

АНОТАЦІЯ

Об'єм пояснювальної записки 73 сторінки. Робота містить 15 рисунків, 4 таблиці та 5 додатків.

Метою даної дипломної роботи було покращення способу збереження даних у вигляді часових рядів, що будуть пізніше піддаватися обробці, аналізу та візуалізації.

Для цього було створено спеціальне програмне забезпечення для збереження даних у вигляді часових рядів (Time Series Data Storage), що може бути використано найефективнішим чином для збереження великих об'ємів даних вимірювань. Було розглянуто різного роду готові рішення, проведений аналіз на відповідність ключовим критеріям та зроблено висновки щодо найкращих варіантів реалізації, що будуть корисні для нас під час прийняття рішень в своїй розробці. Було спроектовано ядро сховища даних метрик, розроблено зручний і простий синтаксис БД та систему парсингу (синтаксичний аналізатор) запитів до БД допомогою за API у вигляді інтерфейсу командного рядка. Також реалізовано такі важливі функціональні рішення як: розмежування доступу за ролями користувачів, масштабованість (реплікація і шардинг), компресія даних, асинхронне виконання, журнал транзакцій та можливість відновлення даних

Ключові слова: часові ряди, статистичний аналіз, сховище даних, СУБД, NoSQL, транзакція, реплікація, шардинг, компресія даних, WAL, програмний рушій, API, синтаксис БД, парсер.

SUMMARY

Total capacity of explanatory note is 73 Pages. It contains 15 illustrations, 4 tables, 5 applications.

The goal of the diploma thesis is the improvement of the way of storing time series for the later analysis, processing and visualization.

For this purpose, special software was created for storing time series (Time Series Data Storage), which can be used to store a big volumes of measurements data in the most efficient way. A variety of ready-made solutions were considered, an analysis was carried out on key criteria, and conclusions were drawn about the best implementation options that would be useful for us when making their decisions in their development. The source engine of the metric data storage was designed, a handy database syntax and a parser for querying database using the API with command line interface was developed. Also, such necessary functionalities were implemented: role based access control, scalability (replication and sharding), data compression, asynchronous execution, transaction logger and backup

Keywords: time series, statistical analysis, DBMS, NoSQL, transaction, replication, sharding, data compression, WAL, software engine, API, DB syntax, parser.