Design of energy-loaded systems using the Neo4j graph database

T V Zhidchenko, M N Seredina, N M Udintsova and N A Kopteva

Azovo-Black Sea Engineering Institute of the Federal State Budgetary Educational Institution of Higher Education Don State Agrarian University, 21 Lenina, Zernograd, 347740, Russia

E-mail: tvzh@inbox.ru

Abstract. The article presents a study of the tasks that determine the features of power-loaded networks, and tasks that are solved using graph models for power-loaded networks. The tasks that determine the features of power-loaded networks are communication check, isolation of network components and components, combinatorial tasks on graphs, calculation of flow distribution, graph simplification, optimal impact on the power grid, and determination of power proportions. This article presents the high-level architecture of Neo4j. Neo4j is an open-source graph database management system implemented in Java and is considered the most comprehensive and promising graph database management system. In comparison with modelling a graph database using a relational database management system, this approach allows applying additional optimization in the case of data with a more complex and loaded structure. The article proposes a model of an energy-loaded network, because it is a system of interaction of a large number of loads or consumers, active sources, communication lines. The declarative query language for graph databases Cypher was used for implementation. A graph database of an electrical network with a description of analytical data processing using a mathematical algorithm and software specifics of the work of queries in a multi-relational database environment has been built. When a query is executed for a user-specified network segment, a connectivity component is allocated; then for the selected elements of the network graph, correspondence is made with the elements of the network itself.

1. Introduction

Electric networks are one of the options solved using graph theory. Specific features of power-loaded networks are their currently significant spatial extent, the nodal concentration of the main characteristics. These features allow applying graph model theory for simulation and analysis. Subordination to sequential principles makes it possible to introduce individual network elements into subsystems and identify the topological connectivity of network objects. From sources to consumers, the transfer of the optimization function is carried out by power-loaded networks. Participating in the transfer of the optimization function, the electrical circuits of the network elements form the structure of the network.

Therefore, the transformation and analysis of electrical networks is carried out for:

1. Checking the availability of communication - determination of objects or network elements that are connected to the data and form a continuous circuit for the transmission of electrical energy, taking into account the operability of the object that allows or prohibits transmission (circuit breakers, fuses, disconnectors), as well as restrictions on the direction of movement and connectivity certain network elements. In electrical networks, these will be the parameters of devices, accounting for the voltage...
classes of the mating transmission lines.

2. Allocation of constituent links is a search for network elements connected with another element by a continuous chain of function transfer. Additional conditions for including elements in consideration are a distinctive feature of this task. In electrical circuits, it may be necessary to analyze only networks of the same voltage class or the number of phases.

3. Determination of special network components is an in-depth analysis of the topology of network sections is carried out together with the definition of their types and attributes. For electrical networks, the selection of a set of elements (components) of the network, connected and having one element of a specific type, through which power is supplied (feeder).

4. Classical combinatorial tasks on graphs, such as finding the maximum flow or the shortest (for a specific condition) route passing through a certain number of points, finding the number of nearest nodes.

5. Calculation of steady-state flow distribution is the task that mainly uses the consideration of the network as a graph. All of the above tasks can be used as subtasks when solving this type of tasks.

6. Simplification of the network graph - consists in replacing it with one element, which includes the properties of a group of elements of the same type, which are connected by the structure of the electric network graph.

7. To obtain the specified parameters, search for the best impact on the network. These tasks are based on performing a steady-state flow distribution calculation. In addition, the composition of the unified parameters plays an important role - if the unified part of the network parameters is small, then other parts can also be simplified.

8. Ratio of nutritional shares - the relevance of this calculation is reflected in determining the nutritional shares of elements from various sources. Usually it is necessary to determine the amount of energy consumed from each source for a subscriber connected to several power sources.

2. Purpose, objects and research methods.

The system by which electricity is transmitted from sources to consumers via long communication lines is an energy-loaded network. The article proposes a model of an energy-loaded network, based on the fact that it is a system of interaction of a large number of elements of load consumers, active sources, communication lines. It is characterized by the following values: power consumption, power flow, voltage loss and also the chosen direction. The electrical section of the network directed from the network node to the point with the consumer is considered as a consumer in the network. Communication lines represent the performed action over an extended section of the network. This also includes switching elements and other elements involved in the transmission of electricity electricity (Figure 1).

The structure of an engineering electrical network can be represented by a graph that reflects the nature of the connection between network elements. The total consumption of electricity consumers is equal to the target flow supplied to the grid by sources. The laws of the theoretical foundations of electrical engineering are applicable to such a network, when in any node of the network the sum of costs is zero, and the sum of voltage losses along any closed loop is zero. Thus, the model of an electrical network is a graph (graph model), the nodes of which are active and passive elements of the network, and the edges represent the actions performed.

The role and conditions for including network elements in the graph model are determined from the schematic construction when generating a graph model. Actions will be the edges of the graph, transmission lines, equipment involved in the transmission of electrical energy (transformers, switching devices) - nodes. Points of connection of network elements, “open” switches, special elements (collectors, sections) become the vertices of the graph. In addition, the nature of the connection between the inputs and outputs of these elements is reflected in the fact that some elements of the network turn into sets of vertices or edges. The model generated in this way is ready for solving the above tasks on its basis.
Figure 1. Electrical section of the network, where T1, T2, T3, T4 are transformer substations (TS) 35/10 kV and 10 / 0.4 kV, QW1, QW2, QW3, QW4 are load break switches, QS1, QS2, QS3, QS4, QS5, QS6, QS7 – disconnectors, FU1, FU2, FU3 – fuses, QF1, QF2, QF3 – circuit breakers, OPL – overhead power lines

For generation of a graph model database schema, we use the Cypher programming language. It is a declarative graph database query language Neo4j.

Neo4j is an “open source” multi-relational database [2] and uses a directed graph model [4] - it works with nodes, relationships, and properties. Directional, introduced relations connect marked points (for information sets). Both node points and relations between them contain arbitrary properties (key-value pairs). Elements of the electrical network in Neo4j can be implemented with the following data types: boolean variables (Bool), bytes (bytes), signed short integers (short), signed integers (Int), signed large numbers (long), real numbers floating point (float), double-precision floating-point (double), character types, and text strings.

The Neo4j database is written in the Java programming language so that Neo4j can be embedded in a Java application. Otherwise, the application development interface of the architectural style of interaction between the components of a distributed application in the network can be used. Neo4j comes with its own declarative Cypher query language. Cypher queries are entered into an input field that supports syntax highlighting and query history viewing.

Figure 2 shows the high-level architecture of Neo4j. At the top of the Java core, there are several other regulated ways of communicating one computer program to another to store and manage graph data together, including the architectural style of interaction between components of a distributed application on a network and other applications to the Java language. Through an object cache, which contains materialized chunks of a dataset, and a file system cache, which stores parts of the file system in order to speed up I / O operations, data is saved to and read from the hard disk. The data is physically stored in a specific number of separate record files. Figure 2 presents four of which (Nodes, Relationships, Relationship Types, Properties).
Figure 2. High-level architecture of Neo4j

We created the nodes and relationships using the Cypher programming language command syntax that is "adapted to the graph model" [5] CREATE or CREATE UNIQUE.

In addition to merely describing the shape of a node in a template, we can also describe a label as a parameter of a simple value. Moreover, we can also assign names to links, and describe certain data in such a way that links can get any set of types, and then can be listed in the template. Properties can be expressed in templates using a construct map: curly braces that contain a specific number of key-value pairs, separated by commas. Thus, we set the properties: element voltage, wire cross-section, quantity.

The program code is presented below:

```
CREATE (substation1: Substation {voltage: '35/10kV'}),
(load_switch1: Load_switch {voltage: '10kV'}),
(substation1)-[: CONNECT]->(load_switch1),
(disconnector1: Disconnector {voltage: '10kV'}),
(load_switch1)-[: WITH]->(disconnector1),
(disconnector2: Disconnector {voltage: '10kV'}),
(disconnector1)-[: TO]->(disconnector2),
(load_switch2: Load_switch {voltage: '10kV'}),
(disconnector2)-[: TO]->(load_switch2),
(air_line1: Air_line {voltage: '10kV'}, {section: '25mm'}),
(load_switch2)-[: SUPPLY]->(air_line1),
(disconnector3: Disconnector {voltage: '10kV'}),
(air_line1)-[: CONNECT]->(disconnector3),
(fuse1: Fuse {voltage: '10kV'}),
(disconnector3)-[: TO]->(fuse1),
(substation2: Substation {voltage: '10/0,4kV'}),
(fuse1)-[: CONNECT]->(substation2),
```
Results

Figure 3 represents the resulting "graph" [3] database of the electrical section of the network. The tops of the graph are transformer substations, load break switches, disconnectors, fuses, circuit breakers, and power lines.

The main task solved using this model is the search for electrically connected network sections with a particular condition.

This task is important for control operators, network designers, when determining the causes of congestion, analyzing the characteristics of the area receiving power. This task is solved using the MATCH operator. For a user-specified network segment, a connectivity component is allocated; then, for the selected elements of the network graph, correspondence is made with the elements of the network itself, which forms it. Selective choice of network elements that form the graph is an essential condition of the problem. As a rule, the user task is expressed as follows: "For a given voltage class or other system parameter, determine all the elements connected to this element." This task is most important for identifying priority consumers where, often a condition is put on the type of electrical network element or consumers.

This defines the conditions for including elements in the graph and in the query, for example::

MATCH (air_line:Air_line)
WHERE air_line.section=’25mm’
RETURN air_line
Figure 3. Graph database of the electrical section of the network implemented in Neo4j

Figure 4. Executing a query in Neo4j
The result of solving the request (Figure 4) is a list of network elements, for example, overhead power lines, electrically connected to the initial element, a 35/10 kV transformer substation.

Analytical processing of such a graph model can be carried out using the depth-first search algorithm or Dijkstra's algorithm of mathematical graph theory. The depth-first search algorithm is described by enumerating all the edges outgoing from the considered vertex. Dijkstra's algorithm (Figure 5) determines the optimal path from a given vertex to all the others.

Figure 5. Dijkstra's algorithm

However, using Neo4j for design allows increasing system performance and speed up queries.

4. Conclusions
Neo4j is one of the most promising databases that allows designing and process energy-loaded
systems as graph models of considerable length and with a large number of nodal points and characteristics. At the same time, the database makes it possible to "easily increase the functions of the system" [1] and is modified when the parameters or requirements for the system change.

References
[1] Zhidchenko T V, Seredina M N and Shipik L Y 2017 Achievements of modern science and education Theory of development and formation of databases 5(2) 64-69
[2] Desyatov S V and Voytyuk T E 2016 Almanac of scientific works of young scientists of ITMO University "Review of graph databases ORIENTDB and NEO4J" 2 31-33
[3] Robinson I, Weber D, and Eifrem E 2016 Graph databases: new opportunities for working with linked data (Moscow: DMK Press)
[4] Abramskiy M M and Timerkhanov T I 2018 Novosibirsk State University Bulletin. Series: Information technology Comparative analysis of the use of relational and graph databases in the development of digital educational systems 16 (4) 5-12
[5] Prytkov V A, Piletsky I I and Volorova N A 2018 Big Data and Advanced Analytics Analysis of the university repository using a graph database 4 179-185