

цель чтения, что пользователь хочет получить в результате прочтения и основной критерий любой книги – жанр. По сути, фильтр является выбором ключевых слов, которые необходимы для подбора похожих книг.

Рассмотрим другой рекомендательный сервис. «Имхонет» является проводником не только в мир книги, но и в галактику фильмов и игр. Данный сервис собирает как можно больше информации от пользователя путем оценки некоторого количества книг, фильмов. Затем на основании поставленных оценок ищет единомышленников. Именно эти люди будут выступать в роли рекомендателей. Алгоритм, использующийся в сервисе, является коллаборативной фильтрацией, основанной на сравнении пользователей. Проблема «холодного старта» решена тем, что пользователь, желающий получить рекомендацию, будет вынужден оценить некоторое количество объектов.

На основе проведенного анализа, можно сделать вывод, что для онлайн-сервиса рекомендации художественной литературы наилучшим вариантом является метод анализа содержимого объекта. Из текстовой информации проще извлекать ключевые слова, чем отслеживать оценки или просмотры той или иной книги.

Список литературы

1. Гомзин А.Г., Коршунов А.В. Системы рекомендаций: обзор современных подходов // Труды Института системного программирования РАН. – 2012. – Т. 22. – С. 401–417.
2. Филиппова Е. Системы рекомендаций: задачи, подходы, алгоритмы. URL: <http://datareview.info/article/sistemyi-rekomendatsiy-zadachi-podhodyi-algoritmyi/> (дата обращения: 03.03.2016).
3. Джонс Т. Рекомендательные системы: Часть 1. Введение в подходы и алгоритмы. URL: <http://www.ibm.com/developerworks/ru/library/os-recommender1/index.html> (дата обращения 03.03.2016).

УДК 004

ПОСТРОЕНИЕ АРХИТЕКТУРЫ ЛОКАЛЬНОЙ СЕТИ МНОГОПРОЦЕССОРНОЙ ВЫЧИСЛИТЕЛЬНОЙ СИСТЕМЫ

Петрухина И.А., Погребной А.В.

Научный руководитель: Погребной А.В., доцент кафедры ИПС

Национальный Исследовательский Томский политехнический университет,

634050, Россия, г. Томск, пр. Ленина, 30

E-mail: irk-2009@yandex.ru

The following article represents method of finding the best solution for connecting stations to the core network. The main criterion of choosing the variation is a minimum data transfer time, which is achieved by parallelizing dataflow in network. This method is used in modeling real-time system.

Key words: *real-time system, cutting dataflow graph, matrix for conflicts, software load, maximum empty subgraph, core network.*

Ключевые слова: *система реального времени, разрезание графа передачи данных, матрица наличия конфликтов, программная нагрузка, максимально пустой подграф, базовая сеть.*

Система реального времени – это система, в которой правильность вычислений зависит не только от логической корректности результата, но и от времени, в течение которого этот результат получен. Часто различные станции находятся в разных местах и соединяются через некоторый канал связи (Интернет, кабель), т. е. системы реального времени имеют распределенную структуру. Это могут быть вычисления на метеостанциях, атомных станциях, в авиационной отрасли и т. д. Для предотвращения неверных результатов моделируют разрабатываемую систему и проверяют работоспособность на тестовых данных.

Целью данной работы является определение и программная реализация варианта подключения станций к магистралям сети по критерию минимизации суммарного времени на передачу данных в одном цикле моделирования. Минимизация времени достигается за счет распараллеливания передач данных в магистралях сети, что также снижает общую загрузку магистралей сети.

При решении данной задачи будем исходить из того, что нам известно:

- количество станций, для примера возьмем шесть станций;
- информационный граф программной нагрузки, в котором определены объемы данных, передаваемых между модулями и позициями за один цикл моделирования;
- модули и позиции распределены по станциям, т. е. мы имеем информационный граф с дугами, требующими сетевой ресурс.

Выберем базовую сеть из трех связанных между собой магистралей (рис. 1, а).

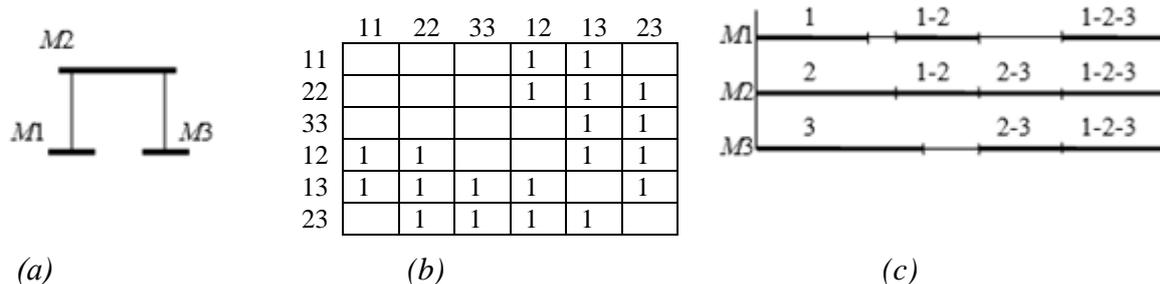


Рис. 1. Базовая сеть (а), матрица конфликтов передач (b) и пример диаграммы загрузки магистралей сети (с) [1]

Главным вопросом является то, как лучше всего подключить станции к магистралям для минимизации времени на передачу данных в сети. Очевидно, что дугам, по которым передаются большие объемы данных, желательно поставить в соответствие передачи данных между станциями, подключенными к одной магистрали. К каждой магистрали логично подключать не менее двух станций.

Метод решения задачи основан на выявлении в структуре сети возможностей параллельных передач данных. С этой целью выполняется последовательность следующих операций:

- строится граф передачи данных между станциями сети;
- выполняется разрезание графа передач данных на минимально связанные подграфы (рис. 2, а, вариант разрезания графа на три подграфа выделен пунктирными линиями);
- выбирается вариант подключения станций к магистралям сети (рис. 2, б);
- строится матрица наличия конфликтов Q между передачами данных при доступе к магистралям сети;
- строится диаграмма совмещения параллельных передач данных [1].

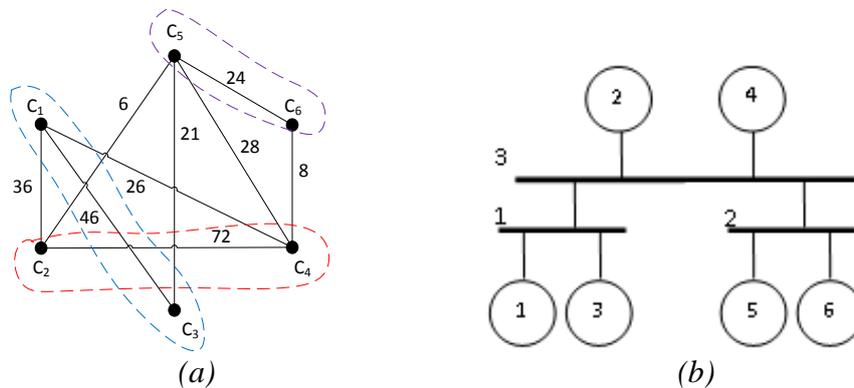


Рис. 2. Варианта разрезания графа (а) и базовая сеть для него (b)

Размерность матрицы $Q = \|Q_{ij,kl}\|$ определяется числом ребер в графе передач данных. Ребра (c_i, c_j) в примере матрицы Q на рис. 2, а обозначены кодовыми номерами ij , например, ребро (c_1, c_2) обозначено 12. Элемент $Q_{ij,kl}$ обозначает наличие ($Q_{ij,kl} = 1$) или отсутствие ($Q_{ij,kl} = 0$) конфликта по доступу к магистралям при одновременной передаче данных между станциями по ребрам (c_i, c_j) и (c_k, c_l) .

Выделим основные шаги методики поиска наименьшего времени передач данных для конкретного распределения вершин по магистралям:

Шаг 1: *получение матрицы наличия конфликтов*. Значения элементов $Q_{ij,kl}$ определяются с использованием матрицы конфликтов M (рис. 1, b) по логическому выражению [2] (где g, h, s – номера магистралей от 1 до 3):

$$\forall c_i \in c(M_g), c_j \in c(M_h) \& c_k \in c(M_s), c_l \in c(M_g) [M_{gh,gs}] \Rightarrow (Q_{ij,kl} = 1), \text{ иначе } (Q_{ij,kl} = 0).$$

Шаг 2: *получение сокращенной матрицы конфликтов*. Из матрицы Q удаляются строки и столбцы, у которых все элементы $Q_{ij,kl} = 1, ij \neq kl$, что означает, что при передаче данных между соответствующими станциями заняты все три магистрали. Суммарный объем передач данных в сети R равен 0.

Шаг 3: *выделение максимального пустого подграфа*. Сокращенная матрица Q принимается в качестве матрицы связности вершин графа (в скобках у номеров вершин графа указаны объемы данных, передаваемых между соответствующими станциями). В графе Q выделяется максимальный пустой подграф. При этом вершины, которые уменьшили свой объем данных на предыдущем шаге, выбираются в первую очередь.

Шаг 4: *удаление вершины*. Вершина выделенного подграфа с минимальным объемом передач данных исключается из графа Q (если в подграфе существует вершина с таким же минимальным объемом, она исключается тоже). Из объемов передач данных других вершин подграфа вычитается объем данных удаленной вершины r_{ij} . К суммарному объему передач данных в сети R прибавляется r_{ij} . Если в графе не удалены все вершины, переходим на шаг 3. Иначе к R прибавляется объем данных удаленных вершин на шаге 2.

Таким образом, в ходе данного алгоритма вычисляется суммарный объем передач данных R , приходящийся на каждую магистраль, что и характеризует время передач данных в сети. На рис. 3, а и 4 показана пошаговая работа алгоритма для варианта распределения станций на рис. 2, b. Когда граф становится полным, выделение параллельных передач невозможно – к R прибавляются r_{ij} оставшихся вершин. Для наглядности строится диаграмма совмещения параллельных передач данных, показывающая загруженность каждой магистрали (рис. 4, с).

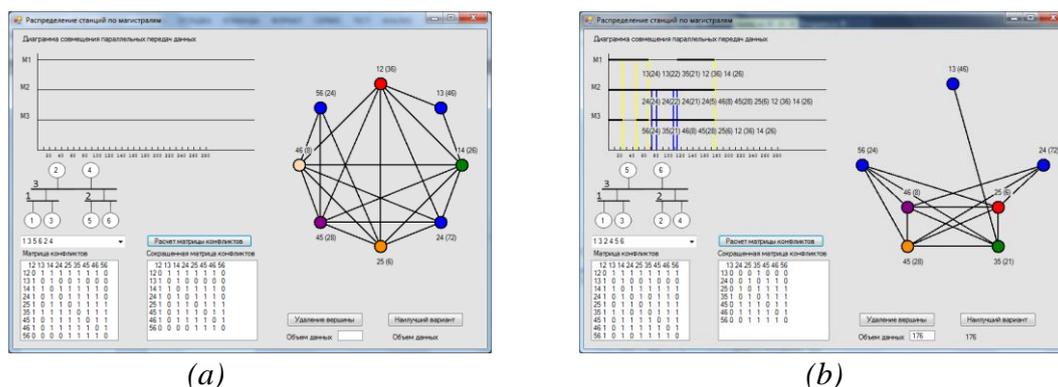


Рис. 3. Шаги 1–3 (a) и наилучший вариант распределения станций (b)

Задача поиска максимально пустого подграфа решалась как подзадача раскраски графа с помощью приближенного алгоритма последовательного раскрашивания [3].

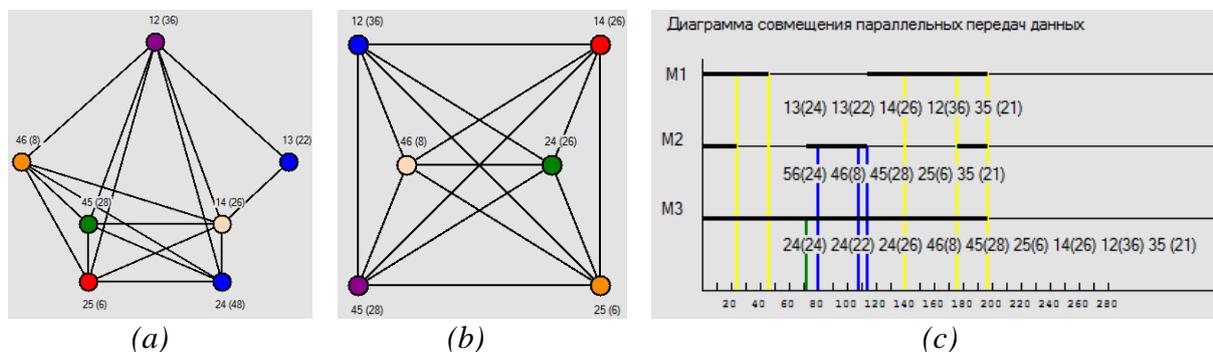


Рис. 4. Шаги 4–3₂–4₂ (a) и диаграмма совмещения параллельных передач данных (b)

Таким образом, для выбранной базовой сети и варианта подключения станций (рис. 2) последовательная цепь передач данных за счет использования параллельных передач сократилась с 267 до 197 единиц. После генерации всевозможных перестановок [4] и расчета R для каждой был найден наилучший по времени вариант распределения станций с $R = 176$ единиц (рис. 3, b). Как мы видим, вариант разрезания графа (рис. 2, a) был удачен, но также играет роль и распределение подграфов на магистралях. Сокращение значения времени передач в сети в рассматриваемом примере удалось достигнуть за счет наличия соответствия между структурой графа передач, содержащего три не пересекающихся подграфа с большими весами ребер и вариантом базовой сети с тремя магистралями.

Список литературы

1. Погребной А.В. Проектирование структуры локальной сети для распределенной вычислительной системы реального времени (Электронный ресурс) / [Известия ТПУ] – 2007. Режим доступа: http://www.lib.tpu.ru/fulltext/v/Bulletin_TPU/2007/v311/i5/19.pdf (дата обращения: 24.02.2016).
2. Погребной В.К. Автоматизированное проектирование распределённых систем реального времени: учебное пособие – Томск: Изд-во ТПУ, 2010. – 326 с.
3. Новиков Ф.А. Дискретная математика для программистов. – СПб: Питер, 2000. – 304 с.
4. Кручинин В.В. Процедурно – ориентированное программирование (Электронный ресурс) Режим доступа: <http://www.ie.tusur.ru/books/ProgDevelTech/index.htm> (дата обращения: 26.02.2016).