

ПЕРЕЛІК ПОСИЛАНЬ

1. Динамика одномерных отображений / А. Н. Шарковский, С. Ф. Коляда, А. Г. Сивак, В. В. Федоренко. – Киев : Наук. думка, 1989. – 216 с.
2. Заяць В. М. Построение и анализ модели дискретной колебательной системы / Заяць В. М. // Кибернетика и системный анализ. – 2000. – С. 161–165.
3. Заець В. М. Моделі дискретних коливних систем / Заець В. М. // Комп'ютерні технології друкарства. – 1998. – С. 37–38.
4. Заець В. М. Аналіз динаміки та умов стійкості дискретних моделей коливних систем / Заець В. М. // Вісник НУ «Львівська політехніка» «Інформаційні технології та мережі». – 2004. – № 519. – С. 132–142.
5. Шустер Г. Детерминированный хаос : Введение : пер. с англ. / Шустер Г. – М. : Мир, 1988. – 240 с.
6. Zayats V. Chaos searching algorithm for second order oscillatory system / Zayats V. // Proc. International Conf. «TCSET-2002». – Lviv – Slavsk, 2002. – P. 97–98.
7. Андронов А. А. Теория колебаний / Андронов А. А., Вит А. А., Хайкин С. Е. – М. : Наука, 1981. – 400 с.
8. Бутенин Н. В. Введение в теорию нелинейных колебаний / Бутенин Н. В., Неймарк Ю. И., Фужаев Н. А. – М. : Наука, 1976. – 354 с.
9. Ван-дер-Поль. Нелинейная теория электрических цепей / Ван-дер-Поль. – М. : Связь, 1935. – 186 с.
10. Видаль П. Нелинейные импульсные системы / Видаль П. – М. : Энергия, 1974. – 336 с.
11. Харкевич А. А. Опознание образов / Харкевич А. А. // Радиотехника. – 1959. – Том 14. – С. 15–19.
12. Фукунага К. Введение в статистическую теорию распознавания / Фукунага К. – М. : Наука, 1979. – 512 с.
13. Горелик А. Л. Методы распознавания / Горелик А. Л., Скрипник В. А. – М. : Высшая школа, 1989. – 232 с.
14. Дуда Р. Распознавание образов и анализ сцен / Дуда Р., Харт П. – М. : Мир, 1976. – 512 с.
15. Заяць В. М. Алгоритмічне та програмне забезпечення системи розпізнавання людини за її рукомоторними реакціями / Заяць В. М., Уліцький О. О. // Вісник ДУ «Львівська політехніка» «Комп'ютерна інженерія та інформаційні технології». – 2000. – № 392. – С. 73–76.
16. Заяць В. М. Визначення пріоритету детермінованих ознак при побудові системи розпізнавання об'єктів / Заяць В. М., Шокира О. // Зб. праць науково-практичної конф. ЛДІНТУ імені В. Чорновола «Математичне моделювання складних систем». – Львів, 2007. – С. 135–137.
17. Заяць В. М. Підхід до опису системи розпізнавання користувача комп'ютера / Заяць В. М. // Комп'ютерні технології друкарства. – 2006. – С. 46–53.
18. Заяць В. М. Математичний опис системи розпізнавання користувача комп'ютера / Заяць В. М., Заець М. М. // Фізико-математичне моделювання та інформаційні технології : зб. – Львів, 2005. – Вип. 1. – С. 146–152.
19. Березин И. С. Методы вычислений / Березин И. С., Жидков Н. П. – М. : Физматиздат, 1962. – 639 с.
20. Заець В. М. Приведення неперервної автоколивної системи до дискретної моделі та спрощення її аналізу / Заець В. М. // Відбір і обробка інформації. – 2005. – Вип. 23 (99). – С. 35–39.
21. Алексеев А. Алгоритм розпізнавання символів на основі структурного підходу / Алексеев А., Заяць В., Іванов Д. // Вісник НУ «Львівська політехніка» «Комп'ютерна інженерія та інформаційні технології». – 2002. – № 468. – С. 129–133.
22. Заяць В. М. Проект системи розпізнавання рукописного тексту / Заяць В. М., Іванов Д. О. // Вісник НУ «Львівська політехніка» «Комп'ютерна інженерія та інформаційні технології». – Львів, 2003. – № 481. – С. 78–83.

Надійшла 6.10.2008
Після доробки 2.12.2008

В статтю пропонується додати к встановленню пріоритету первичних признаков при построении системы распознавания пользователя компьютера при описании ее в виде дискретной модели путем определения рукомоторных реакций пользователя при введении информации с клавиатуры компьютера. Целесообразность подхода иллюстрируется при реализации процедуры идентификации пользователя компьютера при использовании детерминированного и вероятностного подходов.

In the paper establishment of the priority of primary features in construction of recognition system of computer user while describing it as discrete model by means of manual motor reaction detection by the user while entering in-formation from a key board was offered. Appropriateness method is illustrated in the process of user identification procedure by means of using implementing and probability approaches.

УДК 004.94

И. В. Корольков

ОСОБЕННОСТИ РЕАЛИЗАЦИИ СИСТЕМЫ ПАРАЛЛЕЛЬНОГО МОДЕЛИРОВАНИЯ МАРШРУТИЗАЦИИ В БОЛЬШИХ ВЫЧИСЛИТЕЛЬНЫХ СЕТЯХ

Рассмотрена система параллельного моделирования больших вычислительных сетей. Предложенная архитектура нацелена на моделирование больших сетей за счет параллельного выполнения и эффективного использования памяти. Система использует гибридный метод синхронизации на основе окон и нулевых сообщений и позволяет достичь ускорения выполнения, близкого к линейному. Модульный принцип системы позволяет легко добавлять новые сетевые устройства и протоколы.

© Корольков И. В., 2009

Предложены модули моделирования маршрутизаторов, каналов связи и источников трафика.

ВВЕДЕНИЕ

С ростом Интернет и широким внедрением IP-сетей интерес к крупномасштабному сетевому модели-

рованию значительно вырос. По мере увеличения размера сети и расширения круга использующих ее приложений начинают проявляться явления не обнаруженные ранее, например: штормы перегрузок, осцилляции маршрутов и распространение вирусов, которые требуют дополнительного изучения. Важной составной частью моделирования вычислительных сетей является моделирование протоколов маршрутизации, которое является одним из основных инструментов изучения сетей.

Известно, что сложность моделирования сетей возрастает вместе с их размером. В условиях существующих на данный момент аппаратных ресурсов моделирование традиционными методами сетей масштаба Интернет является невозможным. Сложность лежащих в их основе протоколов затрудняет разработку аналитических моделей.

ПОСТАНОВКА ЗАДАЧИ

Одним из подходов к моделированию вычислительных сетей является использование функционала дискретно-событийного моделирования [1]. Этот метод моделирования обеспечивает возможность реализовать практически неограниченный уровень детализации модели, однако приводит к значительному увеличению ее вычислительной сложности. Для решения задач моделирования больших сетей часто обращаются к параллельному дискретно-событийному моделированию [2].

Существует ряд проблем реализации системы моделирования, связанных с обеспечением механизмов синхронизации и распределением вычислительной нагрузки. При выполнении моделирования должны быть сбалансированы следующие характеристики: скорость, точность и масштабируемость.

Целью исследования является создание и испытание системы параллельного моделирования маршрутизации в больших вычислительных сетях – Parallel Network Simulation System (PNSS).

1 МЕХАНИЗМЫ СИНХРОНИЗАЦИИ ПАРАЛЛЕЛЬНОГО МОДЕЛИРОВАНИЯ

Все механизмы синхронизации могут быть условно разделены на два класса: оптимистические и консервативные [3]. Консервативная синхронизация – механизм, не позволяющий процессам, участвующим в распределенном моделировании, обрабатывать те сообщения, временные метки которых не соответствуют хронологическому порядку. В системах моделирования с консервативной синхронизацией обрабатываются только те события, о которых можно с уверенностью сказать, что в будущем они не получат сообщения с меньшим временем. Оптимистическая син-

хронизация, в отличие от консервативного механизма, позволяет обрабатывать любые события, но при обнаружении нарушения причинной связи восстанавливает одно из предыдущих состояний системы при помощи откатов.

При моделировании больших сетей необходимость постоянного сохранения состояний связана с чрезмерной ресурсоемкостью, особенно из-за того, что вслед за одним откатом к более раннему модельному времени на одном процессоре может последовать каскад откатов на других процессорах. В связи с этим в системе моделирования PNSS применена консервативная синхронизация на основе гибридного метода, использующего нулевые сообщения и окна.

В отличие от классической схемы синхронизации на основе окон, в предложенном методе окно не пересчитывается глобально в конце каждого периода. Вместо этого вначале каждая пара процессоров i и j , производящая соответственно моделирование узлов n_i и n_j , на основании типа моделируемой линии связи определяет окно δ_{ij} . Процессоры i и j должны производить синхронизацию в моменты модельного времени $n\delta_{ij}$, $n > 0$. Требование глобальной минимизации δ_{ij} устраняется, однако процессоры i и j , моделирующие соответственно связанные узлы n_i и n_j , должны производить синхронизацию, даже если на протяжении длительных периодов времени данные между узлами не передаются. На практике в условиях моделирования больших сетей такие пустые синхронизации, как правило, не происходят, особенно если синхронизации агрегируются при наличии нескольких линий связи между узлами.

2 ОБЩИЕ ПРИНЦИПЫ ПОСТРОЕНИЯ СИСТЕМЫ МОДЕЛИРОВАНИЯ

Архитектура PNSS ориентирована на моделирование сетей большого размера, что обуславливает особые требования к масштабируемости и производительности. При этом модульный принцип построения системы обеспечивает значительную гибкость.

Использование метода алгоритмической маршрутизации устраняет необходимость в поддержании таблиц маршрутизации для каждого узла. При этом для работы с моделями большого размера PNSS ориентирована на функционирование в условиях параллельной вычислительной среды с распределенной памятью.

Ядро системы является полностью последовательным. Список событий на каждом процессоре последователен, а за распараллеливание отвечают отдельные модули. Если два связанных узла находятся в одном участке, т. е. моделируются на одном процессоре, то они соединяются при помощи непосредственной связи (НС). Если узлы находятся в отдельных участках, то они соединяются с помощью внешней связи (ВС), которые находятся в обоих из участков. Время от

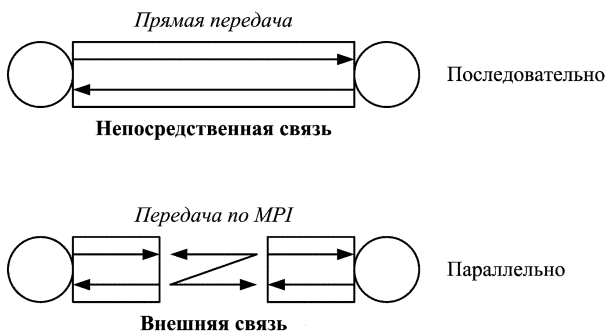


Рисунок 1 – Принцип функционирования НС и ВС

времени две ВС производят соединения для обмена пакетами. Интервал соединения имеет продолжительность, соизмеримую с временем задержки НС. Фактически ВС имеет те же свойства, что и НС, за исключением того, что пакеты принимаются или передаются за пределы участка, а не внутри него (рис. 1).

Снижение коммуникационных накладных затрат возможно обеспечить за счет преобразования в ВС длинных линий связи, задержки которых и так велики. Конечно, ограничения топологии сети могут диктовать необходимость использования в качестве ВС линий с малыми задержками.

Ядро системы моделирования включает набор средств построения компонентов сети и обработчик дискретных событий. Интерфейс обмена данными сетевыми компонентами описывается пятью основными структурами:

- класс – данная структура содержит указатели на функции инициализации подклассов и устройств. Примерами сетевых классов являются маршрутизаторы, каналы связи и источники трафика. Для каждого модуля на этапе загрузки инициализируется и существует один экземпляр данной структуры;

- подкласс – представляет собой специализацию класса. Так, например, подклассом модуля TCP может быть сервер, клиент или клиент с различной частотой запросов к другому клиенту;

- устройство – представляет собой конкретную реализацию подкласса. Устройство обеспечивает интерфейс для связи с другими совместимыми устройствами и может иметь пространство памяти для сохранения своего состояния по мере выполнения моделирования;

- точка_входа – точка входа хранит указатель на функцию обработчика, вызываемую каждый раз, когда пакет поступает на точку входа. Эта функция обеспечивает выполнение соответствующих для данного модуля действий. Устройство может иметь одну или несколько точек входа;

- точка_выхода – устройство может иметь одну или несколько точек выхода, содержащих указатели на точки входа других устройств.

Модуль сетевого компонента заполняет элементы этих структур соответствующими значениями и указателями функций. Сетевые компоненты обмениваются друг с другом данными исключительно через этот API.

События могут относиться к конкретному сетевому компоненту или компонентами или быть глобальными. Обработчик событий процессора сохраняет события в единой упорядоченной по времени очереди представляющей собой скошенное дерево [4, 5]. При этом, сам обработчик не производит синхронизации между процессорами – все необходимые действия по синхронизации выполняются специализированными модулями. Эти модули обеспечивают обмен пакетами, пересекающими внешние связи и блокировку обработки процессором событий для сохранения причинных связей.

Для генерации трафика предлагаются два модуля. Первый модуль представляет собой приемник пакетов и источник, генерирующий последовательность пакетов с заданным временем межпакетных интервалов. Такой модуль может быть рассмотрен как простой источник UDP-трафика. Второй модуль представляет собой реализацию протокола TCP с некоторыми упрощениями. Например, каждый узел может быть или клиентом или сервером, но не одновременно. Тем не менее, модуль отвечает большинству спецификаций RFC 793 [6], и включает такие дополнения, как медленный старт и предотвращение перегрузок [7].

Модуль непосредственной связи отображает двуправленную линию связи типа «точка – точка». Модуль буферизует пакеты, помещает их в канал связи и спустя некоторый промежуток времени, соответствующий вносимой каналом задержке, посылает их адресату.

Модуль внешней связи является ключевым компонентом, обеспечивающим возможность параллельного моделирования в системе PNSS. Два устройства внешней связи (по одному для каждого конца линии связи) заменяют обычную линию связи, когда два соединенных узла моделируются на разных процессорах. Для пакета такое устройство представляется идентичным устройству непосредственной связи.

Как только пакет достигает устройства внешней связи, должно быть обеспечено, чтобы удаленный процессор получил пакет во время или до того, как пакет будет доставлен к удаленному узлу. Если время задержка составляет Δ секунд, тогда два устройства должны производить синхронизацию как минимум каждые Δ секунд модельного времени. Все пакеты, которые поступают и обрабатываются в период времени от t до $t + \Delta$, сохраняются, и в момент времени $t + \Delta$ передаются соответствующему устройству на удаленном процессоре. Этот процессор приостанавливает моделирование в момент времени $t + \Delta$, пока пакеты не будут переданы, после чего процесс моделирования возобновляется. Поскольку рассматри-

вается дуплексная линия связи, локальный процессор может получить пакеты от удаленного процессора. Такие пакеты имеют время прибытия от $t + \Delta$ до $t + 2\Delta$. Пакеты, подлежащие передаче от одного процессора к другому, назовем межпроцессорными пакетами.

В описанной процедуре существуют два проблемных вопроса: во-первых, каждая связь между процессорами, представляется двумя устройствами типа «мост», что вводит дополнительную вычислительную нагрузку; а во-вторых, необходимость синхронизации обуславливает жесткий параллелизм системы.

Первая проблема может быть решена путем упорядочения данных. Допустим, процессор моделирует нескольких узлов, имеющих связи с узлами, моделируемыми на других процессорах. Если эти линии связи однотипны и имеют одинаковое время задержки, тогда пакеты, подлежащие передаче, могут быть объединены для отправки в одно сообщение. Такой подход сокращает количество межпроцессорных сеансов связи и соответственно снижает вычислительные затраты.

Применение консервативного подхода к синхронизации неизбежно приводит к необходимости соблюдения жесткого параллелизма между процессорами. Пусть $P_i(n)$ – время необходимое процессору i для моделирования временного интервала от $(n - 1)\Delta$ до $n\Delta$; при этом $N\Delta$ – общее время моделирования. Полное время выполнения моделирования будет не менее

$$T = \max \left(\sum_{n=1}^N P_i(n) \right).$$

Другими словами, в лучшем случае можно надеяться, что полное время моделирования будет соответствовать времени затраченному процессором, выполняющим наибольший объем вычислений, не учитывая затраты на межпроцессорные пересылки. С учетом жесткого параллелизма, полное время выполнения моделирования может быть

$$T = \sum_{n=1}^N \max(P_i(n)),$$

также не считая межпроцессорных пересылок. Время выполнения моделирования возрастает, если на отдельных процессорах присутствует значительные вариации $P_i(n)$. Для избежания наибольших проблем консервативной синхронизации были приняты следующие меры.

Во-первых, везде, где это возможно, используются асинхронные коммуникационные интерфейсы, обеспечиваемые средствами MPI. Это позволяет отдельному процессору обмениваться данными с несколькими удаленными процессорами одновременно.

Описанный выше исходный механизм синхронизации предполагает, что все процессоры обмениваются межпроцессорными пакетами с периодами $n\Delta$, где $n \in [1 \dots N]$. В эти моменты времени процессор должен собрать все межпроцессорные пакеты, отправить их удаленным процессорам и дождаться получения пакетов от них. Если один процессор не успевает достигнуть точки синхронизации (что возможно из-за большого объема трафика в моделируемом участке сети), все остальные процессоры вынуждены ожидать его.

Вторая мера нацелена на устранение данной проблемы, и предполагает упреждающую передачу межпроцессорных пакетов. Каждый период синхронизации Δ разделен на s временных сегментов. Теперь процессоры обмениваются данными каждые Δ/s секунд модельного времени. При этом межпроцессорные пакеты, посланные в момент модельного времени $(n + k/s)\Delta$, $0 \leq k < s$, могут ожидать обработки удаленным процессором до времени $(n + (s - 1 + k)/s)\Delta$.

Комбинация асинхронной передачи и сегментации передачи межпроцессорных пакетов снижает строгость синхронизации. Если один процессор столкнулся с внезапным переизбытком вычислительной нагрузки, другие процессоры могут продолжить моделирование впереди отстающего процессора вплоть до $((s - 1)/s)\Delta$ секунд модельного времени. Такое смягчение вносит гибкость в процесс параллельного моделирования. Тем не менее, чрезмерное увеличение числа сегментов ведет к увеличению вычислительных затрат на обеспечение пересылок.

Эксперимент по проверке влияния количества сегментов на время выполнения моделирования показал, что наименьшее время выполнения моделирования удалось достичь при разделении периода синхронизации на два временных сегмента. При большем количестве сегментов время выполнения неуклонно растет вместе с коммуникационными накладными затратами. Использование двух сегментов позволяет достичь небольшого, но достойного внимания преимущества при моделировании в условиях интенсивного общего трафика. Преимущества сегментации периодов синхронизации варьируются между различными вариантами топологий сетей.

Модуль маршрутизатора реализует устройство, осуществляющее алгоритмическую маршрутизацию [8]. Это устройство принимает пакеты на входном порту, определяет следующую пересылку, по которой следует отправить пакет, и назначает пакет соответствующей выходной линии связи.

Формирование большой топологии сети вручную представляет собой медленный и утомительный процесс. Модуль устройства формирования топологии (УФТ) автоматизирует эту задачу. УФТ является виртуальным устройством и не отображает часть физической структуры сети и не генерирует трафик.

Вместо этого УФТ создает и соединяет маршрутизаторы, линии связи и источники трафика. Например, УФТ может считывать карту, сформированную при помощи GT-ITM [9] и Inet [10]. Inet генерирует топологию, сходную с топологией автономных систем в сети Интернет.

Главная особенность предложенного подхода состоит в том, что генерируемое устройство само может иметь тип УФТ. Так, например, первичное УФТ может использовать файл Inet для формирования топологии на уровне автономных систем. Каждая вершина в этой топологии отображается формированием узла. Таким образом, если этот узел представляет собой УФТ, формирующее типовую топологию автономной системы, можно легко перейти к формированию двухуровневой иерархической модели сети, подобной Интернет. Узлы второго уровня УФТ могут быть маршрутизаторами или другими УФТ формирующими третий уровень иерархии.

УФТ может производить разделение топологии на участки при помощи библиотеки разделения графов METIS [11]. METIS пытается сформировать сбалансированные участки с минимальным разрывом ребер. Как только топология разделена на участки, производится инициализация узлов, линий связи, мостов и источников трафика и назначение их обработки конкретным процессорам.

3 СРЕДСТВА АВТОНОМНОЙ БАЛАНСИРОВКИ НАГРУЗКИ

Важно, чтобы вычислительная нагрузка распределялась равномерно между процессорами, иначе модуль ВС на одном процессоре будет вынужден ожидать, пока соответствующий модуль на втором процессоре будет готов к обмену данными. Система моделирования PNSS поддерживает возможность балансировки нагрузки в автономном режиме. В процессе моделирования ядро системы фиксирует информацию о вычислительных затратах на моделирование каждого узла сети. В последующих сеансах моделирования эта информация может быть использована для более равномерного распределения вычислительной нагрузки.

Библиотека METIS позволяет производить взвешенное разделение графа. Такая возможность позволяет реализовать автономную балансировку нагрузки для сетей созданных с помощью УФТ. После завершения пробного сеанса моделирования фиксируется количество событий, произошедших на каждом из узлов УФТ. Эти значения сохраняются и используются в качестве весовых характеристик для разделения графа для последующих сеансов моделирования. Если число пакетов, назначенных на обработку разным процессорам, сильно отличается, взвешенное разделение топологии может обеспечить более

равномерное распределение вычислительной нагрузки, и, следовательно, более эффективное параллельное выполнение.

Степень того, насколько балансировка нагрузки может сократить время выполнения, зависит от характеристик сети. Обычная сеть, в которой пакетные события равномерно распределены между узлами, будет сбалансированной сразу после разделения с помощью METIS без использования весовых коэффициентов. При этом, если достаточно большая часть общего числа пакетных событий концентрируется на небольшом участке сети, METIS мало что может сделать для равномерного разделения. Следует заметить, что чем больше отношение количества узлов к числу процессоров, тем лучших результатов балансировки возможно достичь.

Для проверки эффективности балансировки было проведено моделирование 100 разных сетей и проведено сравнение времени выполнения с использованием и без использования балансировки. Каждая из рассмотренных сетей содержала по 1000 УФТ, каждое из УФТ формировало подсеть из 10 маршрутизаторов. К каждому из маршрутизаторов было подключено по 5 серверов, а также по 20 серверов – в точке подключения к более высокому уровню. Кроме того, к каждому маршрутизатору подключено некоторое количество клиентов, определенное случайно в соответствии с распределением Парето. Это распределение было выбрано в соответствии с исследованием топологии сети Интернет, показавшем, что распределение полустепени исхода подчиняется экспоненциальному закону [12, 13]. В среднем каждая сеть содержит 121801 клиента, или более 10 на каждый маршрутизатор. Каждый клиент инициирует соединения с сервером, при этом вероятность, что сервер находится в той же подсети, составляет 40 %. Период простоя подчиняется экспоненциальному распределению со средним значением 0,5 секунды. Сервер в среднем посылает клиенту 1001 пакет – этот размер также подчиняется распределению Парето.

Клиенты подключены к маршрутизаторам линиями связи со скоростью 10 Мбит/с, серверы – 100 Мбит/с. Маршрутизаторы в пределах одной подсети соединены линиями связи со скоростью 100 Мбит/с, подсети соединены с помощью OC-48 (2488 Мбит/с). Задержка на линиях OC-48 составляет 5 мс. Для предотвращения возникновения вариаций маршрутов между сеансами моделирования сформированные сети не содержали петель. Каждый сеанс моделирования включал 30 секунд модельного времени.

Моделирование выполнялось на вычислительном кластере под управлением ParallelKnoppix 2.7.1, состоящем из восьми узлов, каждый из которых включал процессор AMD Athlon64 X2 4800+ и 1 Гб оперативной памяти. Узлы были связаны линиями связи Fast Ethernet со скоростью 100 Мбит/с.

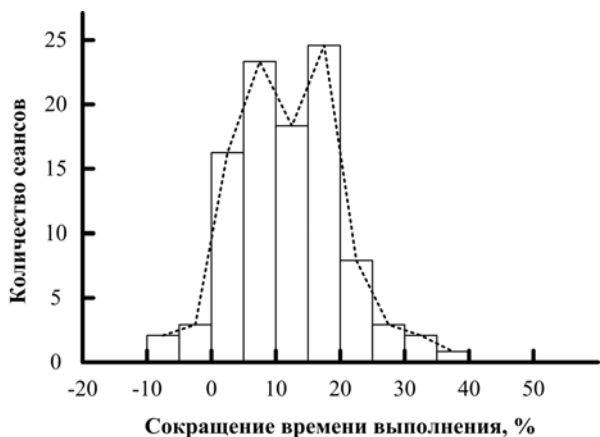


Рисунок 2 – Сокращение времени выполнения моделирования при использовании автономной балансировки нагрузки

Сокращение времени выполнения моделирования при автономной балансировке нагрузки показано на рис. 2. Среднее сокращение составляет $12,1 \pm 0,9$ %, а в некоторых случаях – до 35 %. Тем не менее, в 5 из 100 опытов время выполнения незначительно увеличивалось в условиях, когда исходная сеть была изначально хорошо сбалансирована.

Одним из показателей эффективности параллельного выполнения является разбалансировка обработки:

$$I = \max_j \frac{p_j}{\frac{1}{M} \sum_{i=0}^M p_i},$$

где p_j – количество событий, смоделированных процессором j при M -процессорном параллельном моделировании. Чем выше разбалансировка работы, тем ниже параллельное ускорение.

На рис. 3 показано, что в невзвешенных участках присутствует разбалансировка обработки в пределах от 1,2 до почти 1,8. Значения равномерно распределены до 1,4, значения выше этой границы наблюдаются реже. Практически во всех случаях разбалансировка обработки снижается после взвешенного разделения на участки. Значения практически всегда сгруппированы в интервале 1,0–1,3. Средняя разбалансировка работы при невзвешенных участках составляет $1,38 \pm 0,01$, при взвешенном разделении – $1,13 \pm 0,01$.

4 ИССЛЕДОВАНИЕ ЭФФЕКТИВНОСТЬ ПАРАЛЛЕЛЬНОЙ ОБРАБОТКИ

Параллельное ускорение – показатель эффективности распараллеливания вычислений. Обозначим па-

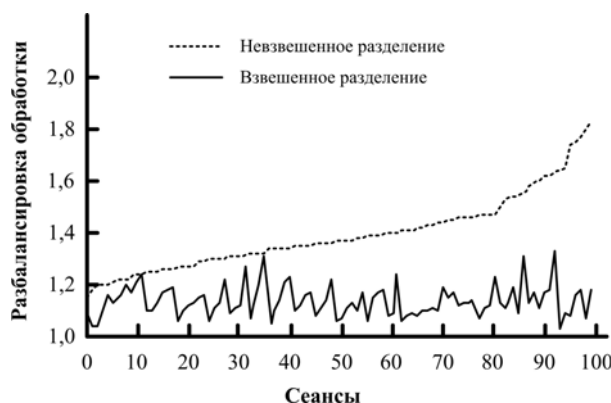


Рисунок 3 – Сокращение разбалансировки обработки при использовании взвешенного разделения сети на участки

раллельное ускорение при M -процессорном моделировании как $S(M)$:

$$S(M) = \frac{T_{\text{seq}}}{T(M)},$$

где $T(M)$ – время выполнения моделирования на M процессорах, а T_{seq} – время однопроцессорного последовательного моделирования. Для некоторых больших моделей выполнение моделирования на одном процессоре является невозможным. В таком случае используется аппроксимация $T_{\text{seq}} = mT(m)$, где m – наименьшее число процессоров, на котором возможно выполнение моделирования. Для многих сравнений в качестве базового используется линейное ускорение, при котором $S(M) = M$.

Для определения эффективности распараллеливания была проведена серия измерений на основе описанной ранее сети. Были проведены сеансы моделирования в системах, включающих от одного до восьми процессоров. На каждой конфигурации было проведено по 10 сеансов моделирования. На рис. 4 показаны графики ускорения при разных соотношениях трафика. Пунктирной линией показано линейное ускорение. При беглом анализе видны хорошие результаты параллельной работы PNSS.

Следует отметить, что графики ускорения имеют некоторые особенности. На всех графиках присутствует сверхлинейное ускорение при числе процессоров до пяти, а после добавления пятого процессора происходит излом кривой.

Такой излом кривой имеет довольно простое объяснение. При моделировании сети на четырех или менее процессорах на каждом двухъядерном процессоре используется только одно из ядер. Каждый логический процесс в реализации MPICH создает два реальных процесса: один процесс инициируется ко-

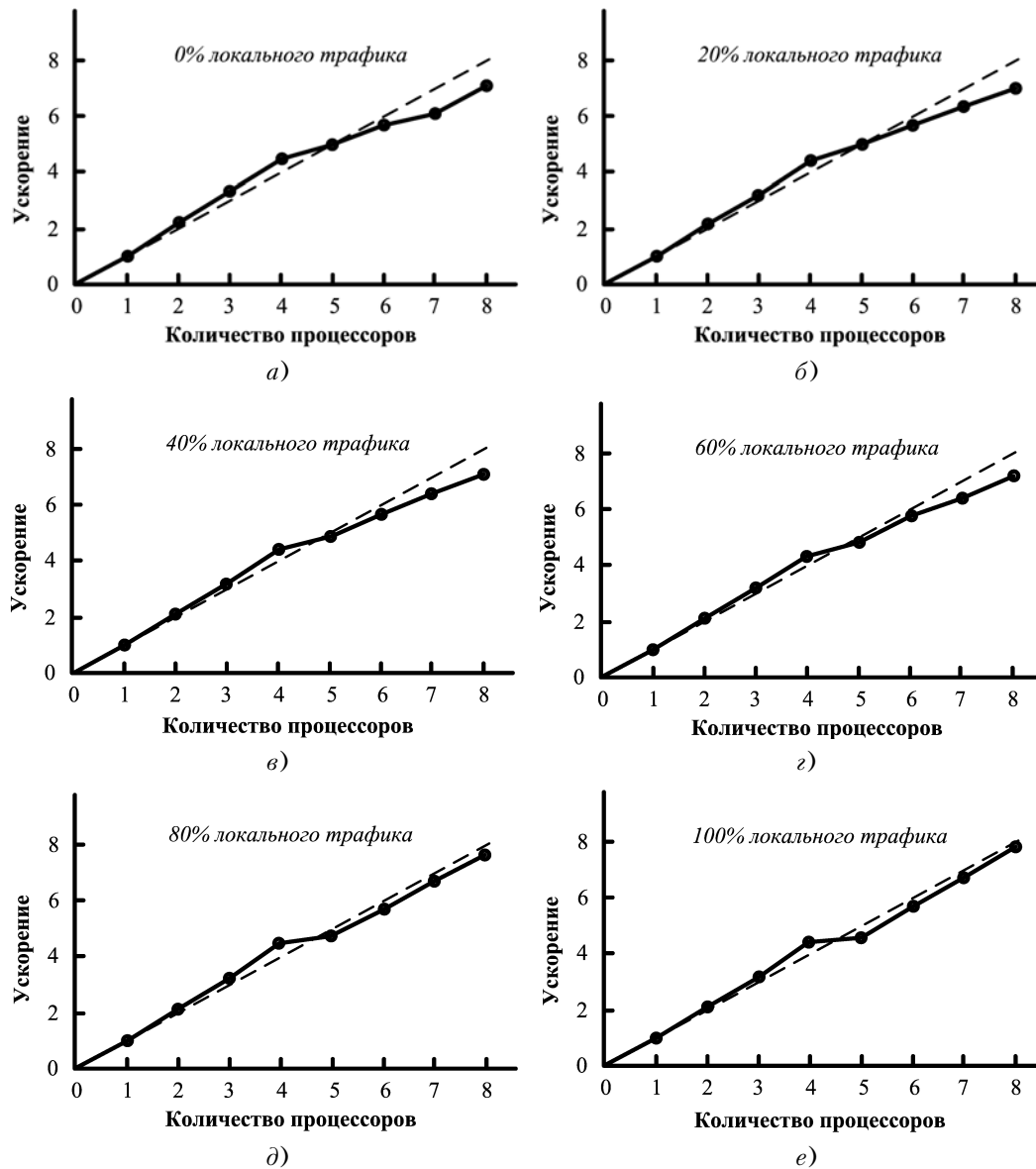


Рисунок 4 – Ускорення паралельної обробки при різних соотношениях трафіка

дом моделювання, а другою – управляє міжпроцесорними пересылками. Якщо один логічний процес виконується двухядерним процесором, то два ядра можуть розділяти між собою MPI-процес і інші системні процеси. Якщо на одному процесорі виконується два логічних MPI-процеса, то на два ядра процесора приходить чотири реальних MPI-процеса, а також системні процеси. В тому випадку, якщо кількість процесів менше п'яти, на кожному з процесорів виконується по одному логічному процесу. Однак при додаванні п'ятого MPI-процеса одному з процесорів приходить виконувати два логічних MPI-процеса. Цим і обумовлено злом кривої при додаванні в систему п'ятого процесора.

Ще однією аномалією графіка є сверхлинейне прискорення. Таке несподіване, але позитивне поведіння обумовлено підвищенням ефективності при зменшенні участків. В частині, розмір множини очікуваних подій має вплив на код обробки подій. По мірі додавання в систему додаткових процесорів на кожен процесор приходить ділячка мережі з меншою кількістю мережних пристроїв, і, відповідно, кожен процесор обробляє менший список подій. Оскільки час додавання і видалення події залежить від загальної кількості подій, на менших ділянках не тільки моделюється менше подій, ніж на більших, але й зменшується час обробки кожної події.

ВЫВОДЫ

Параллельное дискретно-событийное моделирование применяется для сетевого моделирования с переменным успехом. Накладные затраты системы управления событиями весьма высоки, и зачастую метод не очень хорошо масштабируется на большом количестве процессоров. С другой стороны, неограниченное использование методов абстракции может значительно снизить точность модели или вовсе исказить предмет рассмотрения.

В предлагаемой реализации системы сетевого моделирования PNSS достигается баланс использования методов абстракции и распараллеливания. В частности, алгоритмическая маршрутизация позволяет устранить необходимость поддержания таблиц маршрутизации для каждого узла, что ощутимо снижает вычислительные затраты. Предложенный метод распараллеливания частично комбинирует синхронизацию на основе окон с механизмом нулевых сообщений. При этом, по сравнению с исходными методами, устраняется необходимость в проведении глобальной минимизации на каждом шаге оконной синхронизации и снижается количество нулевых посылок. Кроме того, сложные механизмы синхронизации отделяются от кода управления событиями чувствительного к производительности, и напрямую обеспечиваются средствами модуля внешней связи. Преимущество заключается в ограничении межпроцессорных коммуникаций только теми случаями, когда это действительно необходимо.

Также в систему моделирования внедрен функционал автономной балансировки нагрузки, который заключается в анализе предыдущих сеансов моделирования с целью более равномерного распределения работы между процессорами в последующих сеансах.

PNSS обеспечивает возможность моделирования TCP-трафика в высокоскоростных проводных сетях. В ходе экспериментов системой продемонстрированы не имеющие аналогов возможности масштабируемости, как по количеству моделируемых узлов, так и по числу процессоров, задействованных в моделировании. Отдельный процессор может обеспечить моделирование порядка 100 тысяч узлов, при этом 64-процессорная система в состоянии моделировать 10 миллионов узлов, что на порядок больше, чем в ранее известных системах.

Комбинирование возможностей методов абстракции и параллельных вычислений обеспечивает высокий уровень масштабируемости PNSS. Модульный принцип построения системы обеспечивает легкость ее расширения. Использование PNSS делает возможным изучение при помощи моделирования широкого круга больших вычислительных сетей.

ПЕРЕЧЕНЬ ССЫЛОК

1. Zeigler B. Theory of Modeling and Simulation / Zeigler B., Praehofer H., Kim T. – 2nd edition. – Academic Press, 2000. – 510 p.
2. Fujimoto R. Parallel discrete event simulation / Fujimoto R. // Communications of the ACM. – 1990. – № 33(16). – P. 30–53.
3. Корольков І. В. Паралельне моделювання великих обчислювальних мереж: актуальні підходи та проблеми / Корольков І. В., Кудерметов Р. К. // Інформаційні технології та комп'ютерна інженерія. – 2007. – № 1(8). – С. 45–51.
4. Jones D. An empirical comparison of priority-queue and event-set implementations / Jones D. // Communications of the ACM. – 1986. – № 29. – P. 300–311.
5. Sleator D. Self adjusting binary trees / Sleator D., Tarjan R. // Proceedings of the fifteenth annual ACM symposium on Theory of Computing. – 1983. – P. 235–245.
6. RFC 793 Transmission Control Protocol [Электронный ресурс] / J. Postel // Information Science Institute. – September 1981. Режим доступа: <http://rfc.sunsite.dk/rfc/rfc793.html>.
7. Jacobson V. Congestion avoidance and control / Jacobson V., Karels M. // Proceedings of SIGCOMM'88. – 1988. – P. 314–329.
8. Корольков И. В. Повышение производительности моделирования маршрутизации в больших вычислительных сетях / Корольков И. В. // Радіоелектронні і комп'ютерні системи. – 2007. – № 7(26). – С. 20–26.
9. Calvert K. Modeling internet topology / Calvert K., Dor M., Zegura E. // IEEE Communications Magazine. – 1997. – № 35(6). – P. 160–163.
10. Winick J. Inet-3.0: Internet topology generator : Technical Report UM-CSE-TR-456-02, EECS / Winick J., Jamin S. – University of Michigan, 2002. – 19 p.
11. Karypis G. Multilevel k-way partitioning scheme for irregular graphs / Karypis G., Kumar V. // Journal of Parallel and Distributed Computing. – 1998. – № 48. – P. 96–129.
12. Tangmunarunkit H. Network topology generators: degree-based vs. structural / Tangmunarunkit H., Govindan R., Jamin S., Shenker S., Willinger W. // Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications. – ACM Press, 2002. – P. 147–159.
13. Faloutsos M. On power-law relationships of the internet topology / Faloutsos M., Faloutsos P., Faloutsos Ch. // Proceedings of SIGCOMM'99. – 1999. – P. 251–262.

Надійшла 27.06.2008

Розглянуто систему паралельного моделювання великих обчислювальних мереж. Запропонована архітектура націлена на моделювання великих мереж за рахунок паралельного виконання та ефективного використання пам'яті. Система використовує гібридний метод синхронізації на основі вікон та нульових повідомлень і дає змогу досягти прискорення виконання, близького до лінійного. Модульний принцип системи дає змогу легко додавати нові мережні пристрої та протоколи. Запропоновано модулі моделювання маршрутизаторів, каналів зв'язку й джерел трафіку.

Parallel simulation system for large computer networks is discussed. Pro-posed architecture is aimed to simulate big networks by means of parallel computing and effective memory consumption. The system uses hybrid synchronization scheme of null messages and windowing, and allows to reach a close-to-linear acceleration. Modular concept of the system provides for easy addition of new network devices and protocols. Modules for simulating routers, links and traffic sources are offered.