Utilizing Semantic Big Data for realizing a National-scale Infrastructure Vulnerability Analysis System

Sangkeun Lee, Supriya Chinthavali, Sisi Duan, Mallikarjun Shankar
Oak Ridge National Laboratory
Oak Ridge, TN 37831, USA
{lees4,chinthavalis,duans,shankarm}@ornl.gov

ABSTRACT

Critical Infrastructure systems (CIs) such as energy, water, transportation, and communication are highly interconnected and mutually dependent in complex ways. Robust modeling of CIs' interconnections is crucial to identify vulnerabilities in the CIs. We present a vision of national-scale Infrastructure Vulnerability Analysis System (IVAS) leveraging Semantic Big Data (SBD) tools, Big Data, and Geographical Information Systems (GIS) tools. We first survey existing approaches on vulnerability analysis of critical infrastructures and discuss relevant systems and tools aligned with our vision. Next, we present a generic system architecture and discuss challenges including: (1) Constructing and managing a CI network-of-networks graph, (2) Performing analytic operations at scale, and (3) Interactive visualization of analytic output to generate meaningful insights. We argue that this architecture acts as a baseline to realize a national-scale network based vulnerability analysis system.

CCS Concepts

•**General and reference** → **Design;** •**Computing methodologies** → *Model development and analysis;* •**Networks** → Network performance modeling;

Keywords

Critical infrastructure network, interdependency, large-scale, vulnerability analysis, graph analysis, big data

This manuscript has been authored by UT-Battelle, LLC under Contract No. DE-AC05-00OR22725 with the U.S. Department of Energy. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The Department of Energy will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

SBD'16, July 01 2016, San Francisco, CA, USA

DOI: http://dx.doi.org/10.1145/2928294.2928295

1. INTRODUCTION

About a decade ago, the World Wide Web Consortium (W3C) developed a set of standards around the vision of establishing a machine-understandable Web, so called Semantic Web. In particular, the Resource Description Framework (RDF), SPARQL Protocol, and RDF Query Language (SPARQL) were introduced to build a fundamental data model and the query language enables flexible schema-free data interchange on the Semantic Web.

Interestingly, today, we are witnessing that data scientists and practitioners frequently exploit the Semantic Web standards and tools for data analysis. According to the statistics [2], over 85 billion triples from 3426 datasets have been published. Distributed execution environment via multiple SPARQL end points opens up the potential of holistic knowledge discovery from disparate large-scale data sources, which is very attractive to big data researchers. Even more, the advent of advanced large-scale RDF data processing infrastructures such as Cray's uRiKA-GD, which is a super computer specialized for RDF-processing, and their applications have shown the promising capabilities and use cases of Semantic Web technologies for big data research and applications.

In this study, we specifically focus on one of the most important real-world problems – vulnerability analysis and understanding cascading effects due to various perturbations in Critical Infrastructures (CIs). CIs such as transportation, water, and energy are significant to sustaining day-to-day commodity flows of economic and national security concerns. The nature of modern infrastructures is that failures in certain critical components can trigger widespread cascading failures causing ripple effects throughout regional or national scales, as shown in historical worldwide extreme events like the 2003 North American blackout and Hurricane Sandy in 2012. CIs are mutually dependent in complex ways, not just physically but through a host of IT/communication technologies. They also display an adaptive behavior influenced by past experience such as aging and adjustment to disturbances [37], etc. Robust modeling of the complex interdependencies of these CIs and performing vulnerability assessment are significant research challenges. While numerous researchers have focused on critical infrastructure dependencies [15, 41, 26], most of them still consider individual or a small number of critical infrastructures and perform qualitative as opposed to quantitative analyses. They fail to consider a national-scale evolving network due to various challenges, and these challenges are further exacerbated by the breadth and complexity of the large-scale CIs.

We envision that SBD technologies are well-aligned with many challenges of realizing a national-scale Infrastructure Vulnerability Analysis System (IVAS). This study aims to open up a new research area in the SBD research community by providing a vision of a SBD-based network based vulnerability analysis system. However, solely adopting SBD technologies for the problem does not simply solve all the issues. We therefore investigate how we can create a synergy of using other Big Data and GIS technologies to complement limitations or lacking capabilities of SBD tools. The rest of the paper is organized as follows. In Section 2, we present background on vulnerability analysis of critical infrastructures and various related systems and tools that can be employed for an IVAS. In Section 3, we present a generic IVAS architecture and discuss its challenges in detail. Finally, we summarize and conclude our study.

2. BACKGROUND

2.1 Infrastructure vulnerability analysis

Modeling interdependencies is not a trivial task, since it involves consideration of a wide range of dimensions such as the type of failures (common cause failure/cascading failure), type of coupling (tight/loose), state of operation (normal/stressed/post event), etc. [37]. For instance, during the hurricane Sandy in 2012, new interdependency links tend to be formed due to backup generators needing fuel, which depend on oil and gas and transportation networks. One of the most important dimensions is the type of interdependencies (physical, geographic, logical, and cyber). For instance, there exists a physical interdependency between power substations and oil refineries for energy, water stream network and generation power plants for cooling water, and a geographical interdependency among multiple CIs in the same geographic region hit by an earthquake. These interactions exist across multiple scales of space/time [35]. The shear breadth and structural complexity of our infrastructure magnifies these challenges.

Previous research efforts in studying infrastructure interdependencies have used various modeling and simulation approaches which are broadly categorized into empirical, agent based, system dynamics based, economic theory, and network based approaches [34]. All these approaches work at the intersection of at most two CIs and do not expand their scope of investigation to cascading effects in a modern interconnected system. As an example, ORNL's EARSS models limit the prediction of propagation consequences of weather related threats such as hurricanes to two infrastructure networks like electric grid (substations affected) and service areas (population) impacted using GIS based approaches[11]. Most of the existing analysis has been restricted to just a small geographical region/network location due to limited access to real-world datasets, and increased problem and computational complexity.

2.2 Semantic Big Data (SBD)

SBD Standards. The World Wide Web Consortium (W3C) published a specification of Resource Description Framework (RDF) in 1999, which is a standard data model originally designed to represent resources on the Web for data exchange. Formally, A RDF dataset is a set of *triples* in ⟨*subject*⟩ ⟨*predicate*⟩ ⟨*object*⟩, where *subject* denotes a globally unique resource, *object* denotes either a unique resource or a literal (i.e., a string or a number), and *predicate* denotes a relationship between the subject and object. SPARQL Protocol And RDF Query Language (SPARQL) is a standard query language for RDF datasets endorsed by W3C. SPARQL queries are represented by a set of triple patterns that identifies a sub-graph of interest in the RDF dataset.

RDF can be considered as a *linked-data*, or a *graph* data model, since triples in a RDF dataset naturally form a graph, where subjects and objects represent nodes and predicates represent edges between them. Accordingly, SPARQL is a query language for graph (or linked-) datasets which retrieves sub-graph patterns from the graph that a RDF dataset composes. As many critical infrastructures form a network topology (e.g., streams, roads, power transmission lines, etc.), RDF and SPARQL can naturally represent and query CI data sets. The graph representation of CI has an advantage that it can provide intuitive visualization to CI operators. Also note that CI datasets can be heterogeneous in many ways including their schema, level-of-abstraction, granularity of information, unit of measure, etc. Having a specific fixed-schema structure data could significantly limit system's flexibility. RDF and SPARQL are promising standards for storing and querying various kinds of entities and their relationships without having a specific schema, which is another reason why they are well-aligned with the needs of dealing with CI datasets.

SBD for large-scale data analysis. Representing national-scale, heterogeneous CI datasets using the RDF model can result massive number of RDF triples. Despite of RDF and SPARQL's theoretic advantageous features, if there is no way to analyze such large-scale datasets in efficient and scalable ways, it is not a feasible approach to take. Earlier conventional triplestores running on a single machine such as Jena[10], Sesame [14], and RDFSuite [9] are not scalable enough to handle such large datasets. Today, thanks to recently advanced software and hardware technologies, advanced Semantic Web tools and systems, namely SBD technologies, are capable of large-scale RDF data processing.

Researchers developed various distributed triplestores such as SPARQLVerse [31], TriAD [22], and 4Store [24], which store large RDF datasets in multiple commodity machines and process SPARQL queries in parallel on the clustered machines. On the other hand, Cray presented uRiKA-GD, which is a massive-scale RDF data processing super computer, developed around their multi-threaded shared-memory hardware XMT2. These SBD technologies have enabled data researchers and practitioners to exploit the SBD tools not only for RDF storage and query but also for complex analytic purposes. To be more specific, the SBD tools can be used for *Graph analysis*, which is the general term of unveiling useful knowledge implied in graphs [30]. The increasing number of applications and researches in a wide range of domains (e.g., healthcare [25], biology [39], medical research [5], etc.) also show the usefulness of the SBD technologies for data analysis.

2.3 Big Data and GIS

Besides the Semantic Web community's efforts, a number of Big Data tools and Geographic Information Systems (GIS) have been developed so far. In this section, we briefly cover some of these systems that can be utilized to complement SBD technologies for infrastructure analysis.

Google's MapReduce framework [18] is developed for processing large-scale datasets on commodity clusters. MapReduce has been shown to be useful and effective in various Big Data applications. However, it has a limitation that its processing scheme is not suitable for iterative machine learning algorithms, as it writes intermediate data to files at each iteration. Spark [42] has gained much attention in distributed computing with commodity clusters since its processing scheme works well for iterative algorithms. Spark allows programmers to cache data in Resilient Distributed Datasets (RDDs) distributed in memory of cluster nodes without writing in files, it therefore provides much faster iterative MapReduce computations. These tools have been widely used in many fields of data practices, and they can be useful for infrastructure analysis system in many ways.

Infrastructure analysis strongly involves handling geographical data. Conventional SBD tools do not support specialized geographical operations, so combinational usage of SBD tools and Geographical Information Systems (GIS) should be carefully considered. GIS covers a wide range of systems and many research topics, but in general, main capability is to efficiently store, retrieve, analyze, and visualize spatial data. For the storage and retrieval purpose, several existing geographical DataBase Management Systems (DBMS) can be employed, where a geographical DBMS is a specialized DBMS designed to efficiently handle points and shapes on the map. PostGIS [33] is an open source extension that adds support for geographical data to the PostgreSQL DBMS. As achieving scalability of processing large-scale geographical dataset is a challenging problem, there have been approaches to enable GIS features on top of MapReduce frameworks [8, 17]. On the other hand, there is an approach to incorporate geographical data support into Semantic Web standards. GeoSPARQL [12] aims to bridge the gap between GIS and SBD systems by incorporating geographical concepts to support the SPARQL query language's feature.

3. THE VISION OF IVAS

Infrastructure Vulnerability Analysis System (IVAS) refers a system which supports subject matter experts' decision making based on the data analysis on multiple heterogeneous critical infrastructure datasets. We present a generic architecture of an IVAS. As shown in Figure 1, an IVAS is composed of three layers: Data Layer, Analysis Layer, and User-Interface. In this section, we explain the role of each layer, discuss challenges, and how we can leverage various existing technologies.

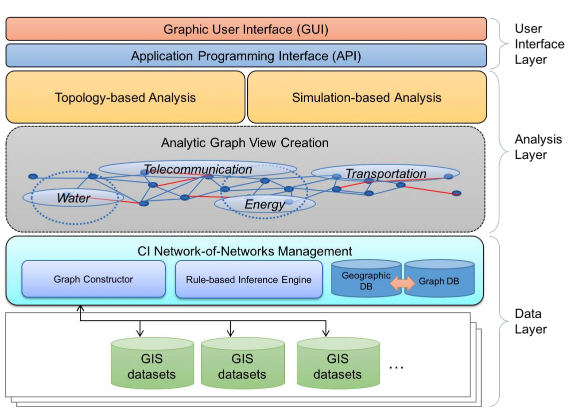

Figure 1: Conceptual architecture of an Infrastructure Vulnerability Analysis System (IVAS).

3.1 The data layer– Constructing and managing a CI network-of-networks graph

The *data layer* is the lowest layer of the architecture. There exists a wide range of public and non-public CI datasets compiled by various organizations such as federal agencies or commercial vendors. For instance, the United State's National Geospatial-Intelligence Agency (NGA) and the Department of Homeland Security (DHS) published a unified infrastructure geospatial data inventory, namely HSIP Gold [1], which includes domestic infrastructure datasets collected from various government agencies and partners. A license free subset of HSIP Gold called HSIP Freedom is available as well. NHDPlus [4] is a dataset created by the US Environmental Protection Agency (EPA), which includes information about the nation's hydrological framework. USGS Current Water Data for the Nation [7] provides real-time stream flow data across the nation. Other examples include open-source Energy datasets from U.S. Energy Information Administration (EIA) [6].

The *CI Network-of-Networks Management* component is responsible for constructing and managing a CI network-of-networks graph from various GIS datasets. The graph is composed of CI entities and their relationships extracted from multiple data sources, where the relationships include both intra relationships within a dataset and inter relationships between entities across multiple datasets. For the graph construction process, higher-level abstracted models are desired. In other words, the *nodes* (URIs), *edges* (predicates), and properties (subset of predicates and literals describing detailed information about nodes and edges) in the CI network-of-networks graph need to be clarified. There can be many ways of constructing the graph and we illustrate a process in Figure 2 that includes the following steps.

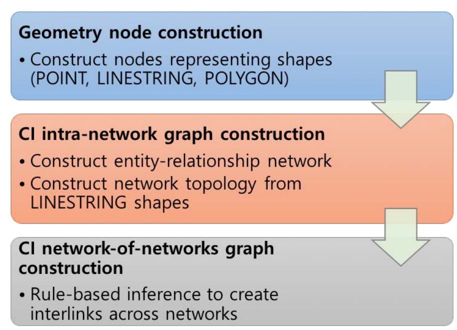

Figure 2: Process of constructing a CI network-of-networks graph.

Geometry node construction: The GIS datasets generally include geographical shapes of entities (e.g., POINT, LINESTRING, POLYGON, etc.). As the first step, the *graph constructor* component extracts shapes information from data sources and construct geometry (shapes). We notice that majority of GIS datasets are published in the *shapefile* format [20]. However, in a generic system, understanding various formats such as plain text and Comma Separated Values (CSV) should also be considered. The following example shows how geometrical shapes can be modeled as a set of triples in the pseudo N-triple format, where P_a and P_b can be considered as nodes and the triples describe the properties of the nodes.

```
P_a "hasType" "Point".
```

```
P_a "hasLatitude" "XX". P_a "hasLongitude" "YY".
...
G_a "hasType" "Poligon".
G_a "isComposedOf" P_a. G_a "isComposedOf" P_b.
G_a "isComposedOf" P_c. G_a "isComposedOf" P_d.
// G_a is a square-shaped geometry,
// composed of four points
```

CI intra-network graph construction: The *graph constructor* component constructs CI intra-network graphs by interpreting given GIS datasets. A CI intra-network graph refers to a graph that can be directly extracted from a dataset describing a CI. GIS datasets with information of CI entities and their explicit relationships can be naturally converted into a CI intra-network graph by making each entity and relationship to a node and an edge respectively. The following example shows an CI intra-network graph in RDF triples constructed from a power transmission network dataset. Note that every node representing an entity has a connection to a geometry node by "hasShapeOf" predicates to represent its shape.

```
//CI intra-network graph
S_a "hasType" "Entity". S_a "isA" "Substation".
S_a "hasShapeOf" P_a
...
S_b "hasType" "Entity". S_b "isA" "Substation".
S_b "hasShapeOf" P_b
...
S_a "hasTransmissionLine" S_b.
```

A CI intra-network graph can be also constructed from multiple *LineString* shapes in a data set. For example, we can create a road network from multiple *LineString* shapes representing the shapes of streets. In this case, nodes representing intersections may need to be created. The following example shows how a CI intra-network graph can be created from two *LineString* geometries. The directions of edges should be carefully considered, and this is particularly important for some infrastructure networks such as water streams where the direction (e.g., water flow) could affect the analysis.

```
//multiple linestrings
L_a "hasType" "LineString".
L_a "isComposedOf:L_a:1" P_a.
L_a "isComposedOf:L_a:2" P_b.
L_a "isComposedOf:L_a:3" P_c.
L_b "hasType" "LineString".
L_b "isComposedOf:L_b:1" P_d.
L_b "isComposedOf:L_b:2" P_b.
L_b "isComposedOf:L_b:3" P_e.
//CI intra-network graph
P_a "connectedBy:L_a" P_b. P_b "connectedBy:L_a" P_c.
P_d "connectedBy:L_b" P_b. P_b "connectedBy:L_b" P_e.
```

CI network-of-networks graph construction: At first, there will be no explicit interlinks between nodes created from disparate datasets. However, there can be various types of interdependencies (e.g., cyber, physical, logical, etc.) between the entities in different infrastructures. This layer is about interlinking nodes created from different CI datasets. The *graph constructor* component exploits the *rule-based inference engine* to infer relationships of entities in different infrastructures and then creates edges between the nodes in CI intra-network graphs. The *rule-based inference engine* should allow users (e.g., CI operators) to expressively define various rules. For instance, operators can choose to connect a substation node with all water pumping station nodes that are within the substation's service area polygon to represent physical dependency of power.

We emphasize that constructing and managing CI network-of-networks graph in a scalable and efficient manner is as important as the other layers, because it provides the fundamental for operations of IVAS. SBD tools cannot resolve all the challenges in this layer, but many existing systems and tools can be exploited to complement the SBD tools. Big Data data processing frameworks such as MapReduce [18] and Spark [42] can be exploited to deal with large-scale data conversion via parallel processing in a clustered system. Our prior work [29] is a good example where MapReduce can be utilized for constructing large-scale graph from non-graph datasets. A geographical database (e.g., PostGIS [33]) and a triplestore (e.g., uRiKA-GD) can be combinationally utilized to store and analyze CI network-of-networks graphs. For instance, we can store a geometry in a geographical database and assign an identifier to the stored geometry entity. Additionally, we can store a node (URI) with a reference to the identifier in the triplestore. Performing geographical operations without proper indexing techniques can be very expensive. For instance, if the rule-based inference engine wants to create new edges between two POINT geometry nodes if they are the *nearest*. Without proper indexing techniques, processing such simple rule already requires $N \times N$ times of measuring distance, where N denotes the number of *POINT* geometry nodes. Geographical operations such as *finding nearest points*, *determining overlaps*, and *measuring distance* can be efficiently performed utilizing the geographical database by taking advantages of indices (e.g., R-Tree-over-GiST [32], etc.). On the other hand, graph operations such as *graph traversals* can be also efficiently processed using triple store. Furthermore, utilizing a triplestore that supports GeoSPARQL [12], such as Parliament [13], can be also considered for utilization.

3.2 The analysis layer– Performing analytic operations at scale

There is not a "one-size-fits-all" graph for all kinds of analysis. Only the nodes and edges that CI operators are interested in should be considered for an analysis. The *analytic layer* is responsible for creating and managing various analytic graph *views* from the constructed graph. Users should be able to define their constraints of sub-selecting entities from the constructed RDF graph (e.g., select *only* highways from a road network dataset). The problem of creating graph analytic views are highly related to subgraph pattern matching query against the constructed graph. Luckily, there are many SBD tools that are developed to efficiently perform this operation. But, it should be considered that there needs to be a mechanism that can incorporate geographical information into the SPARQL query at the same time. Thus, an abstract data interface that covers both of geographical database and triplestores need to be provided for this purpose. Materialization of selected graph views for efficient processing should be considered.

Once a graph view is selected, the components in this layer are responsible for performing various analysis operations on the graph. In general, analytic operations can be grouped into two categories.

The *topology-based analysis* aims to discover useful vulnerability analysis results based on the understanding of the structure of a graph. Understanding how robust a CI graph

We will use the term 'graph' to refer to an analytic graph view the convenience of explanation, as a whole is can be useful, since it gives a big picture to CI operators. However, defining a good quantitative measure that captures the vulnerability of the system modelled as a graph is still an open research problem. We suggest a combinational usage of existing graph-theoretic measures such as *node degree*, *shortest paths*, connected component [36], node eccentricity [23] as starting points. For instance, by assuming that a higher number of shorter re-routing paths between nodes indicates a more robust network, the robustness can be measured by the size of the largest connected component and by the average distance between nodes [28].

In addition, CI operators are often interested in quantifying the vulnerability of graph components (e.g., nodes, edges, subgraphs, etc.), since it can provide more specific information to CI operators. There have been a few researches that incorporate simple graph-theoretic measures such as node degree or betweeness [21] for vulnerability analysis [28, 19], but not much efforts have been made to leverage more sophisticated measures such as PageRank [40].

For computation of these graph measures, existing tools such as gm-sparql [30], which enables performing various graph operations such as *node degree*, connected component [36], node eccentricity [23] within a triplestore, can be utilized. However, as majority of existing graph measures are defined on a homogeneous graph, which is composed of one type of nodes and edges and CI network-of-networks graphs are heterogeneous, existing graph-theoretic measures should be carefully utilized or modified with incorporating sufficient domain knowledge of CI.

The *simulation-based analysis* aims to understand how effects of perturbation events in an infrastructure (e.g., damage in the road network) can spread out across multiple infrastructure networks via simulation. One potential approach for the simulation-based approach is to leverage propagation models such as random-walks [40] and label propagation [43] to incorporate domain knowledge into the graph network. For instance, assuming the Markov property and each node in the network are affected by its adjacent nodes, cascading effects over a graph can be simulated by using a function that decides each node's status based on its adjacent nodes. It is important to also consider temporal features, such as predicting how failures of nodes can cause cascading effects throughout the network after an hour, a day, or a week. Another approach of performing simulation-based analysis is to use the vulnerability measures as discussed previously. Specifically, we can use the changes in the vulnerability score of the entire graph or certain interested graph components before and after removal of a given set of nodes in the graph. Simulation of various what-if scenarios such as random perturbations (inactivating random nodes or edges), targeted perturbations (inactivating nodes with higher number of edges), regional perturbations (inactivating a set of nodes or edges located in a region) can be useful for CI operators. Ultimately, simulation-based analysis should be able to predict consequences of a perturbation event before the effects cascade, which means that not only the scalability but also the processing time responding to changing inputs needs to be deeply investigated.

3.3 The user interface layer – Interactive visualization of analytic output to generate key insights

The user interface layer layer aims to provide an easy-to-use and intuitive interface of IVAS to users. IVAS should have similar capabilities provided by other GIS tools, such as visualization of analytic outputs of CIs as layered maps. It can give intuitive insights to the decision makers and operators by clarifying the relationships among several factors and displaying how seriously a perturbation event can affect a local area or the network as a whole [38]. Without having to develop the whole visualization interface, leveraging standards and widely exploited softwares like ESRI's ArcGIS [27], Maptitude [3] and Google Maps can be a good start. For interactive analysis, providing intuitive interface for defining of perturbation inputs (nodes/edge removal) and the type of interdependencies that will be considered for the graph analysis need to be studied. Presenting discovered knowledge or suggestions based on analysis as a combinational display of information visualization widgets such as line, bar charts, and tree maps [16] are desired by CI operators. Considering the extensibility of system, developing APIs that can be directly used in another software development should be also considered.

IVAS should allow users to easily ingest additional CIs data sets into the constructed heterogeneous infrastructure network, specify the types of interlinking, the type of required analysis such as topological and dynamic cascade analysis. It should also allow users to navigate and prioritize data entities and discovered knowledge.

4. CONCLUSION

In this study, we introduced an important application domain – critical infrastructure vulnerability analysis – for Semantic Big Data community. Critical infrastructures (CIs) which are significant to sustaining day-to-day commodity flows form a very complex heterogeneous network-of-networks. Modeling and performing simulation on these networks for vulnerability and cascading failure analysis are nicely aligned with the capabilities of emerging SBD technologies. However, to complement limitations of the SBD tools, synergistic exploitation of other various Big Data and GIS tools and standards should be carefully considered. We also discussed various types of challenges we might potentially encounter while constructing and managing a CI network-of-networks graph, performing analytic operations at scale, and visualizing analytic output to generate meaningful insights.

5. REFERENCES

[1] Homeland security infrastructure program (HSIP). https://gii.dhs.gov/HIFLD/hsip-guest.

[2] Linked Open Data statistics. http://stats.lod2.eu/.

[3] Maptitude: Geographic Information System. http://www.caliper.com/maptovu.htm.

[4] NHD plus. http://www.horizon-systems.com/nhdplus/.

[5] ORIGAMI:Oak Ridge Graph Analytics for Medical Innovation. http://hypothesis.ornl.gov/.

[6] U.S. energy information administration (EIA). https://www.eia.gov/.

[7] USGS Current Water Data for the Nation. http://waterdata.usgs.gov/nwis/rt.

[8] A. Aji, X. Sun, H. Vo, Q. Liu, R. Lee, X. Zhang, J. Saltz, and F. Wang. Demonstration of hadoop-gis: a spatial data warehousing system over mapreduce. In *Proceedings of the 21st ACM SIGSPATIAL*

International Conference on Advances in Geographic Information Systems, pages 528–531. ACM, 2013.

[9] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis, and K. Tolle. The ics-forth rdfsuite: Managing voluminous rdf description bases. In *SemWeb*, 2001.

[10] Apache Software Foundation. Apache Jena. https://jena.apache.org/.

[11] A. M. Barker, E. B. Freer, O. A. Omitaomu, S. J. Fernandez, S. Chinthavali, and J. B. Kodysh. Automating natural disaster impact analysis: An open resource to visually estimate a hurricane's impact on the electric grid. In *Southeastcon*, pages 1–3, 2013.

[12] R. Battle and D. Kolas. Geosparql: enabling a geospatial semantic web. *Semantic Web Journal*, 3(4):355–370, 2011.

[13] R. Battle and D. Kolas. Enabling the geospatial semantic web with parliament and geosparql. *Semantic Web*, 3(4):355–370, 2012.

[14] J. Broekstra, A. Kampman, and F. Van Harmelen. Sesame: A generic architecture for storing and querying rdf and rdf schema. In *The Semantic Web-ISWC*, pages 54–68. Springer, 2002.

[15] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin. Catastrophic cascade of failures in interdependent networks. *Nature*, 464(7291):1025–1028, 2010.

[16] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann, 1999.

[17] Q. Chen, L. Wang, and Z. Shang. Mrgis: A mapreduce-enabled high performance workflow system for gis. In *eScience, 2008. eScience'08. IEEE Fourth International Conference on*, pages 646–651. IEEE, 2008.

[18] J. Dean and S. Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[19] L. A. Dueñas-Osorio. *Interdependent response of networked systems to natural hazards and intentional disruptions*. PhD thesis, Citeseer, 2005.

[20] E. ESRI. Shapefile technical description. *An ESRI White Paper*, 1998.

[21] L. C. Freeman, S. P. Borgatti, and D. R. White. Centrality in valued graphs: A measure of betweenness based on network flow. *Social networks*, 13(2):141–154, 1991.

[22] S. Gurajada, S. Seufert, I. Miliaraki, and M. Theobald. Triad: A distributed shared-nothing rdf engine based on asynchronous message passing. In *SIGMOD*, 2014.

[23] P. Hage and F. Harary. Eccentricity and centrality in networks. *Social networks*, 17(1):57–63, 1995.

[24] S. Harris, N. Lamb, and N. Shadbolt. 4store: The design and implementation of a clustered rdf store. In *SSWS2009*, pages 94–109, 2009.

[25] R. Henschel, A. Seal, J. J. Yang, D. J. Wild, Y. Ding, A. Thota, S. Michael, M. Gianni, and J. Maltby. Applications of the yarcdata urika in drug discovery and healthcare. 2014.

[26] Å. J. Holmgren. Using graph models to analyze the vulnerability of electric power networks. *Risk analysis*, 26(4):955–969, 2006.

[27] K. Johnston, J. M. Ver Hoef, K. Krivoruchko, and N. Lucas. *Using ArcGIS geostatistical analyst*, volume 380. Esri Redlands, 2001.

[28] V. Latora and M. Marchiori. Vulnerability and protection of infrastructure networks. *Physical Review E*, 71(1):015103, 2005.

[29] S. Lee, S.-H. Lim, and B.-H. Park. Table2graph: A scalable graph construction from relational tables using map-reduce. *BigDataService*, 2015.

[30] S. Lee, S. R. Sukumar, S. Hong, and S.-H. Lim. Enabling graph mining in rdf triplestores using sparql for holistic in-situ graph analysis. *Expert Systems with Applications*, 48:9–25, 2016.

[31] Z. Liu, A. Le Calvé, F. Cretton, and N. Glassey. Using semantic web technologies in heterogeneous distributed database system: A case study for managing energy data on mobile devices. *IJNCAA*, 4(2):56–69, 2014.

[32] T. Nguyen. Indexing postgis databases and spatial query performance evaluations. *International Journal of Geoinformatics*, 5(3), 2009.

[33] R. Obe and L. Hsu. *PostGIS in action*. Manning Publications Co., 2011.

[34] M. Ouyang. Review on modeling and simulation of interdependent critical infrastructure systems. *Reliability engineering & System safety*, 121:43–60, 2014.

[35] P. Pederson, D. Dudenhoeffer, S. Hartley, and M. Permann. Critical infrastructure interdependency modeling: a survey of us and international research. *Idaho National Laboratory*, pages 1–20, 2006.

[36] L. Qin, J. X. Yu, L. Chang, H. Cheng, C. Zhang, and X. Lin. Scalable big graph processing in mapreduce. In *SIGMOD*, 2014.

[37] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly. Identifying, understanding, and analyzing critical infrastructure interdependencies. *Control Systems, IEEE*, 21(6):11–25, 2001.

[38] M. Shankar, J. Stovall, A. Sorokine, B. Bhaduri, and T. King Jr. Visualizing energy resources dynamically on earth. In *Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–4. IEEE, 2008.

[39] R. Singh, J. A. Graves, S. Lee, S. R. Sukumar, and M. Shankar. Enabling graph appliance for genome assembly. In *Big Data*, pages 2583–2590. IEEE, 2015.

[40] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. *ICDM*, 2006.

[41] J. Winkler, L. Dueñas-Osorio, R. Stein, and D. Subramanian. Interface network models for complex urban infrastructure systems. *Journal of Infrastructure Systems*, 2011.

[42] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: cluster computing with working sets. In *HotCloud*, 2010.

[43] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Citeseer, 2002.