



# Trabajo Práctico Especial 2:

## **Peatones**

31 de Agosto de 2022

72.42 - Programación de Objetos Distribuidos

**Arce**, Julián Francisco - 60509

**Catalán**, Roberto José - 59174

**Dell'Isola**, Lucas - 58025

**Pecile**, Gian Luca - 59235

**Torrusio**, Lucía - 59489

## Decisiones de diseño e implementación de los componentes de cada trabajo MapReduce

En el presente trabajo de implementación práctica se busca probar la hipótesis que plantea la materia luego de haber tenido una primer experiencia previamente con threads, dicha hipótesis trata de demostrar que mientras más nodos se agregan, mejor performance se obtiene.

Se diseñó e implementó el uso de la clase *Tuple* para el manejo de pares de datos genéricos.

En ciertos casos, como en la query 4, donde solo se necesitan las lecturas de un año en específico, se realiza un filtrado de los datos antes de enviarlo a la red, para reducir considerablemente la cantidad de datos enviados.

Otra mejora con el fin de reducir el tráfico de red incluye el filtrado de los sensores activos, y a su vez, los readings que pertenecen a sensores activos (en el caso que la query lo requiera).

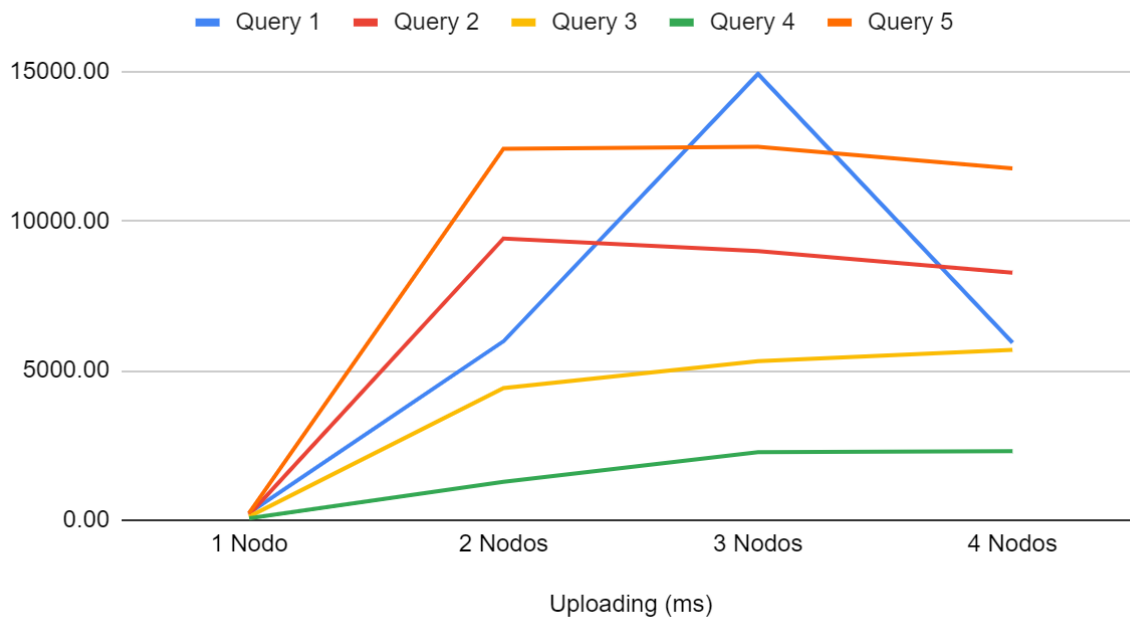
Suponiendo que tenemos una distribución uniforme de readings entre cada sensor, en el dataset propuesto, se cuenta con 80% de sensores activos, por lo que se estima que se enviaría un 20% menos de datos a la red.

## Análisis de tiempos de Queries

El análisis se realiza con un dataset de 100.000 líneas y hasta 4 nodos. Las queries fueron realizadas sin el uso de combiner y poseen los siguientes resultados:

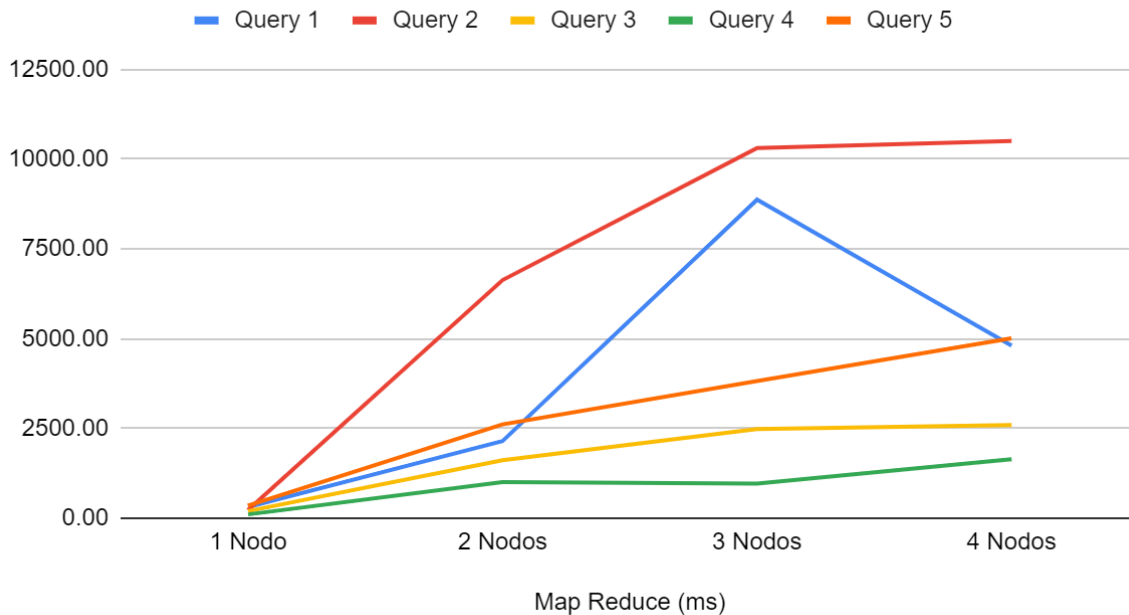
<b><i>Uploading (ms)</i></b>	1 Nodo	2 Nodos	3 Nodos	4 Nodos
Query 1	233.33	5985.67	14918.00	5936.33
Query 2	184.67	9416.00	8994.33	8269.00
Query 3	119.33	4415.00	5318.00	5696.33
Query 4	66.67	1281.67	2276.00	2306.33
Query 5	236.33	12415.67	12481.00	11758.00

### Uploading (ms)



<b>Map Reduce (ms)</b>	1 Nodo	2 Nodos	3 Nodos	4 Nodos
Query 1	312.00	2146.00	8877.00	4809.00
Query 2	229.33	6633.67	10311.00	10509.00
Query 3	197.00	1615.67	2484.33	2598.00
Query 4	106.00	1005.33	963.67	1643.67
Query 5	348.33	2613.67	3819.33	5010.67

### Map Reduce (ms)



Los parámetros usados son:

- Query 3:
  - Min = 500
- Query 4:
  - N = 3
  - año = 2019

## Potenciales puntos de mejora y/o expansión

Realizar optimizaciones para el manejo de subida de archivos por pedazos o *chunks* para aumentar la velocidad y mejorar la performance en sistemas con menores recursos donde al levantar todos los datos no dan abasto.

Otra optimización posible es realizar uso de una librería de streaming para la lectura de archivos en vez del uso de *Streams* de Java.

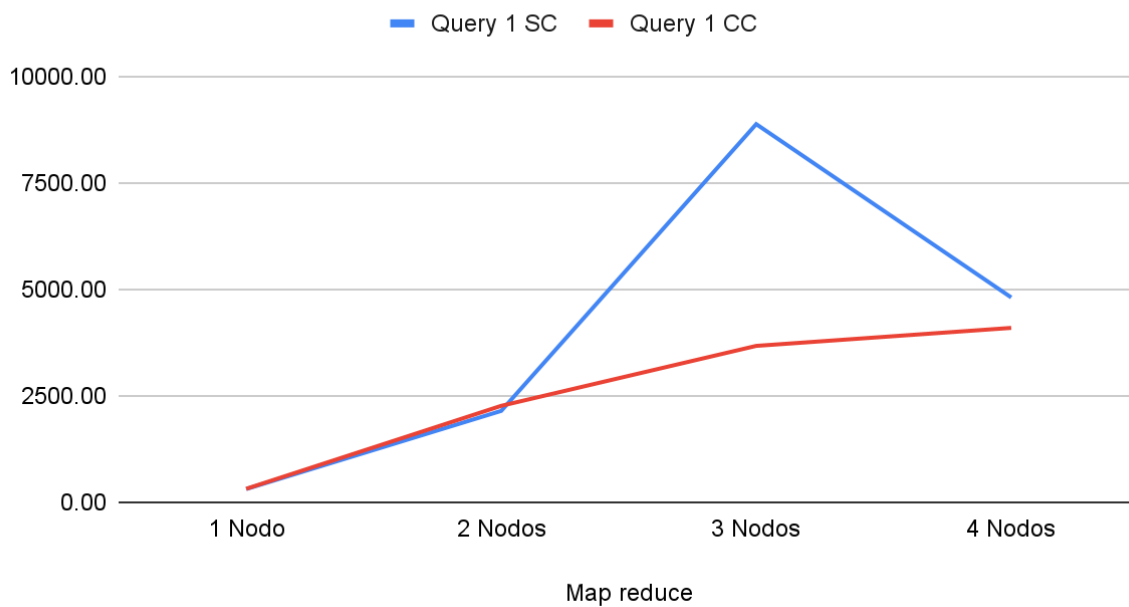
### Query a mejorar con *Combiner*

Se decidió analizar la query 1. Los resultados con combiner son los siguientes:

<b>Uploading (ms)</b>	1 Nodo	2 Nodos	3 Nodos	4 Nodos
Query 1	232.67	5942.33	12480.67	12020.67

<b>Map reduce (ms)</b>	1 Nodo	2 Nodos	3 Nodos	4 Nodos
Query 1	317.33	2265.00	3672.00	4094.33

### Map Reduce (ms)



Se puede observar que con el combiner la velocidad de procesamiento mejora.