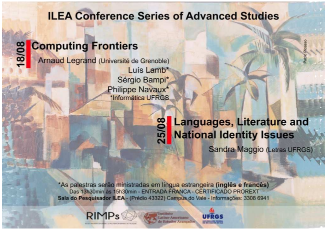
Large Scale Computing Infrastructure Challenges

Arnaud Legrand

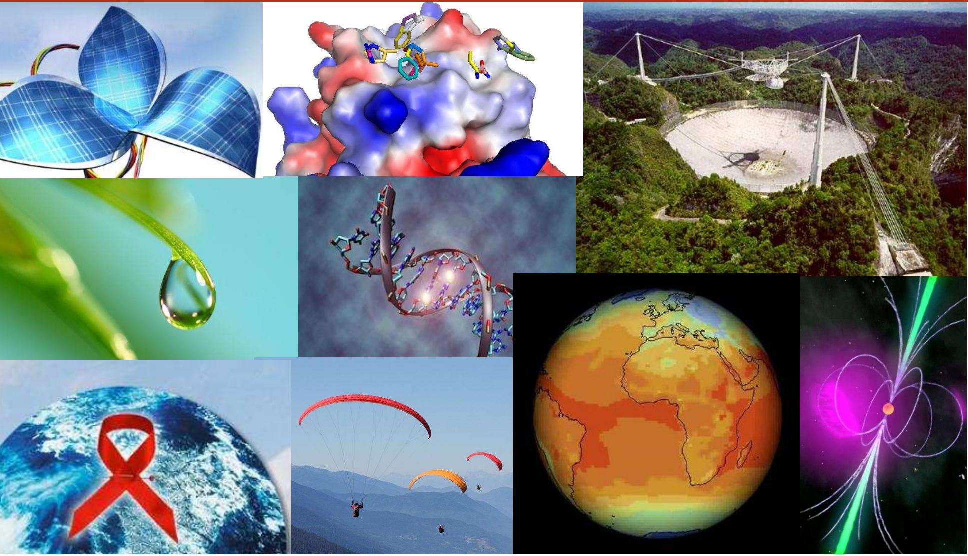
CNRS, University of Grenoble

August 2015

Instituto Latino-americano de Estudos Avançados – UFRGS

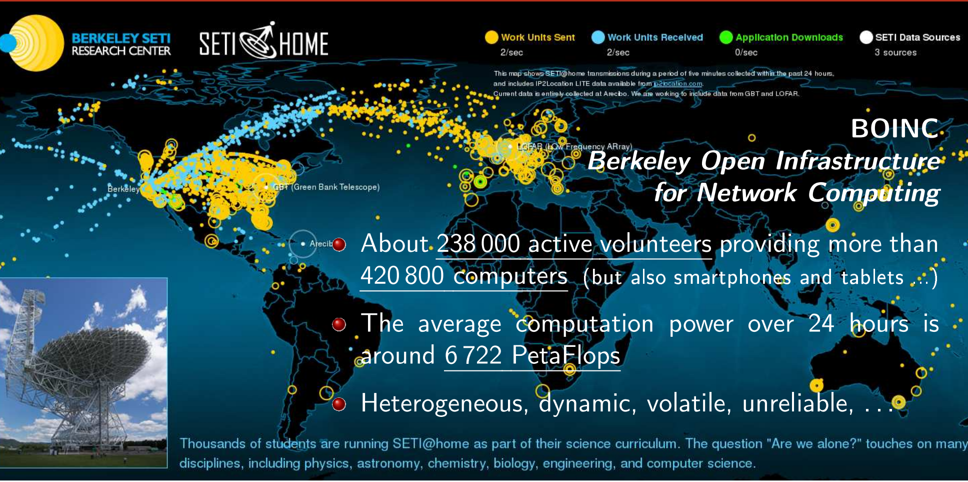


What do . . . have in common?



Clean water, solar cells, new drugs against Ebola/AIDS/Cancer, climate evolution, weather forecast for paragliding, searching for Extra-Terrestrial Intelligence, pulsars,

Volunteer Computing



BOINC

Berkeley Open Infrastructure for Network Computing

- About 238 000 active volunteers providing more than 420 800 computers (but also smartphones and tablets ...)
- The average computation power over 24 hours is around 6 722 PetaFlops
- Heterogeneous, dynamic, volatile, unreliable, ...

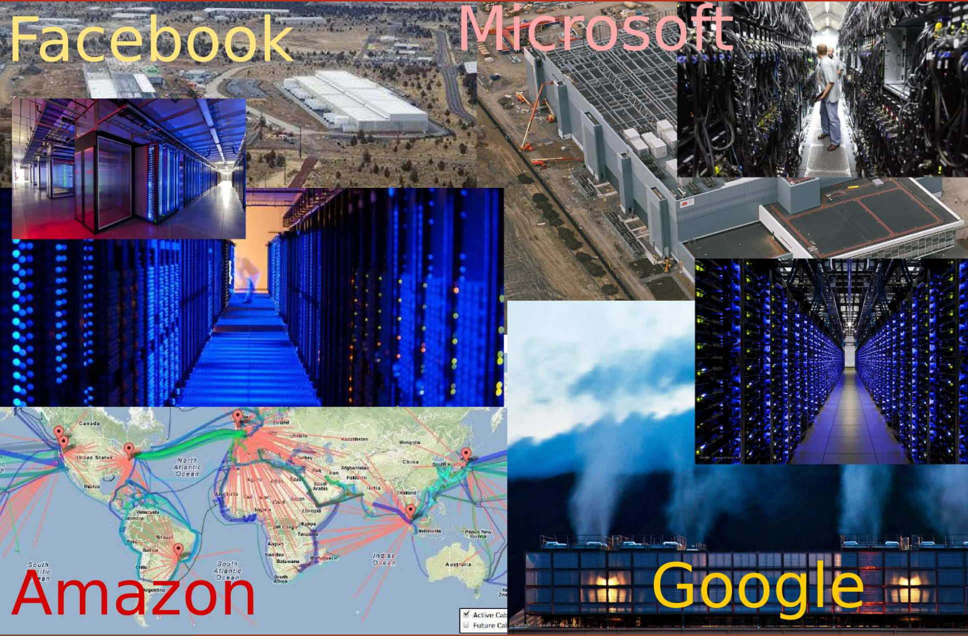
Today the computer is just as important a tool for chemists as the test tube. Simulations are so realistic that they predict the outcome of traditional experiments

– Nobel committee (chemistry), 2013

Supercomputers

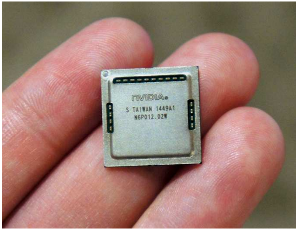
- 100,000 to 1,000,000 cores with accelerators (GPU, Xeon Phi) and a high throughput/low latency interconnection network
- An international race (Top500)

The Cloud


Facebook
Microsoft
Amazon
Google
Canada
North Atlantic Ocean
South Atlantic Ocean
Indian Ocean
China
Mongolia
Kazakhstan
Australia

A Breathtaking Evolution

Hybrid and very large scale parallel architectures to answer computation needs in restricted power envelopes.

1996	2009	2015
		
ASCI Red	ATI Radeon	Nvidia Tegra X1
1 Teraflop	2.4 Teraflop	1 Teraflop
9298 Pentium II	1600 Stream Processors	8-core ARM CPU
1 000 Flops/W	1 600 000 Flops/W	667 000 000 Flops/W

My smartphone is as powerful as a 20 years old supercomputer

Embedded Systems, Sensor Networks, Internet of Things . . .

Cooking analogy

Exploiting Sequoia = **6 million threads** constantly exchanging data!

How can we even conceive a code for such a monster?

Computers are very very fast but also very very dumb

- Computers make very fast, very accurate mistakes
- Computers are not intelligent, they only *think* they are 😉

So coding means writing very, very detailed instructions in a paranoid way

Somehow, we can compare coding to writing a recipe, and running the code like cooking

Preparing a Piperade

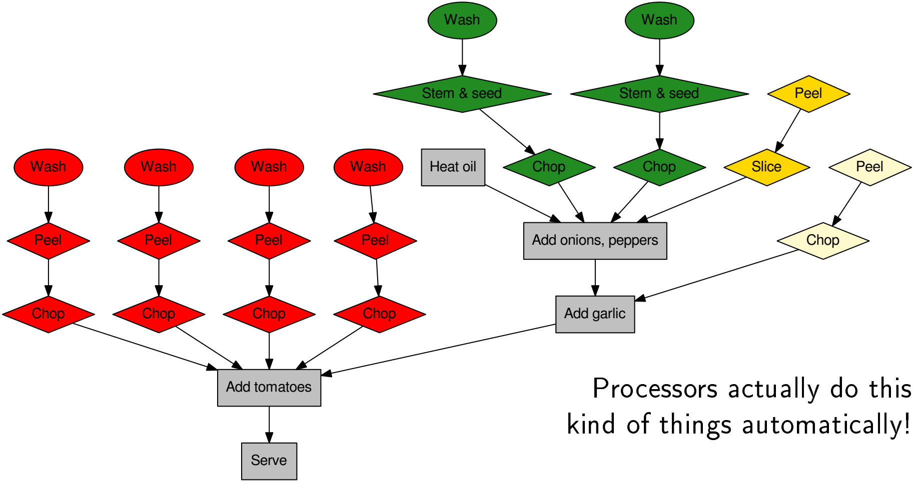
Preparing a Piperade

For 2 persons:

- Wash, Peel and roughly chop 3 plum tomatoes
- Wash, stem and seed, and roughly chop 2 bell peppers
- Peel and chop 4 garlic cloves
- Peel and thinly slice 1 onion
- In a 12-inch skillet over medium high heat, heat olive oil until hot
- Add onions, peppers and $\frac{1}{2}$ teaspoon salt and sauté, stirring frequently, until onions are translucent and peppers have started to lighten in spots, about 10 minutes
- Add garlic and continue to sauté for 1 more minute
- Stir in tomatoes and piment d'Espelette
- Reduce heat to medium, cover and cook until tomatoes are starting to fall apart and peppers are soft but still hold their shape, about 15 minutes
- Remove cover and continue to cook, stirring frequently, until mixture thickens like a slightly runny relish, about 5 minutes more

Cooking = executing the recipe in the *right* order

Note that you should probably change the order to reduce the preparation time



Processors actually do this kind of things automatically!

Going even faster..

This recipe is perfect if you are a single cook and have 1 kitchen sink, 1 knife, 1 cutting board, 1 skillet

- If your wife helps you out and has her own knife and cutting board, you will easily go faster

But if you want to prepare for 200 persons, it will still take a lot of time ☹

Going even faster..

This recipe is perfect if you are a single cook and have 1 kitchen sink, 1 knife, 1 cutting board, 1 skillet

- If your wife helps you out and has her own knife and cutting board, you will easily go faster

But if you want to prepare for 200 persons, it will still take a lot of time ☹

Would asking 40 friends to help really help ?

- Maybe... provided they have their own cutting boards, knives, and you manage to fit them all in your small kitchen
- You will also probably need additional skillets

Coordination is however going to be a nightmare, especially if they do not all chop/peel/slice at the exact same speed

Solving bigger problems requires more processors, more memory, a fast communication medium, and a thorough organization that takes the right decisions on the fly

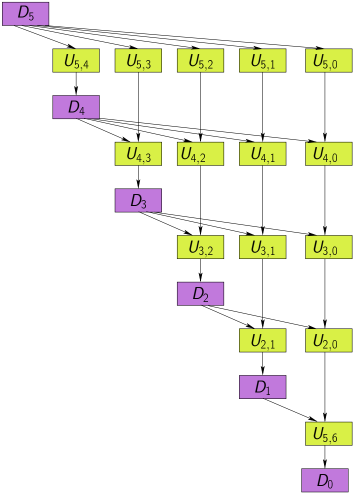
And is it really the same with computers?

If I_1 writes z and I_2 read/writes z, then I_1 and I_2 should be done in the right order [Bernstein66]

Data define dependencies between instructions/tasks

```
import numpy as np
A = np.array([[1, 1, 1, 1], [0, 1, -3, -1],
     [0, 0, 6, 1] , [0, 0, 0, 4]], float)
b = np.array([6, 5, -4, 8], float)

n = len(b)
x = np.zeros(n, float)
S = np.zeros(n, float)
for j in reversed(range(0,n)):
  x[j] = (b[j] - S[j]) / A[j][j]   # pivot            (D_j)
  for i in range(0,j):             # parallel loop
     S[i] = S[i] + A[i][j] * x[j]  # update          (U_i,j)
```



And is it really the same with computers?

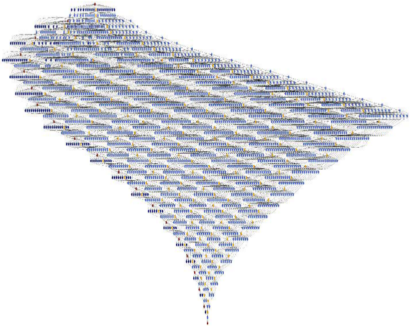
If I_1 writes z and I_2 read/writes z, then I_1 and I_2 should be done in the right order [Bernstein66]

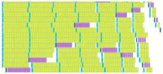
Data define dependencies between instructions/tasks

```python
import numpy as np
A = np.array([[1, 1, 1, 1], [0, 1, -3, -1],
    [0, 0, 6, 1] , [0, 0, 0, 4]], float)
b = np.array([6, 5, -4, 8], float)

n = len(b)
x = np.zeros(n, float)
S = np.zeros(n, float)
for j in reversed(range(0,n)):
  x[j] = (b[j] - S[j]) / A[j][j]    # pivot                  (D_j)
  for i in range(0,j):              # parallel loop
     S[i] = S[i] + A[i][j] * x[j]   # update               (U_i,j)
```

- adapting granularity
- optimized code versions for each resource (CPU/GPU/auto-tuning)
- dynamic load balancing
- portable performances





As a Conclusion

Our society (citizens, companies, science, . . .) relies (often obliviously) on gigantic computation infrastructures

How to design/use/optimize/understand such infrastructures?

- Scalability
- Fair sharing
- Fault tolerance
- Capacity planning
- Energy consumption
- Modeling/analysis/evaluation/experimentation

Similar issues with any large distributed infrastructure

- HPC/cloud/. . .
- Wireless networks
- Smart grids
- Transportation systems