EDIM1: Introduction to the new Data Intensive Machine

Paolo Besana
University of Edinburgh

DIR weekly seminar – 1 April 2011

New Data Intensive Machine

- One hundred nodes. Each node:
  - Intel Pentium 66Mhz
  - 16Mb of SIMM RAM
  - 1.6Gb Seagate Hard Disk
  - 10Mb/s fast Ethernet board
  - £1623 per node
- Software stack:
  - Linux Kernel 1.2.0
  - GCC 2.7.0

Well, sort of…

- That would have been April 1995…
- Now machines are a bit more powerful, and a bit cheaper:
  - Intel Atom dual core, 1.6Ghz
  - 4Gb of Ram
  - 6.256 Tb of HD
  - 1Gb/s connection
  - Less than £1000 per node (main cost being the disk)

Presentation structure

- Paolo Besana (DIR, Informatics):
  - Data intensive problem
  - EDIM solution
  - Dealing with large datasets
  - Test applications
- Adam Carter (EPCC):
  - EDIM1 configuration details
  - Software stack

What is Data Intensive?

- Data volume is increasing:
  - Cost of sensors is decreasing (better CCDs, at lower price, higher speed sensors,…)
  - Ubiquitous networks allow collection and access to data
  - Better HPC allows more detailed simulations (for example, fine grained simulation of geodynamics or climate)

Where is the problem?

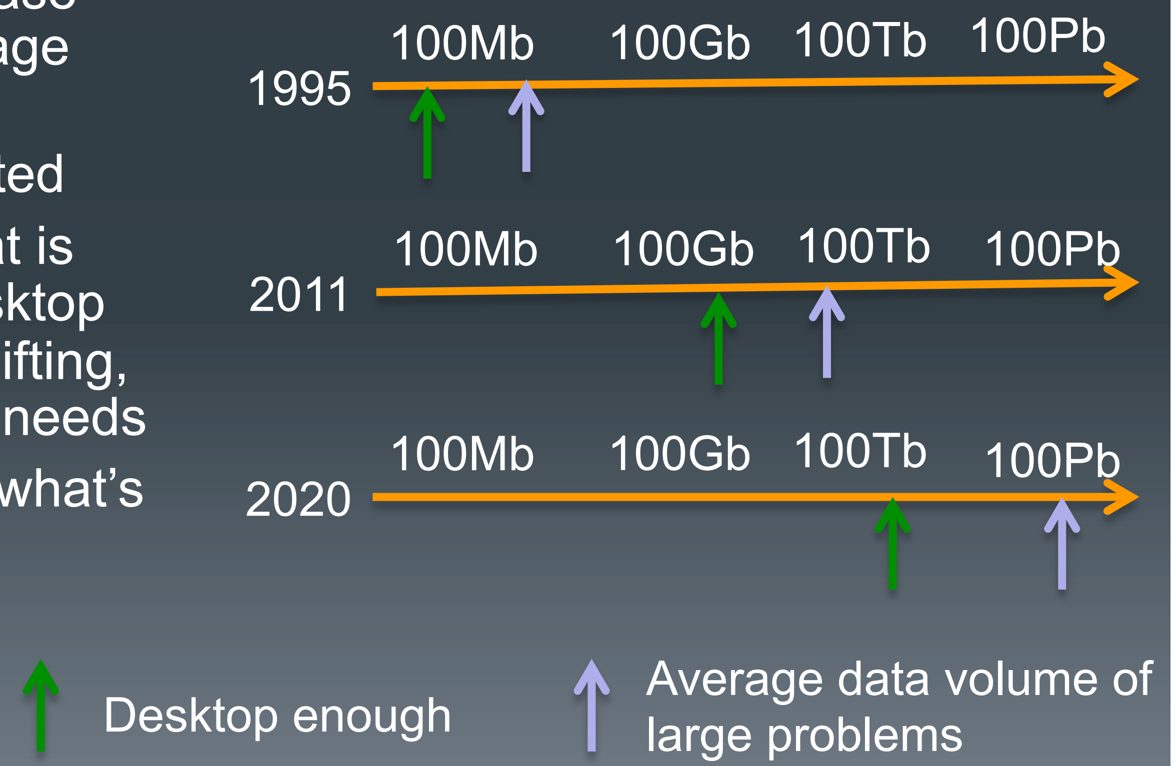
- Processing large amount of data is becoming the bottleneck:
  - CPUs process more data than can be transmitted
- HPC systems oriented towards:
  - computationally intensive tasks (such as simulation, for example)
  - Not data intensive tasks

Can we not care?

- Will technology solve the problem?
- Volumes problematic 5 years ago can be dealt on a desktop
- What is problematic now will be easy in 5 years
- As shown in the example, machines grew in power, and decreased in cost

We probably have to care

- Sensors costs decrease more than CPU/storage costs
- More data are collected
- The threshold of what is computable on a desktop machine is simply shifting, but stays behind the needs
- Techniques apply to what’s above the threshold



Are we making up a problem?

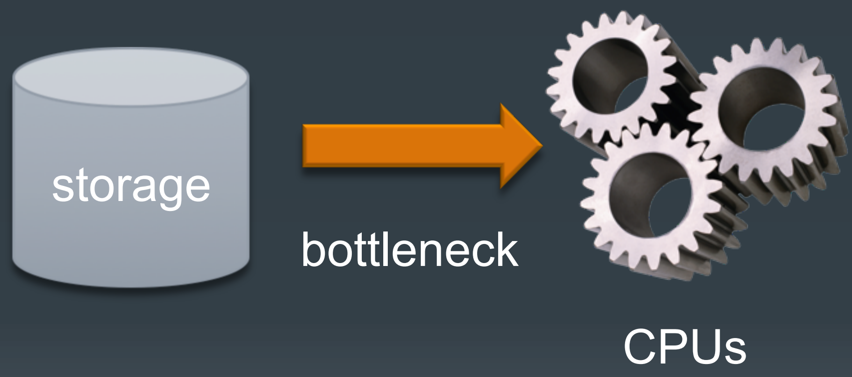
- Will we reach a limit of data volume useful to collect?
  - All library of congress is estimated to have 3 petabytes of data
  - iTunes has 13,000,000 songs (~70Tb)
  - We are "only" 7 billion, and we will grow to 9 billion by 2050: total amount of information we produce is bounded by that (DNA for each inhabitant is bound, images)
- Probably not, at least for a while
  - We always find new ways to produce new data…
  - Did you imagine 10 years ago that 1Tb was just enough for your pictures and videos?

Power consumption

- Data centres absorb large portions of electrical power:
  - To power the machines
  - To cool the machines
  - And costs are increasing
  - prediction that most of the cost of a centre will be power, not hardware
- For example, CWI in Amsterdam is buying a 1M€ cluster
- They expect to spend the same amount in the next 3 years in electricity

In search of a solution

- Traditional model of processing separates processing from storage: it may not be efficient
- It surely very expensive
- Transfer between storage and processing can cause a bottleneck:
  - increasingly fast processing nodes do not correspond to transfer speedup



An alternative

- A data-intensive experimental machine has just been delivered to Edinburgh DIR group
- Ideas from Jim Gray and Alex Szaley:
- Create a network of “*data-bricks*”:
  - low consumptions node, with large storage capability
- Aims at processing data locally, reducing the need to transfer data to processing nodes

Working with large dataset

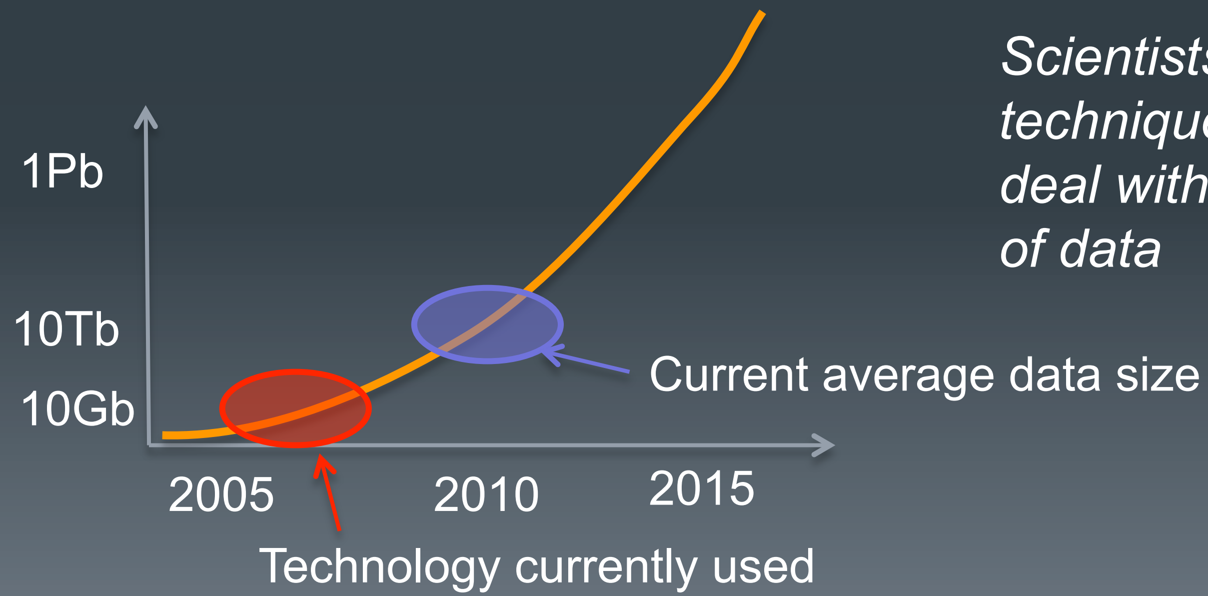
- We have discussed the hardware problems:
  - Bandwidth and power consumption
- Improving hardware is not enough
- Need to study software approach to deal efficiently with large dataset:
  - Minimise data transfer
  - Use streaming to process data as soon as it starts arriving

Transition phase

- Many scientists are still handling data with older technology and some distributed infrastructure (grid, “cloud”, clusters)
- Techniques awkward but still work, but will not scale:
  - Good timing for identifying and proposing new methods

Current situation

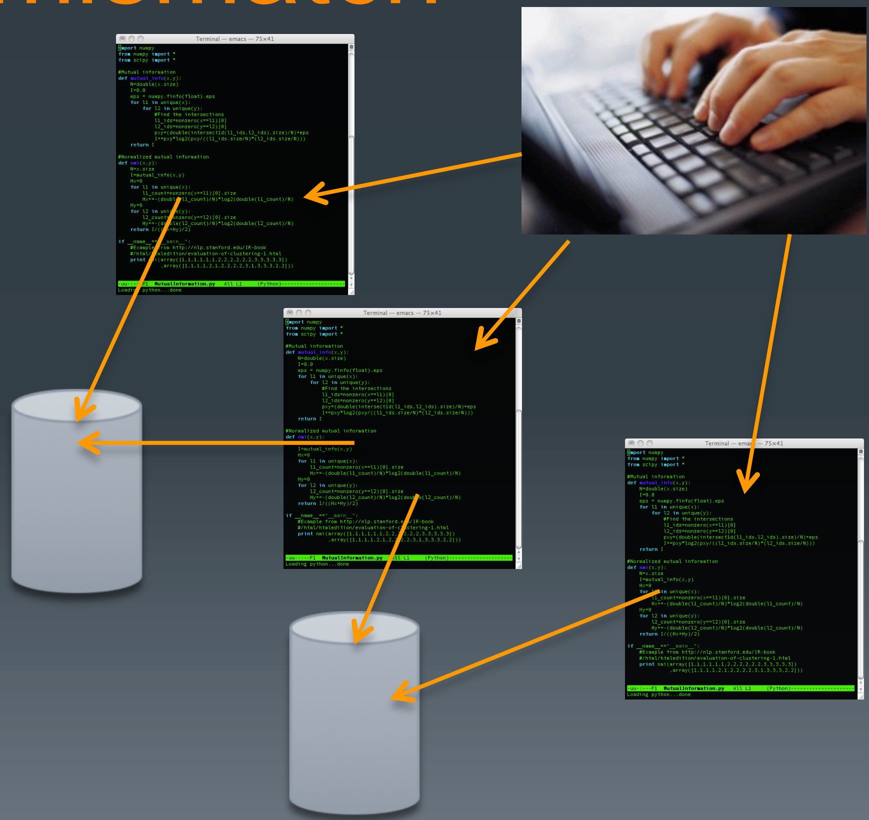
- Talked with scientists to understand what computations are performed, what are data requirement



Scientists are still using techniques developed to deal with smaller volume of data

Technique-data mismatch

- techniques are often scripts, developed years ago, manually run from the shell.
- Data is transferred manually from storage to processing clusters/nodes, scripts launched, result collected and transferred back manually
- Still work, but awkward:
  - Current techniques will not scale
- Good timing for studying a scalable solution



Test beds - 1

EDIM1 will be used on a set of scientific applications that mine information from large amount of data.

Application that handle images are data intensive:

- *Microscopy*: storage/access/rendering of images, analysis of time sequences
- *Gene expression* in mice embryo
- *Cosmology*: galaxy shape, lensing
- *Brain imaging* from MRI
- *Astrophysics*: quasars analysis

Test beds - 2

- But not only images
- *Gene clustering* for breast cancer type detection
- *Seismology*: wave front correlation

Gene interactions

What genes interact during mouse embryo development?

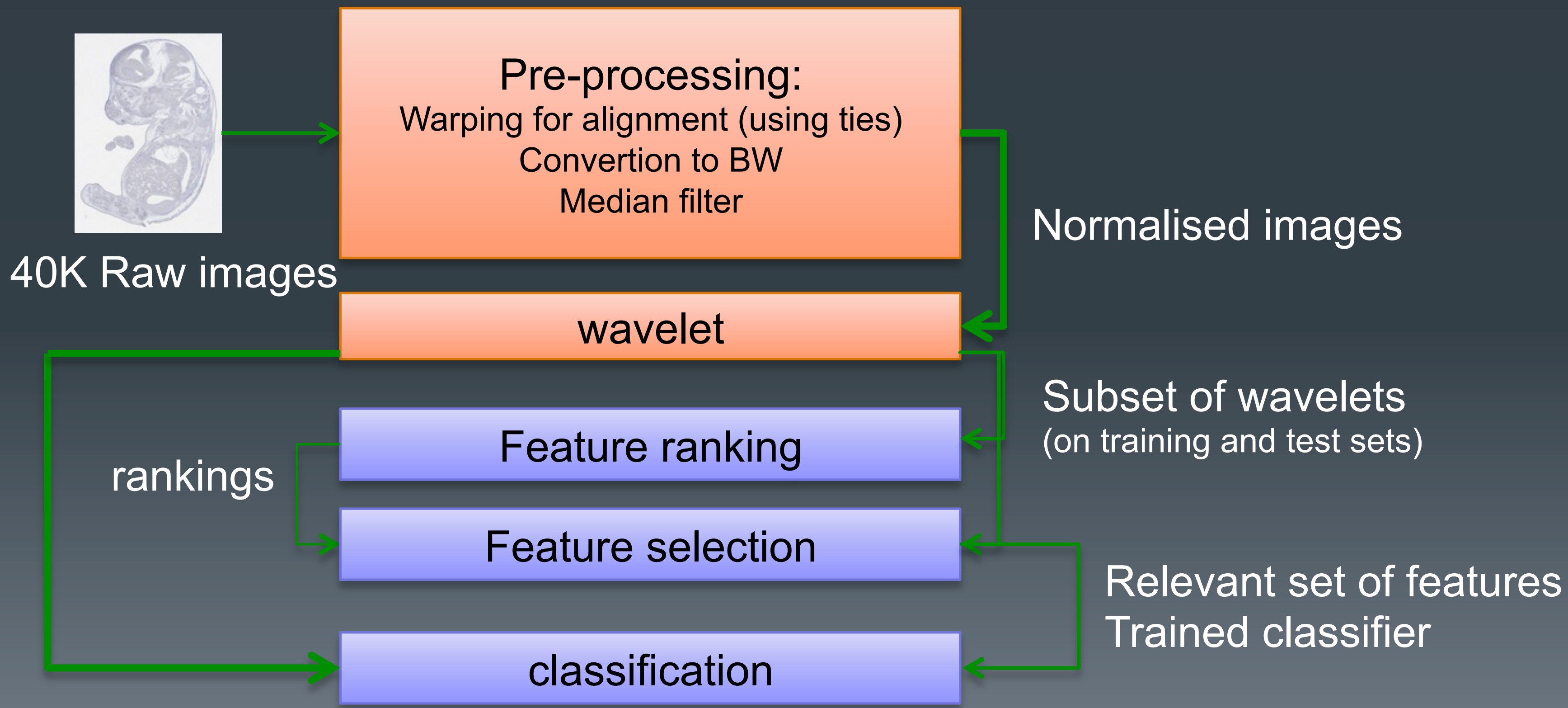
- Question tackled in two phases:
  - An exploratory phase focused only on transcription factors (around 1600 genes: involved in the transcription of protein from corresponding DNA code)
  - A full analysis of the genes (around 15000)
- Use images from Eurexpress: slices of mice embryo treated with gene markers: if a gene is matched by the marker, there is a darker spot.
- Previously, subset of subsampled images used to find gene expressions in anatomical parts

Gene expression interactions

- Each image is one slice marked with a marker for a gene
- Interaction when two genes are expressed in same area in different slices
- Need to find overlapping between areas
- Result of the process, a set of rules specifying:

  Gene *i* interacts with gene *j*
- No reference to anatomical area

Gene expression - workflow



Gene expression computation

- Preprocessing and wavelet computation is "embarrassingly" parallel:
  - Can put raw images in separate nodes, warp, convert and compute wavelet
- In feature ranking and extraction, only the wavelets computed for the training sets need to be transferred on a node for computation:
  - Mutual information
  - Correlation
- In final classification, a full pair-wise analysis is required:
  - However, only the relevant features can be transferred

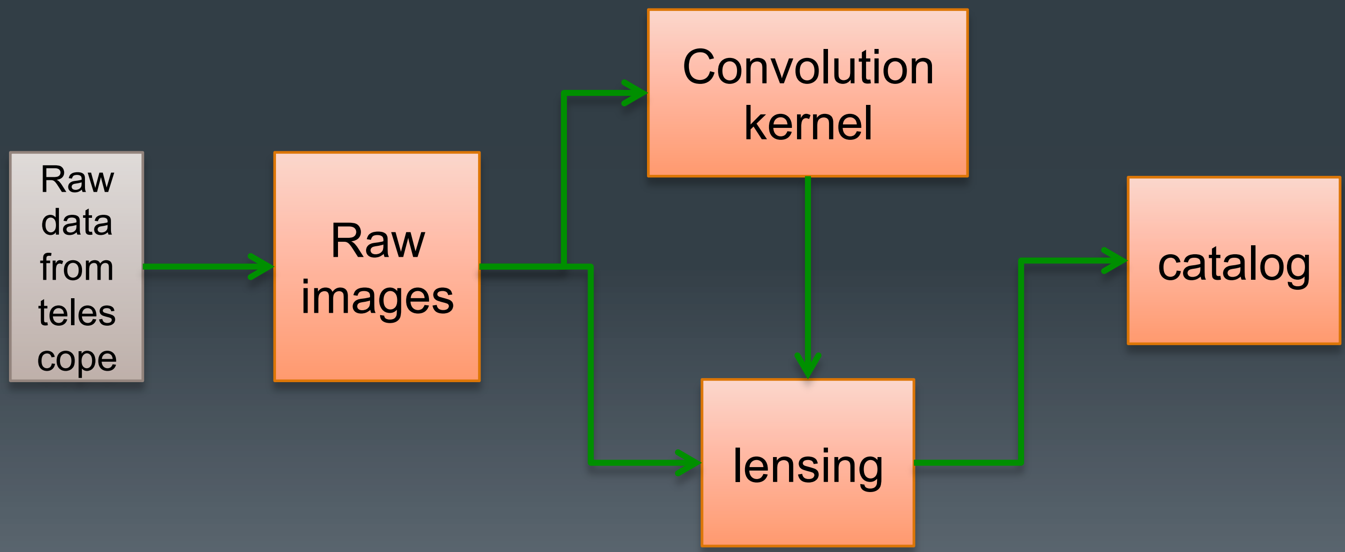
Gene expression resources

- Processing elements are:
  - Unix executables (some from ImageMagick)
  - Perl scripts
  - Python scripts (require numpy, PIL and pywavelets)
  - R scripts

Cosmology

- Analyse the shape of galaxies, and their statistical properties:
  - Aim is to verify the results of cosmological simulations
  - Does the result of a simulation match the real distribution of mass in the universe?

Cosmology workflow



30 billions of galaxies (from different channel – light, radio, …)

Cosmology resources

- Processing elements are in:
  - Executables in C, C++ or Fortran
  - BASH shell

Microscopy

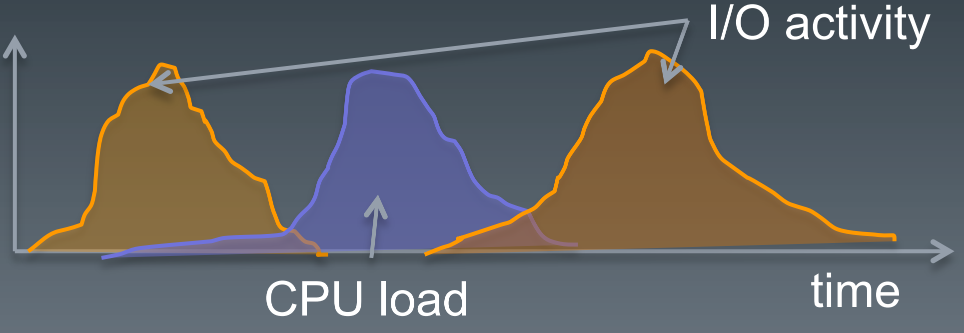
- Studies of architectural innovation for OMERO, current platform for microscopy imaging.
- Two projects:
  - Cluster for intensive computations (on top of OMERO)
  - Alternative technologies for image storage and retrieval (array-based databases, hadoop)
- Mainly java libraries

Problem of managing the experiments

- Each experiment requires different stack of software
- Experiments require initial distribution of data
  - Is uniform allocation good enough?
  - It likely depends on the problem
- We are measuring performances of the system for problem classes:
  - One experiment at the time can run on a node: otherwise there would be interferences and resources would be shared

Measurements

- We will try to measure performances to verify whether we get close to goals
  - Data transfer between nodes
  - CPU usage
  - Power consumption
- What we do not want see:



Thanks!

Questions after Adam