HMusket

Corrector de secuencias mediante el espectro k-mer basado en Hadoop

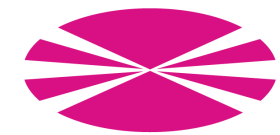
Luis Lorenzo Mosquera

luis.lorenzom@udc.es

Directores: Roberto Rey Expósito y Jorge González Domínguez

Trabajo Fin de Máster - Máster en bioinformática para las ciencias de la salud

28 de junio de 2018



UNIVERSIDADE DA CORUÑA

Índice I

Introducción

- Aparición de las técnicas NGS (Next Generation Sequence)
- Abaratamiento de la secuenciación
- Nuevas áreas de trabajo
- Aumento de tamaño de los conjuntos de datos a procesar
- Necesidad de corregir los datos
- Limitaciones de las soluciones actuales

Representación del ADN

```
>SEQUENCE_1
GATTTGGGGTTCGATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCGTTT
```

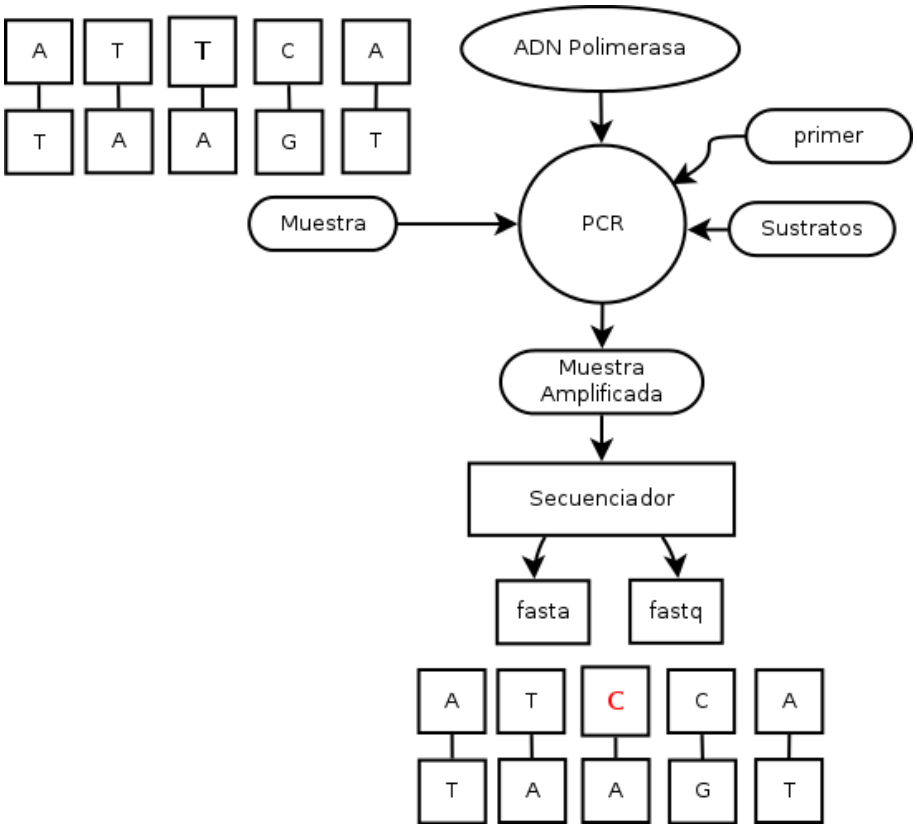
- Formato **fasta**
- Identificador de la secuencia
- Cadena que represente las bases
- Puede añadir opcionalmente un comentario junto con el identificador

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

- Formato **fastq**
- Identificador de la secuencia
- Cadena que represente las bases
- Comentario
- Representación ASCII de las calidades de las bases

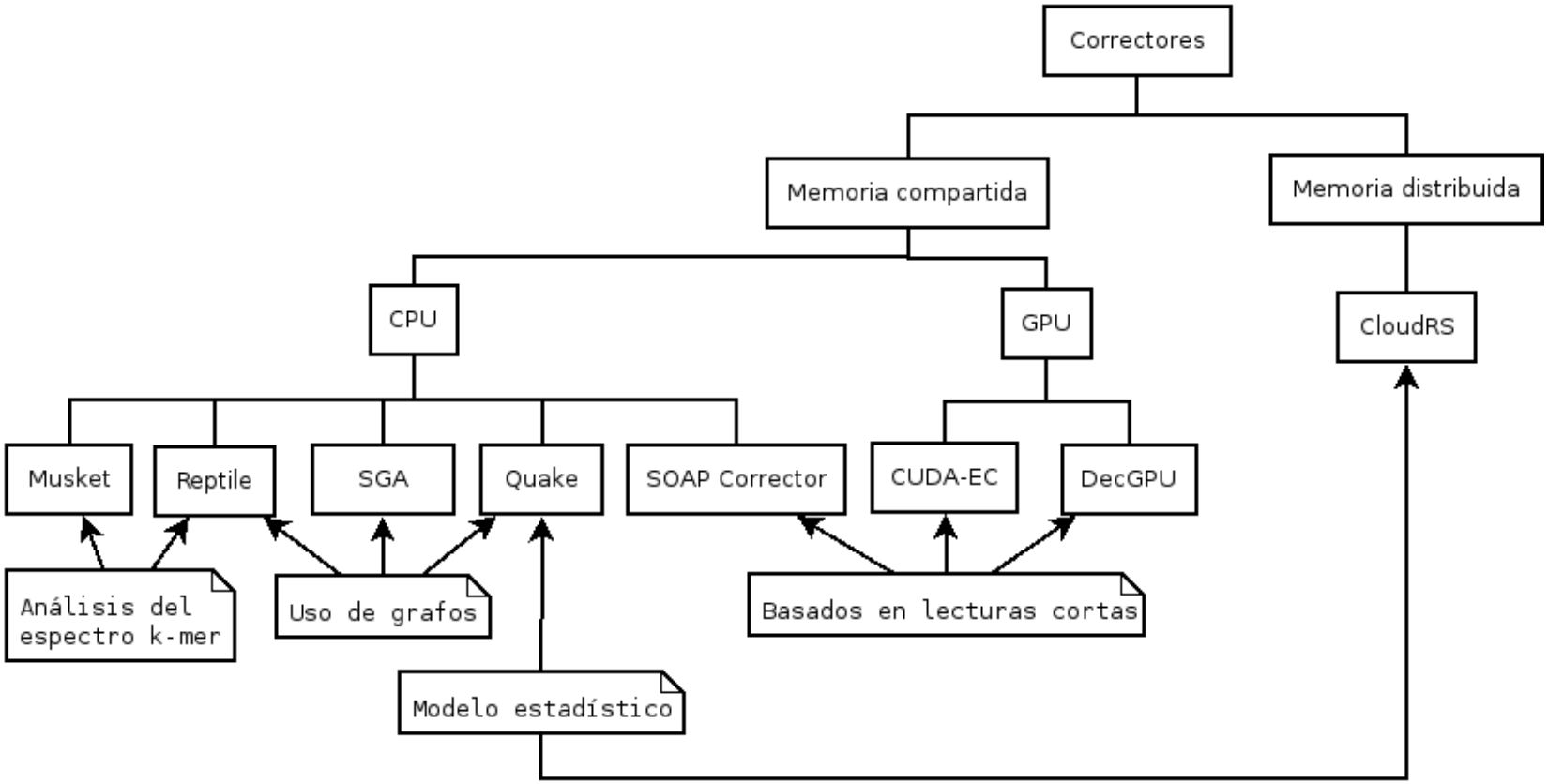
2 modos: **single-end** o **paired-end**

Acerca de los errores de secuenciación



- Los errores siguen un proceso estocástico
- Principales errores (sustitución o translocación)
- Varias soluciones
  - Análisis del espectro k-mer
  - Modelos estadísticos (e.g. probabilidad acumulada)
  - Uso de grafos para ciertas optimizaciones
  - Combinación de varios métodos

Estado del arte

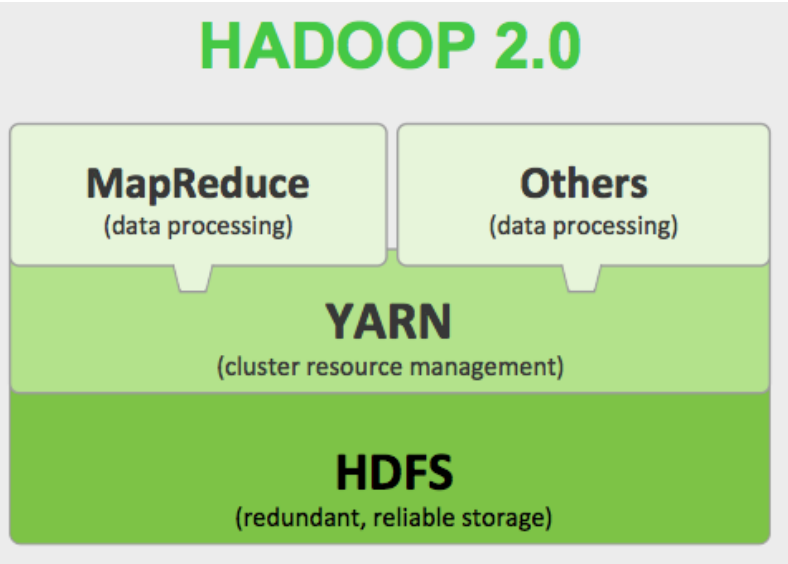


Musket

- Basado en el análisis del espectro kmer
- Desarrollada en C++
- Paralelizado mediante hilos (OpenMP)
- Uso de un patrón maestro/esclavo para evitar desbalanceo de carga e hilos ociosos
- Buen rendimiento
- Proporciona una alta cobertura sobre los conjuntos de datos
- Según estudios recientes es corrector que mejor precisión tiene (aprox. 81 %)

`musket.sourceforge.net`

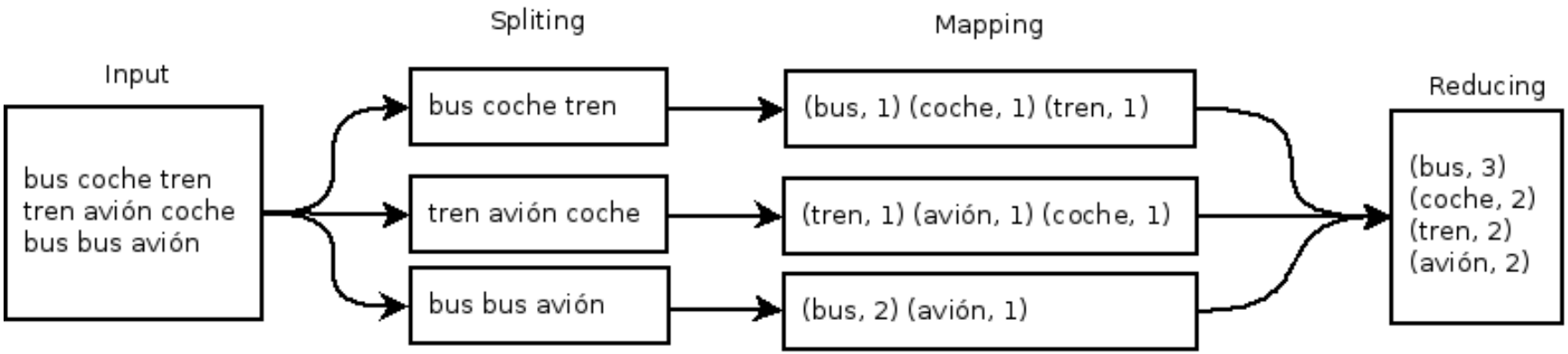
Tecnologías - Hadoop



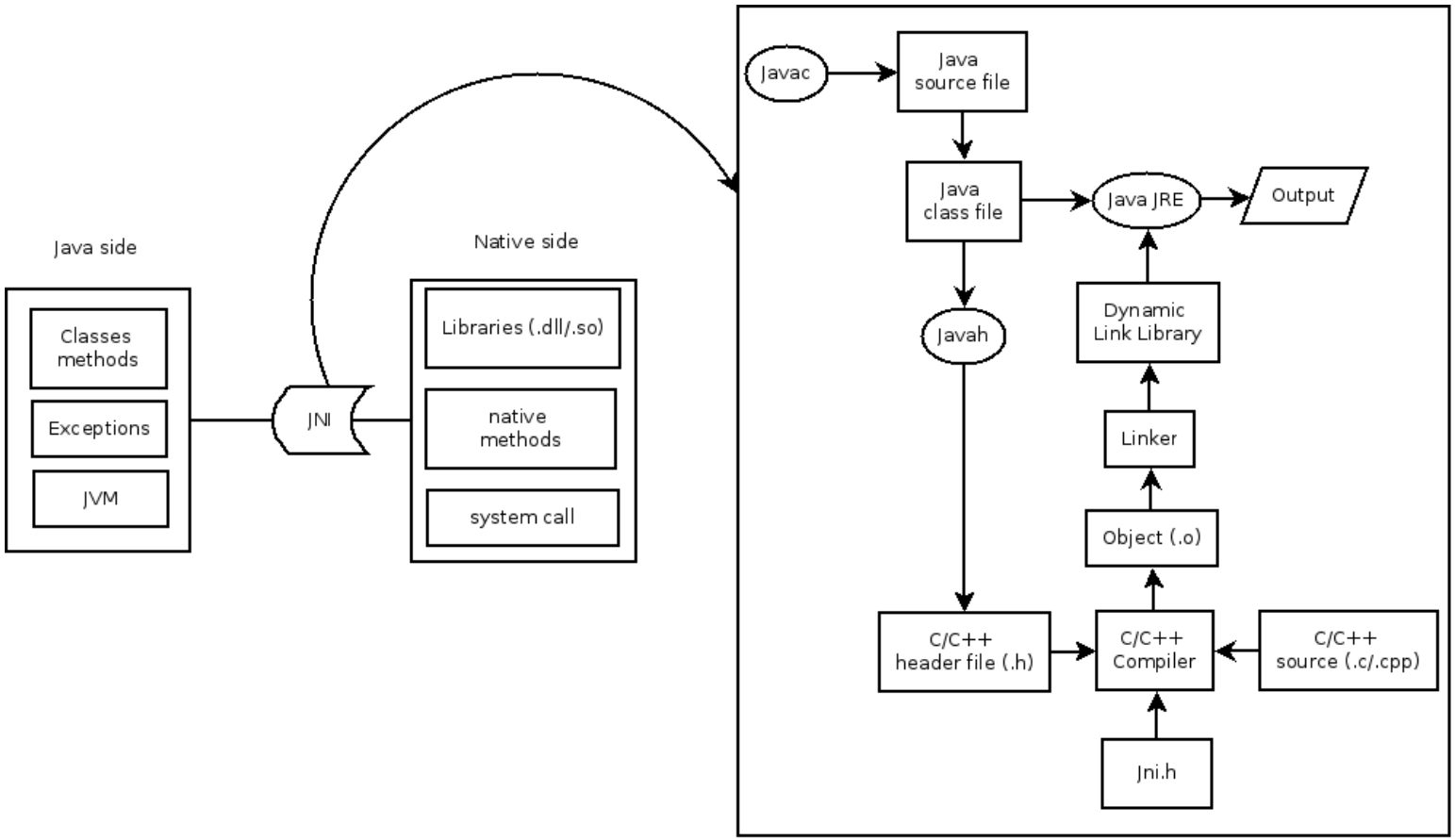
Arquitectura de Hadoop 2.0

- Framework de código abierto desarrollado en Java
- Procesamiento distribuido y eficiente de grandes volúmenes de datos
- Modelo MapReduce
- Orientado a aplicaciones distribuidas y entornos clúster
- Usa una arquitectura maestro/esclavo

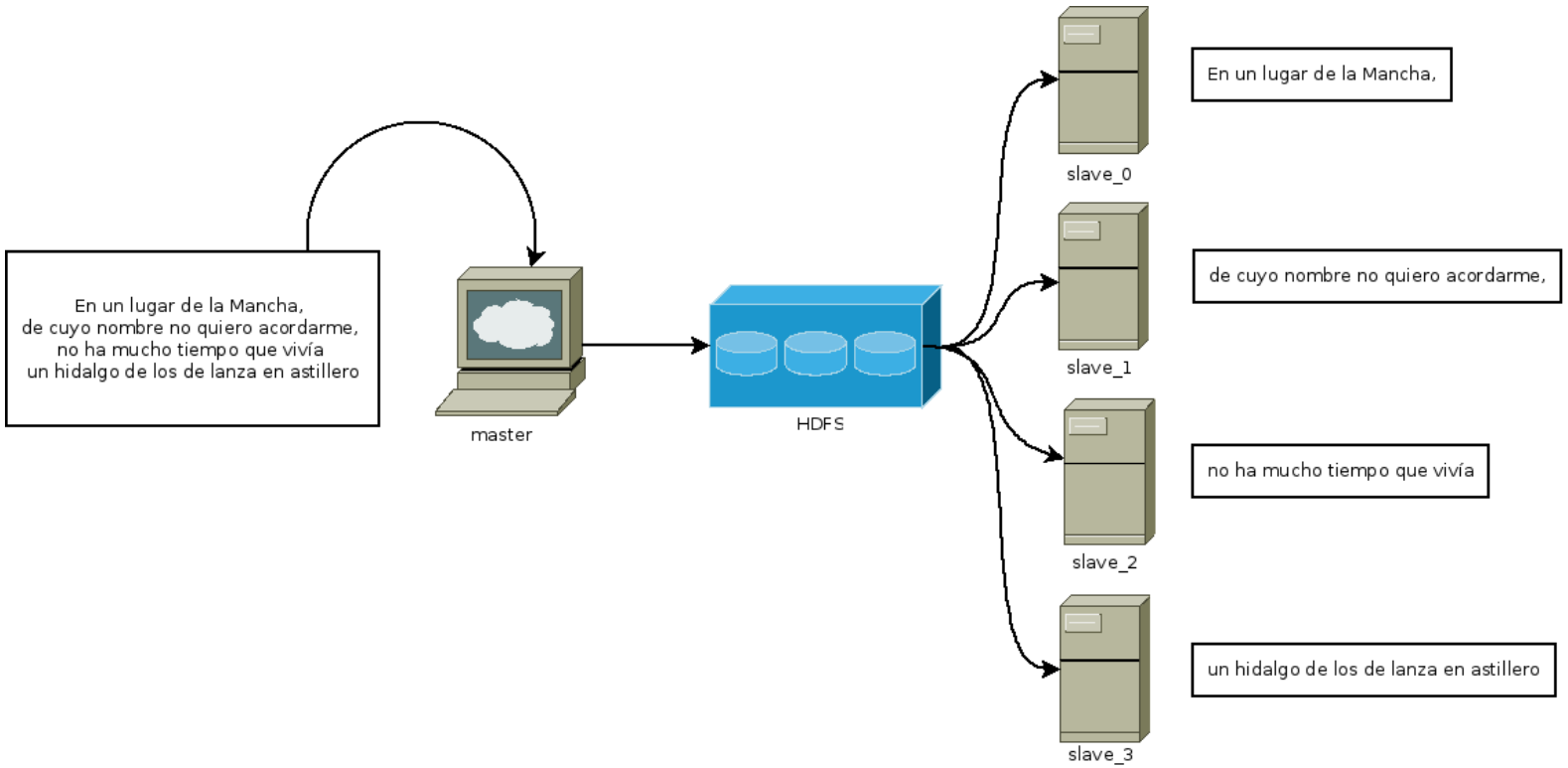
Ejemplo MapReduce (wordcount)



Tecnologías - Java Native Interface (JNI)

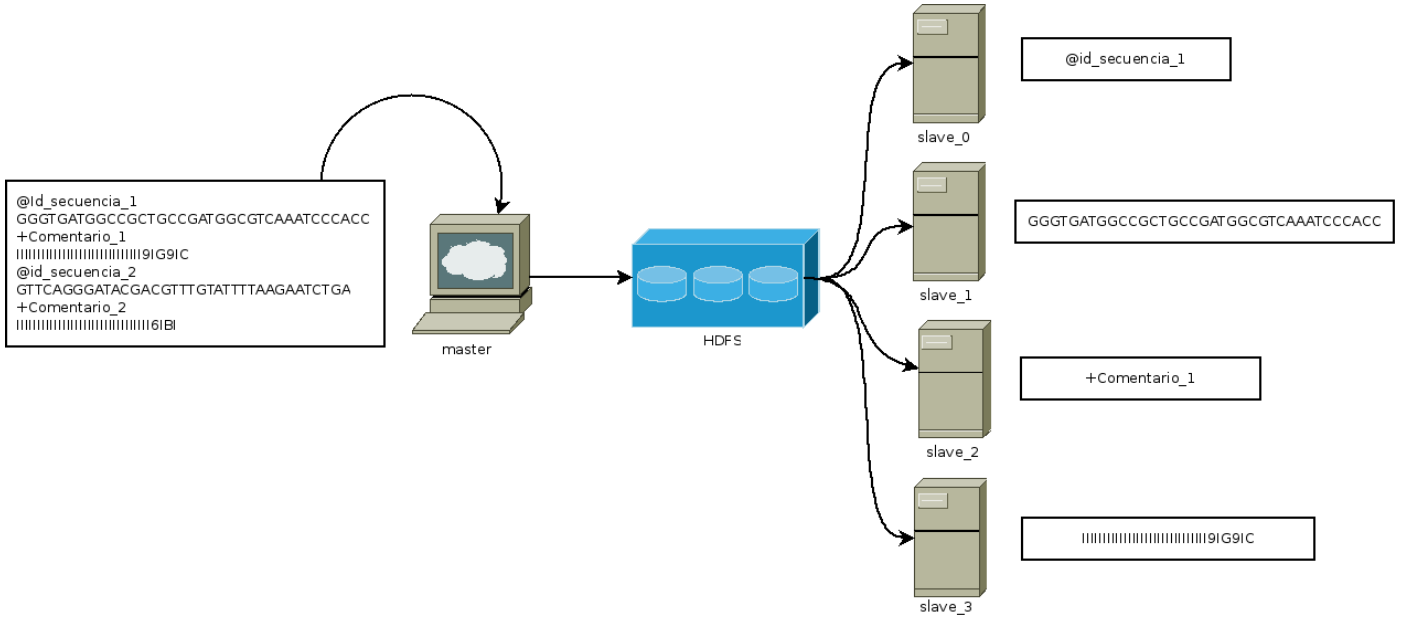


Tecnologías - HDFS



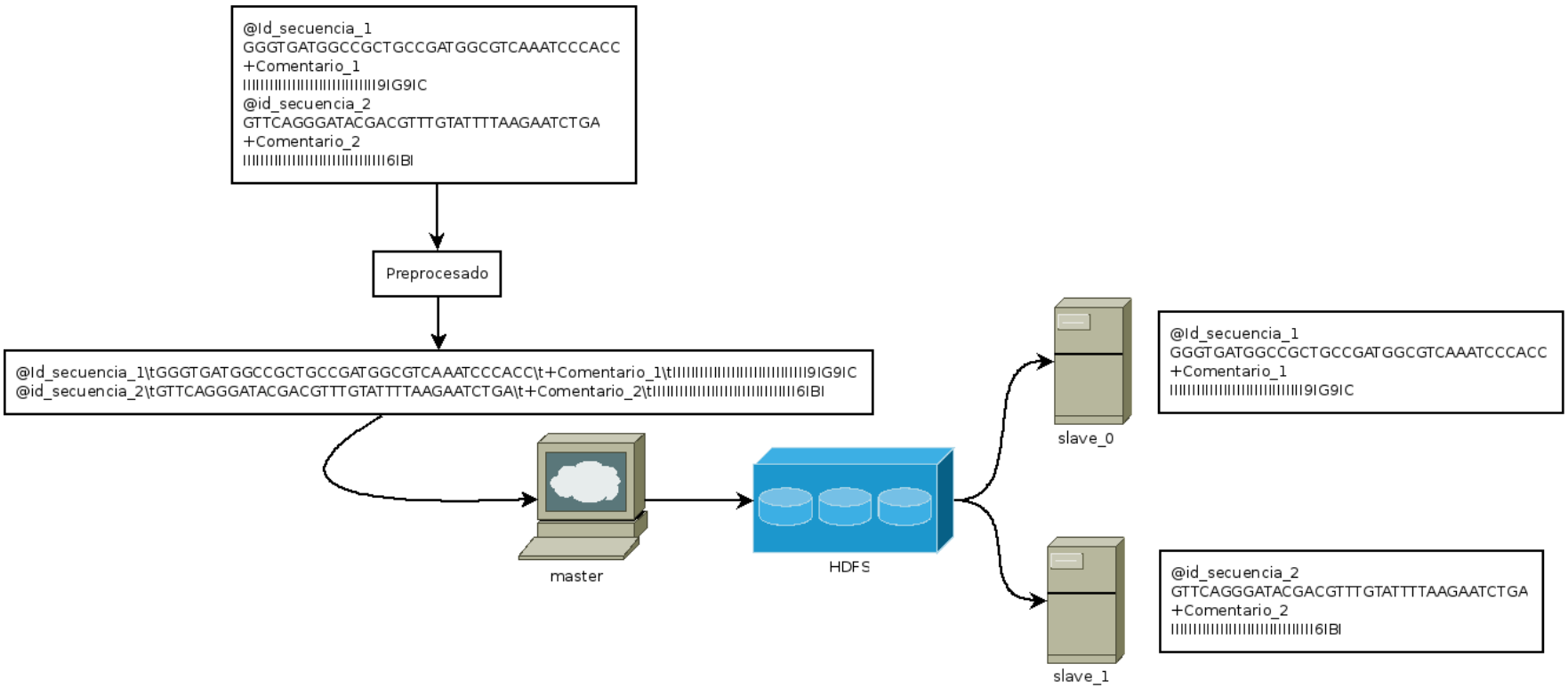
Procesamiento de un texto en HDFS

Tecnologías - HDFS



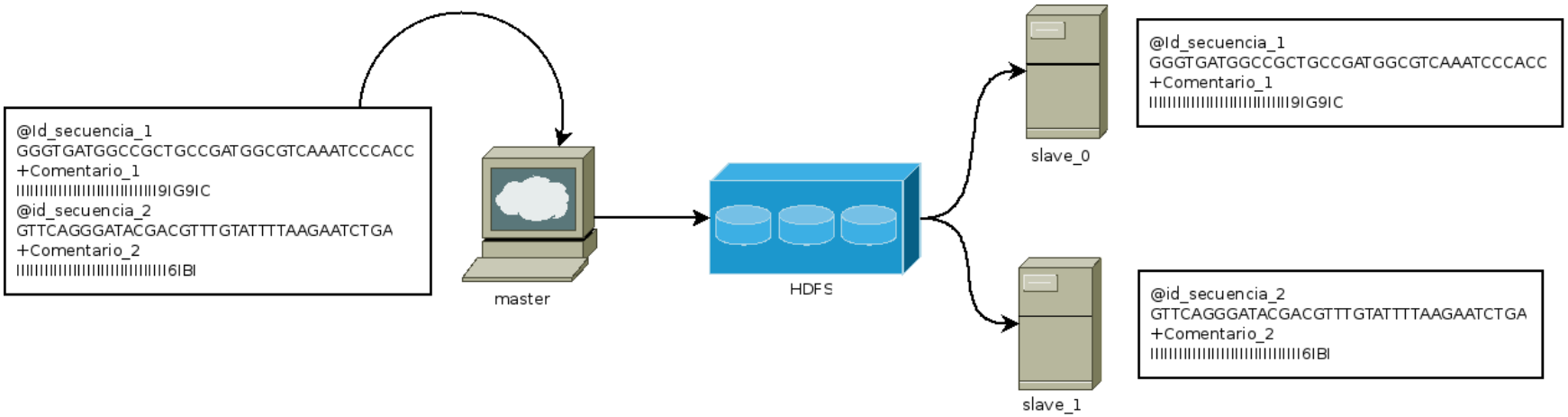
- Problemática con los formatos fast/fastq
- CloudRS resuelve este problema realizando un preprocesado inicial
- Este preprocesado aumenta considerablemente el tiempo

Tecnologías - HDFS



Ejemplo del preprocesado de CloudRS

Tecnologías - Hadoop Sequence Parser (HSP)



- Soporta formatos fasta y fastq
- En modo single-end o paired-end
- Se puede utilizar con cualquier framework del que soporte HDFS (e.g. Spark, Flink, etc)

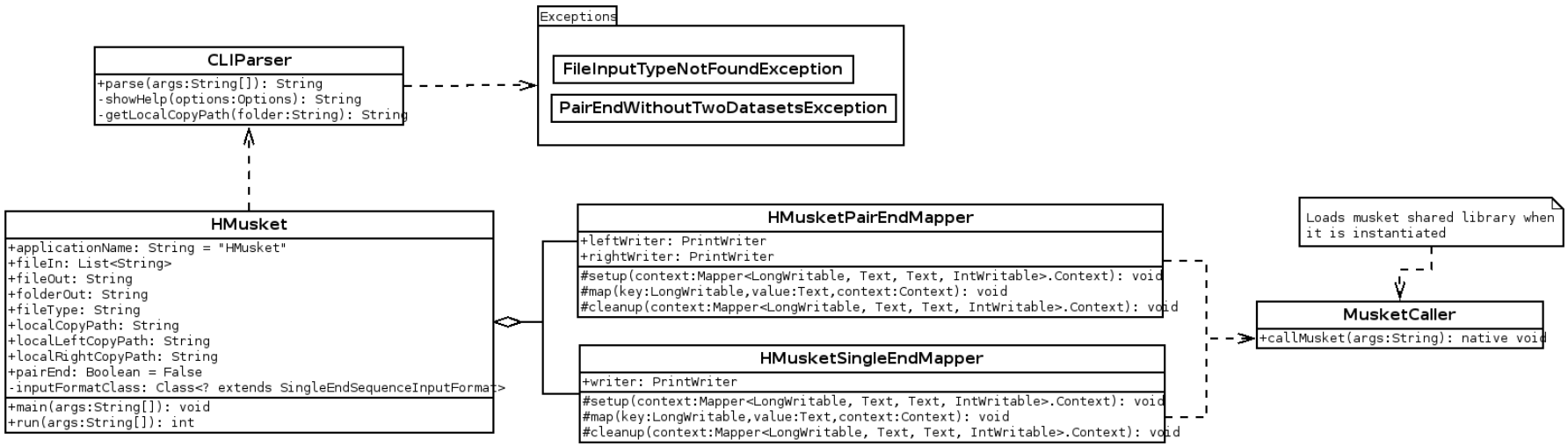
`https://github.com/rreye/hsp`

HMusket

- No requiere de un preprocesado inicial, hace uso de HSP
- Soporta datasets en formato fasta o fastq
- Ya sea en modo single-end o paired-end
- Proporciona precisión similar a Musket
- Paralelizado para un clúster
- Un segundo nivel de paralelismo a nivel de hilo
- Es de código libre

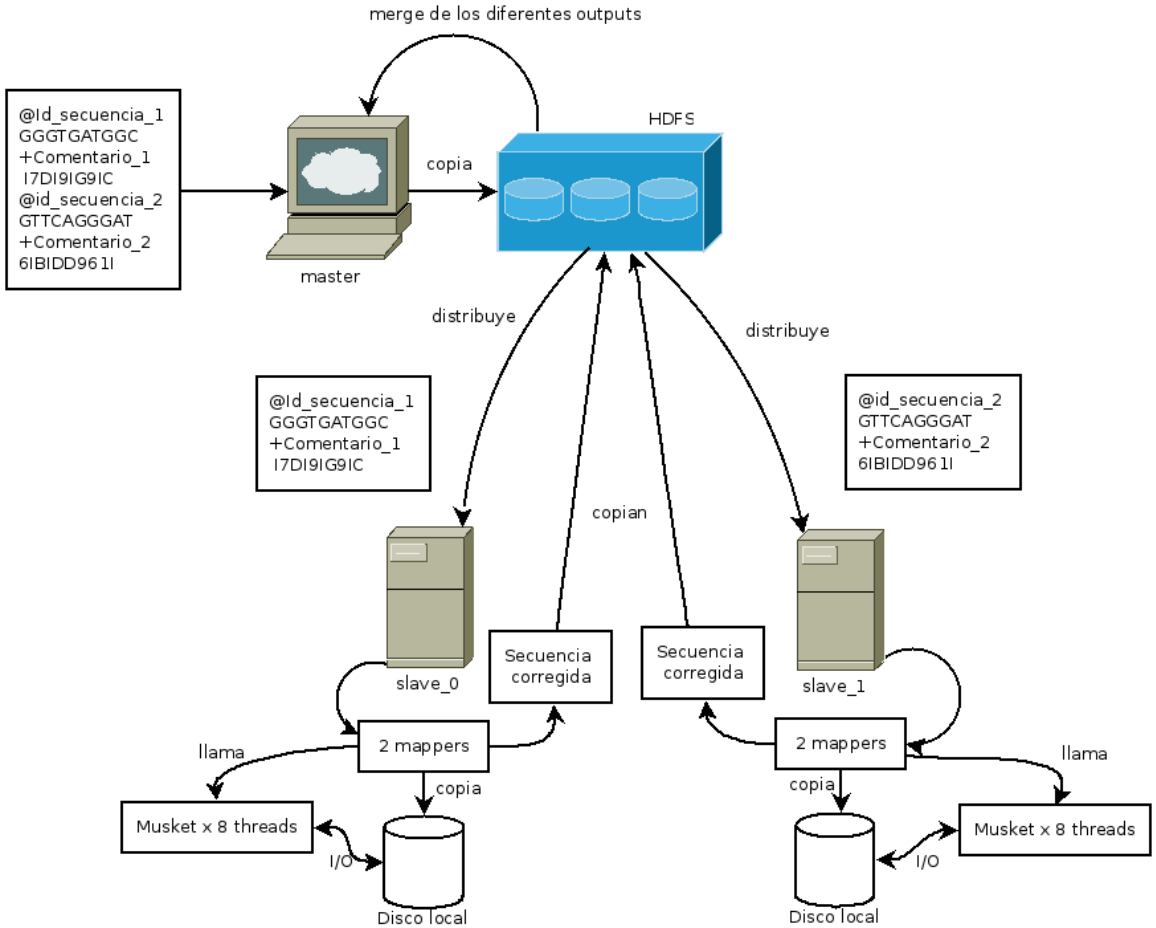
`https://github.com/luislorenzom/hmusket`

Diseño



Diseño de HMusket

Implementación



Pruebas - Entorno de pruebas

Modelo CPU	2 x Intel Xeon E5-2660 Sandy Bridge-EP
Velocidad/Turbo CPU	2.20 GHz/3 GHz
Cores por CPU	8
Threads por core	2
Cores fisicos/virtuales	16/32
Cache L1/L2/L3	32 KB/256 KB/20 MB
Memoria RAM	64 GB DDR3 1600 Mhz
Discos	1 x HDD 1 TB SATA3 7.2K rpm
Redes	InfiniBand FDR y Gigabit Ethernet

Especificaciones Hardware (1 nodo de cómputo)

S.O.	CentOS 6.9 (basado en Red Hat Enterprise Linux)
Kernel	2.6.32-696.23.1.el6.x86_64
Suite GNU	6.3.1
JDK	OracleJDK 1.7.0_80
BDEv	3.1
Hadoop	2.9.0
Musket	1.1

Especificaciones Software

Cabina 0 del clúster Plutón (`http://pluton.des.udc.es`)

Pruebas - Experimentos

Se evaluaron dos conjuntos de datos:

- SRR921889
  - Formato Single-end
  - 50 millones de secuencias
  - 100 bases por secuencia
  - Tamaño de 16 GB
- SRR948355
  - Formato Paired-end
  - 2 x 69 millones de secuencias
  - 101 bases por secuencia
  - Tamaño de 44 Gb (2 x 22 GB)

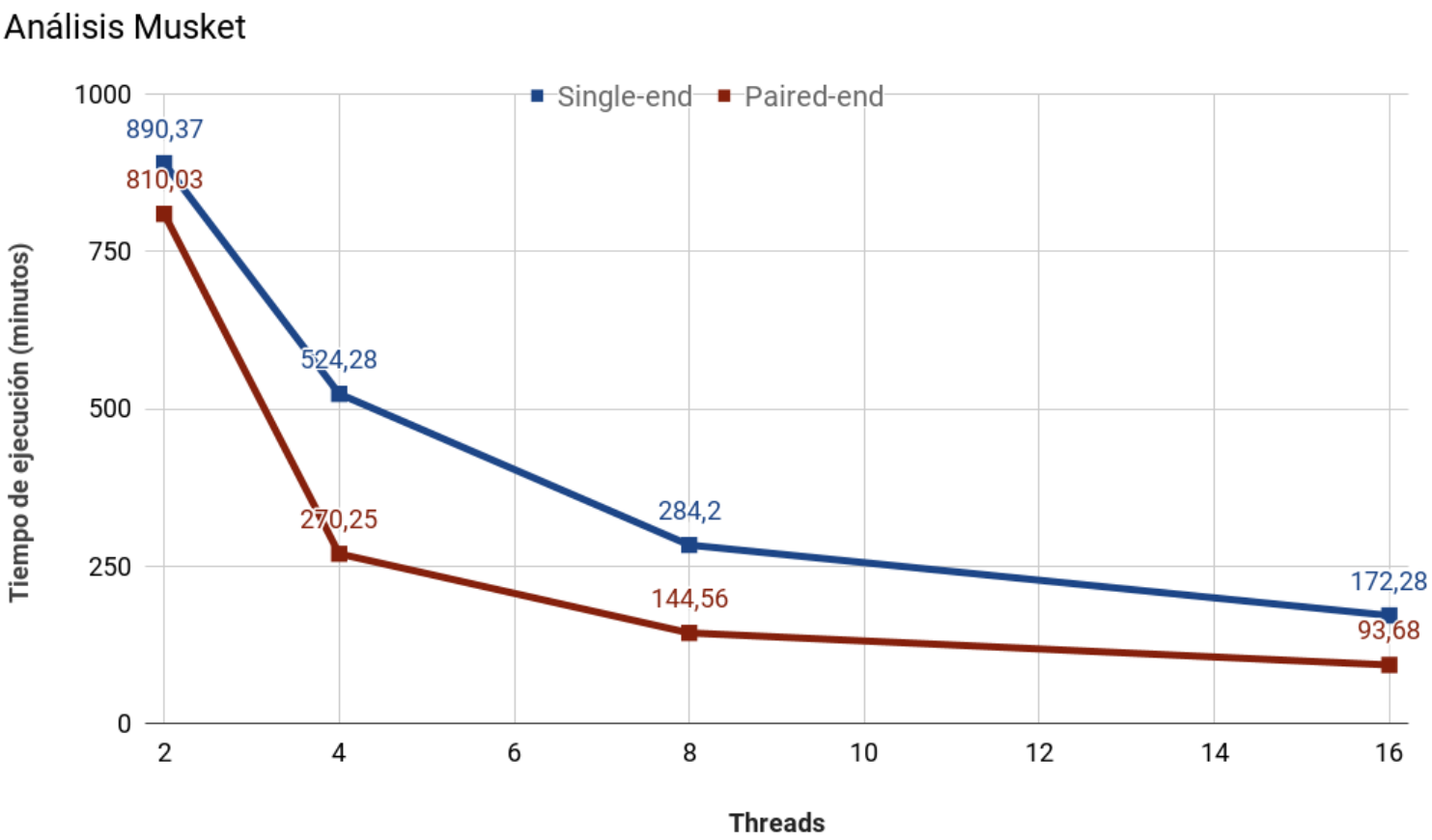
Experimentos Musket

- Un solo nodo de cómputo
- Uso de 2, 4, 8, 16 hilos

Experimentos HMusket

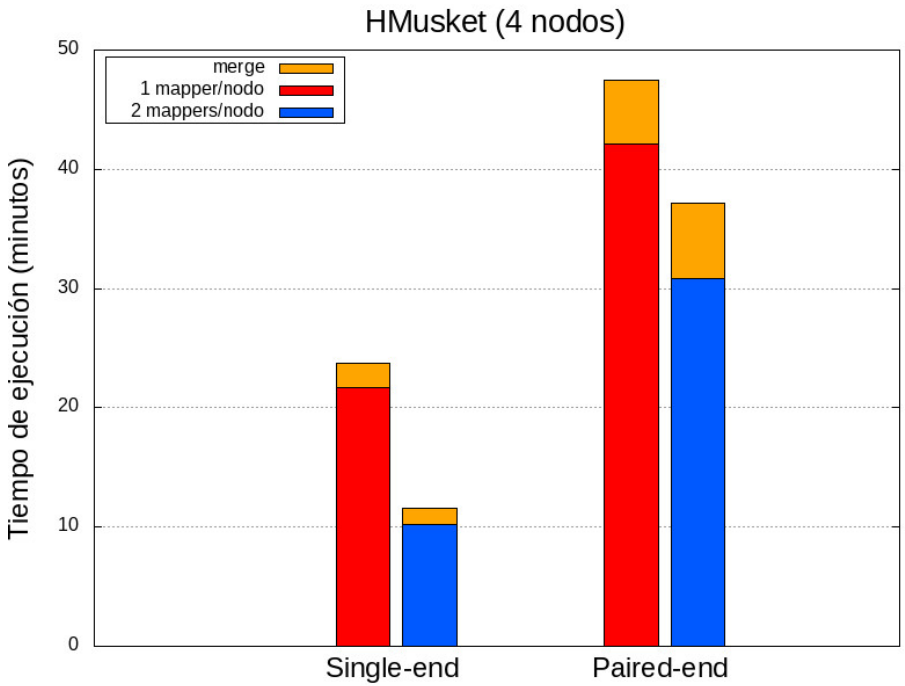
- Búsqueda de la combinación más óptima
- Se prueban con 4 y 8 nodos con 1 y 2 mapper por nodo
- Realización de 5 ejecuciones por prueba (utilizando la mediana)
- Medición por separado de la operación merge

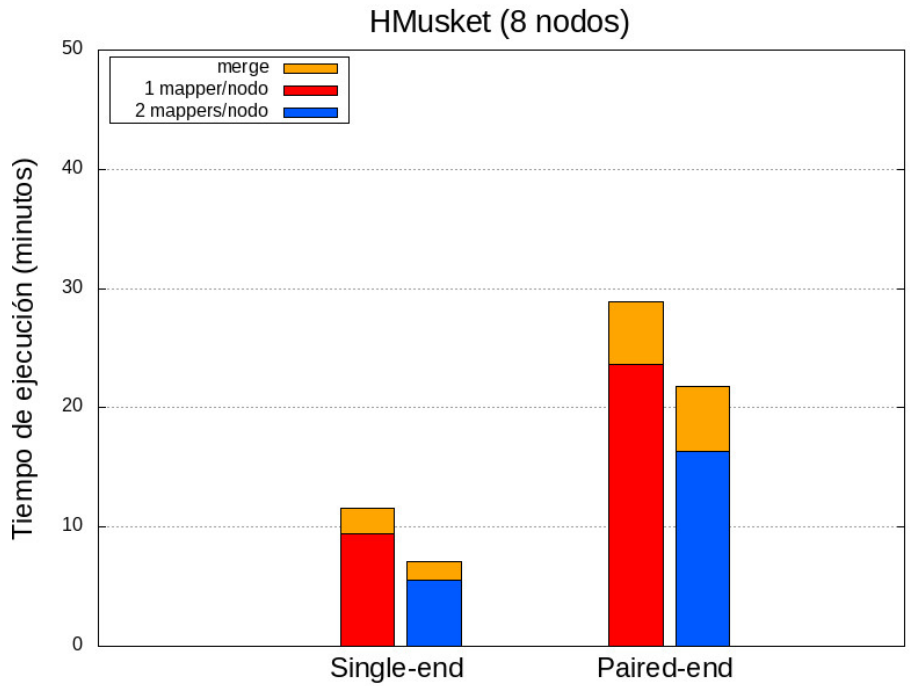
Resultados - Musket



Aceleraciones: 4,69 para Single-end y 8,64 para Paired-end

Resultados - HMusket





- Escalabilidad lineal
- La configuración con 2 mappers por nodo puede acelerar x2
- El merge incrementa los tiempos

Resultados - Comparativa

Dataset	Mejor tiempo Musket	Mejor tiempo HMusket (4 nodos)	Mejor tiempo HMusket (8 nodos)	Aceleración
SRR921889	172,28	9,83	8,30	31,21
SRR948355	93,68	30,83	16,36	5,72

- Para los conjuntos de datos single-end se obtiene una aceleración de 31,21
  - Memoria compartida (16 threads) - Memoria distribuida (8 nodos, 2 mappers/nodo)
- Para los datos en formato paired-end la aceleración es de 5,72
  - Misma comparación que en los casos anteriores
- Los cambios de configuración en el uso de nodos (experimento single-end) implicó una aceleración de 2,13

Conclusiones

- Musket es una herramienta muy potente y precisa
- Su implementación en memoria compartida no puede hacer frente a una alta demanda de datos
- Solución: reutilizar el software en un sistema de memoria distribuida
- Evitando cuellos de botella y preprocesados innecesarios
- Se puede obtener una gran aceleración (hasta 30 veces más rápido)

Trabajo futuro

- Sería interesante realizar modificaciones en HMusket para que utilizase comunicación entre hilos (pipes) y compararla con la versión aquí expuesta
- Evaluar la herramienta la escalabilidad de la herramientas y del merge con otros conjuntos de datos, variando el número de secuencias y tamaño de las mismas

¡Muchas gracias por la atención!

Preguntas, comentarios...

luis.lorenzom@udc.es

Hadoop Sequence Parser (HSP): https://github.com/rreye/hsp
HMusket: https://github.com/luislorenzom/hmusket