

Санкт-Петербургский Академический университет

Дедупликация данных, хранящихся на блочных
устройствах, с использованием подхода
extreme binning

Студент: Антон Афанасьев

Руководитель: Баталов Е. А.

2015 г.

Метод сжатия данных при помощи исключения копий повторяющихся данных.

Стандартная процедура дедупликации

- ▶ Файлы разбиваются на блоки
- ▶ От каждого блока вычисляется хэш
- ▶ Блоки считаются одинаковыми, если хэш совпадает
- ▶ Хэши складываются в индекс
- ▶ Уникальные блоки записываются в хранилище

Основная проблема: индекс становится очень большим, в оперативную память не помещается.

Располагать индекс на диске медленно.

Extreme binning

Алгоритм дедупликации, предложенный D. Bhagwat et al.

Основные идеи:

- ▶ Используются два вида индекса: главный индекс в памяти и много маленьких индексов на диске (корзины)
- ▶ Из каждого файла выбирается “главный” блок, который определяет корзину
- ▶ Соответствующая корзина поднимается с диска, и хэши добавляются в нее
- ▶ Главный блок выбирается так, чтобы с большой вероятностью назначать похожие файлы в одинаковые корзины
- ▶ На сохранение файла требуется одно обращение к диску

Требует мало памяти, дедупликация быстрая.

Дедупликация не идеальная (возможны повторяющиеся блоки).

Дедупликация на блочных устройствах

Extreme binning активно использует информацию о разбиении на файлы.

Иногда хочется дедуплицировать блочное устройство без знания о файловых системах.

- ▶ Поддерживать много разных файловых систем дорого
- ▶ Не всегда известно какая ФС используется
- ▶ Работа через файловую систему медленнее, чем без нее

Extreme binning показывает хорошие результаты.

Если перенести его на блочные устройства, возможно, там он тоже будет работать хорошо.

Цель

Адаптировать алгоритм extreme binning для дедупликации данных, хранящихся на блочных устройствах

Задачи

1. Найти способ разбиения данных на сегменты при отсутствии файлов
2. Написать приложение для тестирования extreme binning и его модификаций с различными параметрами
3. Провести оценку работы подходов на реальных данных

Разбиения на файлы нет, значит, нужно разбить на сегменты.

Используем подходы для разбиения на блоки

- ▶ Сегменты фиксированного размера
- ▶ Размер сегмента определяется содержимым
 - ▶ якорное сегментирование — в скользящем окне вычисляем хэш, когда хэш делится на заданную константу заканчиваем сегмент.
 - ▶ TTTD (два порога, два делителя) — то же, плюс устанавливаем минимальный и максимальный размеры сегмента. При достижении верхней границы на размер пытаемся переразбить с меньшим делителем.

Приложение для тестирования алгоритмов дедупликации

1. Дедупликация данных

Программа на C++

- ▶ Принимает архивы с хэшами с размеченными границами файлов
- ▶ Производит дедупликацию выбранным методом, собирая статистику о работе метода
- ▶ Генерирует json-файл с собранной статистикой

2. Анализ результатов

Программа на python

- ▶ Принимает json от основной программы
- ▶ Вычисляет производные статистики, строит гистограммы
- ▶ Генерирует отчет в Latex

Фреймворк для дедупликации

1. Разбиение на блоки
 - ▶ С учетом границ файлов
 - ▶ Без учета границ файлов
2. Разбиение на сегменты
 - ▶ Сегмент — файл
 - ▶ Сегменты фиксированного размера
 - ▶ Якорное сегментирование
 - ▶ Якорное сегментирование с двумя делителями
3. Дедупликация
 - ▶ Extreme binning
 - ▶ Файловая дедупликация
 - ▶ Идеальная блочная дедупликация

Реализация Extreme binning

- ▶ Хранилище корзин вторичного индекса на основе файлов
- ▶ Алгоритм выбора главного блока в сегменте — $\text{minhash}(k)$
- ▶ Сбор статистики
 - ▶ качество дедупликации (коэффициент дедупликации)
 - ▶ размеры индекса в памяти, индексов на диске
 - ▶ распределение размера корзин, размера сегментов, популярности корзин
 - ▶ распределение средней доли переиспользования корзин
 - ▶ количество загрузок корзин с диска
 - ▶ количество дубликатов сегментов, количество обнаруженных дубликатов сегментов
 - ▶ доля корзин запрашиваемых последовательно

Резервные копии облачного хранилища компании Acronis.
Общий объем данных порядка 3Тб, $\approx 11 \cdot 10^6$ файлов.

Максимальный размер файла: 40.4Гб

Медиана: 8Кб

Коэффициент дедупликации — отношение объема данных до и после дедупликации

Коэффициент файловой дедупликации: 1.4313

Коэффициент идеальной дедупликации: 1.731066

Результаты тестирования

Средний размер сегмента: 512Кб

Метод сегментирования	Количество корзин	Коэффициент дедупликации	Среднее использование корзины
по файлам	5178926	1.56569	0.57294
фикс. размер	4383838	1.32452	0.74210
якорное	3709741	1.52890	0.67128
два делителя	3709789	1.52886	0.67336

Сегменты фиксированного размер показывают плохое качество дедупликации. Два делителя ничего не дают в нашем случае.

Методы, опирающиеся на содержимое сегмента, достигают качества дедупликации 88% от идеальной, 96% от обычного extreme binning.

- ▶ Найдены эффективные способы разбиения данных на сегменты при отсутствии файлов
- ▶ Написана программа для тестирования базовых алгоритмов дедупликации, extreme binning и его модификаций
- ▶ Алгоритмы протестированы на реальных данных
- ▶ Подход extreme binning успешно перенесен на блочные устройства и показал хорошие результаты

Спасибо за внимание!