Trueno: A High Performance Graph Datastore and Computational Engine

Victor Santos, Servio Palacios, Edgardo Barsallo, Tyler Cowman, Miguel Rivera,
Peng Hao, Chih-Hao, Mehmet Koyutürk, Ananth Grama

Abstract

Handling large and complex dynamic graph datasets poses significant computational challenges. Existing graph databases such as Neo4J, TitanDB, OrientDB, and others, offer limited functionality making the management of large networks a challenge, from points of view of analyses and presentation. TruenoDB is a novel integration of highly optimized algorithms and implementations, with distributed search engines, graph-parallel computations on top of a data flow-framework, and a rich set of drivers. TuenoDB provides a facile API for developing plugins, has extensive language support, interfaces with commonly used execution engines such as Spark and Mapreduce, and includes a library of graph analytics kernels. Through a number of micro and macro benchmarks, we demonstrate the excellent performance, scalability, and flexibility of TruenoDB in the context of diverse applications ranging from computational systems biology to information retrieval.

Building Scalable Graph Database Software

- ❏ Scalability in graph sizes and distributed processing infrastructure is critical
- ❏ Support for advanced features, including dynamic networks, compression, and replication
- ❏ Versioning in graph databases is a critical (and often missing) feature
- ❏ A rich analytics library, along with ability to interoperate with other analytics systems and execution engines is important.
- ❏ Finally, scalable high performance is key.

What is Trueno?

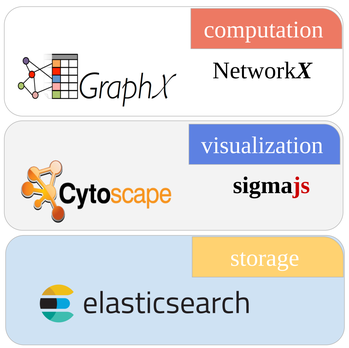
A High Performance Graph Datastore and Computational Engine

Features

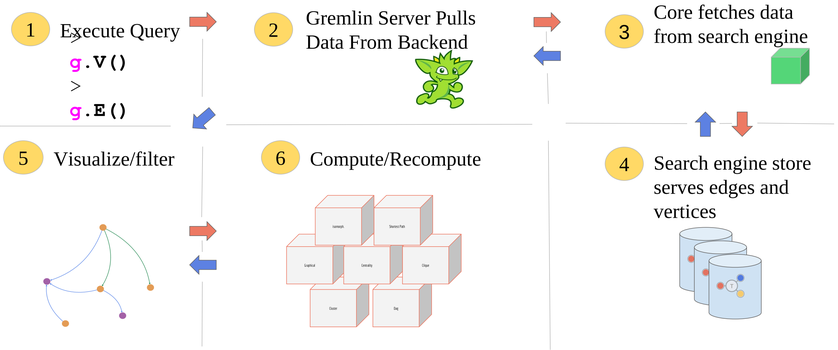
- ❏ Distributed, resilient, and highly optimized graph store.
- ❏ Support for distributed computation.
- ❏ Rich set of online queries.
- ❏ Scalable to billions of nodes and edges and tested to tens of compute nodes.
- ❏ Easy setup for both cluster and single instance installations.
- ❏ User friendly and intuitive interface for graph analysis, fast processing and visualization.

Leveraging existing infrastructure

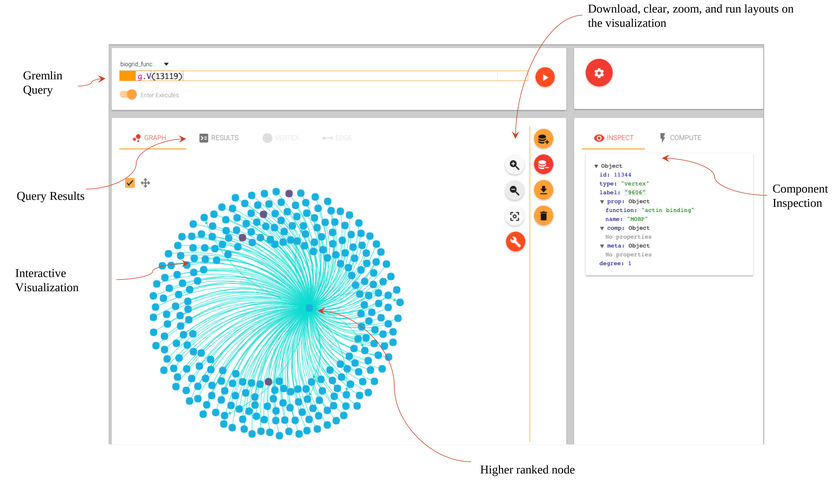
- ❏ **Gremlin(Apache Tinkerpop):** A graph traversal language for intuitive and easy graph analysis.
- ❏ **Web Console**: Web Interface for graph processing, analytics, visualization, and database management. Data laboratory that connects directly to the database/processing engine.
- ❏ **Trueno Core**: Database/Computational Engine Core.
- ❏ **ElasticSearch**: A distributed, RESTful search and analytics engine capable of handling a growing number of use cases.
- ❏ **Apache Spark:** a fast and general engine for large-scale data processing. Used for Distributed Graph Processing (**GraphX**).



Graph Exploration Execution Flow

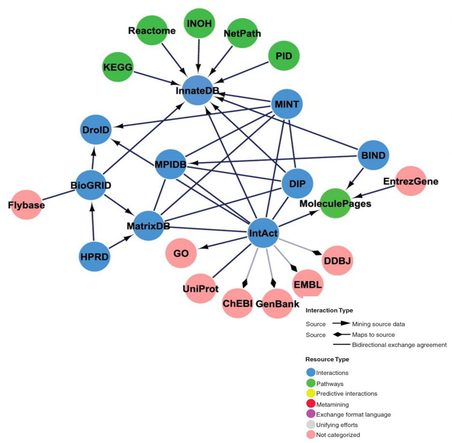


Trueno Laboratory (Preview)



Case Study: Biological Network Databases

- ❏ Protein-protein interactions
- ❏ Protein-DNA interactions
- ❏ Regulatory elements and genes
- ❏ Small molecule-protein interactions
- ❏ Enzyme-substrate interactions
- ❏ microRNA-mRNA interactions
- ❏ Functional or statistical interactions among genes and/or genomic elements (e.g., synthetic lethality, eQTL)
- ❏ Orthologs across species
- ❏ Functional associations
- ❏ Phenotype associations



Case Study: Constructing Tissue Specific Interactomes

[Mohammadi & Grama, ISMB 2016]

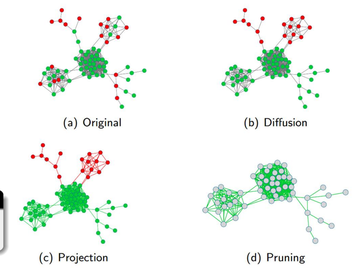
Constructing Tissue-Specific Interactomes

Available data sources:

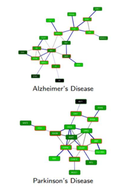
1. A global interactome, which contains the set of *possible* interacting pairs.
2. A tissue-specific measurement of gene/protein activity within each tissue/cell type.

Definition

How can we optimally utilize transcriptional activity of gene products to construct the most informative tissue-specific sub-network?



Results: Identifying Novel Disease Related Pathways

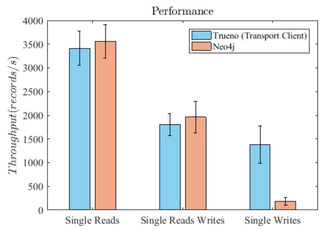


- Prize Collecting Steiner Tree (PCST)

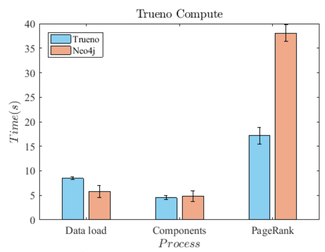
$$\underset{<v,e>\in T}{\operatorname{argmin}} \left\{ \sum_e c_e - \lambda \sum_v b_v \right\}$$

- Solved using known message-passing algorithm
- *ActPro* successfully reconstructs disease-related pathways in tissue-specific disorders

Evaluation:

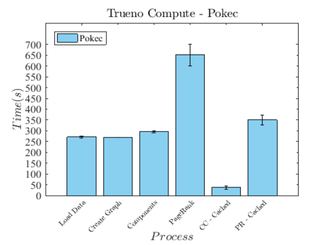


Trueno and Neo4J Performance Comparison



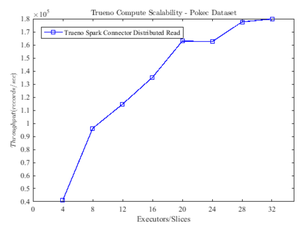
Trueno and Neo4J Compute Engine Comparison

Biogrid dataset (15,034 Vertices, 301,685 Edges)



Trueno Compute Evaluation

Pokec dataset (1,632,803 Vertices, 30,622,564 Edges)



Trueno Spark Connector - Distributed Read Scalability

Future Work and Acknowledgments

Forthcoming releases include support for dynamic graphs, compressed querying, a full versioning API and storage optimization, and support for trusted computations on graphs.

We would like to thank NSF for supporting this project and future research. We also thank faculty and students at Case Western Reserve and UC San Diego for contributing to this project.

Trueno's Github repository **github.com/TruenoDB**