И. Бурдонов, А. Косачев.
Исследование графа взаимодействующими автоматами.
Новые информационные технологии в исследовании сложных структур. Материалы 10-ой российской конференции с международным участием. 2014, изд. Томского госуниверситета, стр.47-48.
2 стр.

Исследование графа взаимодействующими автоматами

И.Б. Бурдонов, А.С. Косачев

ИСП РАН, Москва, Россия

igor@ispras.ru, kos@ispras.ru

Исследование графов автоматами является корневой задачей во многих приложениях, к которым относятся верификация и тестирование программных и аппаратных систем, а также исследование сетей, в том числе сети интернета и GRID на основе формальных моделей. Модель системы, в конечном счёте, сводится к графу переходов, свойства которого нужно исследовать. За последние годы размер используемых систем и, следовательно, размер их моделей и, следовательно, размер исследуемых графов непрерывно растёт. Проблемы возникают тогда, когда исследование графа одним компьютером либо требует недопустимо большого времени, либо граф не помещается в его память, либо и то и другое. Поэтому возникает задача параллельного и распределённого исследования графов. Эта задача формализуется как задача исследования графа коллективом автоматов (несколькими параллельно работающими компьютерами с достаточной суммарной памятью).

В докладе подразумевается тестирование детерминированных систем: граф — это граф автомата тестируемой системы, автомат на графе — тестирующая система, а проход по дуге — это тестовое воздействие и наблюдение результата. В качестве практического примера можно привести функциональное тестирование различных подсистем модели процессора [1]: кэш третьего уровня, управление прерываниями и пр. Модельные графы содержат от нескольких тысяч до нескольких миллионов узлов и несколько миллионов дуг. Тест выполнялся максимально на 150 компьютерах.

Автомат-обходчик выполняется на одной машине (процессор с памятью), а наличие нескольких автоматов на разных машинах позволяет распараллелить работу. При тестировании клонирование тестируемой системы обычно возможно только в начальном состоянии. Поэтому автомат начинает работу с начальной вершины графа. Будем считать, что за один такт создаётся не более одного клона тестируемой системы.

Нижняя оценка времени обхода для одного или ограниченного числа автоматов равна $\Omega(nm)$, где n — число вершин графа, а m — число дуг. Если число автоматов не ограничено, нижняя оценка $\Omega(m)$.

Для того чтобы автомат мог обходить любой конечный граф, требуется неограниченная рабочая память, в которой накапливается информация о пройденной части графа. Если эта память — суммарная память автоматов, число которых ограничено, автоматы не конечны на классе всех графов. Существуют алгоритмы с оценкой $\Theta(nm)$. Распараллеливание ускоряет обход, но не меняет порядок времени обхода в наихудшем случае. Если число автоматов не ограничено, то существуют алгоритмы с оценкой $\Theta(m)$.

Проблема возникает, когда автоматы конечны, а граф слишком большой. Есть два подхода.

Первый подход применим, когда рабочая память существует отдельно от памяти автоматов и реализуется на вершинах графа: автомат может писать/читать в текущей вершине символы конечного алфавита. Такой подход может применяться, например, для сети интернета, когда вершина — это узел сети, а проход по дуге — передача сообщения между узлами. Для одного конечного автомата известен алгоритм с оценкой $\Theta(nm+n^2 loglogn)$ [2], а при повторном обходе $\Theta(nm+n^2 l(n))$, где $l(n)$ — число логарифмирований, при котором достигается соотношение $1 \le log(log...(n)...) < 2$ [3]. Отличие от нижней оценки $\Omega(nm)$ объясняется тем, что автомату бывает нужно «вернуться» в начало только что пройденной дуги. Если автоматов несколько, каждый из них может читать пометки в вершинах, оставленные другими автоматами, и обмениваться с ними сообщениями. Для двух автоматов оценка $\Theta(nm)$.

Второй подход применяется при тестировании, когда вершина графа — это состояние тестируемой системы, и автомат не может в неё писать. Рабочая память — это суммарная память коллектива конечных автоматов, обменивающихся сообщениями. Для k машин можно обходить графы в k раз большие, чем для одной машины. Если размер графа не ограничен, число автоматов в коллективе также должно быть не ограничено. В докладе предлагается алгоритм для расширенных автоматов (сводимых к конечным автоматам) с оценкой $\Theta(m+nD)$, где D – максимальная длина пути в графе от корня.

1. *Demakov A., Kamkin A., Sortov A.* High-Performance Testing: Parallelizing Functional Tests for Computer Systems Using Distributed Graph Exploration in Open Cirrus Summit. – Moscow, 2011.
2. *Бурдонов И.Б.* Обход неизвестного ориентированного графа конечным роботом // Программирование. – 2004. – №4. – С. 11-34.
3. *Бурдонов И.Б.* Проблема отката по дереву при обходе неизвестного ориентированного графа конечным роботом // Программирование. – 2004. – №6. – С. 6-29.