The Abdus Salam International Centre for Theoretical Physics

2494-27

Workshop on High Performance Computing (HPC) Architecture and Applications in the ICTP

14 - 25 October 2013

GC3 Use cases for the Cloud

Antonio Messina
University of Zurich
Switzerland

University of Zurich[UZH]

GC3: Grid Computing Competence Center

GC3 Use cases for the Cloud

Some real world examples suited for cloud systems

Antonio Messina <antonio.messina@uzh.ch>

Trieste, 24.10.2013

Who am I

System Architect at the GC3 - Grid Computing Competence Center of the University of Zurich

our mission is to **foster research, education, infrastructure, and usage of distributed computing** at the University of Zurich.

we maintain computational infrastructures:

- HPC cluster(s)
- A Cloud (called Hobbes)
- An ARC grid site

we develop software to enable users to run on these infrastructures

we organize schools and training events for Python, Cloud systems, (Grid) and more. . .

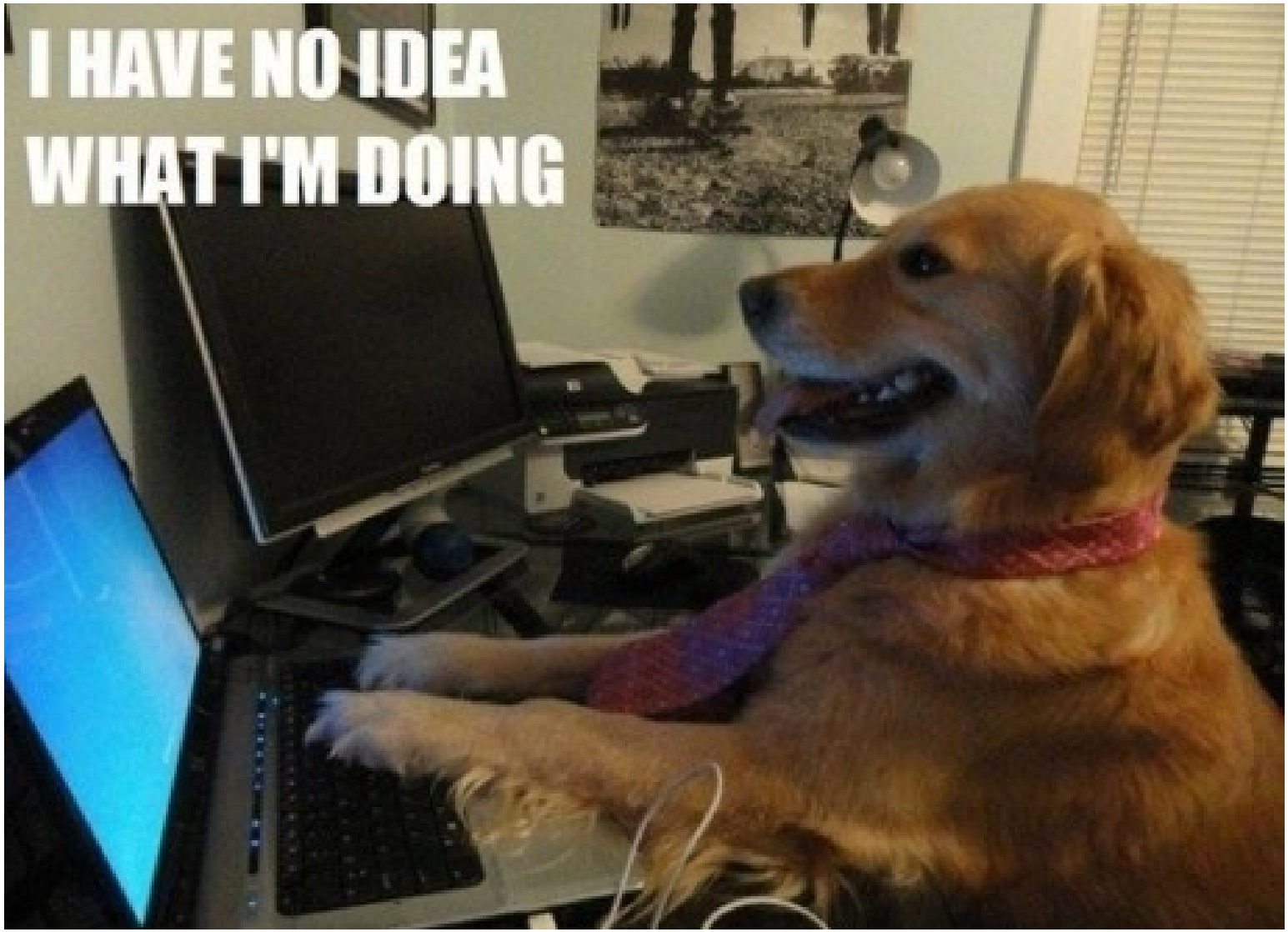
Take-home message

Two real-world examples to demonstrate that

- Scientists have many kind of computational needs
- Some of them run on cloud infrastructures
- and they are happier than before...

Disclaimer

Disclaimer

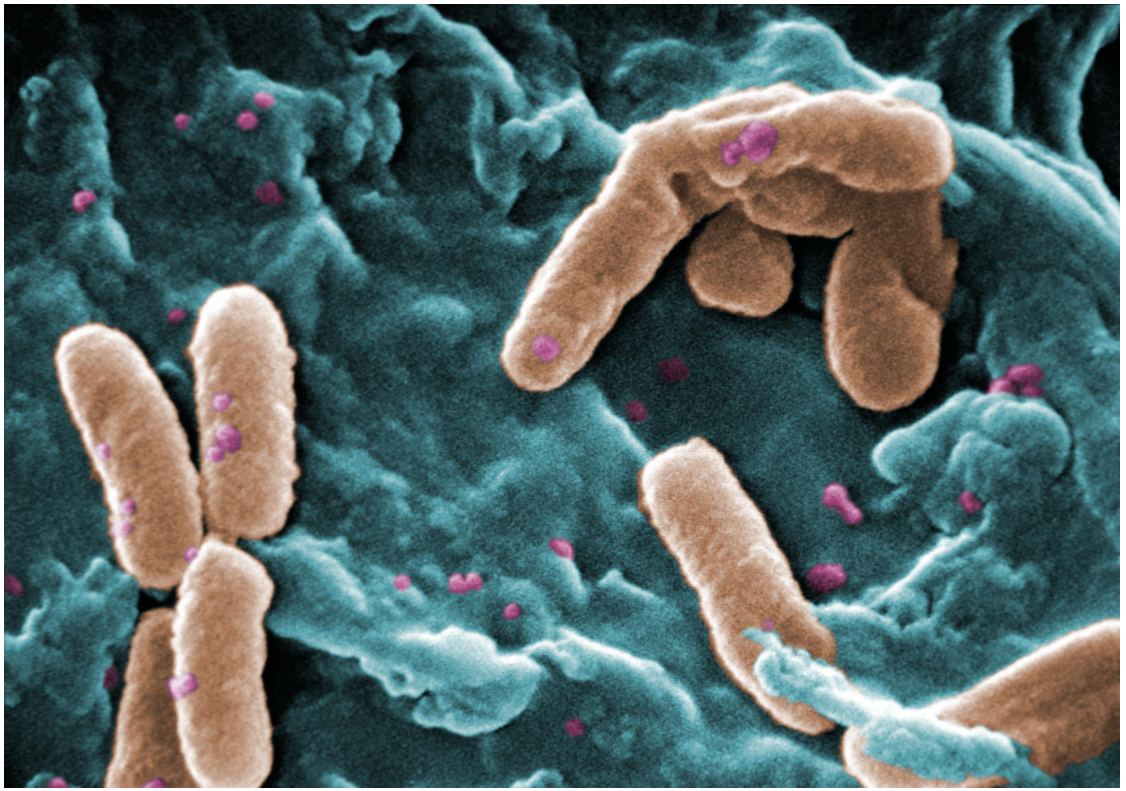


Akos Dobay

Institute of Evolutionary Biology and Environmental Studies

University of Zurich

- Epidemiology and infectious diseases
- Evolutionary biology
- Gene regulation and epigenetics
- Computational and theoretical biology
- Computer-based modelling of biological systems



Pseudomonas Aeruginosa

Like many other bacteria, they

- **produce** molecules (public goods)
- public goods goes around and bind to environmental *resources* (e.g. iron particles)
- the bacteria then **consume** these public goods

Cheaters

- Some bacteria are *cheaters*: they only consume public goods.
- . . . like people who do not pay taxes.
- If there are too many cheaters, the population will face extinction.
- Otherwise, the population will grow until there is physical space.

The goal is to study the evolution of this kind of systems, possibly without having a lab full of *in vitro* cultures.

Evolution of a bacteria population

This movie will show a simulation of a population of bacteria.

They produce public goods, consume them, reproduce and die.

What decides the evolution of the population?

Many factors have to be studied:

- environment
- initial disposition of the cells
- production rate of public goods
- durability of the public goods
- death rate of the cells
- reproduction rate

biointeract

- compute random starting conditions
- run 2000 simulations for one set of parameters
- accepts 4 different parameters from command line
- configurable number of cells
- output of multiple simulations are later used for statistical analysis.

For each parameter, we investigate $\approx$ 10 different values.

which means $\approx$ 1000 different runs

biointeract (2)

Computational needs:

- each simulation takes at least 1 minute (150 cells)
- each run takes from 12hours to 3 days (depending on the number of cells)
- no input files
- small output file
- low memory consumption
- single-core

gbiointeract

- compute all the combinations of parameters
- run on one or more cloud and on batch systems

Why not HPC clusters

(g)biointeract can run *also* on HPC clusters.

However, it would be a waste of precious resources:

- no need for a fast parallel storage
- no need for a fast, low-latency (and expensive) network
- no need for a big storage
- no need for an SMP machine

parameter sweep

- Periodically re-run an application/a workflow when new data is generated and coping with peak cycles.
- Process large data sets, requiring large-scale tightly coupled systems, large memory systems of large, fast scratch storage.
- Modify input parameters applied to a computational model, which requires any combination of large-scale tightly coupled system, large memory system, large fast scratch storage.
- Assess a range of parameter combinations based on a prototypical embarrassingly parallel use case, e.g. high energy physics.

Other parameter sweep use cases (1)

Vanessa, *Alpine cryosphere-related slope movements: measurements and analysis.*

- Data coming from GPS stations planted on the Alps.
- they send information on their position and inclination
- used to understand movements of the slope

Small set of R scripts to calibrate the geographical model.

- 17,598 runs
- 48,747 cpu-hours

Other parameter seep use cases (2)

Joel, *Mountain cryosphere subgrid parameterisation and computation.*

- GEOtop is a distributed hydrological model.
- GEOtop uses a DEM. The bigger and more detailed the area, the bigger is the DEM, the longer takes the simulation.
- Preprocessing system to intelligently reduce the points where GEOtop has to run
- 4 order of magnitude less computation

Need to calibrate the model:

- 8,588 runs
- 114,000 cpu-hours

Dr. Michal Okoniewski

Marek Wiewiorka

Functional Genomics Center Zurich

- Data analyses in genomic and transcriptomic projects.
- Support of experimental design, statistical analysis, data integration and annotation processing.

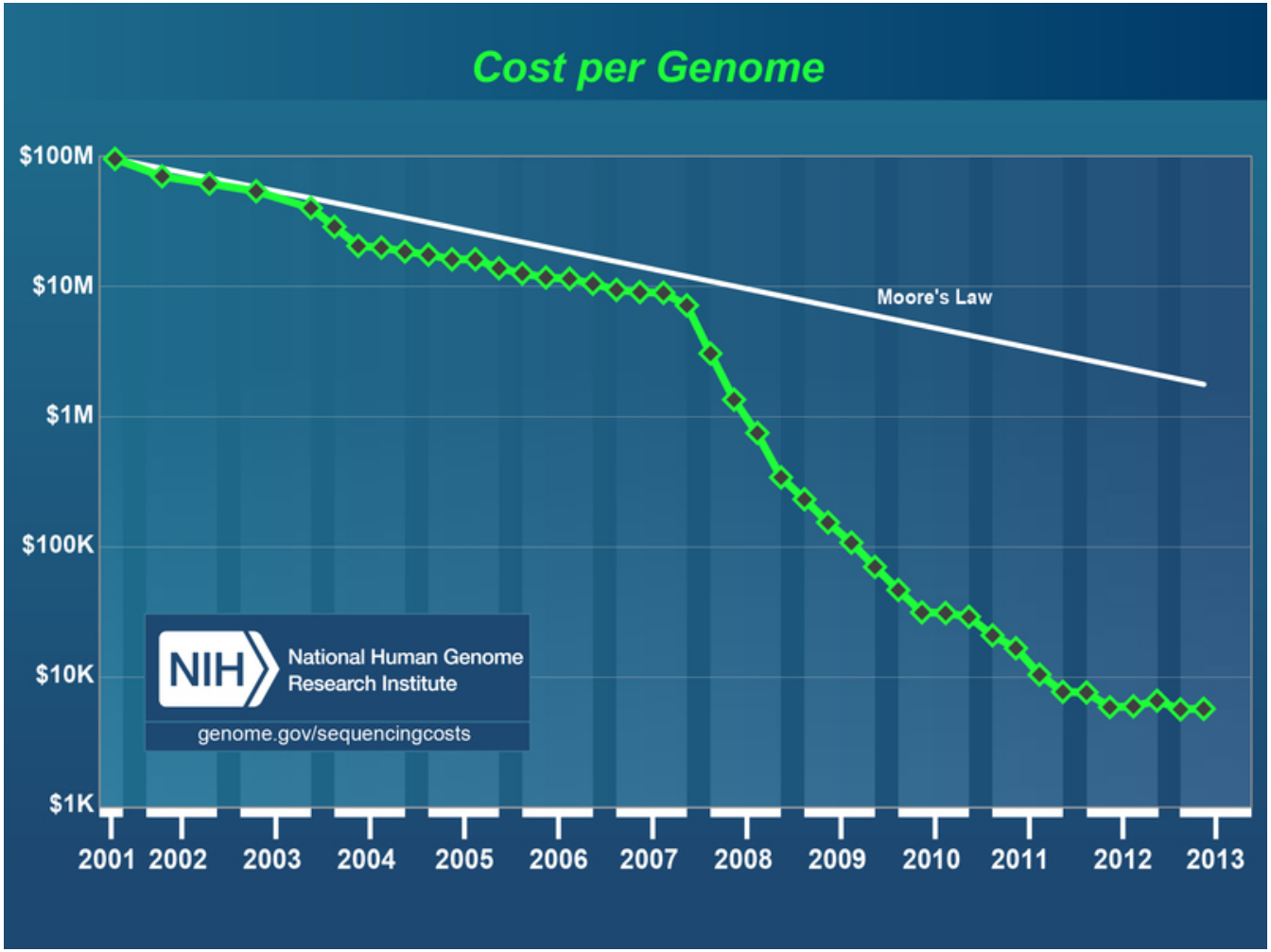
Did you know that horses can suffer from asthma?

What makes this

into this?

RNA and DNA sequencing is needed to be able to answer this kind of questions.

Sequencing machines prices is dropping



Cost per genome. link

RNA sequencing

Illumina HiSeq 2000

- blood sample from horses
- sequenced to a raw file
- then converted to a BAM file (1GB big)
- a BAM file can contain a full genome
- FGCZ has already 300 samples.

Goal is: finding genes that switch on or off in case of horse asthma.

RNA sequencing

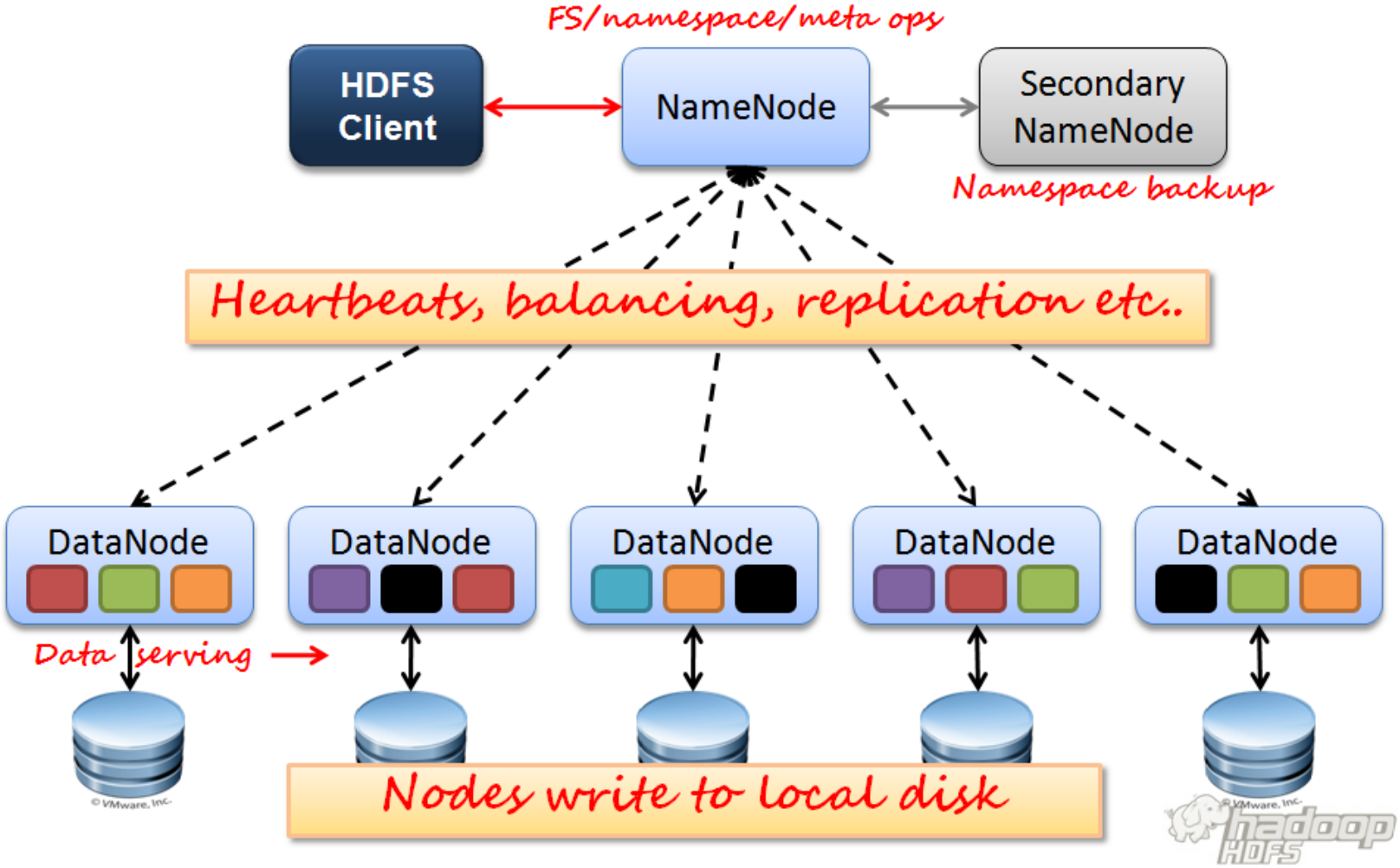
BAM files contains information on RNA presence and quantity from a genome at a given moment in time.

RNA continually changes, as opposed to static genome. RNA sequencing is useful for instance to observe cellular pathway alterations during infection and gene expression level changes in cancer studies.

Having multiple samples, it's possible to do statistical analysis, aggregating reads mapped to a specific genome region in genes or in exons, thus looking for *patterns*, and find specific shapes of coverage of the genome, and statistically significant differences between them.

Computational infrastructure

Hadoop is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models



Hadoop

- HDFS spreads data over multiple (thousands) servers
- each machine offer local computation and storage
- jobs are sent to the machines which holds the data to compute (**data locality**)
- no need for hardware high-availability, failures are detected at application level
- the cluster can be easily grown or shrunk
- HDFS works very well for mostly immutable files.

Implementation

- Queries are written in Scala (in the future also R will be supported)
- the framework generates multiple jobs from the query
- jobs are sent on the nodes of the Hadoop cluster
- each job works on some of the BAM files stored on that specific node
- results are then collected and aggregated

Deployment of the Hadoop cluster on the cloud is done using elasticluster

Why not HPC clusters?

- In the future, near real-time computations will be performed, which do not play well with batch systems
- thanks to HDFS, no need for parallel storage: data is read locally.
- no need for fast network interconnect
- a framework which uses Hadoop is used, that allows for easy and fast implementation of statistical queries over the dataset

Thank you

and many thanks to Akos, Michal and Marek!

Questions?