

РАСПРЕДЕЛЕННАЯ ОБРАБОТКА ИЗОБРАЖЕНИЙ НА ОСНОВЕ ДИНАМИЧЕСКОЙ ОРГАНИЗАЦИИ ДАННЫХ

С.Б. Попов

При рассмотрении проблемы отображения задач обработки изображений на архитектуру распределенных вычислительных систем с различными типами параллелизма выделяются следующие основные подходы к организации программного обеспечения параллельной обработки изображений [1]: параллельная обработка на многоядерных процессорах в рамках модели общей памяти; параллельная обработка с использованием гетерогенных многопроцессорных вычислительных систем (CPU+GPU); параллельная обработка на распределенных вычислительных системах, в том числе кластерных.

При обработке крупноформатных изображений использование первых двух вариантов архитектуры влечет за собой необходимость решения проблем распараллеливания операций ввода/вывода изображений и ограниченности размера оперативной памяти системы.

Для параллельных систем обработки изображений узким местом является последовательный доступ к централизованно хранящимся данным крупноформатных изображений.

Централизованный доступ к данным в значительной степени связан с тем, что вычисления в большинстве существующих систем параллельной обработки организованы с использованием программы-менеджера, которая выполняет декомпозицию изображений непосредственно перед обработкой, причем в случае достаточно популярной процедуры локальной обработки скользящим окном декомпозиция осуществляется с перекрытием. Размер перекрытия обрабатываемых фрагментов выбирается в соответствии с размером окна обработки. При выполнении последовательности таких локальных операций обработки возникают дополнительные затраты на коммуникацию, необходимые для согласования перекрывающихся данных. Положительным моментом использования централизованного менеджера является возможность балансировки вычислений при параллельной обработке.

В докладе предлагается альтернативный подход, основанный на концепции распределенного изображения, которая обеспечивает распараллеливание операций доступа к данным, используя принцип "обрабатываем там, где храним". При реализации этой концепции декомпозиция выполняется при создании или импорте изображения в систему и не изменяется в процессе хранения. Это порождает целый ряд проблем: выбор оптимальной декомпозиции (разбиения на фрагменты) изображения в условиях отсутствия априорной информации о параметрах, запускаемых в системе задач обработки, проблемы, связанные с обеспечением отказоустойчивости распределенного хранения фрагментов изображений, сбалансированности загрузки компьютеров, участвующих в обработке при заранее выполненной декомпозиции данных, интерактивности системы при визуализации распределенных изображений.

В работе [2] предложен комплексный подход к решению этих проблем на основе декомпозиция распределенного изображения в виде перекрывающихся фрагментов, в соответствии с которым размер необходимого перекрытия фрагментов определяется не параметрами последующей задачи обработки, поскольку она априори неизвестна, а необходимостью решения проблем отказоустойчивости распределенного хранения фрагментов изображений и сбалансированности загрузки компьютеров.

Суть подхода на примере одномерной декомпозиции заключается в следующем: выполняется предварительная декомпозиция на удвоенное (относительно количества узлов хранения/обработки) количество непересекающихся блоков, каждый узел распределенной системы хранит и обрабатывает два собственных блока предварительной декомпозиции, а также дополнительно хранит и, при необходимости использует при обработке, по одному ближайшему блоку от каждого соседнего узла.

Такая структура данных распределенного изображения позволяет восстановить его при отказе одного из узлов хранения, а также обеспечивает возможность формирования отсчетов основных блоков нового изображения без передачи данных между соседними узлами при многоэтапном выполнении локальных операций в процессе распределенной обработки изображения.

Предложенный способ организации распределенных изображений по построению обеспечивает отказоустойчивость за счет использования полного перекрытия областей при формировании фрагментов изображений.

В докладе описываются принципы динамической организации хранения распределенных изображений, которые обеспечивают размещение данных, минимизирующее время решения задач обработки изображений.

Распределенное изображение в системе может быть создано двумя способами: импорт файла изображения или набора файлов изображений, составляющих панораму; создание нового изображения в процессе обработки существующего распределенного изображения.

При импорте изображения выполняется разбиение изображения на фрагменты в соответствии с предложенным методом формирования децентрализованных структур данных изображений на основе

декомпозиции в виде перекрывающихся фрагментов. Размещение полученных таким образом фрагментов изображения выполняется на доступных узлах хранения в соответствии с выбранными критериями.

Данные распределенного изображения после его создания не изменяются, при обработке создается новое распределенное изображение. Фрагменты нового изображения, создаваемого при обработке существующего распределенного изображения, размещаются на тех же узлах, где размещались фрагменты исходного изображения.

В процессе работы системы фрагменты не перемещаются. Фрагменты только дублируются или удаляются. Фрагменты дублируются, во-первых, если при доступе к распределенному изображению пользователя, у которого на компьютере еще нет ни одного фрагмента данного изображения. Пользователю дублируется тот фрагмент, время доступа к которому с компьютера пользователя максимально. Если разброс времени доступа ко всем фрагментам лежит в допустимом диапазоне (это параметр системы), то на компьютер пользователя дублируется тот фрагмент, который имеет минимальный уровень репликации в системе, в противном случае – дублируемый фрагмент выбирается случайно. Во-вторых, фрагменты дублируются при отказе (тайм-ауте) доступа к узлу хранения. При этом формируется копия недоступного фрагмента по данным тех узлов хранения, которые хранят теневые копии его основных блоков; новое размещение выбирается из числа доступных узлов в системе, имеющих минимальную загрузку предоставленной системе области хранения данных.

Фрагменты удаляются в процессе оптимизации системы хранения или по команде пользователя. Удаление выполняется если фрагмент имеет уровень репликации выше, чем заданный уровень репликации в системе и время последнего доступа к этому фрагменту превышает заданное время. Когда пользователь дает команду на удаление распределенного изображения, с которым работал только этот пользователь, то все фрагменты изображения удаляются из системы, если изображение использовалось и другими пользователями, то удаляется (или перемещается на другой узел, в зависимости от уровня репликации фрагмента) только фрагмент, размещенный на компьютере пользователя, издавшего запрос.

Соответственно, на компьютере пользователя хранится по одному фрагменту каждого распределенного изображения, с которыми он работает. Если пользователь длительное время не использовал какое-либо изображение, т.е. не обрабатывал и не просматривал его, то данный фрагмент распределенного изображения может быть удален с компьютера пользователя с тем, чтобы оптимизировать размещение этого распределенного изображения в системе.

Метод декомпозиции распределенного изображения является основой для оригинального алгоритма динамического распределения нагрузки процессоров при выполнении локальной обработки. Данный алгоритм в варианте с равномерным начальным распределением данных по узлам обработки описан в работе [2]. В докладе рассматривается модификация алгоритма динамической балансировки, при которой начальная декомпозиция выполняется с использованием статической балансировки на основе априорной информации о производительности узлов хранения/обработки. Приводятся результаты экспериментальных исследований децентрализованного алгоритма динамического распределения нагрузки процессоров при выполнении локальной обработки изображений на вычислительных узлах (14 вычислительных узлов по 2 CPU: Quad-Core Intel Xeon 5430 2.33 GHz, cache 12 Mb, 1333MHz; RAM: 8Gb DDR2-667; HDD: 120Gb) и 1-го сервера HP ProLiant BL260c в качестве управляющего узла. Вычислительные эксперименты показали, что при возрастании вычислительной нагрузки вдвое на 7 узлах (из 14) использование предлагаемой динамической балансировки позволяет обеспечивать время решения модельной задачи, в целом, пропорциональное общей нагрузке кластера, однако время решения зависит от взаимного расположения нагруженных узлов. Если нагруженные узлы распределены в системе таким образом, что они чередуются с менее загруженными узлами, то время обработки соответствует времени, получаемому при статической балансировке. Если нагруженные узлы образуют "кластеры" размером более трех, обработка становится менее сбалансированной, и время выполнения увеличивается.

Таким образом, если в системе распределенного хранения и обработки изображений статистика ее использования показывает, что сформировались кластеры нагруженных узлов, то необходимо выполнить перераспределение хранимых фрагментов изображений так, чтобы нагруженные узлы чередовались с менее загруженными.

Работа выполнена при поддержке программы фундаментальных исследований Президиума РАН «Проблемы создания национальной научной распределенной информационно-вычислительной среды на основе развития GRID технологий и современных телекоммуникационных сетей», гранта Президента РФ для ведущих научных школ № НШ-7414.2010.9 и гранта Российского фонда фундаментальных исследований № 10-07-00553.

ЛИТЕРАТУРА:

1. С.Б. Попов "Моделирование информационной структуры параллельной обработки изображений" // Компьютерная оптика, 2010, т. 34, № 2, с. 231-242.
2. С.Б. Попов "Концепция распределенного хранения и параллельной обработки крупноформатных

изображений" // Компьютерная оптика, 2007, т. 31, № 4, с. 77-85.