Разработка программного комплекса для конструирования программ обработки данных на высокопроизводительных вычислительных системах

А.Б. Купчишин, В.Г. Сарычев

Новосибирский государственный технический университет, Новосибирск

Представлены проект и прототип программного комплекса, предназначенного для визуального автоматизированного конструирования программ обработки сейсмических данных, параллельная реализация алгоритма когерентного суммирования для поиска эпицентра микросейсмической активности, полученные результаты эффективности распараллеливания.

Введение

Комплекс Madagaskar[1] представляет собой коллекцию процедур для обработки геофизических данных и язык сценариев для конструирования схем вычислений из таких процедур. Коллекция процедур расширяется за счет усилий различных авторов, которым требуются те или иные операции с данными, и качество реализации таких процедур различно. В некотором смысле Madagaskar – это набор спецификаций и реализаций-примеров тех процедур, которые часто требуются для решения практических задач, и актуальной работой является повышения качества реализаций наиболее востребованных процедур для современных вычислительных систем. С повышением количества процедур и с необходимостью делать схемы обработки данных относительно большими возрастает сложность конструирования таких схем посредством языка сценариев. Необходимо создание инструмента для высокоуровневого конструирования программ обработки данных из процедур, позволяющего сделать конструирование более наглядным и автоматизировать выбор реализаций процедур под особенности конкретной вычислительной системы.

В настоящей работе представлен проект и прототип программного комплекса, предназначенного для решения задач обработки сейсмических данных, параллельная реализация алгоритма

[1] Madagaskar, http://www.ahay.org/wiki/Main_Page

когерентного суммирования для поиска эпицентра микросейсмической активности

Архитектура программного комплекса

Программный комплекс, призванный облегчить написание управляющей программы (сценария), задача которой сводится к запуску библиотечных подпрограмм в нужном порядке, включает в себя:

- визуальный конструктор программ,
- расширяемую библиотеку процедур обработки данных, характерных для геофизических задач,
- интерпретатор сценариев.

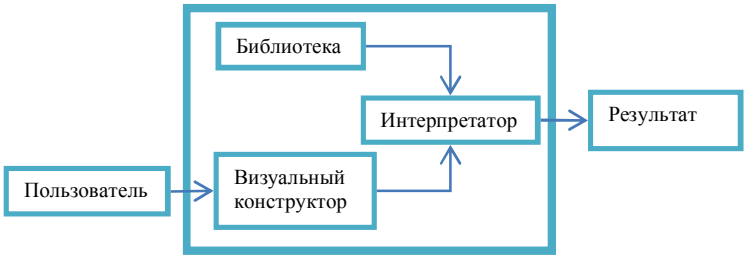

Рис. 1. Схема программного комплекса

Визуальный конструктор включает в себя:

1) инструменты визуального конструирования схем обработки данных;
2) модуль проверки корректности связей, который выполняет проверки на соответствие связываемых объектов, типов входных и выходных данных;
3) модуль кодогенерации, создающий для визуального сконструированного сценария его текстовое представление (скрипт-файл) в формате, понятном для интерпретатора.

Интерпретатор – это модуль, который исполняет сконструированную программу, используя при этом предоставленные библиотекой операции и процедуры.

Визуальный конструктор программ

Пользователь при конструировании программы оперирует тремя объектами: переменными, операциями, стрелками (см. пример на

рис. 2). Переменная – это ячейка памяти, характеризуется типом данных и содержит некоторые значения, может выступать в качестве параметра для операций. Операция – процедура, которая характеризуется набором входных и выходных данных и выполняет преобразование одного в другое. Стрелки устанавливают одностороннюю связь между объектами (операция-операция, переменная-операция), обозначающую направление передачи данных.

Рис. 2. Пример сконструированной схемы программы

Переменные представлены прямоугольниками var, операции – прямоугольниками operation, стрелки отражают направление потоков данных и аттрибутированы типом данных.

Схема на рис. 2 представляет собой цепочку из трех операций, последовательно передающих свой результат на вход последующей. Результат последней операции посылается выходной переменной, а значения входных переменных передаются первой операции.

Результаты отработавшей операции могут передаваться сразу нескольким соответствующим объектам, создавая потенциал для одновременного исполнения множества операций. Типы данных определяются предметной областью, и возможные для использования типы поставляются библиотекой процедур (в геофизике, например, приняты определенные форматы для представления данных, собранных с датчиков, которыми и могут характеризоваться уже конкретные типы данных).

Параллельная реализация метода когерентного суммирования

При добыче углеводородов методом гидроразрыва пласта требуется находить координаты разрывов среды. В момент порождения разрывы являются источниками (эпицентрами) микросейсмической активности. Для их локализации используется метод когерентного суммирования [1].

Модель задачи

Чтобы определить координаты эпицентра микросейсмической активности, на поверхности исследуемого объема земли раскладывают

датчики, улавливающие распространяющееся возмущение. Время, за которое волна достигает различных датчиков, неодинаково. Датчики делают замеры через промежутки времени. Каждый такой замер являет собой картину состояний датчиков на определенный момент времени. За промежуток времени набирается набор состояний для различных моментов времени, который будем называть разверткой по времени.

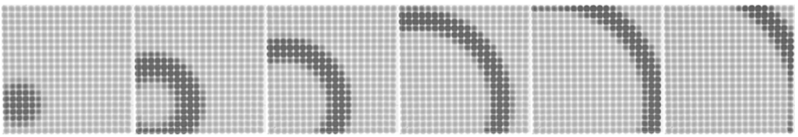

Рис. 3. Регистрация датчиками распространения волны в идеальных условиях без шумов

Датчики, помимо возмущения от разрыва, улавливают еще и шум, который сопоставим по силе с самим возмущением. Поэтому наблюдаемая картина соответствует рис. 4.

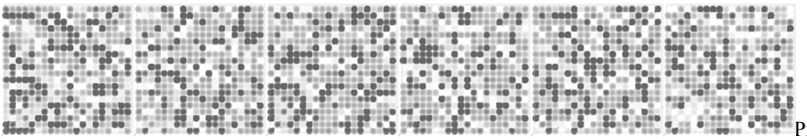

Рис. 4. Состояния датчиков с учетом шума

Когерентное суммирование

На рис. 5 представлен срез куба земли, треугольниками обозначены датчики, звездочкой – эпицентр, справа – развертка по времени. Каждая из вертикальных линий отображает временную развертку состояния датчика. Направление времени указано стрелкой. Пики на линиях отмечают время прихода сигнала до соответствующего датчика. Метод когерентного суммирования предполагает следующую процедуру:

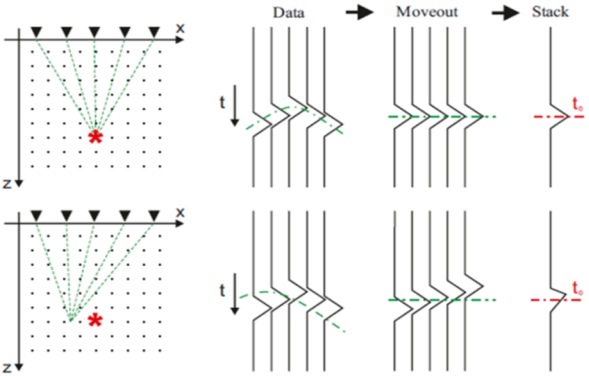

Рис. 5. Обнаружение эпицентров сейсмической активности с помощью метода когерентного суммирования

1) в кубе среды строится сетка;
2) для каждого узла:
   a) рассчитывается время распространения сигнала из этого узла до датчиков. Таким образом, получается некоторый профиль волны, которая могла бы распространяться из этого узла, если бы эпицентр был в нем;
   b) выбирается датчик, до которого время прихода волны минимально, берется значение состояния датчика в начальный момент времени (первое состояние с начала замеров), состояния других датчиков выбираются в моменты времени, соответствующие приходу волны до этих датчиков. Все выбранные состояния суммируются. Далее фронт сдвигается на следующий шаг по времени и повторяется процедура суммирования. Так продолжается до конца таблицы измерений. Из всех сумм выбирается сумма с максимальным значением;
3) из всех максимальных для каждого узла сумм выбирается максимальная. Узел, соответствующий этой сумме, считается наиболее близким к эпицентру.

Реализация

Реализован параллельный алгоритм когерентного суммирования, сочетающий в себе следующие особенности:

- параллелизм в распределенной памяти – исходные данные являются разверткой по времени, которую мы можем поделить на некоторое

количество отрезков, назначив каждому узлу свой отрезок, который он будет обрабатывать;

- параллелизм в общей памяти – внутри вычислительного узла потоки параллельно вычисляют суммы для разных узлов пространственной сетки;
- двойная буферизация – подгрузка данных в память на фоне обработки ранее загруженных данных;
- алгоритм уточнения исследуемой области – позволяет сократить количество обрабатываемых точек за счет поэтапного сужения исследуемой области локализации эпицентра.

На рис. 6 представлена схема распараллеливания в распределённой памяти, таблица состояний датчиков делится поровну на отрезки времени между процессами, и каждый поток ищет эпицентр на ограниченном отрезке времени, такой эпицентр будем называть локальным.

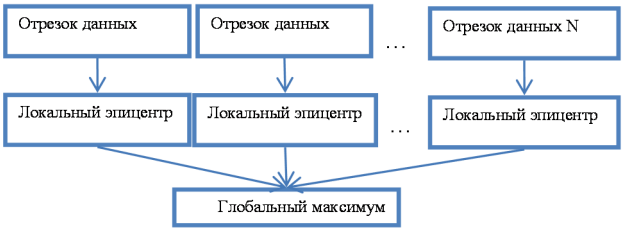

Рис. 6. Схема распараллеливания в распределенной памяти

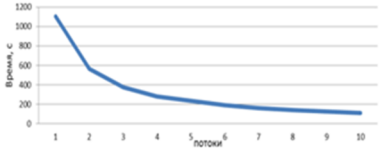

Рис. 7. Зависимость времени от количества потоков

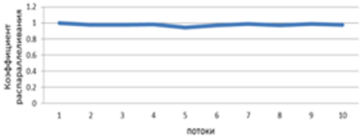

Рис. 8. Эффективность распараллеливания

При таком распределении нагрузки потоки работают с минимальными накладными расходами и независимо друг от друга, чем и обусловлены близкие к идеальным показатели эффективности распараллеливания (рис. 7, 8).

На рис. 9 представлена схема распараллеливания в общей памяти, потоки распределяют между собой узлы сетки, для которых проводится суммирование, т.е. делят исследуемое пространство.

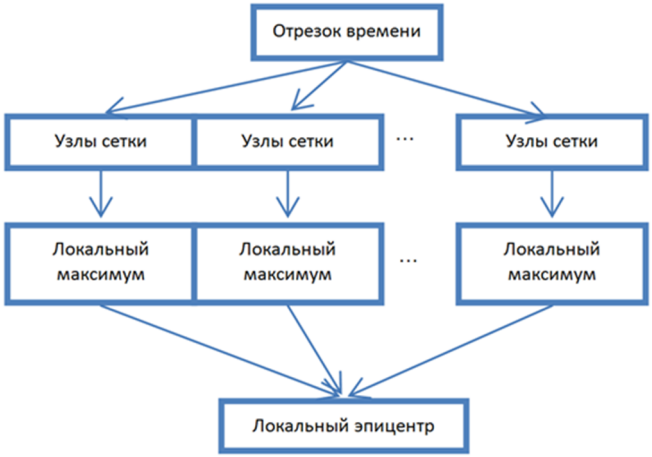

Рис.9. Схема распараллеливания в общей памяти

Внутри каждого потока ищется узел с максимальным результатом суммирования, который будем называть его локальным максимумом.

При таком распределении нагрузки потоки выполняются независимо друг от друга, т.е. отсутствует простой одних вычислительных устройств в ожидании завершения работы других, чем и обусловлены близкие к идеальным показатели эффективности распараллеливания (рис.10, 11).

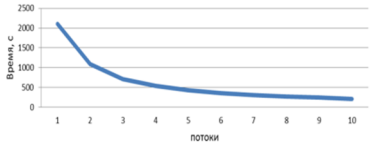

Рис.10. Зависимость времени от количества потоков

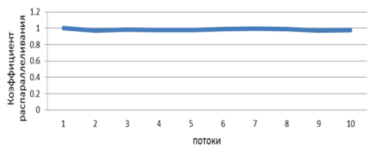

Рис.11. Эффективность распараллеливания

Для обработки больших объемов данных, которыми характеризуется задача, используется предварительная подгрузка данных в память на фоне обработки ранее загруженных данных (рис. 12).

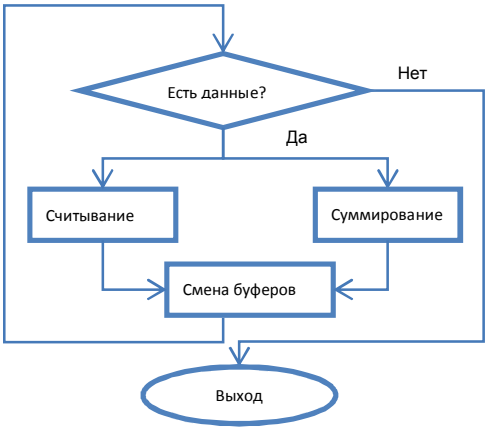

Рис.12. Схема подгрузки данных

Для организации подгрузки используются два буфера данных.

Порождаются два потока. Пока один проводит суммирование (используя первый буфер), второй тем временем готовит новую порцию данных (заполняя второй буфер).

После того как данные обработаны, буферы меняются местами.

Уточняющий алгоритм:

1) разбиваем исследуемую область на подобласти;
2) узнаем результат суммирования для центра подобластей;
3) выбираем ту подобласть, в которой значение оказалось максимальным;
4) теперь так же исследуем полученную подобласть;
5) делаем так до тех пор, пока исследуемая область не станет достаточно малой;

Ускорение достигается за счет непосредственного уменьшения числа обсчитываемых точек.

Заключение

Разработана архитектура и реализован прототип программного комплекса для визуального конструирования программ обработки геофизических данных, который позволяет конструировать

линейные цепочки процедур обработки данных. Реализован набор операций, решающий задачу поиска эпицентра микросейсмической активности методом когерентного суммирования. Реализован параллельный алгоритм когерентного суммирования.

Дальнейшая работа предполагает наполнение библиотеки эффективными реализациями процедур, специфицированных в Madagaskar, реализацию возможности конструировать нелинейные схемы программ, интеграцию инструмента визуального конструирования в среду HPC Community Cloud [2,3].

Литература

1. *Колесников Ю.И., Хогоев Е.А., Полозов С.В., Донцов М.В.* Применение сейсмоэмиссионной томографии для локализации сейсмических источников // Сборник докладов Международной конференции, посвященной 90-летию академика Пузырева Н.Н. «Сейсмические исследования Земной коры». Новосибирск, 2004, С. 129–134.
2. *Городничев М.А., Малышкин В.Э., Медведев Ю.Г.* HPC Community cloud: эффективная организация работы научно-образовательных суперкомпьютерных центров // Научный вестник НГТУ. 2013. №3(52). С. 91–96.
3. *Вайцель М. А. Городничев М.А.* HPC Community Cloud: разработка инструментария для повышения уровня взаимодействия пользователей с объединенными HPC-системами // Седьмая Сибирская конференция по параллельным и высокопроизводительным вычислениям. Программа и тезисы докладов. Томск: Изд-во Том. ун-та, 2013. С. 82–84.