Creating a Scalable Database for Weather Research

Seth Cook and Stephen Harrell

Research Computing, Purdue University

Introduction

Atmospheric scientists store some meteorological data in the GRIB (GRIdded Binary)[4] format, which "is an efficient vehicle for transmitting large volumes of gridded data"[4]. Data sets stored in the GRIB format may be split across multiple files, each file containing compressed, binary data, which adds complexity to querying meteorological data across multiple variables. To simplify the process of analyzing meteorological data across multiple dimensions, we wrote a tool that uses a generalized format for each meteorological data point to store and query a collection of data.

Our tool, known as Wintx, permits queries focused on a location over a duration of time within a single command. Performing a similar query with data in the GRIB format would require analyzing multiple GRIB files, expanding their compressed binary data, filtering the geospatial grids on multiple levels, then referencing the geospatial locations with the variable data. Adding additional constraints with Wintx, such as limiting the data in the query to a temperature range, is also done in the same query command. In order to add the same constraint for data in the GRIB format, additional filters are required to accommodate each file.

Requirements

Our requirements were formed to satisfy the need for scalability, space efficiency, and geospatial searches:

- Generalized format for storing and retrieving meteorological data
- Interface to store and query data across multiple variables
- Stores data in a scalable architecture
- Allow geospatial queries from shape files
- Imports the data to keep up with daily weather prediction operations
- Efficiently utilizes storage space
- Provides access to data through web based services

Design

Wintx is a Python library written to provide an abstract interface for Python scripts to use. The interface encapsulates interactions with a sharded database, allowing developers to write programs that interact with their data without using database or sharding functions. A RESTful[2] service has been developed using this interface to expand Wintx's functionality to web based services. The generalized format, developed to interact with meteorological data, allows for the database to change without impacting the users interaction with Wintx.

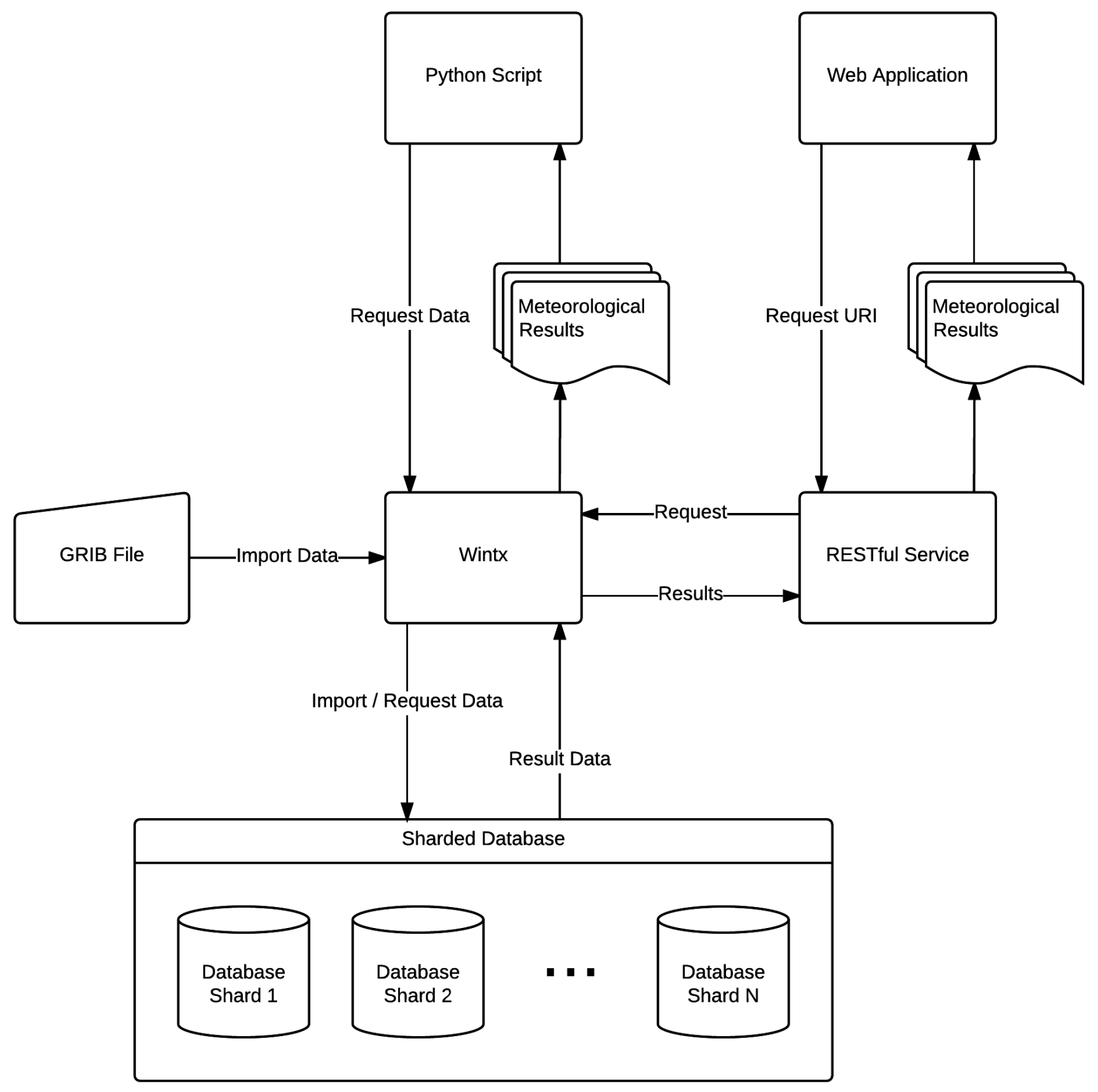

Figure 1: The Wintx workflow model

Database Disk Space Usage

A NoSQL database (MongoDB[3]) and a relational database (MySQL Fabric[1]) were tested as solutions for Wintx. MongoDB was tested with and without indexes.

NoSQL databases store uncompressed metadata with each data point, unlike relational databases which can normalize data and compress the duplicate metadata. As seen in Figure 2, the MySQL Fabric solution stores a data point in the same space as the MongoDB, no-index solution and in less space than the MongoDB, indexed solution. MySQL Fabric is the ideal solution in terms of storing data efficiently. GRIB files no longer need to be maintained once ingested into Wintx. Only database sizes are taken into consideration for these tests.

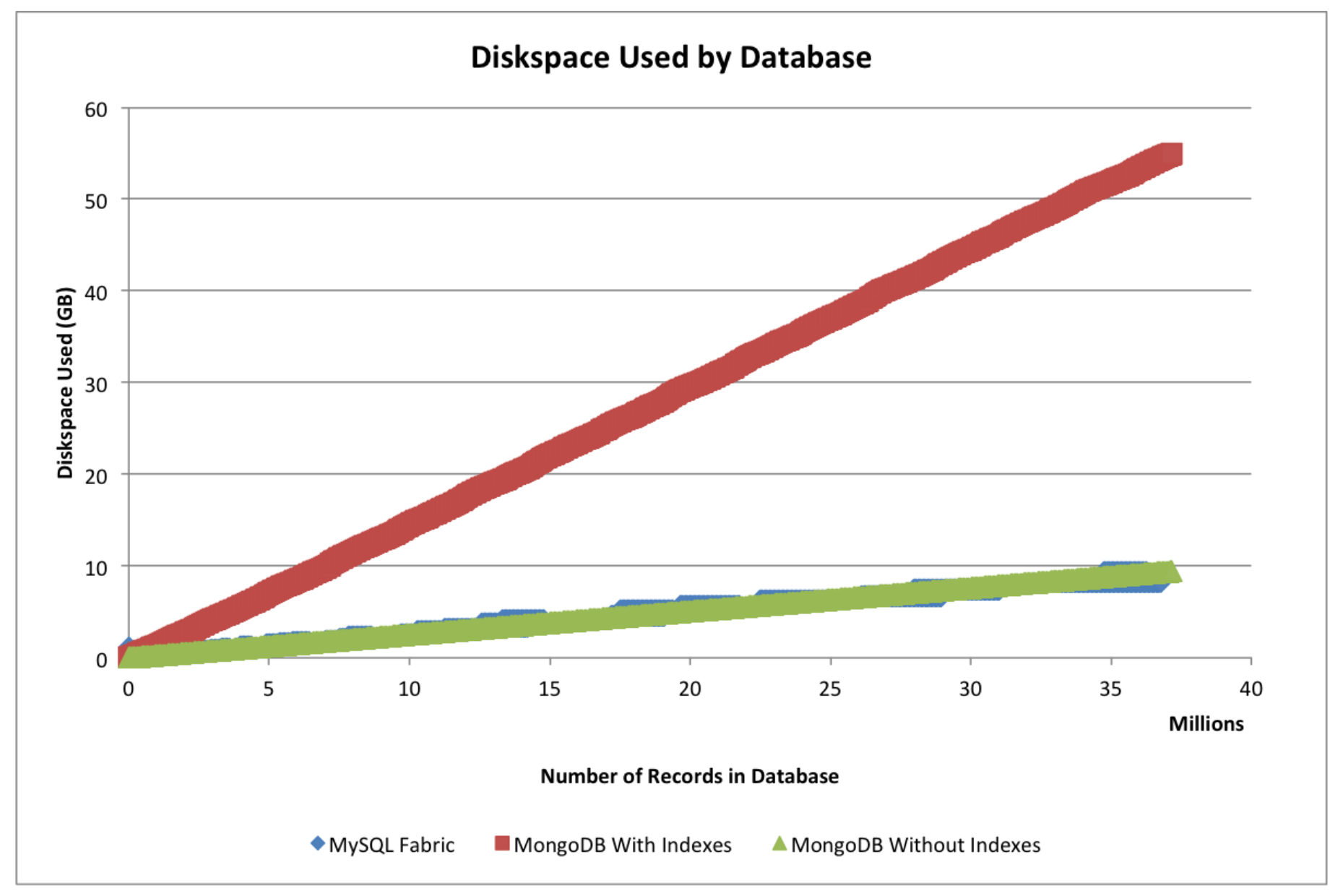

Figure 2: Amount of space (in gigabytes) used on disk by each database. Each GRIB file tested contained # records.

Database Import Speeds

The underlying database needs to be capable of importing GRIB files as quickly as possible to keep up with daily weather prediction operations. Figure 3 displays how many records per second each database solution can import from a GRIB file. MySQL Fabric and MongoDB without indexing have about the same upper bound import speed, though MySQL Fabric's lower bound import speeds are faster than MongoDB's. MySQL Fabric will import records consistently faster than MongoDB, especially with indexes.

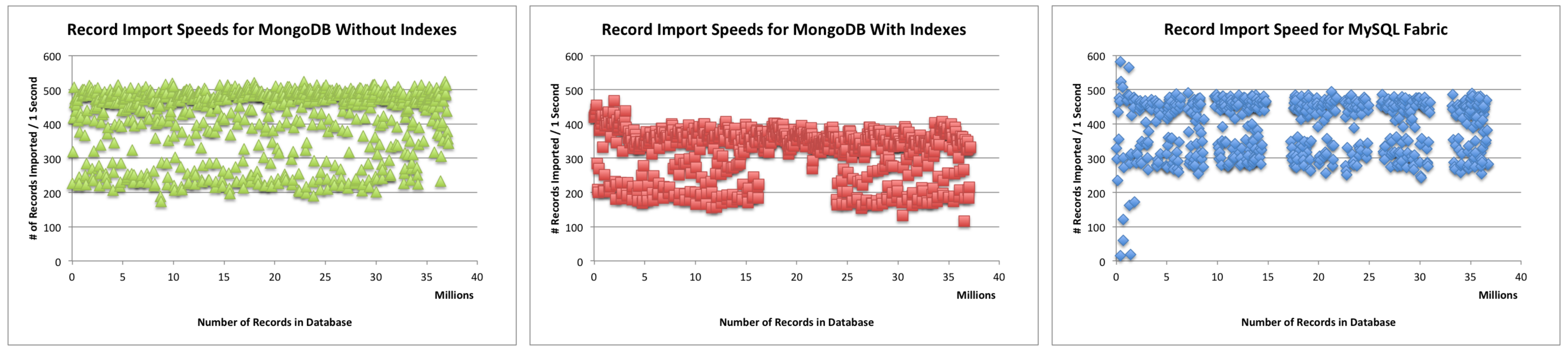

Figure 3: The speeds of individual databases.

Conclusion

Wintx was created to satisfy the need for scalability, space efficiency, and geospatial searching with meteorological data. Our solution, Wintx, provides a generalized method for interacting with data. It includes an easy query mechanism, and is built using MySQL Fabric rather than MongoDB as the underlying database. MySQL Fabric used disk space more efficiently and consistently imported records at a faster rate while still providing the ability to shard the data set and perform geospatial queries. Wintx offers atmospheric scientists a resource to trivially store, access, and query their data. A web REST API on top of Wintx enables collaboration between scientists and developers to build web based applications.

References

[1] Oracle Corporation.
MySQL Fabric.

[2] Todd Fredrich.
REST API Tutorial, August 2013.

[3] MongoDB Inc.
MongoDB.

[4] World Meteorological Organization.
Guide to GRIB.

Acknowledgements

This work was supported by the Indiana Department of Transportation. The contents of this paper reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein, and do not necessarily reflect the official views or policies of the sponsoring organizations. These contents do not constitute a standard, specification, or regulation.

Contact Information

- Email: sethcook@purdue.edu