УДК 004.271.3

МОДЕЛЬ ПОТОКОВОЙ ОБРАБОТКИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

В. А. Щапов[1], А. Г. Масич[2], Г. Ф. Масич[2]

Современные экспериментальные установки порождают продолжительные по времени и интенсивные потоки данных. Ограниченность вычислительных ресурсов при установках и рост требований к точности алгоритмов приводят к невозможности их локальной обработки. Однако большая пропускная способность оптических лямбда-сетей позволяет перенести вычисления на удаленные суперкомпьютеры, при этом крайне важно организовать эффективный параллелизм во всех компонентах такой распределенной системы. Рассматривается модель потоковой обработки экспериментальных данных, ориентированная на использование высокоскоростных оптоволоконных сетей передачи данных большой протяженности для связи экспериментальной установки с высокопроизводительными вычислительными системами. Исследование выполнено при поддержке РФФИ (код проекта 11–07–96001-р_урал_а) и Уральского отделения РАН (код проекта 12–П–1–2012). Статья рекомендована к публикации программным комитетом Международной научной конференции “Научный сервис в сети Интернет: поиск новых решений” (http://agora.guru.ru/abrau).

Ключевые слова: экспериментальная установка, суперкомпьютер, скоростная оптическая сеть, протокол, BDP, распределенные вычисления, потоковая обработка данных.

1. Введение. Некоторые экспериментальные установки и системы визуализации генерируют большие объемы данных, нуждающихся в обработке. Одной из таких задач является обработка экспериментальных данных, получаемых по методу PIV (Particle Image Velocimetry) — оптическому методу измерения полей скорости жидкости или газа в выбранном сечении потока. Метод основан на обработке пар фотографий трассеров — мелких частиц, взвешенных в потоке, в моменты, когда они подсвечиваются импульсным лазером, создающим тонкий световой нож. Скорость потока определяется расчетом перемещения трассеров за время между вспышками лазера. Интенсивность порождаемого потока данных зависит от числа, разрешения и частоты работы камер и может достигать нескольких гигабит в секунду [1]. Обработка таких потоков данных с использованием только локального компьютера экспериментальной установки (ЭУ) чрезвычайно затруднена. В то же время развитие математического аппарата и появление новых высокоточных алгоритмов расчетов увеличивают требования к необходимой вычислительной мощности. Поэтому перенос вычислений на удаленные суперкомпьютеры достаточной производительности позволит применять новые высокоточные алгоритмы, обрабатывать данные в реальном времени, уменьшить объемы сохраняемых на ЭУ данных. Сопутствующей задачей является 4 K-визуализация в 24 кадра в секунду (FPS) в формате RGBA (32 бит на пиксель: с красными, зелеными, синими и альфа-каналами по 8 бит в каждом), которая требует 6.4 Gbps сетевой полосы пропускания к потоку от источника рендеринга до панелей отображения.

Существует значительный дисбаланс между вычислительной мощностью и скоростью ввода/вывода (I/O) суперкомпьютера, а также ограниченность традиционных протоколов передачи данных между оконечными системами, сопрягаемыми по сетям с большим BDP (Bandwidth Delay Product). Предлагаемое в настоящей статье архитектурное решение направлено на частичное устранение указанных проблем. Мы развиваем данное направление в рамках проекта “Распределенный PIV” [2] с использованием высокоскоростной оптической сети проекта “Инициатива GIGA UrB RAS” [3].

Оставшаяся часть статьи организована следующим образом. Во втором разделе описана инфраструктура распределенной системы, третий раздел поясняет ограниченность существующих решений. В чет-

[1] Институт механики сплошных сред Уральского отделения Российской академии наук (ИМСС УрО РАН), ул. Академика Королева, 1, 614013, г. Пермь; инженер, email: schapov@icmm.ru; Пермский национальный исследовательский политехнический университет (ПНИПУ), ул. профессора Поздеева, 7, 614013, г. Пермь; аспирант, email: schapov@icmm.ru

[2] Институт механики сплошных сред Уральского отделения Российской академии наук (ИМСС УрО РАН), ул. Академика Королева, 1, 614013, г. Пермь; А. Г. Масич, мл. науч. сотр., email: mag@icmm.ru; Г. Ф. Масич, зав. лабораторией, email: masich@icmm.ru

© Научно-исследовательский вычислительный центр МГУ им. М. В. Ломоносова

вертом разделе рассматривается модель потоковой обработки экспериментальных данных и реализующий ее протокол PIV. В пятом разделе приводятся результаты измерений и оценка эффективности модели.

2. Инфраструктура распределенной системы (рис. 1). *Коммуникационный фрагмент инфраструктуры* создается в рамках проекта "Initiative GIGA UrB RAS" — скоростная DWDM научно-образовательная магистраль Уральского Отделения РАН "Архангельск–Сыктывкар–Ижевск–Пермь–Екатеринбург".

Доступная для использования в проекте "Распределенный PIV" реализация на участке "Пермь–Екатеринбург" [8] состоит из трех λ-каналов по 10 Гбит/с и имеет следующую структуру/порты для сопряжения оконечных систем: $\lambda 1$ — 1x10GE (гарантировано), $\lambda 2$ — 8x1GE (независимы и гарантированы), $\lambda 3$ — размножается на 24x1GE [3]. Время распространения сигнала по оптоволокну составляет 5 мкс/км и для оптической трассы "Пермь–Екатеринбург" (456 км$\times$5 мкс/км) равно примерно 2.3 мс без учета задержек в очередях L2-коммутационного облака оконечных систем. Эта величина задержки экспериментально подтверждается измерением RTT (Round Trip Time): 5 мс. Оценку размера окна транспортных протоколов с обратной связью для гарантированных каналов в оптических лямбда-сетях можно выполнить посредством вычисления BDP = bandwidth * RTT, которая будет равна 0.625 или 6.25 Мбайт для скоростей 1 Гбит/с или 10 Гбит/с соответственно.

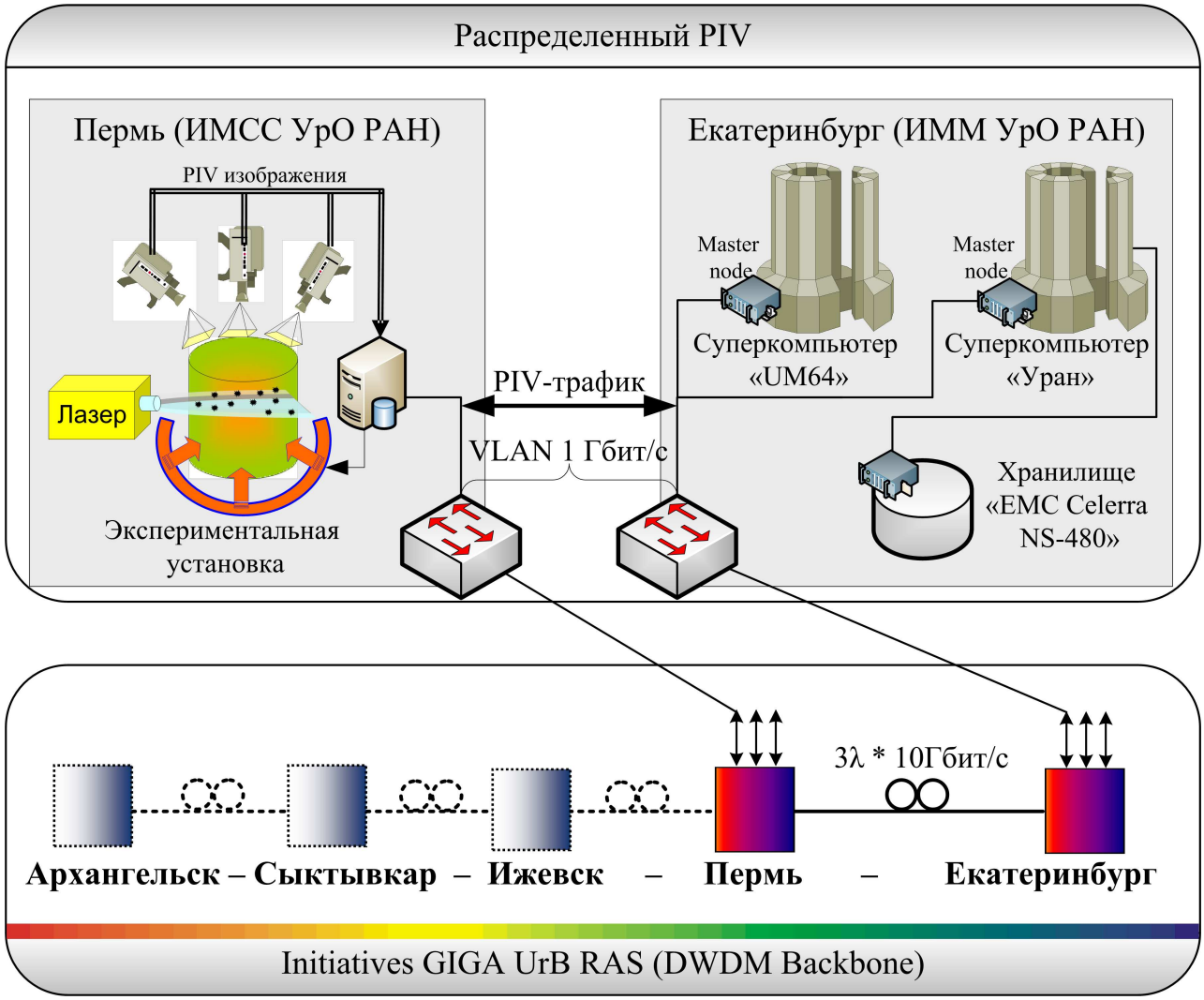

Рис. 1. Инфраструктурная схема проектов GIGA UrB RAS и "Распределенный PIV"

Сопрягаемые оконечные системы. ЭУ (Пермь, ИМСС УрО РАН) соединяется с суперкомпьютерами "Уран" и "UM64" и системой хранения данных (СХД) EMC Celerra NS-480 (Екатеринбург, ИММ УрО РАН) по Ethernet-технологии на скорости 1 Гбит/с. Обеспечивается прямое соединение на втором уровне модели OSI RM с использованием технологии VLAN. В оконечных системах используется стек протоколов TCP/IP. На последующих этапах планируется апробировать лямбда-параллелизм потоков данных в

DWDM-тракте передачи данных.

3. Оценка существующих решений. Классическим способом передачи данных на суперкомпьютер является подключение к СХД суперкомпьютера по протоколам CIFS и NFS или загрузка данных через управляющий узел по протоколам SCP, GridFTP и др.

Существенным недостатком этих подходов является высокая сложность реализации расчета на нескольких супервычислителях, так как это требует внедрения общей для суперкомпьютеров СХД, разработки алгоритмов межсуперкомпьютерной синхронизации вычислительных узлов и алгоритмов распределения данных по узлам между суперкомпьютерами (при этом данный алгоритм придется реализовывать на вычислительных узлах). Вторым недостатком использования СХД для передачи данных является недостаточная эффективность названных протоколов на протяженных линиях связи. Как показали исследования, она может не превышать 50–60% при использовании выделенного канала Пермь–Москва пропускной способностью 1 Гбит/с [2].

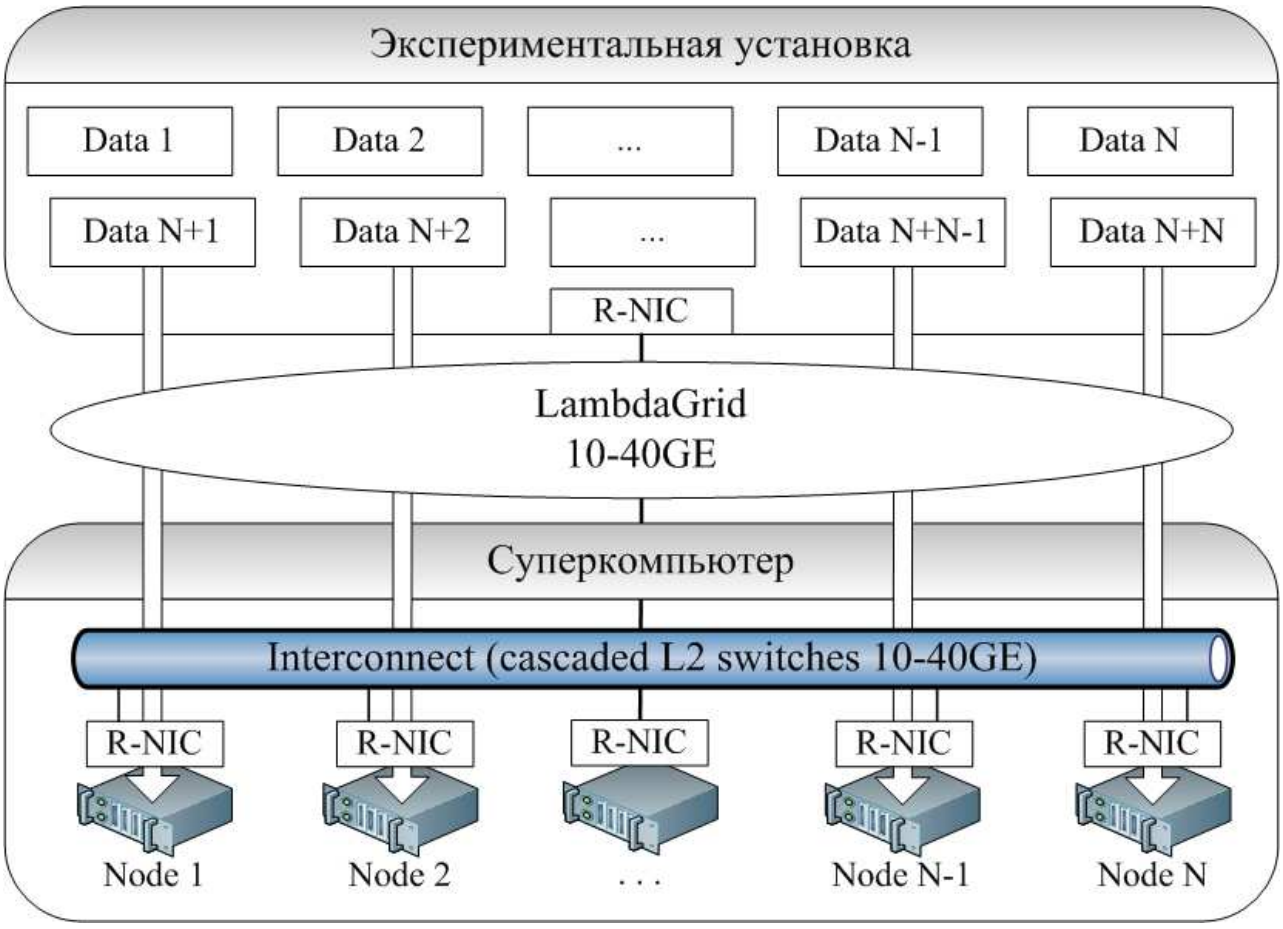

Рис. 2. Модель обработки интенсивных потоков данных

4. Модель обработки данных на основе сетевого взаимодействия. Суть модели — прямой ввод потока данных в оперативную память вычислительных узлов удаленного суперкомпьютера без предварительного сохранения на внешних носителях ЭУ или СХД суперкомпьютера (рис. 2). Такой способ передачи данных позволит минимизировать задержки на двойную запись/чтение данных и отказаться от копирования данных между хранилищами с использованием классических протоколов SCP, FTP, NFS или CIFS.

Благодаря тому, что каждое измерение в эксперименте PIV может обрабатываться независимо от других [4] и существует один источник данных на ЭУ, было предложено отказаться от однозначного отображения измерений на вычислительные узлы. На экспериментальной установке организуется очередь измерений, которые раздаются вычислительным узлам по принципу First-In-First-Out (FIFO), как показано на рис. 3. Эта подсистема программного обеспечения называется диспетчером.

Несмотря на то что извлечение элемента из очереди — это блокирующая операция, после извлечения измерений из очереди они могут передаваться клиентам параллельно. В случае наличия большого

количества конкурентных запросов число параллельных передач определяется доступными ресурсами на компьютере ЭУ, при этом остальные вычислительные узлы будут простаивать в ожидании появления доступных ресурсов (в основном, оперативной памяти). Это позволяет использовать канал связи максимально эффективным образом.

Взаимодействие ЭУ и вычислительных узлов реализуется с использованием протокола, получившего название “Протокол PIV”, построенного на идее модели RPC. При этом ЭУ является сервером, а на вычислительных узлах работают клиенты.

Полный перенос задачи диспетчеризации в программное обеспечение, устанавливаемое на ЭУ, позволил отказаться от межузлового обмена данными на стороне суперкомпьютера, что дает следующие преимущества:

— автоматическая балансировка нагрузки по вычислительным узлам: более быстрые узлы будут чаще посылать запросы на новые данные и, таким образом, будут получать больше данных для обработки;

— возможность изменять число используемых вычислительных узлов во время проведения эксперимента;

— возможность использовать вычислительную мощность нескольких суперкомпьютеров;

— минимизация потери данных в случае выхода из строя одного или нескольких вычислительных узлов (в худшем случае потеряется только текущее обрабатываемое измерение сбойного узла). Если потеря данных недопустима, то на время расчета измерения необходимо сохранять в памяти сервера и в случае выявления сбоя повторно добавлять эти блоки в очередь готовых к обработке измерений.

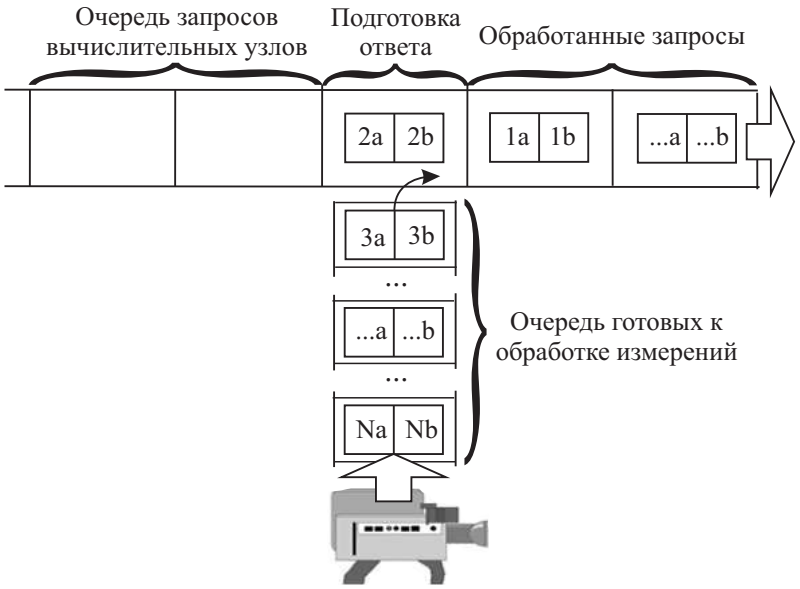

Рис. 3. Модель обработки очереди исходных данных

Необходимо отметить, что предлагаемый подход не ограничивает нас в выборе транспорта для передачи данных. Текущая реализация использует для передачи данных протокол PIV. Однако возможно и использование общей СХД, подключенной и к ЭУ, и к супервычислителю. В этом случае задачей диспетчера будет распределение по вычислительным узлам не самих данных, а путей к файлам с данными на СХД, при этом вычислительные узлы будут считывать данные непосредственно из файлов на СХД. Это позволит анализировать и сравнивать поведение как классической схемы с общим дисковым пространством, так и предлагаемой схемы без промежуточного хранения данных на дисковых массивах.

В настоящий момент реализован подход, когда при наличии данных в очереди они будут отдаваться сервером в ответ на все поступающие запросы при наличии доступных ресурсов на компьютере ЭУ. Однако возможно, что с точки зрения загрузки сети и отзывчивости системы данный подход не будет являться оптимальным. Исследование поведения системы и выбора оптимальных стратегий распределения очереди в условиях большого числа вычислительных узлов (более 512) будут предметом дальнейших исследований. Предполагается определить такую конфигурацию параметров, при которой будет соблюдаться баланс между числом активных параллельных соединений, уровнем загрузки сети и временем получения результата расчета после начала его отправки (отзывчивость системы).

Протокол PIV является протоколом прикладного уровня, реализующим идею модели RPC. Протокол работает по схеме запрос-ответ и может использовать в качестве протокола транспортного уровня любой надежный потоковый протокол передачи данных. Текущая реализация протокола PIV поддерживает транспортные протоколы TCP и UDT [5]. Положение протокола PIV в стеке сетевых протоколов показано на рис. 4.

Протокол UDT — это основанный на UDP протокол передачи данных для высокоскоростных сетей. Он был разработан в Университете штата Иллинойс в Чикаго. Функциональные возможности протокола UDT аналогичны протоколу TCP. UDT является дуплексным протоколом передачи потока данных с предварительной установкой соединения. Особенностью протокола UDT является оригинальная архитек-

тура и реализация, а также оригинальный алгоритм управления перегрузкой. При этом протокол UDT позволяет программисту реализовать и использовать свой алгоритм управления перегрузкой.

Протокол PIV рассчитан на передачу нескольких блоков бинарных данных. В одном пакете протокола можно передать от нуля до 65 535 блоков, каждый из которых может иметь размер до 4 Гбайт. Формат пакета протокола PIV приведен на рис. 5. Поля заголовка пакета протокола кодируются в сетевом порядке байт.

Протокол поддерживает три типа пакетов (поле “Cmd” протокола PIV):

1) GET — предназначен для запроса данных вычислительными узлами у ЭУ;

2) POST — предназначен для передачи данных с вычислительных узлов на ЭУ;

3) RESPONSE — пакет ответа на запрос вычислительного узла.

Информация о подтипе RESPONSE-пакета кодируется в поле статуса. Через поле статуса вычислительным узлам сообщается, успешно ли был обработан запрос или нет. В случае неуспеха конкретное значение поля статуса определяет произошедшую ошибку.

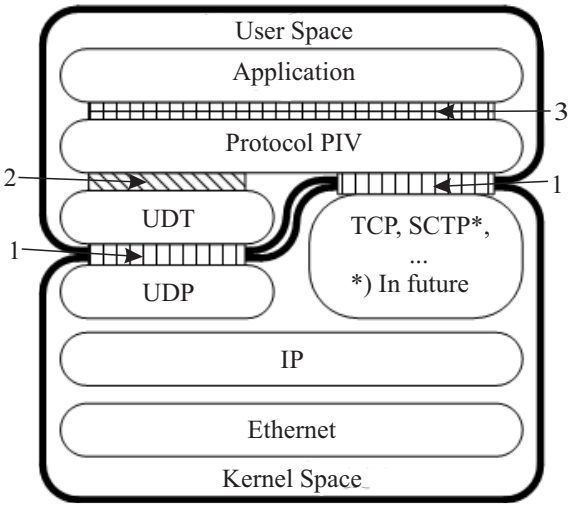

Рис. 4. Положение протокола PIV в стеке сетевых протоколов: 1) Socket API, 2) UDT Socket API, 3) PIV library API (C++)

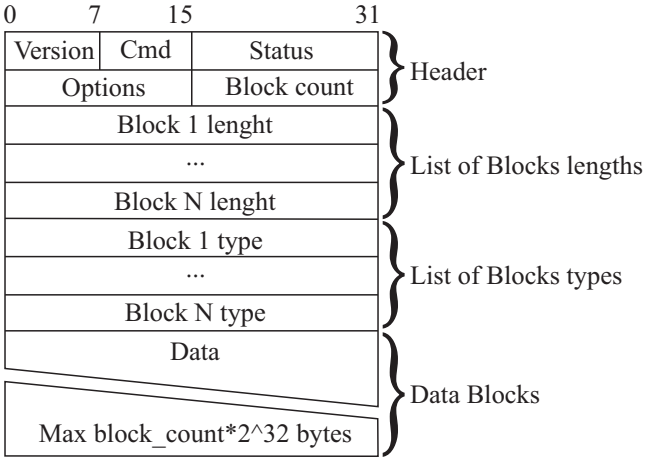

Рис. 5. Формат пакета протокола PIV

Программное обеспечение, реализующее передачу данных с помощью предложенной модели, разработано с использованием языка программирования C++ и библиотек Boost.

Язык C++ был выбран по причине удобства разработки и его распространенности на суперкомпьютерах. Использование библиотек Boost позволило упросить кросплатформенную реализацию асинхронной работы с сетевыми соединениями.

Программное обеспечение состоит из трех логических компонентов: подсистема загрузки данных; диспетчер (менеджер очередей); клиентская библиотека.

В текущей реализации подсистема загрузки данных и диспетчер объединены в одном приложении (сервер), однако при необходимости они могут быть реализованы в виде двух разных приложений. Клиентская библиотека предоставляет API для написания расчетных приложений, запускаемых на суперкомпьютере. В дополнение к этому реализовано расчетное приложение с использованием клиентской библиотеки, реализующее кросскореляционный алгоритм расчета PIV-данных.

Сервер работает непосредственно на экспериментальной установке и отвечает за загрузку данных и их распределение по вычислительным узлам. Для минимизации затрат ресурсов сетевое взаимодействие реализовано по асинхронной модели, что позволяет эффективно обрабатывать большое количество параллельных соединений в небольшом числе процессов.

Клиентская библиотека предоставляет API, которое позволяет запрашивать у сервера новый блок исходных данных и отправлять на сервер результаты расчетов (или подтверждение об успешном расчете, если отправка полных результатов на ЭУ не требуется).

5. Оценка эффективности применения предложенной модели. Оценка эффективности проводилась с использованием приведенной на рис. 1 инфраструктуры. Вычислительные узлы получали доступ к компьютеру экспериментальной установки через головной узел суперкомпьютера с использованием технологии PAT. Тестовый набор данных состоял из 1000 ранее полученных измерений размером 13.6 Мбайт

каждое, которые хранились на жестком диске. Реальный расчет данных не проводился, вместо этого в программу расчета был добавлен вызов функции sleep, при помощи которого можно было эмулировать различное время расчета. На ЭУ возвращалось только подтверждение успешного "расчета" без передачи дополнительных данных.

На рис. 6 приведен график зависимости суммарного времени обработки всего множества данных от числа задействованных вычислительных узлов для случаев с разным временем обработки одного блока данных. Для наглядности ось "число вычислительных узлов" приведена в логарифмическом масштабе. Эмулировались четыре значения времени расчета одного блока данных — 0 с, 1 с, 2 с и 4 с.

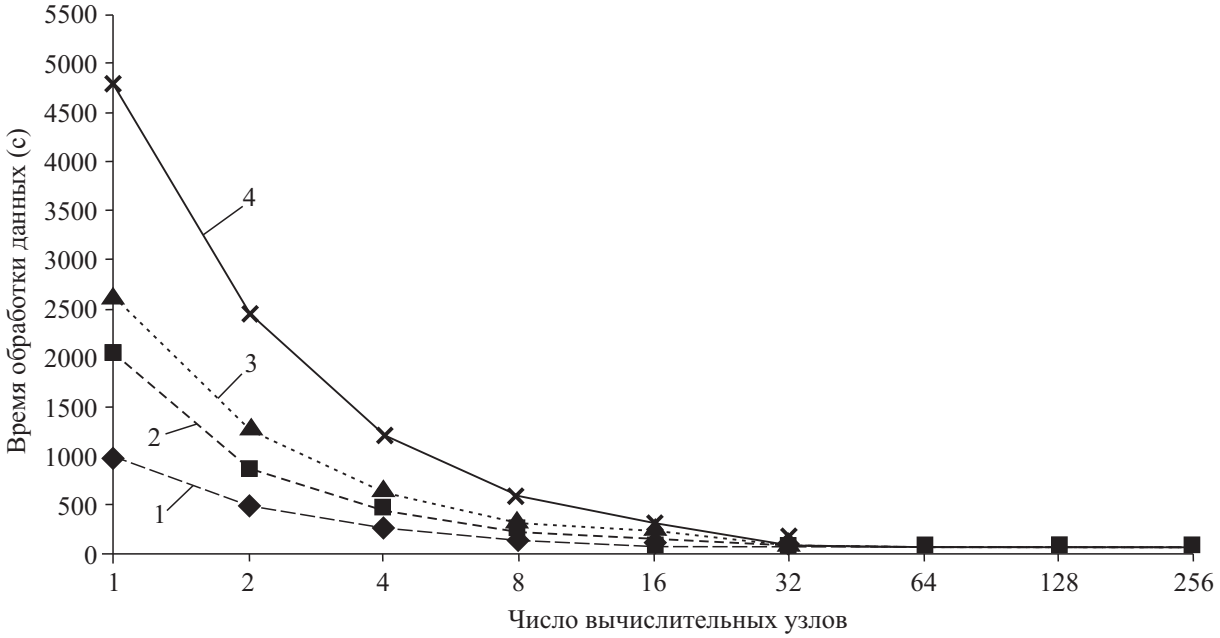

Рис. 6. Суммарное время обработки множества данных: 1) время расчета [0 с]; 2) время расчета [1 с]; 3) время расчета [2 с]; 4) время расчета [4 с]

На графике можно четко выделить две области. В первой области с ростом числа вычислительных узлов время обработки уменьшается, во второй — остается неизменным.

Горизонтальный участок графика характеризует ситуацию, когда общая пропускная способность системы ограничивается пропускной способностью сети передачи данных. Минусом работы в этой области является то, что чем больше задействовано вычислительных узлов, тем больше времени они простаивают в ожидании новых данных. Плюсом является повышение надежности системы с увеличением числа используемых узлов, так как в этом случае аварийное отключение некоторых узлов не приведет к выходу с горизонтального участка графика и тем самым обеспечит неизменную пропускную способность системы.

Наклонный участок графика показывает, как растет пропускная способность системы с ростом числа вычислительных узлов из-за того, что параллельные потоки в сети начинают использовать все больший процент доступной пропускной способности канала связи. Такая картина наблюдается по причине того, что из-за характеристик транспортных протоколов TCP (UDT) малое число параллельных потоков протокола PIV не могут занять всю пропускную способность канала передачи данных. Необходимо отметить, что параметры наклонного участка напрямую зависят от времени расчета одного измерения. Чем дольше вычислительный узел участвует в расчете данных, тем больше вычислительных узлов необходимо для того, чтобы в системе оставались доступные ресурсы для приема новых измерений.

Эмпирически можно установить следующие закономерности. Для расчета уже существующего набора данных число вычислительных узлов лучше выбирать из области, расположенной левее начала горизонтального участка графика. Эта область даст разумный баланс между используемыми вычислительными ресурсами и временем расчета. В случае расчета данных в реальном времени число вычислительных узлов должно находиться на горизонтальном участке графика. В этом случае часть вычислительных ресурсов будет простаивать, но так как лимитирующим фактором будет являться сеть передачи данных, то общая пропускная способность системы будет близка к пропускной способности сети. В результате будет достигнута максимальная для данной конфигурации пропускная способность системы. Для повышения отказоустойчивости число вычислительных узлов рекомендуется выбирать с запасом, который возьмет

на себя возросшую нагрузку и не даст перейти системе в левую часть графика при выходе из строя части вычислительных узлов.

Разработанное решение используется лабораторией физической гидродинамики ИМСС УрО РАН в работах по экспериментальному изучению модели формирования циклонов и антициклонов в атмосфере [6, 7].

6. Заключение. Предложена модель потоковой обработки экспериментальных данных в распределенных системах. Спроектирован протокол, алгоритм диспетчеризации данных и разработано программное обеспечения для передачи данных с экспериментальной установки на узлы вычислительного кластера, тестирование которых подтверждает работоспособность предложенной модели потоковой обработки экспериментальных данных.

Оценка эффективности показала уменьшение времени обработки массивов исходных данных и увеличение пропускной способности системы при использовании разработанной технологии. Показан способ эмпирической оценки необходимого числа вычислительных узлов. Разработанная технология предоставляет принципиально новый инструмент проведения экспериментальных исследований, позволяя обрабатывать быстропротекающие процессы в течение длительного времени, например, при лабораторном изучении начальной стадии формирования тропических циклонов.

Дальнейшие исследования будут направлены на детальную разработку методики оценки необходимого вычислительного и коммуникационного ресурса: числа λ-каналов и количества конкурирующих потоков в одном λ-канале в зависимости от параметров эксперимента и расчетного алгоритма (время расчета). Кроме того, предполагается разработка оптимальных стратегий распределения очереди данных по вычислительным узлам с учетом требований к пропускной способности, а также анализ отзывчивости системы, повышение эффективности использования сети передачи данных и изучение влияния настроек сетевых подсистем и коммутационного оборудования на эффективную пропускную способность сети. Планируется совершенствование модели с целью максимального использования ресурсов внутренних сетей суперкомпьютера, таких как сеть ввода/вывода и сеть передачи сообщений MPI.

СПИСОК ЛИТЕРАТУРЫ

1. *Степанов Р.А., Масич А.Г., Сухановский А.Н., Щапов В.А., Игумнов А.С., Масич Г.Ф.* Обработка на супервычислителе потока экспериментальных данных // Тр. Междунар. суперкомпьютерной конференции “Научный сервис в сети Интернет: экзафлопсное будущее”. М.: Изд-во Моск. ун-та, 2011. 168–174.
2. *Степанов Р.А., Масич А.Г., Масич Г.Ф.* Инициативный проект “Распределенный PIV” // Тр. Всероссийской суперкомпьютерной конференции “Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность”. М.: Изд-во Моск. ун-та, 2009. 360–363.
3. *Масич А.Г., Масич Г.Ф.* Инициатива GIGA UrB RAS // Совместный вып. журн. “Вычислительные технологии” и журн. “Вестник КазНУ им. Аль-Фараби. Сер.: матем., механ., информатика” № 3 (58). Т. 13, Ч. II. Алматы, 2008. 413–418.
4. *Масич А.Г., Масич Г.Ф., Степанов Р.А., Щапов В.А.* Скоростной I/O-канал супервычислителя и протокол обмена интенсивным потоком экспериментальных данных // Тр. X Междунар. конференции “Высокопроизводительные параллельные вычисления на кластерных системах HPC-2010”. Т. 2. Пермь: Изд-во ПГТУ, 2010. 119–128.
5. *Gu Y., Grossman R.L.* UDT: UDP-based data transfer for high-speed wide area networks // Computer Networks. 2007. **51**, N 7. 1777–1799.
6. *Баталов В.Г., Сухановский А.Н., Фрик П.Г.* Экспериментальное исследование спиральных валов в адвективном потоке, натекающем на горячую горизонтальную поверхность // Изв. РАН. Механ. жидкости и газа. 2007. № 4. 50–60.
7. *Batalov V., Sukhanovsky A., Frick P.* Laboratory study of differential rotation in a convective rotating layer // J. Geophys. Astrophys. Fluid Dynam. 2010. **104**, N 4. 349–368.
8. *Масич А.Г., Масич Г.Ф., Матвеенко В.П., Тирон Г.Г.* Инициатива GIGA UrB RAS: методология построения и архитектура научно-образовательной оптической магистрали Уральского отделения РАН // Тр. Междунар. конференции “Математические и информационные технологии” MIT-2011. Белград, 2012. 257–265.

Поступила в редакцию
16.09.2012