

Вольныя СКБД для захоўвання часавых шэрагаў.

Дубіцкі А.У., Касцюк Д.А., Маркіна А.А.

Брэсцкі дзяржаўны тэхнічны ўніверсітэт, dmitriykostiuk@gmail.com

A review of Time Series Databases is presented, with focus on free/libre and open source software. Time-arranged list of TSDB with notes on specifics of their usage is presented as far as the generalized functionality overview and the analogy in relational database approach.

Базы даных часовых шэрагаў (time series database, TSDB) - гэта спецыялізаваныя СКБД для захоўвання праіндэксаваных па часе масіваў даных. Прычыны іх з'яўлення - неабходнасць збору, захоўвання і апрацоўкі вялікіх масіваў разнастайных метрык (па відавочным прычынах, асабліва часта можна сустрэць TSDB ў сістэмах маніторынгу), а таксама той факт, што выкарыстанне рэляцыйных БД у сістэмах са складанай логікай і высокім аб'ёмам транзакцый для даных часавых шэрагаў аказваецца не самым практычным выбарам.

У параўнанні з рэляцыйнымі СКБД, TSDB з'явіліся адносна нядаўна, і ў сапраўдны момант знаходзяцца на стадыі ўстойлівага росту папулярнасці. Акрамя задач маніторынгу, даныя такога роду збіраюцца і захоўваюцца пры самых розных задачах. Прывядзем некалькі прыкладаў з іншых абласцей:

- **статыстыка выкарыстання** (наведванне сайтаў, задзейнічаны апаратных і сеткавых рэсурсаў і г.д.);

- фінансавая, біржавая, актуарная статыстыка;
- макраэканамічная статыстыка.

Рэляцыйныя БД не дазваляюць эфектыўна апрацоўваць спарадкаванае мноства элементаў часовага шэрагу, а іншыя тыпы класічных СКБД таксама апынуліся неэфектыўнымі пры вялікай нагрузцы. У выніку шырокай сферы патэнцыйнага прымянення, на рынку прысутнічаў дастаткова высокі попыт на эфектыўнае захоўванне часовых шэрагаў з магчымасцямі іх апрацоўкі, характэрнымі для больш традыцыйных рэляцыйных БД, уключаючы транзакцыі, матэматычныя і лагічныя аперацыі над выбаркамі.

Значнае месца сярод TSDB займаюць свабодныя праекты (табл. 1 змяшчае тыя з іх, з якімі ў большай ці меншай ступені ўдалося пазнаёміцца аўтарам).

У дачыненні да прадстаўлення даных ў TSDB можна вылучыць дзве мадэлі: мадэль «wide-table», якая спрашчае ўзаемадзеянне з данымі па-за часовых шэрагаў і робіць працу больш звыклай для распрацоўшчыкаў рэляцыйных БД, і «narrow-table», якая не патрабуе змены структуры табліцы для збору новых метрык, аднак з'яўляецца больш складанай для ўспрымання чалавекам (малюнак 1).

У плане рэалізацыі TSDB можа быць прымяненне спецыяльнай схемы для захоўвання часовых шэрагаў у больш універсальнай традыцыйнай СКБД (1), альбо спецыялізаваная СКБД, спраектаваная з нуля (2).

«wide-table»:

Person	Age	Weight
Bob	32	128
Alice	24	86
Steve	64	95

«narrow-table»:

Person	Variable	Value
Bob	Age	32
Bob	Weight	128
Alice	Age	24
Alice	Weight	86
Steve	Age	64
Steve	Weight	95

Малюнак 1 — Мадэлі прадстаўлення даных ў TSDB

Табліца 1 - Вольныя TSDB

TSDB (ліцэнзія, год)	Мова запытаў	Платформа
InfluxDB (MIT, 2013)	InfluxQL, JSON	Go
Druid (Apache 2.0, 2012)	SQL, REST	Java
TimescaleDB (Apache 2.0, 2017)	SQL	C

Prometheus (Apache 2.0, 2015)	PromQL, REST	Go
KairosDB (Apache 2.0, 2013)	REST	Java
Hawkular Metrics (Apache 2.0, 2014)	REST	Java
Riak TS (Apache 2.0, 2015)	SQL	Erlang
Blueflood (Apache 2.0, 2013)	REST	Java
DalmatinerDB (MIT)	DQL	Erlang

Характерны приклад TSDB першаго типу - TimescaleDB, яка ґрунтуецца на PostgreSQL, з аптымізацыямі для хуткага пошуку і складаных запытаў. Па відавочных прычынах TimescaleDB падтрымлівае SQL і ўсе функцыі, тыповыя для рэляцыйнай СКБД: JOIN, другасныя індэксы, складаныя прэдыкаты WHERE, GROUP BY, аґрэгацыі, пад-запыты, статыстычныя і апраксімацыйныя функцыі і інш. Таксама TimescaleDB выкарыстоўвае мадэль «wide-table», характэрную хутчэй для рэляцыйных СКБД, чым для TSDB.

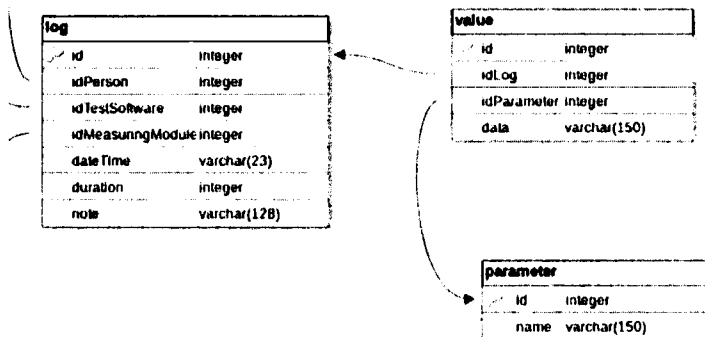
Добрым прыкладам TSDB другога тыпу з'яўляецца InfluxDB, першы рэліз якой адбыўся ў 2013 годзе. InfluxDB спрашчае аналіз даных за кошт уласнай мовы запытаў InfluxQL, але падтрымлівае і запыты ў фармаце JSON. Большасць кліенцкіх бібліятэк выкарыстоўвае HTTP API для ўзаемадзеяння з серверам. Прыклад каманды, якая дадае запіс, выглядае наступным чынам:

```
curl -X POST -d '{"name": "<table name>", "columns": [{"<column1>", "<column2>", ..., "<columnN>"}], "points": [{"<value1_column1>", [<value1_column2>], ..., [<value1_column1>, <value2_columnN>]]}]'
'http://<host>:<port>/db/<db_name>/series?u=<login>&p=<pass>'
```

дзе <table name> - імя табліцы, <columnX> - імёны слупкоў, <valueY_columnX> - значэнне ячэйкі, <host> - імя хаста, <port> - нумар порта сервера, <db_name> - імя базы даных, <login> - лагін, а <pass> - пароль карыстальніка базы даных (замест ўводу лагіна і пароля ў URL можна выкарыстоўваць HTTP-аўтэнтыфікацыю).

Яшчэ адным характэрным прыкладам папулярнай TSDB другога тыпу з'яўляецца DalmatinerDB. Механізм захоўвання даных гэтай TSDB ґрунтуецца на ўласцівасцях магутнай і прадукцыйнай файлавай сістэмы ZFS (Zettabyte File System), што забяспечвае вельмі высокую эфектыўнасць захоўвання (і вельмі нізкую пераноснасць). Гэтая асаблівасць DalmatinerDB, а таксама яе алгарытмічныя рашэнні забяспечваюць маштабаванасць і надзвычай высокую ўстойлівасць. Выкарыстоўваецца мова запытаў, вельмі падобная на SQL, і гэта заўважна зніжае патрабаванні да навучання распрацоўшчыкаў, чаго нельга сказаць аб сістэмных патрабаваннях СКБД (рэкамендаваная канфігурацыя ўключае 16 працэсарных ядраў, 60 Гб аператыўнай памяці і дыскавую падсістэму на базе SDD).

У заключэнне заўважым, што з-за фактараў, якія ўскладняюць прымяненне TSDB (прывязка да канкрэтнай СКБД, мовы праграмавання, аперацыйнай сістэмы, альбо сістэмных патрабаванняў) распрацоўшчык можа апынуцца ў сітуацыі, калі ўласная рэалізацыя механізму захоўвання часовых шэрагаў у рэляцыйнай СКБД пераважней, чым выкарыстанне спецыялізаванага рашэння - нягледзячы на тое, што апошнія прадастаўляюць лепшыя функцыянальныя магчымасці і аптымізацыю. У гэтым выпадку ў рэалізацыі звычайна можна прасачыць наяўнасць у тым ці іншым выглядзе цэнтралізаванай "табліцы фактаў", якая захоўвае даныя аб канкрэтных падзеях (значэння часовых шэрагаў), з унікальнымі складовымі ключамі, якія аб'ядноўваюць першасныя ключы "табліц вымярэнняў", што ўтрымліваюць атрыбуты падзей, захаваных ў табліцы фактаў. Пры злучэнні табліц выкарыстоўваецца альбо схема зоркі (з памяншэннем ліку табліц і паскарэннем запытаў за кошт дэнармалізацыі табліц вымярэнняў), альбо схема сняжынкi (з вялікім лікам табліц і больш поўнай нармалізацыяй).



Малюнак 2 — Захоўванне часовага шэрагу ў рэляцыйнай СКБД (фрагмент)

У якасці прыкладу можна разгледзець злучэнне табліц на малюнку 2, якое аўтары ўжылі ў праекце UXDump (<https://bitbucket.org/AsyaAliset/uxdump>) для захоўвання вынікаў біямэтрычных і іншых вымярэнняў, выкананых падчас тэставання эфектыўнасці чалавека-машыннага ўзаемадзеяння. У дадзеным праекце выкарыстоўваецца СКБД MySQL, для якой на момант распрацоўкі адсутнічала адкрытая TSDB-надбудова (і, наколькі аўтарам вядома, сітуацыя не змянілася).

Як можна бачыць, вынікі вымярэнняў паралельна з некалькіх крыніц заносзяцца ў пару табліц parameter і values (набор вымяраных параметраў загодзя невядомы, што адпавядае мадэлі "narrow-table"). Табліца values, дзе захоўваюцца атрыманыя значэнні, мае сувязі з даведнікам назваў вымяраных параметраў parameter і з табліцай log, у якой захоўваецца інфармацыя аб серыях вымярэнняў.

Спис літератури

1. Knowledge Base of Relational and NoSQL Database Management Systems // <https://db-engines.com/en/>
2. Top 10 Time Series Databases // <https://blog.outlyer.com/top10-open-source-time-series-databases>
3. Introduction to Time Series Database // <https://www.linkedin.com/pulse/introduction-time-series-database-pinglei-guo/>