SATELLITE IMAGE ORTHORECTIFICATION OVER HDFS USING SPARK

IRT, OCE PROJECT
• Mathias ORTNER <mathias.ortner@irt-saintexupery.com>
• Gregory FLANDIN <gregory.flandin@irt-saintexupery.com>
• Marc SPIGAI <marc.spigai@irt-saintexupery.com>
BIG DATA?

• store, process, analyse **large** volumes of data
• maximal volume not know a priori
• cost = \( c \times \text{Volume} \)
• nature of data or processing may change
SATELLITE IMAGE GROUND SEGMENT

We have / expect :

- large volume of images to store
- large volume of images to explore
- new competitors from “big data world”
OUR WORK!

- well known case: orthorectification of SPOT 6 images (L1 to L2)
- study HDFS + spark over google cloud
- which advantages / drawbacks
- how does it fit with usual High Performance approaches
WHAT IS HADOOP?
IN SHORT

- **Distributed data storage**
  1. store large volumes
  2. ... over low cost hardware
- scalable
- MapReduce
- Not Posix
DATA REPLICATION

- Data is stored on several computers (a cluster)
- Data is split into pieces (typically 64M Bytes)
- HDFS is not POSIX (set of commands)
- One namenode, several datanodes
- FS aware of data locality
AND DISTRIBUTE
MAPREDUCE (MAP)

Files (data collections)

Map operation
- □ ➔ □
- □ ➔ □
- □ ➔ □
- □ ➔ □

Y = mean(X)
MAPREDUCE (REDUCE)

Reduce operation
(★,★) ➔ ★
(★,★) ➔ ★
(★,★) ➔ ★
X = max(X,X)

Result:
★
★
★
EXAMPLE (LS)

[ortner@cluster-data-master ~]$ hdfs dfs -ls /
Found 2 items
-rw--r--r--  2 ortner supergroup 1030588144 2015-04-28 19:09 /brisbane1-T400
EXAMPLE (PUT)

[ortner@cluster-data-master ~]$ hdfs dfs -put temp/brisbane1-T1000-BB.avro
[ortner@cluster-data-master ~]$ hdfs dfs -ls /

Found 2 items
-rw-r--r--  2 ortner supergroup  281398027 2015-04-30 14:57 /brisbane1-T1000-BB.avro
-rw-r--r--  2 ortner supergroup 1030588144 2015-04-28 19:09 /brisbane1-T4000-BP.avro
EXAMPLE (STATUS OF FILE)

```
[ortner@cluster-data-master ~]$ hdfs fsck /brisbane1-T4000-BP.avro -blocks
Connecting to namenode via http://cluster-data-master:50070
FSCK started by ortner (auth:SIMPLe) from /10.240.11.148 for path /brisbane1-T4000-BP.avro 1030588144 bytes, 8 block(s): OK
0. BP-1139583743-10.240.69.224-1430248096224:blk_1073741825_1001 len=134217728 repl=2
1. BP-1139583743-10.240.69.224-1430248096224:blk_1073741826_1002 len=134217728 repl=2
2. BP-1139583743-10.240.69.224-1430248096224:blk_1073741827_1003 len=134217728 repl=2
3. BP-1139583743-10.240.69.224-1430248096224:blk_1073741828_1004 len=134217728 repl=2
4. BP-1139583743-10.240.69.224-1430248096224:blk_1073741829_1005 len=134217728 repl=2
5. BP-1139583743-10.240.69.224-1430248096224:blk_1073741830_1006 len=134217728 repl=2
6. BP-1139583743-10.240.69.224-1430248096224:blk_1073741831_1007 len=134217728 repl=2
7. BP-1139583743-10.240.69.224-1430248096224:blk_1073741832_1008 len=91064048 repl=2

Status: HEALTHY
Total size:  1030588144  B
Total dirs:  0
Total files:  1
Total symlinks:  0
Total blocks (validated):  8  (avg. block size 128823518  B)
Minimally replicated blocks:  8  (100.0 %)
Over-replicated blocks:  0  (0.0 %)
Under-replicated blocks:  0  (0.0 %)
Mis-replicated blocks:  0  (0.0 %)
```
FSCK ended at Thu Apr 30 15:04:22 UTC 2015 in 1 milliseconds

The filesystem under path '/brisbane1-T4000-BP.avro' is HEALTHY
WHAT IS ORTHORECTIFICATION?
ON GROUND PROJECTION

- image is acquired in sensor geometry (L1)
- but the user needs a projection on ground (L2)
- Projection accounts for:
  - Time measurements
  - Line of sight calibration
  - Satellite attitude measurements
  - Satellite orbit measurements
  - Terrain model
SPOT6 SPECIFICITY: MULTIPLE SENSORS

The system swath is 60km, native resolution is 2.2m.

- two cameras (1 and 2),
- and in each camera, two retinas (A and B),
- in each retina, 5 bands (1 Pan and 4 Multispectral),
- Panchromatic band: 7000 pixels, and in the Multispectral bands 1500 pixels.

The L1 product is therefore actually made of 20 images.
INPUT IS L1

We start from a Level 1 image which is the native image in the focal plane geometry
We produce a level 2 image, i.e. an image that is projected on the ground using a Digital Terrain model and a cartographic frame.
IN BETWEEN :

The production relies on four steps :

- Dtm interpolation
- Inverse localization (from ground to focal plane)
- Image interpolation (pixel lookup, B spline resampler)
- Fusion for overlapping parts
TERRAIN MODEL IS A SHARED RESOURCE (SRTM)

SRTM terrain model is a standard, freely available, and widely used terrain model.
LARGE VOLUMES OF DATA TO BE HANDLED

We focus in this study on a full PAN image production, with typical dimensions of

- 44754 x 49135 pixels (1X)
- 110858 x 81327 pixels (4X)
- 352829 x 42817 pixels (7X)
WHAT IS SPARK?
APACHE PROJECT

Use distributed computation and map reduce easily

It is written in

- akka, on top of
- scala, on top of
- java

It has bindings in python, scala, java.
RESILIENT DISTRIBUTED DATASETS (RDDs)

All data is stored in *collections* of objects called *resilient distributed datasets* (RDDs).

Collections are distributed on the network.
ACTIONS ON RDDS

We have different possible actions:

- Creation
- Mapping (transform a RDD into another one)
- Reductions
- Writing, collection
LAZY EVALUATION

Spark is a functional language

- Define rules...
- ... only needed rules are applied
PAY (MODERATELY) FOR (HIGH) USAGE!
WHAT IS IT?

Create and use virtual machines....
EXAMPLE: CREATE A MACHINE!

```bash
#!/bin/bash

gcloud compute instances create cluster-data-master
  --image centos7-image-java-xvfb
  --disk name=data-disk device-name=sdb mode=rw
  --local-ssd interface=SCSI
  --metadata-from-file startup-script=startup.sh
  --machine-type n1-highmem-16

Created [https://www.googleapis.com/compute/v1/projects/XXXX/zone=europe-west1-b/instances/

<table>
<thead>
<tr>
<th>NAME</th>
<th>ZONE</th>
<th>MACHINE_TYPE</th>
<th>INTERNAL_IP</th>
<th>EXTERNAL_IP</th>
<th>STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>cluster</td>
<td>data-master</td>
<td>europe-west1-b</td>
<td>10.240.5.82</td>
<td>104.155.0.44</td>
<td>RUNNING</td>
</tr>
</tbody>
</table>
```
## How Much Does It Cost?

<table>
<thead>
<tr>
<th>Machine Type</th>
<th>Virtual CPUs</th>
<th>Memory</th>
<th>Typical Price (USD) per Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>n1-standard-1</td>
<td>1</td>
<td>3.75GB</td>
<td>$0.038</td>
</tr>
<tr>
<td>n1-standard-2</td>
<td>2</td>
<td>7.5GB</td>
<td>$0.076</td>
</tr>
<tr>
<td>n1-standard-4</td>
<td>4</td>
<td>15GB</td>
<td>$0.152</td>
</tr>
<tr>
<td>n1-standard-8</td>
<td>8</td>
<td>30GB</td>
<td>$0.304</td>
</tr>
<tr>
<td>n1-standard-16</td>
<td>16</td>
<td>60GB</td>
<td>$0.608</td>
</tr>
<tr>
<td>n1-highmem-2</td>
<td>2</td>
<td>13GB</td>
<td>$0.096</td>
</tr>
<tr>
<td>n1-highmem-4</td>
<td>4</td>
<td>26GB</td>
<td>$0.192</td>
</tr>
<tr>
<td>n1-highmem-8</td>
<td>8</td>
<td>52GB</td>
<td>$0.384</td>
</tr>
<tr>
<td>n1-highmem-16</td>
<td>16</td>
<td>104GB</td>
<td>$0.768</td>
</tr>
</tbody>
</table>
Distribution of algorithm *incredibly easy to write*
WHY USE HDFS / MAPREDUCE?
FIRST AND MAIN RESULT

- input is on HDFS
- output is on HDFS
- production is made using Spark
PRODUCE 4 UNITS ON 4 SLAVES...
... OR 24 UNITS ON 12 SLAVES ...
... 36 UNITS ON 18 SLAVES ...
OR 48 ON 24 SLAVES ...
IN SAME AMOUNT OF TIME!
INCREASE PRODUCTION BY ADDING CORES

Mpix / sec function of nb Cores

0  50  100  150  200  250  300  350  400
0  50  100  150  200  250  200  150  100
WHAT WE HAVE DONE:

- Analysis of *High Performance mono computer* orthorectification algorithm
- Full implementation of *scalable* orthorectification algorithm
- Deployment on *google cloud engine*
- Analysis of *distribution performances*