

Востриков А.А.

Декодирование видеосигнала формата H.264 посредством распределенных вычислений в локальной сети.

Аннотация.

В данном докладе представлены предпосылки к использованию распределенных вычислений для решения задачи просмотра видео на персональных компьютерах, процессоры которых не могут справиться с объемом вычислений при декодировании видеопотока формата H.264/MPEG-4 AVC в реальном времени. Исследована возможность применения распределенных вычислений для декодирования такого видео. На примере небольшой сети обоснована технология с применением вторичного кодирования. Показаны проблемы использования данного подхода и методы минимизации накопления ошибок. В докладе также предложены методы оценки эффективности использования предложенной схемы для декодирования видео.

Востриков А.А.

Декодирование видеосигнала формата H.264 посредством распределенных вычислений в локальной сети.

Из-за повышенного коэффициента сжатия, более чем на 25% по сравнению с предшествующими, стандарт H.264/MPEG-4 AVC начал использоваться повсеместно. Математический аппарат кодирования видеосигнала по стандарту H.264 значительно сложнее ранее применяемых алгоритмов. Данное обстоятельство предъявляет очень высокие требования не только к кодеру, но и к декодеру. Вследствие этого в мире существует огромное количество устаревших персональных компьютеров (PC от англ. *Personal Computer*), которые не могут справиться с объемом вычислений, требуемым для декодирования такого видеопотока в реальном времени, особенно, если дело касается форматов телевидения высокой чёткости (HDTV). Данное обстоятельство делает такие PC непригодными для просмотра видео.

Предпосылки.

Вследствие развития технологий P2P сетей (от англ. *peer-to-peer* - *точка-точка*) наблюдается значительный подъём использования глобальной сети в качестве единого, распределенного хранилища данных.

P2P сети (их также называют одноранговыми, децентрализованными или пиринговыми) – это компьютерные сети, основанные на равноправии участников. В таких сетях отсутствуют выделенные серверы, а каждый узел (peer) является как клиентом, так и сервером. В отличие от архитектуры клиент-сервера, такая организация позволяет сохранять работоспособность сети при любом количестве и любом сочетании доступных узлов [1].

BitTorrent (букв. англ. «битовый поток») — пиринговый (P2P) сетевой протокол для кооперативного обмена файлами через Интернет. Файлы передаются частями, каждый клиент, получая (“скачивая”) эти части, в это же время отдаёт (“закачивает”) их другим клиентам, что снижает нагрузку и зависимость от каждого клиента-источника и обеспечивает избыточность доступа данных [2]. Самым распространенным программой BitTorrent-клиентом является µTorrent, интерфейс которой переведен на 52 языка а число пользователей, на декабрь 2008г, во всем мире более 28 миллионов [3]. Torrents.ru — крупнейший российский BitTorrent-трекер, насчитывающий около 4 миллионов активных учётных записей [4]. На трекере зарегистрировано более 777 тысяч раздач, суммарный размер которых составляет более петабайта [5].

Другим хорошо известным представителем P2P сетей является Direct Connect. Наиболее известной программой-клиентом является DC++, которая охватывает аудиторию в 90% всех пользователей Direct Connect. Интерфейс программы переведен на 56 языков, а количество пользователей сложно поддается оценке [6].

Использование BitTorrent™ и Direct Connect дают рядовому пользователю ощущение того, что он является частицей единого, распределенного хранилища данных, где он всегда может найти все, что ему необходимо. Для чего ему, также как и всем другим участникам, помимо пользования чужими ресурсами необходимо предоставлять и свои. Говоря простыми словами, одним из ключевых преимуществ является то, что участие в проекте дает пользователю круглосуточный доступ к фильмам, музыке, книгам, итп. всех пользователей проекта.

Помимо пиринговых сетей в современном мире активно развивается инфраструктура распределенных вычислений. Суть данной технологии заключается в том, что при решении сложных задач, требующих часы, недели, месяцы и, даже, миллионы лет процессорного времени одного РС, вместо суперкомпьютера используются все компьютеры глобальной сети, участвующие в проекте. Причем участвовать в проекте может каждый, для этого требуется лишь доступ в интернет и желание пользователя предоставить мощности своего РС для решения поставленных задач.

Например Folding@home - проект распределённых вычислений для проведения компьютерной симуляции свёртывания молекул белка. По состоянию на 21 сентября 2009 года, производительность проекта составляет 8,207 Пфлопс [7]. Для сравнения, самый мощный на сегодняшний день суперкомпьютер Jaguar, размещенный в Национальном Центре компьютерных исследований в Окридже, штат Теннесси, США имеет пиковую производительность 1.38 Пфлопс [8]. (флопс - величина, используемая для измерения производительности компьютеров, показывающая, сколько операций с плавающей запятой в секунду выполняет данная вычислительная система)

Технологии распределенных вычислений позволяют использовать процессорное время простоя компьютеров. Говоря простыми словами, для решения сложной задачи, для которой могли бы потребоваться миллионы лет процессорного времени рядового РС, нет необходимости покупать суперкомпьютер. Такие задачи можно решить посредством множества процессоров глобальной сети. А время решения задачи зависит лишь от того, скольких пользователей ты сагитируешь.

Помимо распределенных вычислений и пиринговых технологий предпосылкой к исследованию является концепция Web 2.0, суть которой также состоит в том, что пользователь, предоставляя свою информацию, получает взамен огромную структурированную базу данных информации от таких же, как он. В качестве примеров можно привести: социальные сети - Odnoklassniki.ru™ VKontakte.ru™, платформы для блогов LiveJournal.com™ Myspace.com™, проекты имеющие, как пользовательскую, так и научную ценность: Wikipedia.org™, Wikimapia.org™, Google Wave™.

Чем больше пользователей, тем целостнее система, тем большую ценность она несет. Это и есть феномен Web 2.0 .

Помимо сказанного предпосылкой исследования является и то, что обратная задача, то есть задача кодирования, распределённая на несколько процессоров, успешно реализована в некоторых кодерах, например [x.264™](#)

Исследование.

Исходя из изложенных предпосылок, задачей исследования является установить, эффективность концепции распределенных вычислений, если применить её в задаче декодирования видеосигнала стандарта H.264/MPEG-4 AVC

Условно возьмем фильм продолжительностью $t=90$ минут, $k=24$ кадра в секунду, разрешение $1920*1080$ пикселей, глубина цвета 24 бита на пиксель.

Подсчитаем сколько информации будет занимать такой фильм в несжатом состоянии:

$$M = t*60*k*1920*1080*24 = 90*60*24*1920*1080*24 = 6449725440000 \text{ бит} = 750 \text{ Гбайт}$$

Подсчитаем скорость передачи данных необходимую для показа такого фильма:

$$V = M/(90*60) = 1.11 \text{ Гбит/с}$$

Таким образом, для декодирования исходного файла посредством распределенных вычислений необходимо, чтобы соединение с каждым пользователем в локальной сети было не менее 1.11 Гбит/с. С учетом того, что сетевые адаптеры в большинстве своём бывают трех типов 10Мбит/с, 100 Мбит/с и 1 Гбит/с данное ограничение ставит под вопрос концепцию использования «устаревших» персональных компьютеров.

С другой стороны раздающий задания РС (назовем его *сервер*) может адаптироваться под конкретные условия. Т.е. если необходимая скорость передачи (V) видеопотока меньше скорости сетевого канала (V_k), то весь алгоритм декодирования будет производиться на удаленных компьютерах (здесь и далее под *удаленными компьютерами* подразумеваются те РС, которые не являются сервером для исполняемой задачи). Далее видеоданные будут передаваться серверу для воспроизведения. Если же $V > V_k$, то появляется необходимость разделить задачу на две части. Рассмотрим это на примере:

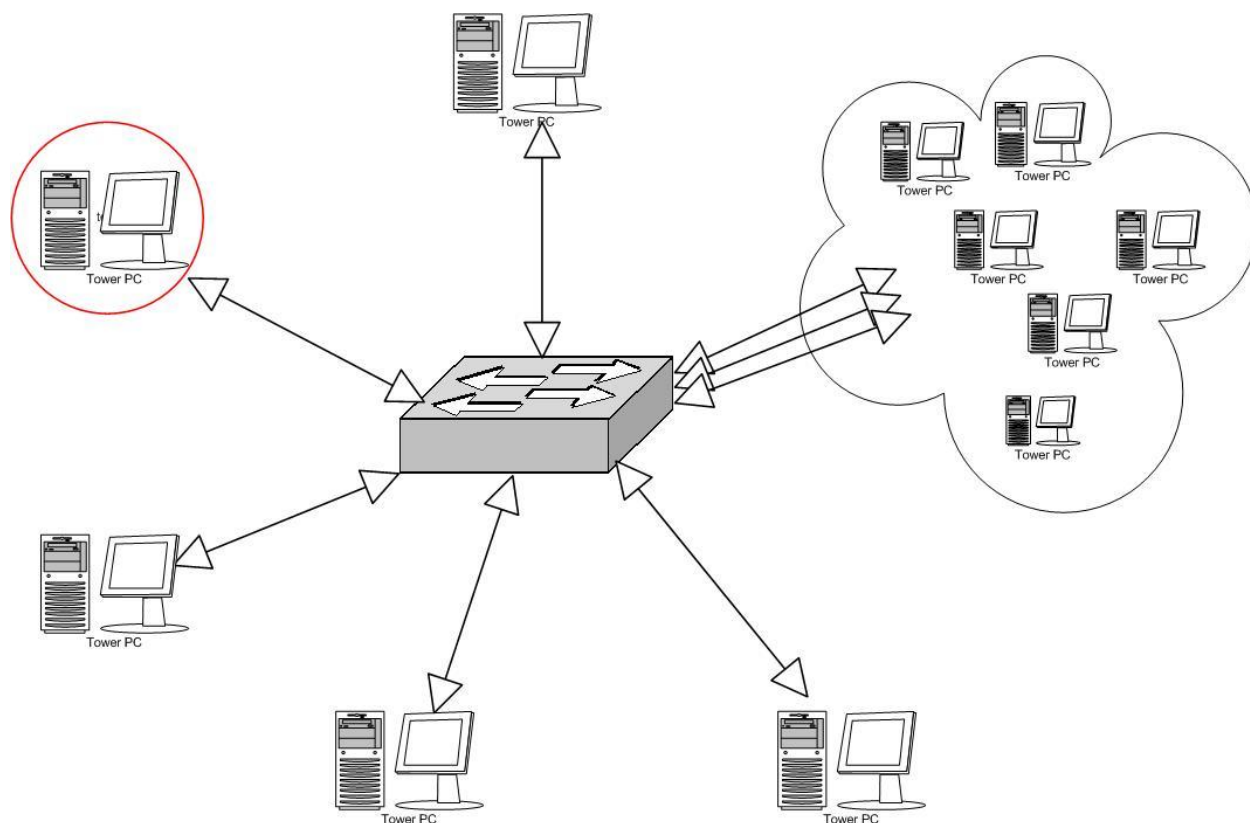


Рисунок 1. Физическая схема соединения распределенной вычислительной сети.

К сетевому концентратору по витой паре, через сетевые интерфейсы (100 Мбит/с) соединены n компьютеров (рисунок 1). После инициализации декодирования исполняемого файла компьютер (который с этого момента становится сервером – обведен в круг) создает соединения с удаленными компьютерами в своей локальной сети. В момент инициализации соединений сервер выясняет производительность и загруженность всех компьютеров локальной сети. После чего посредством многоадресной рассылки (multicast) сервер начинает передавать задания всем удаленным компьютерам. Наиболее производительным и наименее загруженным компьютерам задание отправляется в первую очередь.

Получив сжатый фрагмент видео формата H.264/MPEG-4 AVC, удаленный компьютер декодирует его, сжимает менее «тяжелым» кодеком и отправляет обратно серверу. Алгоритм повторного сжатия выбирается адаптивно, в зависимости от загруженности канала связи и загруженности процессора на сервере. Важным критерием при использовании повторного сжатия является накопление ошибок, т.к. искажения при неправильном выборе метода кодирования могут существенно возрасти. Для уменьшения данного эффекта, во-первых, можно уменьшить расстояние между I-кадрами в два или более раза [9]. Во-вторых, можно пренебречь использованием внутрикадрового разделения на макроблоки переменного размера. Не стоит забывать и про то, что уменьшение расстояния между I-кадрами и использование макроблоков минимального (постоянного) размера, например 4x4, увеличит объем информации

по отношению к исходному, сжатому фрагменту. Зато существенно будет снижена скорость передачи данных необходимая для показа видео, по сравнению с передачей несжатого видео.

Так же не стоит забывать про форматы сжатия видео без потерь, коэффициенты сжатия которых в настоящее время достигают 3 и более [10, 11,12].

Вдобавок, использование вторичного кодирования дает преимущество в тех случаях, когда разрешения экрана компьютера меньше разрешения исходного видео. Например, если взять видео, рассматриваемое в примере (см. выше) - 1920x1080 (16:9), а разрешение экрана 1024x768 (4:3), то нет смысла вторично кодировать видео формата 1920x1080, достаточно будет 1024x576 (16:9). Данное преобразование позволит уменьшить количество пикселей в k раз, где k – отношение количества пикселей исходного кадра, к количеству пикселей требуемого видеопотока.

$$k = \frac{1920 \cdot 1080}{1024 \cdot 576} = 3,515625$$

Подсчитаем эффективность использования предложенного метода распределенных вычислений для декодирования видео. Допустим, что все компьютеры локальной сети идентичны и одинаково загружены, тогда время декодирования одного видеофрагмента:

$$T_i = T_i \text{ отправки} + T_i \text{ декодирования} + T_i \text{ вторичного кодирования} + T_i \text{ доставки} + T_i \text{ вторичного декодирования}$$

Тогда общее время декодирования всего потока:

$$T_{\hat{1}} = \frac{1}{n} \sum_i T_i$$

Тогда эффективность предложенного метода :

$$E = \frac{\sum_i T_i}{\frac{1}{n} \sum_i T_i} = \frac{n \cdot \sum_i T_i}{\sum_i T_i}$$

Список литературы:

- 1) http://ru.wikipedia.org/wiki/Пиринговые_сети
- 2) <http://ru.wikipedia.org/wiki/BitTorrent>
- 3) <http://lenta.ru/news/2008/12/26/millions/>
- 4) http://services.bal-con.ru/torrents_ru/
- 5) <http://ru.wikipedia.org/wiki/Torrents.ru>
- 6) <http://en.wikipedia.org/wiki/DC++>
- 7) <http://ru.wikipedia.org/wiki/Folding@Home>
- 8) [http://ru.wikipedia.org/wiki/Jaguar_\(суперкомпьютер\)](http://ru.wikipedia.org/wiki/Jaguar_(суперкомпьютер))
- 9) Стандарт сжатия H.264
- 10) http://compression.ru/video/codec_comparison/pdf/lossless_codecs_test.pdf

- 11) L.Hanzo, P.Cherriman, J.Streit. Video compression and communications. Second Edition. Wiley, 2007. ISBN: 9780470518496, 704 p.
- 12) В.А.Грибунин MPEG-4 vs. H.264 "Технологии Защиты" №4, 2008