CFSA Seminar, University of Warwick, 19 February 2019

Tutorial: Real time big data handling

ALEKSANDR KOLOTKOV

PENZA, RUSSIA

EMAIL: ALEXANDERKOLOTKOV@GMAIL.COM

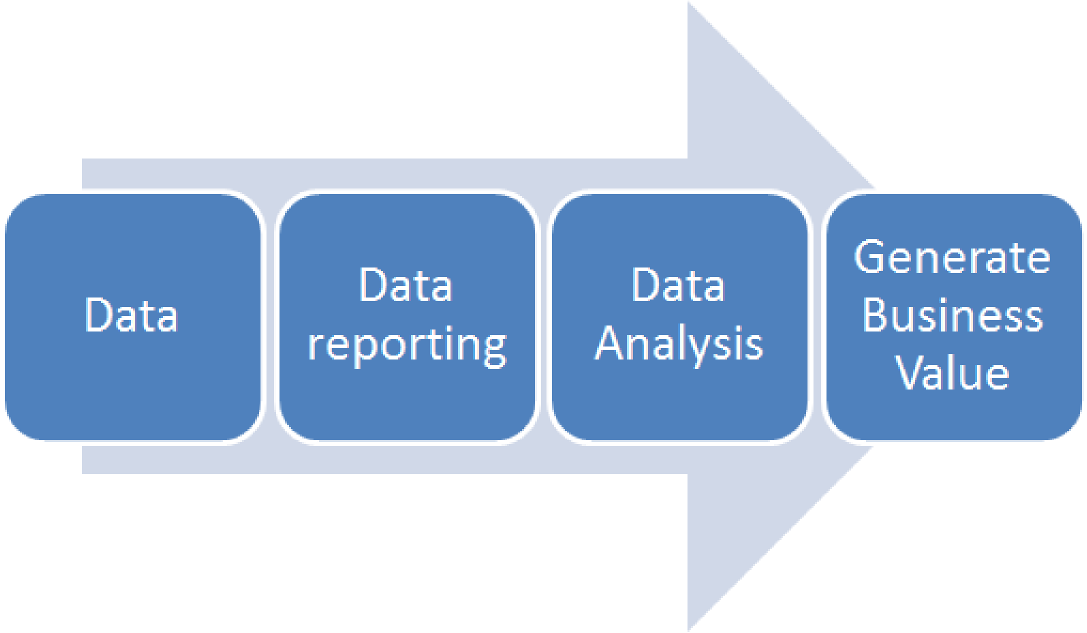
Process of data handling

Raw data collecting – choosing meaningful event parameters and storing them in a separate row for each event; at this stage we do not have to know exactly which dependencies and between which parameters will be further analyzed

Data reporting – the tool for aggregation and filtration collected raw data, as well as for searching for dependencies between parameters

Data analysis – the process of obtaining new knowledge, hypotheses testing and discovering dependencies between parameters using reporting

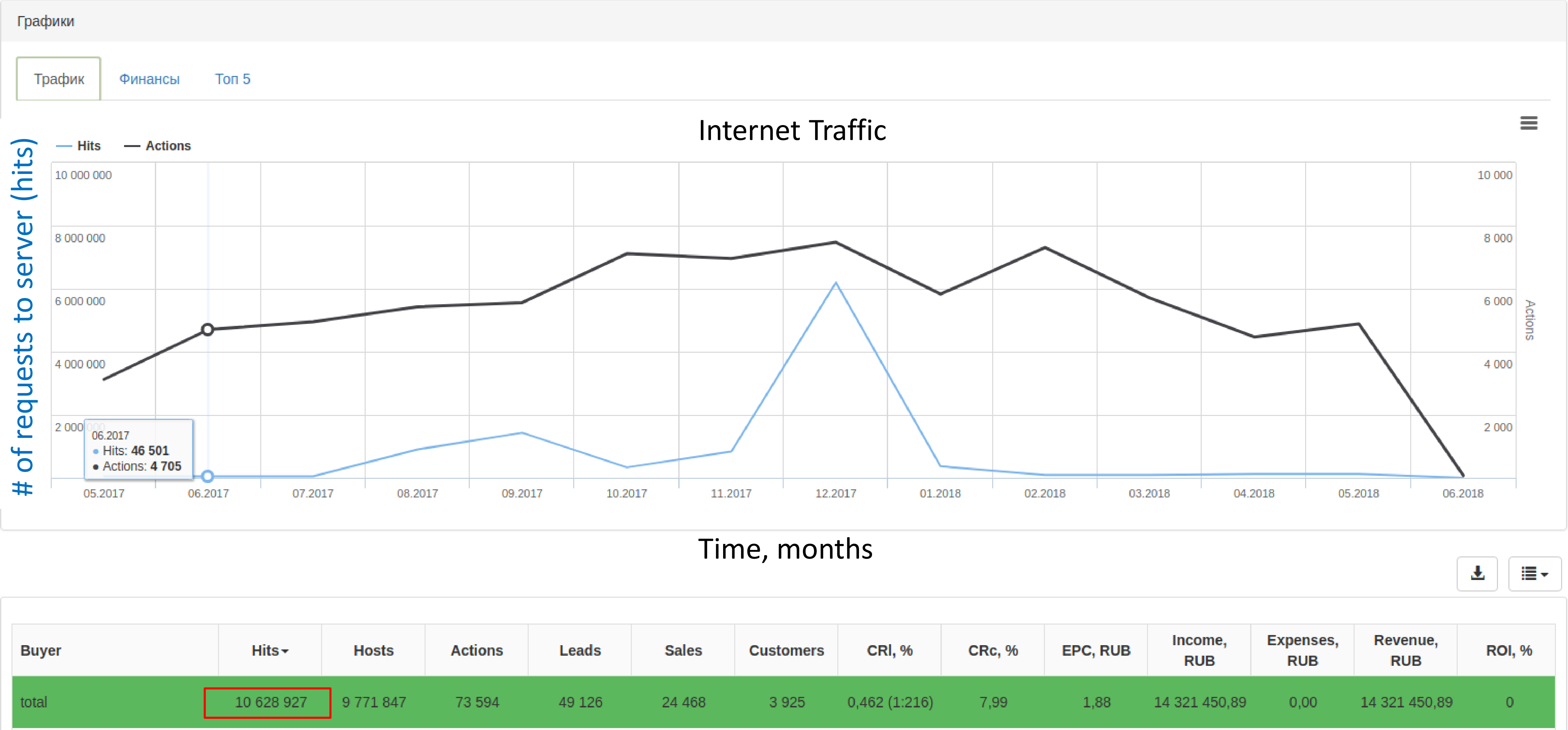
Dealing with new trends and insights we received after the data analysis



Raw data collecting

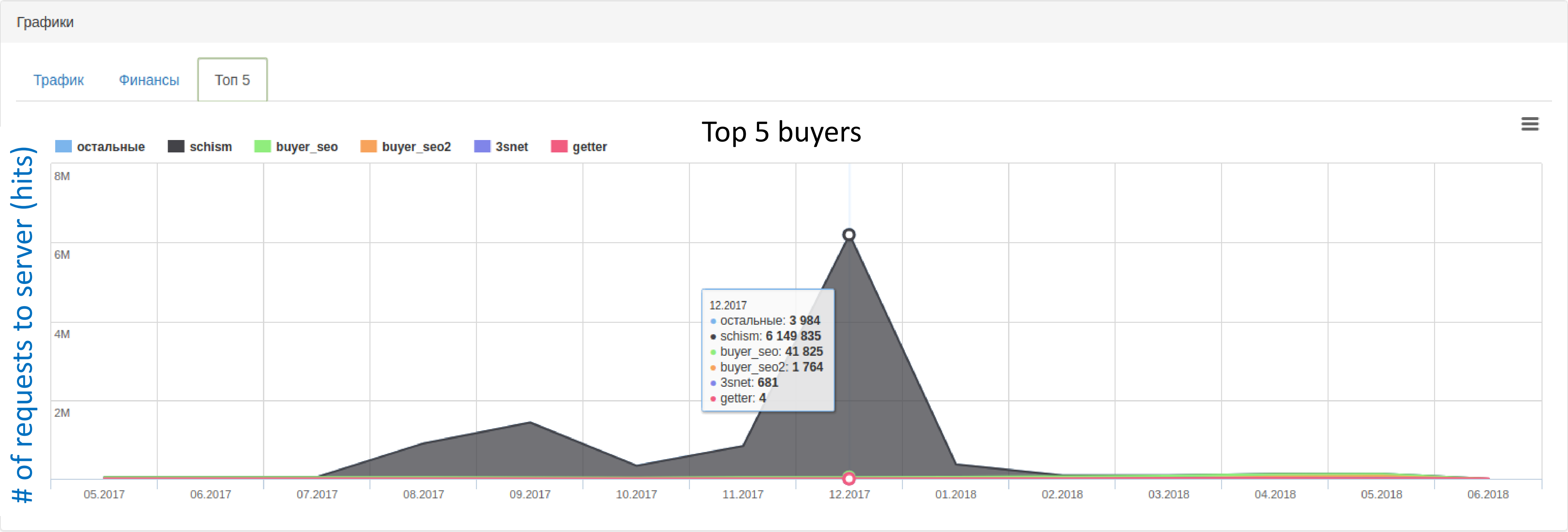
id	buyer	dt
1	buyer_seo	2017-07-01 00:00:01
2	buyer_seo2	2017-07-01 00:00:01
3	getter	2017-07-01 00:00:03
4	getter	2017-07-01 00:00:04
5	getter	2017-07-01 00:00:05
6	buyer_seo2	2017-07-01 00:01:16
7	buyer_seo2	2017-07-01 00:01:17
8	buyer_seo	2017-07-01 00:11:20
9	buyer_seo	2017-07-01 00:11:55
10	buyer_seo	2017-07-01 00:15:22
11	buyer_seo2	2017-07-01 00:10:01

Reporting: detecting unusual behavior in data



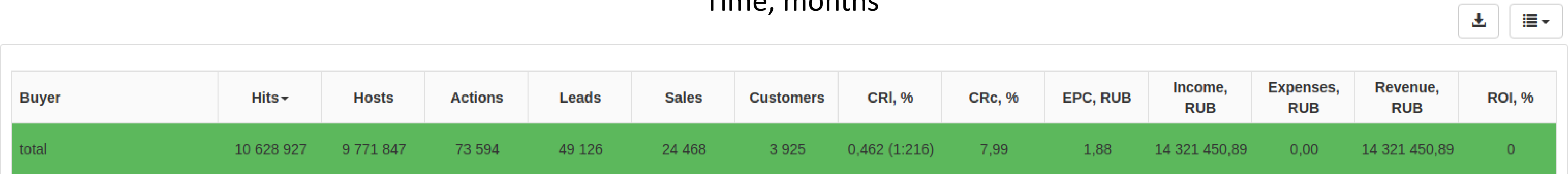
Buyer	Hits▾	Hosts	Actions	Leads	Sales	Customers	CRl, %	CRc, %	EPC, RUB	Income, RUB	Expenses, RUB	Revenue, RUB	ROI, %
total	10 628 927	9 771 847	73 594	49 126	24 468	3 925	0,462 (1:216)	7,99	1,88	14 321 450,89	0,00	14 321 450,89	0

Analytics: hypothesis testing



12.2017
- остальные: **3 984**
- schism: **6 149 835**
- buyer_seo: **41 825**
- buyer_seo2: **1 764**
- 3snet: **681**
- getter: **4**

Time, months

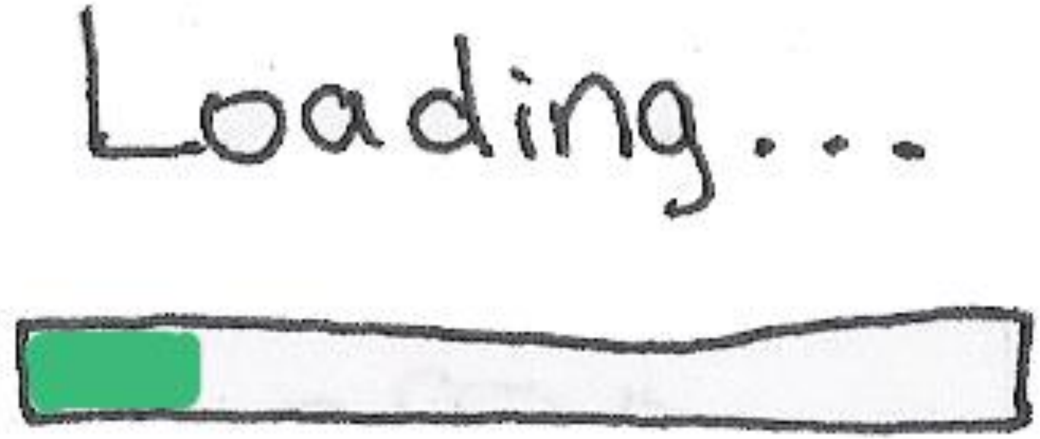


Buyer	Hits▾	Hosts	Actions	Leads	Sales	Customers	CRl, %	CRc, %	EPC, RUB	Income, RUB	Expenses, RUB	Revenue, RUB	ROI, %
total	10 628 927	9 771 847	73 594	49 126	24 468	3 925	0,462 (1:216)	7,99	1,88	14 321 450,89	0,00	14 321 450,89	0

- Reasonable computational resources

- Adequate time-scales needed for the report making

- And thus fast hypothesis testing



ClickHouse

…is an open source column-oriented database management system capable of real time generation of analytical data reports using SQL queries.

Why ClickHouse?

Simple and handy

Highly reliable

Blazing fast

Hardware efficient

Linearly scalable

Feature reach

Fault tolerant

https://clickhouse.yandex/

All one need to know to start with ClickHouse in a nutshell...

- Here's the command to check if your CPU is suitable (we will use Intel Core i7):

```
$ grep -q sse4_2 /proc/cpuinfo && echo "SSE 4.2 supported" || echo "SSE 4.2 not supported"
```

- ClickHouse can run on any Linux, FreeBSD or Mac OS X
  (we will use Docker container with Linux on host computer with Mac OS X)

- Docker - a computer program that performs operating-system-level virtualization.
  Docker installation documentation: https://www.docker.com/get-started
  ClickHouse Server Docker Image: https://hub.docker.com/r/yandex/clickhouse-server/

- Tabix - visual interface for ClickHouse allowing one to perform data querying in web browser.
  Documentation link: https://github.com/tabixio/tabix

- User-friendly SQL dialect for data querying: https://en.wikipedia.org/wiki/SQL

- A sample dataset. We will use the USA civil flights data since 1987 till 2015 from the open sources (contains 166 millions rows, 63 Gb of uncompressed data)
  Download link: https://yadi.sk/d/pOZxpa42sDdgm

Sample dataset file format

Year
Quarter
Month
Day of Month
Day of Week
Flight Date
Unique Carrier
Airline ID
Carrier
Tail Number
Flight Number
Origin Airport ID
Origin Airport Seq ID
Origin City Market ID
Origin
Origin City Name
Origin State
Origin State Fips
Origin State Name

```
1987,4,10,19,1,1987-10-19,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,20,2,1987-10-20,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,21,3,1987-10-21,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,22,4,1987-10-22,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,23,5,1987-10-23,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,24,6,1987-10-24,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,25,7,1987-10-25,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,26,1,1987-10-26,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,27,2,1987-10-27,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,28,3,1987-10-28,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,29,4,1987-10-29,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,30,5,1987-10-30,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,31,6,1987-10-31,"CO",19704,"CO","","597",10821,1082102,30852,"BWI","Baltimore, MD","MD","24","Maryland", ...
1987,4,10,1,4,1987-10-01,"CO",19704,"CO","","598",12266,1226601,31453,"IAH","Houston, TX","TX","48","Texas", ...
1987,4,10,2,5,1987-10-02,"CO",19704,"CO","","598",12266,1226601,31453,"IAH","Houston, TX","TX","48","Texas", ...
```

Create table for sample dataset in ClickHouse

```
:) CREATE TABLE ontime
(
    Year UInt16,
    Quarter UInt8,
    Month UInt8,
    DayofMonth UInt8,
    DayOfWeek UInt8,
    FlightDate Date,
    UniqueCarrier FixedString(7),
    AirlineID Int32,
    Carrier FixedString(2),
    TailNum String,
    FlightNum String,
    OriginAirportID Int32,
    OriginAirportSeqID Int32,
    OriginCityMarketID Int32,
    Origin FixedString(5),
    OriginCityName String,
    OriginState FixedString(2),
    OriginStateFips String,
    OriginStateName String,
    ...
)
ENGINE = MergeTree(FlightDate, (Year, FlightDate), 8192);
```

Questions we will try to answer (obtain new knowledge):

- the most popular destinations in 2015;
- the most popular cities of departure;
- cities of departure which offer maximum variety of destinations;
- flight delay dependence on the day of week;
- cities of departure with most frequent delays for 1 hour or longer;
- flights of maximum duration;
- distribution of arrival time delays split by aircompanies;
- aircompanies who stopped flights operation;
- most trending destination cities in 2015;
- destination cities with maximum popularity-season dependency.

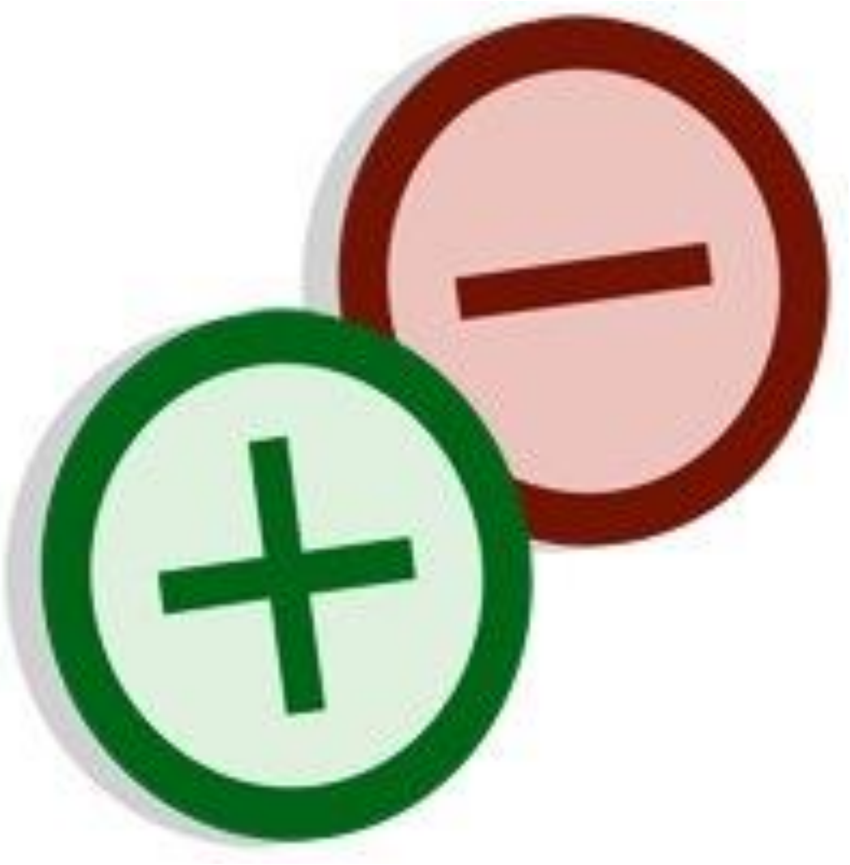
Pros and cons of using ClickHouse on your own computer

Pros

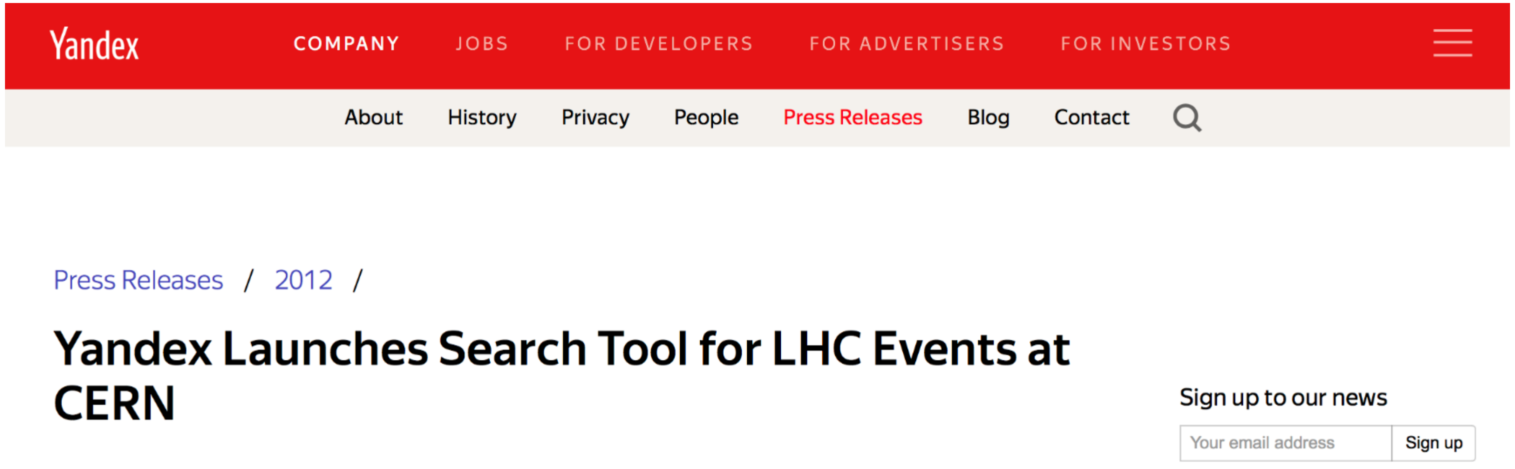
- It is absolutely possible
- It allows not only store big data sets on your hard drive in a compact form but it provides you real-time access to any part of that data
- It is really fast in data aggregation and filtration
- You should not be database administrator or any other kind of technical specialist to start using it
- It has excellent and detailed documentation

Cons

- It requires some preparations of data before loading to database
- To be able use it in a more efficient way you still have to learn a little something new

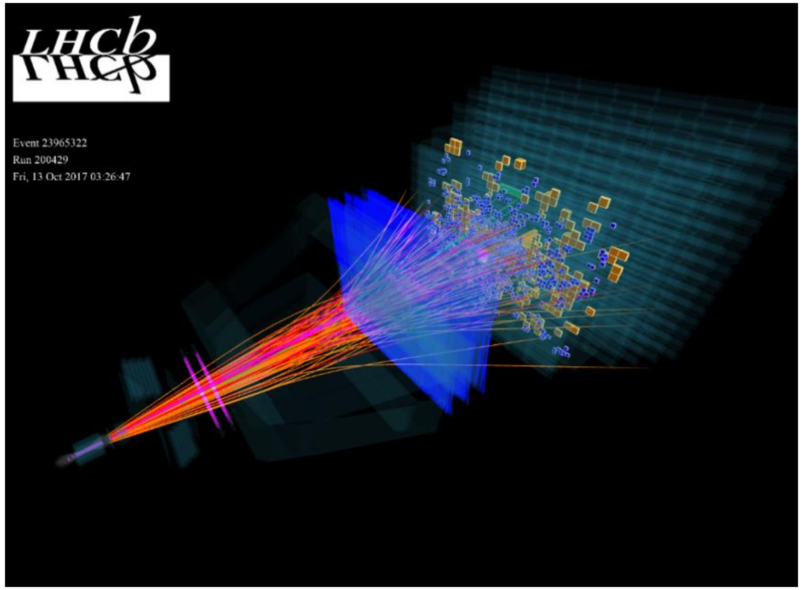


ClickHouse has already been successfully implemented at CERN's LHCb experiment to store and process metadata on 10 billion events with over 1000 attributes per event.



Press Releases / 2012 /

Yandex Launches Search Tool for LHC Events at CERN



“It’s a pleasure to work with the European Organization for Nuclear Research, as we welcome any opportunity to apply our technologies across different fields. Also, it’s nice to do something useful for physics and basic science. We will keep refining our LHCb event search, which may take us to the stage where we could contribute to other experiments at CERN,” says Ilya Segalovich, Yandex’s CTO.

https://www.yandex.com/company/press_center/press_releases/2012/2012-04-10/

From business intelligence to solar and space climate data?

https://github.com/kolotkov/clickhouse-tutorial