Multi-Scale Community Visualization of Massive Graph Data

Scott Langevin, *Member, IEEE*, David Jonker, David Giesbrecht, and Michael Crouch

Abstract— Graph visualizations increase the perception of entity relationships in a network. However, as graph size and density increases, readability rapidly diminishes. In this paper, we present a tile-based visual analytic approach that leverages cluster computing to create large-scale, interactive graph visualizations in modern web browsers. Our approach is optimized for analyzing community structure and relationships.

Index Terms—Large data visualization, graph/network data, data clustering, distributed computing, multi-resolution techniques, scalability issues, interaction design, zooming and navigation techniques

INTRODUCTION

As scientists, government agencies, and businesses increasingly require insight from massive sets of relational data, there is a growing need for scalable visual graph analytics that can reveal patterns and anomalies in big data. However, visualizing terabytes or even petabytes of information is not without prohibitive perceptual and computational costs: traditional graph visualization tools that run on a single workstation often cannot process or render massive data and, when they do, produce overcrowded "hairball" results with ineffective labelling that make nuanced interaction and investigation extremely difficult.

Graph visualizations are frequently employed to support understanding of relationships between entities. Of particular importance is the detection of communities of highly related nodes and awareness of their relations both internally and externally to other communities. However, as the size and density of a graph increases the ability to perceive and understand these nuanced relations quickly deteriorates.

Our approach to addressing these issues is to leverage open-source cluster-computing frameworks (Apache Hadoop [1] and Spark [2]) and a tile-based visual analytics methodology [3][4] to create interactive large-scale graph visualizations by: 1) extracting hierarchical communities in the data; 2) applying a distributed, recursive layout algorithm to align nodes according to their hierarchical community membership; and 3) producing a two-dimensional, multi-scale graph visualization with a familiar, web-based map interaction scheme that supports simple pan and zoom navigation.

1 BACKGROUND

While computational algorithms can quickly derive complex answers from massive node-link datasets, they continue to lag behind the human ability to perceive and understand visual patterns and anomalies [5]. As such, technical and non-technical audiences alike require interactive visual graph analytics to:

- Assess the believability or perception of truth in answers computed with machine learning.
- Place the aforementioned answers in the proper context.
- Discover nuances or patterns that are not typically identified by computational algorithms.

By placing node-link data in an interactive visual analytic tool, users are able to apply their natural visual acuity to identify clusters and communities of related nodes, discover closely connected nodes that suggest similarity or affinity, and understand the structure of organizations [5]. This visual mapping of data allows users to comprehend the spatial representation of complex data, retain mental models of data organization, and detect anomalies and patterns for further investigation.

Many current graph visualization tools are designed to operate on a single workstation. These tools are limited to node-link data that can fit in the memory of a single machine, which often means they cannot scale past thousands of nodes [6]. This places rigid restrictions on a user's ability to work with big data.

More recently, graph analytic tools (e.g., GraphCT and SNAP) are taking advantage of distributed and parallel computing systems to handle node-link datasets with millions or billions of nodes [6]. In addition to scaling to massive data sizes, these approaches are fault-tolerant preventing costly restarting of computations [7].

Large-scale graph data pose challenges to existing visual graph analysis approaches, requiring new techniques to overcome the following issues:

- **Resource-intensive computations are required to establish optimal graph layouts** that reveal structures such as communities and connectivity.
- **Community clustering and layouts are typically applied separately**, causing occlusion of clusters and loss of information about the structure of individual clusters and inter-cluster connectivity.
- **Large graphs are too big to fit in memory** of a single machine.
- **Rendering graphs of millions+ of nodes and edges** is time consuming. Showing all nodes and links often results in hairballs that hinder sense making.

1.1 Related Work

Previous work has investigated community-based layouts for large graph visualizations. In HC-drawing [8], graphs are hierarchically clustered, and a graph layout algorithm aligns nodes according to community membership. Fitted-Rectangles [9] uses a similar non-hierarchical approach, but also provides aggregation of node links. In Group-in-a-Box [10], nodes are grouped either semantically along attributes of interest or community, and then laid out using a treemap space filling technique. NodeTrix [11] visualizes graphs with communities as nodes and links depicting inter-community connections. Community nodes are visualized as adjacency matrices to reveal community structure. Zame [12] uses a tile-based approach to provide an interactive multi-scale visualization of a graph adjacency matrix.

Our work differs from previous approaches: 1) we leverage cluster computing to efficiently cluster, lay out and compute analytics on graphs from datasets scaling to terabytes; 2) a tile-based visualization approach facilitates efficient interactive multi-scale graph analysis of node-link diagrams; and 3) flexible tile-based

- *Scott Langevin, David Jonker, David Giesbrecht, and Michael Crouch are with Uncharted Software Inc.*
  E-mail:{ slangevin, djonker, dgiesbrecht, mcrouch} @uncharted.software

visual analytics provide information overlays summarizing community attributes.

2 Tile-Based Visual Analytics

Our approach to creating interactive visualizations of massive node-link graph data employs tile-based visual analytics to enable investigation using common web browsers. The methodology is implemented as open-source software built on Apache Spark and Hadoop [13].

A cluster-computing and parallelization framework generates multi-resolution tiled datasets with analytics and aggregate summaries for each tile. Tiles are served and rendered on demand as images in an interactive web-based map client. The client allows users to pan and zoom through increasingly detailed views of the source data, a "tile pyramid" that spans global to local scales, much like Google Maps offers both aggregate views of the entire world and street-level depictions.

The tile generation process uses Apache Spark to convert character-delimited or GraphML source data into a set of Apache Avro data tiles that summarize the individual data points at multiple resolutions. A tile service delivers the tiled data to the web client either as a set of rasters or a JSON payload for client-side rendering.

Users can filter tile views by attributes such as time to apply visual metaphors on the fly. All analytic overlays leverage the same underlying data. Tile-based visual analytics support cross-plot, geospatial, time series, and force-directed graph layouts of big data.

3 Visual Analysis of Graph Data

Generating a visual graph analytic (Fig. 1) is a multi-stage process of detecting communities of nodes, laying them out in a hierarchical manner, and generating the data tiles used to serve the transformed source data to the web client for presentation.

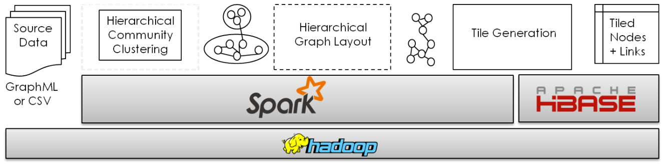

Fig. 1 Graph Tiling Pipeline

Once all of the stages have been executed, an efficient representation of the graph data is available for interaction in any modern web browser.

3.1 Hierarchical Community Clustering

To detect and cluster nodes that are highly connected, we apply a distributed version of the Louvain Modularity algorithm [14] to the source data using the Apache Spark GraphX library. Strongly connected nodes form communities at several different hierarchical levels: low-level communities are detected from the raw data, those communities are then clustered accordingly at the next highest level, and so on up the chain to the highest (global) hierarchical level.

This stage is optional if the source data already has location information (e.g., x/y coordinates) associated with the nodes.

3.2 Hierarchical Graph Layout

Informed by the network and community structure, the *Hierarchical Graph Layout* stage positions the nodes to highlight visual groupings of communities. A distributed, recursive, force-directed layout algorithm was developed to lay out communities from the top-level hierarchy down to raw nodes.

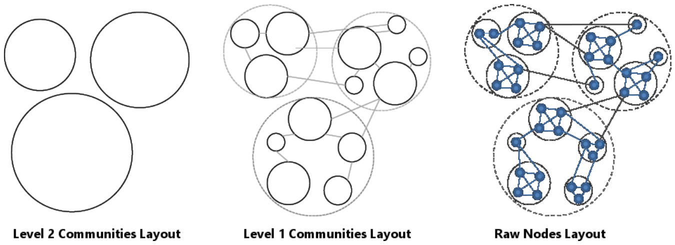

Fig. 3 Hierarchical Graph Layout

As shown in Fig. 3, each community is drawn as a virtual node sized to indicate the number of nodes that it contains. The force-directed algorithm [15] simulates repellent forces between nodes and spring forces on links to naturally group and cluster connected nodes. The algorithm starts with the top-level communities, laying them out independently of the lower-level structure. The next lowest level of communities is then laid out within the spatial constraints of the parent level, and so on.

Applying a recursive force-directed layout to communities reduces hairball results by increasing visual separation and distinguishing communities and the relationships between them. The resulting proximity reflects strength of relationship. Further, the recursive nature of the layout algorithm is parallelizable into independent layout subtasks that can be efficiently computed.

The Hierarchical Graph Layout stage is run on Apache Spark, allowing parallelization and reducing computational costs.

As with the Hierarchical Community Clustering, this stage is not required if the source data already has location information associated with the nodes.

3.3 Graph Tile Generation

Once the source graph data has been clustered and laid out, the positioned node and links are passed to the *Tile Generation* component to create a hierarchical pyramid of data tiles that summarize the graph at multiple resolutions, from global to local scales. Tiling data instead of graphics allows for responsive interfaces with capabilities such as runtime filtering, which can be used, for example, to filter out all links below a certain weight.

The tile hierarchy created during this stage enables smart decisions about the level of detail of nodes and links in the user interface, which can increase rendering efficiency. For example, low-level or off-screen links can be omitted at each zoom level. Moreover, only the data necessary to render the current views needs to be delivered to the browser, making it possible to explore billions of linked entities at interactive speeds.

Massive graph tile pyramids are saved in an Avro format, typically to an HBase table to enable distributed file storage and scalability to billions of tiles.

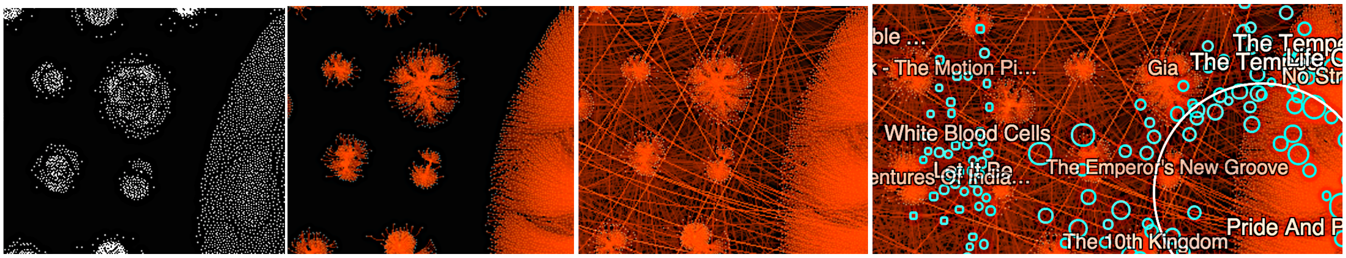

Fig. 2 Graph layout elements: (a) nodes, (b) intra-community links, (c) inter-community links, and (d) communities and labels

3.4 Tile-Based Visual Analytics for Graphs

When the Tile Generation stage is complete, the tile pyramid is served to the web client, where the entire graph is made available for analysis through pan, zoom and layer interactions familiar to web-based map services. Each visual element type is displayed as a separate layer that can be independently filtered or hidden resulting in an interactive graph with a trillion or more "pixels" of resolution.

Nodes (Fig. 2a) are spatially arranged to reflect relationships and the communities to which they belong. Node weight may be optionally indicated through color. A consistent pixel size is used for display at each zoom level to ensure clarity. Controls are provided to dynamically adjust node diameter for greater or lesser emphasis.

Intra-community (Fig. 2b) and inter-community (Fig. 2c) links are rendered as separate layers, allowing end users to tailor relative emphasis to support analytic interest. Links can be weighted to represent the strength of relationships between the nodes they connect, and are visualized as a heatmap to depict strength, distribution and density of clusters of edges. To avoid visual clutter that can otherwise interfere with visibility of local connections, links leading to distant off screen nodes can be attenuated using opacity fall off.

Communities (Fig. 2d) are treated as virtual nodes within the graph layout. They are denoted by an interactive circular boundary that reveals additional community metadata when clicked. Each community is sized according to the number of child nodes that it contains. Zooming in on a community reveals sub-communities and nodes that are aggregated together at higher levels. This increasingly detailed hierarchical structure helps to maintain high rendering speeds in the browser and introduces an efficient map-service paradigm that serves only the data needed for the current viewport.

To add semantics to the display, community labels are derived hierarchically from the underlying child node with highest weighted degree centrality (i.e., sum of weights of incident edges for a give node). Additional metadata for a community can be presented such as a distribution of its member attributes.

To better express the character of communities, additional tile-based analytics can be overlaid on top of the graph. Each analytic summarizes key attributes about the nodes or links underlying the corresponding tile. These overlays summarize aspects with which to characterize visible communities, such as common topics of conversation shown as a word cloud.

4 Proof-of-Concept Applications

Our tile-based visual analytic approach to enabling analysis of large-scale graphs was developed empirically over time using a variety of real-world applications. We present two examples involving social media and e-commerce data. *Chelsea FC Fan Communities* examines social media influence amongst individuals and organizations using the Twitter social network, and *Amazon Product Affinity* maps clusters of products that interest the same people.

4.1 Chelsea FC Fan Communities

The *Chelsea FC Fan Communities* application attempts to highlight communities within the sphere of Twitter users who used Chelsea Football Club keywords in tweets during 2014. In total, the dataset contains 248,747,072 tweets with 554,430 unique account nodes (users). The application contains 100,700 relationships (links) between users who have mentioned each other in tweets.

4.1.1 Geospatial Mapping

The first approach to discovering communities in the Chelsea FC data involved mapping the geo-located tweet data based on latitude/longitude coordinates (Fig. 5). Directed, clockwise arcs between tweet locations indicate user mentions, while arc color indicates tweet density (dark blue for low density and white for high density).

Tiling tweets using their geolocation data allows for geospatial analysis of social network data and avoids the computational cost of generating a graph layout. Displaying the graph as a map focuses on spatial communities and patterns, allowing end users to analyze the high-level geographic structure and regional communication flow (e.g., between large communities in England, Spain, and West Africa).

Geolocated data is aggregated across several increasingly detailed heatmap views. End users can drill down from a global view to street level, revealing patterns at each scale.

Additional community characterization data is provided in tile analytic overlays. For example, a Player Mentions Sentiment overlay summarizes the sentiments (positive, negative, or neutral) of tweets within the region that referenced specific players. Drill down information reveals individual posts and trends over time.

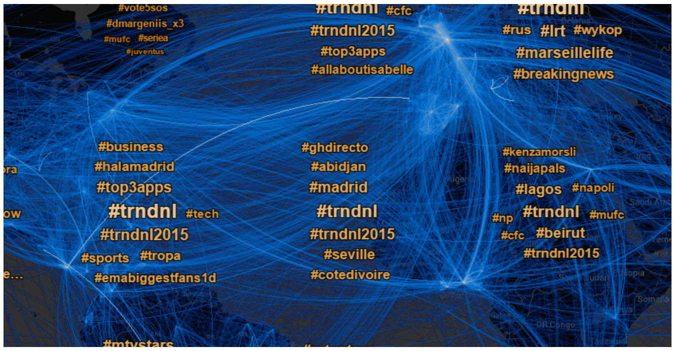

Fig. 5 Geospatial mapping of Chelsea FC Twitter mentions with directed arcs representing user mentions with top hashtag overlays

4.1.2 Community Graph

Geospatial layouts of social media are limited in that they cannot plot posts lacking location data, and only 2-3% of Twitter messages are geocoded [16]. An alternative approach to tiling the Chelsea FC Fan Communities is to use a graph layout, which supports all of the tweets plotted in the geospatial map and adds tweets that do not have latitude/longitude data. In contrast to the geospatial layout, the layout of the community graph is determined by the structure of related users, where directional arc links and the proximity of communities indicate the strength of the relationship between them.

The graph layout reveals several details that are obscured in the map layout. First, a multitude of disconnected groups exist outside the core Twitter activity, indicating that they do not interact with the community at large (Fig. 6a).

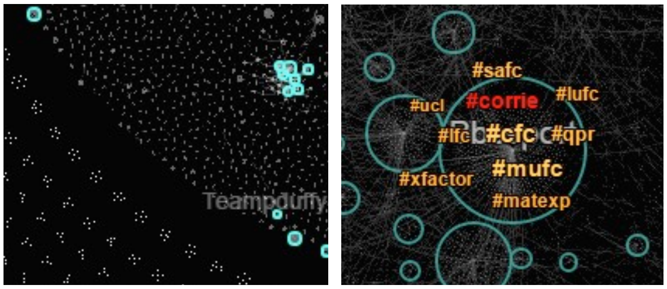

Fig. 6 Graph layout features (a) fringe communities and (b) tile analytic overlays

Each circle in the graph is a community of nodes with high cohesiveness. Reviewing community labels can reveal unexpected correlations; for example, the most central community in the Chelsea application appears to be defined by its users' shared interest in a rival football club, Manchester City FC.

As with the geographic plot, the graph layout supports multiple tile analytic overlays that provide more context of the behaviours and trends of the underlying communities. In Fig. 6b, a BBC Sports community appears to correlate with a high degree of tweets with the hashtag *#corrie*, a reference to the popular British soap opera, Coronation Street.

4.2 Amazon Product Affinity

The *Product Affinity* application is based on a Stanford dataset of Amazon product information. Compiled over nine years, the dataset includes anonymized customer product reviews and links between products based on co-purchase patterns (i.e., "customers who bought this also bought..."). Nodes in the Amazon graph represent products and anonymized customers, while the links indicate weighted customer reviews and co-purchases. The application attempts to represent spheres of interest among Amazon customers.

While the Amazon Product Affinity dataset (with 2,372,409 nodes and 9,909,551 links) is an order of magnitude larger than the Chelsea FC Fan Communities dataset, the speed and interactivity of the application remains unchanged.

4.2.1 Graph Layout

The graph layout of the Amazon dataset suggests product affinity. The closeness of individual products and communities in the graph indicates co-purchase habits. Reviewing the hierarchical communities or related products can reveal social demographic data about customers.

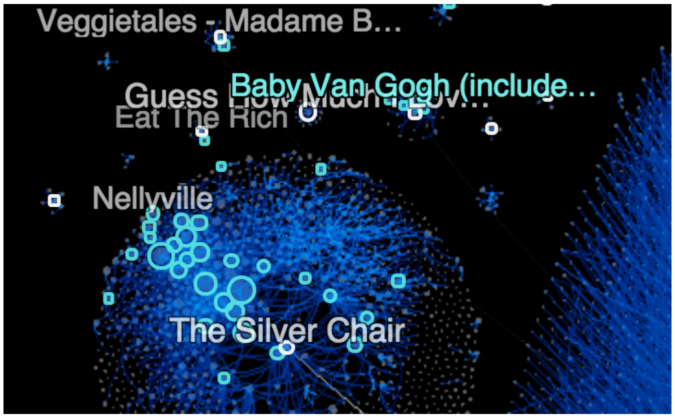

Fig. 7 Community labels reveal co-purchase habits

For example, zooming in on a community (Fig. 7) reveals clusters of products marketed primarily at young adults: rap and alternative music like Good Charlotte, fantasy novels like *Eragon* and *The Silver Chair*, and edgy animated shows like South Park. However, further analysis reveals unexpected affinities with products nearby: a cluster of pregnancy books and interactive Baby Einstein products (which itself contains surprising products: *Spanish for Health Professionals* and a KISS – Immortals DVD). Visualizing nuanced correlations in customer appeal suggests demographic patterns, such as families of a certain size and age.

Drilling down into any of these large-scale Amazon product communities reveals constellations of smaller communities, illustrating how tile-based visual analytics can introduce structure to massive market data.

5 Future Work and Conclusions

We plan to explore alternative community clustering algorithms to evaluate the resulting clustering and graph layout quality. Additionally, we will investigate developing a runtime graph analytics API for integrating on-demand graph analytic queries such as "path finding" capabilities that allow users to discover the path between any two nodes. Visual analytic overlays will present analytic results in context. Finally, we plan to continue to enrich the graph visualization with additional analytic layers for communities including summary statistics (e.g. distribution properties) that describe their makeup.

Tile-based visual analytics offer a scalable solution to the challenges of creating massive graph visualizations by parallelizing and distributing the generation process. They also offer a user experience that enables investigation of any subset of big data node-link graphs through efficient delivery of scale and context-appropriate data to the user interface. The community-based force-directed layouts, multi-resolution views and interactive labelling in our approach address problems that persist in traditional hairball layouts of graph data. This combination of computational analytics with highly expressive interactive visualization provides the opportunity for deeper understanding and trust.

Acknowledgments

This work was supported by the Defense Advanced Research Projects Agency (DARPA) under Contract Number XDATA (FA8750-12-C-0317). The views, opinions, and findings contained in this report are those of the authors and should not be construed as an official Department of Defense position, policy, or decision.

References

[1] https://hadoop.apache.org/

[2] http://spark.apache.org/

[3] P. Schretlen,N. Kronenfeld, D. Gray, J. McGeachie, E. Hall, D. Cheng, N. Covello, and W. Wright, "Interactive Data Exploration with "Big Data Tukey Plots", IEEE VIS, 2013.

[4] D. Cheng, P. Schretlen, N. Kronenfeld, N. Bozowsky, and W. Wright, "Tile based visual analytics for twitter big data exploratory analysis", IEEE International Conference on Big Data, 2013.

[5] R. Rohrer, C.L. Paul, and B. Nebesh, "Visual Analytics for Big Data", The Next Wave, 20(4), 2014.

[6] D. Ediger, K. Jiang, E.J. Riedy, and D.A. Bader, "GraphCT: Multithreaded Algorithms for Massive Graph Analysis", IEEE Transactions on Parallel & Distributed Systems, 2012.

[7] P. Burkhardt, and C. Waring, "An NSA Big Graph Experiment", Technical Report NSA-RD-2013-056002v1, U.S. National Security Agency, May 20, 2013.

[8] W. Didimo, and F. Montecchiani, "Fast Layout Computation of Hierarchically Clustered Networks: Algorithmic Advances and Experimental Analysis", In 2012 16th International Conference on Information Visualisation (IV), 18 –23, 2012. doi:10.1109/IV.2012.14.

[9] S. Chaturvedi, Z. Ashktorab, and R. Zacharia, "Fitted Rectangles: A Visualization for Clustered Graphs", 2013.

[10] E. Rodrigues, N. Milic-Frayling, M. Smith, B. Shneiderman, and D. Hansen, "Group-in-a-Box layout for multi-faceted analysis of communities", Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on. IEEE, 2011.

[11] N. Henry, J.-D. Fekete, and M. McGuffin, "NodeTrix: a Hybrid Visualization of Social Networks", Visualization and Computer Graphics, IEEE Transactions On 13, no. 6 (2007): 1302–1309.

[12] N. Elmqvist, T-N. Do, H. Goodell, and N. Henry, "ZAME: Interactive Large-Scale Graph Visualization", Visualization Symposium, PacificVIS'08, 2008.

[13] http://aperturetiles.com/

[14] V. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks", Journal of Statistical Mechanics: Theory and Experiment, 2008.10 (2008): P10008.

[15] T. Fruchterman, and E. Reingold, "Graph drawing by force-directed placement", Software – Practice & Experience 21.11 (1991): 1129-1164.

[16] K. Leetaru, S. Wang, G. Cao, A. Padmanabhan, and E. Shook, "Mapping the global Twitter heartbeat: The geography of Twitter", First Monday, [S.l.], apr. 2013. ISSN 13960466. Available at: <http://firstmonday.org/ojs/index.php/fm/article/view/4366/3654>. Date accessed: 13 Aug. 2015. doi:10.5210/fm.v18i5.4366.