

ОПТИМИЗАЦИЯ ТРАНСПОРТНЫХ СРЕДСТВ ДЛЯ КЛАСТЕРНЫХ АРХИТЕКТУР

С. А. Бокач

ВВЕДЕНИЕ

Технология распределенной обработки является эффективным средством для решения задач с большим объемом вычислений. Для организации подобной обработки могут быть применены кластерные архитектуры. Под термином кластер понимается некоторая сеть из набора (N) узлов системы, содержащих 1 или более процессорных элементов, и фабрики межсоединений (некоторого способа соединения узлов) [1]. Наиболее часто используемые технологии это Fast Ethernet, Gigabit Ethernet, SCI, Myrinet, Quadrics, Infiniband. В системах, основанных на технологиях SCI, Myrinet, Infiniband достигается очень маленькая латентность, однако такие системы очень дороги. При использовании же намного более дешевого оборудования Fast и Gigabit Ethernet (т.е. фактически получаем локальную сеть) передача данных происходит с использованием стека протоколов TCP/IP. Существующий же стек протоколов не позволяет эффективно использовать канал передачи данных. Целью работы было построить систему, позволяющую эффективно передавать данные, минуя стек протоколов TCP/IP.

Существует целый ряд коммуникационных средств с высокими параметрами производительности (Active Messages[2], Fast Messages [3], Unet [4], Gamma, VIP). Однако наряду с хорошими показателями передачи данных у них существует и ряд недостатков: неэффективное использование буфера, наличие задержек при проверке поступивших сообщений (Active Messages).

1 МОДЕЛЬ СИСТЕМЫ БЫСТРОЙ ПЕРЕДАЧИ ДАННЫХ

Моделью системы является ориентированный граф сети, вершины которого представляют собой процессы. Причем необязательно чтобы различные процессы принадлежали различным узлам локальной сети. Дуги графа – это однонаправленные каналы связи между парой процессов. Все информационное общение с другими узлами осуществляется с использованием протокола TCP/IP через стандартный интерфейс сокетов (socket).

Для однозначной идентификации каждому процессу узла сети присваивается набор значений {id, mac, ip} – где: Id – уникальный идентификатор узла в сети, вырабатывается в процессе инициализации службы; Mac – MAC-адрес сетевого адаптера, через который будут передаваться

данные; Ip – IP–адрес сетевого адаптера, используется для обновления информации о доступных узлах.

В качестве основы для службы поддержки передачи данных была взята модель Direct Point – технологии, передача данных между процессами происходит по разработанному протоколу BFP [5].

2 ПРОГРАММНАЯ ПОДДЕРЖКА СИСТЕМЫ

2.1 Сервисы сети

На каждом узле системы хранятся таблицы – таблица процессов (представляет собой совокупность троек {id, mac, ip}); таблица связей (поясняет, как связаны между собой процессы); переменная, хранящая информацию о количестве обновлений локальных таблиц. Причем на всех узлах виртуальной сети хранятся одинаковые копии таблиц. Для обеспечения работы системы на каждом узле виртуальной сети устанавливается Win32-служба, контролирующая добавление/удаление процессов из системы.

Описание жизненного цикла службы:

- Процесс инициализации: на данном этапе необходимо построить граф сети на данном узле – достигается путем рассылки широко-вещательного запроса «Get Table». Каждый узел высылает часть данных графе. Построив таблицы, служба создает проецируемый в память файл (ресурсы для которого выделяются из страничного файла, что позволяет избежать многих задержек) и помещает в него данные о системе.
- Обслуживание клиентов: при поступлении запроса сервис организывает поток для данного клиента и выполняет соответствующую обработку. Поток действителен на время сеанса общения с данным клиентом. При обновлении состояния системы служба обновляет информационный файл.
- Процесс удаления службы: удалить все процессы данного узла из графа, сообщить об изменениях, освободить файл ресурсов.

2.2 Протокол BPF

Передача данных между процессами осуществляется через протокол BFP (beta fast protocol). Это протокол передачи данных вне стека протоколов TCP/IP, взаимодействующий напрямую с сетевым адаптером с помощью API библиотеки WinPCap [6].

Протокол разрабатывался с учетом следующих требований:

- Гарантированная доставка данных (возможность ретранслирования утерянных либо поврежденных пакетов);

- Ориентирование на соединение. В общем случае соединение остается открытым для передачи данных;
- Симплексная передача данных.

2.2 Интерфейс пользователя

Взаимодействие с клиентской частью (обмен данными о доступных соединениях) осуществляется посредством файлов, проецируемых в память. Перед началом работы пользователь должен вызвать инициализирующую функцию. При этом процесс делает запрос локальной службе о создании новой строки в таблицах, инициализируется структура дескриптора захвата пакетов (вызывается функция `init()` протокола BFP). По завершении работы необходимо выполнить очистку ресурсов, при этом удаляется информация о данном процессе из всех таблиц системы, освобождаются занятые ресурсы. Ниже представлены основные функции системы.

Таблица

Программная реализация (API системы)

Основные функции библиотеки, доступные конечному пользователю	
<code>int init(int adapterNum)</code>	инициализация работы системы
<code>int SendData(int id, char Buffer[], int Length, bool checkSend)</code>	передача данных по сети
<code>int ReceiveData(int id, char *Buffer, bool TimeOutExit, bool checkReceive)</code>	получение данных от процесса
<code>int GetMyId()</code>	получение идентификатор текущего процесса
<code>int GetAllAvailibleProcesses (int processes[], int maxCount)</code>	получить список всех доступных процессов

3. ТЕСТИРОВАНИЕ

Тестирование системы проводилось в локальной сети Ethernet, 100 Мбит/с. Сетевые карты: Realtek Family PCI Fast Ethernet. (Sender/ Receivers – Pentium III 1,0 Ghz, 512 RAM, Win NT).

Корректность передачи проверялась после полного приема данных. В ходе экспериментов получили, что при базовой скорости локальной сети Ethernet 100 Мбит/с скорость передачи данных с использованием системы составила 87-92 Мбит/с, в то время как при использовании стека протоколов TCP-IP данные показатели не превышают 55 Мбит/с.

В результате тестов получили, что построенная система по производительности превышает большинство систем, и примерно совпадает с наилучшими показателями систем U-net, DP. Загрузка канала передачи составляет примерно 90 процентов, в то время как при использовании

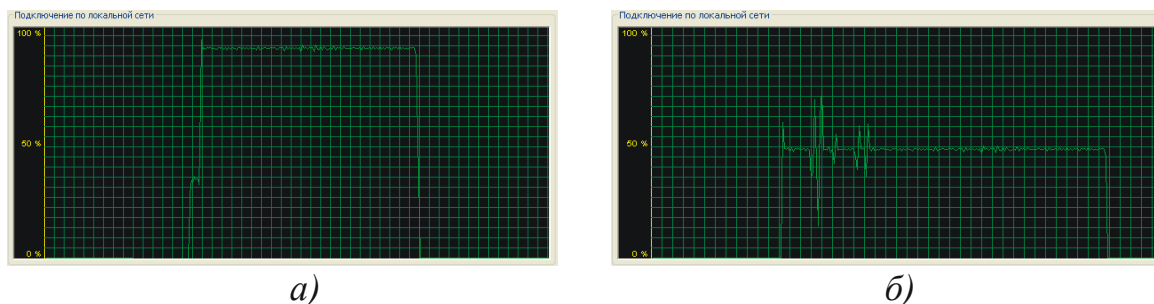


Рис. Загрузка канала передачи данных при использовании:
а) системы передачи данных;
б) стека протоколов ТСР/ІР

стека протоколов ТСР/ІР – не превышает 50 процентов, как показано на рисунке.

Заключение

В работе предложена модель организации обмена данными между процессами и протокол передачи данных, взаимодействующий непосредственно с драйвером сетевого адаптера. Реализация ориентированна на платформу Windows.

Эффективность достигается за счет взаимодействия непосредственно с сетевым оборудованием, расположения буферов приема и передачи в адресном пространстве процесса.

Литература

1. Буза М. К., Зимянин Л. Ф. Модель общей памяти для кластерных архитектур // Международная научн. конф. «Суперкомпьютерные системы и их применение» SSA'2004: Доклады конф. – Мн.: ОИПИ, 2004. – С.77-80.
2. T.von Eicken et al. Active Messages: A Mechanism for Integrated Communication and Computation, Proc. 19th Symp. Computer Architecture, IEEE CS Press, Los Alamitos, Calif., 1992, pp.256-266.
3. Pakin S., Lauria M., and Chien A. High Performance Messaging on Workstations: Illinois Fast Messages (FM) for Myrinet, Proc. Supercomputing'95, ACM Press, New York, 1995.
4. T.von Eicken et al. U-Net: A User-level Network Interface for Parallel and Distributed Computing, Proc.15th Symp. Operating Systems Principles, ACM Press, New York, 1995, pp.40-53.
5. Chun-Ming Lee, Anthony Tam, Cho-Li Wang. Directed Point: An Efficient Communication Subsystem for Cluster Computing - The University of Hong Kong, available at <http://www.srg.cs.hku.hk/homepage/srg/dp.htm>
6. WinPCAP web site, Politecnico di Torino, available at <http://winpcap.polito.it>