

ДЕЛЬТА-КОДИРОВАНИЕ - DELTA ENCODING



ДЕЛЬТА-КОДИРОВАНИЕ - ЭТО СПОСОБ ХРАНЕНИЯ ИЛИ ПЕРЕДАЧИ ДАННЫХ В ВИДЕ РАЗЛИЧИЙ (ДЕЛЬТ) МЕЖДУ ПОСЛЕДОВАТЕЛЬНЫМИ ДАННЫМИ, А НЕ ПОЛНЫМИ ФАЙЛАМИ;
В БОЛЕЕ ОБЩЕМ СМЫСЛЕ ЭТО НАЗЫВАЕТСЯ ДИФФЕРЕНЦИРОВАНИЕМ ДАННЫХ. ДЕЛЬТА-КОДИРОВАНИЕ ИНОГДА НАЗЫВАЮТ ДЕЛЬТА-СЖАТИЕМ, ОСОБЕННО КОГДА ТРЕБУЕТСЯ АРХИВИРОВАНИЕ ИСТОРИИ ИЗМЕНЕНИЙ (НАПРИМЕР, В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ ДЛЯ КОНТРОЛЯ ВЕРСИЙ)

ПРОСТОЙ ПРИМЕР ДЕЛЬТА-КОДИРОВАНИЕ

Возможно, самый простой пример - это сохранение значений байтов в виде разностей (дельт) между последовательными значениями, а не самих значений.

Итак, вместо 2, 4, 6, 9, 7 мы будем хранить 2, 2, 2, 3, -2. Это уменьшает дисперсию (диапазон) значений при коррелировании соседних выборок, позволяя использовать меньшее количество бит для одних и тех же данных. Формат звука IFF 8SVX применяет это кодирование к необработанным звуковым данным перед применением к ним сжатия.

К сожалению, даже не все 8-битные звуковые сэмплы лучше сжимаются при дельта-кодировании, а удобство использования дельта-кодирования еще меньше для 16-битных и лучших сэмплов. Поэтому алгоритмы сжатия часто выбирают дельта-кодирование только тогда, когда сжатие лучше, чем без него. Однако при сжатии видео дельта-кадры могут значительно уменьшить размер кадра и используются практически в каждом кодеке сжатия видео .

ОПРЕДЕЛЕНИЕ

Дельта может быть определена двумя способами: симметричная дельта и направленная дельта. Симметричная дельта может быть выражена как

$$\Delta(v_1, v_2) = (v_1 \setminus v_2) \cup (v_2 \setminus v_1),$$

где v_1 и v_2 представляют две версии. Направлена дельта, которая также называется изменением, представляет собой последовательность операций (элементарные) изменений, которые, при применении к одной версии, дают другую версию (обратите внимание на соответствие журналы транзакций в базах данных). В компьютерных реализациях они обычно принимают форму языка с двумя командами: копировать данные из v_1 и записывать буквальные данные v_1 в v_2 .

ПРОБЛЕМЫ РЕАЛИЗАЦИИ ДЕЛЬТА-КОДИРОВАНИЕ

Природа данных, которые должны быть закодированы, влияет на эффективность конкретного алгоритма сжатия. Дельта-кодирование работает лучше всего, когда данные имеют небольшие или постоянные вариации; для несортированного набора данных сжатие с помощью этого метода может быть незначительным или невозможным. При дельта-кодированной передаче по сети, где на каждом конце канала связи доступна только одна копия файла, используются специальные коды контроля ошибок, чтобы определить, какие части файла были изменены с момента его предыдущей версии. Например, rsync использует алгоритм скользящей контрольной суммы, основанный на контрольной сумме Марка Адлера adler-32

ПРИМЕР КОДА С ДЕЛЬТА-КОДИРОВАНИЕ СЛЕДУЮЩИЙ КОД С ВЫПОЛНЯЕТ ПРОСТУЮ ФОРМУ ДЕЛЬТА-КОДИРОВАНИЯ И ДЕКОДИРОВАНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ СИМВОЛОВ

```
void delta_encode(unsigned char *buffer, int length)
{
    unsigned char last = 0;
    for (int i = 0; i < length; i++)
    {
        unsigned char current = buffer[i];
        buffer[i] = current - last;
        last = current;
    }
}

void delta_decode(unsigned char *buffer, int length)
{
    unsigned char last = 0;
    for (int i = 0; i < length; i++)
    {
        unsigned char delta = buffer[i];
        buffer[i] = delta + last;
        last = buffer[i];
    }
}
```

Д/З

ДЕЛЬТА-КОДИРОВАНИЕ В HTTP, ДЕЛЬТА-КОПИРОВАНИЕ,
ДЕЛЬТА-КОДИРОВАНИЕ,
ОНЛАЙН-РЕЗЕРВНОЕ КОПИРОВАНИЕ,
ДЕЛЬТА-ОБНОВЛЕНИЯ, DIFF, GIT ,
VCDIFF, GDIFF, BSDIFF