РАСПРЕДЕЛЕННАЯ ОБРАБОТКА КРУПНОФОРМАТНЫХ ИЗОБРАЖЕНИЙ

В.А. Сойфер
Самарский государственный аэрокосмический университет имени академика С.П. Королева (национальный исследовательский университет)
e-mail: soifer@ssau.ru,
Н.Л. Казанский, С.Б. Попов
Институт систем обработки изображений РАН
e-mail: ipsi@smr.ru

Аннотация

Для обработки данных дистанционного зондирования Земли предложены эффективные схемы репликации крупноформатных изображений и оригинальный алгоритм динамической балансировки многопроцессорных систем. Реализация предложенного подхода позволила повысить производительность и отказоустойчивость распределенной системы обработки и хранения изображений.

Одной из важнейших проблем использования вычислительной техники является «отображение задач вычислительной математики на архитектуру вычислительных систем» [1]. Эта проблема была обозначена академиком Г.И. Марчуком как фундаментальное научное направление, кратко называемое «проблемой отображения».

В настоящее время наиболее актуальной представляется решение проблемы отображения вычислительных задач на параллельную архитектуру вычислительных систем (ВС), поскольку основным направлением повышения эффективности использования вычислительных средств является использование параллельных методов организации вычислений.

В области обработки изображений большинство задач может быть решено путем последовательного применения к обрабатываемым данным некоторого набора типовых операций обработки [2-3]. Именно эту особенность эксплуатирует большинство систем обработки изображений общего назначения, т.е. они создаются не под конкретную технологию обработки, а для решения широкого спектра задач обработки изображений интерактивным образом или на уровне макропрограммирования.

Естественный при обработке изображений параллелизм, основанный на декомпозиции данных, позволяет достаточно легко адаптировать последовательные реализации широкого спектра задач обработки для их выполнения на многоядерных процессорах. Однако с увеличением размеров изображений время их обработки все более определяется временем, затрачиваемым на операции ввода/вывода. В таких условиях при обработке крупноформатных изображений с использованием относительно простых в вычислительном отношении задач эффективность использования многоядерных процессоров резко падает. Существующие тенденции развития вычислительной техники еще более обостряют проблему: все увеличивающийся разрыв между производительностью процессоров и быстродействием устройств постоянного хранения данных существенно снижает показатели общей эффективности ВС при решении прикладных задач. Вместе с тем скорость передачи данных в локальных сетях растет существенно быстрее, чем аналогичный показатель для дисковых запоминающих устройств (ЗУ). В настоящее время скорость обмена данными в сетях на базе техноло-

гии Gigabit Ethernet (1GbE) находится приблизительно на одном уровне с дисками типовых конфигураций компьютера, но при переходе на технологию 10GbE ситуация кардинально изменится – визуализация крупноформатного изображения будет выполняться быстрее при удаленном, а не локальном хранении данных. В таких условиях использование распределенных систем для хранения и обработки крупноформатных изображений становится актуальным.

Другим важным фактором при обработке крупноформатных изображений является размер оперативной памяти. Несмотря на то, что появление 64-битных операционных систем отодвинуло ограничение на размер оперативной памяти, ее существенное наращивание относительно типовых решений достаточно дорого. Эффективным методом в данном случае является хранение и обработка изображения по частям на разных компьютерах. В сочетании с возможностью параллельной организации таких вычислений в распределенных системах, особенно при организации их на основе кластерных вычислительных систем, этот подход представляется перспективным.

Обоснованное решение проблемы отображения задач обработки изображений на архитектуру распределенных вычислительных систем с различными типами параллелизма должно опираться на моделирование и исследование различных вариантов организации хранения и обработки изображений применительно к целевой архитектуре аппаратно-программных средств.

Выделяются следующие основные подходы к организации программного обеспечения параллельной обработки изображений [4]: 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 размера оперативной памяти системы.

Для параллельных систем обработки изображений узким местом является централизованное размещение крупноформатных изображений на специализированных хранилищах данных и коммуникации согласования данных при выполнении операций локальной обработки скользящим окном (операций преобразования группы соседних отсчетов).

Централизованный доступ к данным в значительной степени связан с тем, как организованы вычисления в большинстве существующих систем параллельной обработки. В этих системах используется программа-менеджер, которая выполняет декомпозицию изображений непосредственно перед обработкой, причем размер перекрытия обрабатываемых фрагментов выбирается в соответствии с параметрами выполняемых операций (в первую очередь, размером окна обработки).

В докладе предлагается альтернативный подход, основанный на концепции распределенного изображения, которая обеспечивает распараллеливание операций доступа к данным, используя принцип "обрабатываем там, где храним".

Распределенное изображение – это структура данных, определяющая способ и параметры разбиения изображения на фрагменты, список компьютеров, где находятся эти фрагменты, место их размещения и формат хранения. Декомпозиция выполняется при создании или импорте изображения в систему и не изменяется в процессе хранения. При этом появля-

ется задача выбора оптимальной декомпозиции (разбиения на фрагменты) изображения в условиях отсутствия априорной информации о параметрах, запускаемых в системе задач обработки. Необходимо также учесть 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.

Анализ вариантов декомпозиции изображений при выполнении различных операций обработки показал, что наиболее целесообразным выбором представляется декомпозиция распределенного изображения в виде перекрывающихся фрагментов.

Новизна предлагаемого подхода заключается в том, что 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.

Рассмотрим основные аспекты предлагаемого подхода на примере одномерной декомпозиции. 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.

Минимизацию избыточности, необходимой для восстановления информации при отказе одного из узлов хранения, обеспечивает одномерная декомпозиция в виде перекрывающихся на половину своей высоты горизонтальных полос. В данном случае суммарный объем данных, занимаемый изображением в распределенной системе хранения, возрастает в два раза. Однако, по сравнению с простым дублированием узлов хранения распределенного изображения, потенциальные возможности распараллеливания предложенного подхода вдвое больше.

Для распределенного изображения разработаны принципы динамической организации хранения, 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:

- 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;
- при отказе (тайм-ауте) доступа к узлу хранения 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:

- если фрагмент имеет уровень репликации выше, чем заданный уровень репликации в системе (параметр системы) и время последнего доступа (использования его данных) к этому фрагменту превышает заданное время (параметр системы);
- если пользователь дал команду на удаление всего распределенного изображения; в этом случае анализируется история использования этого изображения, если с ним работал только пользователь, издавший запрос на удаление, то все фрагменты изображения удаляются из системы; если изображение использовалось 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.

Рассмотрим суть алгоритма на примере взаимодействия двух соседних узлов, m-ого и $(m+1)$-го, при формировании той части результирующего изображения, которая содержится в блоках предварительной декомпозиции данных с номерами $2m$ и $2m+1$. Заметим, что каждый из узлов имеет всю информацию, чтобы сделать эту работу самостоятельно (или почти всю, в случае выполнения операции локальной обработки скользящим окном).

m-й узел начинает формировать строки $2m$-го блока, начиная с первой строки в порядке возрастания номеров строк, $(m+1)$-й узел, в свою очередь, формирует строки $(2m+1)$-го блока, но, начиная с последней строки блока, в порядке убывания номеров строк. После

формирования определенного количества строк каждый узел информирует соседний узел о времени, за которое он выполнил эту работу. На основании этой информации вычисляется прогнозируемый номер строки изображения, на которой узлы завершат такую совместную обработку.

Таким образом, в процессе работы смежные узлы двигаются навстречу друг другу, сообщая о скорости своего процесса вычислений при достижении заранее определенных моментов. При этом прогнозируемый номер строки изображения, на которой эти процессы встретятся, постоянно корректируется в зависимости от текущей загрузки вычислительных узлов. Таким образом, все процессы завершат свою работу практически одновременно. Разница во времени при этом составит не больше, чем время обработки одной строки.

Одновременно каждый узел участвует в двух таких процессах, попеременно формируя строки старшего блока в порядке возрастания номера и строки своего младшего блока в порядке убывания, при необходимости переходя к формированию строк теневых блоков.

В результате будет сформировано новое распределенное изображение в полном объеме, но возможно с неравномерным размещением данных по узлам хранения. Далее в фоновом режиме узлы хранения сформированного распределенного изображения обменяются своими данными с тем, чтобы привести структуру распределенного изображения к необходимому виду. Однако пользователь может получить результат своего запроса сразу по завершении процесса обработки.

Аналитическое исследование эффективности предложенного алгоритма для идеализированного случая несбалансированной вычислительной системы, у которой производительность одного из узлов отличается от остальных, показало, что если распределенная система состоит из четырех узлов, то предлагаемый алгоритм может равномерно распределить нагрузку даже при трехкратном превышении производительности одного из узлов над остальными. Более производительный узел обеспечит обработку половины строк изображения, а на долю остальных узлов достанется по 1/6 части.

При увеличении числа узлов обработки/хранения разброс производительности, который может быть скомпенсирован без временных потерь, постепенно уменьшается до двукратного.

Достоинство предлагаемого алгоритма заключается в том, что он является полностью децентрализованным, и обеспечивает равномерное распределение нагрузки, если производительность соседних узлов не отличается больше чем в два раза.

Особенно актуальна задача распределенного хранения и обработки изображений при обработке данных, получаемых при дистанционном зондировании Земли в Поволжском центре космической геоинформатики [3, 5]. Создание такой системы, использующей децентрализованный подход к организации вычислений в процессе обработки изображений, основано на распределении поступающей в центр приема информации на компьютеры университетской сети ГРИД, а в процессе обработки на компьютеры распределенной системы вместо данных изображений рассылаются процедуры их обработки. Основой сети является ряд вычислительных кластеров: кластер центра приема космической информации (HP BLc3000 производительностью 1,5 ТФлопс), университетский кластер «Сергей Королев» (производительностью 10 ТФлопс: платформа IBM BladeCenter, 112 блейд-серверов IBM BladeCenter HS22; каждый сервер имеет по два четырехядерных процессора Intel Xeon 5560 с частотой ядра 2,8ГГц; общий объем оперативной памяти 1,3Тб; система хранения данных объемом

10Тб), сервер национальной наносети объемом 60 Тб и др. Система исследовалась на вычислительном кластере HP BLc3000 на базе 7 сдвоенных блейд-серверов HP ProLiant BL2x220c G5 в качестве вычислительных узлов и 1-го сервера HP ProLiant BL260c в качестве управляющего узла. Основные характеристики кластера: 1 управляющий узел (CPU: Quad-Core Intel Xeon 5430 2.66 GHz, cache 12 Mb, 1333MHz; RAM: 4Gb DDR2-667; HDD: 240Gb) и 14 вычислительных узлов (2 CPU: Quad-Core Intel Xeon 5430 2.33 GHz, cache 12 Mb, 1333MHz; RAM: 8Gb DDR2-667; HDD: 120Gb).

С помощью данной системы проведено исследование децентрализованного алгоритма динамического распределения нагрузки процессоров при выполнении локальной обработки космических изображений. Проведены вычислительные эксперименты по сравнению времени решения задач обработки изображений с использованием предлагаемой динамической балансировки и без нее. 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, получаемом 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 таким образом, чтобы нагруженные узлы чередовались с менее загруженными.

В заключение отметим следующие основные моменты:

- новый метод формирования децентрализованных структур, определяющих размещение распределенных данных изображений на основе декомпозиции перекрывающихся фрагментов, позволяет динамически балансировать загрузку компьютеров и обеспечивающий отказоустойчивость распределенного хранения фрагментов изображений;
- оригинальный алгоритм динамической балансировки многопроцессорных систем при распараллеливании операций обработки изображений, который опирается на предложенный метод декомпозиции распределенного изображения, является полностью децентрализованным, что оптимально при использовании его в распределенных системах, и позволяет обеспечить равномерное распределение нагрузки при параллельной обработке изображений в условиях гетерогенной вычислительной среды, причем возможно полное выравнивание нагрузки при двукратном различии производительности соседних узлов;
- предлагаемый подход к организации вычислений реализован при создании системы параллельной обработки крупноформатных изображений, получаемых в Поволжском центре космической геоинформатики со спутников дистанционного зондирования Земли.

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]. Марчук Г.И., Котов В.Е. Проблемы вычислительной техники и фундаментальные исследования. Автом. и вычисл. техн. 1979, № 2. С. 3-14.

[2]. Computer Image Processing, Part I: Basic concepts and theory. Edited by Victor A. Soifer. VDM Verlag. 2009. 296 p.

[3]. Computer Image Processing, Part II: Methods and algorithms. Edited by Victor A. Soifer. VDM Verlag. 2009. 584 p.

[4]. Merigot A., Petrosino A. Parallel processing for image and video processing: Issues and challenges. Parallel Computing. 2008. Vol. 34. P. 694-699.

[5]. Chernov A.V., Myasnikov E.V., Treschova E.V., Vorobiova N.S. The development of regional spatial data and metadata geoportal. Proceedings of 9-th International Conference on Pattern Recognition and Image Analysis: New Information Technologies (PRIA-9-2008), Russian Federation, Nizhni Novgorod, September 15—19. 2008. Vol. 2. P. 74-76.