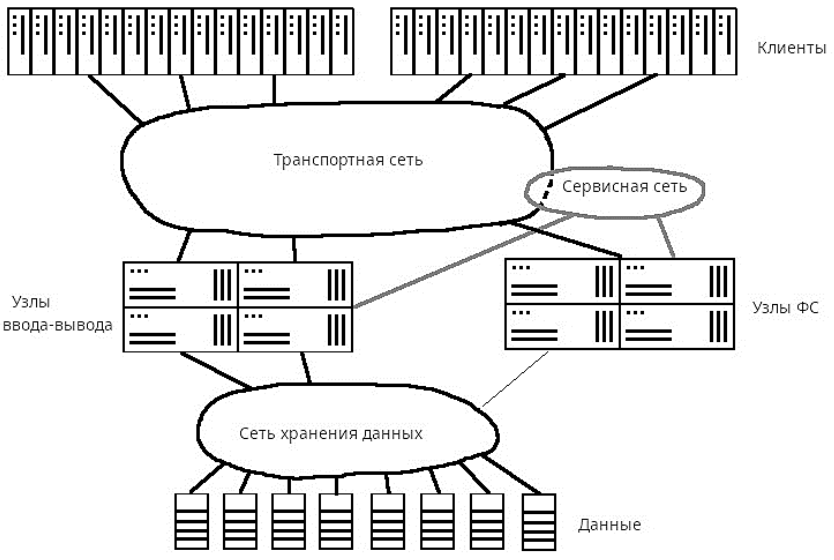

Рис. 1. Упрощенная схема системы хранения данных

Fig. 1. Simplified scheme of the data storage system

Факторы, влияющие на снижение предела производительности

На пределы производительности влияют накладные расходы в организации программно-аппаратной части системы хранения данных.

Уменьшить пределы производительности могут задержка времени отклика, а акже дробление сетевых каналов на несколько частей из-за физических нюансов аппаратуры и протоколов связи. При этом клиенты ФС вынуждены подстраиваться под такую схему, если система хранения сама динамически не подстраивается под характер приложения.

Узкие места, такие как несогласованность сетевой и агрегированной производительности носителей информации (учитывая схемы распределения данных и влияние механизмов обеспечения надежности), тоже снижают пределы. Если подобные нюансы имеются в системе, это, скорее, недостатки проектирования, поэтому в представленной модели учитываться не будут.

В ПФС существует глубокая иерархия блоков данных [23]. Это связано с наличием различных типов носителей информации (дисков, памяти узлов ФС и т.д.) и с организацией единого адресного доступа к данным. То есть данные через различные сети могут распределяться блоками между узлами, задействованными в работе ФС, и только потом непосредственно между носителями. Таким образом, различные схемы распределения данных по-разному могут влиять на предел производительности.

На производительность могут повлиять и пределы производительности сервисной сети, по которой ФС осуществляет обмен служебной информацией, такой как блокировочные токены, синхронизация метаданных и т.п. Если транспортная сеть системы хранения объединена с сервисной, то служебная составляющая трафика будет уменьшать пределы производительности всей системы хранения. Служебная составляющая содержит постоянную и дополнительную компоненту. Постоянная компонента – непрерывный трафик служебной информации, не зависящий от рабочей нагрузки ФС. Дополнительная компонента зависит от характера рабочей нагрузки всей системы хранения.

Расчет пределов производительности

Первый и основной предел производительности составляет аппаратная часть системы хранения. Для его оценки рассчитываются следующие компоненты.

1. Пропускная способность клиентов ФС:

$B_c = N_c C$,

где C – пропускная способность канала, выделенного для одного клиентского процесса; N_c – количество клиентских процессов.

В более общем плане пропускная способность клиентов

$$B_C = \Sigma_i C_i. \quad (1)$$

2. Пропускная способность N_s узлов ввода-вывода в транспортной сети, где S – пропускная способность каналов одного узла:

$$B_s = N_s S. \quad (2)$$

3. Пропускная способность сетевой инфраструктуры между клиентами и серверами ввода-вывода:

$$B_{net}. \quad (3)$$

4. Максимальная производительность хранилища:

$$B_d. \quad (4)$$

Программная зависимость пределов производительности

В работе [20] аппаратные и программные факторы учитываются в одной модели. То, что эти виды факторов не разделяются, во-первых, существенно усложняет процесс выяснения характеристик системы хранения, а во-вторых, вносит неопределенность в понимание степени влияния конкретного фактора.

В настоящем исследовании авторы полностью разделяют эти два вида факторов, что позволяет пошагово оптимизировать систему хранения данных для определенного класса нагрузки. При этом к аппаратным факторам относят такие, как количество носителей и серверов данных, сетевая инфраструктура всей системы хранения, аппаратное распределение блоков данных по носителям. К программным факторам, влияющим на производительность ФС, относят механизмы блокирования данных в ФС, методы организации единого адресного пространства в иерархии данных, механизмы консолидации и распределения блоков данных непосредственно по носителям, кроме аппаратных. Аппаратные настройки обычно устанавливаются статически при начальной инсталляции или настройке системы хранения, в дальнейшем они остаются неизменными. В современных динамически создаваемых ФС эти настройки могут быть изменены, но в очень узких границах. Вклад таких настроек в пределы производительности системы хранения требует дополнительных исследований.

К программным факторам относится также и программное распределение данных по раз-

личным по скорости доступа носителям информации (диски, память узлов ФС и т.п.).

Программные факторы

Чтобы достичь максимальной производительности при выполнении операций чтения и записи в один большой файл, система хранения распределяет блоки данных по всем дискам и дисковым контроллерам. Количество дисков, подключенных к одному узлу ввода-вывода, может быть большим, особенно при наличии сети хранения данных. Например, GPFS делит большие файлы на блоки данных одинаковой длины и последовательно размещает их на разных дисках: первый блок на первом диске, второй – на втором и так далее, блок N – на диске N, блок $N + 1$ – на первом диске и т.д. Чтобы свести к минимуму относительно длительную операцию установки жесткого магнитного диска на нужное место, используются блоки подходящей длины, согласующейся с аппаратными характеристиками дисков, RAID-контроллеров и т.п. Например, когда у дисков наименьший размер блока равен 4 Кб, RAID имеет соответствующий размер блока, равный произведению количества дисков (без избыточности) и размера блока диска, размер блока ФС можно варьировать в зависимости от ее назначения и количества серверов ввода-вывода. Если больших файлов много, то большой размер блока будет иметь преимущество и за одну операцию ввода-вывода возможно прочитать или записать больше данных без накладных расходов. Небольшие файлы и концы больших файлов могут храниться в небольших блоках фиксированной длины, называемых подблоками ФС, например, 1/32 от размера блока ФС, как в первых версиях GPFS. Этот метод экономит место на диске и не должен приводить к существенному снижению производительности. Размер блока ФС – основной программный фактор, влияющий на пределы производительности.

Механизмы опережающего чтения и кэширования записи – второй существенный программный фактор, влияющий на пределы производительности. Однако для большинства рабочих нагрузок именно для высокопроизводительного вычислительного кластера опережающее чтение или кэширование записи только уменьшает накладные расходы непосредственно при работе с носителями информации.

Существуют различные способы работы ПФС с большими каталогами. Если в одном каталоге может быть очень много файлов, то механизмы распределения метаданных по точкам доступа (серверам) существенно влияют на архитектуру системы хранения и непосредственно на производительность ФС.

В большой ФС невозможно выполнять проверку целостности каждый раз, когда ФС монтируется или происходит сбой узла кластера. Одним из способов решения проблемы является ведение специального протокола изменений метаданных каждым узлом ФС отдельно, который обязательно записывает протокол на диск, перед тем как производить сами изменения метаданных на диске. Механизмы регистрации изменений и восстановления при сбоях могут существенно влиять на производительность ФС.

Производительность кластерной ФС может значительно превышать производительность некластерной ФС, если используются параллельные операции чтения и записи на нескольких узлах ФС. С другой стороны, существуют требования к целостности данных и ограничению параллелизма в POSIX. ПФС как минимум должна гарантировать поддержку требований POSIX на одном узле для выполнения операций во всем кластере.

Указанные программные факторы и подобные им являются настраиваемыми. Оптимизацией настроек целесообразно заниматься после определения пределов производительности на более низком уровне иерархии в системе хранения данных.

Механизмы блокировки

Релевантность механизмов блокировки рабочей нагрузки существенно влияет на производительность ФС. В GPFS, например, реализованы два способа POSIX-синхронизации данных:

• *распределенная блокировка*: каждая операция ввода-вывода должна получить блокировку на чтение или запись, чтобы синхронизировать конфликтующую операцию на другом узле перед чтением или записью данных или метаданных;

• *централизованная синхронизация*: все конфликтующие операции ввода-вывода перенаправляются на центральный или указанный узел ввода-вывода, который выполняет запрошенную операцию чтения или записи.

От применяемых способов блокировки существенно зависит величина накладных расходов по параллельному доступу к данным.

Механизм распределенной блокировки обеспечивает больший параллелизм, чем централизованная синхронизация, до тех пор, пока разные

узлы работают с разными частями данных или метаданных с учетом независимости блоков данных в иерархической структуре. С другой стороны, централизованный подход может быть более эффективным при частом чтении или записи данных или метаданных с разных узлов. Когда конфликты блокировок начинают возникать часто, накладные расходы в механизме распределенной блокировки могут превысить стоимость пересылки запросов на центральный узел при централизованной синхронизации. Степень детализации блокировок сильно влияет и на производительность механизма. Так, при высокой степени детализации блокировка небольших блоков данных может привести к большому количеству запросов на блокировку, а при низкой степени детализации блокировка больших блоков данных с большей вероятностью приведет к возникновению конфликтующих запросов. На рисунке 2 изображена зависимость скорости записи данных от вида применяемого механизма их блокировки.

Влияние механизмов блокировки на пределы производительности происходит на двух уровнях. Первый – это внутренние механизмы, не зависящие от характера ввода-вывода в приложении. Здесь механизмы выбираются статично. Например, если размер операции ввода-вывода меньше размера блока ФС, то данные перенаправляются к узлу, который обслуживает весь блок данных, а если больше или равно, то ФС старается распределить нагрузку равномерно между узлами ввода-вывода. Второй уровень – прикладная задача. На этом уровне можно распознавать или статически задавать шаблоны распределения нагрузки в соответствии с необходимостью руководствоваться POSIX или с отсутствием такой необходимости.

Рис. 2. Зависимость скорости записи данных от механизма блокировки

Fig. 2. Dependence of data writing speed on the locking mechanism

Для эффективной работы с широким спектром вычислительных задач недостаточно применять только один подход к процессу блокировки. Применяемые в ПФС механизмы блокировки обычно специфичны для нее. Релевантность этих механизмов рабочей нагрузке в существенной мере влияет на производительность ФС. Выявление степени влияния применяемых механизмов лучше делать раздельно, от нижнего к верхнему уровню.

Методика оценки пределов производительности ПФС

Оценку пределов производительности системы хранения данных, а именно аппаратной и программной составляющих, целесообразно проводить поэтапно, начиная с аппаратной составляющей и заканчивая наблюдением за рабочей нагрузкой конкретного приложения.

Для оценки пределов производительности аппаратной части можно воспользоваться несколькими способами.

1. Расчет пределов производительности исходя из технических характеристик сетевых составляющих по модели (1)–(4). В этом случае минимум будет основным пределом. Если постоянный сервисный трафик системы хранения включается в транспортную сеть, то вместо (3) оценивается (3) за вычетом той производительности, которую берет на себя сервисный трафик.

2. Выполнение различных тестов по оценке максимальной производительности сетевых составляющих, причем каждой в отдельности. Вместе с первым способом это позволит выявить несоответствия документированных и реальных технических характеристик аппаратуры, которые могут вызывать, например используемые драйверы устройств и протоколы обмена.

3. Создание рабочей нагрузки, которая будет идеальной для исследуемой системы хранения, и замер параметров производительности. Этот способ является менее точным. В данном случае предел будет выявляться автоматически с учетом всех накладных расходов, в том числе и на сервисный трафик.

4. Мониторинг производительности при идеальной и неидеальной для системы хранения рабочих нагрузках. Рабочая нагрузка может создаваться специальными имитационными те-

стами или приложениями. В данном случае оценивается максимальная производительность сетевых составляющих и далее рассчитывается по модели (1)–(4).

Для лучшего понимания точности оценки пределов производительности желательно использовать все четыре способа.

После оценки пределов производительности аппаратной части можно оценивать программную зависимость и определять эффективность выбранной ФС с различными настройками и в условиях всевозможных рабочих настроек.

Пусть результат теста производительности ФС равен B. Очевидно, что он не может превышать ни одного из пределов аппаратной части (1), (2), (3), (3) минус сервисный трафик, (4). Если он превышает минимум, значит, для теста не были созданы правильные условия. Возможно, механизм кэширования данных оказал существенное влияние на результаты теста.

Принимая во внимание соотношение

$$\eta = B/\min(B_c, B_s, B_{net}, B_d), \quad (5)$$

получаем эффективность η ФС, то есть ее реальную производительность в процентах от теоретического предела. Соотношение

$$P = \min(B_s, B_{net}, B_d)/C \quad (6)$$

даст предел распараллеливания операций ввода-вывода. Здесь предполагается, что B_c больше B_s и не влияет на значения (5) и (6), поскольку количество клиентских узлов обычно не меньше количества узлов ввода-вывода.

Производительность ФС должна увеличиваться линейно с увеличением количества клиентских процессов, пока (5) не достигнет значения «1». Дальнейшее добавление параллельных клиентских процессов не приведет к увеличению общей производительности, но может вызвать ее снижение из-за роста накладных расходов.

Эксперименты и результаты

Стало очевидным, что для тестирования пределов программной составляющей системы хранения данных необходимо оценить пределы производительности аппаратной части. Это можно сделать теоретически, рассчитав B_c, B_s, B_{net} как сумму пропускных способностей сетевых каналов для соответствующего уровня и экспериментально оценив, является ли B_d пределом. Можно проверить правильность теоретической оценки, выполнив сетевые тесты и/или тесты параллельного ввода-вывода. Для этого необходимо уменьшить влияние ограничений, устранить зависимость операций ввода-вывода, то есть минимизировать количество блокировок данных во время теста. Например, массово и максимально параллельно записывать новые данные в различные файлы клиентскими процессами. ПФС Lustre, в которой есть единый сервер метаданных, распределяет получаемые непосредственно с клиентов выводимые данные по всем узлам ввода-вывода и по всем носителям данных, если операции ввода-вывода проводятся последовательно и большими блоками. Таким образом, Lustre является идеальной ПФС для тестирования аппаратной составляющей системы хранения.

Разработанная экспериментальная система хранения данных состоит из одного узла ввода-вывода метаданных, пяти узлов ввода-вывода данных и большого количества клиентских узлов (более 64). Узлы соединены сетью Intel OmniPath пропускной способностью 100 Гбит/с на интерфейс каждого узла. Носители данных подключаются к узлам ввода-вывода через те же интерфейсы Intel OmniPath. Носители данных находятся на клиентских узлах. Сетевой интерфейс является двунаправленным, поэтому влияние на производительность сетевого интерфейса при одновременном выводе с клиента и записи на носители данных минимально. При тестировании аппаратной части выполняются массовые параллельные операции последовательной записи большими блоками случайных данных множеством параллельных процессов с клиентов в новые файлы в одном каталоге. Наличие зависимости выводимых данных от единого каталога компенсируется наличием только одной точки синхронизации метаданных в Lustre – одного централизованного сервера ФС, сервера метаданных. Чтобы компенсировать возможное дробление каналов связи, на каждом клиентском узле запускается несколько параллельных процессов. Число процессов обычно делают кратным количеству процессоров на клиентском узле ФС.

Тестовая конфигурация:

- M узлов, $M \in \{1, 2, 4, 8, 16, 32, 64\}$;
- N параллельных процессов ввода-вывода на узел, $N \in \{1, 2, 4, 8, 16, 32 ,64\}$;
- количество клиентских процессов $N_s = N \times M$;
- пропускная способность канала одного клиентского узла – 100 Гбит/с;
- пропускная способность канала, выделенного для одного клиентского процесса, $C = \frac{100 \text{ Гб/с}}{N}$;
- пропускная способность клиентов ФС $B_c = N_c C = M \times 100$ Гбит/с;

– пропускная способность узла ввода-вывода в транспортной сети S = 100 Гбит/с = 12 500 Мбит/с;

– количество узлов ввода-вывода N_s = 5:

– $B_c = N_c S$ = 5 × 100 Гбит/с = 500 Гбит/с = = 62 500 Мбит/с.

Поскольку сеть является полносвязной, пропускная способность сетевой инфраструктуры транспортной сети $B_{net} = \min(B_c, B_s)$.

В представленном примере транспортная сеть объединена с сетью хранения, так как носители данных находятся на клиентских узлах. Сетевые каналы имеют пропускную способность 100 Гбит/с как на ввод, так и на вывод, тем самым общая пропускная способность сети хранения данных $B_d = B_s$.

Предполагается, что распределение данных по носителям в сети хранения данных является идеальным.

Intel OmniPath (4xEDR) использует кодировку 64b/66b, накладные расходы составляют около 3 %. Максимальная производительность пяти интерфейсов узлов ввода-вывода составляет 60 606,06 Мб/с. Тем самым теоретический предел производительности экспериментальной системы хранения равен 60,6 Гб/с.

Для уточнения аппаратных пределов производительности воспользуемся еще двумя описанными способами.

Модель рабочей нагрузки в тесте производительности будет наиболее приближена к тому, для чего создавалась ПФС Lustre – массовый параллельный вывод данных, имитация записи контрольной точки. Чтобы обеспечить независимость по данным операций вывода, записываем данные в различные независимые файлы. Файлы будут располагаться в одной и той же директории, но это не должно вызвать существенных дополнительных накладных расходов в ПФС Lustre, так как управление метаданными централизуется на одном сервере метаданных (узле ФС). Для оценки влияния процесса дробления каналов связи со стороны транспортной сети и сети хранения данных и его компенсации создадим несколько независимых параллельных процессов на одном вычислительном узле. Известно, что сервер метаданных в Lustre распределяет нагрузку по объектам, а не по блокам данных, то есть при операциях ввода-вывода с одного пользовательского процесса передача информации на носители будет происходить только через один узел ввода-вывода. Для второго процесса, даже если он запущен на том же вычислительном узле, будет назначен следующий узел ввода-вывода.

Сетевой трафик отслеживается с помощью специальных средств, что даст понимание полноты использования пропускной полосы в сетевых каналах.

Результаты тестирования с помощью имитации контрольной точки (массовый и продолжительный вывод данных в независимые файлы) приведены в таблице. Из нее видно, что один процесс на вычислительном узле не может полностью утилизировать пропускную полосу сетевых интерфейсов узлов ввода-вывода. Пропускная способность вычислительных узлов в транспортной сети используется полностью, если на одном вычислительном узле выполняются как минимум 8 процессов, на двух узлах – 16 процессов. Когда количество вычислительных узлов становится существенно больше, чем узлов ввода-вывода (в данном случае 32 к 5), для наполнения всей полосы пропускания доста-

Зависимость производительности операций вывода от количества используемых вычислительных узлов и параллельных процессов вывода на один узел, Мб/с

Dependence of output operation performance on the number of used computational nodes and parallel output processes per node, Mb/sec

		Количество параллельных процессов на узел *N*						
		1	2	4	8	16	32	64
Количество вычислительных узлов *M*	1	905,6	1 878,4	3 831,2	6 731,8	6 900,8	6 944,8	6 974,4
	2	1 864,0	3 811,2	7 081,6	12 694,4	13 759,2	14 424,0	14 992,0
	4	3 692,2	8 013,6	15 024,8	25 159,2	34 524,0	37 726,4	33 695,2
	8	7 106,9	16 326,4	28 477,6	40 768,8	43 799,2	42 072,8	43 302,4
	16	14 063,3	31 775,2	45 634,4	50 011,2	50 189,6	50 108,0	49 491,2
	32	23 555,6	47 838,4	50 400,0	49 600,0	50 160,0	50 943,2	48 209,6
	64	26 436,2	49 270,4	50 400,0	49 840,0	49 690,4	49 760,0	50 310,4

точно двух процессов на узел. Эффективная схема распределения параллельных процессов ввода-вывода по вычислительной системе требует отдельного анализа.

Наилучшая производительность системы хранения составляет B = 50942,2 Мб/с и достигается при 32 вычислительных узлах (M = 32) и 32 параллельных процессах ввода-вывода на вычислительный узел (N = 32). Примерно такая же производительность достигается уже при M = 16 и N = 8 или M = 32 и N = 4. Это подтверждает теоретические расчеты и то, что самым узким горлом в системе является общая пропускная способность каналов узлов ввода-вывода в транспортной сети (сети хранения данных). При этом пределы производительности B_c = 3 103,03 Гб/с, B_s = 60,60606 Гб/с и $B_{net} = B_s$.

На рисунке 3 изображены графики производительности сетевых интерфейсов узлов ввода-вывода. Во время теста достигалась максимальная скорость передачи данных на узел ввода-вывода – 10,4 Гб/с, что на 16,8 % меньше пропускной полосы.

Эффективность файловой системы (5) составляет $\frac{50{,}9422\ \text{Гб/с}}{60{,}60606\ \text{Гб/с}} \approx 0{,}84$.

Анализ результатов экспериментов

Теоретический предел производительности экспериментальной установки составил 60,6 Гб/с. Влияние программных факторов в проведенных экспериментах по выяснению аппаратных пределов производительности ослабляется различными приемами. Параллельные операции вывода проводятся максимально параллельно и независимо. Для ПФС Lustre с централизованным сервером метаданных не требуется проводить синхронизацию операций управления метаданными. В отказе от POSIX тоже нет необходимости, поскольку метаданные хранятся в быстрой оперативной памяти централизованного сервера. Механизмы журналирования записи, выделения свободного места для записи, работы с большими каталогами в проведенном эксперименте также в какой-либо значительной степени не влияют на производительность, так как сервер метаданных централизован, файлов немного, записываемые данные независимы. Единый каталог для записываемых файлов, если и влияет на производительность, то тоже в очень незначительной мере из-за наличия единой точки управления метаданными. Влияние эффекта кэширования на уровне ФС (на клиентах или серверах) исключается в схеме ввода-вывода, когда параллельные операции записи проводятся массово и большими блоками. Схема распределения (распараллеливания) потоков данных по носителям данных в ПФС Linux такова, что при большом количестве этих потоков дробление данных и распределение по независимым каналам и носителям данных, перевод блоков данных на нижние уровни иерархии данных проводятся без особых ухищрений и задержек. При выполнении операций записи большими порциями и параллельно при наличии единого сервера метаданных механизмы блокировок, которые могут оказать эффект на скорость записи данных, не задействуются. Таким образом, в проведенных экспериментах выявленная производительность системы хранения данных близка к пределу возможностей ее аппаратной части.

По результатам экспериментов получено снижение пределов производительности на 16,8 %.

Из графиков производительности сетевых интерфейсов узлов ввода-вывода (рис. 3) видно, что основное влияние на предел оказывают

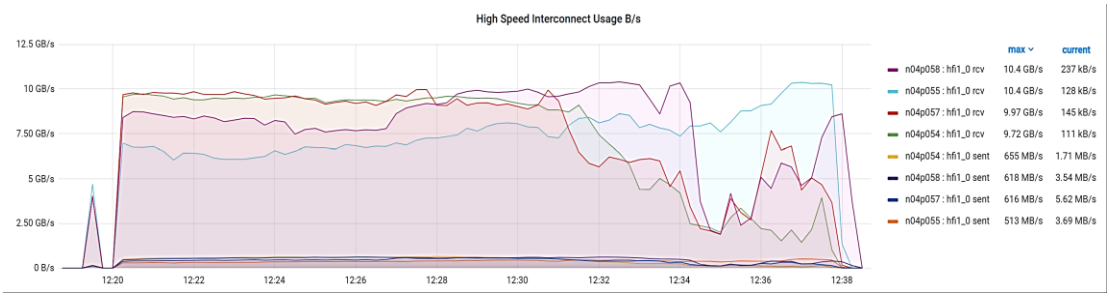

Рис. 3. Максимальная производительность интерфейсов ввода-вывода транспортной сети на узлах ввода-вывода во время работы теста

Fig. 3. Maximum performance of transport network I/O interfaces on I/O nodes during the test operation

факторы, связанные с уменьшением пропускной способности интерфейсов узлов ввода-вывода. Эффект от процесса дробления каналов связи компенсируется количеством клиентских узлов и параллельных процессов на один узел. В данном случае конфликты между параллельными потоками данных не возникают, что можно проверить сетевыми тестами. Авторы теоретически показали, что узким горлом в представленной системе хранения является максимальная пропускная способность интерфейсов узлов ввода-вывода в транспортной сети и сети хранения данных (в данном случае одна и та же сеть). Этот факт подтверждает такую несогласованность и дает повод изменить архитектуру системы хранения данных, увеличить количество узлов ввода-вывода и/или их сетевых интерфейсов.

На снижение пределов производительности для архитектуры системы хранения данных могло повлиять то, что сервер метаданных и серверы обслуживания ФС (узлы ФС) располагаются на узлах ввода-вывода. Данные системы мониторинга показали, что узлы ввода-вывода неравнозначны: два узла ввода-вывода имеют меньшую производительность, чем три других. Такой эффект создается дополнительной нагрузкой на узлы ввода-вывода – наличием сервера метаданных и/или сервера мониторинга системы хранения. Сам эффект можно оценить, если определить разницу в средней производительности узлов ввода-вывода в период тестирования.

В данном случае транспортная сеть и сеть хранения данных объединены и влияют друг на друга в случае, когда для вывода данных возникает обратный трафик. Этот эффект хорошо виден на рисунке 3, где максимальный обратный трафик составляет порядка 5 % от пропускной полосы 12,5 Гб/с. Таким образом, теоретически максимальную производительность интерфейса можно увеличить до 11 Гб/с, если разделить транспортную сеть и сеть хранения данных.

ПФС Lustre устроена так, что трафик по сервисной сети между узлами ввода-вывода между собой и узлами ФС ничтожен, то есть данный фактор несущественен.

Таким образом, выявленный предел производительности аппаратной части экспериментальной системы хранения составляет 51 Гб/с. После этого можно проводить сравнительные тесты для определения эффекта на предел производительности, оказываемого различными программными факторами.

Выводы

Способы оценки производительности ПФС для высокопроизводительного вычислительного кластера зачастую сложны и недостаточно точны. Предлагаемый способ поэтапной оценки пределов производительности по сравнению с другими способами имеет ряд неоспоримых преимуществ. Во-первых, он позволяет проще и глубже понять основные факторы, влияющие на производительность ПФС для различного рода прикладных приложений. Во-вторых, оценив основной предел производительности системы хранения еще до этапа выбора ПФС, можно понять, какой будет предел производительности вычислительного кластера на операциях ввода-вывода. Если производительности недостаточно для предполагаемой рабочей нагрузки, то необходимо корректировать проект суперкомпьютера. В-третьих, постепенное уточнение пределов производительности позволяет более точно и полно учитывать многочисленные факторы, влияющие на производительность ввода-вывода, а также нюансы архитектур и настроек систем хранения данных и ПФС. Сложная системная зависимость многочисленных факторов друг от друга может не позволить проведение моделирования или тестирования производительности всевозможных вариантов конфигурации системы хранения данных в один шаг. Оценка влияния факторов на разных архитектурных уровнях позволяет поделить все факторы на независимые группы. Влияние малых групп факторов легче оценивать, а при постепенном снижении пределов производительности появляется вероятность избежать ошибок и неточностей при анализе этой зависимости.

Принято считать, что максимальную производительность будущей системы хранения данных можно оценить по архитектурным параметрам системы хранения и аппаратной части с какой-либо ПФС, например, оценить агрегированную пропускную способность сетевых интерфейсов и количественные параметры ресурсов, задействованных в операциях ввода-вывода на кластере с учетом характера ФС. Но на самом деле эти оценки могут быть очень далеки от реальных показателей.

Производители суперкомпьютеров обычно тщательно изучают будущую рабочую нагрузку системы хранения. Имея уже готовые варианты решений, они предлагают самые эффективные. Суперкомпьютерные технологии развиваются очень быстрыми темпами. Все ча-

ще приходится использовать суперкомпьютерные установки для новых видов приложений, в которых характер рабочей нагрузки может сильно отличаться от проектной. Для этого применяют динамически создаваемые системы хранения, которые разворачиваются на оборудовании непосредственно перед запуском задачи. Раньше суперкомпьютеры оснащались одной ПФС, которая могла настраиваться или автоматически настраивалась под характер рабочей нагрузки от разных вычислительных заданий. Но все чаще в суперкомпьютерах стали применять облачные технологии, когда вычислительные ресурсы динамически выделяются под задачи непосредственно по требованию. Появляются простейшие файловые системы (ad-hoc). Для таких вариантов предложенный метод оценки пределов производительности системы хранения и последующего подбора ресурсов, вида ФС, настроек ее работы подходит намного лучше, чем известные способы оценки.

Список литературы

1. Vef M.-A., Miranda A., Nou R., Brinkmann A. From static to malleable: improving flexibility and compatibility in burst buffer file systems. In: LNCS, 2023, vol. 13999, pp. 3–15. doi: 10.1007/978-3-031-40843-4_1.
2. Oeste S., Kluge M., Tschüter R., Nagel W.E. Analyzing parallel applications for unnecessary I/O semantics that inhibit file system performance. In: LNCS, 2023, vol. 13999, pp. 161–176. doi: 10.1007/978-3-031-40843-4_13.
3. Aladyshev S., Kiselev E.A., Zakharchenko A.V., Shabanov B.M., Savin G.I. Influence of external memory characteristics of supercomputer complexes on parallel programs execution. Lobachevskii J. of Math., 2021, vol. 42, pp. 2493–2502. doi: 10.1134/S1995080221110044.
4. Schwan P. Lustre: Building a filesystem for 1,000-node clusters. Proc. Linux Symposium, 2003, pp. 380–386.
5. Hildebrand D., Honeyman P., Adamson W.A. pNFS and Linux: Working towards a heterogeneous future. CITI Tech. Report, 2007. URL: https://deepblue.lib.umich.edu/bitstream/handle/2027.42/107906/citi-tr-07-1.pdf?sequence=1&isAllowed=y (дата обращения: 19.07.2024).
6. Shvachko K., Kuang H., Radia S., Chansler R. The Hadoop distributed file system. Proc. IEEE 26th Symposium MSST, 2010, pp. 1–10. doi: 10.1109/MSST.2010.5496972.
7. Varki E., Merchant A., Xu J., Qiu X. Issues and challenges in the performance analysis of real disk arrays. IEEE Transactions on Parallel and Distributed Systems, 2004, vol. 15, no. 6, pp. 559–574. doi: 10.1109/TPDS.2004.9.
8. Thomasian A., Liu C. Comment on “Issues and challenges in the performance analysis of real disk arrays”. IEEE Transactions on Parallel and Distributed Systems, 2005, vol. 16, no. 11, pp. 1103–1104. doi: 10.1109/TPDS.2005.132.
9. Uysal M., Alvarez G.A., Merchant A. A modular, analytical throughput model for modern disk arrays. Proc. MASCOTS, 2001, pp. 183–192. doi: 10.1109/MASCOT.2001.948868.
10. Kelly T., Cohen I., Goldszmidt M., Keeton K. Inducing models of black-box storage arrays. Tech. Report HPL-2004-108, 2004. URL: https://www.researchgate.net/publication/244334840_Inducing_Models_of_BlackBox_Storage_Arrays (дата обращения: 19.07.2024).
11. Wang M., Au K., Ailamaki A. Storage device performance prediction with CART models. Proc. MASCOTS, 2004, pp. 588–595. doi: 10.1109/MASCOT.2004.1348316.
12. Anderson E. Simple table-based modeling of storage devices. SSP Tech. Report HPL–SSP–2001–4, 2001. URL: https://www.researchgate.net/publication/2364923_HPL--SSP--2001--4_Simple_table-based_modeling_of_storage_devices (дата обращения: 19.07.2024).
13. Yu W., Vetter J.S., Canon R.S. Exploiting lustre file joining for effective collective IO. Proc. CCGrid, 2007, pp. 267–274. doi: 10.1109/CCGRID.2007.51.
14. Yu W., Oral H.S., Canon R.S. Empirical analysis of a large-scale hierarchical storage system. In: LNTCS. Proc. Euro-Par, 2008, vol. 5168, pp. 130–140. doi: 10.1007/978-3-540-85451-7_15.
15. Piernas J., Nieplocha J., Felix E.J. Evaluation of active storage strategies for the lustre parallel file system. Proc. Int. Conf. SC, 2007, art. 28. doi: 10.1145/1362622.1362660.
16. Shan H.Z., Shalf J. Using IOR to analyze the I/O performance for HPC platforms. Proc. Cray User Group Conf., 2007. URL: https://crd.lbl.gov/assets/pubs_presos/CDS/ATG/cug07shan.pdf (дата обращения: 19.07.2024).
17. Yu W., Oral S., Vetter J. Efficiency evaluation of Cray XT parallel IO stack. Proc. CUG Meeting, 2007. URL: https://www.cs.fsu.edu/~yuw/pubs/2008-IPDPS-CrayXT-IO.pdf (дата обращения: 19.07.2024).
18. Yu W., Vetter J. ParColl: Partitioned collective I/O on the Cray XT. Proc. ICPP, 2008, pp. 562–569. doi: 10.1109/ICPP.2008.76.
19. Logan J., Dickens P. Towards an understanding of the performance of MPI-IO in lustre file systems. Proc. IEEE Int. Conf. on Cluster Computing, 2008, pp. 330–335. doi: 10.1109/CLUSTR.2008.4663791.
20. Zhao T., March V., Dong S., See S. Evaluation of a performance model of lustre file system. Proc. Annual ChinaGrid Conf., 2010, pp. 191–196. doi: 10.1109/chinagrid.2010.38.
21. Kluge M. Performance evaluation of the CXFS file system on the HPC/Storage complex for data-intensive computing at the TU Dresden. CMST, 2006, no. 1, pp. 29–32. doi: 10.12921/cmst.2006.SI.01.29-32.
22. Aguilera A., Kluge M., William T., Nagel W.E. HPC file systems in wide area networks: Understanding the performance of lustre over WAN. In: LNTCS. Proc. Euro-Par, 2012, vol. 7484, pp. 65–76. doi: 10.1007/978-3-642-32820-6_9.

23. Aladyshev O.S., Shabanov B.M., Zakharchenko A.V. Expectations of the high performance computing cluster file system selection. Lobachevskii J. of Math., 2023, vol. 44, pp. 5132–5147. doi: 10.1134/S1995080223120041.

Software & Systems doi: 10.15827/0236-235X.148.472-486 2024, 37(4), pp. 472–486

Computing cluster performance expectations when selecting a parallel file system

Oleg S. Aladyshev [1, 2]✉, Andrey V. Zakharchenko [1, 2], Vladimir F. Ogaryshev [1, 2], Boris M. Shabanov [1, 2]

[1] Joint Supercomputer Center of RAS, Moscow, 119334, Russian Federation
[2] National Research Centre "Kurchatov Institute", Moscow, 123182, Russian Federation

For citation
Aladyshev, O.S., Zakharchenko, A.V., Ogaryshev, V.F., Shabanov, B.M. (2024) 'Computing cluster performance expectations when selecting a parallel file system', *Software & Systems*, 37(4), pp. 472–486 (in Russ.). doi: 10.15827/0236-235X.148.472-486
Article info
Received: 20.06.2024 After revision: 27.08.2024 Accepted: 30.08.2024

Abstract. When creating a high-performance computing cluster, one of the most urgent tasks is to provide a productive external storage system for future workload. The paper studies the principles of parallel file systems' operation, which can determine their performance for different workloads; it proposes a method for determining the performance limits of external storage systems. The main advantage of the proposed method is the phased approach to identifying performance limits. Initially, the authors determine the hardware (infrastructure) limits based on theoretical calculations. Then they refine the limits using tests and/or readings of the storage hardware monitoring system. In addition to selecting an appropriate file system and configuring hardware, the authors form program factors that can affect the file system performance for the required workload. Through various subject tests or models, the limits continue to be refined. Finally, the revealed limits are checked in the storage system preliminarily tuned for the required workload. The proposed approach to selecting a parallel file system for a high-performance computing cluster and to tuning a storage system for a certain range of parallel supercomputer applications allows avoiding using complex models and the need to analyze large amounts of test results. Besides, it helps to understand better the characteristics of the storage system being created. The proposed method of systematic performance evaluation of parallel file systems allows simplifying and speeding up the process of developing a storage system with a parallel file system. The method is also applicable to modern ad-hoc file systems dynamically created for a supercomputer application.
Keywords: high-performance computing cluster, parallel file systems, external storage systems, supercomputer
Acknowledgements. The paper was carried out under the government assignment, project no. FNEF-2024-0016

References

1. Vef, M.-A., Miranda, A., Nou, R., Brinkmann, A. (2023) 'From static to malleable: improving flexibility and compatibility in burst buffer file systems', in *LNCS*, 13999, pp. 3–15. doi: 10.1007/978-3-031-40843-4_1.
2 Oeste, S., Kluge, M., Tschüter, R., Nagel, W.E. (2023) 'Analyzing parallel applications for unnecessary I/O semantics that inhibit file system performance', in *LNCS*, 13999, pp. 161–176. doi: 10.1007/978-3-031-40843-4_13.
3. Aladyshev, S., Kiselev, E.A., Zakharchenko, A.V., Shabanov, B.M., Savin, G.I. (2021) 'Influence of external memory characteristics of supercomputer complexes on parallel programs execution', *Lobachevskii J. of Math*., 42, pp. 2493–2502. doi: 10.1134/S1995080221110044.
4. Schwan, P. (2003) 'Lustre: Building a filesystem for 1,000-node clusters', *Proc. Linux Symposium*, pp. 380–386.
5. Hildebrand, D., Honeyman, P., Adamson, W.A. (2007) 'pNFS and Linux: Working towards a heterogeneous future', *CITI Tech. Report*, available at: https://deepblue.lib.umich.edu/bitstream/handle/2027.42/107906/citi-tr-07-1.pdf?sequence=1&isAllowed=y (accessed July 19, 2024).
6. Shvachko, K., Kuang, H., Radia, S., Chansler, R. (2010) 'The Hadoop distributed file system', *Proc. IEEE 26th Symposium MSST*, pp. 1–10. doi: 10.1109/MSST.2010.5496972.
7. Varki, E., Merchant, A., Xu, J., Qiu, X. (2004) 'Issues and challenges in the performance analysis of real disk arrays', *IEEE Transactions on Parallel and Distributed Systems*, 15(6), pp. 559–574. doi: 10.1109/TPDS.2004.9.
8. Thomasian, A., Liu, C. (2005) 'Comment on "Issues and challenges in the performance analysis of real disk arrays"', *IEEE Transactions on Parallel and Distributed Systems*, 16(11), pp. 1103–1104. doi: 10.1109/TPDS.2005.132.
9. Uysal, M., Alvarez, G.A., Merchant, A. (2001) 'A modular, analytical throughput model for modern disk arrays', *Proc. MASCOTS*, pp. 183–192. doi: 10.1109/MASCOT.2001.948868.
10. Kelly, T., Cohen, I., Goldszmidt, M., Keeton, K. (2004) 'Inducing models of black-box storage arrays', *Tech. Report HPL-2004-108*, available at: https://www.researchgate.net/publication/244334840_Inducing_Models_of_Black-Box_Storage_Arrays (accessed July 19, 2024).
11. Wang, M., Au, K., Ailamaki, A. (2004) 'Storage device performance prediction with CART models', *Proc. MASCOTS*, pp. 588–595. doi: 10.1109/MASCOT.2004.1348316.

12. Anderson, E. (2001) ‘Simple table-based modeling of storage devices’, *SSP Tech. Report HPL–SSP–2001–4*, available at: https://www.researchgate.net/publication/2364923_HPL--SSP--2001--4_Simple_table-based_modeling_of_storage_devices (accessed July 19, 2024).

13. Yu, W., Vetter, J.S., Canon, R.S. (2007) ‘Exploiting lustre file joining for effective collective IO’, *Proc. CCGrid*, pp. 267–274. doi: 10.1109/CCGRID.2007.51.

14. Yu, W., Oral, H.S., Canon, R.S. (2008) ‘Empirical analysis of a large-scale hierarchical storage system’, in *LNTCS. Proc. Euro-Par*, 5168, pp. 130–140. doi: 10.1007/978-3-540-85451-7_15.

15. Piernas, J., Nieplocha, J., Felix, E.J. (2007) ‘Evaluation of active storage strategies for the lustre parallel file system’, *Proc. Int. Conf. SC*, art. 28. doi: 10.1145/1362622.1362660.

16. Shan, H.Z., Shalf, J. (2007) ‘Using IOR to analyze the I/O performance for HPC platforms’, *Proc. Cray User Group Conf.*, available at: https://crd.lbl.gov/assets/pubs_presos/CDS/ATG/cug07shan.pdf (accessed July 19, 2024).

17. Yu, W., Oral, S., Vetter, J. (2007) ‘Efficiency evaluation of Cray XT parallel IO stack’, *Proc. CUG Meeting*, available at: https://www.cs.fsu.edu/~yuw/pubs/2008-IPDPS-CrayXT-IO.pdf (accessed July 19, 2024).

18. Yu, W., Vetter, J. (2008) ‘ParColl: Partitioned collective I/O on the Cray XT’, *Proc. ICPP*, pp. 562–569. doi: 10.1109/ICPP.2008.76.

19. Logan, J., Dickens, P. (2008) ‘Towards an understanding of the performance of MPI-IO in lustre file systems’, *Proc. IEEE Int. Conf. on Cluster Computing*, pp. 330–335. doi: 10.1109/CLUSTR.2008.4663791.

20. Zhao, T., March, V., Dong, S., See, S. (2010) ‘Evaluation of a performance model of lustre file system’, *Proc. Annual ChinaGrid Conf.*, pp. 191–196. doi: 10.1109/chinagrid.2010.38.

21. Kluge, M. (2006) ‘Performance evaluation of the CXFS file system on the HPC/Storage complex for data-intensive computing at the TU Dresden’, *CMST*, (1), pp. 29–32. doi: 10.12921/cmst.2006.SI.01.29-32.

22. Aguilera, A., Kluge, M., William, T., Nagel, W.E. (2012) ‘HPC file systems in wide area networks: Understanding the performance of lustre over WAN’, in *LNTCS. Proc. Euro-Par*, 7484, pp. 65–76. doi: 10.1007/978-3-642-32820-6_9.

23. Aladyshev, O.S., Shabanov, B.M., Zakharchenko, A.V. (2023) ‘Expectations of the high performance computing cluster file system selection’, *Lobachevskii J. of Math.*, 44, pp. 5132–5147. doi: 10.1134/S1995080223120041.

Авторы

Аладышев Олег Сергеевич [1, 2], к.т.н.,
ведущий научный сотрудник, o.aladyshev@jscc.ru
Захарченко Андрей Викторович [1, 2],
научный сотрудник, avz@jscc.ru
Огарышев Владимир Федорович [1, 2],
к.ф.-м.н., старший научный сотрудник,
ogaryshev@jscc.ru
Шабанов Борис Михайлович [1, 2],
д.т.н., чл.-корр. РАН, зам. директора,
shabanov@jscc.ru

[1] Межведомственный суперкомпьютерный
центр РАН, г. Москва, 119334, Россия
[2] Национальный исследовательский центр
«Курчатовский институт», г. Москва, 123182, Россия

Authors

Oleg S. Aladyshev [1, 2], Cand. of Sci. (Engineering),
Leading Researcher, o.aladyshev@jscc.ru
Andrey V. Zakharchenko [1, 2],
Research Associate, avz@jscc.ru
Vladimir F. Ogaryshev [1, 2],
Cand. of Sci. (Physics and Mathematics),
Senior Researcher, ogaryshev@jscc.ru
Boris M. Shabanov [1, 2], Dr.Sci. (Engineering),
Corresponding Member of the RAS,
Deputy Director, shabanov@jscc.ru

[1] Joint Supercomputer Center of RAS,
Moscow, 119334, Russian Federation
[2] National Research Centre “Kurchatov Institute”,
Moscow, 123182, Russian Federation