Storage and Data Recovery

Exercise 4: Map reduce

Master's Degree in Intelligent Systems
Iván López Muñoz

What is map reduce?

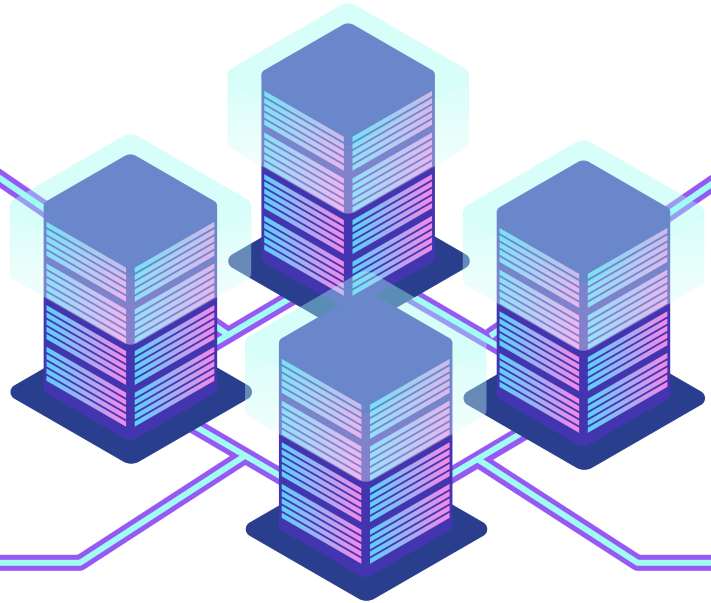
Map-reduce is a data processing paradigm for condensing large volumes of data into useful aggregated results

1 - Splitting

Split large volume of data of a dataset into differents small datasets

2 - Mapping

Assign a value for every key of the data source



3 - Shuffling

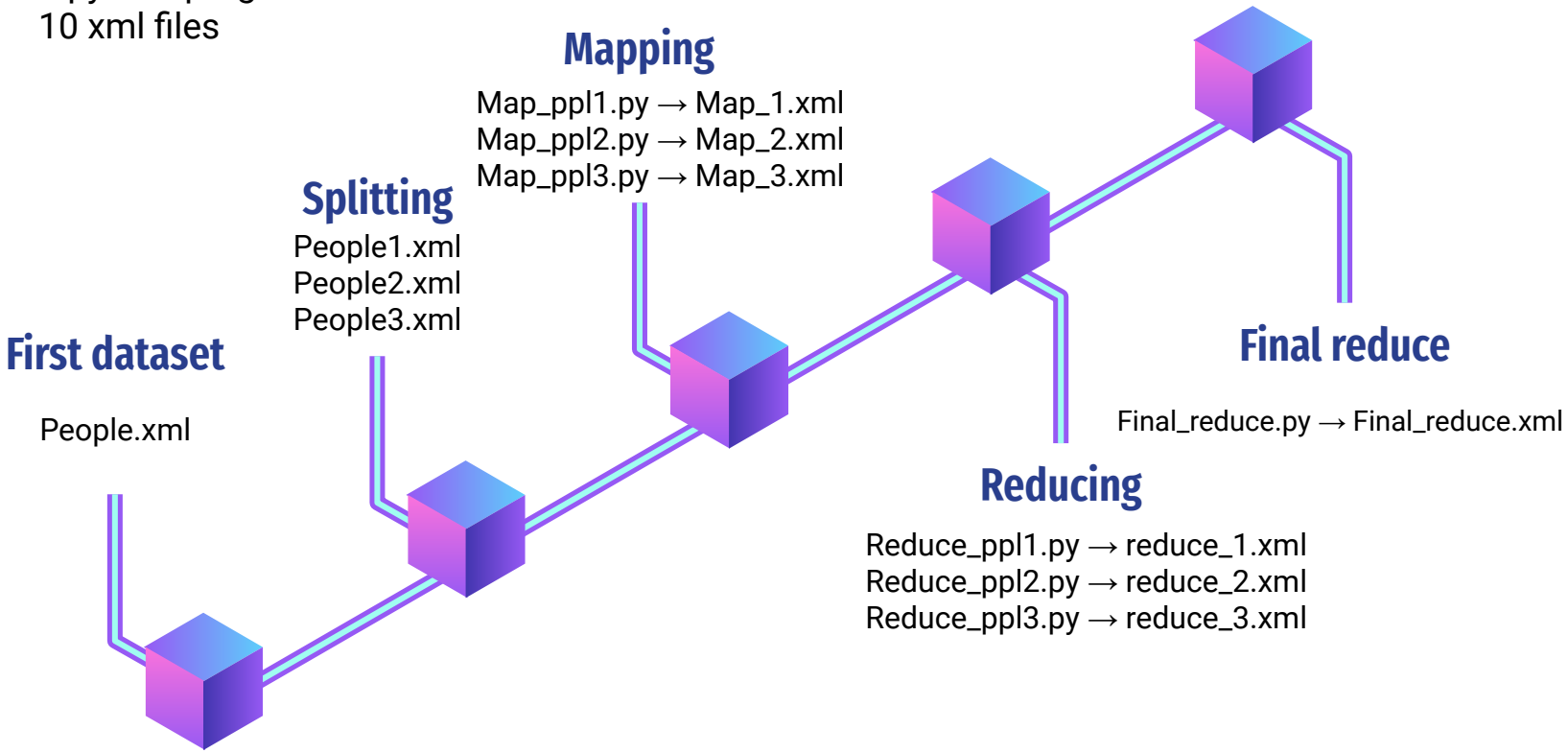
Join information with the same key to reduce data volume

4 - Reducing

Join all the information of every key. The reduced output is the final result.

File summary

- 7 python programs
- 10 xml files



Splitting

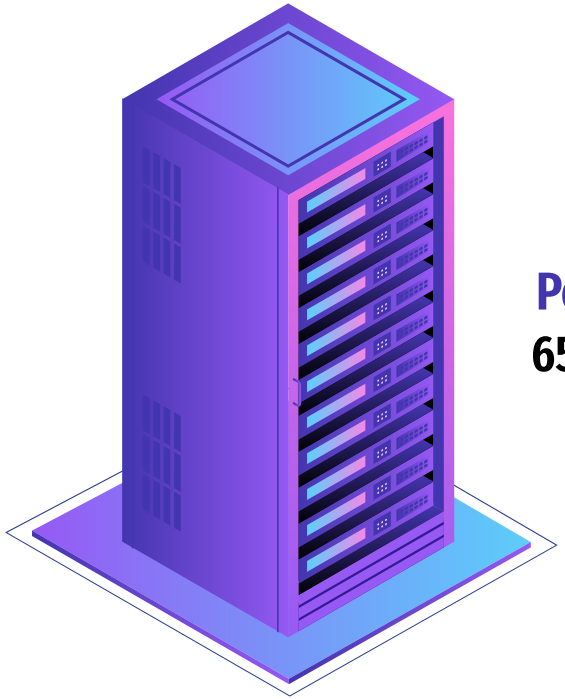
First, we need to split our dataset into smaller datasets. Here we have our People.xml dataset with 654 persons

```
<person><id>1</id><sex>H</sex><name>Juan</name><surname1>Prats</surname1><surname2>Ramis</surname2><birthdate>1816-1-1</birthdate></person>
<person><id>2</id><sex>D</sex><name>Antonia</name><surname1>Ferregut</surname1><surname2>Jaume</surname2><birthdate>1816-1-1</birthdate></person>
<person><id>3</id><sex>D</sex><name>Juana</name><surname1>Perello</surname1><surname2>Figuerola</surname2><birthdate>1816-1-6</birthdate></person>
<person><id>4</id><sex>H</sex><name>Jaime</name><surname1>Pujades</surname1><surname2>Tortella</surname2><birthdate>1816-1-7</birthdate></person>
<person><id>5</id><sex>D</sex><name>Francisca</name><surname1>Garriga</surname1><surname2>Bauza</surname2><birthdate>1816-1-8</birthdate></person>
<person><id>6</id><sex>D</sex><name>Geronima</name><surname1>Coll</surname1><surname2>Pujades</surname2><birthdate>1816-1-9</birthdate></person>
<person><id>7</id><sex>H</sex><name>Gabriel</name><surname1>Estrañy</surname1><surname2>Garau</surname2><birthdate>1816-1-9</birthdate></person>
<person><id>8</id><sex>H</sex><name>Arnaldo</name><surname1>Mulet</surname1><surname2>Perello</surname2><birthdate>1816-1-9</birthdate></person>
<person><id>9</id><sex>H</sex><name>Gabriel</name><surname1>Llinas</surname1><surname2>Alomar</surname2><birthdate>1816-1-10</birthdate></person>
<person><id>10</id><sex>D</sex><name>Margharita</name><surname1>Domenech</surname1><surname2>Pujades</surname2><birthdate>1816-1-10</birthdate></person>
<person><id>11</id><sex>H</sex><name>Pedro</name><surname1>Ferregut</surname1><surname2>Rayo</surname2><birthdate>1816-1-11</birthdate></person>
<person><id>12</id><sex>D</sex><name>Margarita</name><surname1>Genestra</surname1><surname2>Estrañy</surname2><birthdate>1816-1-11</birthdate></person>
<person><id>13</id><sex>D</sex><name>Antonia</name><surname1>Pujades</surname1><surname2>Gual</surname2><birthdate>1816-1-12</birthdate></person>
<person><id>14</id><sex>D</sex><name>Catalina</name><surname1>Llompard</surname1><surname2>Campins</surname2><birthdate>1816-1-12</birthdate></person>
<person><id>15</id><sex>D</sex><name>Juana</name><surname1>Munar</surname1><surname2>Siquier</surname2><birthdate>1816-1-12</birthdate></person>
```

⋮

```
<person><id>639</id><sex>H</sex><name>Pere</name><surname1>Mulet</surname1><surname2>Perello</surname2><birthdate>1820-11-24</birthdate></person>
<person><id>640</id><sex>D</sex><name>Margalida</name><surname1>Bertran</surname1><surname2>Compañy</surname2><birthdate>1820-11-30</birthdate></person>
<person><id>641</id><sex>D</sex><name>Margarita</name><surname1>Gual</surname1><surname2>Vicens</surname2><birthdate>1820-12-2</birthdate></person>
<person><id>642</id><sex>H</sex><name>Miguel</name><surname1>Ferregut</surname1><surname2>Rayo</surname2><birthdate>1820-12-3</birthdate></person>
<person><id>643</id><sex>D</sex><name>Margarita</name><surname1>Pujades</surname1><surname2>Ferregut</surname2><birthdate>1820-12-4</birthdate></person>
<person><id>644</id><sex>D</sex><name>Francisca</name><surname1>Martorell</surname1><surname2>Martorell</surname2><birthdate>1820-12-4</birthdate></person>
<person><id>645</id><sex>D</sex><name>Leocadia</name><surname1>Desconegut</surname1><surname2>Desconegut</surname2><birthdate>1820-12-9</birthdate></person>
<person><id>646</id><sex>H</sex><name>Andres</name><surname1>Aguilo</surname1><surname2>Forteza</surname2><birthdate>1820-12-16</birthdate></person>
<person><id>647</id><sex>D</sex><name>Francisca</name><surname1>Bestard</surname1><surname2>Rossello</surname2><birthdate>1820-12-18</birthdate></person>
<person><id>648</id><sex>H</sex><name>Onofre</name><surname1>Llompard</surname1><surname2>Rubi</surname2><birthdate>1820-12-19</birthdate></person>
<person><id>649</id><sex>H</sex><name>Gabriel</name><surname1>Coll</surname1><surname2>Fiol</surname2><birthdate>1820-12-20</birthdate></person>
<person><id>650</id><sex>D</sex><name>Juana</name><surname1>Ferrer</surname1><surname2>Tortella</surname2><birthdate>1820-12-21</birthdate></person>
<person><id>651</id><sex>H</sex><name>Martin</name><surname1>Coll</surname1><surname2>Ferrer</surname2><birthdate>1820-12-28</birthdate></person>
<person><id>652</id><sex>H</sex><name>Gabriel</name><surname1>Matheu</surname1><surname2>Figuerola</surname2><birthdate>1820-12-28</birthdate></person>
<person><id>653</id><sex>D</sex><name>Juana</name><surname1>Vallespir</surname1><surname2>Garau</surname2><birthdate>1820-12-29</birthdate></person>
<person><id>654</id><sex>H</sex><name>Antonio</name><surname1>Martorell</surname1><surname2>Llobera</surname2><birthdate>1820-12-31</birthdate></person>
```

Splitting



People.xml
654 persons

People1.xml
200 persons

People2.xml
250 persons

People3.xml
204 persons

Mapping


People1.xml
200 persons


People2.xml
250 persons


People3.xml
204 persons


Map_1.xml
200 persons
Juan: 1
Antonia: 1
Juana: 1
etc


Map_2.xml
250 persons
Francisco: 1
Miguel: 1
Juan: 1
etc


Map_3.xml
204 persons
Sebastian: 1
Margarita: 1
Cathalina: 1
etc

How to Map with Python

Map_ppl1.py

```python
# -*- coding: utf-8 -*-
"""
Created on Wed Oct 18 11:57:25 2023

@author: Ivan
"""

import xml.etree.ElementTree as ET

# Create an empty map element
map_element = ET.Element("map")

# Specify the input XML file
input_file = 'people1.xml'

# Parse the input XML file
tree = ET.parse(input_file)
root = tree.getroot()

# Iterate over the 'name' elements and create item elements
for person in root:
    name = person.find('name').text

    # Create an item element
    item_element = ET.SubElement(map_element, "item")

    # Add the 'name' and 'value' elements to the item
    name_element = ET.SubElement(item_element, "name")
    name_element.text = name

    value_element = ET.SubElement(item_element, "value")
    value_element.text = "1"

# Create an ElementTree object with the map element
result_tree = ET.ElementTree(map_element)

# Write the results to an XML file
result_tree.write("Map_1.xml", encoding="utf-8")
```

- Read each smaller dataset splitted previously
- Assign a value for every key of the data source, in this case for every name in the dataset
- From each ‘people’ dataset we obtain a list of names with a value ready to shuffle

Map_1.xml

```xml
<map>
    <item>
        <name>Juan</name>
        <value>1</value>
    </item>
    <item>
        <name>Antonia</name>
        <value>1</value>
    </item>
    <item>
        <name>Juana</name>
        <value>1</value>
    </item>
    <item>
        <name>Jaime</name>
        <value>1</value>
    </item>
    <item>
```

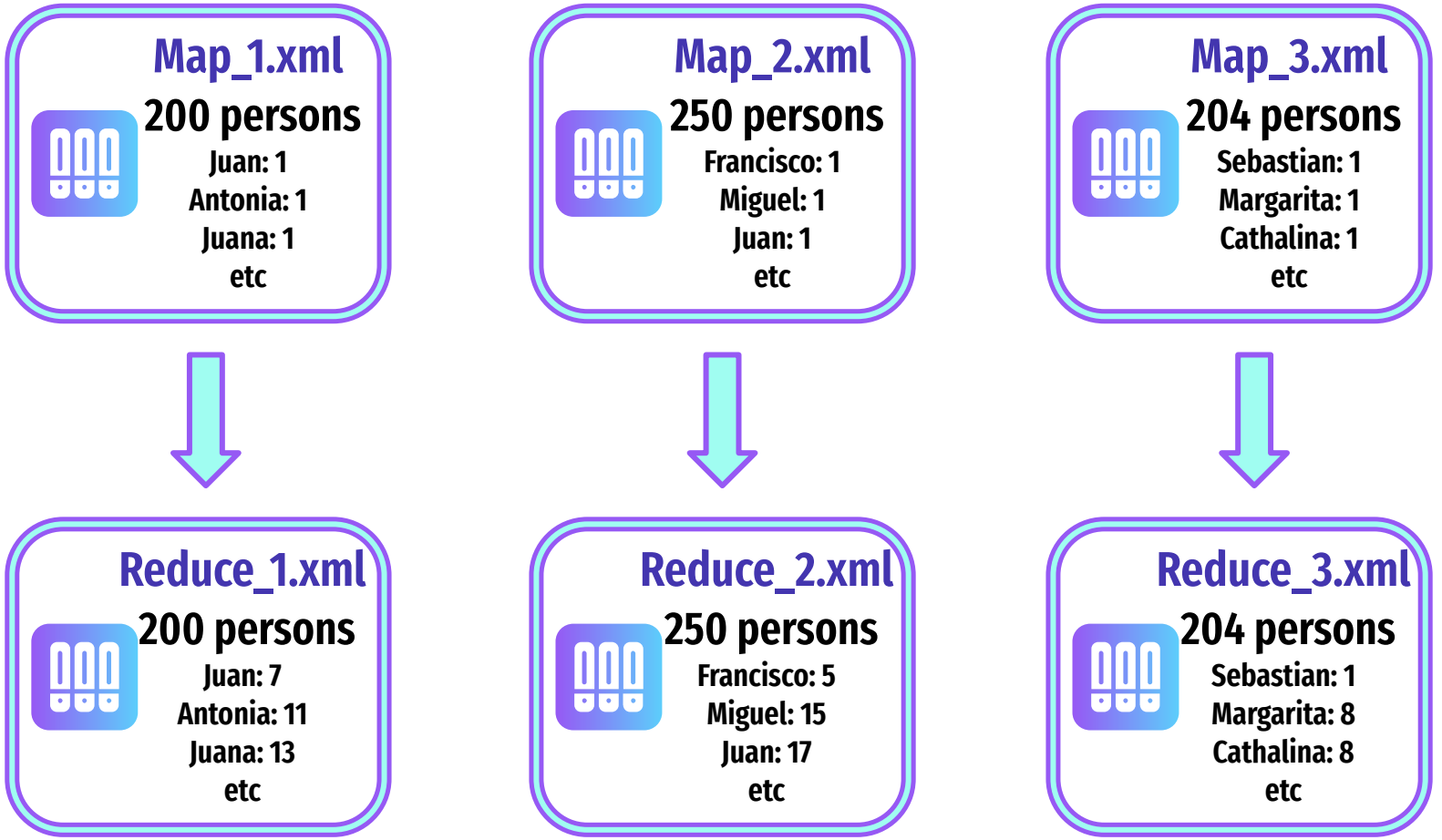
Map_2.xml

```xml
<map>
    <item>
        <name>Sebastian</name>
        <value>1</value>
    </item>
    <item>
        <name>Margarita</name>
        <value>1</value>
    </item>
    <item>
        <name>Cathalina</name>
        <value>1</value>
    </item>
    <item>
        <name>Ana</name>
        <value>1</value>
    </item>
```

Map_3.xml

```xml
<map>
    <item>
        <name>Francisco</name>
        <value>1</value>
    </item>
    <item>
        <name>Miguel</name>
        <value>1</value>
    </item>
    <item>
        <name>Juan</name>
        <value>1</value>
    </item>
    <item>
        <name>Guillermo</name>
        <value>1</value>
    </item>
```

Reducing


Map_1.xml
200 persons
Juan: 1
Antonia: 1
Juana: 1
etc
Map_2.xml
250 persons
Francisco: 1
Miguel: 1
Juan: 1
etc
Map_3.xml
204 persons
Sebastian: 1
Margarita: 1
Cathalina: 1
etc
Reduce_1.xml
200 persons
Juan: 7
Antonia: 11
Juana: 13
etc
Reduce_2.xml
250 persons
Francisco: 5
Miguel: 15
Juan: 17
etc
Reduce_3.xml
204 persons
Sebastian: 1
Margarita: 8
Cathalina: 8
etc

How to Reduce with Python

Reduce_ppl1.py

```python
"""
Created on Wed Oct 18 12:04:38 2023

@author: Ivan
"""

import xml.etree.ElementTree as ET
from collections import defaultdict

# Create a defaultdict to store name counts
name_count = defaultdict(int)

# Parse the input XML file
tree = ET.parse("Map_1.xml")
root = tree.getroot()

# Iterate over the 'item' elements and
# extract names and values
for item in root.findall("item"):
    name = item.find("name").text
    value = int(item.find("value").text)

    # Increment the count for each name
    name_count[name] += value

# Create a new map element for the results
result_map = ET.Element("reduce")

# Iterate over the unique names and their counts
for name, count in name_count.items():
    # Create an item element
    item_element = ET.Element("item")

    # Add the 'name' and 'value' elements to the item
    name_element = ET.Element("name")
    name_element.text = name
    item_element.append(name_element)

    value_element = ET.Element("value")
    value_element.text = str(count)
    item_element.append(value_element)

    # Add the item to the result map
    result_map.append(item_element)

# Create an ElementTree object with the result map
result_tree = ET.ElementTree(result_map)

# Write the results to a new XML file
result_tree.write("reduce_1.xml", encoding="utf-8")
```

- Read each smaller dataset splitted and mapped previously
- Iterate on each person's name and add the values for each one
- From each ‘mapped’ dataset we obtain a list of names with the number of times they appear in each dataset

reduce_1.xml

```xml
<reduce>
    <item>
        <name>Juan</name>
        <value>7</value>
    </item>
    <item>
        <name>Antonia</name>
        <value>11</value>
    </item>
    <item>
        <name>Juana</name>
        <value>13</value>
    </item>
    <item>
        <name>Jaime</name>
        <value>1</value>
    </item>
```

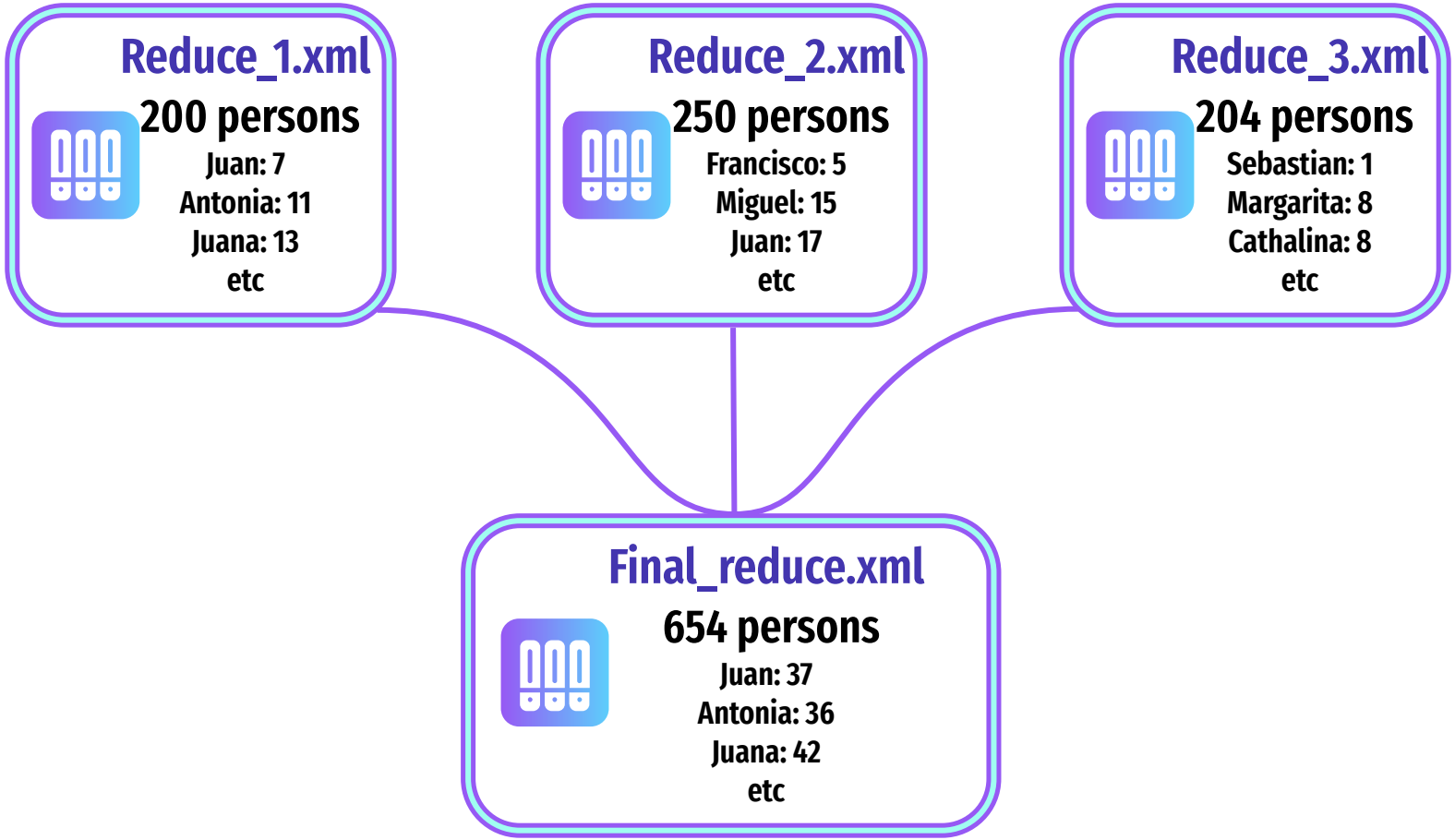
reduce_2.xml

```xml
<reduce>
    <item>
        <name>Francisco</name>
        <value>5</value>
    </item>
    <item>
        <name>Miguel</name>
        <value>15</value>
    </item>
    <item>
        <name>Juan</name>
        <value>17</value>
    </item>
    <item>
        <name>Guillermo</name>
        <value>5</value>
    </item>
```

reduce_3.xml

```xml
<reduce>
    <item>
        <name>Sebastian</name>
        <value>1</value>
    </item>
    <item>
        <name>Margarita</name>
        <value>8</value>
    </item>
    <item>
        <name>Cathalina</name>
        <value>8</value>
    </item>
    <item>
        <name>Ana</name>
        <value>5</value>
    </item>
```

Final reduce



Final reduce

Final_reduce.py

```python
"""
Created on Wed Oct 18 12:07:06 2023

@author: Ivan
"""

import xml.etree.ElementTree as ET
from collections import defaultdict

# Create a defaultdict to store name counts
name_count = defaultdict(int)

# Iterate over the XML files
for filename in ['reduce_1.xml','reduce_2.xml','reduce_3.xml']:
    tree = ET.parse(filename)
    root = tree.getroot()

    # Iterate over the 'name' elements and increment the count
    for person in root.findall("item"):
        name = person.find('name').text
        value = int(person.find("value").text)
        name_count[name] += value

# Create the XML result document
result_root = ET.Element("reduce")

# Add name count elements to the result document
for name, count in name_count.items():
    # Create an item element
    item_element = ET.SubElement(result_root, "item")

    # Add the 'name' and 'value' elements to the item
    name_element = ET.SubElement(item_element, "name")
    name_element.text = name

    value_element = ET.SubElement(item_element, "value")
    value_element.text = str(count)

# Create an ElementTree object and write it to an XML file
result_tree = ET.ElementTree(result_root)
result_tree.write("Final_reduce.xml", encoding="utf-8")
```

- Read all reduced datasets
- Iterate on each person's name in every reduced file and add the values for each one
- Finally, we obtain a list with the number of people who have the same name

```xml
<reduce>
    <item>
        <name>Juan</name>
        <value>37</value>
    </item>
    <item>
        <name>Antonia</name>
        <value>36</value>
    </item>
    <item>
        <name>Juana</name>
        <value>42</value>
    </item>
    <item>
        <name>Jaime</name>
        <value>2</value>
    </item>
```

Final_reduce.xml

Conclusions about map reducing

- **Scalability**: Map Reduce efficiently processes substantial volumes of data, making it suitable for handling the vast amount of information within these XML files.

- **Data Flexibility**: Map Reduce can work with structured, semi-structured, and unstructured data, making it a versatile choice for the diverse content found in the XML files.

- **Distributed Data Processing:** The ability to process data in a distributed manner helps minimize communication overhead and maximizes computational efficiency.

- **MapReduce Phases**: The Map Reduce process involves distinct phases like splitting, mapping, shuffling, and reducing, providing a structured approach to data processing that simplifies complex tasks.

  → By modifying the final program by counting the times it writes the names of people in the final xml file, from the data set of 654 people in People.xml, we can find 86 different names

  → That's why the shuffling part has not been done, since we would have 86 xml files, each of them with a name and the number of apparitions in the dataset