DATA ENGINEERING AUDIT AND RECOMMENDATION

KEVIN JOHNSON

Management Overview

Precision agriculture is an analytics-based approached to farm management. By observing, managing, and analyzing the needs of individual fields and crops, farmers are able to apply more focused attention to their operations. Use cases of analytics for farmers are numerous. Granular data on rainfall patterns, water cycles, and fertilizer requirements help decide what crops to plant for better profitability and decide the right time of year to harvest. Farmers are also able to better manage production of their crops by pinpointing operational inefficiencies, tracking the quality of their soil, and avoid overuse of chemicals. Lastly, farmers are able to use data to optimize farm equipment and cut down on maintenance costs.

One recent entrant to the precision agriculture market is FarmOS. FarmOS is a web-based application for farm management, planning, and record keeping. The application was founded by Michael Stenta, a software developer and former farmer. FarmOS runs on two different tiers. The first tier is DIY, where the user hosts it themselves. Code for this tier is available through the FarmOS Github (https://github.com/farmOS). Users can download this code themselves and modify and to fit their needs. This software is built on Drupal, which is a free and open-source web application framework. The second tier of FarmOS is Farmier, their subscription hosting service. Farmier provides hosting, automatic updates, backups, and SSL security. The standard subscription for Farmier is $75 per year.

Design Methodology

FarmOS uses an open source development process. The rationale behind using an open source process is that as farmers spend their winter months unable to tend to crops, they can develop new solutions for improving their operations and share ideas with other farmers. The open network allows for individual contributors to submit their own modules. Users and developers communicate through their open forums (https://farmos.discourse.group/) to discuss support requests, workflow ideas, improvements, etc. The application is updated by users submitting feature requests, bug reports, or new capabilities through an issue queue.

System Architecture

Back-End

FarmOS was built using Drupal as its back-end framework. Drupal is a free and open-source web-based framework written in PHP and distributed under GNU General Public License. FarmOS is built off the Drupal Core codebase with a set of pre-selected entity types. The four Drupal entity types for FarmOS are as follows:

1. Logs: This is used to represent various types of events that are recorded on a farm. Logs can be labeled as "Done" or "Not Done," and can also be Rescheduled, Assigned, and Cloned. FarmOS was primarily structured around Logs. Below are the standard Log options:
    - Activities
    - Eggs
    - Harvests
    - Inputs
    - Maintenance
    - Medical
    - Observations
    - Seedings
    - Soil Tests
    - Transplantings
    - Logs with Movements
2. Farm Assets: This entity type is used to represent assets on the farm. The core set of assets types on the platform include Plantings, Animal, and Equipment.
3. Areas: There are the geometric and field areas of a farm. There are three layers built under Areas:
    - Area Layers: Bed, Building, Field, Greenhouse, Landmark, Paddock, Property, Water
    - Base Layers: Google Hybrid, OpenStreetMap
    - Soil Layers: NRCS Soil Survey
4. Taxonomy Terms: These are used for categorization and tagging. Terms are organized into different "Vocabularies," and the FarmOS platform provides a number of different Vocabularies that are used throughout the system.
5. Users: These provide the mechanism through which users can log in and use the system.

Database

FarmOS is dependent on Docker, which is an open-source project that automates the deployment of applications inside software containers. The FarmOS Docker image can be used in both development and production environments. This Docker image contains the FarmOS codebase with all dependencies.

The database Docker defaults to is MariaDB. MariaDB is a popular open source relational database that provides a SQL interface for accessing data. Within MariaDB, FarmOS uses a hierarchical data structure based on entity types. In addition to MariaDB, Drupal's database abstraction layer supports PostgreSQL. Use of PostgreSQL allows the ability to leverage PostGIS, which supports spatial data and allows for location queries to be run in SQL. Users interact with the database using a REST API. The API provides CRUD operations for all the entity types.

Because of the open source nature of FarmOS, much of the data is user submitted or manually updated. This is especially true for Logs. However, the application also provides a framework for receiving data from automated environment sensors.

<u>Front-End</u>

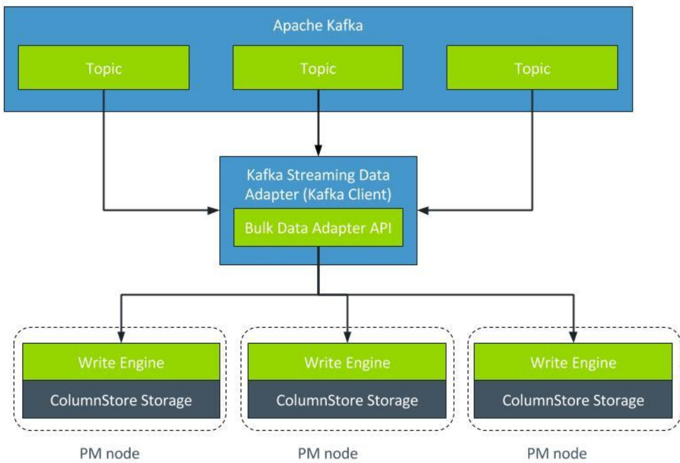
The FarmOS front-end framework uses Bootstrap, a collection of HTML, CSS, and JavaScript tools for creating and building web pages and web applications. It is a free and open source project hosted on Github. One of the benefits of Bootstrap is its grid system, which allows the application to look good on all screen sizes.

Streaming Recommendation

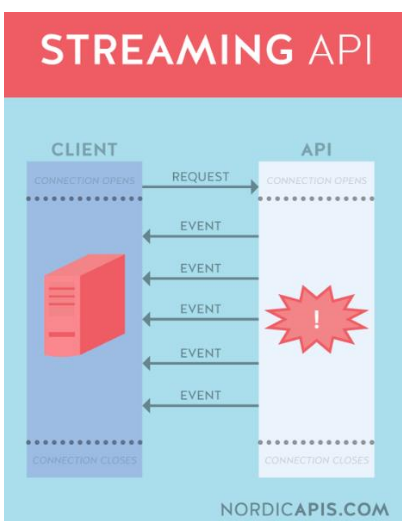
One of the most important aspects for farming is monitoring soil health. Vegetative growth of crops is sensitive to soil moisture, as optimal soil moisture conditions help maintain cell division and expansion, and transport nutrients to the actively growing sites of the plant. For many farmers, collecting hard data can still be a very manual process. The farmer must first scout and collecting soil and then sent them to a lab for testing. The whole process can take days to weeks. Technological development in field-based sensors have been shrinking these timelines. Now, farmers are able to receive real-time measurements from the field to help them make quicker and more efficient decisions (Tzach). There is an opportunity for FarmOS to harness this technology for their users.

Setting up stream processing for soil data can be done using field sensors and Rasberry Pi. One widely used sensor that allows for this is the Wireless Leaf & Soil Moisture/Temperature Station by Davis Instruments.

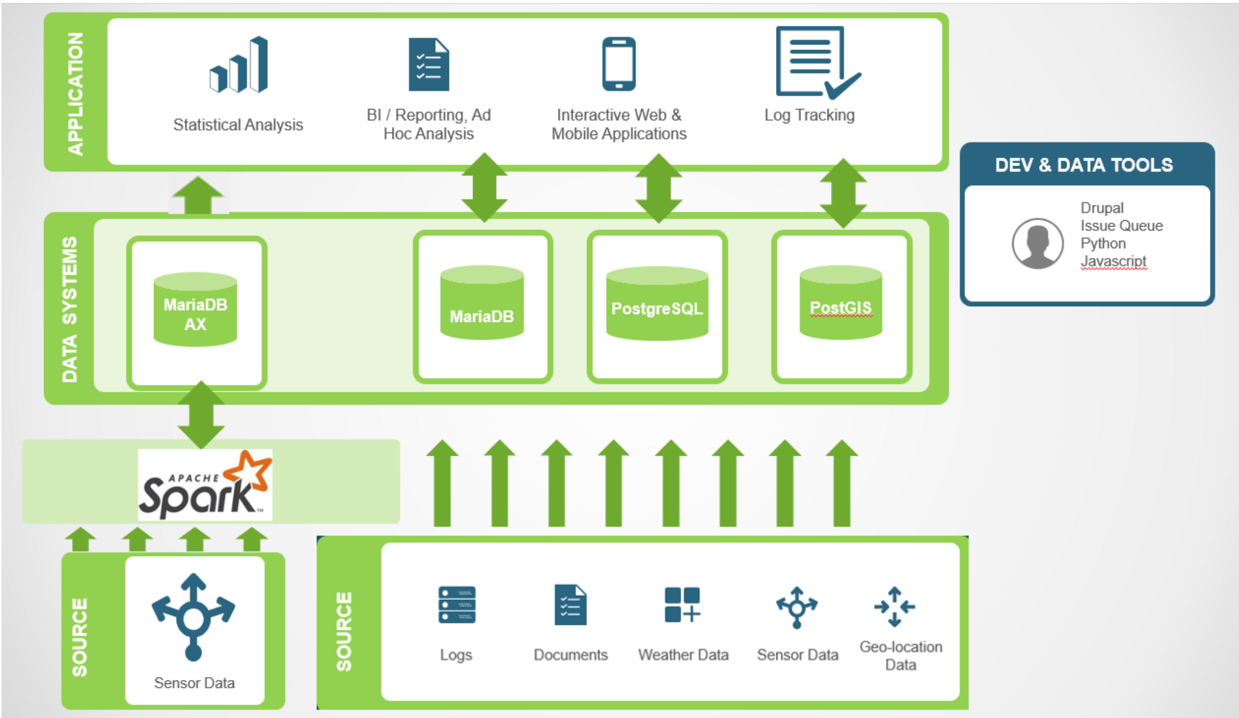
After the data has been collected it needs to be sent to the database. Rasberry Pi will push this data to the message queue, where the streaming system can push it to the database. The streaming system we can use for this task is Apache Kafka. Apache Kafka is an open-source distributed streaming platform that allows for publishing and subscribing to stream of records. When the data is sent to the queue in Apache Kafka, the Kafka Adapter will send it through to the database. Since MariaDB already serves as the default database for FarmOS, we will use MariaDB AX as our database for the streaming data. MariaDB AX is the analytics solution for MariaDB and allows for OLAP. Below is a graph of how this process works, as provided by MariaDB.



Now that the streaming data is being collected into the database, the last step will be to make it available to the end user. This will be done using a Streaming API. While REST APIs are request/response tools between client and server, Streaming APIs have a limited response format. Rather than relying on response-oriented servers, Streaming APIs utilize event brokers to manage this interaction.



In summary, we will be setting up a stream processing pipeline for soil moisture data. The data collection will be done using the Wireless Leaf & Soil Moisture/Temperature Station by Davis Instruments. This data will be pushed to Rasberry Pi and then sent to the queue in Apache Kafka. Apache Kafka will then use its adapter to push to the MariDB AX database, where it can then be sent to the client via a Streaming API. The chart of the system is included below.



Resources

https://farmos.org/

https://farmos.discourse.group/

Crocker-White, Chris. “Build an Air Quality Monitor with InfluxDB, Grafana, and Docker on a Raspberry Pi.” *Balena*, Balena, 9 Jan. 2020, www.balena.io/blog/build-an-environment-and-air-quality-monitor-with-raspberry-pi/.

MariaDB. “Real-Time Data Streaming with MariaDB AX.” *MariaDB*, 26 Aug. 2019, mariadb.com/resources/blog/real-time-data-streaming-with-mariadb-ax/.

Tzach, Tomer, et al. “Soil Sensors: A New Direction in Precision Agriculture to Improve Crop Production.” *PrecisionAg*, 25 Sept. 2019, www.precisionag.com/in-field-technologies/connectivity/soil-sensors-a-new-direction-in-precision-agriculture-to-improve-crop-production/.