

$$L = \left(\sum_i \frac{L_i}{L_{max}} \right) * \frac{I}{N_{used}}, \quad (7)$$

где L_i – нагрузка на процессоре i , L_{max} – максимальная нагрузка, N_{used} – используемое реально число процессоров.

Условия экспериментов по оценке средней загрузки были такими же, как и в случае оценки оптимизации по времени выполнения. Результаты оценки средней загрузки процессоров показывают, что алгоритм виртуальной сети обеспечивает более рациональное использование процессоров и равномерность их загрузки, нежели метод поиска решения с помощью генетических алгоритмов. Соответствующие результаты приведены на рисунке 5 и рисунке 6.

ЗАКЛЮЧЕНИЕ

Проведенные эксперименты показали эффективность использования модели виртуальной сети для решения задачи о назначении модулей параллельной программы на процессоры многопроцессорной вычислительной системы. С увеличением количества задач рост производительности алгоритма виртуальной сети заметно увеличивается. Кроме того, алгоритм виртуальной сети наряду с улучшением времени выполнения параллельной программы увеличивает равномерность загрузки процессоров и способствует более эффективному их использованию для вычислений.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. И Евсеев. MPI – программный инструмент для параллельных вычислений.
<http://www.csa.ru/~il/Parallel/myMPI/intro.htm>.

УДК 681.3.001.57

Садыхов Р.Х., Радишевский В.А., Отвагин А.В.

АЛГОРИТМ СТАТИЧЕСКОЙ МАРШРУТИЗАЦИИ СООБЩЕНИЙ В УВК РЕАЛЬНОГО ВРЕМЕНИ НА БАЗЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ

(Примечание: Работа выполнена при поддержке гранта БРФФИ Т00-050)

ВВЕДЕНИЕ

В современных управляющих вычислительных комплексах (УВК) и высокопроизводительных вычислительных системах наряду с проблемой оптимального распределения задач по устройствам обработки существует проблема взаимодействия и синхронизации процессов. Эти проблемы наиболее остро стоят при использовании режимов реального времени для функционирования управляющих вычислительных комплексов (УВК РВ). Современные вычислительные системы часто используют для обеспечения взаимодействия процессов специализированные средства операционных систем, либо библиотеки обмена сообщениями. В качестве примера можно привести широко распространенный в мире стандартизованный интерфейс передачи сообщений (Message Passing Interface - MPI)[1]. Данный интерфейс предполагает использование идентификаторов для обмена сообщениями между процессами. Для гарантированного обмена при старте программы создается специальная коммуникационная область, объединяющая все процессоры в единое коммуникационное пространство. Данный подход применяется, как правило, в сильно локализованных высокопроизводительных системах. Однако, применение его в достаточно распределенных системах, таких, как промышленные SCADA-системы (Supervisory Control And Data Acquisition Systems) неоправданно, поскольку управление полным коммуникационным пространством

достаточно трудоемкая процедура.

Оптимизация распределения задач по процессорам системы снимает часть проблем взаимодействия и синхронизации процессов. Однако оставшиеся межпроцессные связи, ставшие «межпроцессорными», нуждаются в продуманной и оптимальной организации для обеспечения требуемых режимов функционирования. Отсюда и вытекает задача оптимальной маршрутизации в УВК и особенно УВК РВ.

1. ПОСТАНОВКА ЗАДАЧИ

Пусть существует множество процессоров P , каждый из которых имеет производительность по обработке собственных или транзитных сообщений $p_i \{i = 1, \dots, n\}$. Процессоры соединены между собой линиями связи пропускной способности C_{ij} . Данную систему можно представить в виде направленного ациклического графа G_p . Пусть также существует параллельный алгоритм, в котором определено множество программных модулей M . Модули обмениваются между собой информацией, при этом между двумя модулями существует поток информации объемом $f_{k,i}$. Все модули некоторым образом распределены по узлам коммуникационной сети G_p , в которой и происходит обмен информацией. Информация каждой взаимодействующей пары модулей передается

через коммуникационную сеть по любому возможному маршруту, конечными точками которого являются процессоры, на которые назначены взаимодействующие модули, причем допустимый маршрут не имеет циклов.

Для обеспечения режимов реального времени данная постановка может быть расширена введением понятия требуемого времени передачи (deadline time), которое определяет максимально возможное время передачи, не нарушающее условий выполнения всех задач в реальном времени.

Результатом работы процедуры оптимизации будет такой набор маршрутов в G_p между парами взаимодействующих процессоров, который обеспечивает наиболее быструю передачу всей информации без существенных задержек и блокировок.

2. МЕТОДЫ МАРШРУТИЗАЦИИ В КОММУНИКАЦИОННЫХ СЕТЯХ

Алгоритмы маршрутизации обычно классифицируют по группам распределенных или централизованных алгоритмов. Преимущества и недостатки этих групп хорошо известны: централизованная маршрутизация, как правило, более оптимальна, поскольку может учитывать все особенности проблемы при наличии полной информации, однако, при высокой динамике изменений в сети, влияющих на качество обслуживания (Quality of Service) и обособленности ее сегментов, предпочтительной является распределенная маршрутизация.

Определяя процедуры обновления информации о маршрутах, выделяют два типа алгоритмов – статические и динамические. Статические алгоритмы эффективно используются в сетях с низкой динамикой и ограниченным перечнем возможных взаимодействий между узлами сети. К таким сетям можно отнести и коммуникационные сети УВК. Особенностью статических алгоритмов является либо однократное построение таблицы маршрутов, либо достаточно долгий период ее обновления, либо обновление по запросу супервизора.

Динамические алгоритмы используют периодическое обновление информации в таблице маршрутов. Этот класс алгоритмов наиболее подходит для сетей с переменной структурой и непостоянным трафиком. Для обеспечения качества обслуживания такая сеть должна уметь гибко перестраиваться. Однако, для обеспечения адаптивности необходима высокая производительность и тесное взаимодействие всех структурообразующих компонентов сети.

Хорошо известным алгоритмом статической маршрутизации является алгоритм Дейкстры [2], называемый также алгоритмом «кратчайшего пути». Суть его заключается в определении кратчайшего пути между двумя вершинами по определенной метрике. В современных телекоммуникационных сетях обычно используют динамическую маршрутизацию с помощью алгоритмов Distributed Bellman-Ford (RFC1075) [3], OSPF (Open Shortest Path First – RFC 1131) [4], OSPF V2 (RFC 2178) [5].

Существует множество целей, которые должны быть достигнуты при построении эффективной таблицы маршрутов в сети. Во-первых, необходимо обеспечить быструю и надежную передачу данных в сети. Кроме этого, необходимо уменьшить среднюю задержку данных при передаче, обеспечивая наибольшую пропускную способность линии связи. Для достижения этих целей часто выдвигается требование обеспечить кратчайший путь, сократив количество транзитных узлов для передачи.

Учитывая специфику задач реального времени и наличие требуемого времени передачи, задача маршрутизации для УВК РВ должна также решать связанные с этим вопросы построения оптимальных маршрутов, удовлетворяющих данному критерию.

Однако использование технологии кратчайшего пути часто приводит к несбалансированной загрузке сети, в то время как надежная передача обычно требует сбалансированного использования линий связи и узлов во избежание перегрузки и потери данных. Еще одной целью может быть минимизация стоимости оборудования проектируемой сети связи [6].

3. ОПРЕДЕЛЕНИЕ ЦЕЛЕВОЙ ФУНКЦИИ МАРШРУТИЗАЦИИ

Поскольку практически все алгоритмы базируются на метафоре «кратчайшего пути», а в качестве метрики обычно выступает расстояние между узлами, очевидным для оптимальной маршрутизации является минимизация целевой функции вида

$$Z = \min \sum_{i,j} H_{i,j}, \quad (1)$$

где $H_{i,j}$ – количество транзитных вершин на пути от вершины i к вершине j . Целью является избежание построения длинных маршрутов, приводящих к неоправданной перегрузке и чрезмерному использованию ресурсов телекоммуникационной сети.

Однако, принимая во внимание ограниченную емкость узлов и линий передачи, использование только кратчайших путей приводит к перегрузке отдельных частей сети – ее «узких» мест. Поэтому является оправданным введение еще одной цели

$$Z_2 = \min (CG_N + CG_L), \quad (2)$$

где CG_N и CG_L – количество совместно используемых узлов и линий связи.

Объединяя две цели и опираясь на время передачи всего потока информации между двумя узлами m и n , как на метрику качества, можно определить данное время в виде

$$T_{m,n} = \sum_{i \in N} t_i + \sum_{j \in L} t_j + \alpha \sum_{i \in N} T_{all} + \beta \sum_{j \in N} T_{all}, \quad (3)$$

где первое слагаемое – время передачи пакета через транзитные узлы, второе слагаемое – время передачи пакета через транзитные линии связи, третье слагаемое – время передачи всех пакетов всех информационных пар через транзитные узлы маршрута, четвертое слагаемое – время передачи всех пакетов всех информационных пар через транзитные линии связи маршрута, α и β – коэффициенты учета влияния всех пакетов ($\alpha > 0, \beta < 1$).

После определения времени передачи информации для каждой из пар можно определить общую оценку для всей совокупности маршрутов

$$T_{opt} = \min \sum_{i,j} T_{i,j}. \quad (4)$$

4. ИСПОЛЬЗОВАНИЕ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ ДЛЯ СТАТИЧЕСКОЙ МАРШРУТИЗАЦИИ

Поскольку задача построения таблицы маршрутов при наличии нескольких оптимизационных целей достаточно сложна и сложность ее растет с увеличением количества возможных связей, можно с уверенностью говорить о ней, как о задаче класса NP-сложных. Для решения таких задач в последнее время часто применяют аппарат генетических алгоритмов.

Генетические алгоритмы основаны на механизмах естественной селекции и генетики. Их преимущество состоит в простоте, надежности и внутреннем параллелизме. Они менее подвержены ограничениям, накладываемым на пространство

поиска и выполняют параллельный поиск в нескольких направлениях на одной итерации. Генетические алгоритмы наиболее подходят для оценки множества возможных комбинаций, представляющих решение. Для определения задачи в терминах генетических алгоритмов используется термин «популяция». Каждая популяция состоит из «хромосом», представляющих собой один из вариантов условного описания исследуемой модели. Описание условно, поскольку методика генетических алгоритмов применима к достаточно широкому кругу задач.

Решение задачи о маршрутизации с помощью генетических алгоритмов рассматривается в работах [6-8]. При этом в качестве популяции может быть выбран как набор возможных маршрутов (хромосома - маршрут), так и несколько возможных комбинаций маршрутов (хромосома - комбинация маршрутов). В первом случае решение представляет собой комбинацию путей, покрывающих все возможные передачи информации. При этом генетические операторы (селекция, мутация и скрещивание) реализуются достаточно просто. Во втором случае алгоритм более трудоемкий и требует больше ресурсов, но позволяет получить более оптимальные решения.

В данной работе предлагается использовать вариант второго подхода. В этом случае каждая хромосома будет состоять из генов, представляющих собой отдельный маршрут. В качестве гена может выступать текстовая строка вида 1-3-4-5-6-8-10. Здесь каждое число означает номер вершины в маршруте. Для такого представления гена достаточно легко можно реализовать оператор мутации при выполнении требования достоверности маршрута, то есть возможности его реального существования. При этом мутация – это простое изменение данного маршрута, то есть получение нового маршрута с теми же конечными точками.

На уровне хромосомы существует цепочка генов, соответствующая каждой возможной передаче информации или информационной паре модулей. Количество информационных пар определяется количеством межпроцессорных передач. Поскольку возможно организовать взаимно однозначное соответствие между позицией в хромосоме и идентификатором информационной пары, соответствующие позиции в разных хромосомах будут определять маршруты с одинаковыми конечными точками. Это позволяет использовать любой вариант классического скрещивания – одно- или двухточечное.

Для оценки хромосомы и проведения селекции необходимо произвести ее отображение на структуру коммуникационной сети и определить по формуле (3) качество каждого маршрута. Отображение происходит путем суммирования нагрузки, полученной каждым процессором и каждой линией связи, в результате последовательного просмотра всех текущих маршрутов. Затем общая оценка определяется согласно (4).

Для учета требований реального времени полученные оценки маршрутов сравниваются с их требуемыми временами передачи. Если в хромосоме присутствует маршрут, не удовлетворяющий этому условию, такая хромосома помечается как непригодная и в результате генетических операций разрушается.

Результатом работы является набор маршрутов для каждой информационной пары с общим минимальным временем передачи.

5. РЕЗУЛЬТАТЫ ЭКСПЕРИМЕНТОВ

Для проведения экспериментов была построена программная модель, сочетающая в себе процедуру оптимизации распределения задач по процессорам и процедуру статической маршрутизации межпроцессорных сообщений.

Для оценки качества маршрутизации полученная в результате оптимизации совокупность маршрутов сравнивалась с совокупностью маршрутов, сформированных алгоритмом «кратчайшего пути». Оценка производилась по единой метрике с использованием выражений (3) и (4).

При проведении экспериментов использовалась программа-генератор, создающая случайные графы с различной плотностью связей. Частные оценки для каждого графа суммировались по категориям, определяемым средним количеством оптимизируемых связей. Результаты приведены на рисунке 1.

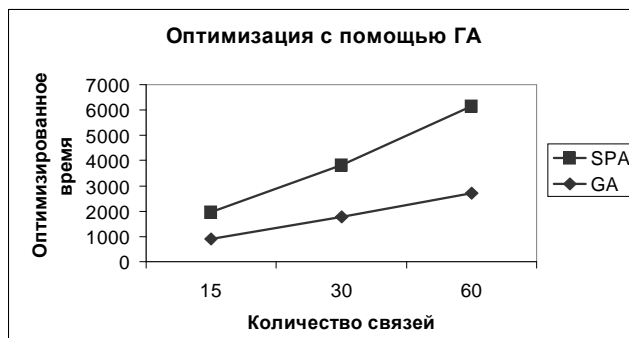


Рисунок 1 – Результаты сравнительных экспериментов.

Здесь SPA – алгоритм кратчайшего пути, GA – генетический алгоритм. На диаграмме хорошо заметна тенденция к увеличению разрыва между графиками с увеличением количества оптимизируемых связей. Это связано с уже указанным недостатком алгоритма кратчайшего пути, а именно перегрузке отдельных линий связи.

ЗАКЛЮЧЕНИЕ

Разработанный алгоритм статической маршрутизации с помощью методики генетических алгоритмов позволяет получить лучшие решения по сравнению с традиционно используемыми алгоритмами. При этом качество функционирования алгоритма повышается с увеличением сложности задачи, что свидетельствует о потенциальной возможности его успешного применения именно в задачах высокой степени сложности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- И Евсеев. MPI – программный инструмент для параллельных вычислений. <http://www.csa.ru/~il/Parallel/myMPI/intro.htm>.
- Dijkstra E. W. A note on two problems in connection with graph. Numer. Math. 1, pp. 269-271(1959).
- RFC 1075, D. Waitzman, C. Partridge, S. E. Deering. Distance Vector Multicast Routing Protocol. Nov-01-1988.
- RFC 1131, RFC Editor. OSPF, <ftp://ftp.isi.edu/in-notes/rfc1131.txt>.
- RFC 2178, RFC Editor. OSPF V2c, <ftp://ftp.isi.edu/in-notes/rfc2178.txt>.
- L. He, N. Mort. Hybrid genetic algorithms for telecommunications network backup routing. // BT Technol. J. Vol.18, No.4, October 2000.
- Cox L A Jr, Davis L and Qiu Y: 'Dynamic anticipatory routing in circuit-switched telecommunications networks', in Davis L (Ed): 'Handbook of Genetic Algorithms', Van Nostrand Reinhold, Ch 11, pp 124—143 (1991).
- Davis L, Orvosh D, Cox A and Qiu Y: 'A genetic algorithm for survivable network design', Proc 5th Intl Conf on GAs, University of Illinois, USA, pp 408—415 (1993).