электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

УДК 004.021

Основные показатели в веб-аналитике и их расчёт с помощью MapReduce

***Липкин В. Н.**, студент*
Россия, 105005, г. Москва, МГТУ им. Н. Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»

Научный руководитель: Рудаков И.В., к.т.н., доцент
Россия, 105005, г. Москва, МГТУ им. Н. Э. Баумана,
кафедра «Программное обеспечение ЭВМ и информационные технологии»
irudakov@bmstu.ru

Введение

Работу любого приложения можно оценить по промежуточным данным. Эта данные содержат информацию по различным параметрам, показывающим статус окончания и успешности процесса на текущий момент. Веб-сайт, если рассматривать его с точки зрения серверного приложения, не является исключением — для его оценки необходимо анализировать различные характеристики посещаемости. Для этого существует особый раздел поискового маркетинга, который именуется веб-аналитикой. С помощью комплекса мер можно получить данные о посещаемости веб-сайта за любой момент и провести по ним глубокий анализ[1].

Одним из возможных источников данных для анализа работы серверных приложений являются их лог-файлы, в частности, для веб-серверов можно использовать так называемый лог доступа (access-log). Основываясь на них, можно рассчитать большое количество статистических показателей.

С развитием Интернета и ростом посещаемости сайта растет и объем логов доступа, что, в конечном счете, приводит к невозможности их обработки на одном компьютере за адекватное время. В этом случае спасают либо приближенные методики расчетов, либо распределенные вычисления. Модель распределеных вычислений MapReduce хорошо подходит для решения описанной задачи.

Результат расчета, в свою очередь, необходимо сделать человекочитаемым. При исследовании простейших зависимостей (например, [дата → количество посетителей]) для

этого не надо прилагать специальных усилий. Но при добавлении в исследование дополнительных измерений и показателей результат расчета может возрасти до такой степени, что человек просто не сможет воспользоваться таким результатом. В таком случае целесообразно воспользоваться дополнительными системами визуализации и пост-обработки результатов расчета. В связи с универсальностью в применении результата расчета для вышеописанных задач, в статье будет рассмотрено получение данных в виде OLAP-куба.

Общая постановка задачи

Расчетом будем называть последовательность операций Map (равномерное распределение записей между экземплярами программы обработки без гарантий их порядка) и Reduce (группировка записей по ключу и их распределение в сгруппированном виду между экземплярами программы), в комплексе дающих данные, отвечающие на поставленный вопрос аналитики.

На входе расчета имеем один или более лог-файлов, содержащих информацию за некоторый фиксированный временной период. В статье будет рассмотрен только лог доступа, в общем случае имеющий следующие поля:

1. **ip** — IP-адрес, с которого произведён запрос к веб-серверу;
2. **datetime** — время запроса к серверу и часовой пояс сервера;
3. **http_status** — код состояния HTTP;
4. **size** — количество отданных сервером байт;
5. **referer** — URL-источник запроса;
6. **user_agent** — HTTP-заголовок, содержащий информацию о запросе (клиентское приложение, язык и т. д.);
7. **vhost** — имя Virtual Host, к которому идет обращение;
8. **request** — адрес страницы и параметры http;
9. **cookies** — небольшой фрагмент данных, отправленный веб-сервером и хранимый на компьютере пользователя; в данном случае пользователь отдает их веб-серверу при обращении.

Каждую запись из лога доступа представим в виде кортежа (<ip>, <datetime>, <http_status>, <size>, <referer>, <user_agent>, <vhost>, <request>, <cookies>).

На выходе расчета имеем OLAP-куб, многомерный массив данных, индексами которого являются измерения куба, а значениями элементов массива — меры куба:

$$w: (x,y,z) \rightarrow w_{xyz},$$

где x, y, z — измерения, w — мера. Исходя из поставленной задачи, в результирующем кубе мерами будут являться статистические показатели веб-сайта, а измерениями — некоторые параметры сегментации этих показателей. В вырожденном случае куб может не иметь измерений и содержать в себе один показатель в одном экземпляре.

Основные показатели в веб-аналитике

Перечислим основные элементы, участвующие в веб-аналитике[3].

• **Хит** — это запрос к веб-серверу для получения файла (веб-страницы, изображения, JavaScript'a, таблицы стилей и т.д.). Когда страница загружена с сервера, то число "хитов", или "хитов страницы", равно числу запрошенных файлов, поэтому одна загруженная страница не всегда равна одному хиту, потому что часто страницы составлены из изображений и других файлов, которые влияют на подсчёт числа хитов[2]. Будем исходить из того, что одна запись лога доступа соответствует одному хиту.

• **Просмотр страницы** — это хит получения файлов веб-страниц.

• **Посетитель** — это человек, сделавших хотя бы один хит на веб-сайте. При расчете данного показателя встает проблема идентификации посетителя, которую можно решить с помощью установки cookie на браузер пользователя, содержащий идентификатор пользователя с высокой степенью уникальности, например, конкатенацию времени установки cookie и случайного числа.

• **Визит (сессия)** — сеанс взаимодействия посетителя с сайтом, включающий один и более просмотров страницы. Более конкретное определение визита выбирается в зависимости от предметной области и задачи, например, визит можно определить как последовательность хитов одного посетителя, совершенных с промежутками, не превышающими 30 минут.

• **Показ** — показ конкретного элемента веб-страницы. В простейшем случае, факт показа можно однозначно согласовать с просмотром веб-страницы, что может оказаться невыполнимым в случае динамической веб-страницы. В таком случае, необходимо организовать отдельный лог показов.

• **Клик** — нажатие мышью на некоторый элемент страницы, в частности, на гиперссылку или рекламный баннер. Факт клика можно установить по логу доступа веб-сайта, на который ведет гиперссылка посредством добавления в ссылку специального http-

параметра с признаком конкретной ссылки. Если это невозможно, следует завести лог кликов.

Отсюда получим базовые показатели: количество просмотров страниц, количество посетителей и т.д.

Рассмотрим следующие возможные измерения для количества хитов, просмотров, показов, кликов:

- дата совершения действия;
- географическое местоположение пользователя в момент совершения действия;
- адрес веб-страницы, на которой совершено действие;
- адрес источника запроса;
- HTTP-статус запроса;
- различные параметры веб-браузера пользователя в момент совершения действия (название, версия браузера) и т.д.

Можно заранее определить область допустимых значений для каждого из измерений исходя из аналитических потребностей. Хиты, клики и просмотры, которые имеют свойства, выходящие за рамки области допустимых значений, простым образом исключаются из расчета. Будем называть этот процесс **фильтрацией**.

Перечислим важные фильтрации, обычно не зависящие от характера задачи:

• Фильтрация по HTTP-статусу (например, можно оставлять только хиты со статусами 200 и 304);

• Фильтрация по странице (отделение просмотров веб-страниц от всех остальных хитов);

• Фильтрация по user-agent (исключение хитов, совершаемых поисковыми роботами).

Сегментация количества посетителей и визитов происходит исходя из сегментации действий, которые к ним принадлежат. Таким образом, один посетитель может одновременно относиться к разным сегментам, например, за одну дату он мог совершать хиты из разных городов.

Визит также имеет очень важный параметр сегментации — время визита.

Основываясь на перечисленных показателях и возможных сегментациях, можно получить сколь угодно сложные производные показатели посредством следующих приемов:

• расчета среднего значения показателя по одному из измерений;

• расчета перцентилей значения показателя по одному из измерений;

• расчета отношения одного показателя к другому.

Перечислим наиболее важные производные показатели:

• возвращаемость — среднее количество дней в месяце, в которые посетители совершают доступ к веб-сайту;

• конверсия — отношение посетителей, совершивших некие целевые действия (покупку, регистрацию, подписку), ко всем посетителям;

• CTR (click-through rate, показатель кликабельности) — отношение количества кликов на рассматриваемый элемент веб-страницы к количеству показов этого элемента, в простейшем случае — к количеству просмотров страницы;

• средняя продолжительность визита.

Рис.1. Простой расчет количественных показателей

Расчет показателей с помощью MapReduce

Рассмотрим расчет простого количественного показателя, показанный на рис. 1. Первая стадия расчета, Map Access Log, принимает на вход записи лога доступа и выполняет следующие функции:

• фильтрация ненужных записей;

• выделение из записи лога параметров (x, y, z) хита, которые по совместительству являются значениями измерений результата расчета;

• преобразование записи лога в запись с составным ключом, содержащим параметры (x, y, z) хита, и пустым значением.

Вторая стадия расчета, Reduce Sum, принимает на вход сгруппированные записи с ключом (x, y, z) и для каждого такого ключа считает количество записей. На выходе этой стадии для каждого ключа имеем один элемент OLAP-куба.

При проектировании подобного расчета следует обратить внимание на следующие вещи:

- Проверки записей на попадание под фильтрацию занимают много времени: «нужная» запись подвергнется всем проверкам, «не нужная» — только части. Поэтому следует разместить проверки в порядке доли фильтруемого входа.
- Для различных ключей количество записей может различаться, причем существенно, что может привести к тому, что отдельные экземпляры Reduce-операции займут больше времени, чем остальные. Этого можно избежать с помощью кеширования записей в Map Access
- Log (см. рис. 2) посредством промежуточного подсчета количества записей с данными значениями измерений внутри экземпляра Map-операции.
- Кеширование необходимо организовать таким образом, чтобы не произошло переполнения оперативной памяти внутри экземпляра операции.
- Кеширование также позволяет уменьшить размер промежуточных данных.

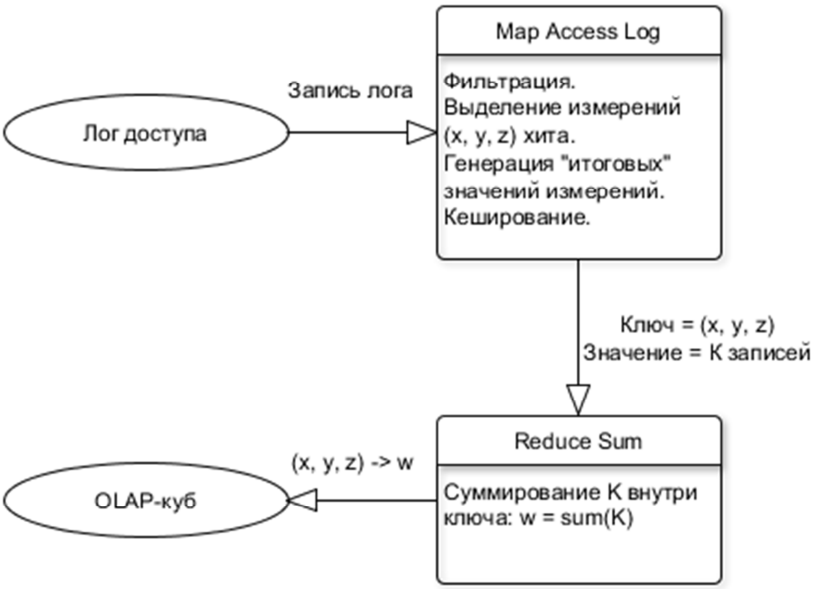

Рис. 2. Расчет количественных показателей с применением кеширования и подсчетом «итоговых значений»

Для некоторых измерений может потребоваться посчитать итоговое значение, например: «просмотров из Ростова», «просмотров из Москвы», «просмотров из Ростова и Москвы». Опустим то, что подобную величину легко посчитать на основе уже полученного OLAP-куба в силу аддитивности количества просмотров. Для решения подобной задачи достаточно генерировать в Map Access Log на каждую запись с одним из двух простых значений измерения вторую запись с подобным «итоговым» значением измерения (см. рис. 2), что, в свою очередь, приведет к опасному различию между количеством записей на уникальный ключ, если не использовать кеширование в первой стадии.

Покажем, как необходимо модифицировать расчет для подсчета количества уникальных посетителей (см. рис. 3):

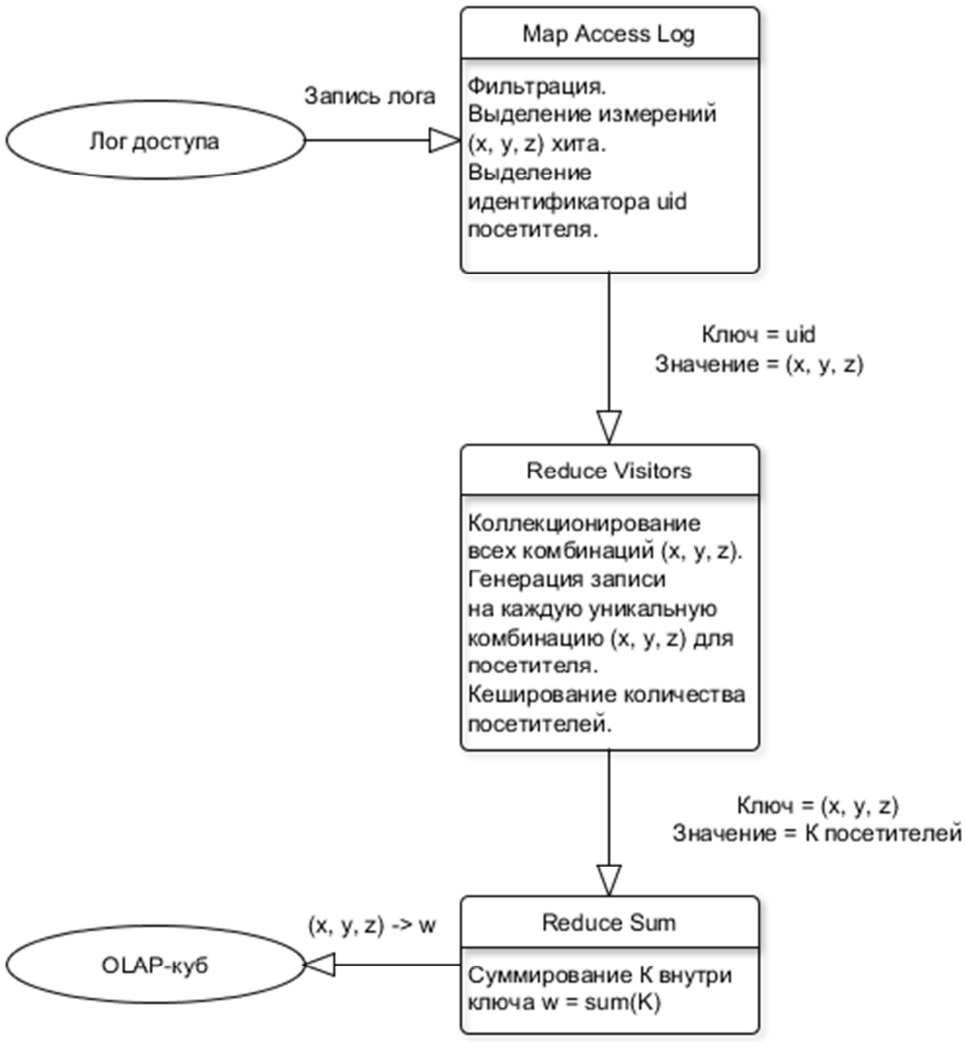

Рис. 3. Расчет количества посетителей

1. Map Access Log на выходе теперь помещает в ключ идентификатор пользователя, а значения измерений — в значение.

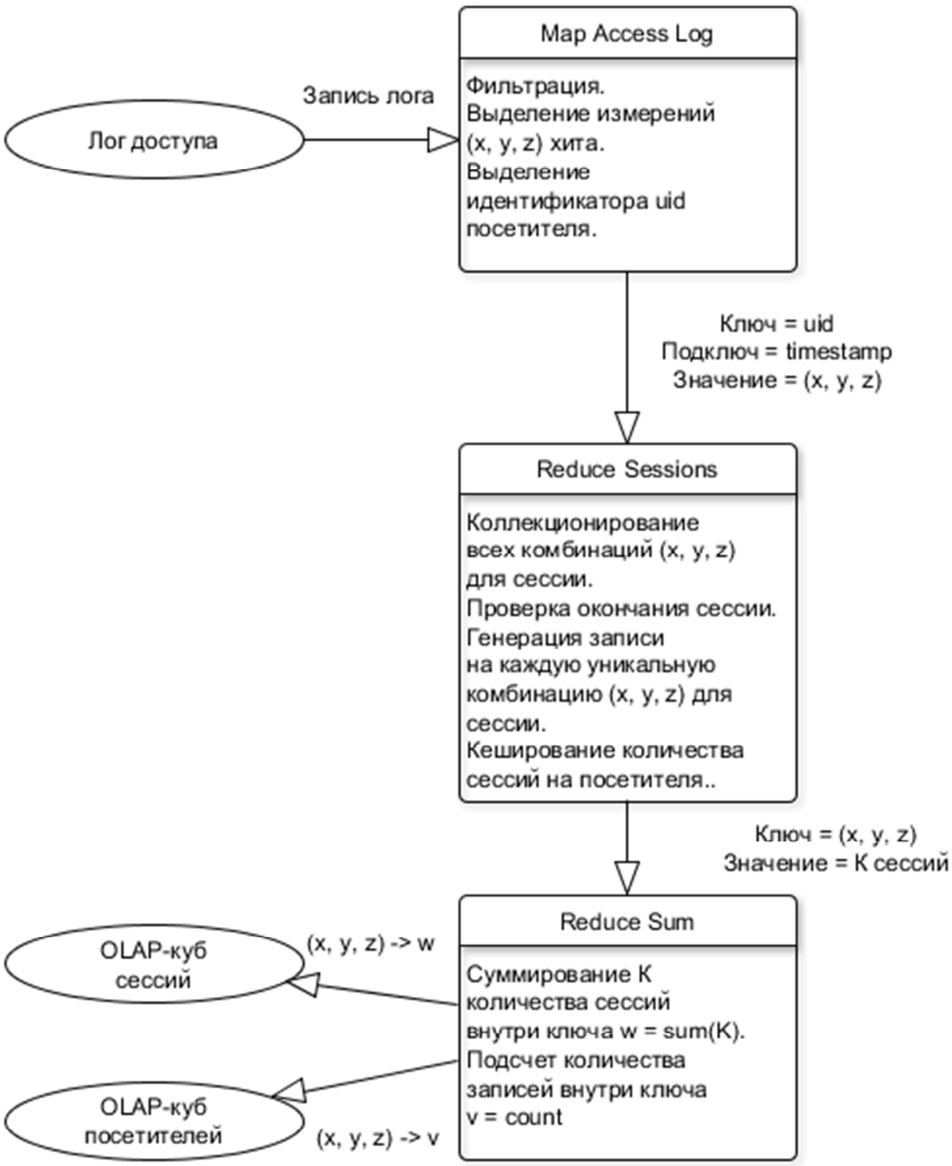

Рис. 4. Комбинированный подсчет количества сессий и посетителей

2. Добавляется стадия Reduce Visitors, имеющая на входе записи, сгруппированные по одному посетителю. Здесь идет поиск всех возможных комбинаций измерений для данного

посетителя. После обработки всех записей, относящихся к посетителю, на каждую найденную комбинацию измерений генерируется по записи с ключом (x, y, z) ,подобно предыдущим расчетам.

3. Во избежание перекоса по ключам в третьей стадии, введем в Reduce Visitors кеширование записей с одинаковым ключом посредством промежуточного подсчета количества записей с данными значениями измерений внутри экземпляра Reduce-операции.

Подсчет уникальных посетителей имеет ключевое отличие от подсчета аддитивных величин, а именно то, что посетитель может принадлежать сразу нескольким значениям одного измерения. Это влияет на подсчет «итоговых» значений: их нельзя получить простым суммированием, как при подсчете количества хитов. Так, в Reduce Visitors есть множество всех комбинаций измерений посетителя. Для подсчета итогового значения по измерению x необходимо на каждую комбинацию (x, y, z) добавить в множество комбинацию (“Total”, y, z). Использование множества обеспечит уникальность и посетитель засчитается в «итоговое» значение измерения один раз.

Наконец, рассмотрим подсчет количества сессий (см. рис. 4). Для этого введем понятие подключа и гарантируем, что на вход Reduce-операции все записи одного ключа будут сортированы по подключу. Тогда поместим на выходе Map Access Log в подключ правильно сортируемое время совершения хита. В Reduce Sessions при обработке первой записи будем считать, что обрабатываем первую сессию посетителя, и начнем коллекционировать все комбинации измерений для этой сессии. При обработке каждой следующей записи в стадии будем проверять разницу с значением времени предыдущей записи и при превышении установленного лимита будем считать, что сессия закончилась.

Несложно комбинировать расчеты базовых показателей, покажем это на примере сессий и посетителей (см. рис. 4). При завершении сессии будем инкрементировать счетчик комбинаций для соответствующих комбинаций посетителя. При окончании обработки посетителя сгенерируем множество записей на каждую уникальную комбинацию, в значении которых будет находиться количество сессий. В Reduce Sum сумма значений внутри одного ключа будет представлять собой количество сессий, а количество самих записей — количество посетителей.

Важным этапом разработки программного обеспечения является задача анализа и контроля правильности функционирования системы на ранних этапах ее разработки[4]. В приложении к расчетам, основанным на больших данных, этому следует уделить большое

внимание, т.к. цена ошибки может быть крайне высокой из-за стоимости эксплуатации кластерного оборудования.

Заключение

В данной статье рассмотрены некоторые особенности проведения расчётов показателей веб-аналитики с помощью MapReduce. Предложена оптимизация алгоритма расчёта, в том числе с использованием кеширования. Показаны основные приёмы составления расчётов с примерами.

Комбинируя рассмотренные подходы, можно посчитать множество производных и более сложных показателей. При расчете конверсий и CTR достаточно посчитать два отдельных показателя (делитель и делимое) с помощью показанных приемов и в результирующей стадии расчета поделить одно на другое.

В реальности совершенно необязательно ограничиваться логом доступа при расчетах: можно пересекать логи по посетителям, присутствующим в двух и более логах; пересекать показы элементов веб-страницы и клики по ним по идентификатору показа. Также в реальности остро стоит вопрос очистки логов от роботов и «накруток», которая решается посредством введения эвристических правил фильтрации.

Список литературы

1. Максим Довженко. WorkFormation. Режим доступа: http://www.workformation.ru/ocenka-i-analiz-poseshhaemosti-sajta.html (дата обращения 27.03.2015).
2. Vanie Beal, Forrtst Stroud. Webopedia. IT Buisiness Endge Network. Режим доступа: http://www.webopedia.com/TERM/H/hit.html (дата обращения 27.03.2015).
3. Яковлев А., Довжиков А. Веб-аналитика: основы, секреты, трюки. СПб.: БХВ-Петербург, 2010. 272 с.
4. Рудаков И.В. Методика иерархического исследования сложных дискретных структур. Наука и образование: электронное научно-техническое издание. 2012. № 6. Режим доступа: http://technomag.bmstu.ru/doc/370230.html (дата обращения 01.04.2015).