Lenstool-HPC

A high performance computing approach to lens-modelling

C. Schaefer, M. Rexroth, G. Fourestey, J-P Kneib

Collaboration between LASTRO and SCITAS

Lenstool

- Parametric mass-modelling of gravitational lenses using Bayesian MCMC sampler
- Multiply imaged sources serve as constraints:

$$\chi_i^2 = \sum_{j=1}^{n_i} \left[x_{obs}^j - x_{pred}^j(\theta) \right]^2 / \sigma_{ij}^2$$

- Parameter space is huge
- Process can take several weeks.

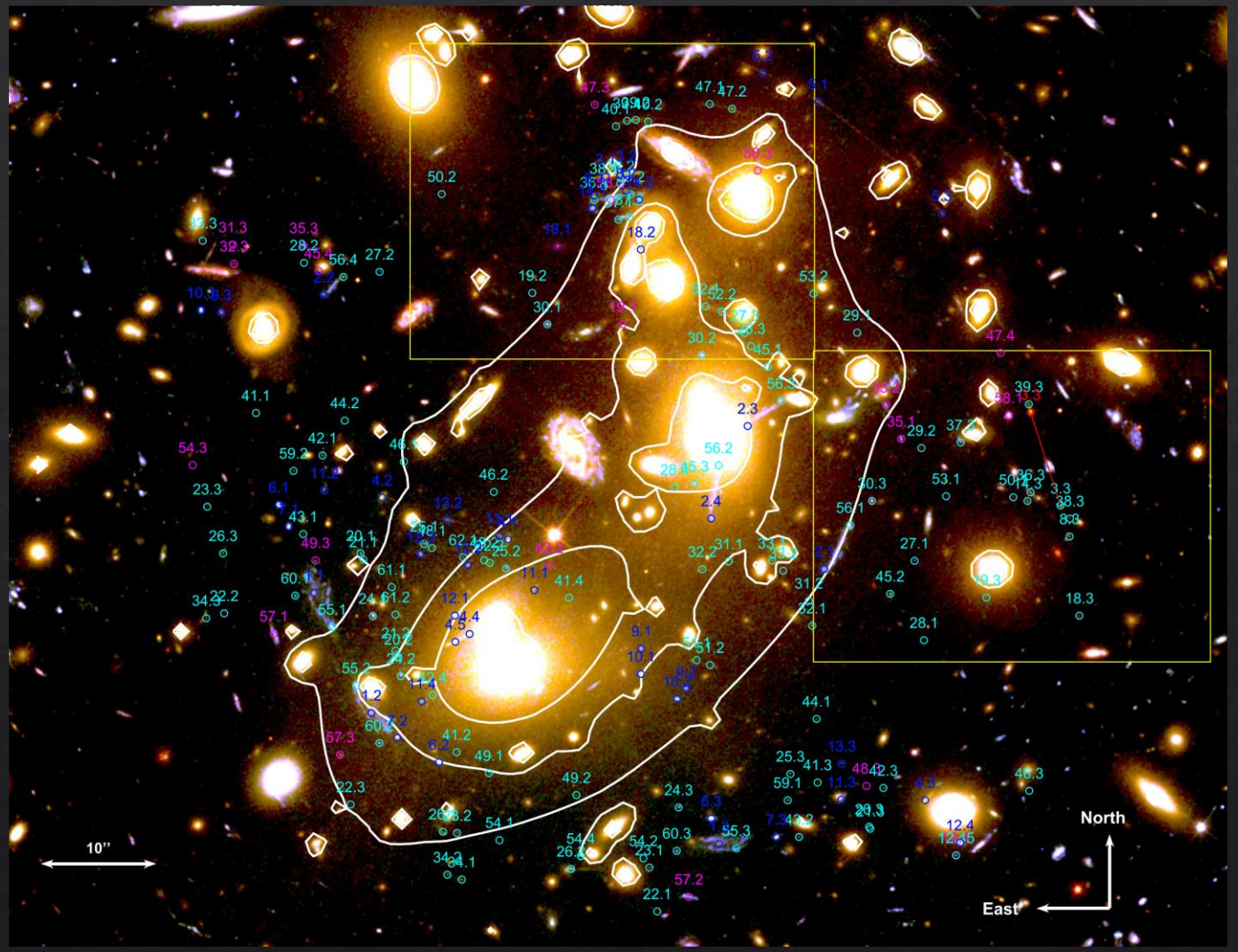
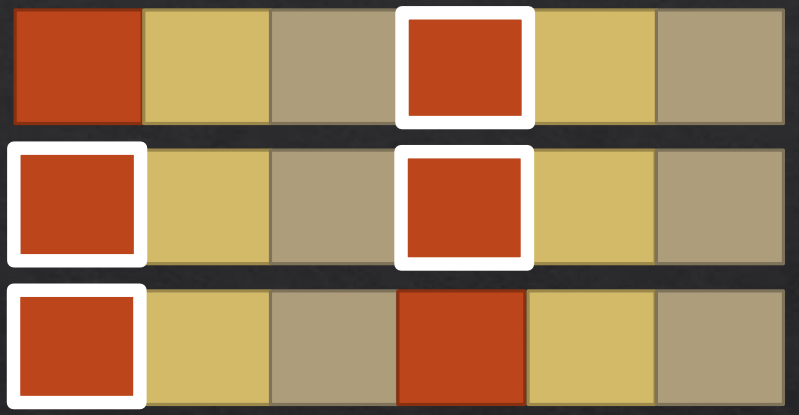


Fig. 1 Credit for Image: Jauzac et al. 2015b

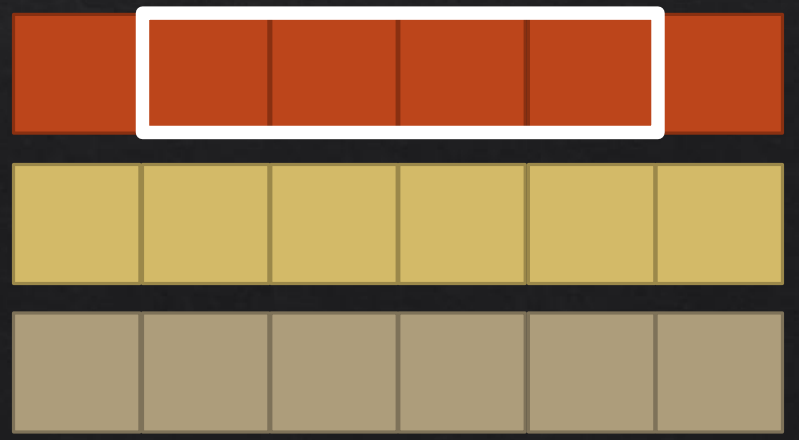
Lenstool-HPC

- Goal: Speed-up Lenstool by re-coding Lenstools mass-modelling capability using HPC techniques for HPC clusters.
- Challenge: Gradient computation of the deflection potential for each mass-model.
- A typical case of SIMD (Single Instruction, multiple data) operations.
- SOA layout VS AOS layout:
  - SOA components are stored contiguously in the memory, so only 1 memory access is needed to feed the CPU.
  - With AOS, memory jumps are needed, since CPUs do not load all the memory at once, and you need to "pack" your data in a vector.

Array of structure (AOS)



Structure of array (SOA)



Data Level Parallelism

- Modern processors are able to process multiple SIMD instructions simultaneously (DLP)
- Lenstool-HPCs SOA implementation shows a significant speed-up (1.4 at worst, 2 at best) already only due to easier memory access
- When allowing DLP, on AVX2 capable machines (vector length of 8) we get a speed-up of 10