

НАХОЖДЕНИЕ ЦЕНТРОВ И МЕДИАН В СЕТЯХ ПРОИЗВОЛЬНОЙ ТОПОЛОГИИ

В статье рассматривается задача нахождения центров и медиан графов в распределенных сетях произвольной топологии. Предлагается новый асинхронный алгоритм для решения данной задачи, который состоит из трех модифицированных подалгоритмов: проверки связности, нахождения кратчайших расстояний между парами вершин графа и выбора лидера. Реализована моделирующая программа, основанная на данном алгоритме.

Ключевые слова: центр графа, медиана графа, распределенный алгоритм, нахождение кратчайших расстояний, выбор лидера.

Введение. С недавних пор мы являемся свидетелями широкого и возрастающего интереса к распределенным алгоритмам, а также к проектированию и анализу децентрализованных сетей (например, MERIT или ARPANET).

Геометрические понятия центральности тесно связаны с задачей размещения, и поэтому в применении задачи нахождения центров и медиан графов возникают два интересующих нас направления: первое направление связано с понятием центральности графа, второе – с широко известными минимаксными и минисуммными задачами размещения.

Центральность. Пусть G – связный граф, а x и y – две его несовпадающие вершины. Длина кратчайшего (x, y) маршрута называется расстоянием между вершинами u и v и обозначается через $d(x, y)$. Эксцентриситет вершины $e(x)$ в графе G и радиус $r(G)$, соответственно, определяются как $e(x) = \max_{y \in V} d(x, y)$, $y \in V$ и $r(G) = \min_{x \in V} e(x)$. Центром графа G называется множество $C(G) = \{x \in V \mid e(x) = r(G)\}$. Граф может иметь единственную центральную вершину или несколько центральных вершин. Наконец, центр графа может совпадать с множеством всех вершин. Например, центр простой цепи P_n при четном числе вершин n состоит ровно из двух вершин, а при нечетном – из одной; для цикла же C_n все вершины являются центральными.

Статус вершины $d(x)$ и статус графа $\sigma(G)$ определяются, соответственно, как $d(x) = \sum_{y \in V} d(x, y)$ и $\sigma(G) = \min_{x \in V} d(x)$. Медианой графа G является множество $M(G) = \{x \in V \mid d(x) = \sigma(G)\}$.

Рейтинги узлов сети по *центральности*, под разными названиями, применяются в социальных сетях, сетях цитирования, гипертекстовых и других сетях. Модель случайного блуждания показывает себя эффективной применительно к поведению пользователя, путешествующего по связям-ссылкам гипертекстовой сети. Она заложена в популярной поисковой системе *Google*, в ее методе получения рейтинга страниц Всемирной Паутины [1, 2]. Узлы (вэб-страницы), принадлежащие центру или медиане, – это “авторитеты”, в которые сеть чаще всего приводит путешественников [1].

В сети, состоящей из информационных узлов и локальных смысловых связей, центральность узла коррелирует с его принадлежностью к макроединицам смыслового содержания сети. Рейтинг узлов сети по “макровости”, в сочетании с обходом узлов методом поиска в глубину, позволяет получать связанные тематические конспекты содержания сети [3, 4], что может быть использовано в автоматизированной системе реферирования.

Минимаксные и минисуммные задачи размещения. Задача нахождения центра в графе тесно перекликается с задачей нахождения медианы. Эти задачи также называются минимаксными и минисуммными задачами размещения, в первой задаче требованием является минимизация наибольшего расстояния от произвольной вершины графа до

ближайшего центра графа. Вторая задача минимизирует сумму всех расстояний от вершин графа до ближайшей медианы.

Размещение центров и медиан имеет большое разнообразие применений, так как размещение общего ресурса в центре графа минимизирует затраты при разделении ресурсов между узлами с различным местоположением.

В распределенных сетях информация о топологии играет важную роль. Например, в сетях хранения и отправления с пакетной коммутацией, пакеты, оставляющие источник, направляются к промежуточным узлам, таким образом, для каждого узла сети существенно иметь некоторую информацию о топологии сети (матрицу смежности [5, 6], матрицу расстояний [7] и др.)

В общем, можно полезно использовать информацию о топологии сети для развития эффективных алгоритмов сети: например, число шагов, необходимых для синхронизации узлов, может быть минимизировано, если центр сети известен.

Распределенные системы. Термин «распределенная система» подразумевает взаимосвязанный набор автономных компьютеров, процессов или процессоров. Компьютеры, процессы или процессоры упоминаются как узлы распределенной системы. Будучи определенными как «автономные», узлы должны быть, по крайней мере, оборудованы своим собственным блоком управления. Чтобы быть определенными как «взаимосвязанные», узлы должны иметь возможность обмениваться информацией.

Так как процессы могут играть роль узлов системы, определение включает программные системы, построенные как набор взаимодействующих процессов, даже если они выполняются на одной аппаратной платформе. В большинстве случаев, однако, распределенная система будет, по крайней мере, содержать несколько процессоров, соединенных коммутирующей аппаратурой.

Более ограничивающие определения распределенных систем могут быть также найдены в литературе. Таненбаум [8], например, называет систему распределенной, только если существуют автономные узлы, прозрачные для пользователей системы. Распределенная система в этом смысле ведет себя как виртуальная самостоятельная компьютерная система, но реализация этой прозрачности требует разработки замысловатых алгоритмов распределенного управления.

Модель вычислений. В изучении распределенных алгоритмов часто используются несколько различных моделей распределенной обработки информации. Выбор определенной модели обычно зависит от исследуемой в распределенной системе задачи и от типа представленного алгоритма. Различные модели распределенных алгоритмов представлены в работах [9], [10], [11] и др.

Распределенные вычисления обычно понимаются как набор дискретных событий, где каждое событие - это атомарное изменение в конфигурации (состояния всей системы).

Связь в распределенной системе обеспечивается путем передачи сообщений. Узел является реактивным (или управляемым сообщением), в том смысле, что обычно он только исполняет вычисление (включая отправление сообщений другим узлам) как ответ на получение сообщения от другого узла.

Новый алгоритм нахождения центров и медиан. Главной особенностью нашего алгоритма является его асинхронность и применимость к произвольным сетям.

В зависимости от сложности задач, выполняемых подсистемами (узлами) распределенной системы, может потребоваться, чтобы эта подсистема была разработана структурированно. Программное обеспечение для осуществления данного вида распределенной сети обычно структурируется в зависимых модулях, каждая из которых выполняет определенную функцию и зависит от действий, предлагаемых другими модулями. Модули называются слоями или уровнями в контексте реализации сети. При проектировании сети важно определить число слоев и интерфейсов.

В рамках нашего алгоритма будем предполагать, что распределенная система состоит из n -идентичных узлов, обозначенных уникальными именами $N_0, N_1, N_2, \dots, N_{n-1}$. У каждого узла есть три различных процессора, программа которых составлена из

атомарных шагов. Эти три процессора соответствуют трем различным слоям алгоритма (рис.1). Эти три слоя будем называть проверка связности, вычисление эксцентриситета (статуса) и вычисление центра (медианы). У каждого процессора имеется свой собственный буфер передачи сообщений. Каждый узел хранит информацию о своих соседях и имеет возможность обмена сообщениями непосредственно со своими соседями. Под соседями мы подразумеваем узлы, расположенные от него на ограниченном расстоянии. Связи между узлами считаются надежными. Связи и буфера осуществлены в порядке FIFO. Задержка передачи сообщений не определена.



Рис. 1. Архитектура узла.

Так как описываемый алгоритм является распределенным, каждый узел выполняет одинаковый алгоритм и задача возлежит на взаимодействии или передаче сообщений каждого узла. Но сообщения передаются не только между процессорами одинакового типа, а в рамках алгоритма внутри одинаковых слоев. На рис. 2 показана передача сообщений между двумя узлами.



Рис. 2. Передача сообщений между слоями узлов.

В алгоритме существуют четыре типа сообщений, и они являются одинаково структурированными. В структуру сообщения входят: идентификатор (адрес) узла назначения, идентификатор (адрес) узла отправителя, тип и данные сообщения.

Основу алгоритма составляют три модифицированных подалгоритма (проверка связности, нахождение кратчайших расстояний и выбор лидера), каждый из которых функционирует в различном слое. Основные алгоритмы, которые использованы для решения задачи нахождения центра и медианы графа для общих сетей были выбраны по следующим причинам: прежде всего, они являются асинхронными и применяются в произвольных сетях. Существующие алгоритмы для распределенных систем обычно охватывают только специфические типы сетей и топологии. Вторая причина – простота, для их лучшего понимания мы можем вычислить и протестировать их на маленьких сетях.

Потоковый граф процесса для данного алгоритма показан на рис. 3.



Рис. 3. Поточковый граф процесса.

После получения процессором сообщения должен быть обработан следующий шаг, но перед этим должно быть создано новое сообщение, то есть должны быть доступны необходимые данные. Каждый слой алгоритма зависит от более низкого слоя, так как он должен обеспечить его информацией, необходимой для обработки, иначе вычисления продолжаются и процессор бездействует. Алгоритм начинается с выполнения тестового слоя, проверяется связность сети, если сеть не связна, возвращается её связная часть, содержащая узел `node0`. Затем начинается выполнение второго слоя, для каждого узла вычисляется эксцентриситет (статус). Эта информация передается следующему слою для вычисления центра (медианы).

Решая задачи в распределенной асинхронной системе, необходимо обратить внимание на две важные проблемы: завершение задачи и возникновение отказа.

Возникновение отказа не может быть предсказано. Данная проблема является одной из важнейших, которую должны решить проектировщики асинхронных систем.

Опишем метод для обнаружения завершения.

Мы рассматриваем ситуации, где во время выполнения алгоритма каждый процессор в состоянии контролировать свои собственные вычисления и решать, поддерживается ли «условие локального завершения». Если данное условие завершения сохраняется в некотором процессоре, то никакие сообщения не могут быть им переданы. Более того, локальное условие завершения сохраняется, пока не будет получено сообщение от другого процессора.

Завершение происходит в некоторое время t , если:

- а) Локальное условие завершения выполняется во всех процессорах во время t .
- б) Никакое сообщение не находится на пути линии связи во время t .

Завершение происходит во время \bar{t} , если \bar{t} - наименьшее время t , для которого выполняются вышеупомянутые условия а) и б). Наша цель состоит в том, чтобы обнаружить завершение в пределах конечного промежутка времени после его возникновения. Заметим, что если завершение произошло в некоторое время t , тогда оно верно для каждого последующего времени $t' > t$, так как никакие сообщения не будут переданы после времени t и локальное условие завершения останется верным во всех процессорах.

Литература:

1. Kleinberg J.M. Authoritative sources in a hyperlinked environment. – J. of the ACM, 1999. – Vol.46. N.5. – P. 604-632.
2. Brin S., Page L. The anatomy of a large-scale hypertextual Websearch engine. – Proc. of the 7-th WWW Conference. – Brisbane, Australia, 1998. – P.107-117.

3. Chelnokov V.M., Zephyrova V.L. Hypertext macrodynamics. – Lecture Notes Comput. Sci. – Berlin: Springer-Verlag, 1995 – Vol. 1015. – P.105-120.
4. Chelnokov V.M., Zephyrova V.L. Collective phenomena in hypertext networks. – Proc. of the Hypertext'97 Conference (Southampton, UK). – ACM, 1997. – P. 220-221.
5. Tajibnapis W. D. The design of a topology information maintenance scheme for a distributed computer network. – Proc. ACM Conference, 1974. – Vol.1. – P. 358-364.
6. Tajibnapis W. D. A correctness proof of a topology information maintenance protocol for a distributed computer network. – CACM, 1977. –Vol.20. N. 7. – P.477-485.
7. Chang E.G. Decentralized algorithms in distributed systems. – Tech. Rep. CCRG-103. – Computer Systems Research Group, University of Toronto, 1979.
8. Таненбаум Э.С. Компьютерные сети. – СПб.: Питер, 2003. – 992 с.
9. Barbosa V.C. An introduction to distributed algorithm. – MIT Press, Cambridge, MA, 1996.
10. Bertsekas D.P., Tsitsiklis J.N. Parallel and distributed computation. – McGraw-Hill, 1996.
11. Lynch N.A. Distributed algorithms. – San Francisco, Morgan Kaufmann, 1997.