eXsight: An Analytical Framework for Quantifying Financial Loss in the Aftermath of Catastrophic Events

Matthew Coelho and Andrew Rau-Chaplin
Faculty of Computer Science
Dalhousie University
Halifax, NS
Email: {coelho, arc}@cs.dal.ca

***Abstract*—Property catastrophe insurance and reinsurance companies are financial institutions that provide for the equitable transfer of the risk due to catastrophic events such as earthquakes, hurricanes, and floods, in exchange for a premium. Immediately after a catastrophic event these insurers face an acute situational analysis and management challenge. They want to immediately start to flow funds to their affected clients, they need fast estimates of likely losses so they can reserve the necessary capital, and they need to be able to communicate their potentially changed financial situations to regulators, rating agencies, and stockholders.**

In this paper we explore the design of an analytical framework for quantifying financial loss in the aftermath of catastrophic events. The idea is to aggregate the thousands of Exposure databases received by a single reinsurer into a giant loosely structured exposure portfolio and then to use Big Data analysis technology, originally developed in the context of web-scale analytics, to rapidly perform natural but ad-hoc loss analysis immediately after an event. As in many situational analysis problems, the challenge here is to work with both categorical and geospatial data, deal with partial data often at varying levels of aggregation, integrate data from many sources, and provide an analysis framework in which natural but ad-hoc analysis can be rapidly performed in the hours, days, and weeks immediately after an event.

***Keywords*—Risk analytics, framework, exposure data, post-event, MongoDB, catastrophe, reinsurance.**

I. Introduction

In 2013 two-hundred and ninety-six global natural disasters (including earthquakes, floods, and hurricanes) caused a total economic loss of $192 billion dollars (USD) [1]. Of this total economic loss, $45 billion dollars was insured meaning that the financial resources required to help effect rapid recovery were available and being held in reserve by the insurance and reinsurance companies who had underwritten the risk. In the hours, days, and weeks immediately after the event these insurers face an acute situational analysis and management challenge. They want to immediately start to flow funds to their affected clients, they need fast estimates of likely losses so they can reserve the necessary capital, and they need to be able to communicate their potentially changed financial situations to regulators, rating agencies, and stockholders.

However, the situation on the ground after a natural disasters is often unclear. The extent of the affected area, the intensity of the hazard, and its impact on the value and usability of buildings are all unknown. Eventually, when initial claims have been filed, the buildings repaired or replaced, and the insurance for lost use or business interruption covered, the total loss will be know. Until then, insurers and reinsurers need systems to help them estimate losses so that they can quickly get initial funds to the right clients in the right amounts and communicate their financial situation to counter parties, regulators, and the public.

Exposure data, in the context of property catastrophe insurance, refers to data that describes what is being insured and under what terms. Broadly it consists of three types of data: 1) Location data such as latitute/longitude, street address, postal code, state, province, country, etc. 2) Physical data such as building type, construction, number of stories, roof type, age, etc. and 3) Contractual data such as coverage value, limits, deductibles, and other financial terms defining the risk transfer contract.

Primary insurance companies collect exposure data from their clients (home owners and businesses), place it in Exposure databases, and pass it on to reinsurance companies. Thousands of these Exposure databases are collected by reinsurance companies (from their clients - the primary insurers) and are used in the pricing process. The reinsurers typically take each individual database and run it through a pricing model to produce an expected loss table (ELT) and then archive it. The individual Exposure databases are currently considered to be too big and granular to be of much further use in the reinsurer's analytical pipeline.

In this paper we explore the design of an analytical framework for quantifying financial loss in the aftermath of catastrophic events. The idea is to aggregate the thousands of Exposure databases received by a single reinsurer into a giant, loosely structured exposure portfolio, and then to use Big Data analysis technology, originally developed in the context of web-scale analytics, to rapidly evaluate natural, but ad-hoc, loss analysis immediately after an event. As in many situational analysis problems, the challenge here is to work with both categorical and geospatial data, deal with partial data often at varying levels of aggregation, integrate data from many sources, and provide an analysis framework that in which natural but ad-hoc analysis can be rapidly performed in the hours, days, and weeks immediately following an event.

II. Scenarios

In this section to help illustrate the challenges of post-event analytics and to build up a set of use cases and a definition of core framework operations, we explore scenarios built around three recent events. While governments, non-governmental agencies, insurers, and reinsurers are all involved in post-event exposure analysis, to keep our scope manageable we will focus on analysis from a reinsurer's perspective.

A. Tōhuku Earthquake, Tsunami, and Radiation Disaster

On March 11, 2011 a magnitude 9.0 earthquake occured just off of the coast of Japan. The earthquake also caused a subsequent tsunami whose total run-up height measured 38.9 meters, approximately the size of a 12 story building. The combination of the earthquake and tsunami severely damaged a number of the reactors in the Fukushima I Daiichi nuclear power plant which caused a nuclear incident, leaking radiation into the surrounding environment. This earthquake was the fourth largest earthquake in recorded history, the largest in Japan[1], and resulted in 15,854 deaths and 3,203 missing persons[2] in Japan [2].

After a catastrophic event, reinsurance companies need to calculate loss information and provide it to regulation agencies, stockholders, and other parties. To do this they need to 1) determine the boundary of the region impacted, 2) determine a map of hazard intensities within that region, 3) estimate mean damage ratios (MDR) maps for the impacted by building type, and finally estimate losses taking into account financial terms under a variety of assumptions.

For an event like the Tōhuku event, reinsurers would overlay a variety of hazard maps to generate the affected area. They would gather shakemaps for earthquake intensity, inundation maps for tsunami intensity, and wind-borne debris maps for the radiation fallout. Overlaying these maps provides a picture of the affected event area and hazard intensities sustained within it.

After constructing the boundary of the affected region, reinsurers would then need to identify the impacted exposure. In areas like the US, where detailed high quality exposure data is typically available, this might be as simple as performing a geospatial query in the Exposure portfolio to identify locations with lat/longs within the boundary. In a case like the Tōhuku event which involves Japanese exposure, the specific exposure would likely be unknown. In this case, aggregated exposure collected at a district or even prefecture level would need to be spatially disaggregated using data such as daytime or nighttime population numbers to produce detailed representative lat/long based exposure.

Finally, by combining the identified exposure, the MDR maps of for the event, vulnerability curves by exposure type, and a financial terms simulator, an ad-hoc event specific loss model can be constructed and used. This will provide loss estimate summaries, a breakdown of losses by a set of filterable fields, and a mapping of losses over the area. The reinsurer can then use this information to drive reserving and early loss settlement processes, as well as to provide updated solvency information to regulators, rating agencies, and other interested parties.

[1]Since instrumental recordings began in 1900

[2]As of March 8, 2012

B. 2011 Thailand Flood Disaster

While the Tōhuku Earthquake was an event that rapidly unfolded, the 2011 Thailand Floods is an example of a slow event that unfolds over months in which the issue is unknown risk hidden in the exposure data.

In 2011 heavy rains throughout Thailand, the remnants of tropical depressions Haima and Nock-Ten, and an active monsoon season caused severe flooding across the country [3]. Although Thailand has a history of flooding, this was the most expensive event resulting in approximately $45.7 Billion USD in economic losses. The biggest contributor to these losses was the manufacturing sector, contributing approximately $32 billion [4]. Many companies in Thailand's manufacturing industry are hard drive manufacturers. The impact of flooding was so extreme that it interrupted the global supply chain of hard drives, driving up world prices substantially [5].

Before 2011, the average reinsurer would have told you that they had little financial exposure to Thailand floods. Thailand's manufacturers were largely insured by Japanese primary insurers who provided reinsurers with only aggregated exposure data. It was only when the early claims started trickling in that the reinsurers realized they might have a problem. But how much of a problem? How big were the eventual claims likely to be? Answering such questions required an ad-hoc analysis process that combined aggregate exposure data with publicly available industry and economic data.

The first task was to estimate the commercial exposure, and generate a detailed representative exposure set. Given knowledge of the aggregate exposure and the average value of a disk manufacturing facility, an estimate on the number of facilities and their values could be obtained. Then using a description of the transportation network and estimates of daytime population (as a proxy for the spatial distribution of commercial activity) detailed representative exposure sets could be generated.

The second task was to create, from the early claims data, industrial building vulnerability curves, and flood inundation maps, a crude aggregate loss model. Such models could be used for reserving (i.e. the process of reserving capital to pay-out future claims). In addition, based on the detailed exposure data, spatial accumulation modeling could be performed to identify potential loss hotspots. Spatial accumulation modeling identifies the largest exposure accumulations within circles of a given radius. This helps companies model circles of maximum potential loss, which is particularly important if the event may spread.

C. Hurricane Sandy

On the evening of October 29, 2012 Hurricane Sandy, a post-tropical cyclone, made landfall near Brigantine, New Jersey. Although a minimal hurricane as measured by the Saffir-Simpson scale, the storm covered a massive area and caused high storm surge over large parts of the coastline. The highest inundations were located in New York, New Jersey, and Connecticut, with the above ground storm surge ranging

between 2-9 ft with an average of 4.5 ft in New York [6]. Although this was a low intensity event, the huge size of the affected area and the high value of the exposure in the affected region caused damages of approximately $50 billion USD, resulting in Sandy being the second most expensive hurricane in US history [7].

In the case of Hurricane Sandy, post event analysis was greatly helped by the rich and detailed nature of the available exposure data. When you know exactly what is being impacted (ie. the exposure) you can concentrate your analysis on getting a more detailed and closer to real-time view of the evolving event. One interesting opportunity that became apparent during Hurricane Sandy was the potential for building real-time event intensity and impact maps from information gleaned from social media data. Social media is a new form of data we can look at for pre, peri, and post-event analysis. The idea is to supplement physical intensity measures with observed intensity measures using Geotagged tweets, Instagrams, and Facebook posts as data sources. The goal is to try and build up new high quality, real-time hazard maps from on-the-ground observations. To do this we can use time-based analysis to measure the intensity of the tides and the landfall, and also use text analytics to help build these new observed hazard maps. While this is a much more speculative form of post-event analysis than those previously discussed, in an increasingly networked world it has significant potential to provide detailed real-time data for post-event situation analysis and management.

III. The Exposure Analysis Framework

In this section we describe our approach to designing of a framework that can make better use of existing exposure data to provide new ways of storing and analyzing this data. The framework provides a way of storing many types data, operating on this data, and analyzing the results from these operations all on a high-performance platform that can scale to handle very large exposure portfolios.

A. The Data Life Cycle

Currently, exposure data is often only used as input into pricing models. After being used for pricing, the exposure data is not used again. This process is shown in Figure 1.

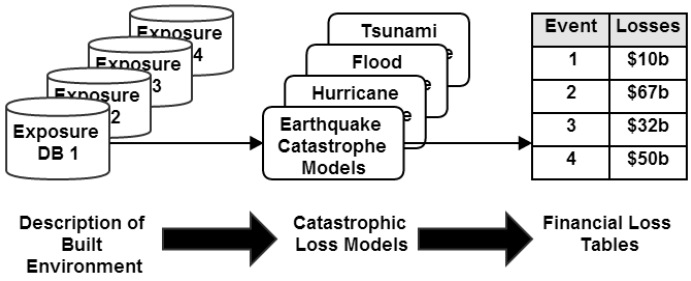

Fig. 1. The current data life cycle for exposure data.

With the Exposure Analysis Framework we hope to change this life cycle and ensure that we can tap into this rich data source. To do this we still use the exposure data for pricing, but after generation we move the data into a data warehouse where we can perform analytical operations on it. Not only can we import exposure data, but we can also store client, claims, event, and historical data inside this warehouse. By importing this other data we can perform many new types of cross analysis, and visualize an exposure portfolio and risk in ways that were not possible previously. Figure 2 shows the proposed data life cycle.

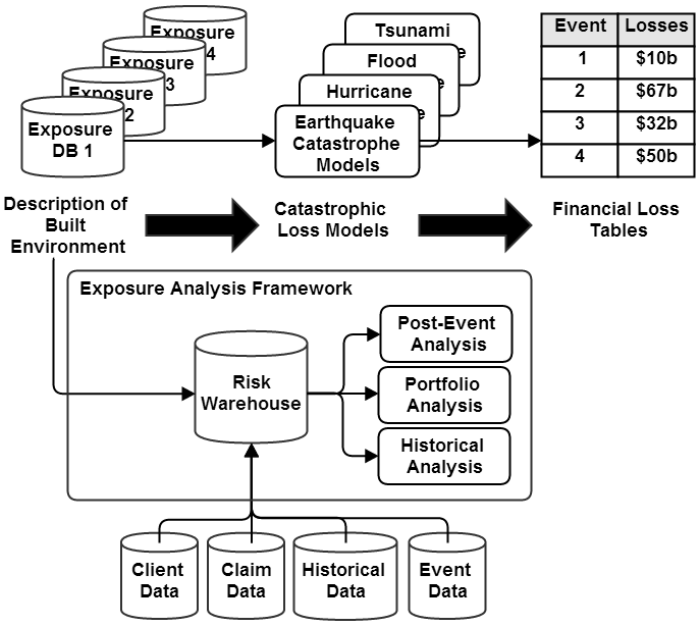

Fig. 2. The proposed data life cycle for exposure data using the framework.

B. Operations

The analytical framework is designed around five fundamental types of operations, namely: 1) Aggregation, 2) Disaggregation, 3) Geospatial, 4) Loss Modeling, and 5) Spatial Accumulation.

1) Aggregation Operations: The framework supports both standard and specialized aggregation operations. All aggregation operations group the data by some specified key or compounded key, then perform some meaningful calculations on the data in order to reduce it to some generalized data value. Typical aggregation operations include minimum, maximum, average, total values, element count, difference, and percent difference operations. Specialized aggregation operations perform actuarial and insurance-specific financial calculations.

2) Disaggregation Operations: Disaggregation operations take aggregated industry data and transform it into meaningful, finer-detailed exposure data. These operations are particularly useful when you have aggregated data and want to run it through a model. Since aggregated data is a coarse grained representation of data, and models require data with finer details, desegregation is used to transform this coarse data into finer detail. The disaggregation process is shown in Figure 3.

The idea for the disaggregation is to transform this high-level, coarse data into some reasonable finer granularity, ending up into geographical point-based exposure data. Disaggregation operations can transform data by in a variety of ways by first dividing it and distributing the exposure data randomly within a smaller region, or distributing the data based on population density.

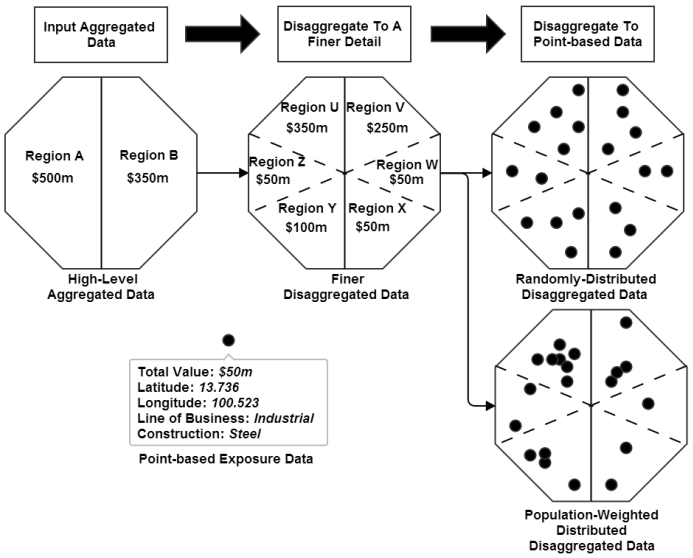

Fig. 3. A graphical representation of how a disaggregation operation works.

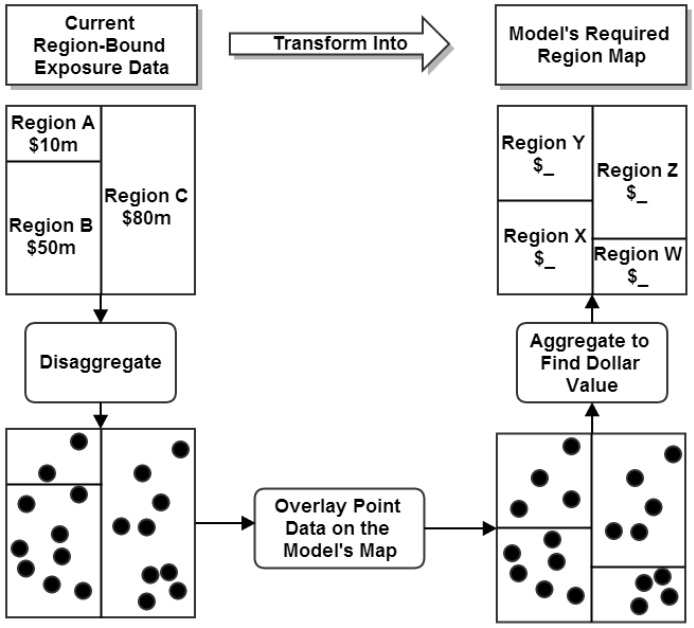

Fig. 4. A graphical representation of how a boundary transformation operation works.

3) Geospatial Operations: Geospatial operations provide a set of tools that can execute geographical queries and handle region data. With these operations you can perform point location queries in polygonal subdivisions, geocoding operations, and geometric operations on points and polygons. These operations also provide regional comparison tools. For example, these tools provide a way to "redistrict" the regional boundaries of older, aggregated exposure data into newer regions in order for the data to be correctly represented in the new model. This process is shown in Figure 4. This is done by disaggregating the data and turning it into point-based exposure data, and placing the new data points onto the old regional boundary map, either by assuming an even spread of points, or by weighting the point data based on population. Lastly the new boundaries are placed on top of the points and the regions are aggregated to reflect the new boundary changes.

4) Loss Model Operations: Post-event loss modeling operations come in two flavors: single-event footprint loss models, and aggregate loss models.

Footprint loss models are used to compute the losses for a given single event. Event data is collected from a variety of hazard maps that provide a collection of areas affected by the event and the intensity in each of the areas. Using these hazard maps, a mean damage ratio (MDR) is calculated for each of the event regions. Exposure data is then used and overlaid on top of the event regions and total losses are calculated by taking the total value of the exposure in the region and multiplying it by the MDR to find the total losses in the area. The result is a table of losses that can be filtered using a variety of specified keys.

Aggregate loss models are similar to footprint models, except instead of one event you have multiple events. The process still follows the same structure as the footprint model, but instead of having a single hazard map, multiple hazard maps are used for each of the events. These models are used to illustrate the total combined losses of events and can also be filtered using a variety of specified keys.

5) Spatial Accumulation Operations: Spatial accumulation operations provide a method for identify regions of largest risk. These operations allow you to find a list of the largest non-overlapping risk or exposure concentrations within query circles of a given radius. Such circles represent exposure or risk accumulations and can provide companies with insight into the spatial distribution and/or clustering of their risks.

IV. The eXSight Framework - v0.1

Our current implementation of the Exposure Analysis Framework, called the eXsight Framework, is built on the Java platform and a MongoDB backend. MongoDB provides a robust suite of NoSQL operations which manage data handling and calculations for the framework.

A. Components

All of the framework's operations are Java classes that utilize the MongoDB Java driver [8]. This driver is responsible for communicating with a MongoDB cluster to manage data, execute operations, and handle results.

MongoDB is a versatile and scalable NoSQL database system that not only provides a way to store and manage data, but also provides a suite of tools to efficiently operate on it [9]. Unlike regular SQL servers, where data is stored in tables and rows, and relationships are drawn between them, MongoDB stores data as a document that contains various fields represented by a JSON-like structure, and multiple documents are contained within a collection [10]. Because of these differences in structure, traditional SQL tools and techniques will not work, and thus MongoDB provides a suite of robust tools and query engines that provides the same functionality that works with this new structure.

The eXsight framework utilizes MongoDB's geospatial indexes and query engine [11], Aggregation Pipeline [12], and

MapReduce Framework [13] to perform the various operations, while also utilizing the built-in NoSQL database functionality to handle data management.

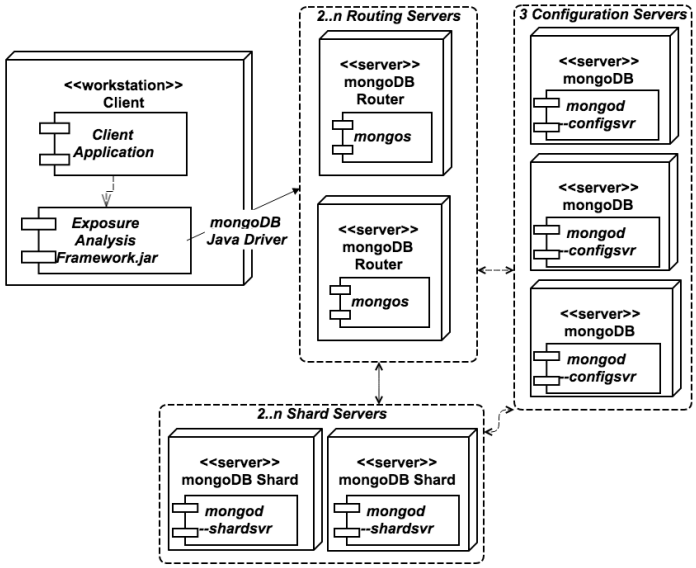

Fig. 5. Deployment diagram depicting the recommended hardware architecture for the framework.

MongoDB can easily scale vertically by adding in more servers and horizontally with the use of sharding. Sharding is a form of horizontal scaling which allows data to be stored across multiple machines. As you can see in Figure 5, a sharded MongoDB backend consists of multiple servers. A sharded production database consists of two or more routing servers, exactly three configuration servers, and two or more shard servers.

1) Query Routing Servers: The routing servers run instances of `mongos`. These instances are used to pass queries and operations on to the correct shards and then pass the results to the client application. In most cases you will have multiple `mongos` instances in order to ease the query load.

2) Configuration Servers: The configuration servers run special instances of `mongod`. These instances are started with the `--configsvr` flag and are responsible for storing the cluster's metadata. This metadata keeps track of a map of the entire data set, and knows which shard contains what data. This mapping is used to direct operations and queries from the routing servers to the appropriate shards.

3) Shards: The final part of the cluster is the shard servers. The shard servers, similar to the configuration servers, run a special instance of `mongod` except it uses the `--shardsvr` flag. The shard servers are used to store a subset of the total data based on the type of data partition that is in use. MongoDB shards the data at the collection level based on a specified *shard key* which exists in every document in the collection [14].

Sharding is a crucial part of improving the performance and efficiency of the framework, however the current prototype framework only implements a single MongoDB backend instance. We will be working on extending on this in a future release.

To use the framework you have to create a Java application which calls the framework's API methods in order to import data, perform operations, and handle results. Figure **??** shows the control flow for the current framework prototype. As you can see, the user can import data into the MongoDB backend and then perform a selection of aggregation and geospatial operations by simply following this diagram and calling the appropriate methods.

B. Data Handling

The framework implements a generic data importer that lets the user define the overall schema of the data. Figure **??** shows the various packages contained in the framework. Notably the `io` package contains the classes and sub-packages that are responsible for handling data input and output for the framework.

The importer provides a way of ingesting exposure data without knowing its structure. To do this the user must describe their data using classes in the `io.utils.importer` package. The importer class takes as a parameter a configuration object, which is an instance of the class that defines the data schema. This schema has a list of data sources that can include one or more files, directories, database tables, models, or NoSQL collections. Inside of each source you would describe the data that requires importation. You would tell the framework where the data is, the type of the data, and the name of the data. This is done for each data element in order to build up a data schema that the importer can read to understand how to import the data.

The `io` package also contains data exporter classes. These two classes are used in conjunction with the operations, mentioned in the next section, to provide a way of presenting the user with different methods of handling and consuming results from executed operations. The exporters allow the user to retrieve a string or list of the results, export data to a MongoDB collection, or even return the raw operation output.

C. Operations

In this section we describe how the core aggregation and geospatial operations are implemented to using MongoDB's Aggregation Pipeline and MapReduce Framework. We hope to eventually provide single implementation of these core operations once the size limitations on MongoDB document is lifted as is proposed in the MongoDB development roadmap.

1) Aggregation Operations: Figure **??** shows the types of aggregation operations you can perform on the exposure data. These operations offer two different ways of executing the operation, using MongoDB's Aggregation Pipeline or using MongoDB's MapReduce Framework.

MongoDB's Aggregation Pipeline is used to perform simpler aggregation functions, bypassing the complexity of the MapReduce framework. The pipeline works in a similar fashion to the UNIX pipe operator. The entire collection is passed through multiple operators to eventually reduce the collection to a single document[3] containing the results [12].

[3]Because the result of the operation is a document, the size of the returned result must be no more than 16MB, which is based on the limitations of a MongoDB document. If you need to get results that are larger than this, you must use the MapReduce Framework.

Similar to the Aggregation Pipeline, MongoDB's MapReduce Framework is also used for aggregation operations. This framework provides the user with an expressive set of methods that allow for more complex operations than the pipeline. Each MapReduce operation has a set of user-defined JavaScript functions: a `map` function, a `reduce` function, and a `finalize` function[4]. The `map` function is the first step in the map-reduce process. It matches a value to a certain key and returns a key-value pair. After mapping, the `reduce` function takes the emitted key-value pairs and condenses all of the values for a given key to a single value. Lastly the `finalize` function takes the reduced key-value pair and edits the output into its final form [15] .

Both aggregation method types are different and have their separate pros and cons, however both will return the same values for the same input.

2) Geospatial Operations: The geospatial package in Figure **??** provides a set of classes that can execute geographical queries, and populate or correct administrative region data.

The `RegionQuery` class provides methods to query the administrative regions that contain a given point. For example, when given a lat/long coordinate the methods in this class will return the names of all of the administrative regional boundaries of the point. The point [28.418749, -81.581211], if passed to the class methods, will return United States, Florida, and Orlando County as the country, state, and county of the point.

The `Geocode` class uses the `RegionQuery` class however instead of querying a single point, it looks at the current exposure data and populates the regional data based on the point data inside of the exposure collection. It provides methods to populate missing data, inaccurate data, and overwrite data.

V. Performance

MongoDB was the chosen backend for this project, not only because of its flexible data storage system, but also because MongoDB is very efficient and fast. Because the project is in the early stages, the current prototype is running on a single machine, thus all of these tests were performed on a single running instance of MongoDB and not a sharded cluster.

A. Aggregation Pipeline vs MapReduce Framework

For testing the difference in performance of these two aggregation methods, we used a simple count aggregation operation. This operation ran through 10001082 documents and counted the total number of elements that were in each county. The Aggregation Pipeline took an average of 15.4 seconds to complete, while the MapReduce framework took an average of 106.3 seconds to complete.

As you can see MongoDB's Aggregation Pipeline executes about 7x faster than the MapReduce framework, however these tests were run on a single MongoDB back end. Future tests will be performed on future versions of the framework which utilize a sharded cluster to determine if the MapReduce framework can be improved to perform as well as the Aggregation Pipeline.

[4]The map and reduce functions are required, the finalize function is optional.

VI. Future Work

We are currently working to refine the design of and complete the implementation of the remaining core operations. Planned future work on the eXsight framework includes implementing the missing operations and performance tuning of the current operations. Since our goal is to have the framework operating on a sharded MongoDB cluster, we will also be migrating from a single MongoDB instance to a multiple MongoDB backend.

We also plan a trial with industry partners to explore the practical applicability of the framework and to identify any critical new operations or functionality, input formats that need to be accepted, or output formats and data visualization options that need to be provided.

References

[1] AON Benfield, Annual Global Climate and Catastrophe Report, Impact Forecasting 2013. [Online]. Available: http://thoughtleadership.aonbenfield.com/Documents/20140113_ab_if_annual_climat [Accessed: 13 Apr. 2014].

[2] P. Dunbar et al., "Tohoku Earthquake And Tsunami Data Available From The National Oceanic And Atmospheric Administration/National Geophysical Data Center," Geomatics, Natural Hazards and Risk, vol. 2, no. 4, Nov. 2011, pp. 305-323. [Online]. Available: Taylor and Francis Group2, doi: 10.1080/19475705.2011.632443 [Accessed: 30 Mar. 2014].

[3] Thai Meteorological Department, "Rainfall and severe flooding over Thailand in 2011," Nov. 2, 2011. [Online]. Available: http://www.tmd.go.th/en/event/flood_in_2011.pdf. [Last accessed: 31 Mar. 2014].

[4] "The World Bank Supports Thailand's Post-Floods Recovery Effort," The World Bank, Dec. 13, 2011. [Online]. Available: http://www.worldbank.org/en/news/feature/2011/12/13/world-bank-supports-thailands-post-floods-recovery-effort. [Last accessed: 31 Mar. 2014].

[5] "Hard Drive Prices Rise Due To Thai Floods," InformationWeek, Jan. 9, 2012. [Online]. Available: http://www.informationweek.com/data-protection/hard-drive-prices-rise-due-to-thai-floods/d/d-id/1102133. [Last accessed: 31 Mar. 2014].

[6] E. Blake et al., "Tropical Cyclone Report Hurricane Sandy (AL182012) 22 29 October 2012," Feb. 12, 2013. [Online]. Available: http://www.nhc.noaa.gov/data/tcr/AL182012_Sandy.pdf. [Last accessed: 31 Mar. 2014].

[7] D. Porter, "Hurricane Sandy Was Second-Costliest In U.S. History, Report Shows," Feb. 12, 2013. [Online]. Available: http://www.huffingtonpost.com/2013/02/12/hurricane-sandy-second-costliest_n_2669686.html. [Last accessed: 31 Mar. 2014].

[8] MongoDB Java Language Center website: http://docs.mongodb.org/ecosystem/drivers/java/ [Last accessed: 25 Mar. 2014].

[9] MongoDB Overview website: https://www.mongodb.com/mongodb-overview [Last accessed: 25 Mar. 2014].

[10] MongoDB Introduction Documentation website: http://docs.mongodb.org/manual/core/introduction/ [Last accessed: 28 Mar. 2014].

[11] MongoDB Geospatial Indexes and Queries Documentation website: http://docs.mongodb.org/manual/applications/geospatial-indexes/ [Last accessed: 26 Mar. 2014].

[12] MongoDB Aggregation Pipeline Documentation website: http://docs.mongodb.org/manual/core/aggregation-pipeline/ [Last accessed: 26 Mar. 2014].

[13] MongoDB MapReduce Framework Documentation website: http://docs.mongodb.org/manual/core/map-reduce/ [Last accessed: 26 Mar. 2014].

[14] MongoDB Sharding Documentation website: http://docs.mongodb.org/manual/sharding/ [Last accessed: 27 Mar. 2014].

[15] MongoDB mapReduce Command Reference website: http://docs.mongodb.org/manual/reference/command/mapReduce/ [Last accessed: 30 Mar. 2014].

[16] RMS(one) Information website: http://www.rms.com/rms-one/rms-one#cloud [Last accessed: 1 April 2014].

[17] "RMS Revolutionizes Risk Management for Insurance Industry with Secure Platform Built on MongoDB," MongoDB, January 15, 2014. [Online]. [Last accessed: 1 April 2014].