УДК 004.451

На правах рукописи

Кинсбурский Станислав Александрович

РАЗРАБОТКА И ИССЛЕДОВАНИЕ МЕТОДОВ ПОВЫШЕНИЯ СКОРОСТИ ДОСТУПА К УДАЛЁННЫМ ДАННЫМ В РАСПРЕДЕЛЁННЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Специальность 05.13.11 – Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

АВТОРЕФЕРАТ
диссертации на соискание ученой степени
кандидата технических наук

Москва – 2008

Работа выполнена в ОАО «ИНЭУМ им. И.С. Брука» и ЗАО "МЦСТ".

Научный руководитель: доктор технических наук,
старший научный сотрудник
Егоров Геннадий Алексеевич

Официальные оппоненты: доктор технических наук,
профессор
Семенихин Сергей Владимирович

кандидат технических наук,
старший научный сотрудник
Дубовик Евгений Александрович

Ведущая организация: Институт точной механики и вычислительной техники им. С. А. Лебедева РАН

Защита состоится **« »** _____ **2008 г. в ___ ч. ___ мин.** на заседании диссертационного совета Д.409.009.01 при ОАО «Институт электронных управляющих машин имени И. С. Брука» по адресу: 119334, г. Москва, ул. Вавилова, 24.

С диссертацией можно ознакомиться в учёном совете ОАО «Институт электронных управляющих машин имени И. С. Брука».

Автореферат разослан **« »** _____ **2008 г.**

Ученый секретарь
диссертационного совета
к.т.н., профессор Красовский В.Е.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность работы. Современные технологии позволяют создавать распределённые вычислительные системы в виде отдельных модулей - одноплатных многомашинных вычислительных комплексов (МВК). При этом организация параллельных вычислений для определенного класса задач может оказаться технически более простой и со сравнительно небольшими потерями на реализацию синхронизирующих действий в виде сообщений. Все это является причиной растущей популярности МВК для распределенной обработки данных.

Под распределенной обработкой данных будем понимать организацию и обработку данных в многомашинных вычислительных комплексах (МВК). МВК представляют собой объединённые в сеть для совместной работы удалённые компьютеры (узлы МВК), которые не имеют доступа к общей памяти, и обладают доступом только к своим собственным ресурсам.

МВК имеют самое широкое распространение. Они нашли применение практически во всех отраслях народного хозяйства, включая оборонно-промышленный комплекс.

Современное развитие микроэлектронных технологий позволяет создавать производительные и при этом компактные многомашинные комплексы, предназначенные для встраиваемых применений и ориентированные на работу в реальном масштабе времени. Компактность многомашинных комплексов достигается путем объединения нескольких узлов на одной материнской плате и за счёт предоставления доступа к периферийным устройствам (в том числе к носителям информации) только одному из узлов многомашинного комплекса.

Примером современного отечественного компактного МВК является вычислительный комплекс, построенный на основе процессорных модулей «МВС/С» разработки ЗАО «МЦСТ». «МВС/С» состоит из четырёх узлов, размещённых на одной плате и соединённых между собой быстрыми каналами связи. При этом доступом к внешним устройствам обладает только один из узлов МВК. Для загрузки всего МВК необходимо обеспечить возможность загрузки и рестарта (перезапуска) бездисковых узлов, которые не имеют прямого доступа к внешним устройствам. Скорость загрузки и рестарта такого МВК является особенно важными параметром с точки зрения обеспечения его функционирования в реальном масштабе времени.

Под временем загрузки и рестарта понимается время, необходимое для приведения МВК в состояние готовности к работе после сигнала сброса. Готовность МВК к работе – это состояние, в котором запуск целевой задачи возможен на любом из его узлов. Сигнал сброса – это аппаратный сигнал, инициирующий старт либо перезапуск вычислительного комплекса.

Загрузка бездисковых узлов многомашинного комплекса предполагает использование программных методов передачи данных от узла, имеющего аппаратный доступ к носителям информации (сервера), к бездисковым узлам (клиентам). Для передачи данных используются каналы связи,

объединяющие вычислительные узлы в МВК. Передача данных по каналам связи приводит к дополнительным задержкам в получении данных узлом–клиентом по сравнению с узлом–сервером. Однако эта задержка зависит не только от скорости передачи данных по каналам связи, но и от эффективности программного обеспечения, обеспечивающего доступ к удалённым данным.

Организация доступа к удалённым данным в распределённых вычислительных системах, как правило, обеспечивается с помощью распределённых файловых систем. Как показал проведённый анализ, современные распределённые файловые системы не обеспечивают скорость доступа к данным и скорость загрузки и рестарта, требуемые в вычислительных системах, работающих в реальном масштабе времени.

В этой связи актуальной является задача, связанная с реализацией методов повышения скорости доступа к удалённым данным в МВК с учётом требований работы в реальном масштабе времени.

Цель диссертационной работы. Целью диссертационной работы является разработка и исследование методов доступа к удалённым данным в распределённых вычислительных системах, обеспечивающих повышение производительности, скорости доступа (реактивности) к удалённым данным при обеспечении низкой нагрузки на вычислительные узлы. Для достижения поставленной цели были определены и решены следующие задачи:

- обзор и анализ существующих методов доступа к удалённым данным в современных распределённых вычислительных системах. Основными параметрами оценки являются время, необходимое для приведения МВК к готовности к работе после рестарта системы, скорость доступа узлов–клиентов к удалённым данным и нагрузка, оказываемая на узел–сервер;
- разработка методов и алгоритмов доступа к удалённым данным в распределённых вычислительных системах, позволяющих увеличить скорость доступа, уменьшить время загрузки и рестарта, а также снизить нагрузку на узел–сервер по сравнению с известными распределёнными файловыми системами;
- разработка архитектуры быстрой распределённой файловой системы, отличительными особенностями которой являются реализация программ обслуживания распределённой файловой системы в адресном пространстве ядра операционной системы (ОС) и обмен данными на уровне блоков, а не файлов, как реализовано в большинстве распределённых систем;
- практическая реализация предложенных методов и алгоритмов в виде быстрой распределённой файловой системы;
- оценка эффективности предложенных методов на основе имитационного моделирования.

Методы исследования. Для решения поставленных задач в диссертации использовались методы теории алгоритмов, методы и технологии системного программирования, методы математического и имитационного моделирования.

Научная новизна исследования. К составляющим научную новизну диссертационной работы решениям следует отнести:

- анализ распределённых файловых систем на основе модели взаимодействия открытых систем (OSI), обеспечивающей наглядность архитектуры распределённой файловой системы и методов её реализации и позволяющей оценить её эффективность;
- разработка метода доступа к удалённым данным на уровне блоков файловой системы, обеспечивающего по сравнению с доступом на уровне файлов меньшее время доступа к данным, уменьшение нагрузки на узел–сервер и уменьшение времени загрузки и рестарта;
- разработка методов доступа к удалённым данным, размещаемых в адресном пространстве ядра;
- разработка алгоритмов единого прозрачного доступа к удалённым данным, расположенных на разных носителях.

Практическая ценность и реализация результатов работы. Предложенные методы и алгоритмы реализованы в виде быстрой распределённой файловой системы в составе операционной системы Linux. В частности, на основе исследований, выполненных по теме диссертации, была реализована модель «клиент–сервер» в ядре ОС Linux для процессорного модуля «МВС/С».

Результаты имитационного моделирования показали эффективность разработанных методов и алгоритмов удалённого доступа к распределённым данным по сравнению с наиболее известными современными распределёнными файловыми системами.

Апробация работы. Основные положения и результаты работы докладывались на Международных и других научных конференциях: XXXII Гагаринские чтения (Москва, 2006 г.); XXXIII Гагаринские чтения (Москва, 2007 г.); XXXIV Гагаринские чтения (Москва, 2008 г.); XXIII научно–техническая конференция на тему "Направление развития и применения перспективных вычислительных систем и новых информационных технологий в ВВТ РКО" (Москва, 2007 г.), а также на семинарах НТС ИМВС РАН, ЗАО “МЦСТ” и ОАО “ИНЭУМ”.

Публикации. По теме диссертационной работы опубликованы 7 печатных работ, в том числе в издании, рекомендованном ВАК РФ.

Структура, объём работы. Диссертация состоит из введения, четырёх глав с выводами, заключения. Основная часть работы изложена на 109

страницах, содержит 21 рисунок, 8 таблиц и библиографический список, включающий 37 наименований.

ОСНОВНОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении определяется область исследования в ряду актуальных проблем компьютерных технологий, обосновывается актуальность диссертационной работы, формулируются цель и научные задачи исследования.

В первой главе приведён общий обзор структуры распределённых вычислительных систем, определены требования к методам доступа к удалённым данным в современных распределённых вычислительных системах, к которым, прежде всего, следует отнести:

- высокую скорость обмена данными;
- контроль целостности данных;
- высокую скорость доступа к данным;
- предсказуемость времени доступа к данным;
- высокую скорость инициализации методов доступа после старта/рестарта;
- снижение нагрузки на узел–сервер.

Необходимо отметить, что последние четыре требования являются особенно актуальными для распределённых вычислительных систем, работающих в реальном масштабе времени.

Анализ распределённых вычислительных систем показывает, что они, как правило, реализуются под управлением Unix–подобных операционных системах. Для обеспечения безопасности, надёжности и стабильности вычислительной системы в Unix–подобных операционных системах программы ядра и пользователя исполняются в разных режимах процессора и размещаются в разных областях адресного пространства. Прямой доступ к данным реализуется обычно на уровне ядра ОС, а обращение к данным осуществляется при помощи служб ядра и системных вызовов.

В современных наиболее распространённых методах удалённого доступа передача данных осуществляется обычно на уровне файлов, поэтому они объединены под общим названием «распределённые файловые системы». Анализ наиболее распространённых распределённых файловых систем позволяет выделить следующие характеризующие их параметры:

- прозрачность – представление удалённых данных частью локальных данных;
- масштабируемость – обеспечение присоединения к системе новых узлов;
- безопасность – обеспечение безопасности связи и гарантирование корректных разрешений на доступ к данным;

- отказоустойчивость – обеспечение сохранности удалённых данных в стабильном и непротиворечивом состоянии в случае, если на одном из узлов МВК происходит сбой;
- непротиворечивость – обеспечение отсутствие противоречий в локальных кэшированных копиях файла на узлах–клиентах в случае разделения его на запись.

Для доступа к удалённым данным используются методы, объединённые под общим названием «удалённый вызов процедур». То есть в процессе обмена удалёнными данными узлу–серверу пересылается также служебная информация, позволяющая реализовать вызов необходимой пользователю узла–клиента процедуры (системного вызова). Таким образом, программы, обслуживающие удалённый вызов процедур, реализованы преимущественно в адресном пространстве пользователя, что приводит к двойному увеличению количества операций обмена, необходимых для получения данных, по сравнению с локальным доступом. Для ускорения обмена данные кэшируются на узле–сервере, что позволяет увеличить скорость обмена, но приводит к дополнительной нагрузке на узел–сервер.

Для получения оценки эффективности современных методов доступа к удалённым данным был проведён анализ распределённых файловых систем NFS, AFS, Coda, xFS, GPFS, Lustre и Sprite с помощью модели взаимодействия открытых систем OSI. Каждый уровень модели рассматривается с точки зрения влияния его реализации в современных распределённых файловых системах на время доступа (реактивность) к удалённым данным и на нагрузку на узел–сервер. Результаты анализа показывают общий для современных распределённых файловых систем недостаток, связанный с плохой реактивностью (большим временем доступа к удалённым данным), большим временем инициализации в случае старта и рестарта и высокой нагрузкой на узел–сервер по сравнению с методами доступа к локальным данным. Это является недопустимым в распределённых вычислительных системах, работающих в реальном масштабе времени.

По результатам проведённого анализа сформулированы задачи диссертационного исследований. В связи с тем, что существующие распределённые файловые системы не обеспечивают должной реактивности, скорости загрузки и рестарта и снижения нагрузки на узел–сервер, необходимых в распределённых вычислительных системах, работающих в реальном масштабе времени, необходима разработка принципиально новой архитектуры распределённой файловой системы.

Во второй главе, исходя из анализа требований доступа к удалённым данным в реальном масштабе времени, автором предложена архитектура быстрой распределённой файловой системы (БРФС), представленная на рис.1.

Основное отличие предлагаемой архитектуры по сравнению с существующими распределёнными файловыми системами состоит в том, что

доступ к удалённым данным предлагается реализовать на уровне блоков. Доступ к блокам при этом доступен только в режиме ядра ОС, поэтому ещё одним принципиальным отличием является то, что программы обслуживания распределённой файловой системы полностью выполняются в режиме ядра ОС.

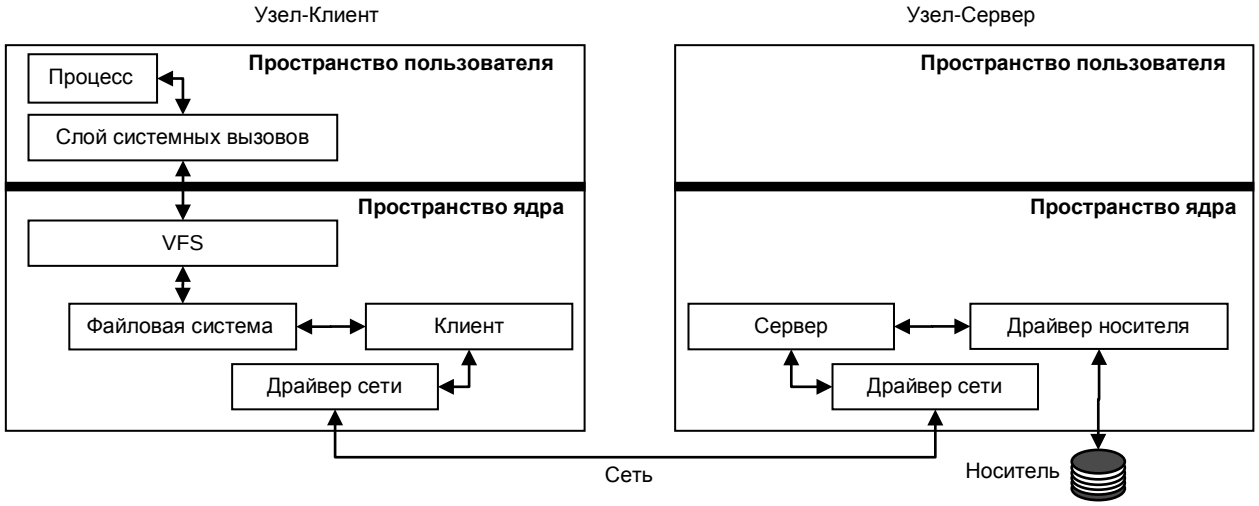

Рис. 1. Архитектура быстрой распределённой файловой системы.

На узле–клиенте запрос процесса пользователя к удалённым данным при помощи системного вызова передаётся ядру, а именно VFS (Virtual File System – Виртуальная Файловая Система). VFS исполняется в режиме ядра и обеспечивает прозрачность представления разных файловых систем. Она определяет тип файловой системы, которой адресован запрос, и передаёт его соответствующему драйверу файловой системы. Файловая система преобразует запрос пользователя к данным в запрос к соответствующему носителю информации и передаёт его клиенту. Клиент создаёт удалённый запрос и передаёт его с помощью драйвера сети узлу–серверу. На узле–сервере сервер принимает запрос клиента от драйвера сети, анализирует его, затем формирует запрос к носителю информации и передаёт его драйверу носителя. Драйвер носителя осуществляет обмен с носителем и возвращает данные серверу. Получив данные, сервер создаёт ответ на удалённый запрос, и передаёт его с помощью драйвера сети узлу–клиенту. На узле–клиенте клиент получает ответ сервера от драйвера сети, анализирует его, затем возвращает данные драйверу файловой системы, который возвращает их VFS, и далее – пользователю.

Клиент БРФС предоставляет процессу пользователя возможность обращения к удалённым данным. Поскольку клиент выполняется в режиме ядра ОС и не имеет независимой от пользователя активности, то его целесообразно реализовать в виде блочного драйвера, подобного драйверу носителя информации. Такая реализация приводит к тому, что запросы от пользовательского процесса к клиенту поступают в виде запросов к секторам носителя информации. Такие запросы можно передавать в пределах ядра

любому блочному драйверу с помощью штатных функций. Таким образом, задачей клиента является пересылка запроса к блочному драйверу на сервер.

Сервер БРФС должен иметь возможность обрабатывать запросы клиентов независимо от активности процессов пользователя на узле-сервере. Функционирование сервера в режиме ядра накладывает определённые ограничения на способы его реализации. Фактически методы современных ядер Unix-подобных операционных систем позволяют реализовать независимый сервер внутри ядра только с помощью потоков ядра. Для обеспечения высокой скорости реакции на запрос клиента, а также для обеспечения незначительного уменьшения скорости обработки запросов в случае высокой нагрузки клиентов на сервер, предлагается использовать как минимум три разных типа потоков. Первый тип потоков принимает запросы клиентов. Второй тип перенаправляет запрос клиента к носителю информации. Так как клиентов может быть несколько, то необходим также третий тип, контролирующий передачу запросов клиентов между этими двумя потоками.

Необходимо отметить, что предлагаемая архитектура БРФС для её реализации требует некоторой аппаратной поддержки программного обеспечения, что накладывает определённые ограничения на её применение:

- центральные процессоры узлов распределённой вычислительной системе должны иметь одинаковый формат порядка следования байтов;
- аппаратура, связанная с обменом данными (а именно контроллеры носителей информации и каналов связи), должна иметь возможность прямого обмена с памятью (без использования центрального процессора), обеспечивать контроль целостности данных при передаче и возможность сборки/разборки пакета данных, а также формировать признак конца обмена.

В этом случае архитектура БРФС гарантирует, что при передаче данных не происходит лишних копирований данных. При дальнейшем рассмотрении БРФС предполагается, что в распределённой вычислительной системе такая аппаратная поддержка присутствует.

Предлагаемая архитектура БРФС обладает следующими важными отличиями от существующих распределённых файловых систем:

- унифицированный доступ к удалённым данным. Доступ к данным и метаданным файлов является унифицированным, так как блоки файловой системы являются более низким уровнем хранения информации, чем файлы. Единый доступ к данным обеспечивается простым протоколом удалённого доступа, который является модификацией протокола запросов к блочным устройствам ядра ОС;
- обработка файловой системы на узле–клиенте. Преобразование процедурных запросов пользователя к удалённым данным в запросы к секторам носителя информации осуществляет операционная система

узла–клиента. Между клиентом и сервером пересылаются только запрашиваемые данные и необходимая служебная информация;

- использование стандартных механизмов кэширования данных. В БРФС отсутствуют специальные механизмы кэширования. Поэтому кэширование данных происходит только на узле–клиенте с помощью стандартных механизмов операционной системы. Сервер не накапливает информацию о запросах клиентов и анализирует их исключительно с точки зрения наличия прав доступа клиента к данным;
- отсутствие разделения данных на запись. Поскольку сервер не кэширует запросы клиентов, то возможность разделения данных на запись в такой системе отсутствует;
- быстрая загрузка и рестарт. Реализация программ обслуживания БРФС в пространстве ядра позволяет добиться быстрого по сравнению с современными распространёнными распределёнными файловыми системами старта программ обслуживания быстрой распределённой файловой системы;
- независимое адресное пространство. Адресное пространство узла–клиента независимо от границы и объёма носителя на узле–сервере. Поэтому на узле–сервере возможна реализация единого прозрачного для узла–клиента адресного пространства на нескольких носителях, либо организация нескольких независимых адресных пространств на одном носителе.

Для оценки эффективности предлагаемой архитектуры БРФС с точки зрения реактивности, нагрузки на узел–сервер и скорости инициализации методов доступа после старта и рестарта был проведён анализ на основе модели OSI. Модель OSI быстрой распределённой файловой системы представлена на рис. 2.

Оперирование блоками файловой системы при обмене позволяет существенно уменьшить количество необходимых служб и сервисов распределённой файловой системы по сравнению с известными аналогами, и тем самым отказаться от реализации наиболее ресурсоёмкого уровня модели OSI – уровня приложения. В современных распределённых файловых системах он реализован на уровне пользователя и представляет собой службы обслуживания и поддержки удалённого вызова процедур и целостности данных.

Реализация однородной распределённой вычислительной системы позволяет отказаться от ещё одного ресурсоёмкого уровня OSI – уровня представления. Поскольку вычислительные узлы МВК имеют одинаковый формат порядка следования байтов, то необходимость в контроле единого представления данных отсутствует, что снижает нагрузку на узел–сервер и увеличивает скорость доступа и обмена данными.

На уровне сессии реализован протокол удалённых запросов к носителю информации. По сути, он является упрощённым аналогом протокола

удалённого вызова процедур (Remote Procedure Call – RPC) и обозначается в работе аббревиатурой SRPC (Simple RPC – Простой RPC). Протокол SRPC обеспечивает клиенту возможность подключаться и отключаться от сервера и обмениваться с ним блоками файловой системы. Так как протокол SRPC работает на уровне драйвера носителя информации, то он должен иметь запросы на чтение и запись блоков файловой системы. К тому же, для обеспечения поддержки нескольких клиентов и гибкого выделения ресурсов необходимы также запросы, обеспечивающие подключение и отключение клиента от сервера. Четырёх типов удалённых запросов вполне достаточно для обеспечения всех необходимых пользователю на клиенте операций с удалёнными данными. Важно отметить, что подключение и отключение производится соответственно в начале и в конце сессии. Для обмена данными используются только запросы чтения и записи. Простота протокола и отсутствие дополнительных запросов при обмене данными существенно снижает нагрузку на каналы данных и узел–сервер и увеличивает скорость доступа и обмена данными.

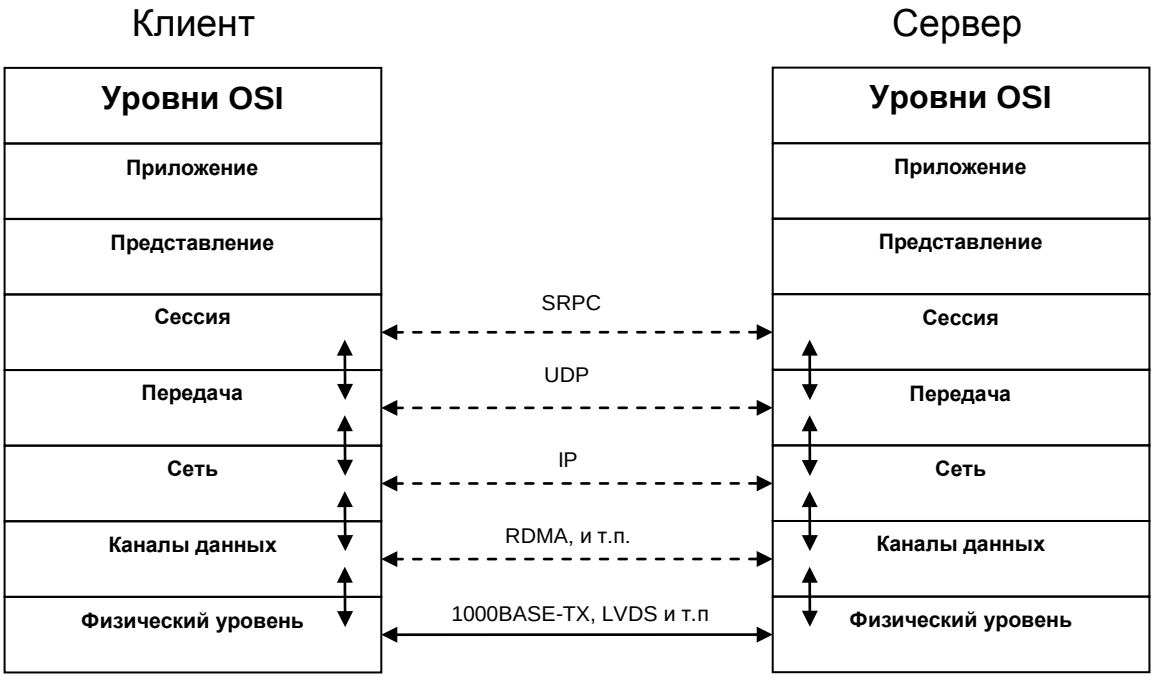

Рис. 2 Модель OSI быстрой распределённой файловой системы.

Поскольку аппаратура поддерживает целостность передаваемых данных, то уровень передачи предлагается реализовать в виде модификации протокола UDP (а не TCP), что позволяет снизить нагрузку на узел–сервер и увеличить скорость доступа и обмена данными.

На уровне сети реализована модификация протокола IP.

На уровне каналов данных предполагается использование высокоскоростных каналов передачи данных, позволяющих осуществлять сборку и разборку пакетов данных. Сборка/разборка пакетов позволяет

снизить нагрузку на процессор и увеличить скорость доступа и обмена данными.

На физическом уровне предлагается использовать протоколы высокоскоростных связей, например LVDS или 1000BASE–TX.

Независимость адресного пространства на узле–клиенте от границ физических устройств на узле–сервере позволяет реализовать в БРФС доступ к удалённым данным, который невозможен в других распределённых файловых системах.

Одним из способов реализации адресного пространства является реализация логических дисков для нескольких узлов–клиентов на одном носителе. Такая реализация позволяет уменьшить размеры и энергопотребление многомашинного комплекса. Однако она не позволяет достичь высокой скорости доступа к данным в распределённых системах с интенсивным обменом данными между узлами и носителем информации.

Более целесообразной является реализация единого адресного пространства на нескольких носителях, в том числе разных типов. Она позволяет предоставить узлу больший объём и распараллелить доступ к разным областям адресного пространства благодаря одновременным запросам к разным носителям.

Время доступа и скорость обмена данными во многом зависит от типа носителя информации. Анализ разных типов носителей показывает, что они имеют свои преимущества и недостатки. Так, твердотельные диски по сравнению с жесткими дисками имеют следующие преимущества:

- малое время начальной инициализации;
- высокая скорость доступа к данным;
- высокая скорость чтения и записи данных;
- низкое энергопотребление;
- высокая надёжность,

и следующие недостатки:

- малый объём;
- высокая стоимость;
- ограниченное число циклов перезаписи.

Реализация единого адресного пространства на жестком и твёрдотельном дисках (разнородного носителя) позволяет предоставить узлу–клиенту преимущества обоих типов носителей. Примером адресации разнородного носителя может служить схема, представленная на рис. 3.

Единое для узла–клиента адресное пространство реализовано на узле–сервере на трёх разных типах запоминающих устройств. В диапазоне от 0 до X адресного пространства узла–клиента на узле–сервере адресуется твёрдотельный диск. В диапазоне от X+1 до Y – регион оперативной памяти. В диапазоне от Y+1 до Z – жесткий диск.

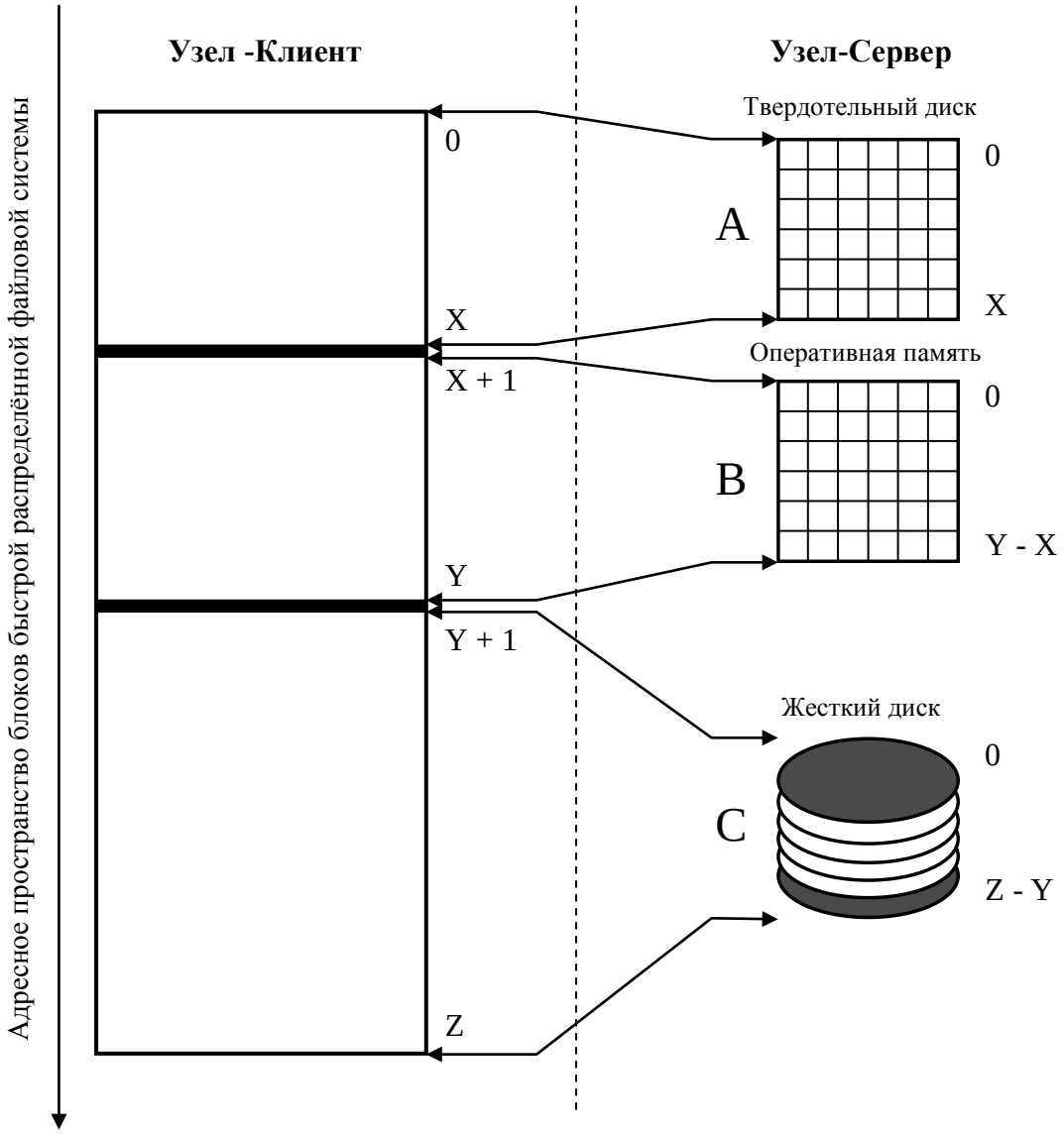

Рис. 3. Схема адресации разнородного носителя на узле–сервере.

Размещение в диапазоне от 0 до X ядра и служебных программ ОС узла–клиента позволяет реализовать быструю загрузку и рестарт бездискового узла–клиента многомашинного комплекса. При этом недостатки твердотельного диска являются несущественными, так как для размещения загрузочных данных требуется сравнительно мало адресного пространства и, как правило, они используются только в режиме чтения.

Размещение в диапазоне от X+1 до Y файла подкачки ОС узла–клиента позволяет достичь высокой скорости свопинга временных данных.

Диапазон от Y+1 до Z предназначен для размещения программ и данных пользователя узла–клиента.

Важно отметить, что архитектура БРФС позволяет узлу–клиенту получить доступ к разнородному носителю, даже если один из составляющих его носителей ещё не готов к работе. Это означает, что после инициализации на узле–сервере твердотельного диска узел–клиент имеет возможность

обмениваться с ним данными даже в том случае, если жесткий диск ещё не инициализирован.

В третьей главе рассматриваются вопросы практической реализации БРФС. Реализация предлагаемой архитектуры БРФС была выполнена в составе операционной системы Linux версии Suse 7.3 (версия ядра 2.4.25) для вычислительного комплекса «МВС/С» разработки ЗАО «МЦСТ». В качестве каналов связи использовались каналы RDMA разработки ЗАО «МЦСТ».

Реализация протокола обмена данными между программами БРФС и драйвером каналов связи позволяет избавиться от избыточного копирования данных и реализовать их передачу внутри ядра с помощью указателей.

Протокол удалённых запросов SRPC включает в себя четыре запроса: Connect, Disconnect, Read и Write. Каждый запрос и ответ на него содержат как минимум заголовок запроса.

В силу аппаратной поддержки заголовок является достаточно простым и включает следующие параметры:

- IP–адрес узла–клиента – необходим для передачи ответа от сервера клиенту;
- тип запроса – уникальный идентификатор запроса;
- номер логического канала – уникальный идентификатор, выдаваемый сервером клиенту при подключении;
- номер стартового сектора в обмене – стартовый номер сектора на носителе информации в запросе;
- количество секторов в обмене;
- результат запроса – результат, возвращаемый сервером клиенту.

Предполагается, что IP–адрес узла–сервера в сети заранее определён, поэтому он в заголовок не входит. Размер заголовка выровнен на максимальный размер блока файловой системы. После заголовка размещаются данные. Объём передаваемых данных ограничен максимальным размером запроса к носителю информации. Такая реализация протокола позволяет не передавать предварительно размер пакета, так как его максимальный размер ограничен константой. Отсутствие лишних обменов пакетами данных позволяет снизить нагрузку на каналы данных, увеличить скорость доступа и обмена данными.

Запрос Connect обеспечивает логическое подключение клиента к серверу. Результатом запроса является выделение сервером ресурсов клиенту и установление логического соединения между ними. Сервер выделяет клиенту уникальный номер, однозначно описывающий выделенные клиенту ресурсы, и передаёт ему данные, характеризующие удалённые носители. Клиент использует эти данные для регистрации удалённого носителя в ОС узла–клиента.

Запросы Read и Write используются соответственно для получения и передачи удалённых данных клиентом. Клиент, получив от пользователя запрос на чтение данных, отправляет запрос серверу, включающий в себя

номер логического канала, тип запроса, абсолютный номер начального сектора запроса в пределах устройства и число секторов в запросе. Сервер осуществляет запрос к носителю и отправляет клиенту запрашиваемые данные. Запрос пользователя на запись обрабатывается зеркально запросу на чтение.

Запрос Disconnect предназначен для логического отключения клиента от сервера и позволяет реализовать «горячую» замену носителя для узла–клиента на узле–сервере. Главным параметром в запросе Disconnect является номер логического канала. Сервер, получив запрос Disconnect, освобождает ресурсы, выделенные клиенту. Клиент, получив ответ от сервера, удаляет информацию об удалённом носителе из системы.

Описанная реализация протокола SRPC является достаточной для поддержки всех операций с файлами и позволяет полностью соблюсти семантику доступа к носителю информации в UNIX–подобной ОС.

Клиент БРФС представляет собой блочный драйвер. При инициализации он посылает серверу запрос Connect, получает от него необходимые данные и с помощью служебных функций ядра регистрирует в системе новое блочное устройство. Запросы пользователя преобразуются клиентом в удалённые запросы и пересылаются серверу с помощью запросов READ и WRITE. Обращение к удалённым данным является абсолютно прозрачным и ничем не отличается от обращения к локальному диску.

Сервер БРФС реализован как многопоточный демон ядра, состоящий из трёх типов потоков. Схема его работы представлена на рис. 4.

Поток-«приёмник» обеспечивает приём запросов от клиентов. Количество «приёмников» соответствует количеству независимых каналов связи. Получив запрос от клиента, он передаёт его потоку-«контролёру», после чего ожидает следующий запрос.

Поток-«контролёр» обеспечивает обработку заголовков и анализ запросов клиентов, а также обрабатывает запросы клиента на подключение и отключение. Если запрос является корректным и клиент запрашивает обмен данными, то такой запрос передаётся третьему типу потоков – «исполнителю». После чего, если очередь пуста, поток-«контролёр» «засыпает» и ждёт следующего запроса.

Поток-«исполнитель» обрабатывает запросы клиента на обмен данными. Он формирует запрос к драйверу носителя информации, дожидается от него ответа, после чего отправляет данные клиенту. Количество «исполнителей» соответствует количеству логических каналов, то есть количеству подключённых клиентов. Если клиент использует распределённый носитель, то для каждого независимого носителя, входящего в его состав, создаётся свой поток.

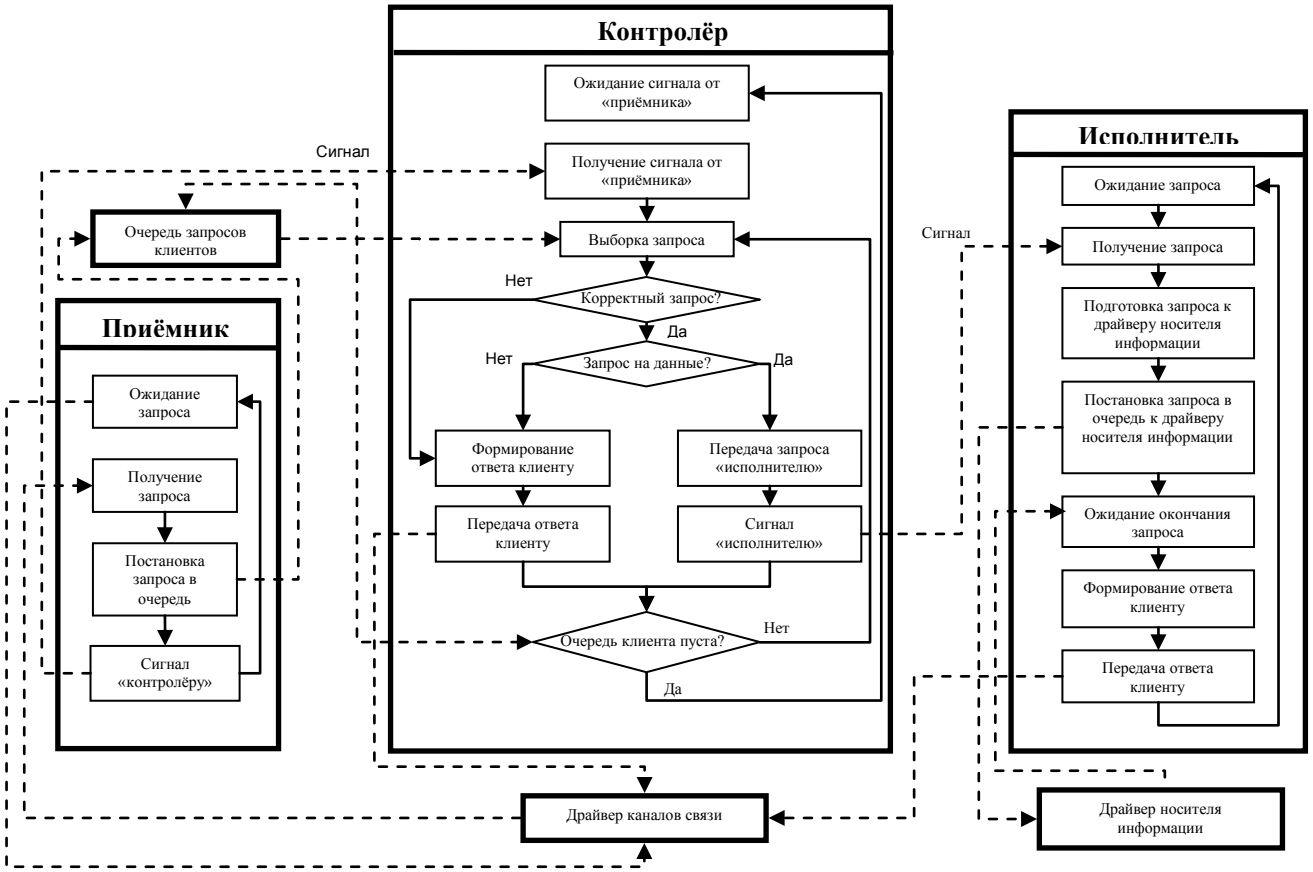

Рис. 4. Блок-схема работы сервера быстрой распределённой файловой системы.

В четвёртой главе приведена оценка эффективности БРФС с точки зрения скорости доступа к данным и нагрузки, оказываемой на узел–сервер. Было проведено сравнение времени чтения файлов разных размеров в локальной файловой системе, в БРФС и в NFS. В качестве стенда использовался макет вычислительного комплекса «МВС/С», состоящий из двух узлов, соединённых между собой каналами RDMA и управляемых операционной системой Suse Linux 7.3 (версия ядра 2.4.25).

Необходимо отметить, что в NFS в качестве каналов связи также использовались каналы RDMA. Таким образом, проведённое тестирование является сравнением эффективности архитектур распределённых файловых систем, и его результаты являются достаточно представительными для оценки эффективности БРФС.

Так как тестирование проводилось на макете вычислительного комплекса, то абсолютные значения не дают объективной оценки эффективности БРФС. Поэтому все результаты моделирования получены в относительных величинах по отношению ко времени чтения в локальной файловой системе.

Чтение файлов было реализовано с помощью программы dd. Время чтения было замерено с помощью программы time. Файл считывался в системное устройство /dev/null.

Замеры времени проводились в четырех тестах:

1. Первичное чтение файла на узле–клиенте. Результаты этого теста демонстрируют реальное время подкачки данных с узла–сервера. Полученные результаты представлены на рис. 5.
2. Повторное чтение файла на узле–клиенте. Этот тест демонстрируют степень и эффективность кэширования данных узлом–клиентом.
3. Первичное чтение файла при «нагруженном» узле–сервере. Результаты этого теста демонстрируют время подкачки данных узлом–клиентом в случае, если узел–сервер нагружен выполнением задачи, полностью загружающей вычислительные ресурсы.
4. Время выполнения счётной задачи, полностью загружающей вычислительные ресурсы, на узле–сервере при поддержке интенсивного обмена данными с узлами–клиентами. Результаты этого теста демонстрируют нагрузку, оказываемую узлом–клиентом на узел–сервер при подкачке данных. Полученные результаты приведены на рис. 6.

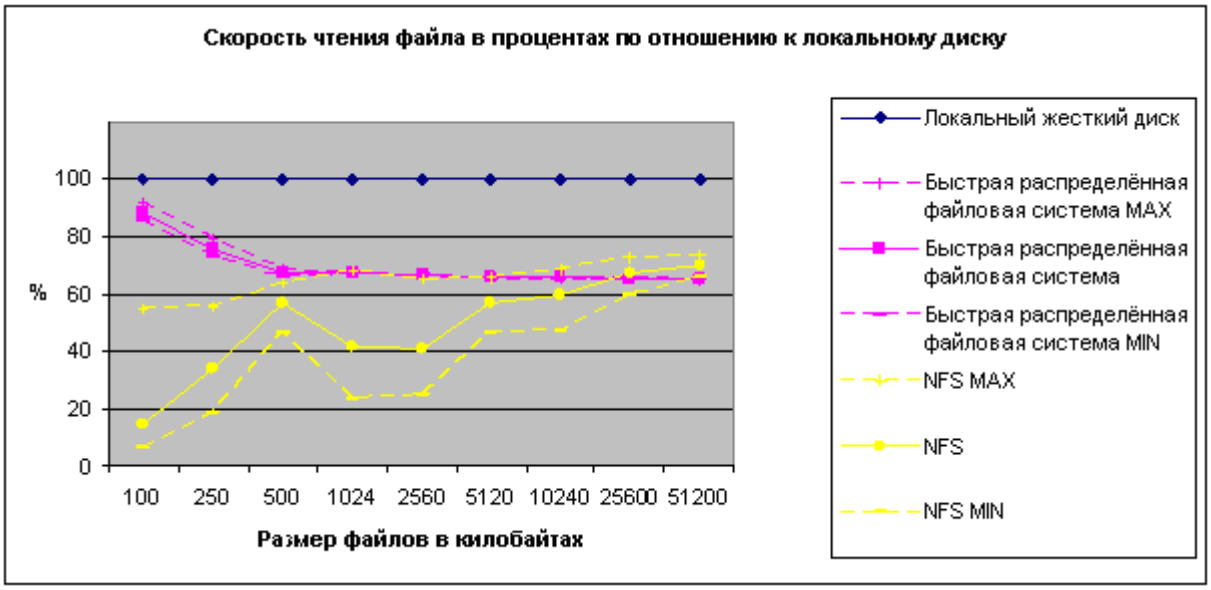

Рис. 5. График зависимости скорости первичного чтения файлов на узле–клиенте от типа файловой системы.

Рис. 5 позволяет оценить реактивность и скорость передачи данных в использовании БРФС. Реактивность при обращении к файлам, по сути, представляет собой время, затрачиваемое узлом–клиентом на получение метаданных файла. Метаданные файла, как правило, составляют не более 100КБ. Таким образом, в БРФС реактивность составляет не менее 90% от реактивности локальной файловой системы, тогда как в NFS – порядка 18%. При этом разброс этой величины для БРФС составляет порядка ±4%, тогда как для NFS – порядка ±30%.

Скорость чтения файлов в БРФС составляет порядка 65% – 90% скорости чтения файлов в локальной файловой системе, причём скорость падает с увеличением размера файлов. Такая динамика связана с тем, что с

увеличением размера файла увеличивается количество служебной информации файловой системы, которое в отличие от NFS также необходимо считать и обработать узлу–клиенту. Скорость чтения файлов в NFS составляет порядка 18% – 70%, причём с увеличением размера файлов она растёт. Такая динамика связана с предварительной подкачкой данных и кэшированием их на узле–сервере.

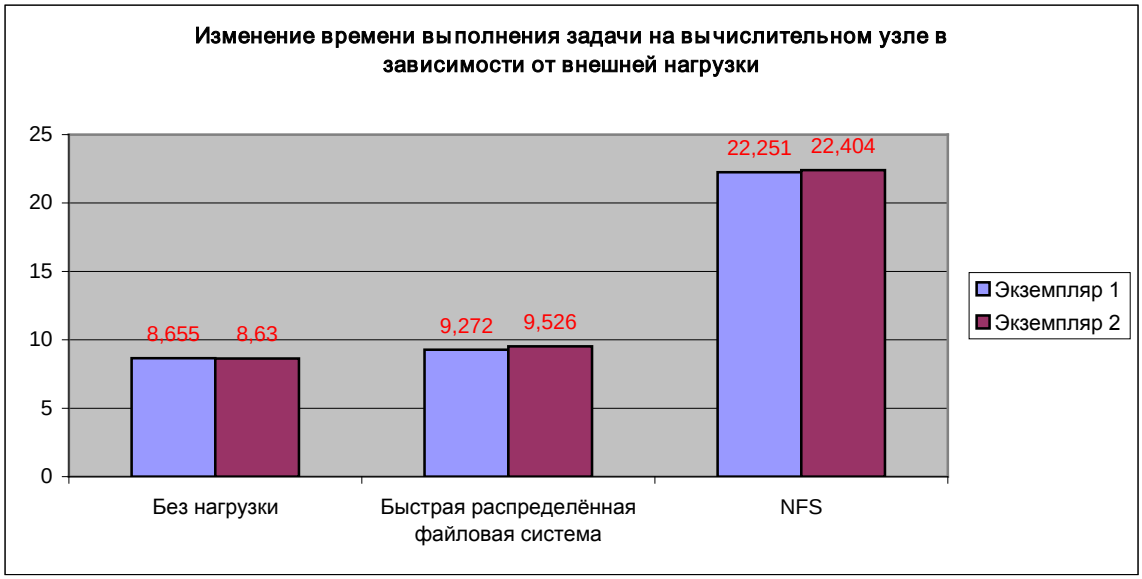

Рис. 6. График зависимости времени выполнения задачи на узле–сервере от внешней нагрузки.

Рис. 6 позволяет оценить нагрузку, оказываемую БРФС на узел–сервер, которая приводит к увеличению времени выполнения счётной задачи в среднем на 9%, тогда как использование NFS приводит к увеличению времени выполнения счётной задачи на 158%.

Результаты исследования показывают, что БРФС обладает лучшей реактивностью, существенно лучшей предсказуемостью времени доступа к данным и оказывает существенно меньшую нагрузку на узел–сервер по сравнению с NFS.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ РАБОТЫ

Основные результаты диссертационных исследований связаны с разработкой и реализацией методов доступа к удалённым данным для многомашинных вычислительных комплексов, включающих в себя бездисковые узлы и работающих в реальном масштабе времени.

В процессе выполнения диссертационных исследований были получены следующие основные результаты:

1. Проанализированы требования к методам доступа к удалённым данным в многомашинных вычислительных системах, позволяющих реализовать распределённую вычислительную систему, в том числе работающую в реальном масштабе времени.
2. Предложена методика анализа распределённых файловых систем на основе модели взаимодействия открытых систем (OSI), обеспечивающая наглядность архитектуры распределённой файловой системы и методов её реализации и позволяющая оценить её эффективность с точки зрения скорости доступа, скорости загрузки и рестарта и нагрузки, оказываемой на узел–сервер.
3. Проведён анализ современных распространённых распределённых файловых систем с точки зрения использования их в распределённых вычислительных системах, работающих в реальном масштабе времени, и обосновано требование создания новой архитектуры распределённой файловой системы.
4. Разработана оригинальная архитектура быстрой распределённой файловой системы, отличительными особенностями которой являются организация обменов на уровне блоков файловой системы, а не файлов, и реализация программ обслуживания удалённого доступа к данным в адресном пространстве ядра операционной системы.
5. Разработан протокол доступа к удалённым данным на уровне блоков файловой системы, который обеспечивает по сравнению с применяемыми в современных распространённых распределённых файловых системах протоколами лучшую реактивность и меньшую нагрузку на узел–сервер.
6. На основе разработанных методов и алгоритмов выполнена практическая реализация архитектуры быстрой распределённой файловой системы, обеспечивающей удовлетворение требований улучшения реактивности, увеличения скорости загрузки и рестарта, и снижения нагрузки на узел–сервер в распределённых вычислительных системах, работающих в реальном масштабе времени.
7. Проведено математическое и имитационное моделирование функционирования разработанной быстрой распределённой файловой системы и доказана эффективность её использования в распределённых вычислительных системах, работающих в реальном масштабе времени.

8. Основные результаты имитационного моделирования быстрой распределённой файловой системы показали эффективность следующих характеристик системы:
   - высокая скорость доступа к удалённым данным, составляющая порядка 90% от скорости доступа к локальным данным;
   - высокая предсказуемость времени доступа к данным: разброс составляет порядка 4%;
   - высокая скорость инициализации после старта/рестарта;
   - низкая нагрузка на узел–сервер – уменьшение производительности узла–сервера составляет не более 10% при интенсивном обмене данными с одним узлом–клиентом.

Список работ, опубликованных по теме диссертации

1. Кинсбурский С.А. Распределённая виртуальная файловая система для многомашинных комплексов // XXXII Гагаринские чтения. Научные труды Международной молодежной научной конференции в 8 т. Т. 6. М.: “МАТИ”–РГТУ, 2006. С. 159–161.
2. Кинсбурский С.А. Подход к реализации распределённой виртуальной файловой системы для многомашинных комплексов // XXXIII Гагаринские чтения. Научные труды Международной молодежной научной конференции в 8 т. Т. 6. М.: “МАТИ”–РГТУ, 2007. С. 234–235.
3. Кинсбурский С.А. Концепция виртуальной файловой системы на распределенных носителях // Информационные технологии, № 12, 2007. С. 12–15.
4. Кинсбурский С.А. Распределённая виртуальная файловая система // в/ч 03425 (НИЦ 4ЦНИИ МО РВ), XXIII научно-техническая конференция на тему "Направление развития и применения перспективных вычислительных систем и новых информационных технологий в ВВТ РКО", 2007 г.
5. Кинсбурский С.А. Характеристики распределённой виртуальной файловой системы // XXXIV Гагаринские чтения. Научные труды Международной молодежной научной конференции в 8 т. Т. 6. М.: “МАТИ”–РГТУ, 2008. С. 188–190.
6. Кинсбурский С.А. Распределённые файловые системы и основные проблемы повышения скорости доступа к удалённым данным // Компьютеры в учебном процессе. 2008. № 5. С. 3–12.
7. Кинсбурский С.А., ОКР "Созвездие – М 2", «Разработка предложений по использованию средств вычислительной техники в составе мобильного сервера, мобильных, переносных и переносимых ПТК», Шифр "Созвездие – М 2 – МЦСТ", ТВГИ.460659.018 ПЗ, 2008 г.