Санкт-Петербургский Государственный Университет

Математико-механический факультет

Кафедра системного программирования и кафедра информатики

Реализация схемы распределённого поисква с использованием технологии Opera Unite

Курсовая работа студентов 345 групп

Землянского Юрия Андреевича

Научный руководитель — М. Л. Симуни

ст. преподаватель

Санкт-Петербург

2010

Оглавление

1. Введение. 3
2. Существующие системы распределённого поиска.....4
3. Цели.....5
4. Технология Opera Unite.....6
5. Структура системы распределённого поиска.....8
    5.1. Прототип. 8
    5.2. Минусы реализации.....7
    5.3. Возможные оптимизации.....8
    5.4. Реализация.....9
    5.5. Логика приложения при старте.....9
    5.6 Индексация.....10
    5.7. Сервер.....10
6. Итог.....11
7. Направления дальнейшего развития приложения.....12
8. Список литературы.....13

1. Введение

Данная работа посвящается распределенным вычислениям, которые становятся всё более популярными, а точнее одной из прикладным задач, которая решается такими системами — распределённому поиску. У распределённых систем есть важные преимущества — масштабируемость и устойчивость к отказе отдельных частей. Авторов привлекла возможность адаптировать информационный поиск в интернете под такую модель. Поискав материалы по этой теме мы обнаружили, что эта область мало изучена, и решили попытаться реализовать свою систему распределенного децентрализованную поиска.

2. Существующие системы распределённого поиска

Примеров работающих систем распределённого поиска авторы не нашли, однако существуют проекты поисковых машин, основанные на распределённой индексации. Например, Majestic-12 [1] [2] — английский проект распределенных вычислений одноименной компании, направленный на создание поисковой системы, способной составить «конкурентоспособную альтернативу Google», но с более полной и глубокой базой данных проиндексированных интернет страниц.

Работа Majestic-12 основана на использовании не процессорного времени, а интернет-канала пользователя, для того, чтобы его компьютер служил «сервером», собирающим информацию о различных интернет-сайтах, для построения поискового индекса, который позже пересылается на головные сервера. Т.е. распределенной является только индексация, а сам поиск, по сути, остается централизованным.

3. Цели

Цель данной работы была построить систему распределённого поиска по файлам на компьютерах, объединённых в одну сеть. Система должна была удовлетворять следующим требованиям:

1. **Гомогенность**. Компьютеры в системе должны использовать одно и тоже программное обеспечение. Сервера — центральные компьютеры — могут присутствовать, но основная часть вычислений должна производиться на обычных компьютерах сети.

2. **Слабая нагрузка на отдельный компьютер**. Так как сеть будет состоять из отдельным компьютеров, нагрузка на отдельное звено должна быть небольшой - пользователь должен иметь возможность продолжать без помех работу.

3. **Масштабируемость**. Система должна работать при достаточно большом количестве пользователей. В глобальных сетях время необходимое для обработки сообщения почти всегда может быть проигнорировано по сравнению со временем передачи сообщения. Поэтому должны существовать механизмы, позволяющий не нагружать всю систему целиком при каждом отдельном запросе.

4. **Безопасность**. Так как сеть полностью публичная, со свободным доступом, то должны существовать механизмы защиты системы от агрессивных действий отдельных пользователей.

4. Технология Opera Unite.

Следующим шагов после определения целей был выбор платформы и инструментов разработки. И, в связи с этим, возник один из самых важных вопросов: каким образом наше приложение попадет к конечному пользователю? Каким образом он узнает, а самое главное, установит незнакомое приложение, при этом не испытывая особой нужды в нем. Поэтому мы пришли к выводу, что обычное десктопное приложение – вариант проигрышный. Тогда было решено попристальней взглянуть на недавно появившееся расширение для браузера Opera под названием **Opera Unite** [3] [4], предоставляющее полноценный веб сервер прямо в браузере для запуска пользовательских приложений. В кратце, что такое **Opera Unite**:

Opera Unite — расширение браузера Opera [5] [6], с помощью которого можно, например, слушать музыку и смотреть видео в потоковом режиме, делиться фотографиями и другими файлами прямо со своего компьютера, посредством предоставления доступа к определённой директории (и поддиректориям) в файловой системе. И, что самое важное, все это делается напрямую, без участия какого-либо сервера. Еще одна из особенностей **Opera Unite** является то, что она может работать и после закрытия браузера, если только пользователь принудительно ее не отключит, что позволяет приложениям работать в пассивном режиме и продолжать обрабатывать запросы.

При создании приложений для этой платформы используются технологии **HTML**, **CSS**, **JavaScript** [7], **SVG** и **AJAX** [8], а также можно использовать библиотеку шаблонов **Markuper API** [9], которая позволяет связать **JavaScript** и **HTML**, облегчив тем самым процесс создания интерфейса приложения.

Подытожим важные для нас преимущества технологии **Opera Unite**:

- Простота установки: пользователям Opera нужно лишь установить приложение для **Opera Unite**, что немного проще чем устанавливать десктопное приложение.

- Мы получили готовую основу для просмотра и навигации по файлам и папкам в виде встроенного по умолчанию в Opera Unite приложению «***File Sharing***» [10] вместе с исходными кодами.

5. Структура системы распределённого поиска.

5.1. Прототип

В качестве первой задачи была выбрана реализация простого прототипа системы.

Схема распределённого поиска в прототипе была такой — есть граф, где вершины — компьютеры с установленным приложением, и у каждой вершины есть список адресов некоторых других компьютеров — список смежных рёбер.

Поиск основан на обходе этого графа в глубину — на отдельном компьютере осуществляется поиск по локальным файлам и процедура повторяется для соседних вершин. Для того, чтобы поиск не зацикливался результаты каждого поиска сохраняются.

5.2. Минусы реализации

1. Любой запрос “нагружает” каждое ребро.
2. Так как запрос поиска для соседних вершин осуществляется асинхронно, то хорошим показателем времени работы служит максимальная глубина поиска. При такой реализации это никак не учитывается.
3. Проблема с нахождением “соседей”. Граф никак не меняется и пользователю приходится самому прописывать смежные вершины

Прототип служил базой для построения других систем.

5.3. Возможные оптимизации

- *Ограничение на глубину поиска или на количество обработанных вершин.* В этом случае появится возможность контролировать «глубину» удалённых вызовов – основной показатель времени обработки запроса.
- “*Оптимизация” графа*. Со временем можно производить локальные изменения конфигурации графа - например, чтобы увеличивать связность графа (повышение устойчивости к отключению отдельных компьютеров), уменьшение диаметра и т. п.

- *Индексация*. Для осуществления быстрого поиска файлов на отдельно взятом компьютере необходимо произвести некоторый предподсчет и построить некоторую подходящую для этого структуру данных. Этот процесс в общем случае называется индексацией.
- *Распространение индекса поиска на соседние вершины*. ***** Так чтобы информация о локальных файлах хранилась не только у одного компьютера.

5.4. Реализация

Изучив и обдумав некоторые подходы к решению задачи распределенного поиска, мы решили остановиться на очень простой, но довольно эффективной модели. Она заключается в разбиении компьютеров поисковой сети на группы, осуществляемом специальным сервером. Каждый компьютер хранит поисковой индекс всех компьютеров из своей группы, а так же список представителей других групп, по одному на каждую группу. При получении поискового запроса приложение производит поиск по имеющемуся у нее набору идексов компьютеров своей группы, а так же отсылает запрос представителям других групп. В результате поиск затронет около корня из N(N – общее число компьютеров сети) компьютеров, но при этом охватит все файлы сети.

В сетях малого и среднего размера данная схема будет работать достаточно эффективно.

5.5. Логика приложения при старте

При старте приложения первым делом происходит переиндексация файлов компьютера. Затем приложение обращается на сервер передавая свой адрес, сервер назначает этот компьютер в определенную группу и в ответ присылает два списка: список адресов всех компьютеров текущей группы и список адресов представителей всех других групп(представители выбираются случайным образом). Далее приложение скачивает индекс со всех компьютеров своей группы и раздает в свой индекс так же все группе.

Замечание: в процессе разаботки была обнаружена проблема выключенных звеньев: т. е. если пользователь выключит компьютер А, который являлся представителем своей группы для компьютера Б, то у компьютера Б не будет доступа к индексу компьютеров всех группы компьютера А. Одно из решений — при отсутствии соединения с представителем группы запрашивать нового представителя у сервера, но тогда, учитывая довольно большой процент постоянно выключающихся из сети звеньев, нагрузка на сервер будет неприемлимо большой. В результате эта проблема решена не была и до сих пор остается открытой.

5.6. Индексация

Создание хорошего алгоритма индексация не входило в цели работы, поэтому была использована самая простая структура данных – просто список всех доступных файлов. Существует множество методов индексации данных и любой из них можно с легкостью встроить в наше приложение. Заметим, что это направление направление развития приложения является одним из приоритетных и может значительно улучшить скорость поиска.

5.7. Сервер

Сервер выполняет роль менеджера групп. Его задача на запрос регистрации в сети нового компьютера поместить этот компьютер в некоторую группу и вернуть два списка: список компьютеров этой группы и список представителей друих групп. Нагрузка на сервер получается минимальной, т.к. сам собственно поиск сервер никак не затрагивает.

Сервер был реализован с использованием технологии Java Servlet, которая позволяет разработать простой веб сервер довольно быстро.

6. Итог

В приложение "File Sharing" была добавлена возможность полноценного распределённого поиска по файлам компьютеров одной сети. Система устойчива к отключению из сети отдельных компьютеров и даже сервера (в этом случае всего лишь теряется возможность подключения новых компьютеров). Однако мы никак не затронули проблему безопасности.

7. Направления дальнейшего развития приложения

- Стабильность
- Тестирование. Для доказательства работоспособности системы, необходимо протестировать ее на достаточно большом количестве компьютеров. В качестве одного из вариантов распространения приложения хорошо бы выложить его на официальном сайте
- Полнотекстовый поиск
- Улучшенная индексация
- Передача индекса peer-to-peer
- Доработка интерфейса

8. Список литературы:

1. Majectic 12 // Официальный сайт проекта URL : http://www.majestic12.co.uk/ (дата обращения: 03.06.2010)
2. Интервью с создателем Majestic 12 // Сайт Distributed Computing Team URL: http://distributed.org.ua/index.php?go=Pages&in=view&id=172 (дата обращения: 03.06.2010)
3. Opera Unite // Официальный сайт Opera Unite URL : http://unite.opera.com/ (дата обращения: 03.06.2010)
4. Opera Unite // Пресс релиз запуска Opera Unite URL : http://www.opera.com/press/releases/2009/06/16 (дата обращения: 03.06.2010)
5. Браузер Opera // Официальный сайт Opera URL : http://ru.wikipedia.org/wiki/Opera (дата обращения: 03.06.2010)
6. Браузер Opera // Статья на Википедии URL : http://ru.wikipedia.org/wiki/Opera (дата обращения: 03.06.2010)
7. JavaScript // Статья на Википедии URL : http://ru.wikipedia.org/wiki/JavaScript (дата обращения: 03.06.2010)
8. Ajax // Статья на Википедии URL : http://ru.wikipedia.org/wiki/AJAX (дата обращения: 03.06.2010)
9. Markuper API // |Статья на официальном сайте Opera URL : http://dev.opera.com/libraries/markuper/ (дата обращения: 03.06.2010)
10. «File Sharing» - Opera Unite Application // Официальный сайт Opera Unite URL http://unite.opera.com/application/132/ (дата обращения: 03.06.2010)