

Алгоритмы распределения потоков информации в графической поисковой системе в реальном масштабе времени

© И.М. Гостев, А.Г. Подгорбунский

Московский институт электроники и математики
igostev@list.ru, shurikp@gmail.com

Аннотация

В работе рассматриваются преимущества и недостатки динамического и статического распределений видеопотока в графической поисковой системе в реальном масштабе времени в распределенной вычислительной среде. Приводятся модели на основе ориентированных графов. Рассматриваются алгоритмы построения распределенной вычислительной сети. Проводится их сравнение, и обсуждаются особенности реализации.

1 Введение

С развитием вычислительной техники в человеческую жизнь неизбежно вторгаются новые понятия и технологии. Появление новых технологий предоставляет новые возможности, а на основании новых возможностей и новых технологий у людей формируются новые потребности.

Развитие библиотечного дела приводит к появлению гигантских банков данных, которые содержат разнородную информацию. К ней относятся: различные формы представления текстов (форматы .gft, .pdf, .tex, djvu и т. п.); графическая информация в виде рисунков, скетчей и др. изображений в самых различных форматах; видеофильмы; аудио книги и многие другие составляющие мультимедийной информации, включая различные комбинации. Обилие и разнородность информации приводит к необходимости использования современных технологий их хранения и поиска.

Если в настоящее время поиск текстовой информации еще как-то удовлетворительно решен, то с навигацией по мультимедийной информации остается практически в начальном состоянии уже длительное время. Известные системы поиска графической информации по содержанию графического документа, практически не изменились за послед-

ние 10 лет и не вышли за пределы исследовательских прототипов.

В [1] было приведено описание графической поисковой системы (Graphic Search System – GSS) для поиска графического содержания в глобальных банках данных, а в [2] введена математическая модель поисковой системы, которая затем была реализована. Однако хотя эта система и была позиционирована как поисковая система реального времени, она не позволяет производить поиск в видео документах. Это объясняется тем, что, как и в любой другой поисковой системе, в ней создается индексируемый список по некоторым заранее выбранным критериям поиска. Однако это удовлетворительно (по скорости) работает с графическими изображениями, но при переходе к поиску фрагментов в видеопотоках требует существенного повышения скорости обработки.

Оценка таких требований, выполненная в [3] показывает, что скорость вычислительной системы для обработки видео потока должна быть пропорциональна размеру изображения, и для телевизионного формата превышать существующие мощности одиночного компьютера примерно в 25-30 раз. Для решения этой проблемы были разработаны параллельные алгоритмы, позволяющие реализовать поисковую систему в распределенной среде.

При обработке непрерывного видеопотока графической информации необходимо решать несколько проблем. Во-первых, необходимо оценивать быстродействие вычислительной системы, на которой производится такая обработка. Во-вторых, – разработать специальные методы, которые обеспечат работу в реальном времени.

Интуитивно понятно, что при высоких скоростях видео потока включающего несколько этапов его обработки, невозможно обеспечить выполнение всех операций на одном компьютере. Необходимо строить процесс на основе распределенной вычислительной системы. В такой среде каждый из ее элементов будет выполнять свою операцию по обработке изображения и распознаванию образов. Управление всем процессом по обработке и распознаванию должно осуществляться одним контроллером, который будет распределять задания по вы-

Труды 10-й Всероссийской научной конференции «Электронные библиотеки: перспективные методы и технологии, электронные коллекции» – RCDL'2008, Дубна, Россия, 2008.

числительным узлам системы, осуществлять контроль и синхронизацию их выполнения.

В такой системе каждый метод по обработке и распознаванию необходимо строить так, чтобы обеспечить выполнение операции при однократном проходе по изображению. Кроме того методы (по возможности) не должны иметь параметров настройки процесса на изображение. В работах [3] и [4] была предложена и развита идея построения распределенной системы обработки и распознавания изображений в реальном времени, основанной на принципах конвейера и нарезки потока на участки. При проведении последующих исследований возникла проблема распределения отдельных фрагментов потока по процессорам, на которых он должен обрабатываться. В теории параллельных вычислений известны несколько способов моделирования параллельных вычислений. Одним из них является представление процесса при помощи сетей Петри [5]. В настоящей работе рассматривается еще один метод, основанный на применении ориентированных графов [6].

2 Математическая модель процессов

Методы распределения потока можно разделить на статические и динамические. В первом случае разбиение потока определяется на этапе начальной инициализации системы и далее остается неизменным. Для любого фрагмента потока можно заранее сказать, по какому маршруту он пройдет, и какими экземплярами каждого метода будет обработан. При динамическом способе расчета, это определяется на стадии обработки. Статический подход имеет два основных преимущества и позволяет:

- снять с вычислительных узлов нагрузку по расчету маршрутизации фрагментов изображения на стадии выполнения;
- построить гораздо более эффективную и простую структуру для хранения фрагментов изображения в памяти каждого вычислительного узла.

Рассмотрим односвязный, ориентированный, ациклический граф, имеющий один исток и один сток, вершинами которого являются *экземпляры метода* или *узлы вычислительной сети методов*, а дугами – *каналы передачи потока*.

По каналам вычислительной сети методов передаются данные, состоящий из фрагментов потока. Методы, имеют несколько входящих каналов и несколько исходящих, осуществляют «нарезку» потока – разделение потока на несколько потоков, содержащих чередующиеся фрагменты, и «сборку» потока – слияние нескольких потоков в один.

Всем процессом обработки потока управляет компонент, называемый «контроллером потока». Контроллер потока в качестве входной информации принимает набор методов обработки и распознавания изображений и параметры этих методов.

Метод – процедура обработки изображения, реализованная в качестве библиотечной функции. В качестве входных данных метод принимает изобра-

жение (фрагмент изображения) и параметры для процедуры обработки. В результате работы метода получается выходное изображение и/или некоторые другие данные, полученные в результате применения процедуры, формат и смысл которых зависит от конкретного метода.

Под методом также понимается программа, функционирующая как узел вычислительной сети методов.

Далее рассматривается случай, когда:

- графический поток представляет собой одно очень длинное непрерывное изображение.
- *фрагментом* такого потока будет непрерывный участок изображения, состоящий из целого числа строк.
- Методы обработки изображения либо точечные либо пространственные.

Вычислительная сеть методов будет удовлетворять следующим ограничениям:

- Каждому элементу *последовательности методов* соответствует один или несколько узлов, являющихся экземплярами этого метода. Совокупность этих узлов образует *ярус* сети, соответствующий данному методу цепочки.
- Все входные каналы данного узла, соединяют его только с узлами яруса, соответствующего предыдущему методу цепочки. Все выходные каналы узла соединяют его только с узлами яруса, соответствующего следующему методу в цепочке.
- Первый ярус содержит единственный узел, соответствующий первому методу цепочки методов – методу загрузки (считывания) изображения и не имеет входных каналов.
- Последний ярус содержит единственный узел, соответствующий последнему методу цепочки – методу сохранения (выдачи) результирующего изображения и не имеет исходящих каналов.

Каждый метод имеет набор характеристик:

- относительная вычислительная сложность процедуры;
- смещение по высоте (y^-, y^+) ;
- параметры метода.

Относительная вычислительная сложность рассчитывается эмпирически на конкретной ЭВМ относительно эталонного метода, как соотношение средних времен обработки эталонного изображения.

Смещения по высоте представляют собой пару чисел, определяющих сколько строк до и после фрагмента требуется для обработки соответствующего фрагмента результирующего изображения.

Другими словами, смещение (y^-, y^+) означает, что для вычисления $I[s_1, s_2]$ необходим фрагмент $I[s_1 - y^-, s_2 + y^+]$. Эти смещения играют большую роль в процессе расчета схемы потока через вычислительную сеть и определяют некоторые требова-

ния к организации памяти для фрагментов изображения внутри вычислительных узлов.

3 Алгоритм построения вычислительной среды

3.1 Статическое распределения потока

Рассмотрим алгоритм статического распределения потока. Пусть исходный поток представляет собой одно длинное изображение – «ленту». Фрагмент потока – непрерывный участок изображения, содержащий целое число строк. Промежуточные потоки состоят из последовательности фрагментов. Если поток состоит из фрагментов одинаковой длины, следующих через равные промежутки, то он может быть описан схемой потока.

Схема потока представляет собой тройку $\langle S, L, P \rangle$ где S – начальная строка первого фрагмента. L – длина каждого фрагмента в строках, P – промежуток от конца предыдущего фрагмента до начала следующего. Т. о. $\langle S, L, P \rangle$ задает последовательность фрагментов $[S + (L + P) * i, S + (L + P) * i + L], i \in N_0$

Две схемы называются *смежными*, если поток, полученный слиянием двух потоков (состоящий из всех фрагментов первого и второго потока) так же может быть задан схемой потока (такая схема называется *объединенной*).

Символом $\aleph_i(\langle S, L, P \rangle)$ обозначается схема потока, скорректированная в соответствии со смещениями по y для i -го метода (необходимого методу на входе, чтобы производить поток $\langle S, L, P \rangle$ на выходе):

$$\aleph_i(\langle S, L, P \rangle) = \langle S - y_i^-, L + y_i^- + y_i^+, P - y_i^- - y_i^+ \rangle$$

Алгоритм состоит из двух шагов – прямого хода и обратного. На первом шаге происходит определение количества вершин в каждом ярусе и построение связей между ярусами исходя из равномерного последовательного разделения потока в каждом ярусе. На втором шаге происходит коррекция соединений и разметки с целью исключения каналов, по которым передаются очень малые относительно среднего потока на ярусе фрагменты.

3.2 Алгоритм построения вычислительной сети с равномерным статическим распределением потока

Вход: цепочка методов $\langle M_1, K, M_N \rangle$, относительные вычислительные сложности c_1, K, c_N , пропускная способность эталонного метода q_e , требуемая пропускная способность цепочки q .

Выход: Построенный граф вычислительной сети $G = \{V, E\}$, ребра и вершины которого размечены схемами потоков

$$M : E \rightarrow \{\langle S, L, P \rangle\}, \quad M' : V \rightarrow \{\langle S, L, P \rangle\}.$$

Шаг 1. Для каждого $i \in [1, N]$ определить ярус

$$G_i = \{V_i, \emptyset\}, \quad \text{где } V_i = \left\{ v_i^k \mid k = 1.. \left\lceil \frac{q_e}{q} c \right\rceil \right\}$$

– набор упорядоченных узлов методов, являющихся экземплярами метода M_i . Положить

$$V = \bigcup_{i \in [1, n]} V_i; \quad E = \emptyset.$$

Приписать истоку схему потока $\langle 0, L, 0 \rangle$, где L – исходная длина блока, задаваемая контроллером потока на этапе конфигурации. Положить разметку

$$M = \emptyset, \quad M' = \left\{ \left(v_1^1, \langle 0, L, 0 \rangle \right) \right\}.$$

Для каждого узла v_i^k разметить его схемой

$$\left\langle \left\lfloor \frac{k-1}{|V_i|} \right\rfloor, \left\lfloor \frac{k}{|V_i|} \right\rfloor - \left\lfloor \frac{k+1}{|V_i|} \right\rfloor, L - \left\lfloor \frac{k}{|V_i|} \right\rfloor + \left\lfloor \frac{k+1}{|V_i|} \right\rfloor \right\rangle$$

Для каждой пары узлов v_i^k и $v_{i+1}^{k'}$, $i \in [1, N-1]$, $k \in [1, |V_i|]$, $k' \in [1, |V_{i+1}|]$ в случае если $M'(v_i^k) \cap \aleph_i(M'(v_{i+1}^{k'})) \neq \emptyset$ добавить дугу между узлами: $E = E \cup (v_i^k, v_{i+1}^{k'})$ и пометить ее схемой-пересечением:

$$M = M \cup \left\{ \left((v_i^k, v_{i+1}^{k'}), M'(v_i^k) \cap \aleph_i(M'(v_{i+1}^{k'})) \right) \right\}.$$

Шаг 2. Для каждого яруса V_i , $i = N, \dots, 2$ для каждого узла $v_i^k \in V_i$, $k \in [1, |V_i|]$:

а) Проверить, что объединенная схема выходов совпадает со схемой узла:

$$\bigcup_{(v_i^k, v) \in E} \{M(v_i^k, v)\} = M'(v_i^k).$$

В случае, если это не так и объединенная схема включает схему узла, то вычислить множество недостающих схем

$$\bigcup_{(v_i^k, v) \in E} \{M(v_i^k, v)\} \setminus \aleph_i(M'(v_i^k))$$

и каждый недостающий элемент этого множества прибавить к одной из схем входящих ребер узла, имеющую схему смежную с недостающей схемой и наибольшую длину.

б) Удалить входные дуги, длина схемы которых меньше определенного порога и схемы этих дуг приписать дугам наибольшей длины, имеющим смежные схемы.

3.3 Динамическое распределения потока

При динамическом распределении потока между обрабатываемыми узлами, маршрут по которому пройдет конкретный фрагмент изображения определяется на стадии выполнения. При этом из-за смещений по высоте, для обработки данного фрагмента могут потребоваться несколько строк предыдущего или последующего фрагмента, непосредственно прилегающих к данному. А так как, заранее неизвестно какие именно фрагменты пройдут через данный экземпляр метода, невозможно сказать требуется ли сохранять данный фрагмент. Кроме того, встает проблема перемещения фрагментов в памяти – для того чтобы обеспечить обработку данного фрагмента с учетом смещений по высоте, требуется разместить предыдущий и последующий фрагменты непосредственно до и после него. Эти два фактора в совокупности усложняют процедуру управления памятью и требуют реализации дополнительной логики маршрутизации на этапе обработки потока, включающую коммуникацию с контроллером потока, координирующем в этом случае маршрутизацию. Это противоречит одной из основных концепций, положенных в основу проектирования системы, согласно которой взаимодействие контроллера потока с методами на этапе обработки должно быть минимальным. Однако обе проблемы могут быть решены с помощью избыточной передачи фрагментов, так, чтобы каждый фрагмент на каждом ярусе сети мог быть обработан независимо. В этом случае освободить память, занимаемую фрагментом, можно сразу после его обработки. Единственным и очевидным недостатком такого метода является повышение общего объема пересылаемых по сети фрагментов вследствие избыточной передачи строк, однако при больших значениях длины пересылаемых фрагментов и небольшом количестве ярусов эти издержки сравнительно малы.

Далее приводится алгоритм для построения сети с динамическим избыточным распределением потока. При этом в отличие от статического метода, где схема потока, приписанная конкретному узлу или дуге сети, определяет какие в точности фрагменты изображения и в каком порядке пройдут через данную дугу или узел, схема потока при динамическом распределении определяет какие фрагменты могут пройти через узел или дугу. Все узлы одного яруса являются равноправными в том смысле, что любой узел предыдущего яруса может передать обработанный фрагмент любому из узлов следующего яруса.

3.4 Алгоритм построения вычислительной сети с равномерным динамическим распределением потока с избыточными строками

Вход: цепочка методов $\langle M_1, K, M_N \rangle$, относительные вычислительные сложности c_1, K, c_N ,

пропускная способность эталонного метода q_e , требуемая пропускная способность цепочки q .

Выход: Построенный граф вычислительной сети $G = \{V, E\}$, ребра и вершины которого размечены схемами потоков

$$M : E \rightarrow \{\langle S, L, P \rangle\}, \quad M' : V \rightarrow \{\langle S, L, P \rangle\}$$

Для каждого $i \in [1, N]$ определить ярус

$$G_i = \{V_i, \emptyset\}, \quad \text{где } V_i = \left\{ v_i^k \mid k = 1.. \left\lceil \frac{q_e c}{q} \right\rceil \right\} - \text{набор упорядоченных узлов методов, являющихся экземплярами метода } M_i.$$

Положить $V = \bigcup_{i \in [1, n]} V_i$; $E = \emptyset$. Приписать стоку схему потока $\langle 0, L, 0 \rangle$, где L – исходная длина блока, задаваемая контроллером потока на этапе конфигурации. Положить разметку $M = \emptyset, M' = \{\langle v_N^1, \langle 0, L, 0 \rangle \rangle\}$,

Для каждого яруса $i \in [N-1, 1]$ разметить все его вершины схемами

$$\left\langle -\sum_{j=i}^N y_j^-, L + \sum_{j=i}^N (y_j^- + y_j^+), -\sum_{j=i}^N (y_j^- + y_j^+) \right\rangle$$

Для каждой пары узлов v_i^k и $v_{i+1}^{k'}$, $i \in [1, N-1]$, $k \in [1, |V_i|]$, $k' \in [1, |V_{i+1}|]$ добавить дугу между ними: $E = E \cup (v_i^k, v_{i+1}^{k'})$ и пометить ее схемой

$$\left\langle -\sum_{j=i+1}^N y_j^-, L + \sum_{j=i+1}^N (y_j^- + y_j^+), -\sum_{j=i+1}^N (y_j^- + y_j^+) \right\rangle$$

В построенной таким образом сети каждый узел каждого яруса соединен дугой с каждым узлом последующего яруса. При большом количестве узлов в соседних ярусах, это может привести к значительной дополнительной нагрузке на вычислительные и сетевые ресурсы кластера (каждая дуга соответствует двум дополнительным потокам в процессах отправляющего и принимающего узлов). Для снижения количества дуг можно использовать для вычисления множества дуг первый алгоритм, а для разметки дуг и узлов – второй.

Каждый узел может послать обработанный фрагмент любому из узлов следующего яруса. Простейшим способом маршрутизации обработанных фрагментов из данного узла является чередующаяся рассылка фрагментов последовательно каждому из узлов следующего яруса. При этом будет достигаться равномерное распределение загрузки между всеми узлами каждого яруса, в предположении, что каждый узел обладает равным количеством ресурсов.

4 Обсуждение и выводы

Подводя итог, можно выделить следующие преимущества и недостатки статического и динамического способа распределения потока:

Распределенная система обладает высокой отказоустойчивостью – при выходе из строя одного из узлов сети, его работу перераспределяется на остальные узлы того же яруса. Поэтому при динамическом распределении потоков неисправный узел будет заменен. При статическом же методе распределения, отказ узла приведет к появлению в выходном изображении необработанных участков.

Динамическое распределение дает возможность более гибкой балансировки загрузки узлов вычислительной сети. Если какой-то узел перегружен, его работу можно распределить на другие узлы того же яруса. Статическое распределение не позволяет этого делать.

При динамическом распределении суммарный объем пересылаемых данных возрастает, вследствие избыточной передачи строк. Появляется дополнительная нагрузка на узлы, связанная с маршрутизацией и балансировкой исходящих фрагментов.

По сравнению со статическим методом возрастает число потоков приема и передачи фрагментов и число сетевых соединений.

В целом преимущества динамического метода превосходят его недостатки, что делает его предпочтительным в дальнейшей разработке системы.

Литература

- [1] Гостев И.М. Графическая поисковая система GSS2001 // Сборник «Информационные технологии в системах вычислительной техники». Вып. 2. – М. : МГИЭМ, 2002.
- [2] Гостев И.М., Мирошкин А.В. Математическая модель одного класса поисковых систем // Вестн. РУДН. Сер. Прикладная и компьютерная математика. – 2004, Т. 3, №1. – С. 93–98.
- [3] Gostev I. M., Sevastianov L. A. About construction distributed calculation in real time system of image processing and pattern recognition // Proc Int. conf. «Distributed computation and GRID in science and education » Dubna JINR June 2004. – Pp.79–85.
- [4] Gostev I. M., Sevastianov L. A. Internal Architecture of Distributed Real-time System of Image Processing and Pattern Recognition // Proc. XX Int. Symp. on Nuclear Electronics & Computing (NEC 2005), Varna, Bulgaria, Sept. 2005.
- [5] Питерсон Дж. Теория сетей Петри и моделирование систем. – М. : Мир, 1984. – 264 с.
- [6] Мелихов А.Н. Ориентированные графы и конечные автоматы. – М. : Наука, 1971. – 416 с.

Information distribution algorithms for the real-time image search system

I.M. Gostev, A.G. Podgorbunsky

This paper deals with issues of distribution of video-flow used in the real-time distributed image search system. It introduces architecture based on graphic-flow controller and distributed network of nodes, applying image recognition and analysis methods to flow parts, static and dynamic methods of distribution of flow among these nodes, their benefits and drawbacks. The paper describes models of distribution and algorithms of distributed computational network building, compares them and discusses details of implementation.