



Use OpenCellID Data To Estimate GSM/UMTS/LTE Coverage

Practical Exam



Goal

OpenCellID.com provides regular exports of worldwide cell data:

- <https://www.opencellid.org>
- Latest Full Dump and Diffs: <https://www.opencellid.org/downloads.php>

Full Database

- [cell_towers.csv.gz](#)
Updated: 2018-10-04 (907MB)

Differential

- [OCID-diff-cell-export-2018-10-04-T000000.csv.gz](#) (1385KB)
- [OCID-diff-cell-export-2018-10-03-T000000.csv.gz](#) (2350KB)
- [OCID-diff-cell-export-2018-10-02-T000000.csv.gz](#) (2293KB)
- [OCID-diff-cell-export-2018-10-01-T000000.csv.gz](#) (969KB)
- [OCID-diff-cell-export-2018-09-30-T000000.csv.gz](#) (1306KB)
- [OCID-diff-cell-export-2018-09-29-T000000.csv.gz](#) (2279KB)
- [OCID-diff-cell-export-2018-09-28-T000000.csv.gz](#) (2378KB)

```
radio,mcc,net,area,cell,unit,lon,lat,range,samples,changeable,created,updated,averageSignal
UMTS,262,2,801,86355,0,13.285512,52.522202,1000,7,1,1282569574,1300155341,0
GSM,262,2,801,1795,0,13.276907,52.525714,5716,9,1,1282569574,1300155341,0
GSM,262,2,801,1794,0,13.285064,52.524,6280,13,1,1282569574,1300796207,0
UMTS,262,2,801,211250,0,13.285446,52.521744,1000,3,1,1282569574,1299466955,0
UMTS,262,2,801,86353,0,13.293457,52.521515,1000,2,1,1282569574,1291380444,0
UMTS,262,2,801,86357,0,13.289106,52.53273,2400,3,1,1282569574,1298860769,0
UMTS,262,3,1107,83603,0,13.349675,52.497575,3102,222,1,1282672189,1300710809,0
GSM,262,2,776,867,0,13.349711,52.497367,1000,214,1,1282672189,1301575206,0
GSM,262,3,1107,13971,0,13.349743,52.497437,1000,212,1,1282672189,1300710809,0
GSM,262,3,1107,355,0,13.34963,52.497378,1000,198,1,1282672189,1300710809,0
UMTS,262,3,1107,329299,0,13.349223,52.497519,3041,186,1,1282672189,1299860879,0
```

cell_towers.csv



Goal

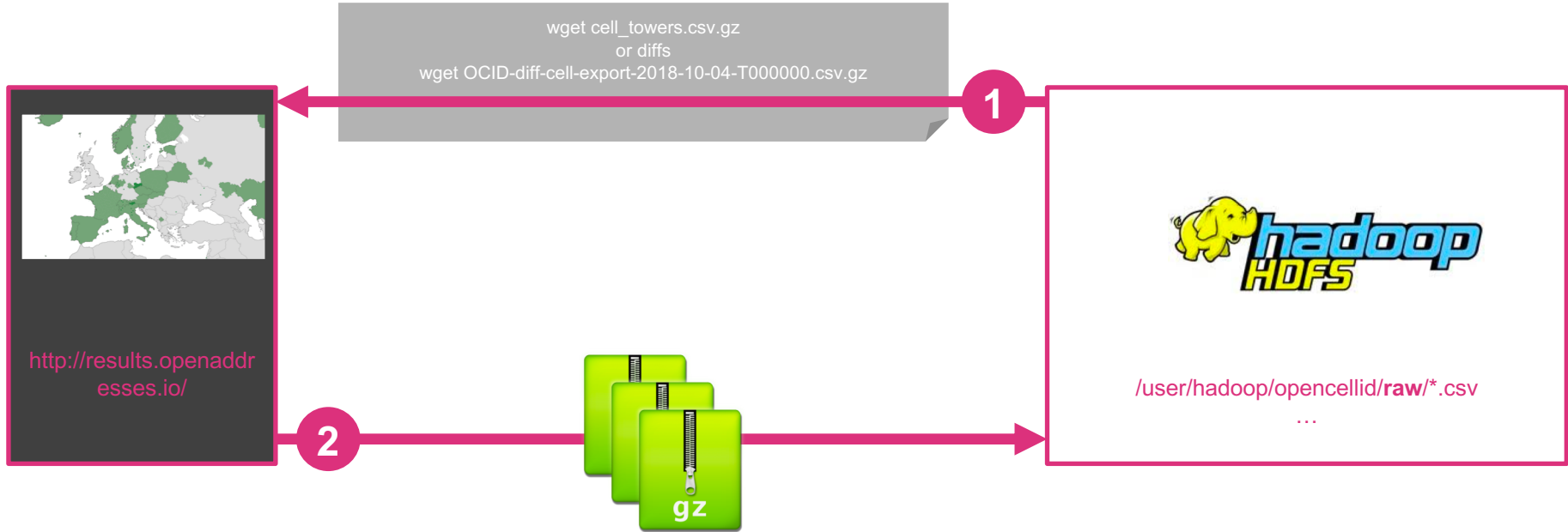
We want to make use of this data to estimate the coverage of GSM, UMTS and LTE for a certain place (*latitude, longitude*).

Workflow:

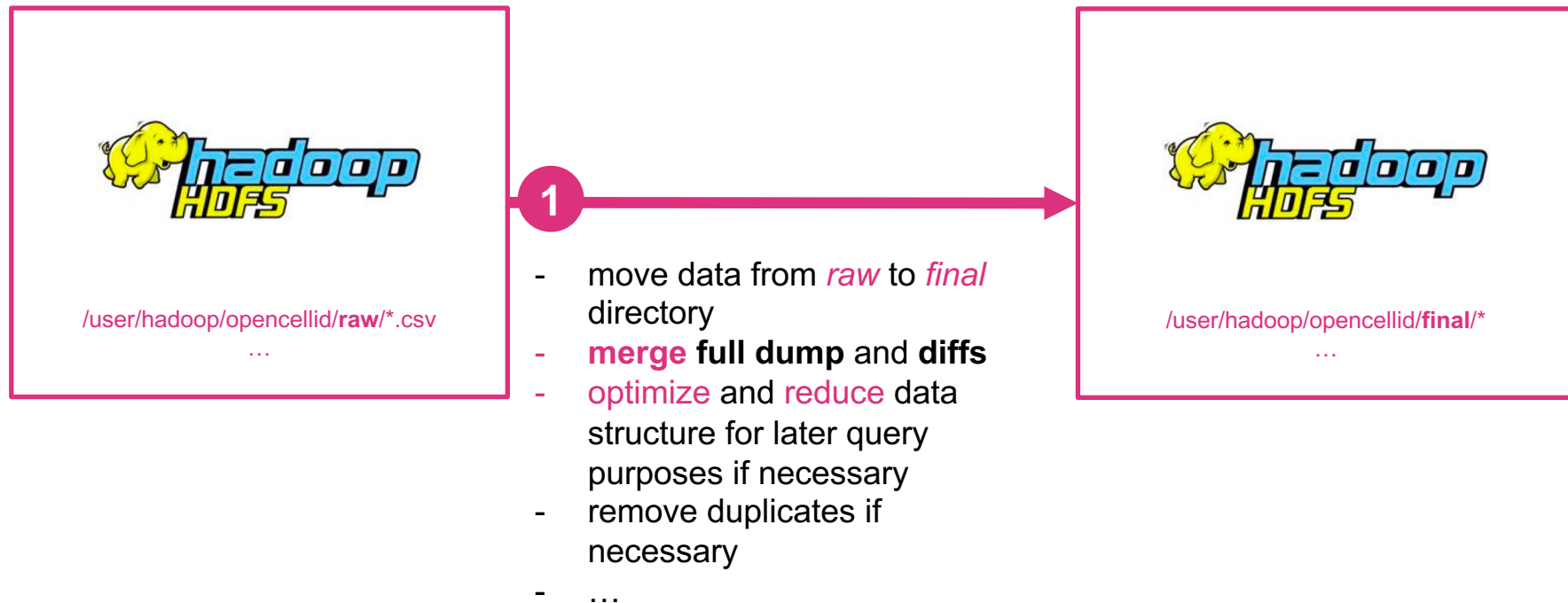
- **Gather data** from OpenCellID.com
- **Save raw data** (CSV files) to HDFS (partitioned by radio, e.g. GSM, UMTS, LTE...)
- **Optimize, reduce and clean raw data** and save it to **final** directory on HDFS
- **Export** address data to **end-user database** (e.g. MySQL, MongoDB...)
- Provide a simple **HTML Frontend** which is able to:
 - read from end-user database
 - process user input (Latitude, Longitude...)
 - checks against OpenCellID data in end-user database
 - Display result (GSM, LTE and UMTS coverage)
- The whole data workflow **must be implemented** within an ETL **workflow tool** (e.g. Pentaho Data Integration or Airflow) and **run automatically**

Latitude:	<input type="text" value="52.254409790039"/>	Result:
Longitude:	<input type="text" value="13.436965942383"/>	GSM: very good
		UMTS: good
		LTE: poor
<input type="button" value="Check Cell Coverage!"/>		

Dataflow: 1. Get Cell Data



Dataflow: 2. Raw To Final Transfer



Dataflow: 3. Enhance Data And Save Results



/user/hadoop/opencellid/final/*

...

1



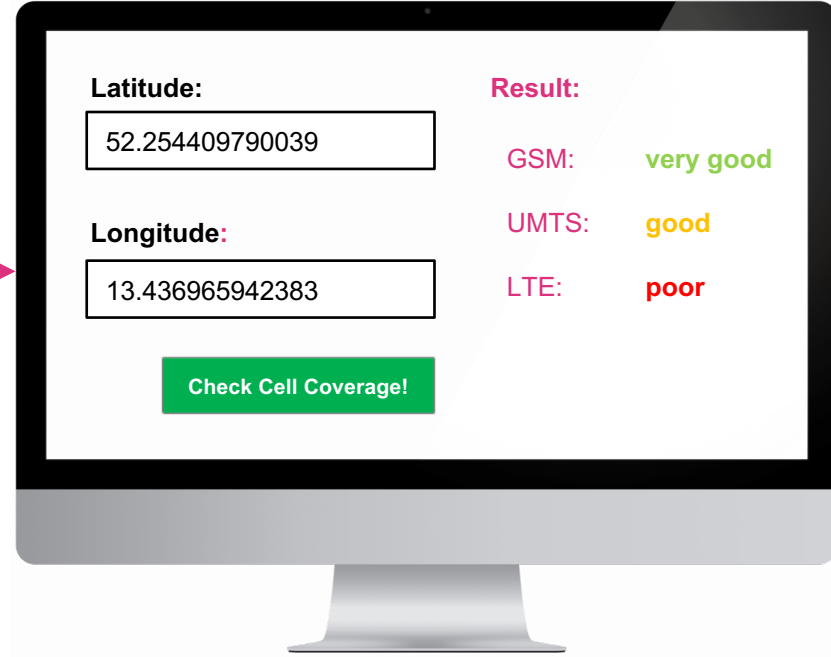
- **enhance data** (e.g. for later querying)
- use *Hive*, *Python*, *Spark* or *PySpark*
- **save** everything to a end-user database (e.g. *MySQL*, *MongoDB*)



Dataflow: 4. Provide Simple Web Interface



1



- Provide a simple **HTML Frontend** which is able to:
 - read from end-user database
 - process user input (Latitude, Longitude...)
 - checks against OpenCellID data in end-user database
 - Display result (GSM, LTE and UMTS coverage)