В ходе работ были проведены испытания разработанной платформы на базе вычислительных инфраструктур, работающих в рамках моделей метакомпьютинга и Грид. Результаты испытаний показали возможность эффективного использования разработанной платформы в качестве надстройки над вычислительными ресурсами, доступными в рамках данных моделей.

Список литературы

1. Boccara N. Modeling Complex Systems. // Springer New York, 2004. — 397 p.

2. Rice J.R., Boisvert R. F. From Scientific Software Libraries to Problem-Solving Environments // IEEE Computational Science & Engineering. – 1996. – Vol. 3. No. 3. – P. 44–53.

3. Бухановский А.В., Ковальчук С.В., Марьин С.В. Интеллектуальные высокопроизводительные программные комплексы моделирования сложных систем: концепция, архитектура и примеры реализации // Изв. высших учебных заведений. Приборостроение. – 2009. – № 10. – С. 5–24.

4. What Are Ontologies, and Why Do We Need Them? / B. Chandrasekaran, J.R. Josephson, V.R. Benjamins // IEEE Intelligent Systems. 1999. – Vol. 14, No. 1. – P. 20–26.

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

[1] Институт механики сплошных сред УрО РАН, г. Пермь
[2] Пермский государственный технический университет

СКОРОСТНОЙ I/O-КАНАЛ СУПЕРВЫЧИСЛИТЕЛЯ И ПРОТОКОЛ ДЛЯ ОБМЕНА ИНТЕНСИВНЫМ ПОТОКОМ ЭКСПЕРИМЕНТАЛЬНЫХ ДАННЫХ

Общие сведения об эксперименте

Экспериментальная установка (ЭУ) использует метод PIV (Particle Image Velocimetry) [4] – оптический метод измерения полей скорости жидкости или газа в выбранном сечении потока. Принцип метода: импульсный лазер создает тонкий световой

нож и освещает мелкие частицы, взвешенные в исследуемом потоке. Положения частиц в момент двух последовательных вспышек лазера регистрируются на два кадра цифровой камеры. Скорость вихревого потока определяется расчетом перемещения, которое совершают частицы за время между вспышками лазера. Измерительная часть установки генерирует поток данных 1-10-100 Гбит/с в зависимости от разрешения, частоты и режимов работ камер (моно/стерео/томография).

Области применения PIV: гидро- и аэродинамика лабораторных течений, физическое моделирование технологических процессов в энергетике, химической промышленности, диагностика обтекания реальных и модельных объектов в авиа- и автомобилестроении и т.д.

Разработанные архитектурные решения

Порты ввода вывода и коммуникации

Идея основана на прямом вводе в память узлов супервычислителя экспериментальных данных (рис. 1).

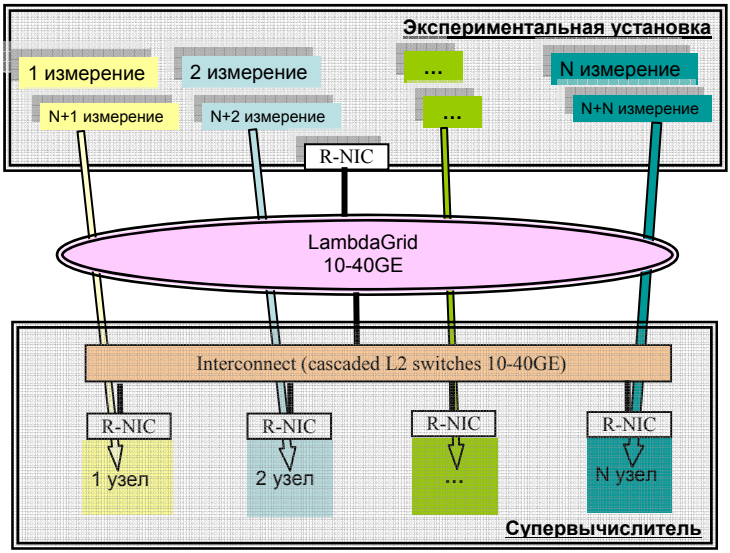

Рис. 1. Идеальная модель обработки интенсивных потоков данных

Задержка обмена данными между системами имеет два компонента: задержки в оконечных системах и задержки в сети. LambdaGrid-технологии [1] решают задачу скоростной передачи данных между взаимодействующими системами.

Задержка приема/передачи в оконечных системах определяется скоростью порта и временем поступления данных от Ethernet порта сетевого адаптера (NIC) в буфер приложения и, наоборот, из буфера приложения в сеть (порт адаптера). Скорости портов непрерывно растут (1-10-40-100GE), и основная проблема заключается в передаче данных приложению. Эта задержка возникает как на передающей, так и на приемной стороне и определяется внутренней сущностью NIC. В качестве I/O-портов связываемых по LambdaGRID оконечных систем целесообразно использование интеллектуальных NIC-карт (Intelligent Ethernet adapter), которые аппаратно поддерживают стеки протоколов передачи данных (TOE NIC - TCP Offload Engine) и технологии удаленного прямого доступа к памяти (R-NIC) для разгрузки CPU-узлов в связи с переходом на скорости 10–40–100GE.

Тестовая инфраструктура

Для разработки и тестирования протокола и программного обеспечения (ПО) на площадке ИМСС УрО РАН была создана инфраструктура (рис. 2).

Основные вычислительные узлы и управляющий узел суперкомпьютера МВС-1000/16П (16 вычислительных узлов, операционная система Linux) объединены двумя приватными подсетями 100 Мбит/с:

192.168.3.0/24 – сеть передачи команд и данных с управляющего узла на вычислительные узлы;

192.168.2.0/24 – сеть внутреннего интерконнекта суперкомпьютера для обмена данными между вычислительными узлами.

Экспериментальная установка (ЭУ) PIV (компьютер для управления камерами и лазером с программным обеспечением Actual Flow под управлением операционной системы Windows)

подключена на скорости 100 Мбит/с непосредственно к сети внутреннего интерконнекта суперкомпьютера.

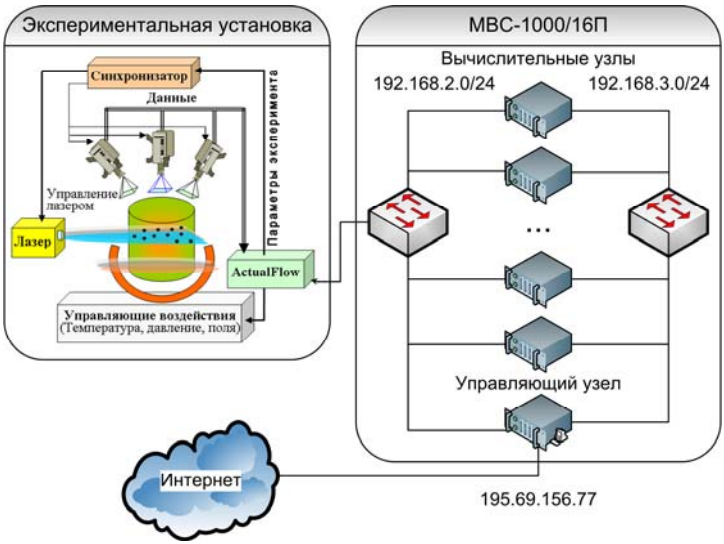

Рис. 2. Структура сети передачи данных

Потоки данных

Каждое измерение, сгенерированное ЭУ, является парой изображений, созданных через определенный интервал времени. Согласно требованиям прикладной задачи каждое измерение можно рассчитывать на суперкомпьютере независимо от других. Это позволяет отказаться от однозначного отображения измерений на вычислительные узлы.

Поскольку измерение является парой изображений, необходимо передать с ЭУ на вычислительный узел суперкомпьютера два блока бинарных данных. Результат обработки каждого измерения (скорости частиц, команды управления экспериментом) является блоком бинарных данных, которые необходимо возвращать на ЭУ. Помимо результата вычисления с вычислительных узлов кластера на ЭУ может передаваться служебная текстовая информация, например, предназначенная для записи

в журнал работы сервера на ЭУ (диагностика, тестирование, анализ производительности).

Программное обеспечение и протокол передачи данных

Разработанное ПО предназначено для обеспечения взаимодействия прикладных процессов ЭУ с вычислительными узлами суперкомпьютера. Взаимодействие осуществляется с использованием разработанного протокола передачи данных, получившего название «Протокол PIV».

Протокол PIV использует TCP, работает по схеме запрос-ответ и предназначен для передачи двух блоков бинарных данных как от сервера к клиенту, так и от клиента к серверу. Поля заголовков кодируются в сетевом порядке байт.

Формат пакета изображен на рис. 3.

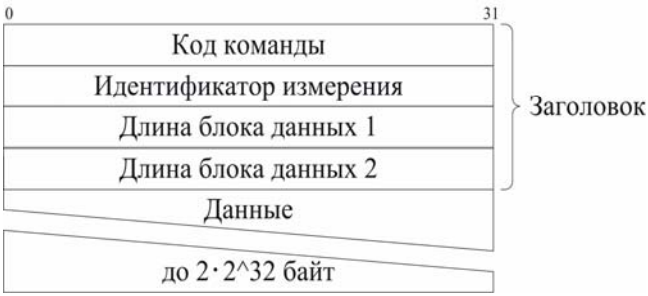

Рис. 3. Формат пакета

Поле «Код команды» содержит один из следующих кодов команд:

- Команды запросов клиента:
  - ✓ OPCODE_GET – запрос у сервера данных следующего измерения.
  - ✓ OPCODE_PUT – передача на сервер результатов обработки измерения.
- Команда ответов сервера:
  - ✓ OPCODE_ERROR – выполнение запроса закончилось ошибкой. TCP-сессия разрывается.
  - ✓ OPCODE_EOD – данных больше нет. Клиент завершает работу.

- ✓ OPCODE_GET_RESP – код ответа на запрос OPCODE_GET. В пакете с этим кодом передаются запрошенные данные.
- ✓ OPCODE_PUT_RESP – код ответа на запрос OPCODE_PUT. Подтверждение от сервера об успешном приеме данных.

Поле «Идентификатор измерения» содержит порядковый номер измерения. Поля «Длина блока данных 1» и «Длина блока данных 2» – длины в байтах соответственно первого и второго подблока данных в поле «Данные» пакета.

Диаграммы допустимых последовательностей кодов команд в пакетах (для одного цикла запрос-ответ) представлены на рис. 4.

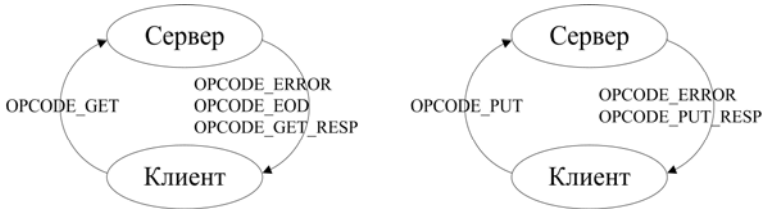

Рис. 4. Диаграммы допустимых последовательностей кодов команд в пакетах

Разработанный протокол добавляет 16 байт заголовочных данных на каждое отправляемое измерение.

Схема работы протокола

На рис. 5 представлена временная диаграмма обмена данными между вычислительным узлом и экспериментальной установкой.

В случае возникновения ошибки передачи данных сторона, обнаружившая ошибку, закрывает TCP-соединение. При закрытии соединения сервер добавляет в очередь готовых к обработке измерений те измерения, которые были переданы через закрытое TCP-соединение, но на которые не был прислан результат расчета. Это позволяет избежать случая, когда из-за

сбоя некоторые измерения окажутся не обработанными. Клиентское ПО при разрыве TCP-соединения прекращает проведение текущего расчета и после таймаута заново инициализирует соединение с серверным ПО на ЭУ.

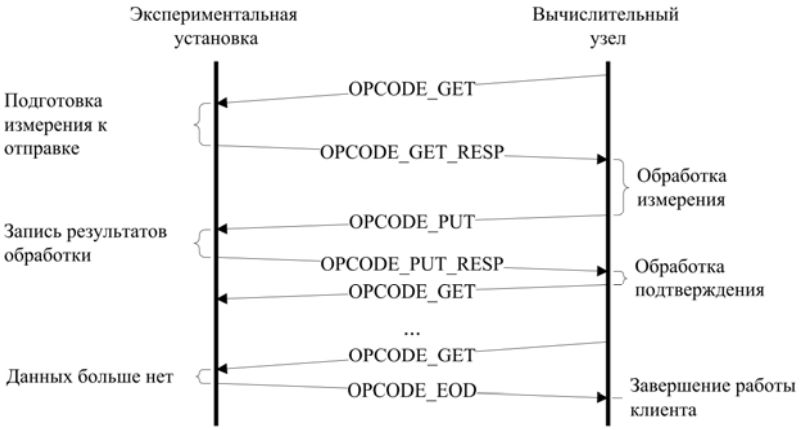

Рис. 5. Временная диаграмма протокола

Серверное программное обеспечение

Серверное ПО запускается на экспериментальной установке. Оно построено по событийно-ориентированной архитектуре. Использование событийно-ориентированной архитектуры и механизма неблокирующих сокетов позволяет в асинхронном режиме обслужить множество подключенных клиентов внутри одной нити исполнения программного кода. Положительный эффект достигается за счет фактического распараллеливания операций подготовки данных и процессов передачи данных через сеть. Для уменьшения влияния на производительность блокирующих операций чтения данных с диска сервер реализован многопоточным.

Клиентское программное обеспечение

Клиентское ПО запускается на вычислительных узлах суперкомпьютера. Оно осуществляет взаимодействие между прикладным алгоритмом, который выполняет необходимые расче-

ты, и серверным ПО. Для параллельного запуска множества клиентов на различных вычислительных узлах суперкомпьютера используется MPI.

Для реализации алгоритма прикладной задачи в код клиента необходимо встроить функцию обработки измерения, написанную на языке C. Прототип этой функции:

```
struct Response {
  char *log;    size_t log_length; /*Поле служебной информации, длина блока данных*/
  char *data; size_t data_length;/*Поле результатов расчета, длина блока данных*/ };
void processing(struct Response *res, uint32_t id, char *a, size_t a_len, char *b, size_t b_len);
```

Первым параметром передается указатель на структуру Response, через которую функция возвращает результаты расчета. Второй параметр – идентификатор измерения, последующие параметры – указатели на блоки данных и длины соответствующих блоков. При заполнении структуры Response для выделения памяти под данные log и data необходимо использовать функцию malloc. После отправки результатов на сервер выделенная память будет автоматически освобождена.

Анализ производительности разработанного решения

Исследование производительности было проведено путем измерения скорости передачи данных между экспериментальной установкой под управлением Windows и вычислительными узлами кластера МВС-1000. График зависимости скорости передачи данных от размера блока и от числа подключенных узлов приведен на рис. 6.

Низкая скорость передачи блоков данных размером 8 Кб (передаваемое измерение содержит два блока данных) показывает недостаточную эффективность стека протоколов TCP/IP в случае, когда IP-пакеты передаются частично заполненными. При размерах блоков, близких к размерам данных реальных измерений скорость достигает максимума, начиная с 2–4 потоков передачи данных.

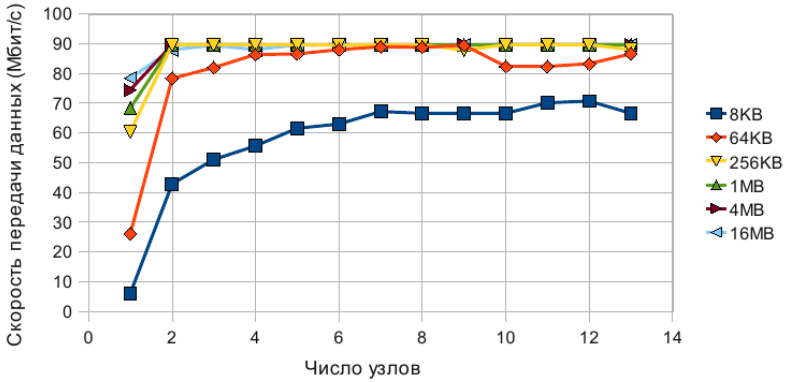

Рис. 6. Зависимость скорости передачи данных от числа подключенных узлов

Максимум скорости имеет значение порядка 90 Мбит/с, что составляет 90 % от пропускной способности канала. Для стека протоколов TCP/IP 90 % использования канала является значением, близким к предельно достижимой эффективности [5].

Спроектирован протокол и разработано программное обеспечение для передачи данных с экспериментальной установки на узлы вычислительного кластера.

Исследование производительности показало, что при передаче блоков данных размером более 256 Кбайт удалось добиться 90% использования канала передачи данных под полезную нагрузку. Результат в 90% и отсутствие падения скорости с ростом числа подключенных клиентов указывает на то, что ограничителем пропускной способности стала существующая локальная сеть. Из этого можно сделать вывод, что разработанное решение имеет запас производительности, который позволит использовать решение в сетях с большей скоростью передачи данных.

Список литературы

1. Масич А.Г., Масич Г.Ф. GIGA UrB RAS подход к LambdaGrid парадигмам вычислений // Научный сервис в сети Интернет: суперкомпьютерные центры и задачи: тр. междунар. суперкомпьютерной конф. – М.: Изд-во МГУ, 2010. (в печати)

2. Степанов Р.А., Масич А.Г., Масич Г.Ф. Инициативный проект «Распределенный PIV» // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: тр. всерос. суперкомпьютерной конф. – М.: Изд-во МГУ, 2009. – С. 360–363.

3. Инфрастуктура распределенного эксперимента / А.Г. Масич (и др) // XVI конф. представителей региональных научно-образовательных сетей «RELARN-2009»: тез. докл. – М.–СПб, 2009. – С. 58–60.

4. Adrian R.J. Scattering particle characteristics and their effect on pulsed laser measurements of fluid flow: speckle velocimentry vs. particle image velocimentry // Appl. Opt. 1984. Vol. 23. Pp. 1690-1691.

5. Dykstra P. Protocol Overhead. – URL: http://sd.wareonearth.com/~phil/net/overhead/.

[1]В.П.Матвеенко, [1]Р.А. Степанов, [1]Б.И. Мызникова, [1]И.Э. Келлер, [2]М.Г. Бояршинов

[1] Институт механики сплошных сред УрО РАН, г. Пермь
[2] Пермский государственный технический университет

ОПЫТ ОРГАНИЗАЦИИ МАГИСТРАТУРЫ ПО ВЫСОКОЭФФЕКТИВНЫМ ВЫЧИСЛИТЕЛЬНЫМ ТЕХНОЛОГИЯМ В МЕХАНИКЕ В ПЕРМСКОМ ГОСУДАРСТВЕННОМ ТЕХНИЧЕСКОМ УНИВЕРСИТЕТЕ

Пермский край известен высокоразвитыми горнодобывающим, химическим, металлургическим, строительным комплексами, машино- и приборостроением. На предприятиях края осуществляется разработка и производство наукоемкой продукции, в том числе авиационных и ракетных двигателей, газотурбинных установок, артиллерийских систем, оборудования для добычи нефти, оптоволоконных гироскопов, новых материалов на основе порошковых нанокомпозиций. Естественно, что в пермских инженерных и научных коллективах возникают задачи обработки информации, математического моделирования, проектирования и оптимизации