Towards the Internet of Water: Using Graph Databases for Hydrological Analysis

Erik Bollen[1,2] , Rik Hendrix[2], Bart Kuijpers[1], Alejandro Vaisman[3]

1) Hasselt University (UHasselt), Computer Science, Data Science Institute, Hasselt Belgium.
2) Flemish Institute for Technological Research (VITO), Data Science Hub, Mol Belgium.
3) Instituto Tecnológico de Buenos Aires, Department of Information Engineering, Buenos Aires Argentina.

Monitoring, analysing and forecasting water-systems, such as rivers, lakes and seas, is an essential part of the tasks for an environmental agency or government. In the region of Flanders, in Belgium, different organisations have united to create the "Internet of Water" (IoW). During this project, 2500 wireless water-quality sensors are being deployed in rivers, canals and lakes all over Flanders. This network of sensors will support a more accurate management of water systems by feeding real-time data. Applications include monitoring real-time water-flows, automated warnings and notifications to appropriate organisations, tracing pollution and the prediction of salinisation.

Despite the diversity of these applications, they mostly rely on a correct spatial representation and fast querying of the flow path: where does water flow to, where can it come from, and when does the water pass at certain locations? In the specific case of Flanders, the human-influenced landscape provides additional complexity with rivers, channels, barriers and even cycles. The river system can be seen as a graph. Therefore, rivers can be modelled, stored and queried using graph databases. Although this normally could be realised using relational databases, an "impedance mismatch" appears between the conceptual representation and the storage models.

In this work, the differences in storing and querying flanders' rivers dataset, "Vlaamse Hydrografische Atlas" - VHA, is studied for graph databases and relational databases. The first group is represented by Neo4j with the query language Cypher, and the latter is represented by PostgreSQL and the language SQL. More specifically, the work will take on the data preparation needed including the creation of appropriate data representations such as binary 'flows-to' relationship or network topologies. Special situations and problems that can be encountered will be discussed and possible solutions will be presented if needed. This work will mainly focus on expressing possible relevant queries in Cypher as well as in SQL and comparing the major differences and similarities in both languages. This is followed by a retrieval speed analysis for all queries in their respective systems.

The work and its preliminary results show that queries are expressed more naturally by a graph-based representation and its dedicated graph query language Cypher in contrast to the relational alternative. The outcome of the speed comparison also shows that the queries are not only more easily expressed in Cypher, but also that the retrieval of the results is more efficient in Neo4j than in PostgreSQL in almost all cases tested. All of this leads to the conclusion that the mismatch between the relational data model and the data has a negative impact on querying, but it also shows that with graph databases this impact can be circumvented. This in turn can help the development of applications in the IoW project and similar topics.