Distributed stream processing for genomics pipelines

Luca Pireddu*, Francesco Versaci and Gianluigi Zanetti
CRS4, Polaris, Ed. 1, I-09010 Pula, Italy
*luca.pireddu@crs4.it

Introduction

We built a *scalable sequence alignment pipeline* based on Apache *Flink* and *Kafka*

- Flink framework for distributed stream-oriented processing
- Kafka is connector service: connect processes without intermediate files

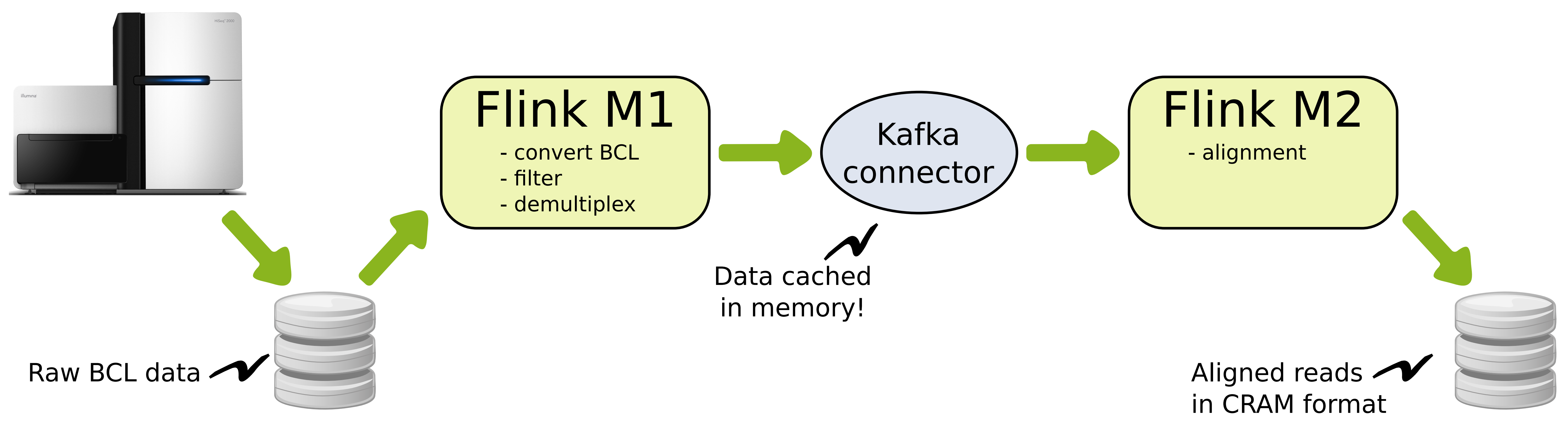
Our solution is:

- *distributed* and *scalable* (runs over computers, efficiently)
- *robust* (resists hardware failures)

Motivation

- As it gets cheaper, sequencing can be a prime tool for personalized medicine
- *Population-wide applications* require scalable analysis
  - need to extract clinically relevant information from raw data
- Conventional processing workflows not scalable
  - sequences of independent tools
  - communicate by intermediate files on shared file system

NGS Alignment Pipeline



Our pipeline is implemented as two Flink modules connected by Kafka.

Module 1: preprocessor

- Reads raw *Illumina* data in BCL format
- Performs BCL conversion
- Filters based on base calling QC
- Demultiplexes reads

Module 2: aligner

- Integrates *BWA-MEM* through the Read Aligner API (http://github.com/crs4/rapi)
- Aligns reads
- Formats *CRAM* output

Streaming

- Flink nodes process data as soon as it arrives
- Not batch oriented!
- Operations run simultaneously, streaming data from one to the next
- Strategy improves pipeline *efficiency* and reduces overall time to result

Architecture can easily be *extended*

- chain more Kafka and operator nodes

Distributed

- Single Flink operations run over multiple computers
- Data and work are spread automatically
- Scalability: more nodes = more *speed*
- *Fault tolerant*: if a node breaks, other nodes complete the job
- Weakness: in our setup, Kafka is not replicated; if the Kafka node breaks the pipeline goes down

Evaluation

We evaluated our pipeline's performance and scalability

Equipment

- Amazon EC2, with up to 12 r3.8xlarge nodes
  - 32 virtual cores, 244 GB RAM, 4x1.9 TB SSD, 10 Gbit Ethernet
- Flink and HDFS over cluster nodes
- One Kafka broker

Dataset

- 1/4 multiplexed run from an Illumina HiSeq 3000 (48 DNA samples; 48 GB)

Baseline

- Pipeline implemented with bcl2fastq2 and bwa-mem
- single r3.8xlarge node, multithreaded

Running times

Running times of our pipeline and the baseline

nodes	time (minutes)
baseline	137
1	152
2	77
4	39.6
8	20.4
12	14.3

On a single node, our pipeline is 11% slower

- due to Flink overhead

Our pipeline achieves *near-optimal scalability*

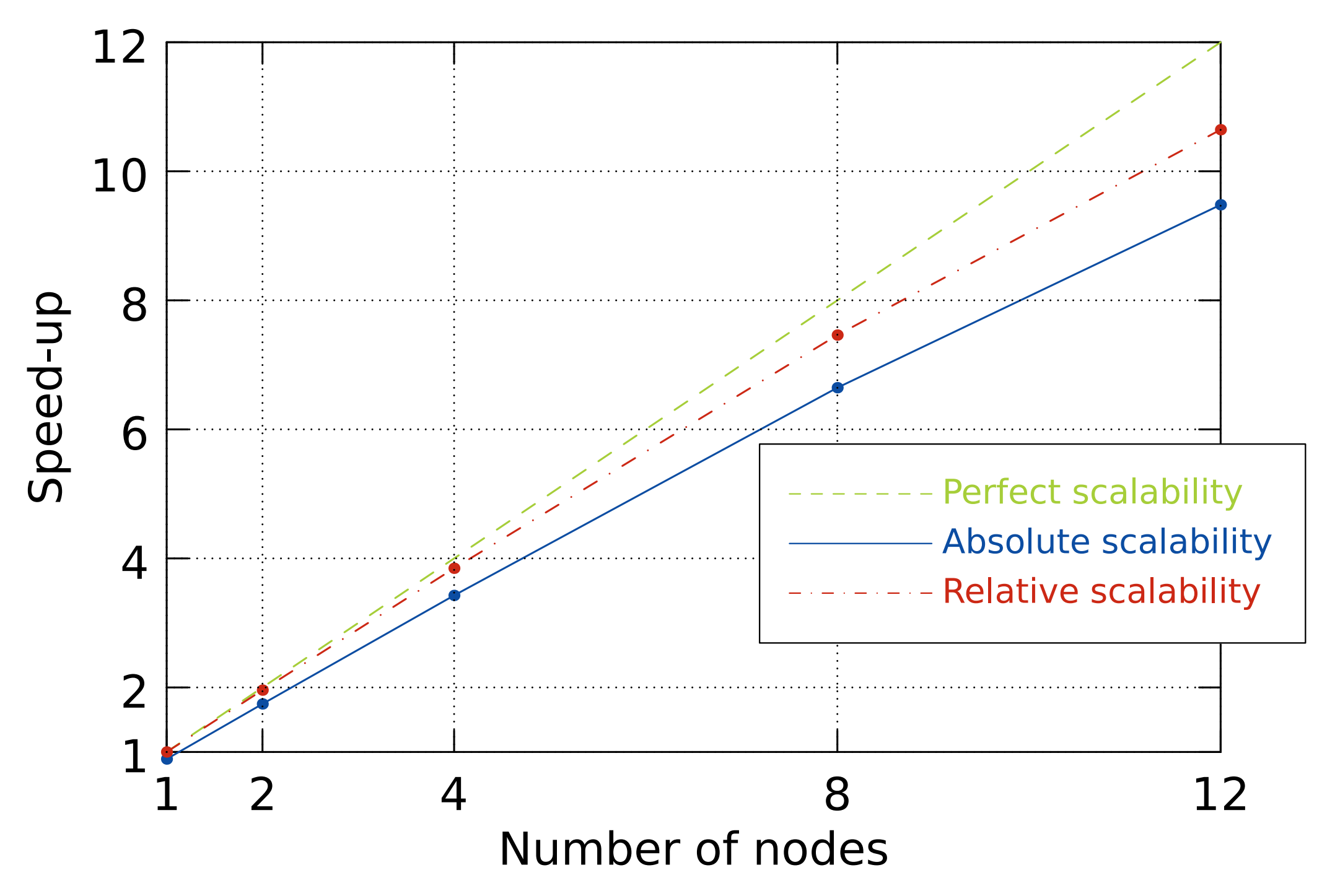
- Relative (compared to itself on 1 node): 10.6 on 12 nodes (88.3%)
- Absolute (compared to baseline): 9.5 on 12 nodes (79.2%)

Especially positive considering 14-min run time

- i.e., fixed-cost overheads take up a significant portion of run time

Scalability

Relative and absolute scalability of our pipeline w.r.t. the number of computing nodes



Future Work

- Fault tolerant Kafka brokers
- Interface with GATK4: full distributed variant calling pipeline