Leveraging Graph Data Structures for Variant Data and Related Annotations

Chris Zawora[1], Jesse Milzman[1], Yatpang Cheung[1], Akshay Bhushan[1], Michael S. Atkins[2], Hue Vuong[3], F. Pascal Girard[2], Uma Mudunuri[3]

[1]Georgetown University, Washington, DC, [2]FedCentric Technologies, LLC, Fairfax, VA, [3]Frederick National Laboratory for Cancer Research, Frederick, MD

Background

The past decade has seen a significant increase in high-throughput experimental studies that catalog variant datasets using massively parallel sequencing experiments. New insights of biological significance can be gained by this information with multiple genomic locations based annotations. However, efforts to obtain this information by integrating and mining variant data have had limited success so far and there has yet to be a method developed that can be scalable, practical and applied to millions of variants and their related annotations. We explored the use of graph data structures as a proof of concept for scalable interpretation of the impact of variant related data.

Introduction

Traditional approaches of data mining and integration in the research field have relied on relational databases or programming for deriving dynamic insights from research related data. However, as more next generation sequencing (NGS) becomes available, these approaches limit the exploration of certain hypothesis. One such limitation is the mining of variant data from publicly available databases such as the 1000 genomes project and TCGA.

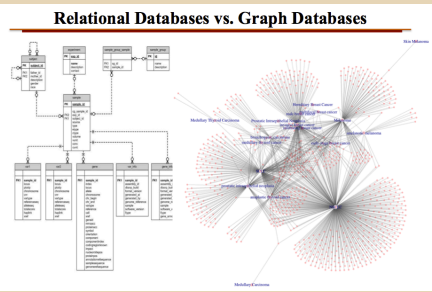

Fig. 1: Graphs handle data complexity intuitively and interactively

Although there are applications available for quickly finding the public data with a certain set of variants or for finding minor allele frequencies, there is no such application that can be applied generically across all the projects allowing researchers to globally mine and find patterns that would be applicable to their specific research interests.

In this pilot project, we have investigated whether graph database structures are applicable for mining variants from individuals and populations in a scalable manner and understanding their impact by integrating with known annotations.

Methods and Materials

Hardware

- FedCentric Labs' SGI UV300 system: x86/Linux system, scales up to 1,152 cores & 64TB of memory
- Data in memory, very low latency, high performance (Fig. 2)

Event	Latency	Scaled	Capacity
1 CPU Cycle	0.3 ns	1 s	KB
Level 1 cache access	0.9 ns	3 s	MB
Level 2 cache access	2.8 ns	9 s	MB
Level 3 cache access	12.9 ns	43 s	MB
Main memory access (RAM)	120 ns	6 min	TB
Solid-state disk (SSD)	50 - 150 us	2-6 days	TB
Rotational disk I/O	1-10 ms	1-12 months	PB
Internet: SFO to NYC	40 ms	4 years	ZB
Internet: SFO to U.K.	81 ms	8 years	ZB
Internet: SFO to AU	183 ms	19 years	ZB
TCP packet retransmit	1-3 s	105-317 years	ZB
OS virtualization system reboot	4 s	423 years	ZB

Fig. 2: Latency matters

Methods and Materials (cont.)

Graph Architecture

- Sparsity Technologies' Sparsity Graph database
- API supports C, C++, C#, Python, and Java
- Implements graph and its attributes as maps and sparse bitmap structures
- Allows it to scale with very limited memory requirements.

Data

- SNPs from 1000 genomes project
- Phenotype conditions from ClinVar
- Gene mappings & mRNA transcripts from Entrez Gene
- Amino acid changes from UniProt

The Graph

- Variants and annotations mapped to reference genomic locations (Fig. 3)
- Includes all chromosomes and genomic locations
- 180 million nodes and 12 billion edges.

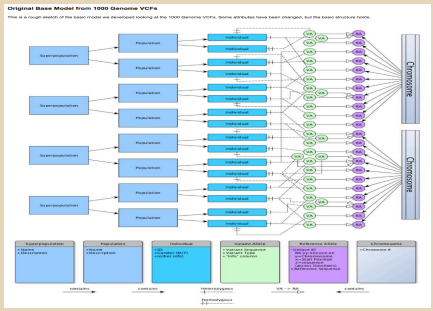

Fig. 3: The Graph Model

Results

Phase I: As an initial evaluation of the graph structures we ran several simple queries, also feasible through a relational architecture, and measured performance speeds.

Simple Query Examples

- Get all information for a single variant
- Find annotations within a range of genomic locations
- Find variants associated with specific clinical phenotypes

Performance speeds

- Query times in milliseconds
- Better or equal to relational database query times

Queries

- Developed a new SQL-like query language called SparkQL
- Eases writing queries for non-programmatic users

Ingestion Times

- Slower than expected
- Sparksee is a multi-thread single write database
- Writes one node/edge at a time
- Each write involves creating connections with existing nodes
- Slows down as the graph size increases
  Solution: Implement multi-threaded insertions in combination with internal data structures to efficiently find nodes and create edges

High Degree Vertices

- Nodes with millions of edges
- Stored in a non-distributed list like format
- Searches for a specific edge might be slow
  Example: nodes representing individuals with millions of variants
  Solution: Explore other graph clustering approaches that can essentially condense the information presented

Results (cont.)

Phase II: We explored complex patterns and clusters inside the graph and spectral clustering queries that were not feasible through the relational architecture.

Complex Query Examples

- Compare variant profiles and find individuals that are closely related
- Compare annotation profiles to find clusters of populations

Phase II Results

- Eight populations with 25 individuals from each population
- Strong eigenvalue support (near zero) for 3 main clusters
- Cluster pattern supported by population genetics (Fig. 4)

Performance speeds

- Spectral clustering took ca. 2 minutes

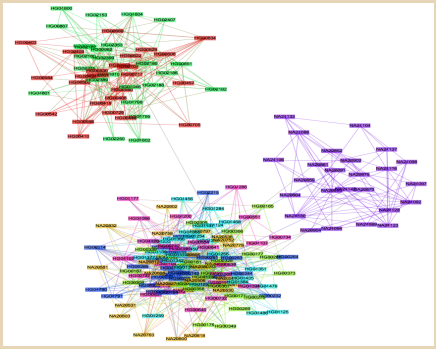

Fig. 4: *Results of spectral clustering of 1000 Genomes data*

Conclusion

Our results indicate that a graph database, run on an in-memory machine, can be a very powerful and useful tool for cancer research. Performance using the graph database for finding details on specific nodes or a list of nodes is better or equal to a well-architected relational database. We also see promising initial results for identifying correlations between genetic changes and specific phenotype conditions.

We conclude that an in-memory graph database would allow researchers to run known queries while also providing the opportunity to develop algorithms to explore complex correlations. Graph models are useful for mining complex data sets and could prove essential in the development and implementation of tools aiding precision medicine.

References

The 1000 Genomes Project Consortium. (2010). A map of human genome variation from population-scale sequencing. *Nature,* 467, 1061–1073.

Gregg, B. (2014). Systems performance: enterprise and the cloud. Pearson Hall: Upper Saddle River, NJ.

Acknowledgments

FedCentric acknowledges Frank D'Ippolito, Shafigh Mehraeen, Margreth Mpossi, Supriya Nittoor, and Tianwen Chu for their invaluable assistance with this project

This project has been funded in whole or in part with federal funds from the National Cancer Institute, National Institutes of Health, under contract HHSN261200800001E. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products, or organizations imply endorsement by the U.S. Government.

DEPARTMENT OF HEALTH AND HUMAN SERVICES • National Institutes of Health • National Cancer Institute

Frederick National Laboratory is a Federally Funded Research and Development Center operated by Leidos Biomedical Research, Inc., for the National Cancer Institute

Contract HHSN261200800001E - Funded by the National Cancer Institute