

Reinventing Infobright's Concept of Rough Calculations on Granulated Tables for the Purpose of Accelerating Modern Data Processing Frameworks

Mateusz Wnuk



Infobright



Analizyczny silnik bazodanowy stworzony w 2005 roku, udostępniony w wersji open-source - ICE (Infobright Community Edition).

- granularyzacja danych
- zaimplementowana idea zbiorów przybliżonych
- zminimalizowana potrzeba dostępu do oryginalnych danych
- szybka i wydajna kompresja i dekompresja

Infobright – koncepcja granularyzacji danych

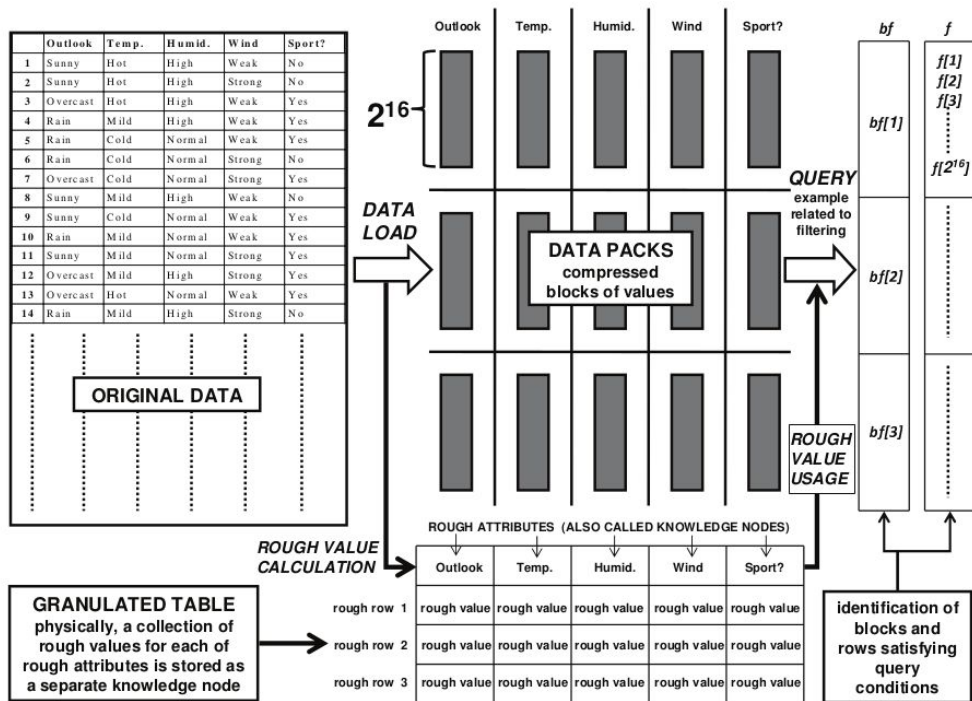


Infobright logicznie dzieli ładowane dane na równej wielkości paczki danych, dla których liczone są podsumowania.

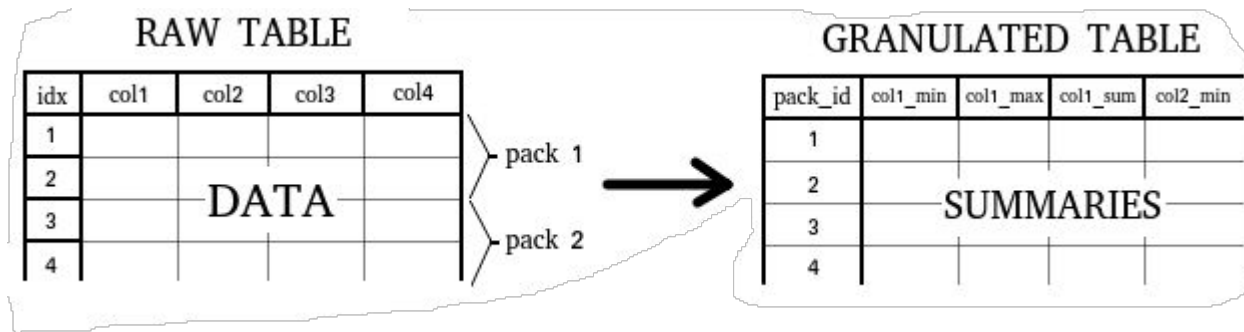
Głównym celem zaimplementowanej zgranulowanej tabeli jest zminimalizowanie potrzeby przetwarzania oryginalnych danych, w celu przyspieszenia pewnych operacji analitycznych.

Ważne kwestie w przypadku generowania zgranulowanej tabeli to wielkość paczki oraz rodzaj przechowywanych podsumowań.

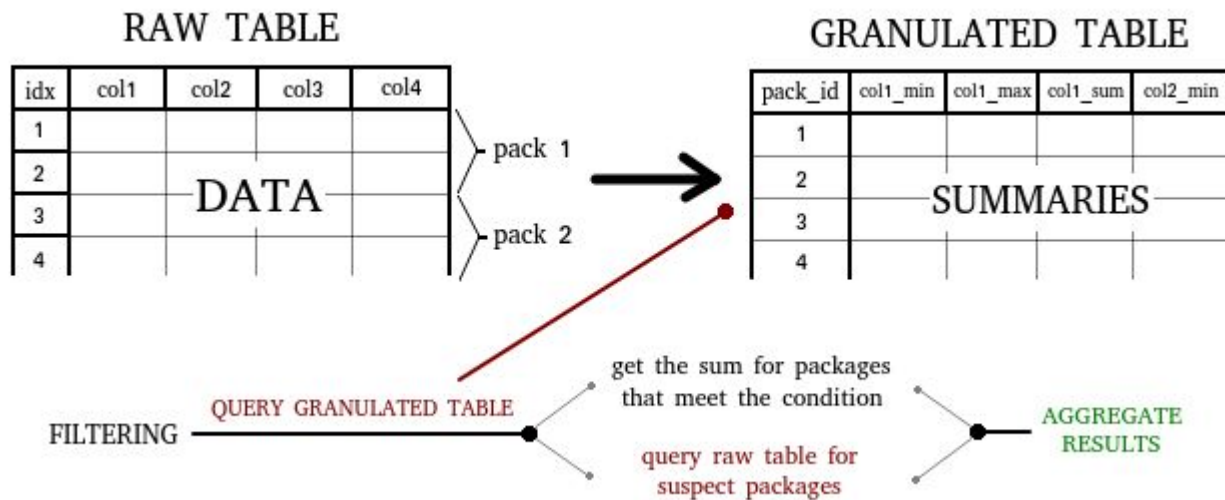
Infobright



Infobright – procesowanie zapytania



Infobright – procesowanie zapytania



Infobright – procesowanie zapytania

Wstępna analiza prowadzi do zidentyfikowania trzech zestawów paczek:

- te, które spełniają warunek w całości (1)
- te, które nie spełniają warunku (2)
- oraz te, które spełniają warunek w części i muszą zostać przeliczone z oryginalnych danych (3)

Idealna sytuacja jest wtedy, gdy zapytanie odfiltrowuje jedynie paczki z grupy (1) i (2).

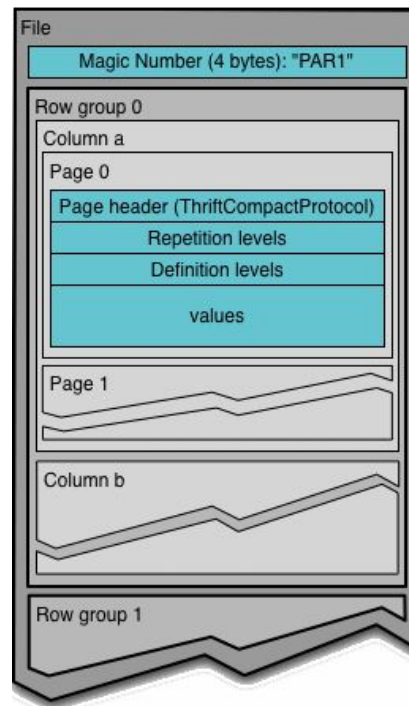
Rozmiar paczek wpływa na:

- podział paczek po odfiltrowaniu (FULL/NEGATIVE/SUSPECT)
- rozmiar podsumowań
- szybkość i wydajność kompresji

Apache Parquet

Parquet to format zapisu danych do których dostęp jest zapewniany przez Apache Arrow.

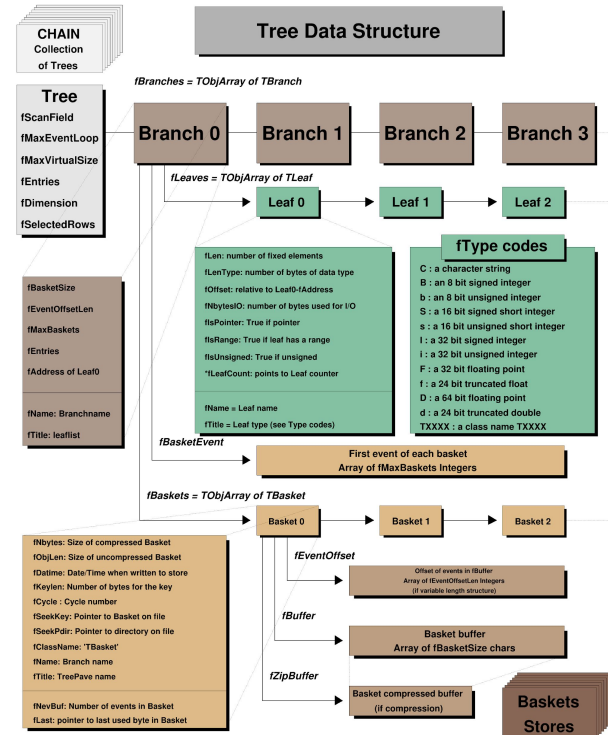
- kolumnowa reprezentacja danych
- wspiera różne typy kompresji
- dzieli dane na paczki i umożliwia dostęp do wybranej paczki bez konieczności czytania całego pliku



ROOT

ROOT jest bardzo złożonym frameworkiem do przechowywania i przetwarzania danych.

- kolumnowa reprezentacja danych
- wspiera różne typy i poziomy kompresji
- wspiera losowy dostęp do danych, jednak nie dzieli ich na paczki, tak jak to robi np. parquet



Eksperyment

Dane testowe użyte do eksperymentu pochodzą z jednego z konkursów z platformy Knowledge Pit.

Charakterystyka danych:

- 4 kolumny: jedna tekstowa i trzy numeryczne
- 476446720 wierszy
- rozmiar paczki - 2^{16}
- 7270 paczek

Table 1
SAMPLE OF ANALYZED RAW DATA TABLE

src_ipcategory	srcport	dstport	relative_timestamp
PRIV-10	0	0	0
PRIV-10	0	0	0
PRIV-10	0	0	7
PRIV-10	60023	33613	0
PRIV-10	60023	33613	0
PRIV-10	60061	25	2
PRIV-10	52555	33613	11
PRIV-10	52555	33613	11
INTERNET	29247	25	0
INTERNET	0	0	1
PRIV-192	443	62449	240
PRIV-192	443	62454	1146

Eksperyment – pierwsze porównanie

Dla porównania wydajności trzech narzędzi, wykonano zapytanie, które w języku SQL może być zapisane:

```
SELECT SUM(col1) WHERE col2 <= 5;
```

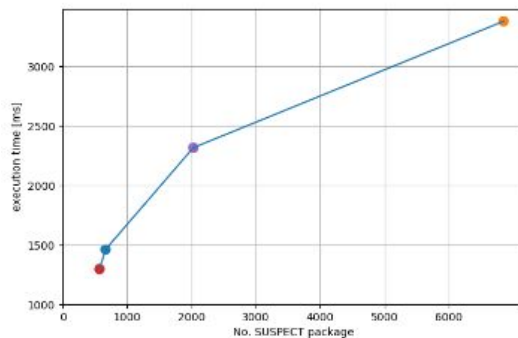
Dla Parquet oraz ROOT wykonano porównanie customowej implementacji zgranulowanej tabeli oraz standardowej formy danego frameworka. Wyniki te zestawiono z wynikami uzyskanymi dla systemu Infobright.

Table II
NUMBER OF PACKAGES OF THREE CATEGORIES BASED ON A QUERY
FOR 2^{16} PACKAGE SIZE

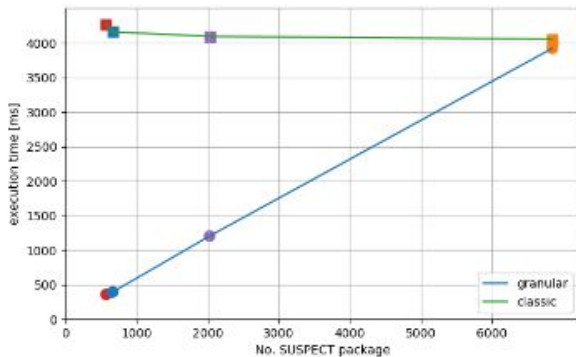
filtering query	FULL	NEGATIVE	SUSPECT
sreport \leq 30000	74	347	6849
sreport \leq 65534	5242	0	2028
dstport \leq 65534	6606	0	664
relative_timestamp \leq 86399	6700	0	570

Eksperyment – pierwsze porównanie

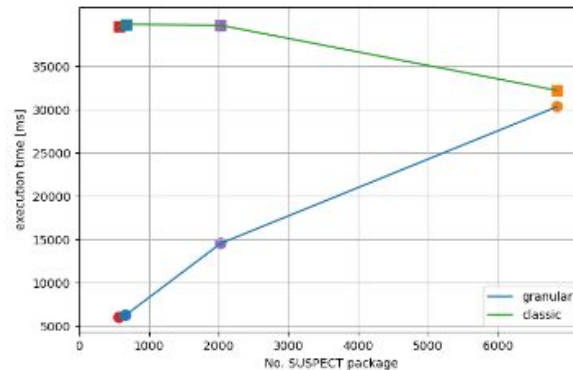
Infobright



Parquet

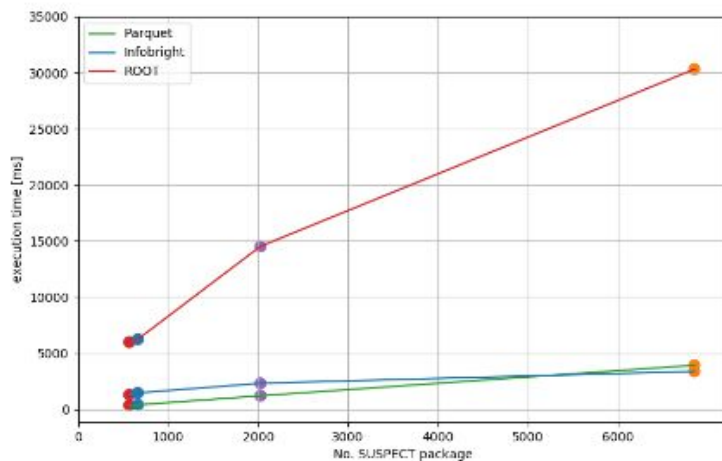


ROOT



Eksperyment – pierwsze porównanie

Porównanie wydajności Infobright, Parquet oraz ROOT dla paczki o rozmiarze 2^{16}



Eksperyment – rozszerzona analiza

Jak zostało to wspomniane wcześniej, rozmiar paczki wpływa znacząco na procesowanie zapytania. Rozszerzono więc analizę porównując dodatkowo dwa inne rozmiary - 2^{15} oraz 2^{17} .

W związku z tą zmianą uległ zmianie również rozkład paczek FULL/NEGATIVE/SUSPECT, który został zaprezentowany w tabeli.

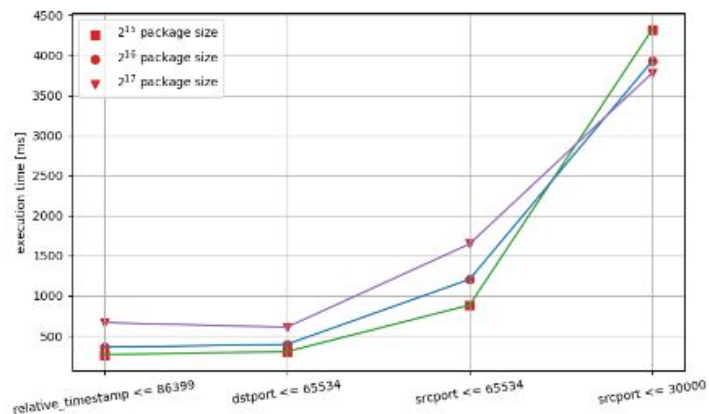
Table III
NUMBER OF PACKAGES OF THREE CATEGORIES BASED ON A QUERY
FOR DIFFERENT PACKAGE SIZE

<i>package size: 2¹⁵</i>			
filtering query	FULL	NEGATIVE	SUSPECT
srcport \leq 30000	182	942	13416
srcport \leq 65534	11877	0	2663
dstport \leq 65534	13762	0	778
relative_timestamp \leq 86399	12651	0	1889
<i>package size: 2¹⁷</i>			
srcport \leq 30000	27	131	3477
srcport \leq 65534	2229	0	1406
dstport \leq 65534	3111	0	524
relative_timestamp \leq 86399	2722	0	913

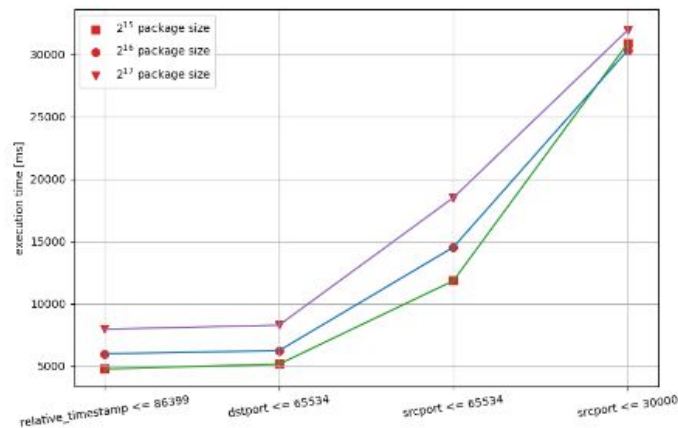
Eksperyment – rozszerzona analiza

Porównanie wydajności dla Parquet oraz ROOT dla trzech różnych rozmiarów paczek

Parquet



ROOT



Eksperyment – rozszerzona analiza

O ile zmiana rozmiaru paczki danych nie wpływa na strukturę pliku ROOT (brak wsparcia podziału danych na paczki), to ma ona wpływ na generowanie pliku parquet. W związku z tym zmienia się jego rozmiar dyskowy.

Rozmiar analizowanego pliku CSV wynosił 11GB. W tabeli zaprezentowano rozmiary plików wynikowych dla Infobright, Parquet oraz ROOT.

Table IV
INFOBRIGHT, ROOT AND PARQUET FILE SIZES

Infobright [MB]	ROOT [MB]	Parquet [MB]		
		2^{15}	2^{16}	2^{17}
878	1867	3105	2849	2533

Podsumowanie eksperymentów

Koncepcja zgranulowanej tabeli pochodząca z silnika Infobright została z powodzeniem zaimplementowana w Parquet oraz ROOT. Widać wyraźne, oczekiwane przyspieszenie dla analizowanych zapytań.

Wzrost wydajności redukuje czasy wykonywania zapytań o rzędy wielkości. Niestandardowa implementacja nie powoduje również żadnego znaczącego narzutu w przypadkach najgorszych - tam, gdzie podsumowania nie mogą zostać użyte (wszystkie paczki są oznaczone jako SUSPECT).

InfoFrame

InfoFrame jako biblioteka programistyczna udostępnia użytkownikowi API za pomocą którego będzie miał możliwość zapisywania, odczytywania, modyfikowania oraz przetwarzania danych zapisanych w formacie infoframe.

Podstawowe operacje analityczne (np. filtrowanie, grupowanie), a także te bardziej zaawansowane (narzędzia oraz funkcje pomocnicze wykorzystywane w definiowaniu algorytmów) będą wspierały koncepcje tabeli zgranulowanej.

Bibliografia

D. Ślęzak, P. Synak, J. Wróblewski, G. Toppin, "Infobright – Analytic Database Engine using Rough Sets & Granular Computing" 2010 IEEE International Conference on Granular Computing, August 2010.

P.W. White, C.D. French, "Database system with methodology for storing a database table by vertically partitioning all columns of the table", US Patent 5,794,229 (1998).

Graham Toppin, Janusz Borkowski, Dominik Ślęzak, Shengli Shi, Piotr Synak, Jakub Wróblewski, Todd Joseph Wongkee, George Charalabopoulos, "System and method for granular scalability in analytical data processing", US Patent Application Publication US 2015/0088807 (2015).

Mateusz Wnuk