ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ ФОРМИРОВАНИЯ OLAP-КУБА НА ПРОЦЕССОРАХ CELL

В.А. Смирнов

В настоящее время одной из актуальных задач в области технологий *оперативного анализа данных (On-Line Analytical Processing)* значится построение эффективных алгоритмов вычисления OLAP-куба. *OLAP-куб* является многомерным представлением реляционной таблицы фактов предметной области.

В данном представлении осями координат являются соответствующие колонки таблицы, в качестве значений координат оси выступает множество всех возможных значений соответствующей колонки, дополненное специальной координатой ALL. Точками OLAP-куба являются значения соответствующих точек таблицы, причем значение по координате ALL вычисляется с помощью специфицированной пользователем агрегатной функции (сумма, среднее и др.). Пример таблицы фактов и соответствующего OLAP-куба приведен на Рис. 1.

Измерения Мера

Ученики	Предметы	Время	Оценка
Иванов	Алгебра	I четверть	3
Петров	История	III четверть	5
Иванов	Алгебра	IV четверть	4
Иванов	История	III четверть	3
Петров	Физика	II четверть	4

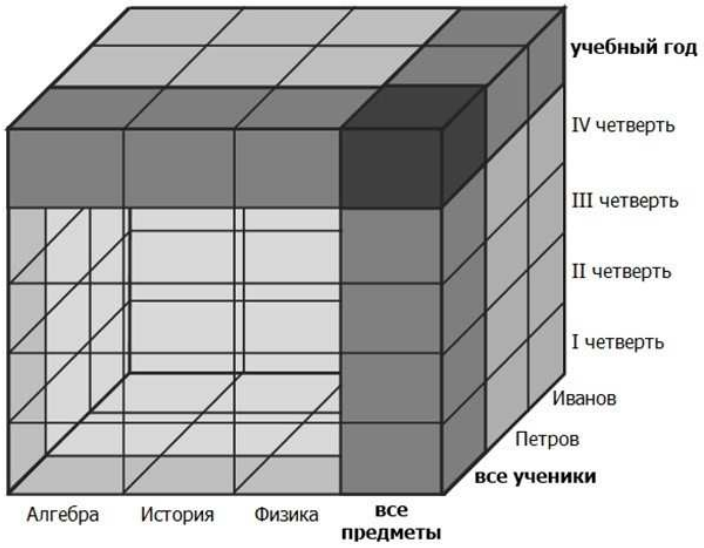


Рис. 1. Таблица фактов и OLAP-куб

В данной работе представлены результаты проекта разработки параллельного алгоритма вычисления OLAP-куба на процессорах Cell.

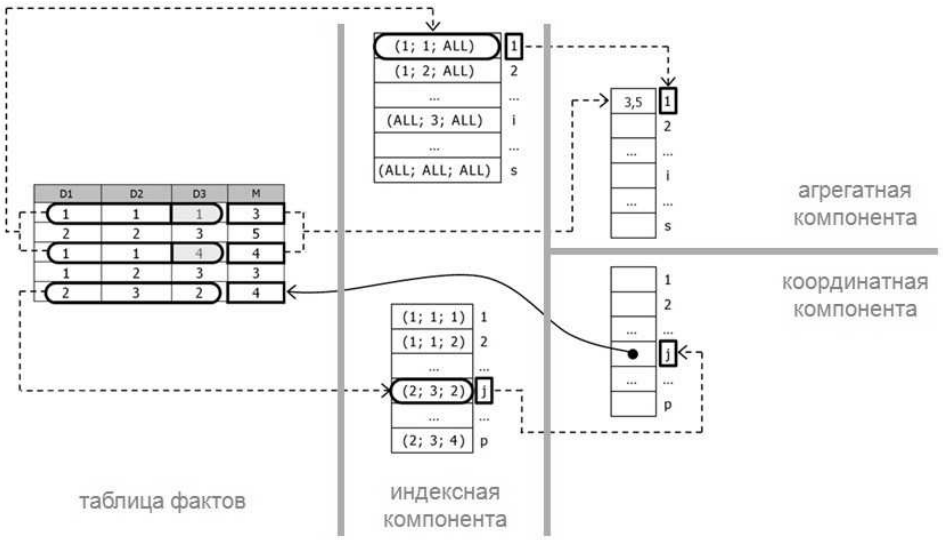


Рис. 2. Структура и схема формирования OLAP-куба

OLAP-куб реализуется в виде одномерной структуры со следующими основными полями: координатная компонента, агрегатная компонента и индексная компонента. Координатная компонента представляет собой массив указателей на ячейки таблицы фактов и не требует вычисления. Агрегатная компонента является массивом значений по координате ALL, при вычислении которых по существу используются векторные функции процессора Cell. Индексная компонента используется в вычислении

агрегатной компоненты и заполнении координатной компоненты и представлена двумя массивами, элементами которых являются многомерные индексы элементов координатной и агрегатной компонент соответственно (см. Рис. 2).

Кратко алгоритм формирования OLAP-куба, представленный на Рис. 2 может быть описан следующим образом.

На первом этапе на основе данных об измерениях формируется индексная компонента, разделённая на два массива: один для адресации имеющихся данных, второй для адресации агрегированных данных.

На втором этапе с помощью первой части индексной компоненты выполняется заполнение координатной компоненты. Поскольку и куб, и таблица фактов находятся в оперативной памяти, в координатную компоненту записываются не значения мер, а указатели на них. В соответствии с каждым индексом из соответствующей части индексной компоненты куба, в таблице фактов выполняется поиск значений мер, а затем указатель на набор этих значений записывается в соответствующую этому индексу ячейку куба. Если для какого-то индекса не найдено значений мер, то в координатную компоненту помещается пустой указатель (NULL).

На третьем этапе с помощью второй части индексной компоненты выполняется вычисление и заполнение агрегатной компоненты куба. Для этого создается маска по каждому индексу с координатой ALL. Затем осуществляется сканирование таблицы фактов, во время которого каждый многомерный индекс сравнивается с маской. В случае совпадения значения меры записываются в один массив. Далее эти значения агрегируются и записываются в соответствующую ячейку агрегатной компоненты.

В соответствии с архитектурой Cell Broadband Engine (Cell BE), процессор Cell представляет собой асимметричный многоядерный процессор, состоящий из одного управляющего ядра (Power Processing Element, PPE) и восьми вычислительных ядер (Synergistic Processing Element, SPE), которые поддерживают набор векторных инструкций, имеют прямой доступ к основной памяти и оперируют 128-битными векторами.

Распараллеливание описанного выше алгоритма для процессора Cell происходит по данным. Каждое вычислительное ядро будет выполнять агрегирование по заданному набору многомерных индексов с координатами ALL из индексной компоненты. Результаты агрегирования будут записываться в агрегатную компоненту. Агрегирование значений мер будет проводиться ядрами SPE с помощью операций векторного сложения.

Схема параллельного формирования агрегатной компоненты следующая. Пусть I2 - это часть индексной компоненты, содержащая индексы с координатами ALL, общее число таких индексов - s, а вычислительных ядер SPE - N. Сначала на управляющем ядре выполняется разбиение I2 на N равных частей (если невозможно разбить на равные части, N-я часть меньше остальных). Затем эти части отправляются на соответствующие вычислительные ядра. Там происходит формирование соответствующих частей агрегатной компоненты. Затем сформированные части отправляются на управляющее ядро, где собираются в агрегатную компоненту.

Рассмотрим схему выполнения операции агрегирования на примере суммирования. Пусть имеется массив, содержащий значения мер, которые необходимо агрегировать с помощью операций над четырехэлементными векторами. До тех пор, пока количество элементов в данном массиве больше четырех, будем выполнять следующую последовательность действий. Дописываем в массив столько нулей, чтобы его длина представляла собой 2^n (n > 2). Затем, начиная с первого элемента, разбиваем массив на вектора и попарно складываем с помощью векторной операции сложения (spu_add). Результаты сложения, представляющие собой вектора той же размерности, записываются в начало исходного массива, перекрывая прежние значения. Длина массива хранится отдельно в виде переменной, значение которой уменьшается вдвое после каждой серии сложений. Это позволяет не задействовать в вычислениях оставшиеся прежние значения. После того, как длина массива становится не больше четырех, его элементы складываются скалярно. Результат

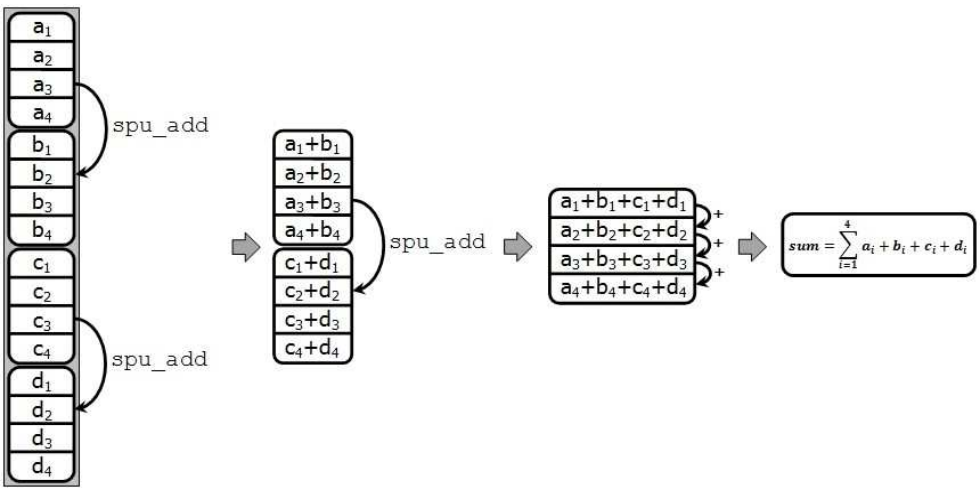
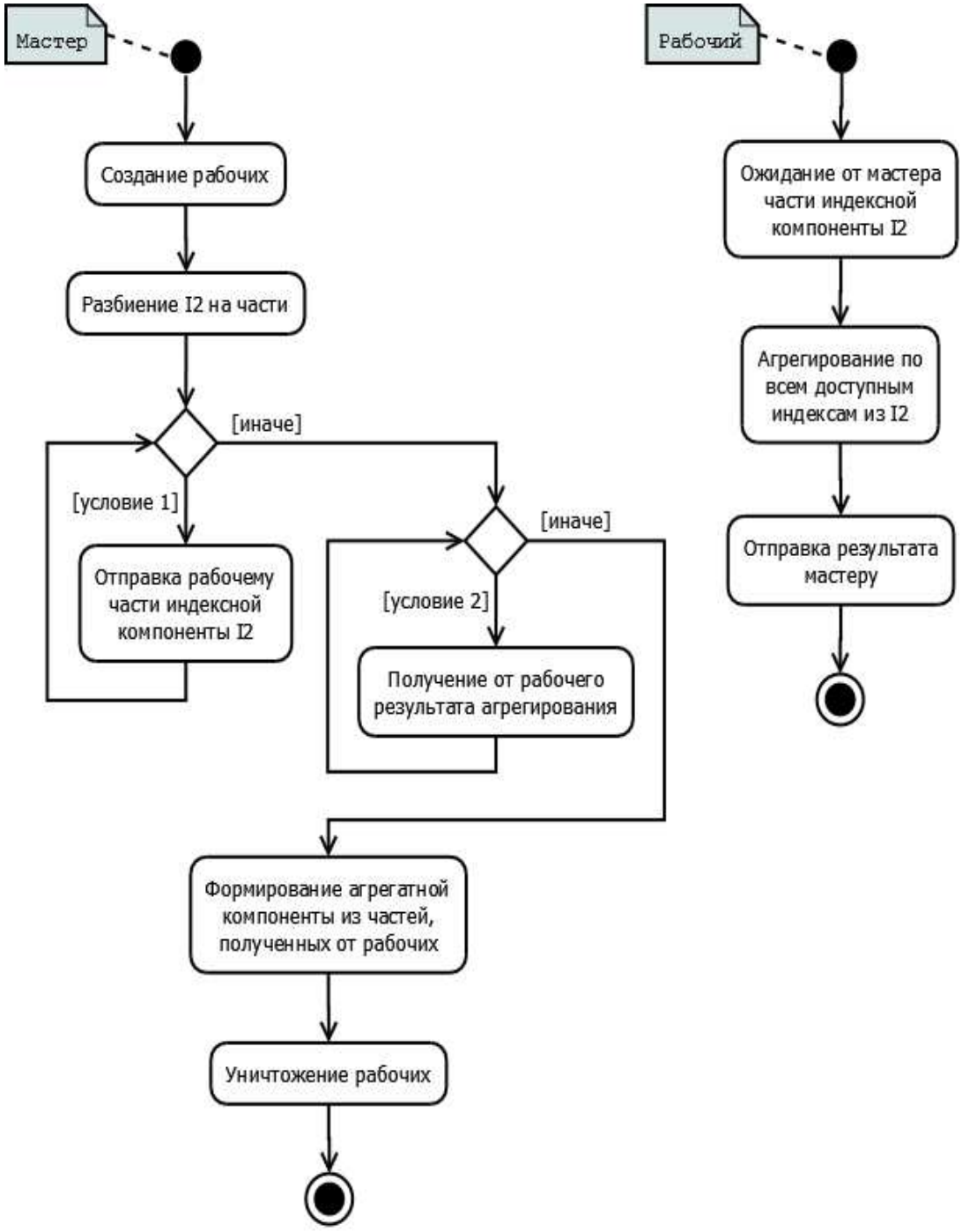


Рис. 3. Схема сложения с использованием векторов

данного сложения будет результатом суммирования элементов исходного массива. Схема сложения представлена на Рис. 3.

В данном алгоритме используется модель «мастер-рабочие». Нить-мастер запускается на управляющем ядре PPE и распределяет задания для рабочих. Нити-рабочие запускаются на вычислительных ядрах SPE и выполняют обработку данных, получаемых от мастера.

Деятельность мастера кратко может быть описана следующим образом. Сперва он создаёт рабочих и разбивает индексную компоненту I2 на равные части, количество которых равно количеству рабочих. Затем он отправляет полученные на предыдущем шаге части рабочим и ожидает результатов агрегирования. После получения результатов, мастер формирует из них агрегатную компоненту, а затем уничтожает рабочих (см. Рис. 4).



условие 1: имеются свободные рабочие
условие 2: не все результаты получены

Рис. 4. Диаграмма деятельности мастера и рабочего

Деятельность рабочего заключается в следующем. Он получает от мастера задание в виде части индексной компоненты I2, выполняет агрегирование по каждому многомерному индексу, который она содержит, и отправляет мастеру результат в виде части агрегатной компоненты (см. Рис. 4).

Результаты экспериментов позволили определить *масштабируемость* алгоритма. Соответствующий график представлен на Рис. 6.

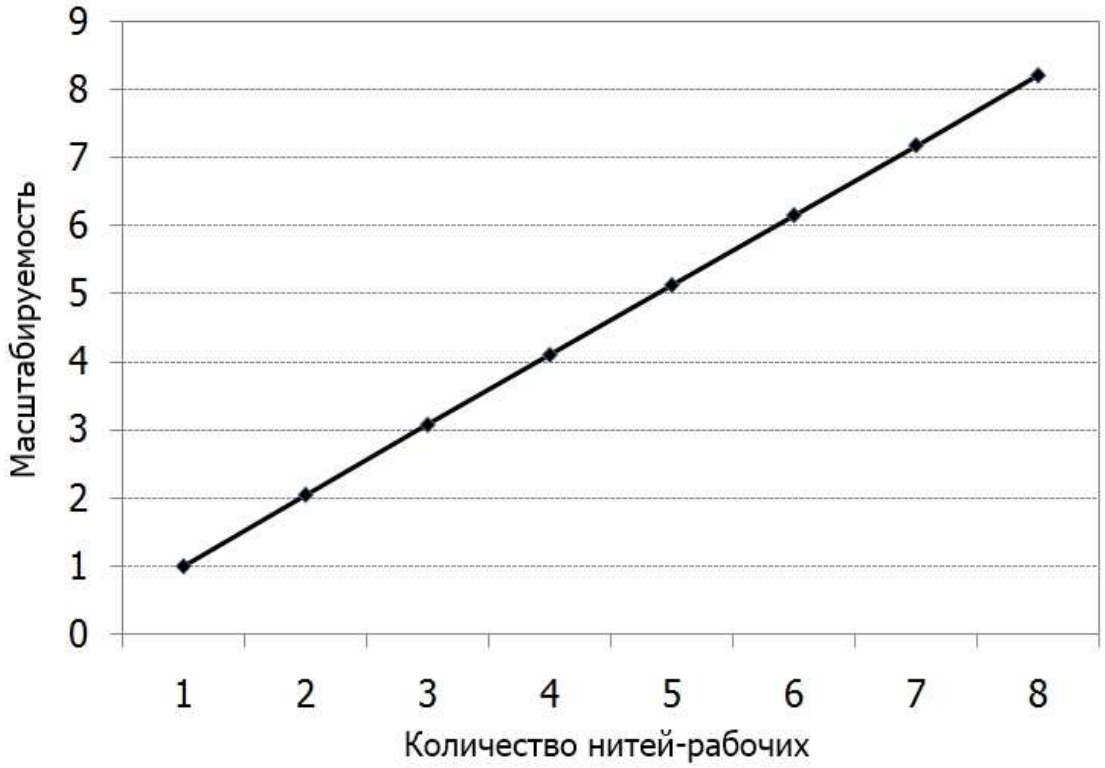


Рис. 5. Масштабируемость

Результатом данной работы является программная система, реализующая стоимостно-оптимальный алгоритм формирования OLAP-куба, оптимизированный для выполнения на процессорах Cell.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (проект 09-07-00241-а).