Pacific Northwest
NATIONAL LABORATORY
*Proudly Operated by* Battelle *Since 1965*

# Graph Query Language

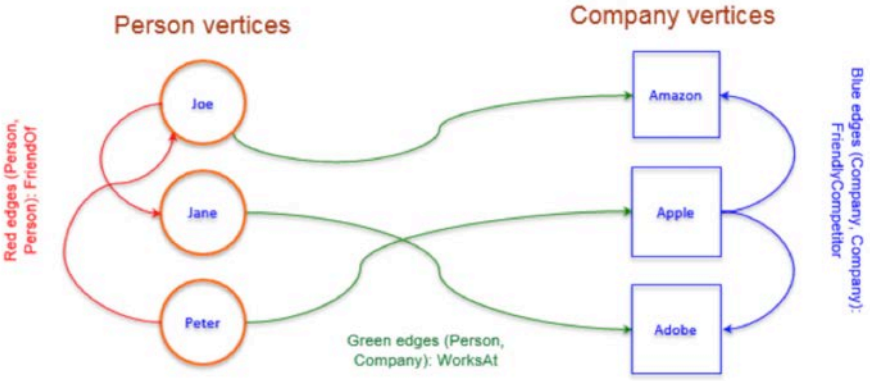
## CHALLENGE

The reality of big data is that working with $10^3$ data items is quite different from working with $10^{12}$ items. Whether streaming or loading, in memory or on disk, methods for data handling, *question* asking, and *answer* handling must be appropriate for the target data—for both engineering and human interaction.

**Building a toolkit to enable the interactive, analyst-driven exploration of very large data represented in a property graph format.**

## CURRENT PRACTICE

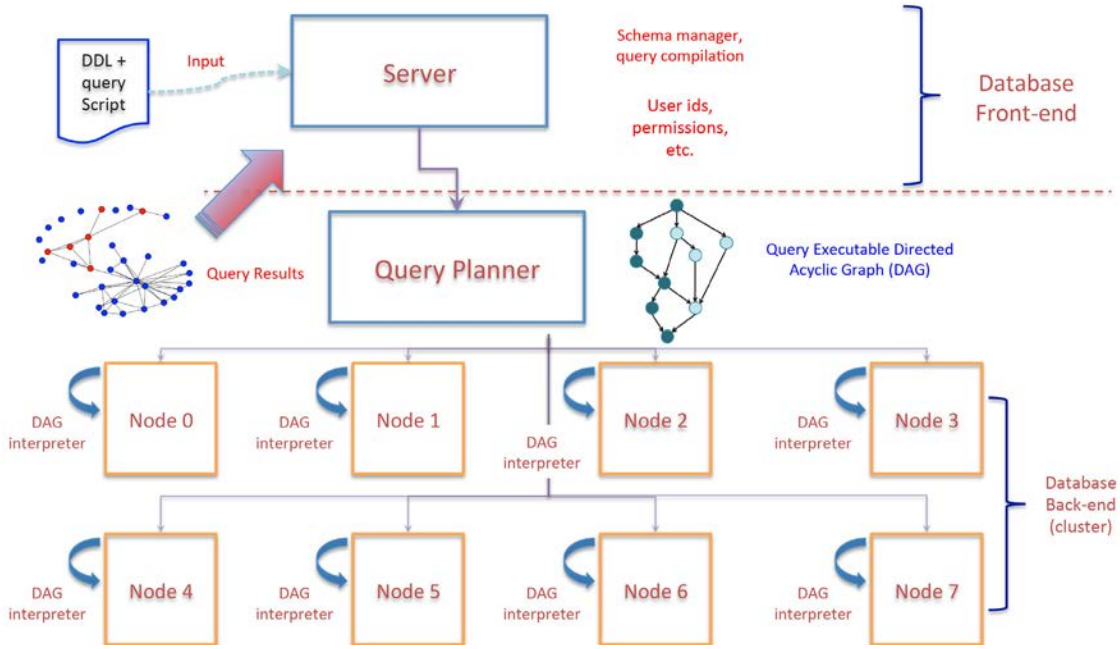
Big data analysts almost always engage in *hierarchical inquisition*. First, coarse summaries or broad filters are applied to the largest collection of information, identifying a small fraction of the data. Statistical, visualization, or other analytic approaches can be applied to those results, which hopefully inform the next round of querying, yielding a third, smaller data set. In this way, analysts can move from the $10^{12}$ through the $10^9$, $10^6$, $10^3$, and finally human-interpretable scales—ultimately identifying specific targets, patterns, or statistics of interest.




Example of an attributed graph with strongly typed vertices and edges.

Most graph database systems available today suffer limitations derived from several sources, including their data model, query language, and scalability of their implementation technology. We have designed a language interface, GraQL, to the GEMS 2.0 graph database backend that addresses these challenges. GraQL enables the definition, ingestion, and query of massive property graph data distributed over the aggregated memories of a high-performance cluster. The language design is based on a tabular data store with a graph view imposed on top of the source tables. All data elements (tables, columns, vertices, and edges) are strongly typed. The rationale behind these design principles is to enable efficient yet rich and flexible data representation for different databases uses, while allowing for a clear mapping of the data and execution to a distributed memory cluster.

While allowing richer data models, other database technologies, suffer because of their choice of implementation technologies, supporting only single-node, shared-memory implementations or relying on technologies that are not specialized for graph databases or technologies with high overhead for distributed data.




Interpretation and execution model for the GraQL query language on the GEMS 2.0 database, executing on an 8-node cluster.

## TECHNICAL APPROACH

We are designing and prototyping an easy-to-use, fully functional, and scalable graph query language. The key factors affecting design choices are scalability (in terms of data size and compute nodes), functionality, and programmability. To overcome the challenges identified in currently available graph database systems with respect to their query and data manipulation languages, we propose that the language and data model should have the following properties:

- Flexible definition of attributes for vertices and edges with *strong static typing properties* to enable better query optimization and validation
- Query interface that enables the expression of complex path queries of arbitrary length, with the possibility of composing individual paths into sophisticated graph patterns
- Flexible representation of *query results* as both lists (or tables) of elements and properly formed subgraphs with equivalent properties to the original graph
- Storage and manipulation of query results to be used by further queries
- Procedural control flow and execution capabilities to enable execution of algorithms that are more sophisticated than what can be expressed in pure declarative forms.

## IMPACT

Current technologies that support graph query and results exploration are limited and, as such, not well suited to the needs of analysts when dealing with massive graphs. Providing the capabilities proposed in terms of a scalable and flexible query language to manipulate massive attributed graphs affords an attractive proposition to enhance the computational capabilities available to analysts.


## Contacts

**Daniel Chavarría**
Principal Investigator
(509) 372-6964
daniel.chavarria@pnnl.gov

**John R. Johnson**
Program Director
(509) 375-2651
John.Johnson@pnnl.gov


Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965