МОДЕЛЬ ПОТОКОВОЙ ОБРАБОТКИ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ

В.А. Щапов, А.Г. Масич, Г.Ф. Масич

Аннотация. Современные экспериментальные установки порождают продолжительные по времени и интенсивные потоки данных. Ограниченность локальных вычислительных ресурсов 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. В статье рассматривается модель потоковой обработки экспериментальных данных в распределенных системах, использующих высокоскоростные оптоволоконные сети передачи данных большой протяженности. Исследование проводится при поддержке РФФИ (грант № 11-07-96001-р_урал_а) и УрО РАН (грант 12-П-1-2012).

1. Введение. Современные 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 тоже 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. Мы развиваем данное направление в рамках проекта «Распределенный PIV» [2] с использованием высокоскоростной оптической магистрали, создаваемой в рамках проекта «Инициатива GIGA UrB RAS» [3].

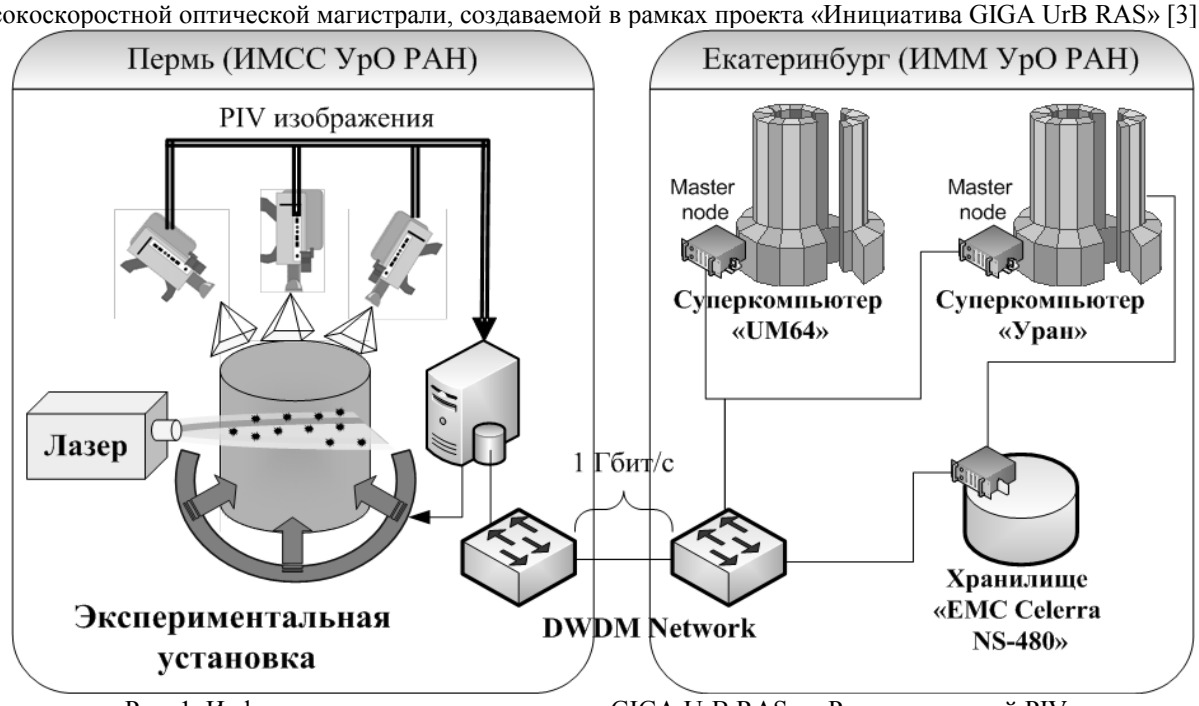

Рис. 1. Инфраструктурная схема проектов GIGA UrB RAS и «Распределенный PIV»

На рисунке 1 показана инфраструктурная схема проектов. ЭУ (Пермь, ИМСС УрО РАН) соединяется с суперкомпьютерами «Уран», «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 уровне 2 модели OSI RM с использованием технологии VLAN. В оконечных системах используется стек протоколов TCP/IP, а L2 OSI RM соединение между оконечными системами позволяет избежать затрат времени на

маршрутизацию и очередей пакетов в IP сети передачи данных. На последующих этапах планируется апробировать лямбда параллелизм потоков данных в DWDM-тракте передачи данных 40 Гбит/с.

Сопутствующей задачей является 4K визуализация в 24 кадра в секунду (FPS) в формате RGBA (32bit 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. Поэтому планируемая скорость в оптической магистрали «Екатеринбург — Пермь — Ижевск — Сыктывкар — Архангельск» —16 лямбда каналов по 10/40 Гбит/с в каждом канале [3].

Существует также, как существенный дисбаланс между вычислительной мощностью и скоростью ввода/вывода (I/O) суперкомпьютера, так и ограниченность традиционных протоколов передачи данных между оконечными системами, сопрягаемым по сетям с большим BDP (bandwidth delay product). Предлагаемое в статье архитектурное решение направлено на частичное решение этих проблем. Оставшаяся часть статьи организована следующим образом. Вторая часть поясняет ограниченность существующих решений. В третьей части рассматривается модель потоковой обработки экспериментальных данных и реализующий ее протокол PIV. В четвертой части приводятся результаты измерений и оценка эффективности модели.

2. 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 т.д.

Существенным недостатком данных подходов является высокая сложность реализации расчета на нескольких супервычислителях, так как это потребует 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 вышеназванных протоколов на длинных линиях связи. Как показали исследования, она может не превышать 50-60% при использовании выделенного канала Пермь-Москва пропускной способностью 1 Гбит/с [2].

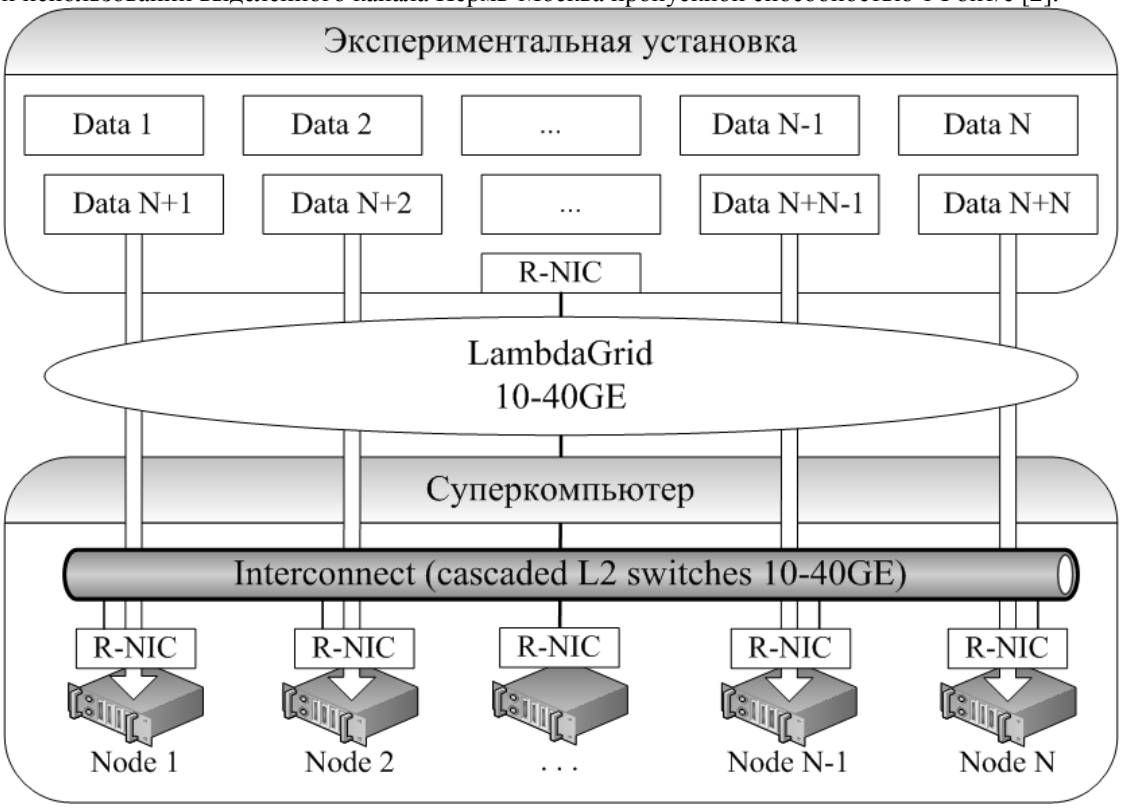

Рис. 2 Модель обработки интенсивных потоков данных

3. 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 (рисунок 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 рисунке 3. Эта подсистема программного обеспечения называется диспетчером.

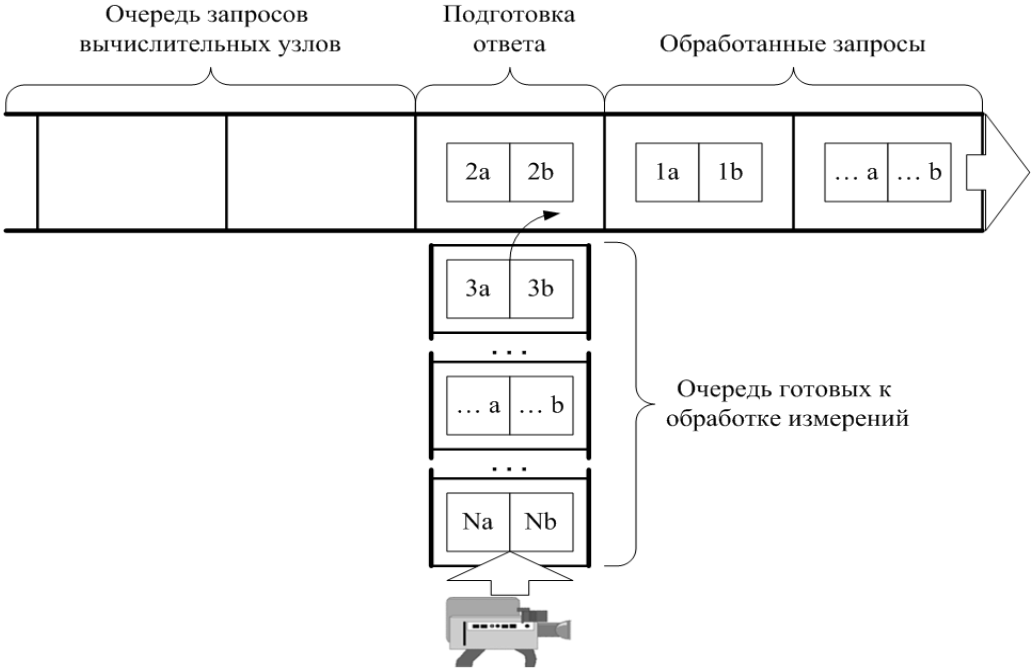

Рис. 3 Модель обработки очереди исходных данных

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 новых данных 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). В случае, 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.

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 общего СХД, подключенного 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 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 рисунке 4.

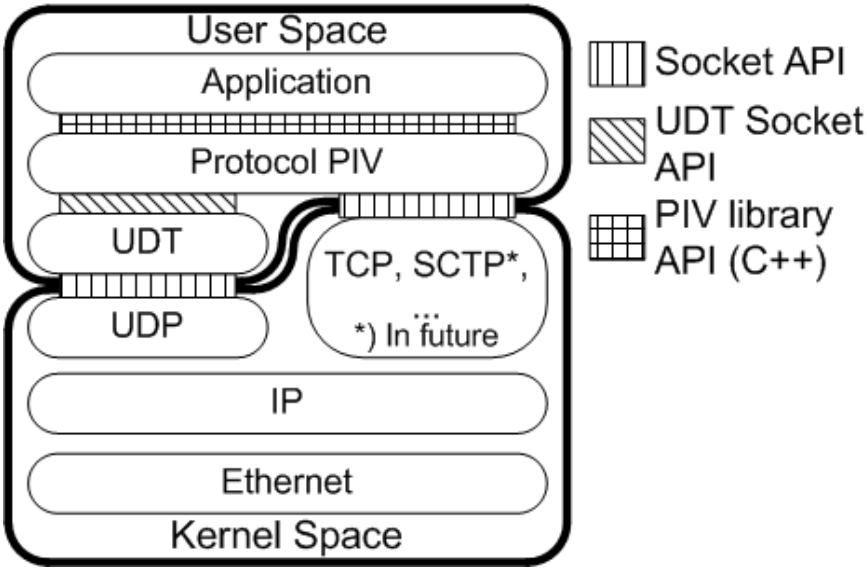

Рис. 4 Положение протокола PIV в стеке сетевых протоколов

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

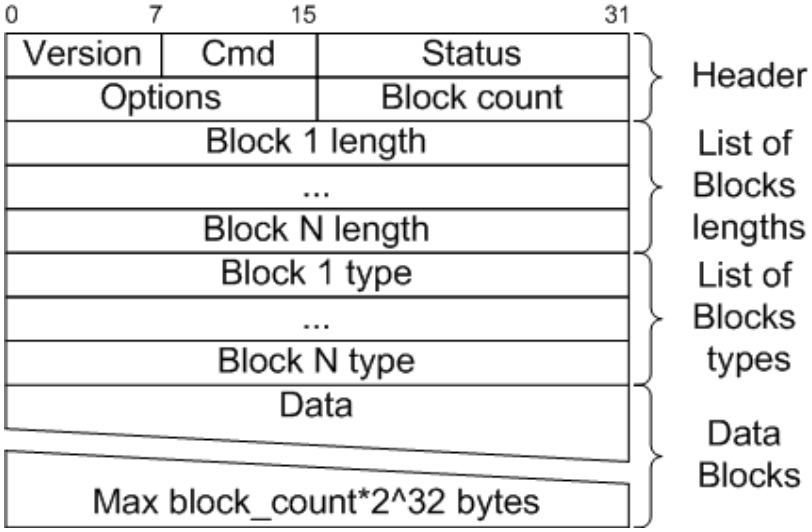

Рис.5 Формат пакета протокола PIV

Протокол PIV рассчитан на передачу нескольких блоков бинарных данных. В одном пакете протокола можно передать от нуля до 65535 блоков, каждый из которых может иметь размер до 4 Гбайт. Формат пакета протокола PIV приведен на рисунке 5. Поля заголовка пакета протокола кодируются в сетевом порядке байт.

Протокол поддерживает три типа пакетов (поле «Cmd» протокола PIV):

- GET – предназначен для запроса данных вычислительными узлами у ЭУ;
- POST – предназначен для передачи данных с вычислительных узлов на ЭУ;
- 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.

4. Оценка эффективности применения предложенной модели. Оценка эффективности проводилась с использованием приведенной на рисунке 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 рисунке 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 — 0c, 1c, 2c и 4c.

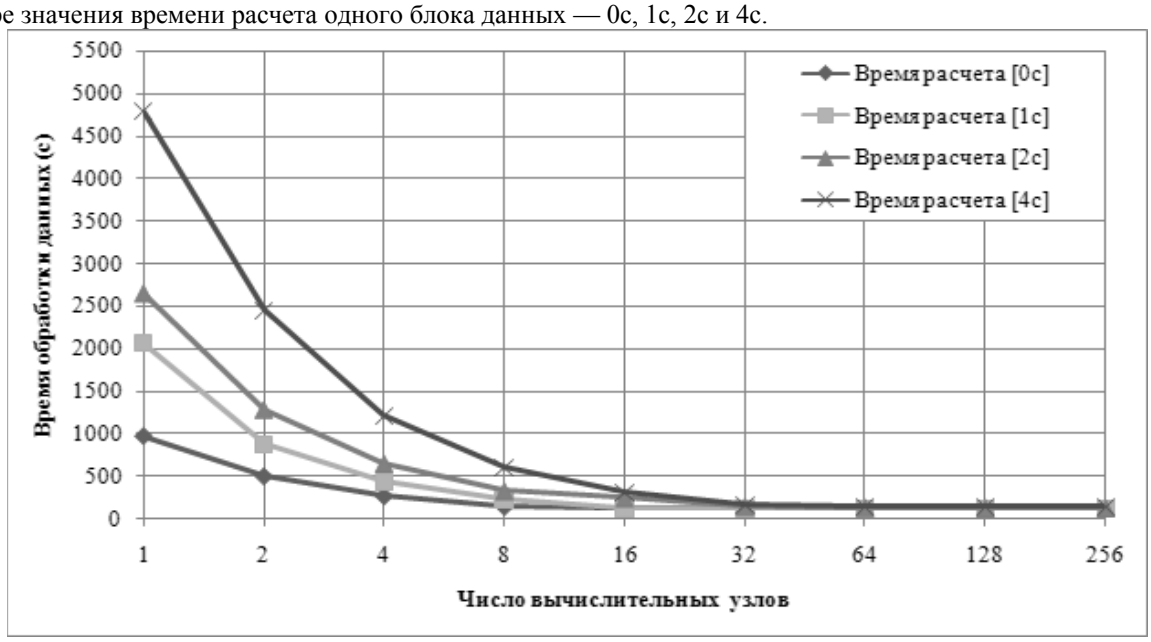

Рис. 6 Суммарное время обработки множества данных

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 больше 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, 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].

5. 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 подтверждают 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, исходя из параметров эксперимента и расчетного алгоритма, изучение влияния настроек сетевых подсистем и коммутационного оборудования на эффективную пропускную способность сети, а также на совершенствование модели для максимального использования ресурсов внутренних сетей суперкомпьютера, таких, как сеть ввода вывода и сеть передачи сообщений MPI.

ЛИТЕРАТУРА:

1. Р.А. Степанов, А.Г. Масич, А.Н. Сухановский, В.А. Щапов, А.С. Игумнов, Г.Ф. Масич «Обработка на супервычислителе потока экспериментальных данных» // Научный сервис в сети Интернет: экзафлопсное будущее: Труды Международной суперкомпьютерной конференции (19-24 сентября 2011 г., г. Новороссийск). – М.: Изд-во МГУ, 2011. С. 168-174. ISBN 978-5-211-06229-0
2. Р.А. Степанов, А.Г. Масич, Г.Ф. Масич «Инициативный проект «Распределенный PIV»» // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: труды Всероссийской суперкомпьютерной конференции – М.: Изд-во МГУ, 2009. – С. 360–363. (ISBN 978-5-211-05697-8)
3. А.Г. Масич, Г.Ф. Масич «Инициатива GIGA UrB RAS» // Совместный вып. Журнала «Вычислительные технологии» и журн. «Вестник КазНУ им. Аль-Фараби». Сер. «Математика, механика, информатика», №3 (58). По материалам Междунар. конф. «Вычислительные и информационные технологии в науке, технике и образовании», - Казахстан, Алматы.-2008.-Т.13.- Ч. II. -С. 413-418 (ISSN 1560-7534)
4. А.Г. Масич, Г.Ф. Масич, Р.А. Степанов, В.А. Щапов Скоростной I/O-канал супервычислителя и протокол обмена интенсивным потоком экспериментальных данных // Материалы X международной конференции «Высокопроизводительные параллельные вычисления на кластерных системах HPC-2010» – Пермь: Изд-во ПГТУ, 2010. - Т. 2. С. 119–128. (ISBN 978-5-398-00506-6)
5. Yunhong Gu and Robert L. Grossman, UDT: UDP-based Data Transfer for High-Speed Wide Area Networks, Computer Networks (Elsevier). Volume 51, Issue 7. May 2007.
6. В.Г. Баталов, А.Н. Сухановский, П.Г. Фрик. Экспериментальное исследование спиральных валов в адвективном потоке, натекающем на горячую горизонтальную поверхность. Известия РАН. Механика жидкости и газа. №4. 2007. С. 50-60.
7. Batalov V., Sukhanovsky A. and Frick P. Laboratory study of differential rotation in a convective rotating layer // J. Geophys.Astrophys.Fluid Dynam. 104: 4, pp. 349 — 368, 2010. – DOI: 10.1080/03091921003759876.