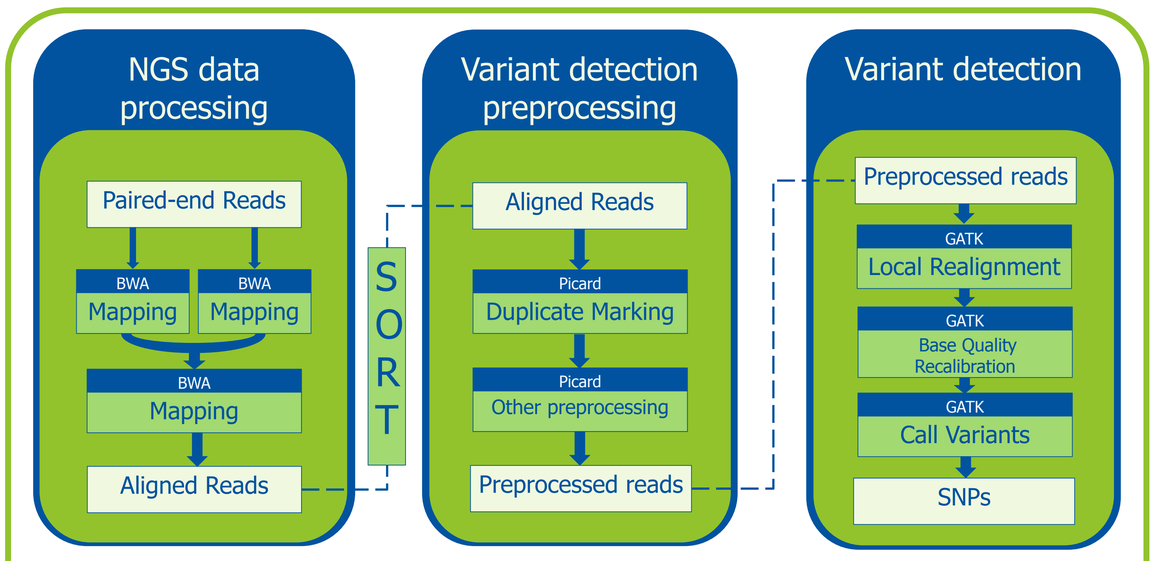
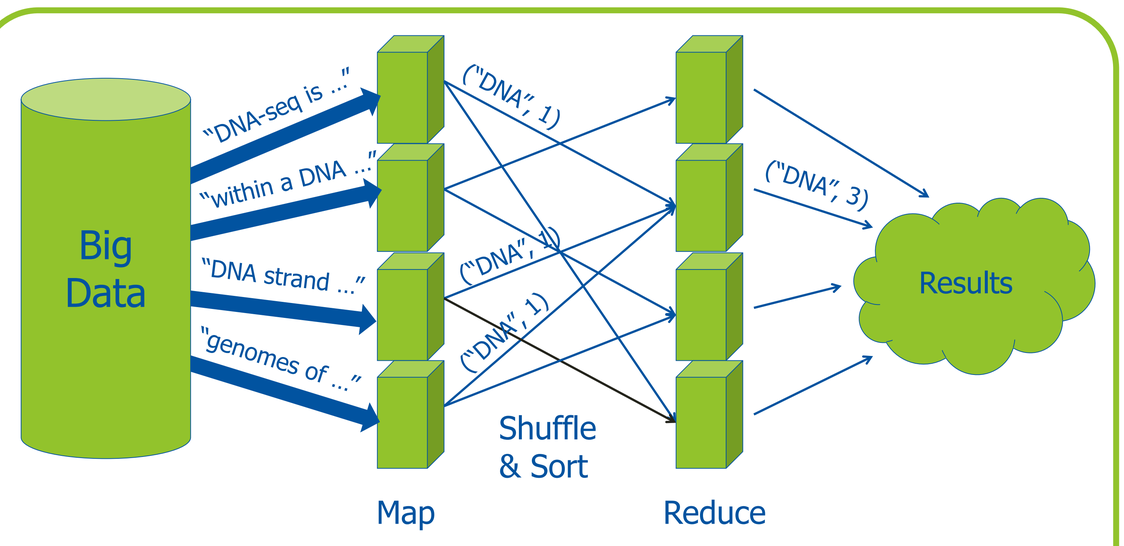
Halvade

whole genome analysis with MapReduce

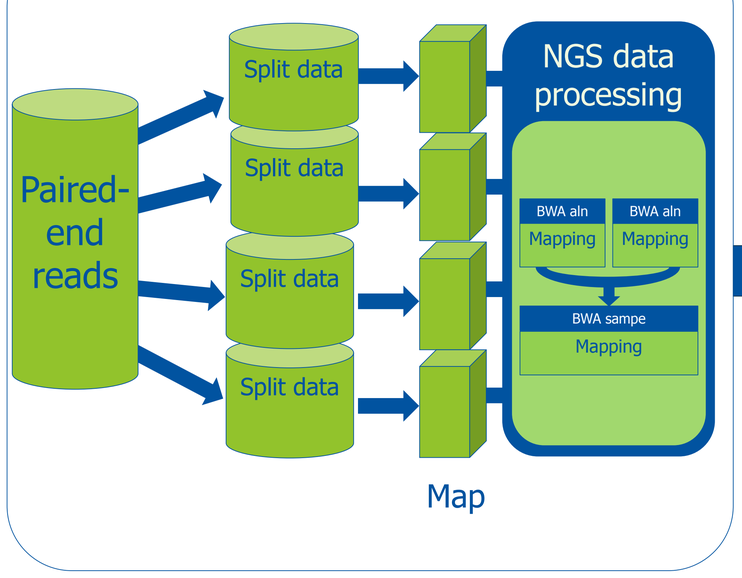
Dries Decap[1,5], Joke Reumers[2,5], Charlotte Herzeel[3,5], Pascal Costanza[4,5], Jan Fostier[1,5]
[1]: Ghent University – iMinds, [2]: JANSSEN R&D - Beerse (Belgium), [3]: IMEC, [4]: Intel Corporation NV/SA,
[5]: ExaScience Life Lab, Kapeldreef 75, B-3001 Leuven, Belgium



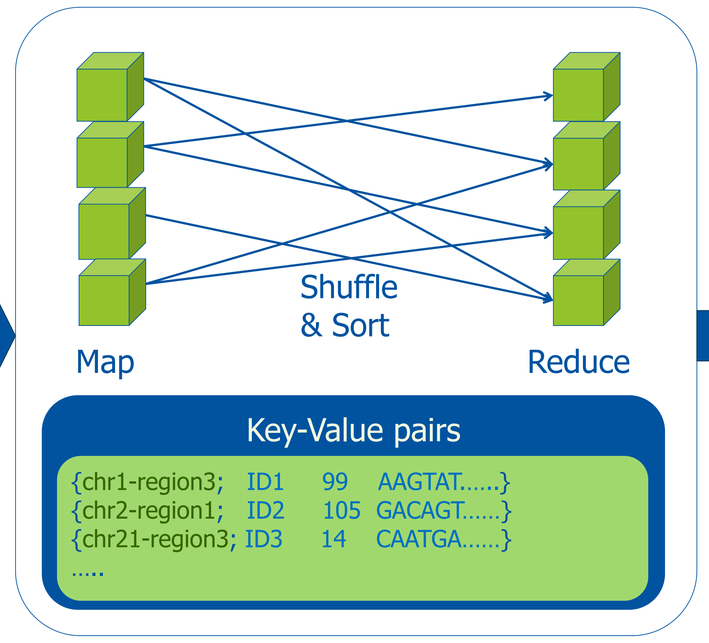
The best practice pipeline for DNA-seq analysis consists of two big phases, **mapping** the paired-end reads and doing **variant detection** with the required preprocessing. The tools used for these two phases are respectively BWA and GATK with SAMtools and Picard for the preprocessing.



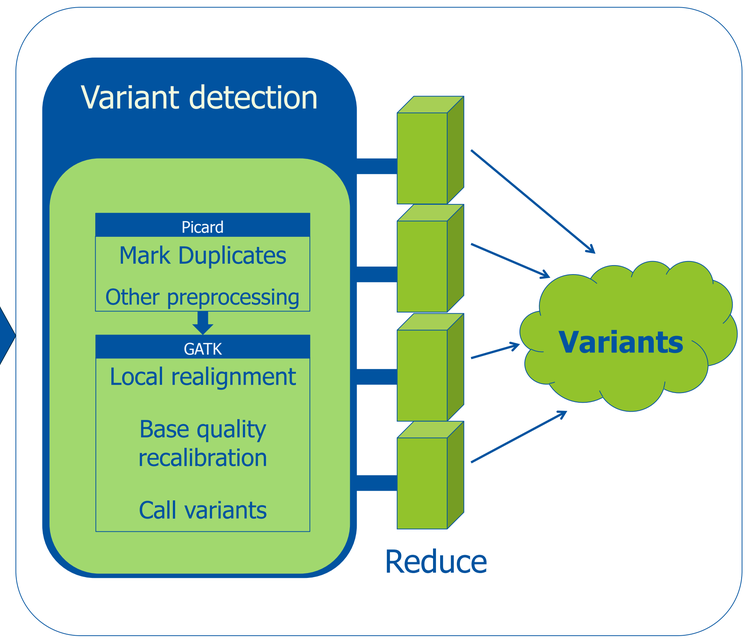
A **MapReduce** program has two phases, the **map** phase performs filtering and makes key-value pairs. The input for the map phase is typically stored on a distributed filesystem called HDFS. These key-value pairs are sorted and grouped per key. Next is the **reduce** phase, which performs an aggregation operation on all values per key.



The input data is preprocessed, paired-end reads are interleaved and stored in smaller files. Each map task has one file as input, this ensure the data is well balanced over all nodes. In the map phase the data is **mapped to a reference genome**.



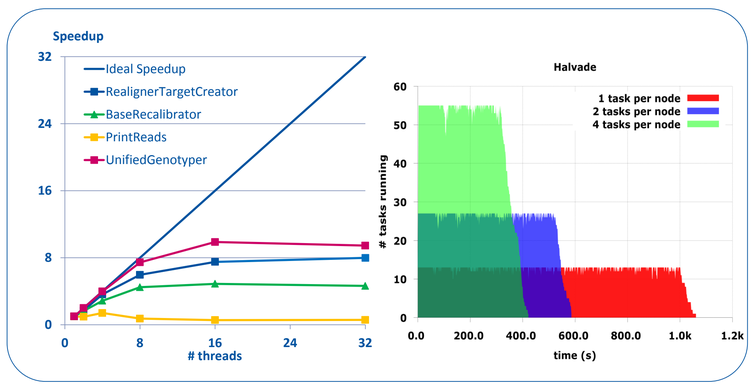
The **efficient sort and shuffle** from Hadoop MapReduce is used to split the mapped reads into groups. Each group belongs to one region of a chromosome. The key-value paires generated are the chromosomal region and the mapped read respectively.

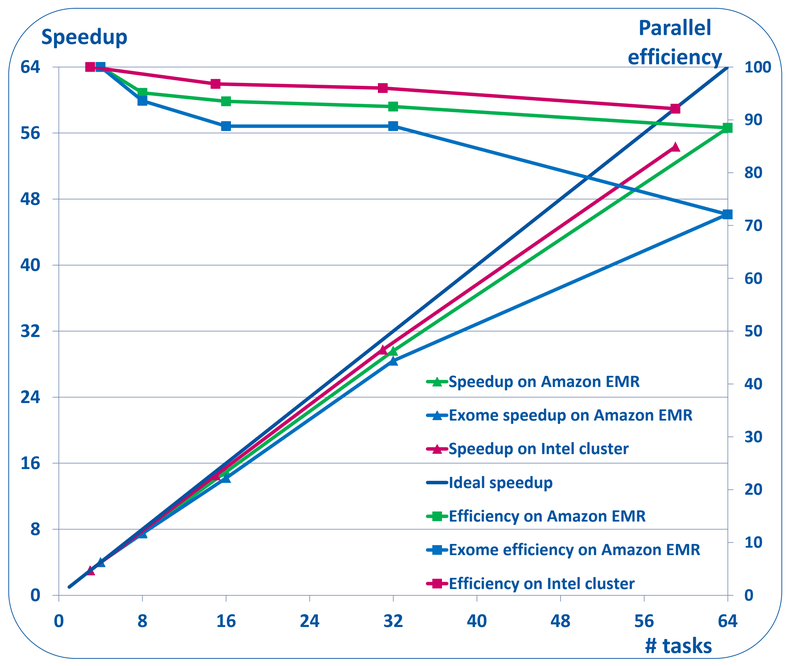


Each Reduce task processes a **chromosomal region** with corresponding mapped reads. These reads are preprocessed using SAMtools and Picard. The variant calling itself is done with GATK and stored on HDFS.

To utilize all available resources the individual tools are run in parallel. However some of the tools have limited parallel speedup (left graph) and cannot achieve optimal performance.

To reduce the absolute runtime, Halvade runs **multiple tasks simultaneously** on each node. The right graph shows the total number of map tasks running over time, which indicates that using multiple tasks per node causes a significant speedup.





The parallel efficiency of Halvade was first assessed on the Intel Big Data cluster in Swindon, UK. This benchmark shows that Halvade has a parallel efficiency of 92,1% using 360 cores. In absolute runtimes, Halvade does **whole genome analysis in under 3 hours** using 15 nodes compared to 120 hours on one node without Halvade.

As a second assessment Halvade was run using Amazon EMR. This benchmark shows that, using 264 cores, Halvade achieves a parallel efficiency of 88,5%. In absolute runtimes this comes down to running whole genome analysis in under 3 hours and the **cost for whole genome analysis is ~111 USD**. Halvade supports exome analysis and gets the results in under one hour for ~20 USD on Amazon EMR.

To asses the accuracy of Halvade, the output was compared with a validation dataset. Halvade has an **accuracy of 99,4% for whole genome analysis** and 97,5% for exome analysis.
Halvade can be used on any Hadoop MapReduce v2.0 or newer distribution including Cloudera and Amazon EMR and is freely available at http://bioinformatics.intec.ugent.be/halvade.

Dataset: NA12878, ~1,5 billion 100bp paired-end reads, 50x coverage, ~86GB compressed
Intel Big Data Cluster: 15 nodes (dual socket Intel® Xeon® CPU E5-2695 v2 @ 2.40GHz, 62GB RAM)
Amazon EMR: 16 nodes (32 vCPU, Intel® Xeon® CPU E5-2680 v2 @ 2.80GHz, 60GB RAM)

Acknowledgments
This work is funded by Intel, Janssen Pharmaceutica, and Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT).