USC Viterbi School of Engineering
DARPA

GoFFish XDATA

Prof. Viktor Prasanna (PI) (prasanna@usc.edu)
Prof. Raghu Raghavendra (Co-PI) (raghu@vsoe.usc.edu)

Overview

Sensors and online instruments performing high fidelity observations are contributing in a large measure to the growing big data analytics challenge. These datasets are unique in that they represent events, observations and activities that are related to each other while being recorded by independent data streams. **GoFFish** (Graph-Oriented Framework for Foresight and Insight using Scalable Heuristics) is a scalable graph-oriented analytics framework well suited for processing reservoirs of interconnected distributed data fed by event data generators. It minimizes the communication overhead by grouping together tightly bind data.

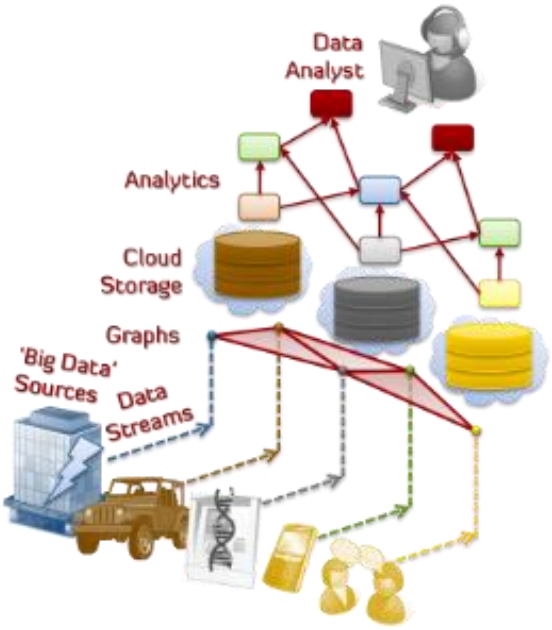

Figure 1. GoFFish concept.

Objectives

Efficiently store interconnected data by using a specialized graph-oriented file system:

- ✓ Support for **widely used graph formats** such as GraphML
- ✓ Take advantage of the various graph information and layout to enable **efficient data loading**
- ✓ Facilitate storing **temporal data** to enable **analytics on evolving graphs**
- ✓ Enable **distributed information storage** in-line with current trends observed in **cloud systems**

Efficiently compose graph analytics on large datasets by grouping data together:

- ✓ Adopt a sub-graph centric approach which **performs more computations locally** and **reduces communication** overhead
- ✓ Offer **high level programming constructs** at sub-graph level that hide low level graph details
- ✓ from the developer
- ✓ Customizable **output** format that can be **mapped to any** format needed by **visualization tools**

Enable fast analytics on certain classes of analytics and graphs:

- ✓ **Performance improvement** for analytics that require sub-graph knowledge, e.g., **community analytics**, **high impact nodes**, **shortest paths**
- ✓ Suited for graphs that can be partitioned evenly across processors with minimal communication between them, e.g., **sparse graphs** (road networks), **Internet/network graphs**

Benefits

GoFFish offers several key benefits to developers…

- ✓ Conversion pipeline from widely used graph formats such as GraphML
- ✓ Specialized graph-oriented file system to easily store interconnected evolving data
- ✓ High level API for composing graph analytics on static and evolving graphs
- ✓ Easily customizable output for interfacing with various visualization tools

Prof. Viktor Prasanna (PI) (prasanna@usc.edu)
Prof. Raghu Raghavendra (Co-PI) (raghu@vsoe.usc.edu)

…and to analysts:

- ✓ Support for multiple graphs and analytics
- ✓ Drive analytics on evolving graphs
- ✓ Ability to incorporate new analytics

Specifications

GoFFish is designed as a layered architecture with two main components: the GoFS graph file system and the Gopher analytics abstraction on top of it.

GoFS: A conversion pipeline from the GraphML format to the GoFS file format allows easy conversion of any graph. The conversion process relies on a graph partitioning stage at which point the graph is split into sub-graphs to balance its subsequent execution on distributed cloud infrastructures.

GoFS is very versatile with respect to the graphs it can integrate. Numerous graphs ranging from road networks to social networks have been successfully converted.

Gopher: GoFFish uses a high level API to intuitively and rapidly compose graph and event analytical models. The composed application enhances data parallel analytics beyond the traditional Map Reduce models using a novel distributed data partitioning approach based on edge distance heuristics. This allows unprecedented insight from the reservoirs of evolving data for commanders to perform causal graph analysis and strategic planning.

Measures of effectiveness

We evaluated our platform on a configuration representative for both clouds and commodity clusters. The system comprised of cluster of 12 nodes, each with an 8-core Intel Xeon CPU, 16 GB RAM, 1 TB SATA HDD, and connected by Gigabit Ethernet. We have compared the platform against the main competitor, Apache Giraph. Both systems were installed using Java 7 JRE for 64 bit Ubuntu Linux. The datasets consisted of three **real world graphs**: California road network (1.6M nodes x 2.7M edges), a network trace route (20M nodes x 23M edges), and the Live Journal social network (5M nodes x 65M edges). Different graph analytics such as connected components, shortest path and page rank were deployed on it and their speed-ups were measured. An **average improvement of 10x** was observed.

Future enhancements

Numerous enhancements are in progress, most of them related to dealing with detecting online events that can occur in evolving streaming graphs. The loop between insight and foresight will be closed by coupling event patterns mined from historical stream reservoirs with graph analytics based on real-time event streams from sensors.

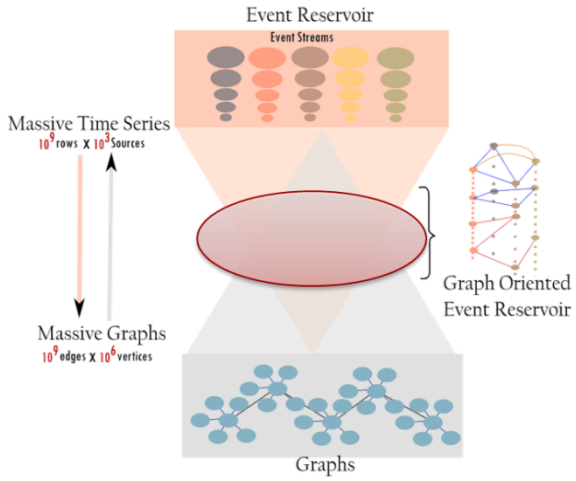

Figure 2. Evolving streaming graphs.

Prerequisites

Java skills

Knowledge of **Linux** and command line interaction

Knowledge of GraphML format