Nispero: a cloud-computing based Scala tool specially suited for bioinformatics data processing

Evdokim Kovach, Alexey Alekhin, Marina Manrique, Pablo Pareja-Tobes, Eduardo Pareja, Raquel Tobes and Eduardo Pareja-Tobes*

Oh no sequences! Research Group. Era7 bioinformatics

*eparejatobes@ohnosequences.com

Abstract. Nowadays it is widely accepted that the bioinformatics data analysis is a r eal b ottleneck i n many r esearch act ivities r elated t o l ife s ciences. H igh-throughput t echnologies l ike N ext Generation S equencing (NGS) ha ve completely r eshaped t he bi ology a nd bi oinformatics l andscape. U ndoubtedly NGS has allowed important progress in many life-sciences related fields but has also p resented i nteresting ch allenges i n t erms o f co mputation cap abilities an d algorithms. M any kinds o f ta sks r elated w ith N GS d ata a nalysis, as w ell as other bioinformatics data analysis, can be computed in a parallel, independent way; taking the maximum advantage of this can obviously help in leveraging the analysis bottleneck.

Given the way NGS data is generated scalability plays also an important role in its analysis. N GS da ta i s not g enerated i n a c ontinous fashion but i n a ba tch way, t hus t he co mputation n eeds can b e d ramatically d ifferent at d ifferent points.

Cloud c omputing pr ovides a pe rfect framework for s ystems with t hese t wo requirements: parallel and scalable. Besides, it allows adjusting the computation power on demand, a nd t hus n ot be ing a ttached t o (and pa ying f or) a f ixed compute infrastructure.

Nispero is a Scala library for declaring stateless computations and scaling them using c loud c omputing, i n pa rticular a c ombination of s ervices f rom AWS (Amazon Web Services). Some highlights are:

- strongly typed configuration based on Scala code
- CRDT-like semantics (a n ispero i nstance i s es sentially a morphism between idempotent commutative monoids)
- automatic deploy/undeploy

Nispero r elies on t he E C2 s ervice (Elastic C ompute C loud) t o carry out t he computations, on the S3 service (Simple Storage Service) for data storage and on S QS (Simple Q ueue S ervice) a nd S NS (Simple N otification S ervice) for communication between the different system components.

A Nispero system is composed by:

- a 'console' instance that tracks at any moment the status of the whole system g iving t he us er t he opp ortunity t o c heck a t a ny poi nt the current status of the computations, workers, etc.
- a 'manager' instance that is in charge of deploying and undeploying the group of workers

- a set of 'workers' that performs the computations/tasks in a parallel, independent way
- SQS queues for 'input', 'output' and 'error' messages
- S3 objects for 'input' and 'output' files

The lifecycle of a Nispero system is simple but robust. It starts with the launch of the 'console' and 'manager' instances, the 'manager' then takes the tasks from an S3 object, publishes them in a SQS queue and launches the workers. The workers take the messages with the tasks from the corresponding SQS queue (i.e. the 'input' queue) in an independent, parallel way. Once they have finished the computation they put the results of the computation in S3 objects, publish a message in the 'output' SQS queue and delete the input message of the corresponding task from the 'input' queue.

Nispero is an open-source project released under AGPLv3 license. The source code is available at https://github.com/ohnosequences/nispero

This project is funded in part by the ITN FP7 project INTERCROSSING (Grant 289974).

Keywords: cloud computing; distributed systems; high performance; Scala; NHS