

Такая интерпретация георадиолокационных данных возможна только при наличии на волновой картине первого класса дефектов земляного полотна.

Таким образом, описанная модель реализует один из вариантов гибридного подхода к решению задачи автоматической интерпретации георадиолокационных данных на железнодорожном транспорте [6]. Идея использования комбинированного подхода является перспективным направлением в исследовании слабо формализованных процессов [7], каким является георадиолокационный процесс. В дальнейшем планируется опираться на анализ накопленной статистики по результатам работы модели, представленной в виде совокупности эмпирических данных, с использованием нескольких классов поддерживающих моделей. Важную роль при этом будут играть интеллектуальные модели, основанные на экспертных знаниях и моделировании человеческой системы обработки информации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Серебряников И.В.* Анализ состояния и основные проблемы усиления инженерных сооружений на сети дорог // Труды II-ой научн.-техн. конф. с международным участием «Современные проблемы проектирования, строительства и эксплуатации земляного полотна и искусственных сооружений». – М.: МИИТ, 2005. – С. 10.
2. *Помозов В.В., Семейкин Н.П.* Георадар как универсальный поисковый прибор // Специальная техника. – 2001, № 2. – С. 2-6.
3. *Долгий А.И., Ковалев С.М., Хатламаджиян А.Е.* Модель локализации неоднородностей в георадиолокационных данных на основе нейросетей Кохонена // Известия ТРТУ. Тематический выпуск «Интеллектуальные САПР». – Таганрог: Изд-во ТРТУ, 2006, №8. – 232 с.
4. *Вершок Д.А.* Алгоритмические средства обработки и анализа изображений на основе преобразования Хафа // Диссертация. Минск, 2002.
5. <http://ict.edu.ru/ft/002407/num1degt.pdf>
6. *Явна В.А., Грицьук В.И. и др.* Проведение измерений и интерпретация георадиолокационной съемки земляного полотна с выдачей рекомендаций для проектов его усиления / Методические указания. – Ростов-на-Дону: Изд-во РГУПС, 2004.
7. *Кулаичев А.П.* Методы и средства комплексного анализа данных. – М.: Форум–Инфра, 2006.

В.А. Литвиненко, С.А. Ховансков

АЛГОРИТМ ОПТИМИЗАЦИИ ПАРАМЕТРОВ КОМПЬЮТЕРНОЙ СЕТИ ДЛЯ УМЕНЬШЕНИЯ ВРЕМЕНИ РЕШЕНИЯ ЗАДАЧИ НА ОСНОВЕ МУЛЬТИАГЕНТНОЙ СИСТЕМЫ*

Существует ряд задач требующих огромных вычислительных затрат и имеющих ограничения по времени решения. Решения таких задач выполняют на специализированной вычислительной системе, состоящей из нескольких центров обработки информации [1].

Существует класс редко повторяемых задач. Создавать под каждую из них новую вычислительную систему не имеет смысла. Проще такие задачи решать на легкодоступном оборудовании. Наиболее приемлемым вариантом является использование для решения задач обычной компьютерной сети, состоящей из нескольких машин обменивающихся между собой пакетами данных по любой сетевой технологии [2]. Решение может быть осуществлено путем распределения ме-

* Работа выполнена при поддержке РФФИ (гранты № 07-01-00511, № 06-01-00272) и программ развития научного потенциала высшей школы 2006-2008 гг. (РНП.2.1.2.3193, РНП 2.1.2.2238).

жду несколькими компьютерами слабо связанных по данным вычислительных модулей задачи.

Для организации вычисления весь вычислительный процесс задачи делят на вычислительные модули и задача представляется в виде графа, где вычислительные модули – вершины графа, маршруты передачи данных – связи между вершинами. Граф компьютерной сети совмещают с графом задачи. Выбирают компьютер сети и ему (вершине графа сети), согласно графа задачи, назначают один вычислительный модуль. При этом наложение графа задачи на граф компьютерной сети, включающей различные компьютеры, носит случайный характер. Задача может быть решена на такой вычислительной системе, но время ее решения очень сильно зависит от полученного варианта случайного совмещения вершин графов задачи и сети, выполненного без учета производительности вершин графа сети и вычислительной сложности вершин графа задачи.

При такой организации вычисления возникает трудность, связанная с тем, что компьютеры используемые для решения задачи могут иметь производительности на порядок отличающиеся друг от друга, т.к. любая сеть представляет собой множество компьютеров с различным аппаратным обеспечением. Это может значительно увеличить время решения [2].

Созданная система может выполнить задачу и за приемлемое время, если структура системы на этапе подготовки к решению заранее оптимизирована по производительности за счет распределения вычислительных модулей между компьютерами с учетом их аппаратных особенностей и вычислительного объема вершин графа выполняемой задачи.

Для оптимизации используют выделенный компьютер, который в этом случае является администратором процесса оптимизации структуры системы и управляющим всей системой в процессе выполнения задачи. Такой вариант имеет существенный недостаток – управляющий узел должен значительное время занимать канал связи между компьютерами, увеличивая общее время решения задачи.

Предлагается другой подход, основанный на использовании быстро развивающегося в последнее время принципа коллективного принятия решения мультиагентной системой.

Под мультиагентной системой понимается множество независимых программных объектов – агентов, каждый из которых работает по одному и тому же алгоритму. Для ее функционирования достаточно обычной компьютерной сети. Каждый агент располагается в отдельном узле сети входящем в организуемую вычислительную систему. Результатом коллективного принятия решения является сокращение времени решения системой задачи.

В функции агента входит:

- ◆ контроль выполнения своего вычислительного модуля;
- ◆ выполнение обмена управляющей информацией – передачу данных другим агентам, имеющими с ним связи согласно связям графа задачи.

Существуют алгоритмы с использованием мультиагентной системы, которые выполняют оптимизацию на этапе подготовки системы к решению задачи. Оптимизация структуры для уменьшения времени выполнения осуществляется путем перебора пар агентов с целью переназначений между ними выполняемых модулей. При этом ведется поиск наиболее оптимального назначения каждому компьютеру сети своего вычислительного блока решения задачи [3].

Недостатком является увеличение общего времени решения. Требуемый при этом временной ресурс решение задачи оптимизации может увеличить на столько, что оно значительно превысит допустимый. Есть и еще немаловажный недостаток

— низкая степень живучести создаваемой вычислительной системы, поскольку при выходе из строя одного из центров обработки данных в ходе решение задачи приводит к остановке процесса решения.

На основе принципа принятия коллективного решения мультиагентной системой предлагается алгоритм оптимизации распределения вычислительных модулей между компьютерами системы позволяющий сократить время обработки данных при их прохождении через вычислительную систему. Его особенностью является то, что он позволяет выполнить оптимизацию конфигурации системы непосредственно во время процесса решения задачи.

Структура графа сложной задачи представляет собой совокупность подмножеств связанных вершин. Эти подмножества образуют структуру в несколько уровней.

Самый верхний уровень – вершина графа, получающая входные данные. Подмножества образуют узел самого высокого уровня и узлы нижнего уровня, которым передаются данные.

Другие подмножества образует каждая вершина нижнего уровня связанными с ней вершинами более нижнего уровня и т.д.

Самым нижним уровнем является вершина, передающая выходные данные.

Работа в сети предполагает использование разделяемой среды передачи данных. В ней каждый компьютер соединяется со всеми другими, поэтому граф передачи данных между компьютерами по графу задачи является условным. Эта среда используется для получения данных, передача результатов обработанных данных, обмена данными между агентами в процессе обработки данных.

Обмен данными включает в себя не только передачу обрабатываемых данных, но и служебную информацию.

Исходные данные приходят на обработку порциями (пакетами). Такими же порциями между агентами передается вся информация. Обработку одного пакета данных можно считать одним шагом процесса обработки данных всей сетью.

За счет передачи данных пакетами при обработке данные передаются от входного компьютера к выходному в пульсирующем режиме. В освободившейся от обработанных данных агент передается новая порция данных, идет следующий шаг обработки. Шаг это получение данных, обработка и передача результата следующему агенту.

На каждом шаге компьютер передающего агента посылает обработанные данные компьютеру агента находящегося на более высоком уровне и обменивается с ним служебной информацией. Потом получает от компьютера агента более нижнего уровня данные и обменивается с ним служебной информацией. Обмен служебной информацией производится между агентами расположенными только на этих компьютерах.

С точки зрения агента расположенного на компьютере параметром является время обработки данных за один шаг и время передачи обработанных данных другому агенту. Последний параметр очень сильно зависит не столько от компьютера, сколько от применяемой сетевой технологии.

Поэтому основным параметром является время обработки пакета данных на одном шаге. Время обработки данных в компьютере зависит от аппаратных параметров компьютера и свойств программного обеспечения, а также от характера обработки данных. Обычно оценку эти параметров выполняют либо в результате выполнения тестовой задачи, либо получают требуемый параметр приближенно, опираясь на информацию об аппаратной части компьютера.

Первый метод реализуется для задач имеющих упрощенный граф решения.

Второй метод дает усредненный результат, очень часто расходящийся с действительными значениями.

Критерием для оптимального назначения вычислительных блоков графа задачи обработки данных является минимальное время обработки одной порции данных всей системой.

При выполнении сложной задачи начало обработки данных в каждой вершине графа начинается только тогда, когда будет выполнена передача из всех связанных с ней агентов нижнего уровня обработанных данных.

Если это один агент, то обработка начинается сразу после получения данных и обмена служебной информацией, если их несколько, то обработка нового пакета начинается после приема данных от всех связанных с ним агентов. Тогда время начала обработки данных на компьютерах агентов текущего уровня будет считаться время приема последнего требуемого пакета данных и соответственно это будет время окончания обработки данных текущего шага $t^{n,i}$ на предыдущем нижнем уровне n . Аналогично на других уровнях.

На графе задачи можно отметить несколько путей передачи и обработки данных от входного компьютера до выходного. Из них можно выделить тот путь передачи данных от агента самого нижнего уровня к агенту самого высокого уровня определяет время обработки данных всей вычислительной системой. Такой путь называется критическим путем:

$$T_{кр} = \max(T_1, T_2, \dots, T_i, \dots, T_k),$$

где k - количество путей между входным агентом $a_{вх}$ и выходным агентом $a_{вых}$.

В предполагаемом алгоритме оптимизируется подбор назначения графа задачи на граф сети.

Эту задачу можно разбить на ряд задач минимизации времени обработки данных на каждом уровне графа задачи G .

Все множество вершин графа задачи разбивается на непересекающиеся подграфы $(g_1, g_2, g_3, \dots, g_j)$ путем их раскраски. Для раскраски графа используются функциональные признаки вычислительных блоков.

Агенты каждого подграфа обмениваются информацией с агентом следующего старшего уровня. Подграфы образуют иерархическую структуру и могут содержать разное количество агентов. Подграфы самого нижнего и самого верхнего уровней выполняют функции соответственно входного и выходного вычислительных блоков. Таким образом, они повторяют основную структуру графа задачи G с информационными потоками.

Для уменьшения времени оптимизации назначения вычислительных блоков графа задачи на компьютеры графа сети P процесс оптимизации распределяется между подграфами. Сначала оптимизация выполняется одновременно в каждом подграфе путем сокращения критического времени выполнения вычислительных блоков. Процесс выполняется во время обмена информацией между шагами обработки данных.

При этом время, затрачиваемое агентами подграфа на оптимизацию, не зависит от мощности подграфа, что позволяет отвести под процесс оптимизации во всех подграфах фиксированный промежуток времени.

При следующем обмене информацией между агентами соседних подграфов выполняется оптимизация между соседними подграфами, аналогичная оптимизации выполняемой в подграфах.

Эти процессы оптимизации между агентами в подграфе многократно чередуются с оптимизацией между агентами в соседних подграфах. Они повторяются, поскольку по алгоритму процесс должен выполняться для каждой передачи данных между агентами вычислительных блоков на фоне выполняемой задачи. Время, затрачиваемое на оптимизацию, сводится к минимуму.

Для оптимизации при обмене информацией между агентами соседних подграфов выполняется передача данных об оценках производительности компьютеров соседних подграфов. В эту информацию включаются все собранные данные о компьютерах других подграфов, что позволяет ускорить и улучшить процесс оптимизации. Через несколько шагов после начала решения задачи каждый агент будет иметь информацию о всех компьютерах вычислительной сети. Количество необходимых шагов определяется формулой $O * 2 + 1$, где O – количество уровней иерархической структуры графа G .

Это количество шагов достаточно для поиска оптимального варианта назначения графа задачи на граф сети и уменьшения времени критического пути.

Алгоритм агента.

1. Прием информации от агентов уровня $n - 1$.
2. Прием служебной информации от агентов уровня $n + 1$.
3. Определение критического пути.
4. Подбор варианта обмена для агента на $T_{кр}$.
5. Все варианты обмена использованы? Если «Да», то перейти к п.7, в противном случае перейти к п.6.
6. Передача служебной информации агентам уровня $n - 1$.
7. Обработка данных.
8. Обработка данных закончена? Если «Да», то перейти к п.9, если «нет», то перейти к п.7
9. Передача всей информации агентам уровня $i + 1$.
10. Переход к п.1.

Повторение этого алгоритма оптимизации соответствия вершин графа задачи G компьютерам сети под управлением мультиагентной системой позволяет выполнять оптимизацию при любом первоначальном назначении графа G на сеть P . Алгоритм не требует предварительной оптимизации назначения. Оптимизация выполняется во время решения самой задачи, используя для этого реальные данные.

Обмен информацией об оценках производительности компьютеров продолжается на всем протяжении решения основной задачи. Это позволяет повысить живучесть вычислительной системы, поскольку при выходе из строя компьютеров оптимизированной вычислительной системы, освободившиеся вычислительные блоки будут распределены между работоспособными компьютерами, а разработанный алгоритм оптимизации оптимизирует вычислительную систему с новой конфигурацией

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Лорин Г. Распределенные вычислительные системы. – М.: Радио и связь, 1984. – 296 с.
2. Таненбаум Э. Распределенные системы: принципы и парадигмы. – СПб: Питер, 2003. – 877 с.
3. Ховансков С.А., Мельник Э.В., Блушвили И.В. Метод организации распределенных вычислений в управляющих системах. Мехатроника, автоматизация, управление // Новые технологии. – 2003, №4.