Enhancing SAMOS Data Access in DOMS via a Neo4j Property Graph Database

Shawn R. Smith, Adam Stallard, Jocelyn Ellya
Center for Ocean-Atmospheric Prediction Studies, Florida State University, Tallahassee, FL, USA
srsmitli@fsu.edu; aps10d@my.fsu.edu; jelya@coaps.fsu.edu

IN21C-1745

Graph Aware is a Neo4j consultancy, training, and software development agency that provides off-the-shelf plugins, including one for generating a Time-tree directly into the graph via Java stored procedures.

This model is event-oriented. An update is made by providing a node, value of time, and resolution of time.

The SAMOS graph uses a single tree structure, which means that all ships are clustered together.

A large requirement of DOMS is managed by Neo4j, provided by Graph Aware technologies.

Graph databases are typically used to capture and store complex, connected data, such as social networks, where entities are related by various types of relationships. Neo4j is a property graph database that allows for the storage and querying of graph data in a highly efficient and scalable manner.

To facilitate this process various Python libraries are exploited including:

- Neo4j consultancy
- Neo4j's Python library
- Neo4j's Cypher

Communication with THREDDS and NetCDF is simple and performant using the libraries provided by Unidata, as these technologies were designed for high throughput of aggregated data and largely backed by a C interface.

When performing an initial loading procedure, Neo4j's CSV bulk import tool is used to send ~28 GB worth of CSV text files into a hierarchical graph modeled exactly as it is in each catalog on the THREDDS server. The advantage of this hierarchy is to provide a mechanism for operating on the graph in chunks to avoid memory intensive write queries.

A major structural change lies at the NetCDF data file level. Rather than storing contiguous large arrays for each variable component, each individual observation is given a node for auxiliary spatial-temporal components, and related nodes for ancillary data variables. This is motivated by the fact that SAMOS observations always incorporate space and time, but vary regarding oceanographic measurements.

Spatial-Temporal Modeling

- Neo4j Spatial is a library plugin which is part of the Neo4j open source contribution ecosystem. It provides an array of spatial procedures, as well as a mechanism to deploy an R-tree index directly into a graph.
- This model was deployed and tested with a subset of SAMOS, but demonstrated significant performance problems. This is likely due to the nature of the data, in that each ship has a dense trajectory reporting spatial measurements at very high resolution in a singly linked fashion.
- A large requirement of accessing SAMOS data is querying on space and time. Originally when testing the Time-tree and R-tree models, a CYPHER statement would request distinct results from each tree, and then intersect each result set. This proved to be unfeasible, however the Time-tree is still a performing candidate. That being said, the plugin is being actively developed and may be useful for other spatial operations.

A Time-Space Model

- Many researchers have studied the problem of spatial-temporal structures in different kinds of applications. For querying SAMOS, a structure is needed that can eliminate large portions of the graph during early query processing given some spatial-temporal criteria.
- Examples covered in literature include a 3D R-tree, MVR-tree, HR-tree, TB-tree, SETI, and others which each have domain specific advantages and disadvantages.
- The TB-tree and SETI models work by aggregating intervals of point data into line polygons for indexing. These will be explored further in future work.

Currently under development is a model featuring a Time-tree that incorporates metadata to suit the needs of spatial queries. This was inspired by pyramid structures used in computer vision applications.

When elements are loaded into the tree, an agglomerative process merges spatial features from the event-leaf nodes up to the root node.

The shapely polygon package is used to merge point data into polygons using a convex hull. Up to 60 leaf nodes for an hour will be used for each ship. This same process is used when merging polygons in the internal nodes of the tree, however a cascaded union precedes the convex hull computation. Shapely is backed by OSGEO, a C++ library, and therefore is fast.

Future Work

- The majority of data engineering work has been done, though there is much to be done for performance testing, and investigation into the internals of Solr and Neo4j.
- Benchmarking is non-trivial since each component is housed on the same machine and competing for resources. This is not an ideal environment, so administration work will be done to normalize each testing suite.
- More research will be covered regarding spatial-temporal databases and other relevant work. It would be interesting to see if other models can be incorporated into the graph for comparison against the model implemented for SAMOS.
- SAMOS is hierarchical in nature and extremely dense. It is possible that graph solutions may not have a competitive advantage for the requirements of DOMS. Graph databases are typically used to discover or represent relationships between objects. While each ship in SAMOS is managed separately in the tree, there may be merit in exploring new relationships between them at various scales of time resolution.

Finally, it is possible that repurposing this graph structure into a metadata utility might be useful for DOMS, or even other work at COAPS. For example, it might be useful as a mechanism to cache matched data in DOMS, whether it be for individual records or even metadata regarding aggregations of matches. Future work might involve using DOMS to collect matched data and generating metadata using the time space model. Depending on the volume of data which is discovered in DOMS, it may be possible to store a global set of matched records which could be accessed directly for a dramatic improvement in query response time to a user.