УДК 519.17+519.24

Некоторые вопросы параллельной реализации алгоритма полного перебора (на примере задач надёжности сетей)

А.С. Родионов, А.В. Сакерин, Д.А. Мигов *

Рассматривается задача эффективного распараллеливания алгоритма полного перебора, применяемого при расчёте некоторых характеристик надёжности сетей с ненадёжными элементами, а также в задачах комбинаторной оптимизации. Описываются общие подходы и конкретные алгоритмы, позволяющие эффективно использовать вычислительные ресурсы при решении задач указанных классов.

Ключевые слова: надёжность сети, случайный граф, параллельный алгоритм.

1. Введение

Многие алгоритмы анализа и оптимизации структур, моделируемых случайными графами (СГ), в частности при рассмотрении их надёжности, требуют полного перебора возможных реализаций этих случайных графов. При этом в большинстве случаев вычисление целевого функционала можно проводить независимо для каждой реализации, что позволяет естественное распараллеливание процесса вычислений. В докладах на конференциях [1–4] представлены предварительные результаты по параллельным алгоритмам вычисления отдельных характеристик надёжности сетей с ненадёжными каналами, моделируемыми случайными графами. В предлагаемой работе представлены обобщения, позволяющие выбор наилучшего способа распараллеливания с учётом мощности и архитектуры инструментальных вычислительных средств. В качестве примеров рассматриваются задачи вычисления:

- вероятности связности подмножества вершин (k-терминальная связность);
- средней вероятности парной связности;
- k-терминальной связности при ограничении количества рёбер в соединяющих путях (на диаметр графа).

2. Вычисление характеристик случайных графов

Будем говорить о вычислении математического ожидания произвольной метрической функции графа G. Рассматривается следующая модель случайного графа.

$G(n, m) = (X, U)$ – неориентированный граф с множеством вершин X и множеством рёбер U.

$n = |X|$ – число вершин графа.

$m = |U|$ – число рёбер графа.

p_{ij} – вероятность того, что ребро e_{ij} исправно.

$q_{ij} = 1 - p_{ij}$ – вероятность того, что ребро e_i разорвано.

$F(G)$ – метрическая функция графа.

*Работа поддержана грантами РФФИ № 12-07-00065, 12-01-31352, 13-07-00589 и грантом Президента России поддержки ведущих научных школ НШ 5176.2010.9.

$\overline{F}(G)$ – математическое ожидание функции $F(G)$.
G/e – граф, полученный из G стягиванием оконечных вершин ребра e.
$G\backslash e$ – граф, полученный из G удалением ребра e.
Рёбра выходят из строя (отказывают) независимо друг от друга.

Если число вершин и рёбер графа ясно из контекста, граф обозначается как просто G. Множество вершин (рёбер) конкретного графа G обозначается как $X(G)$ ($U(G)$).

Согласно этой модели, в общем случае имеем:

$$\overline{F}(G) = \sum_{H \in \Gamma} P(H)F(H), \tag{1}$$

где Γ – множество всех возможных реализаций графа G, которые приходится получать для использования формулы (1).

При вычислении характеристик связности (вероятность связности всех или подмножества вершин, средней вероятности парной связности, среднего размера максимального связного подграфа и др.) большей эффективностью часто обладает метод факторизации [5, 6], при котором для произвольно выбранного ребра e_i

$$\overline{F}(G) = p_i\overline{F}(G/e_i) + (1 - p_i)\overline{F}(G\backslash e_i).$$

При вычислении средней вероятности парной связности можно использовать и формулу получения взаимнооднозначно связанного с ней математического ожидания числа несвязных пар вершин

$$N(G) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} a_{ij} w_i w_j, \tag{2}$$

где w_i – вес вершины, ожидаемое число стянутых в неё вершин при применении на предварительных этапах редукции висячих вершин, факторизации по формуле (2) или ветвления по цепям [7],

$$\overline{R}(G) = \frac{C_n^2 - N(G)}{C_n^2}. \tag{3}$$

Существенное ускорение вычисления некоторых характеристик связности (при сохранении уровня алгоритмической сложности) можно достичь заменой перебора вариантов разрушения графа перебором вариантов разрушения его покрывающего дерева [8]. Основная формула получения вероятности связности графа в этом алгоритме имеет вид:

$$R(G) = R(T_0) + \sum_{i=1}^{N-1} \sum_{j=1}^{C_{N-1}^i} P_S(E_i^j) I(H_i^j), \tag{4}$$

где $P_S(E_i^j)$ есть вероятность удаления из T_0 i рёбер j-ым способом, H_i^j – соответствующий псевдограф связей компонент разрушенного дерева, $I(G)$ – индикаторная функция связности графа. Рассматриваются все возможные сочетания i рёбер.

Удаление из дерева i ребер приводит к его распаду на $i + 1$ часть (блок). Каждый из этих блоков содержит некоторое подмножество вершин графа, связанных между собой. Связность графа в целом может быть получена только за счёт рёбер, не входящих в дерево. Ребро между двумя вершинами графа компонент H существует, только если множество рёбер графа $G\backslash T_0$, соединяющих вершины соответствующих блоков, не пусто. Пусть между некоторыми блоками существуют k рёбер с вероятностями присутствия $p_1, p_2, \ldots, p_k$. Тогда вероятность присутствия соответствующего ребра графа H есть

$$p = 1 - \prod_{i=1}^{k}(1 - p_i). \tag{5}$$

При использовании этого подхода для вычисления математического ожидания числа несвязных пар вершин СГ имеем:

$$N(G) = \sum_{i=0}^{N-1} \sum_{j=1}^{C_{N-1}^{i}} P_S(E_i^j) N(H_i^j). \tag{6}$$

Вес вершины псевдографа равен сумме весов вершин соответствующих поддеревьев.

3. Организация вычислений

При всей кажущейся простоте и «естественности» распараллеливания приведённых выше формул, в основном связанного с получением частичных сумм независимо вычисляемых значений, существует немало «подводных камней», скрывающихся за этой простотой.

3.1. Метод факторизации

При использовании рекурсивной формулы (2) фактически строится дерево разбора, листья которого соответствуют различным вариантам разрушения графа. Преимущество использования метода заключается в возможности прекращения ветвления по отдельным ветвям в случае существования метода непосредственного вычисления функционала. Например, при вычислении вероятности связности графа, получение несвязного графа позволяет прекратить дальнейшее ветвление, поскольку независимо от состояния оставшихся рёбер связность графа не изменится и значение индикаторной функции связности останется равным 0. Кроме этого, существуют конечные формулы вычисления вероятности связности графов малой размерности (до 5 вершин [9]) и графов специальной структуры, не говоря уже о возможности редукции размерности графов, соответствующих промежуточным узлам дерева разбора [10].

«Естественная» реализация метода, когда при рекурсии одна из ветвей обрабатывается на текущем ядре, а другая ветвь отправляется на другое свободной ядро с последующим ожиданием выполнения, очевидно неэффективна в силу существенного разброса времени вычисления по ветвям. От ожидания легко избавиться, передавая ветви не только граф, но и вероятность его получения (т.е. произведение вероятностей присутствия-отсутствия рёбер в пути до отправляемого на расчёт графа) с последующей отправкой результата, в случае окончания ветвления, процессу, осуществляющему суммирование. Отсутствие свободных процессов можно обрабатывать двумя путями: 1) проводить дальнейший расчёт всего поддерева текущим процессом; 2) создать очередь пар вероятность–граф, к которой будут обращаться освобождающиеся процессы. В любом случае, на начальном этапе невозможно загрузить все ядра, и чем больше ядер выделено для задачи, тем больше времени пройдёт до их полной загрузки.

Ещё одно замечание связано с затратами на передачу данных между ядрами: на определённом этапе, при достижении размерности промежуточных графов, ещё не позволяющей применять конечные формулы, но близкой к этому, затраты на передачу данных и запуск/завершение подчинённого процесса начинают превышать затраты на завершение расчётов процессом-мастером, и поэтому отправку графов в очередь необходимо прекращать. Эта размерность зависит от производительности инструментальной вычислительной системы и её коммуникационной среды и должна определяться экспериментально. Так, для кластера НКС-30Т Сибирского суперкомпьютерного центра коллективного пользования СО РАН (`http://www.sscc2.ru`) при вычислении вероятности связности всех вершин случайного графа эта размерность равна 11.

3.2. Прямой перебор реализаций СГ

При использовании формулы (1) проблема загрузки вычислительных ядер не существует, достаточно распределить между ними варианты разрушения графа. Однако тут также необходимо решить несколько задач.

«Естественное» решение заключается в разделении поровну вариантов с удалением 1, 2, 3,... рёбер. Однако это требует построения систематического генератора расстановок k единиц по m местам в маске разрушения (1 означает разрушение соответствующего ребра). Кроме этого, генератор должен обеспечить равномерное распределение масок по вычислительным ядрам.

Вместе с тем, общее число вариантов разрушения графа $G(n, m)$ равно $N = 2^m$ и каждое из чисел $0, 1, \ldots, N$ в двоичном m-разрядном представлении соответствует одной из возможных масок разрушения. Таким образом, достаточно равномерно распределять между ядрами эти числа. Для простоты выражений далее считаем, что число ядер K, выделенных под вычисление $F(G)$, кратно степени двойки, т.е. N делится нацело на K. Отклонение от этого условия легко учесть некоторым простым правилом распределения остатка по ядрам (например, случайным образом).

Отметим, что изначальная передача i-му ядру диапазона $2^m(i-1)/K, \ldots, 2^m i)/K - 1$ не является разумным решением, так как время расчёта $F(G)$, как правило, варьируется в широких пределах и, следовательно, подобное распределение может привести к тому, что одни ядра уже закончат вычисления, тогда как другие ещё далеко не исчерпают выделенные им диапазоны. Решение заключается в динамическом распределении более мелких диапазонов по $2^m(i-1)/K/d, \quad d = 2^l$ вариантов. Диапазоны должны быть не слишком мелкими, чтобы не загружать коммуникационную систему.

3.3. Двойной параллелизм

При вычислении некоторых характеристик СГ, например математического ожидания числа несвязных пар вершин СГ, происходит накопление сумм значений, которые сами получаются с помощью параллельных алгоритмов.

Как отмечалось выше, получить эту характеристику можно с помощью формул (2) либо (6). В обоих случаях можно параллельно организовать вычисление слагаемых сумм либо частичных сумм, при этом значение слагаемого вычисляется нетривиально, в том числе с возможным применением метода факторизации. Таким образом, распределение пар вершин (для формулы (2)) или вариантов разрушения (для формулы (6)) происходит не между ядрами, а между группами ядер, которые используются для применения параллельного алгоритма вычисления соответствующего функционала.

Очевидно, настройка такого двойного параллельного алгоритма представляет собою нетривиальную задачу, требующую проведения большого количества численных экспериментов.

4. Заключение

В своей работе мы показали основные проблемы организации параллельного исполнения полного перебора в некоторых комбинаторных задачах теории графов. Предложены некоторые общие подходы, применимые к различным задачам и позволяющие балансировать нагрузку при использовании большого количества вычислительных ядер при минимизации обменов и синхронизации между ними. Дальнейшие исследования связаны с выбором наиболее подходящих алгоритмов и их параметров при использовании различных архитектур инструментальных ЭВМ, а также с построением имитационных моделей параллельных программ решения рассматриваемых задач для целей анализа их масштабируемости. Следует ожидать, что подобно моделям, рассмотренным в [11], такое моделирование приведёт к выявлению узких мест алгоритмов при их отображении на ЭВМ с очень большим числом вычислительных ядер (сотни тысяч и миллионы).

Литература

1. *Мигов Д.А.* Параллельный метод для расчёта структурной надёжности сети // Тезисы 14-й Всероссийской конференции молодых учёных по математическому моделированию и информационным технологиям (Томск, 2013). Новосибирск, ИВТ СО РАН, 2013. С. 42.
2. *Каневский В.И., Мигов Д.А., Родионов А.С.* Параллельный алгоритм расчёта математического ожидания числа несвязных пар вершин в графе // Труды X Международной Азиатской школы-семинара «Проблемы оптимизации сложных систем», Часть 1, 2014, – Новосибирск: ИВМиМГ СО РАН, – С. 346-351.
3. *Migov D., Rodionov A.* Parallel Implementation of the Factoring Method for Network Reliability Calculation // ICCSA 2014. Springer Lecture Notes in Computer Science. Vol. 8584, issue 6, Heidelberg: Springer, – 2014, P. 654–664.
4. *Migov D.A.* A parallel method for network probabilistic connectivity calculation // Abstracts of the Int. conf. «Advanced mathematics, computations and applications – 2014». ICM&MG SB RAS, 2014, p. 103.
5. *Colbourn Ch. J.* The combinatorics of network reliability. N.Y.: Oxford Univ. press, 1987. 160 p.
6. *Satyanarayana A., Chang M.K.* Network reliability and the factoring theorem // Networks. 1983. V. 13. P. 107-120.
7. *Rodionov, A.S, Rodionova, O.K., Choo, H.* On the Expected Value of a Number of Disconnected Pairs of Nodes in Unreliable Network. In: ICCSA 2007. LNCS, vol. 4707, Heidelberg: Springer, – 2007, P. 534–543.
8. *Chen Y., Li J., Chen J.* A new algorithm for network probabilistic connectivity // Military Communications Conference Proceedings, 1999, MILCOM 1999. IEEE. – 1999. – Vol. 2. – P. 920–923.
9. *Мигов Д.А.* Формулы для быстрого расчёта вероятности связности подмножества вершин в графах небольшой размерности // Проблемы информатики, № 2(6), 2010. – С.10-17.
10. *Rodionova, O.K., Rodionov, A.S., and Choo, H.* Network Probabilistic Connectivity: Exact Calculation with Use of Chains. ICCSA-2004, Springer LNCS. Vol. 3046 (2004) P. 315–324.
11. *Glinsky, B., Rodionov, A., Marchenko, M., Podkorytov, D., and Weins, D.* Scaling the Distributed Stochastic Simulation to Exaflop Supercomputers. High Performance Computing and Communication & 2012 IEEE 9th International Conference on Embedded Software and Systems (HPCC-ICESS), 2012 IEEE 14th International Conference on, Proceedings, P. 1131–1136.

Статья поступила в редакцию 01.09.2014.

Мигов Денис Александрович
к.ф.-м.н., научный сотрудник лаборатории моделирования динамических процессов в информационных сетях Института вычислительной математики и математической геофизики СО РАН (630090, Новосибирск, пр. Лаврентьева, д. 6), тел. (383) 332-69-49, e-mail: mdinka@rav.sscc.ru

Сакерин Алексей Викторович
аспирант Сибирского государственного университета телекоммуникаций и информатики (630102, Новосибирск, ул. Кирова, д. 86.), e-mail: bdlab.adm@gmail.com

Родионов Алексей Сергеевич
д.т.н., профессор кафедры вычислительных систем СибГУТИ (630102, Новосибирск, ул. Кирова, д. 86), тел. (383) 332-69-49, e-mail: alrod@sscc.ru

Some problems of exhaustive algorithm parallel realization (illustrated by network reliability analysis)

A. Rodionov, A. Sakerin, D. Migov

The problem of effective exhaustive algorithm paralleling used for calculation of some network reliability characteristics with unreliable links or/and nodes and for combinatorial optimization of such networks is examined. General approaches and specific algorithms enabling effective usage of computational resources when solving tasks of the classes under consideration are presented.

Keywords: parallel algorithm, network reliability, random graph, exhaustive search.