Matsu Workflow: Web Tiles and Analytics over MapReduce for Multispectral and Hyperspectral Images

Open Cloud Consortium
Open Data Group

September 7, 2012

Materials for Matsu

Matsu is a public project. Currently, code and documentation are maintained on Github. The Open Cloud Consortium has an account on Github where projects may be released from time-to-time. The official Open Cloud Consortium account is `github.com/opencloudconsortium`. The Matsu project page is `github.com/opencloudconsortium/matsu-project`.

Information about the computing resources available from the OCC can be found at `opencloudconsortium.org`.

Summary

Matsu[1] is a project of the Open Cloud Consortium. A goal of the Matsu project is to use an open source cloud-based infrastructure to make high quality satellite image data accessible through an Open Geospatial Compliant (OGC)-compliant Web Map Service (WMS). Additionally, a workflow has been created using MapReduce which generalizes image processing and can be used for running distributed analytics over satellite images.

Matsu incorporates the following open source technologies and applications:

- GlusterFS;
- Hadoop and the Hadoop Streaming Interface;
- Accumulo;
- R;
- Python;
- Augustus; and
- PMML.

Contributors to the project have included the National Aeronautics and Space Administration (NASA), TexelTek, the University of Chicago, the Laboratory for Advanced Computing (LAC), and Open Data.

[1]Matsu, or Mazu, is a goddess of the sea said to protect fishermen and sailors. Project Matsu was initially a response to the 2010 Haiti earthquake and the name was chosen in the spirit of aiding those in need.

Contents

1 Introduction

One of the goals of the Matsu Project is to use an open source cloud-based infrastructure to make high quality satellite image data accessible through an Open Geospatial Compliant (OGC)-compliant Web Map Service (WMS). Another goal is to develop an open source cloud-based analytic framework for analyzing individual images, as well as collections of images. A third goal is to generalize this framework to manage and analyze other types of spatial-temporal data.

2 Matsu Data Flow

The Open Science Data Cloud (OSDC) is used for the storage and processing of EO-1 images. The workflow is:

Performed by NASA as part of their Daily Operations

---

1. Transmit data from NASA's EO-1 Satellite to NASA ground stations and then to NASA Goddard;
2. Align data and generate Level 0 images;
3. Transmit Level 0 data from NASA Goddard to the Open Cloud Consortium's (OCC) OSDC;

Run by NASA on the OSDC for Matsu and other Projects

---

4. Store Level 0 images in the OSDC Storage Cloud for long-term, active storage (GlusterFS).
5. Within the OSDC, Launch Virtual Machines (VMs) specifically built to render Level 1 images from Level 0. Each Level 1 band is saved as a distinct image file (Geo-TIFF);
6. Store Level 1 band images in the OSDC Storage Cloud for long-term storage;

Run Specifically for Project Matsu [Under Development]

---

7. Read Level 1 images, combine bands, and serialize image bands into a single file;

8. Store serialized files on HDFS;

   Map Tiling

9. Run a MapReduce job using the Hadoop Streaming Interface to create web map tiles from the serialized Level 1 images stored in HDFS;

   (a) Divide ("chunk") images into manageable sizes, suitable for processing and display on the WMS. We use *dyadic decompositions* for the zooming / chunking. We start at the natural resolution of the image and increase in area covered by $4x$ for each decrease in zoom level. The image size in pixels is held constant. MapReduce is well-suited for processing images in this way.

   (b) Tag each chunked image with the bounding box, date, time, dyadic level, and bands used. Convert to the PNG format to simplify further processing;

10. Load PNG files into Accumulo for access by the WMS. Accumulo is the tile store;

    Analytics

11. Run a MapReduce job using any of the Hadoop interfaces to run an analysis on the serialized Level 1 images stored in HDFS;

12. Store the results of the analysis in Accumulo for display in the WMS.

Satellite data comes from the Hyperion (hyperspectrum) and ALI (multispectrum) instruments on NASA's Earth Observation Satellite 1 (EO-1). Level 1 images are either radiometrically or geometrically corrected.

3 Satellite Imagery to Web Map Tiles

The available level 1 images have 10 (ALI) or 155 (Hyperion) usable bands (some bands are not calibrated). Any number of bands may be used in an analytic, but for web map tiles, an accurate visible image must be assembled. Level 1 images cover narrow strips of large geographical distances. For web map tiles, we want fixed-size (width and height in pixels) images of increasing resolution.

The process to do this is to read serialized images and select bands representing the visual spectrum. Images are broken apart and sent to reducers handling specific, large, geographical regions. Reducers convert images in web map tiles. This starts at the most granular (highest zoom) level, and moves towards the least granular. The resources required to do each tile decreases with each step. Each decrease in zoom level means a lower resolution over the fixed tile size, which results in a smaller image size, decreasing the memory requirements.

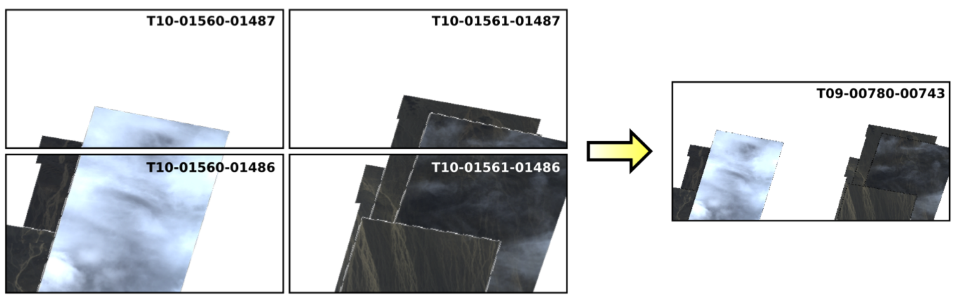

Figure 1: Building a tiles for a zoom level. A single tile at zoom level z is comprised of four tiles from zoom level $z+1$.

A tile may be made up of several level 1 images. Conversely, a level 1 image may span several tiles.

4 MapReduce

To build the tile cache for display in WMS and a dyadic decomposition for use with analytics, the band images have to be broken into smaller pieces and aligned. The mapper loads images with specified date, geographical region, and bands. The bands are assembled to create an image, and then the bytes are partitioned by bounding boxes.

In the reducer, the images are built as PNGs and saved to Accumulo. The saved images are:

- PNG files that can be more easily used with image libraries than the 16-bit Geo-TIFFs of the original band data;
- A combination of specific visual bands; and
- Indexed by date, time, geospatial bounding box, zoom level, map projection, and original instrument.

4.1 Preprocessing

The Level images are separated into files for each band. When the images are uploaded to HDFS, the bands are combined and the image is serialized. When an image is read by the mapper, the desired bands must be specified and are extracted from the serialized image.

All usable bands for an image are serialized so that the resulting file in HDFS is usable for a variety of analytics.

4.2 Map

The map step of the web tiling process runs from serialized image files. The pre-processing uses Python to convert the band images into a file format more easily ingested by Hadoop. The map itself is written in Python for use with the Hadoop Streaming interface. It reads the image into a NumPy array.

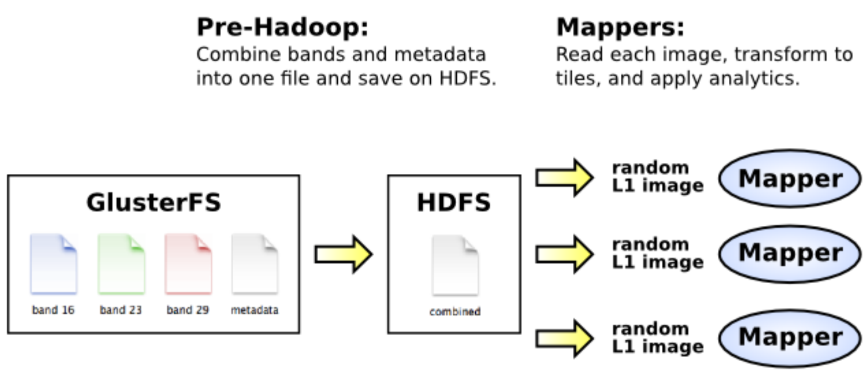

Figure 2: Diagram showing the pre-processing step and mappers acting on pre-preprocessed images in HDFS. An entire image is read as a single unit.

4.3 Reduce

The reduce steps processes images for a geographical region. All tiles, at all zoom levels, which fall in the region are processed together. As an implementation detail, tiles are created at the most granular level first. The images from a zoom level are used to create the images for the zoom level above it.

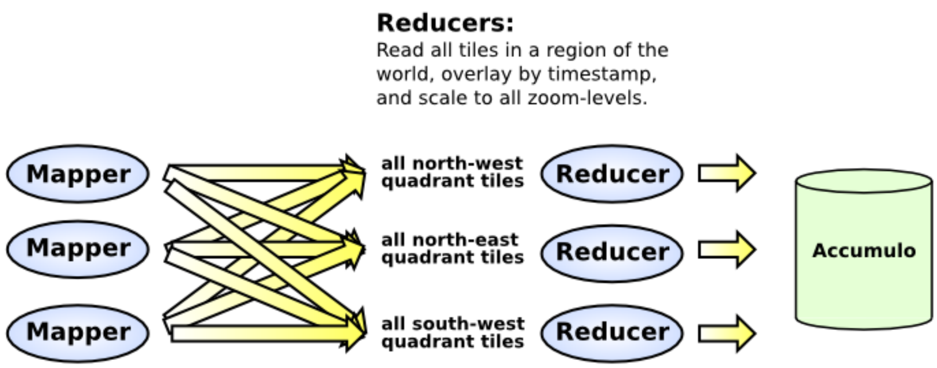

Figure 3: Diagram showing the shuffle of image data to reducers. Each reducer processes a geospatial region and assembles them web map tiles.

4.4 Accumulo

From the reducer, the images are sent to Accumulo. The tiles are indexed in a way that different regions and times can be displayed using the WMS.

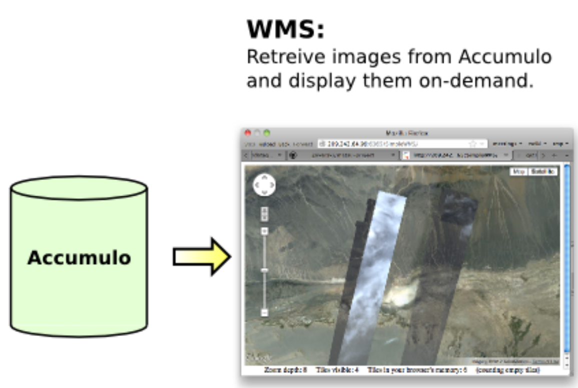

Figure 4: A sample WMS page displayed in a web browser. Google Maps is used for the background. The tiles built from EO-1 data are clearly visible in the center of the image.

4.5 Applying Analytics

In the reducer, we can plug in other computations besides the ones for display over WMS. Statistical models can be built from the data and stored in HDFS. Figure 5 below shows Augustus[2] being used to build PMML models from the data in the reducer.

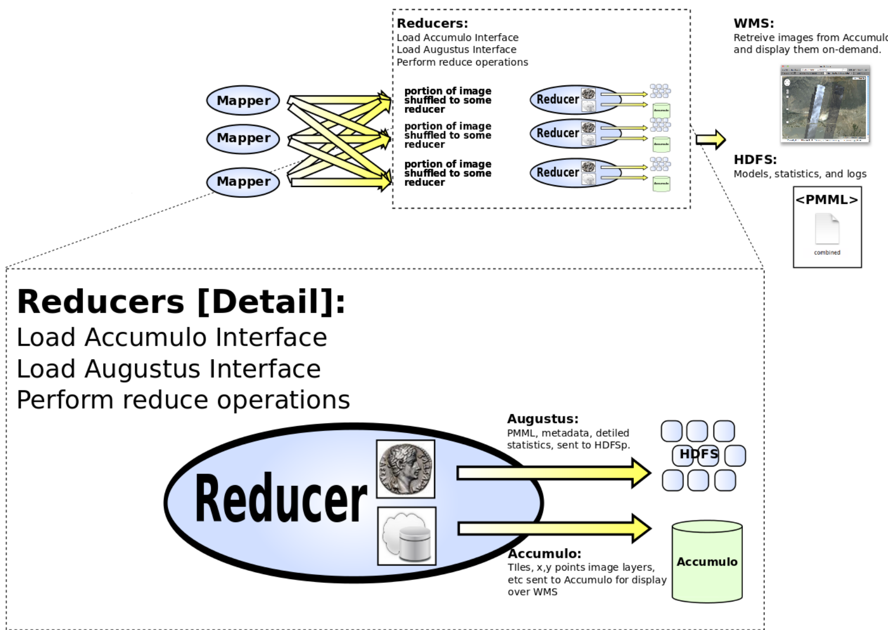

Figure 5: A sample WMS page displayed in a web browser. Google Maps is used for the background. The tiles built from EO-1 data are clearly visible in the center of the image.

5 Storage

Currently, there is over 100 terabytes of level 0 and 1 images stored on GlusterFS. The filesystem is accessed over a wide are network (WAN). The web tiles and analytics require additional space. Storage capacity, accessibility,

[2]Augustus is written in Python and is an open source system for building and scoring statistical models. It is maintained by Open Data and is available from Google Code at `augustus.googlecode.com`.

speed, and fault tolerance is a large concern for this project. How and where some of the components of Project Matsu are stored are given below:

Level 0 and Level 1 images are stored on GlusterFS on the Open Science Data Cloud. GlusterFS is a shared resource and offers a fault-tolerant, distributed file system useful for long-term storage and access. Level 0 and 1 images are accessible by several projects using the OSDC. Older images may be purged if space becomes limited on the OSDC.

Web tiles generated by MapReduce are stored in Accumulo. Accumulo offers distributed, fault-tolerated data storage with built-in, cell level security. The Accumulo instance is dedicated to Matsu. Older tiles may be purged if space on the nodes become limited.

Analytic outputs from MapReduce jobs are stored in Accumulo but may additionally be stored in HDFS or local disk, depending on the nature of the analytic. The output from an analytic may be an image or ascii data. These data will be kept as long as possible.

Frequently requested images can be cached in a layer between the NoSQL database and the servlet to improve performance. This is ephemeral storage.

6 About Project Matsu

Matsu is an Open Cloud Consortium (OCC)-sponsored project. The source code and documentation will be made available on GitHub under Open Cloud Consortium (`https://github.com/opencloudconsortium`).

The OCC is a not for profit that manages and operates cloud computing infrastructure to support scientific, environmental, medical and health care research. The OCC is focused on using this technology to make scientific advances by working with scientists in a variety of disciplines. Visit us at `http://opencloudconsortium.org` or, for more information, email `info {at} opencloudconsortium {dot} org`.

Open Data is a member of the OCC. Open Data began operations in 2001, specializes in building predictive models over big data, and is one of the pioneers using technologies such as Hadoop and NoSQL databases so that companies can build predictive models efficiently over all of their data. More information can be found at `http://opendatagroup.com` or by emailing `info {at} opendatagroup {dot} com`.