CloudBurst: highly sensitive read mapping with MapReduce (Summary)

CloudBurst is a seed-and-extend read mapping algorithm which has been optimised for mapping NGS data to reference genomes, mainly the human one, using the Hadoop implementation of MapReduce for parallel execution across multiple nodes. Millions of short DNA sequences, reads, are collected from the target genome and then mapped to a reference genome to find the read occurrences (allowing a small number of differences – mismatches and indels). Thus, differences between genomes can be analysed.

MapReduce is a software framework whose computation is divided into two phases: map and reduce, separated by an intermediary shuffle. The functions are automatically executed in parallel over multiple processors. The map function generates key-value pairs from the input data based on a relationship of the problem in question. Multiple instances of it can execute in parallel on different parts of the input, if the input is large, leading to running time being divided among the processors used. Once complete, the pairs are shuffled so that the values with same key are grouped together producing a distributed hash table indexed by the key with a list of values for each key. The reduce function runs a user-defined function (must be commutative) on each key-value list; each instance executing independently. Reduce-like functions called combiners can be used for optimisation to run after the map; executing on a subset of the values for a given key. MapReduce is designed for computations with large datasets, beyond what can be stored in RAM. A robust file system is used to support it as the multiple files required for intermediary results and inter-machine communication can cause a large bottleneck overhead. Hadoop and the Hadoop Distributed File System HDFS are such open source versions whereby the developers need only write custom map and reduce functions. The Hadoop framework automatically executes them in parallel.

CloudBurst is a MapReduce-based read-mapping algorithm which runs in parallel on multiple processors using Hadoop and is optimised for mapping many short reads to a reference genome, allowing for a user specified number of mismatches. It is split accordingly; the map phase emits k-mers from the reads and reference genome, the shuffle phase groups together k-mers shared between the reads and reference and the reduce phase extends the shared seeds into end-to-end alignments allowing both mismatches and indels.

During evaluation, it was found that CloudBurst scales linearly in execution time as the number of reads increases and with near linear parallel speedup as the size of the cluster increases. At low sensitivity, the overhead of shuffling and distributing the data is large while at high sensitivity, the opposite is observed. However, the speedup when mapping to the full genome did not improve at high sensitivity due to the increased overhead from the increased data size. This can be minimised by aligning more reads in a single batch. It was also observed that the ad hoc method performs well with speedups similar to CloudBurst but fails to reach linear speedup in most cases. An ad hoc parallelisation scheme would be fragile as it wouldn’t have the benefits from Hadoop. It was also tested on Amazon cloud leading to many analyses concerning core and their respective run times.

Future work for CloudBurst is to include quality values in the mapping and scoring algorithms and to improve support for paired reads. Algorithms which do not use a hash table, such as the BWT-based short-read aligners, can also use Hadoop to parallelise execution and the HDFS. Advantages of this are: scalability, redundancy, automatic monitoring and restart and high performance distributed file access.