UTILIZING A SPARK-BASED CLOUD COMPUTING FRAMEWORK TO ADDRESS BIG REMOTE-SENSING DATA PROCESSING: MULTI-SCALE LARGE LANDSAT 8 DATASET AND MODIS AS AN EXAMPLE

Hai Lan

hlan@terpmail.umd.edu

Department of Geographical Sciences
University of Maryland College Park

INTRODUCTION

- Remotely sensed data is one of the **principal geospatial data sources** that are accessible to support research studies in earth science, environmental science, and urban planning (Wulder & Coops, 2014).
- By processing remote sensing data, it is possible to provide significant support for **open scientific questions** or generate **more detailed and insightful results** for existing research (Yang et al., 2017).
- With the rapid development of remote sensing and computer engineering technologies, increasing **amounts of imagery data, meteorological data, environmental data, hydrological data, biological data etc.** have been collected quickly and routinely (Ma et al., 2015).
- The question is: how to fully exploit the benefits from those **big data** to help better solve geospatial research questions, e.g., dust storm forecasting (Q. Huang *et al. 2013*) and high-resolution global forest cover change mapping(Hansen et al., 2013).

MOTIVATION

A open source cloud computing framework for large and multi-spatial, multi-spectral, multi-temporal remotely sensed data classification and change detection

1. Advances in observational platforms that are routinely generating massive amounts of remotely sensed datasets are increasingly providing further support to answering scientific questions in geographical fields with unprecedented details (Ma et al., 2015).
2. A grand challenge is to process those datasets into valuable information(Yang et al., 2017).
3. Most of the studies with pure remote-sensing data are usually limited to a small study area, with only a few scenes of data, or to low-resolution remotely sensed images for large-area experiments (Wang, Z. et al. 2017).
4. Processing the massive volume of remotely sensed data is not the only problem, as the intrinsic complexity of such data is also an important issue that must be considered.

RESEARCH OBJECTIVES

Identify elements that comprise a cloud computing framework for large and multi-spatial, multi-spectral, multi-temporal remotely sensed data classification and change detection

- Design and implement a universal scalable solution (with alterable core function) to classify and process change detection for multi-sourced massive remote sensing imagery dataset, i.e., existing Landsat TM/ETM+, MODIS, SPOT and near horizon dataset

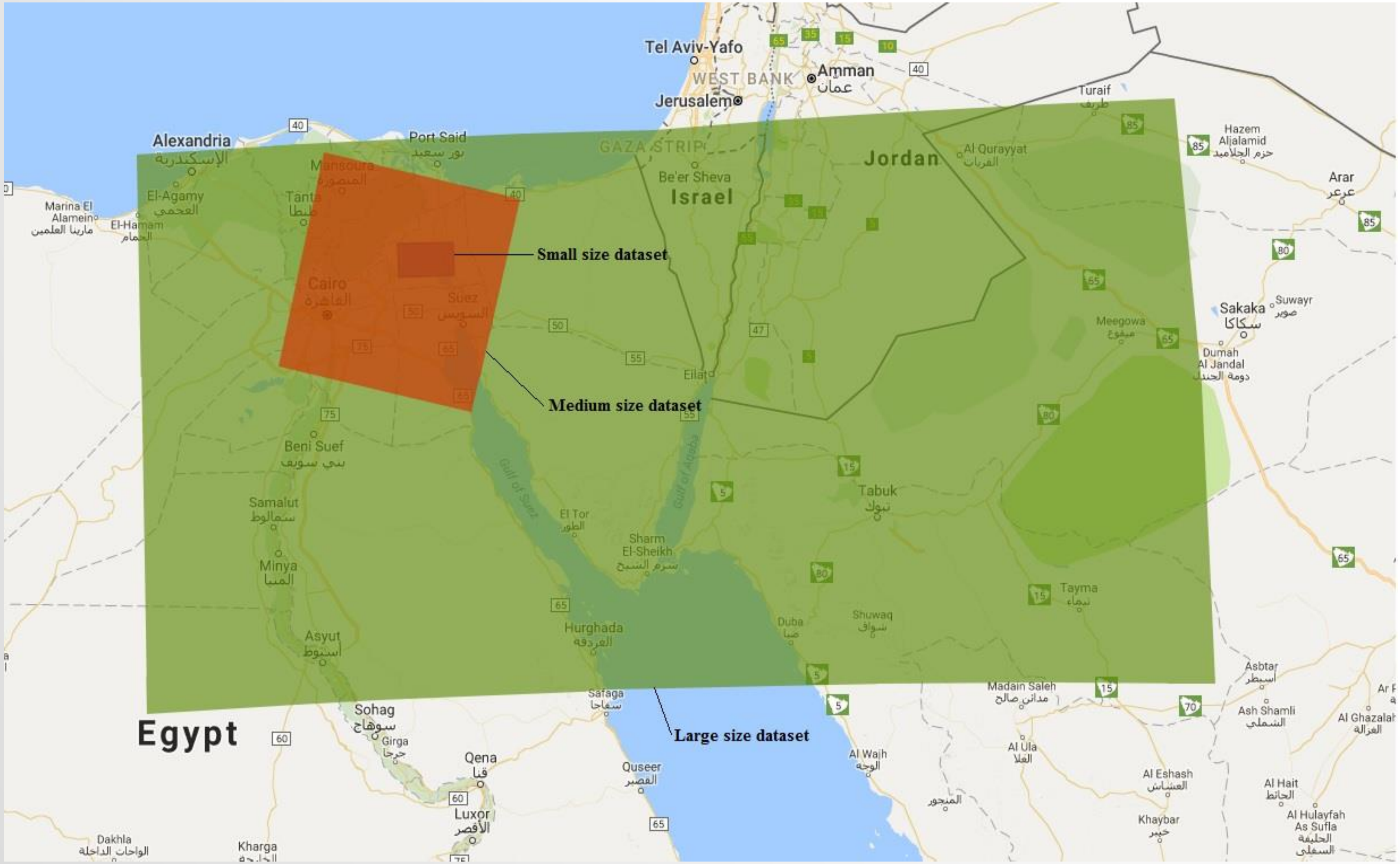
CHALLENGES

1. How to implement a tool that can process different proposed remote sensed tasks with only minor adjustments rather than fully rebuild new toolkits.

2. How to make it fully capable of exploiting benefits from cutting-edge cloud computing technologies, resources and platforms (includes partitioning strategies)

3. How to visualize the result from cloud directly to help users examine results easily and quickly

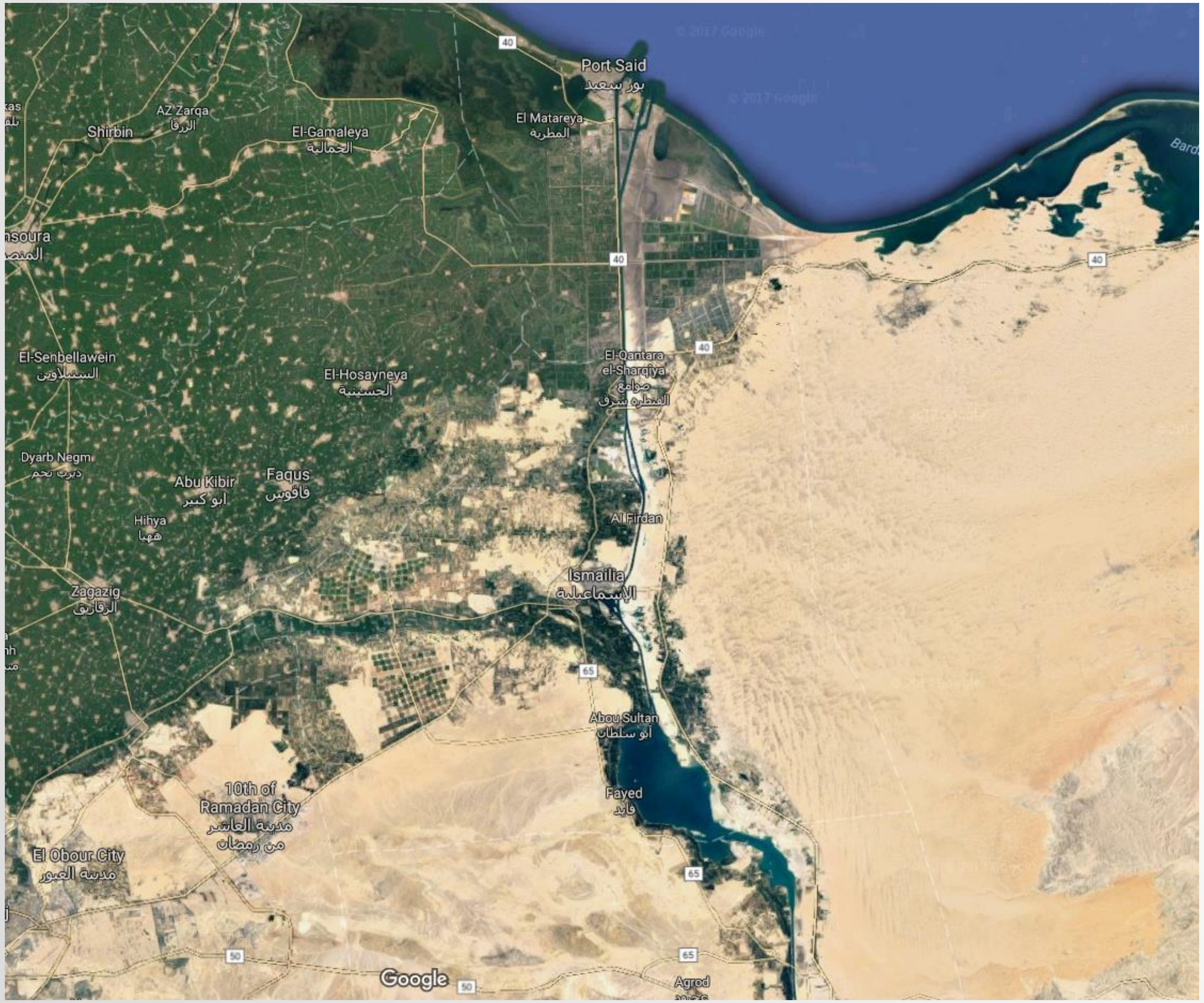
DATA SOURCES



Landsat 8 Operational Land Imager (OLI) dataset in three different scales

MODIS/Terra Surface Reflectance Daily L2G Global 1 km and 500 m SIN Grid V006 (MOD09GA)

DATA SOURCES



Why Suez Canal area

- the land cover mainly includes sparse and dense vegetation, a natural water body, the Suez Canal (with water), and bare sands and rocks.
- Suez Canal was expanded since 2013 to build another branch (Suez Canal Authority, 2013). A good example to test/monitor LULC change

DATA SOURCES

Detailed **acquisition time** of each Landsat 8 dataset

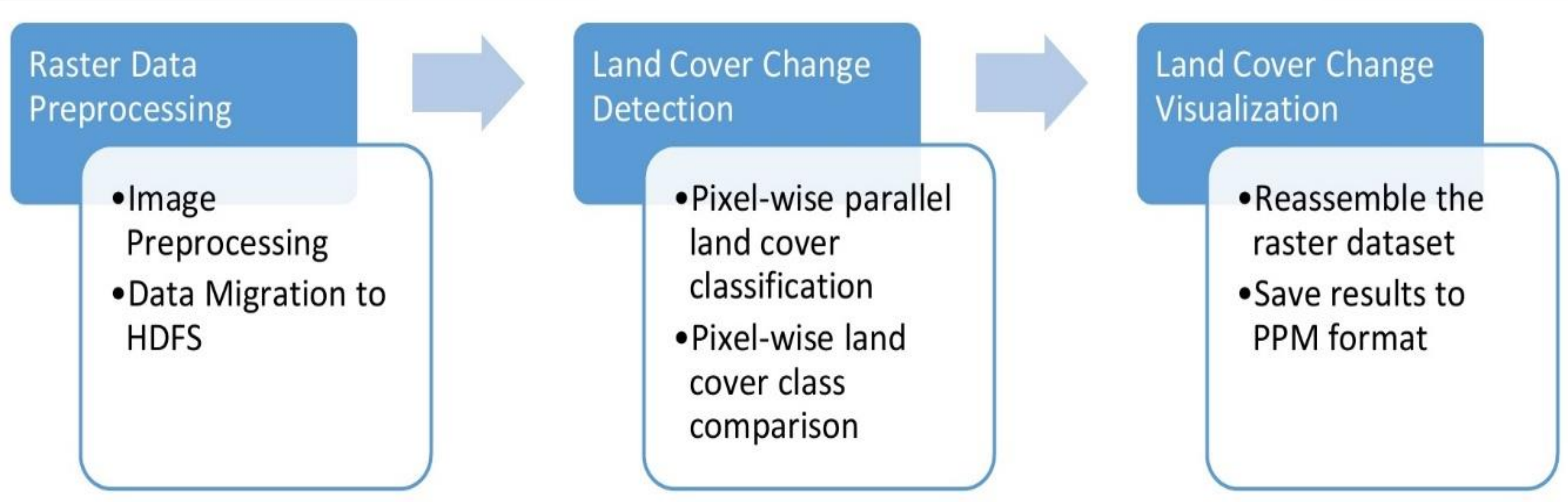
	2013	2014	2015	2016	2017
Landsat_Small	4/29	3/31	4/19	3/4	4/5
Landsat_Medium	4/29	3/31	4/19	3/4	4/5
Landsat_Large	Apr	Mar	Apr	Apr	Apr
MODIS	4/29	3/31	4/19	3/4	4/5

DATA SOURCES

Google Earth Engine(GEE) and Amazon Web Service(AWS) S3

- GEE is a cloud-based platform that can serve remote sensing data source with customized criteria such as the region boundaries and cloud mask (Gorelick et al., 2017).
- AWS has made Landsat 8 and MODIS etc. data freely available on Amazon S3. Anyone can use on-demand computing resources to perform analysis and create new products without cost of storing data or the time to download it.

WORK FLOW



Raster Data Preprocessing

- Image Preprocessing
- Data Migration to HDFS

Land Cover Change Detection

- Pixel-wise parallel land cover classification
- Pixel-wise land cover class comparison

Land Cover Change Visualization

- Reassemble the raster dataset
- Save results to PPM format

- **HDFS:** The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware.
- **PPM format:** Portable Pixel Map format is an easy format to write and manage text-based outputs into human-readable figures.

WORK FLOW

Step 1: extract each band of raster dataset by using the Geospatial Data Abstraction Library (GDAL)

- Resilient Distributed Datasets (RDD) is the basic abstraction of a dataset in Spark.
- For each image, three RDDs will be created for green, red, and near-infrared bands for NDVI and NDWI land cover classification at the initial stage.

$$\mathrm{NDVI} = \frac{NIR - VIS(red)}{NIR + VIS(red)}$$

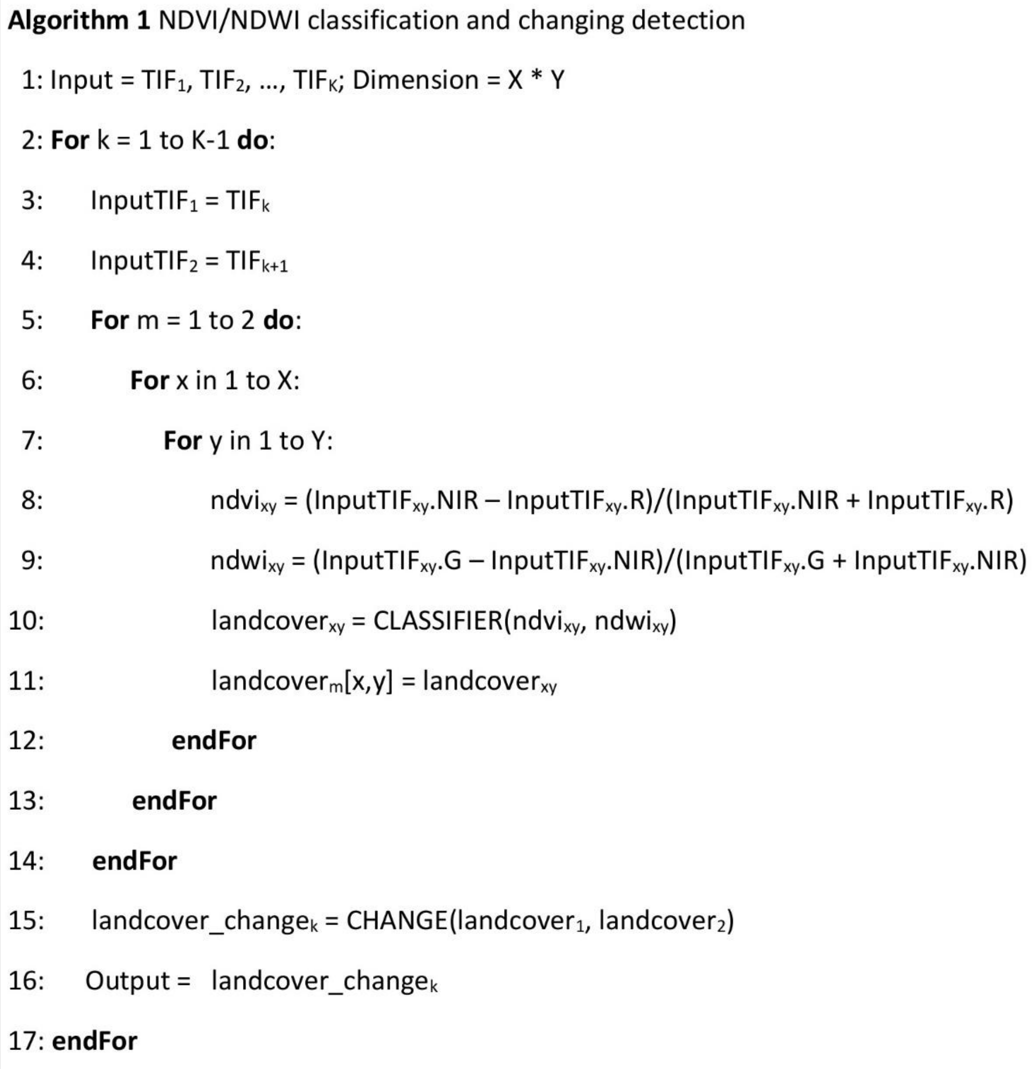
$$\mathrm{NDWI} = \frac{GREEN - NIR}{GREEN + NIR}$$

WORK FLOW

Step 2: perform parallel processing of images classification and land cover change detection.

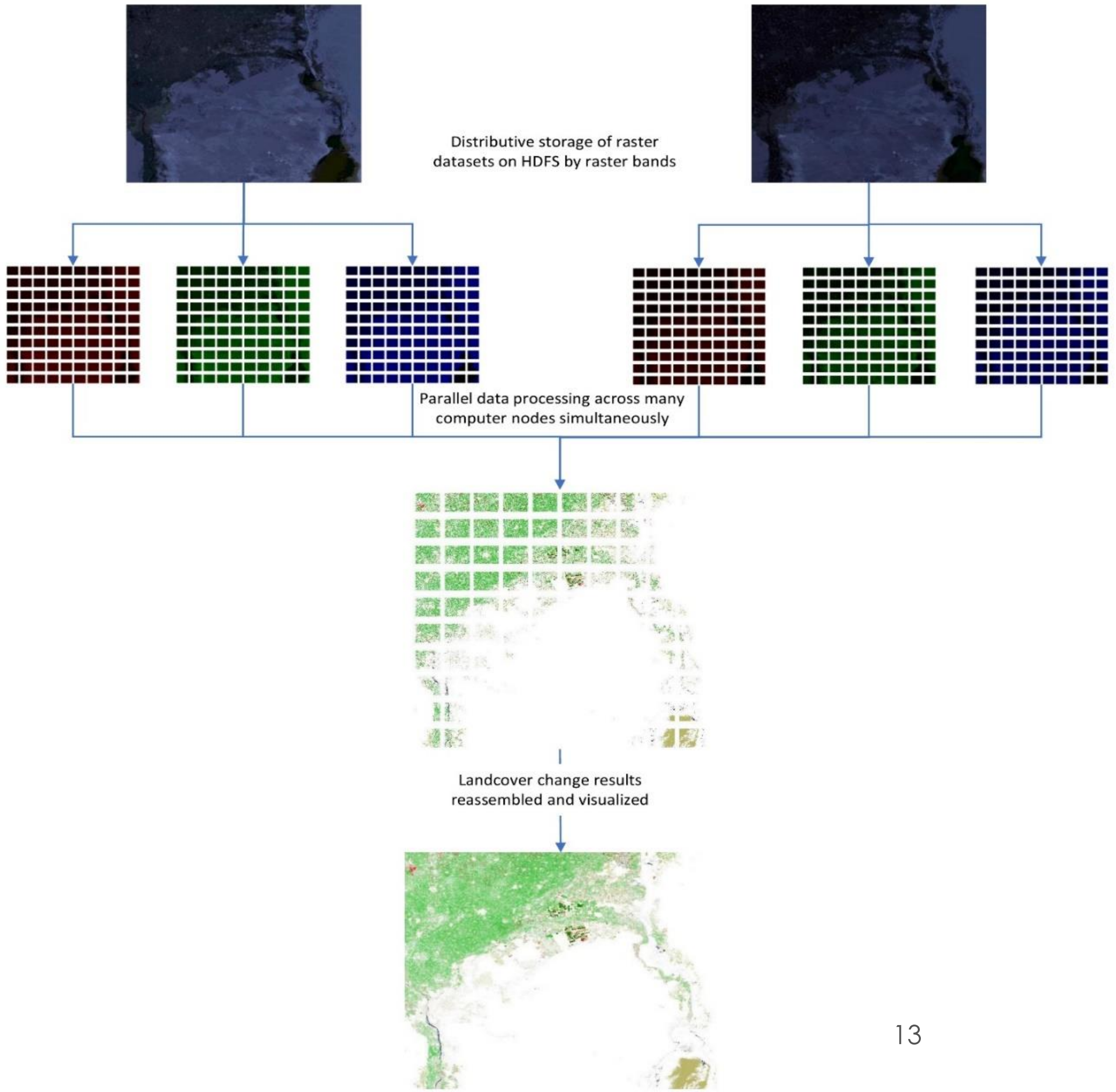
- Because the basic parallel processing unit is pixel-wise, this method can be highly flexible and scalable
- the partition scale cannot be too small or too large. A very small partition scale will result in low-performance computing and even increase the fault recovery cost. A very large partition scale may lead to the out of memory error.
- strip-based partitioning (Ma et al., 2015)

PARALLEL PROCESSING ALGORITHM

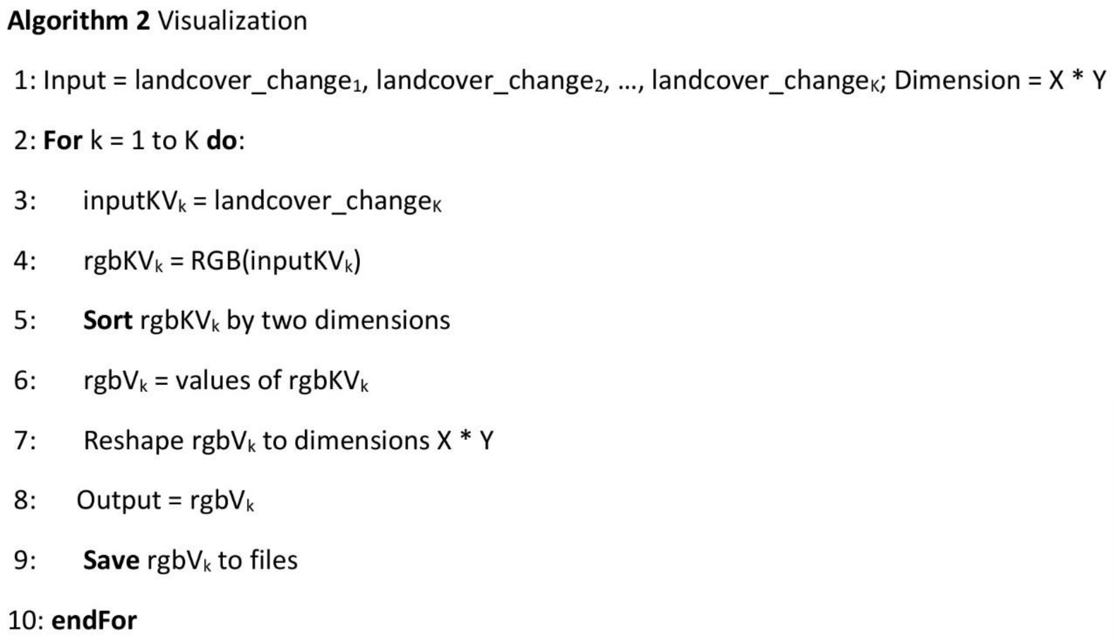


Algorithm 1 NDVI/NDWI classification and changing detection

```
Input = $TIF_1, TIF_2, …, TIF_K$; Dimension = X * Y
For k = 1 to K-1 do:
    $InputTIF_1 = TIF_k$
    $InputTIF_2 = TIF_{k+1}$
    For m = 1 to 2 do:
        For x in 1 to X:
            For y in 1 to Y:
                $ndvi_{xy} = (InputTIF_{xy}.NIR – InputTIF_{xy}.R)/(InputTIF_{xy}.NIR + InputTIF_{xy}.R)$
                $ndwi_{xy} = (InputTIF_{xy}.G – InputTIF_{xy}.NIR)/(InputTIF_{xy}.G + InputTIF_{xy}.NIR)$
                $landcover_{xy} = CLASSIFIER(ndvi_{xy}, ndwi_{xy})$
                $landcover_m[x,y] = landcover_{xy}$
            endFor
        endFor
    endFor
    $landcover\_change_k = CHANGE(landcover_1, landcover_2)$
    Output = $landcover\_change_k$
endFor
```


Distributive storage of raster datasets on HDFS by raster bands
Parallel data processing across many computer nodes simultaneously
Landcover change results reassembled and visualized

VISUALIZATION



Algorithm 2 Visualization

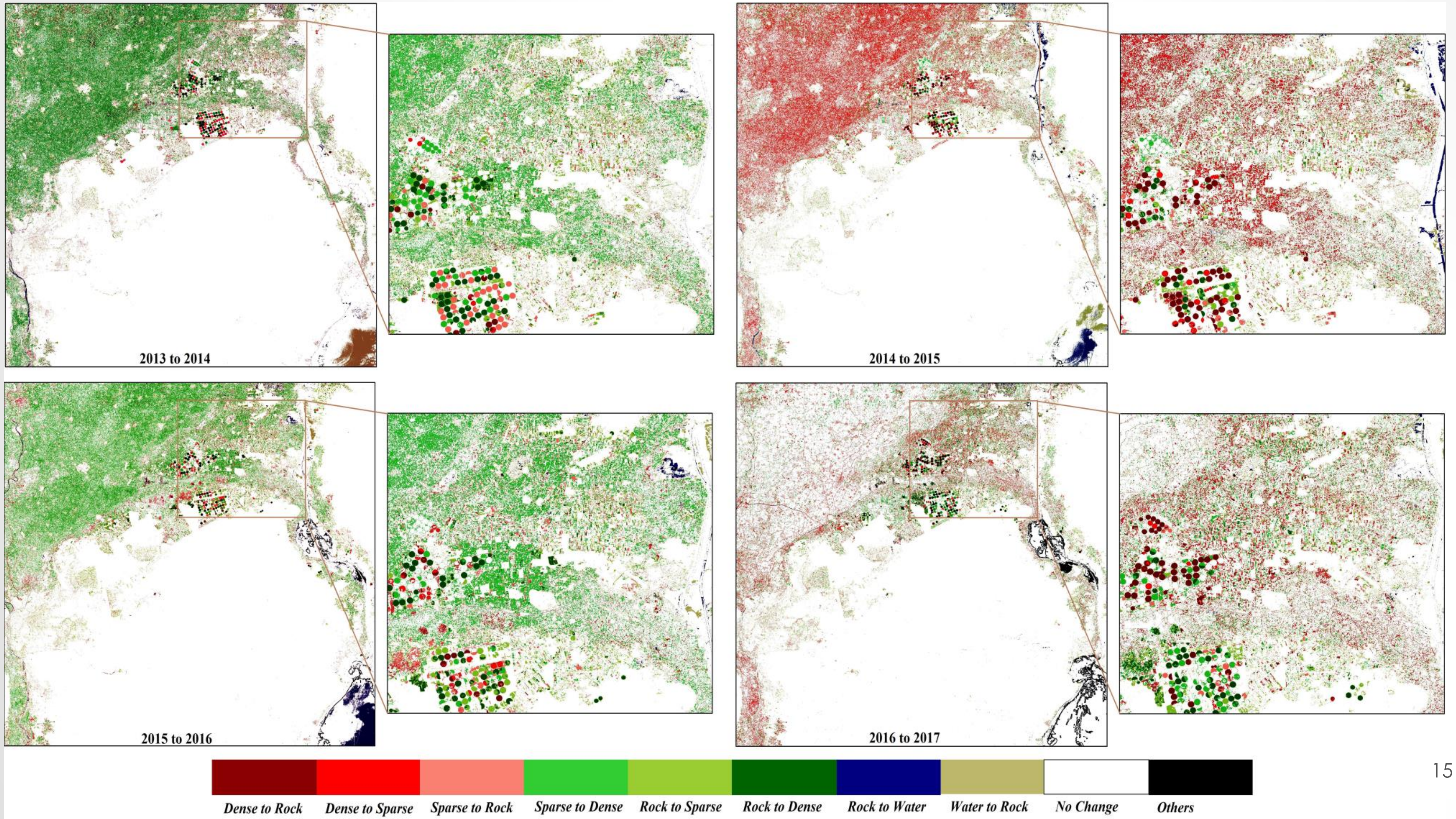
```
Input = $landcover\_change_1, landcover\_change_2, …, landcover\_change_K$; Dimension = X * Y
For k = 1 to K do:
    $inputKV_k = landcover\_change_K$
    $rgbKV_k = RGB(inputKV_k)$
    Sort $rgbKV_k$ by two dimensions
    $rgbV_k$ = values of $rgbKV_k$
    Reshape $rgbV_k$ to dimensions X * Y
    Output = $rgbV_k$
    Save $rgbV_k$ to files
endFor
```

Step 3: Gather results and assign labels to each pixel

Step 4: Reconstruct whole images by sorting output RDDs and visualize them in PPM format.

- values of key-value pairs will be converted to colors according to a user-defined RGB color scheme
- appended to the PPM file header to create a complete PPM image

RESULTS VISUALIZATION



RESULTS

Dense to Rock	Dense to Sparse	Sparse to Rock	Sparse to Dense	Rock to Sparse	Rock to Dense	Rock to Water	Water to Rock	No Change	Others

- For example, “Rock to Water” indicates the land cover feature changed from sand or rock to water body in the past year.
- No change indicates there are no detected changes in the past year.
- Others indicate those parts in the images with no data or with erroneous data.
- can monitor the new branch of Suez Canal being built and filled with water.

PERFORMANCE TEST ENVIRONMENT

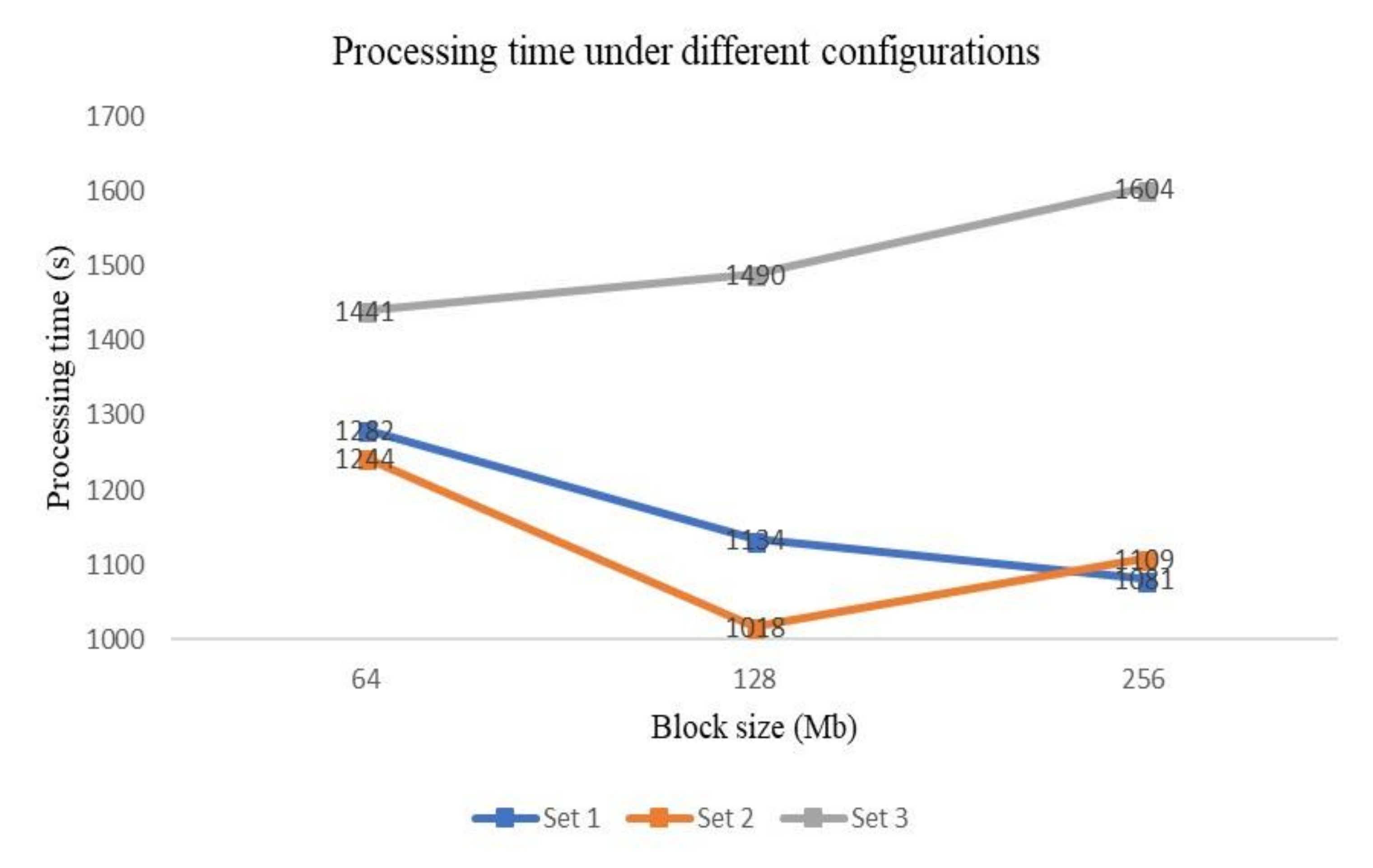
- Env 1
  - 2 master nodes with 2 Intel Xeon E5-2680v4 2.4GHz CPUs and 256 G memory each.
  - 18 computing nodes are equipped with 2 Intel Xeon E5-2690v4 2.6GHz CPUs (28 cores) and 256 G memory each.
  - 10Gbit network is assigned.
  - In total, there are 1008 computing vcores and 5.12 Tb memory available.
  - 2 Pb HDFS is configured and the block size is 128 M as default.

- Env 2
  - 1 t2.xlarge instance with 4 vCPUs Intel Broadwell E5-2686v4 2.3 GHz as master node
  - 2 t2.large instances with 2 vCPUs Intel Broadwell E5-2686v4 2.3 GHz as slaves.
  - The total number of vcores is 8 and the overall memory is 32 G.
  - 20 G storage per node are attached with default 128 M block size.

RESULTS

	Scenes for each year	Size	Processing time
Small	~0.15	~1.2 G	67 s
Medium	~1	~10 G	276 s
Large	~15	~107.4 G	1018 s
MODIS	~0.5	~0.25 G	54 s

RESULTS



- 10 executors with 100 G memory and 100 cores each as set 1
- 50 executors with 20 G memory and 20 cores each as set 2
- 500 executors with 2 G memory and 2 cores as set 3

CONCLUSION

- A prototypical framework to implement big remote sensing imagery classification and change detection on cloud
- Flexibility for processing big remote sensing datasets in multi-spatial, multispectral, and multi-temporal cases
- Shifting between resolutions and spectrums is possible with slight adjustment, thereby significantly saving the time cost of reprogramming brand new toolkits for different purposes.
- Possible to exploit the benefits of cloud platforms to gain (theoretically) unlimited computing resources
- Highly accessible to multisource data storage, even in the cloud, which is useful in reducing data transformation costs

THANK YOU!

Q&A