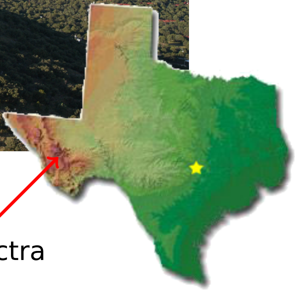
Simulating a Bunch of Galaxies for HETDEX

Henry Gebhardt

Department of Astronomy and Astrophysics,
The Pennsylvania State University

Astro 585 — HPC and Astronomers?
April 23, 2014

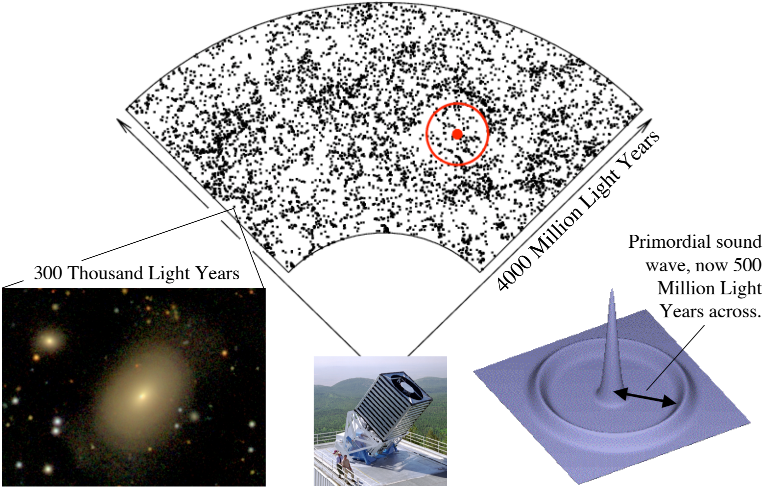
Hobby Eberly Telescope Dark Energy Experiment (HETDEX)



McDonald Observatory, 10 m primary

Every pixel is a spectrograph: 36 000 spectra

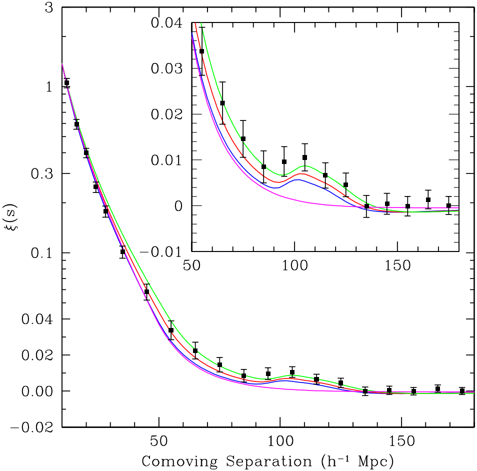
Baryonic Acoustic Oscillations (BAO)

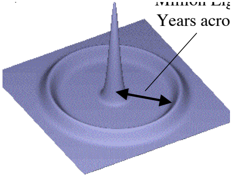


Sloan Digital Sky Survey out to redshift $z \sim 0.5$.

Eisenstein *et al.*, 2005.

Baryonic Acoustic Oscillations (BAO)





- Dark Energy
- Dark Matter
- Baryonic matter

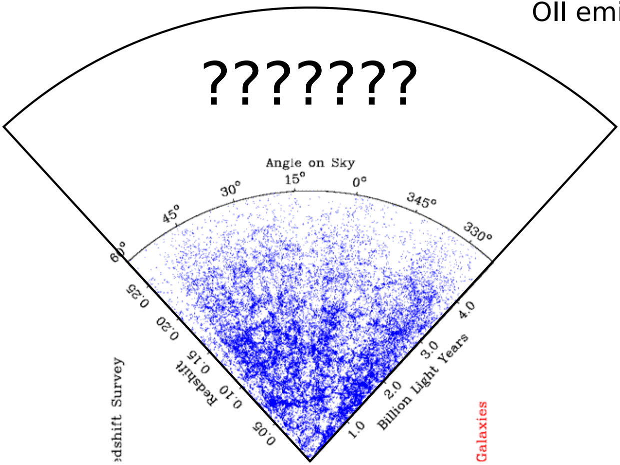
- Oscillations are a standard yardstick

Eisenstein *et al.*, 2005.

HETDEX: How does the Hubble constant change with time?

LAEs: ca. 800 000

OII emitters: 1–2 million

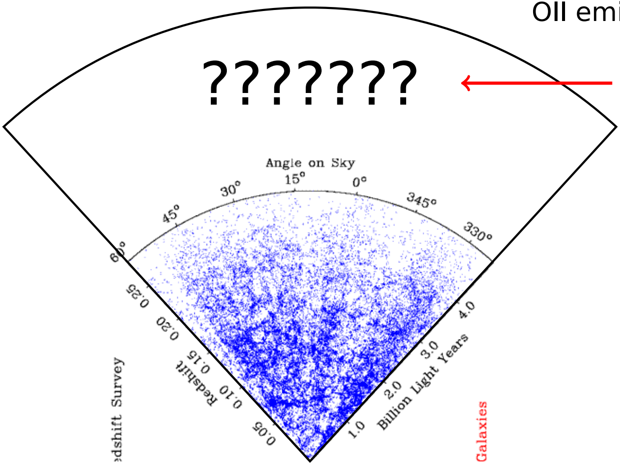


How does the view of the yardstick change as a function of redshift?

HETDEX: How does the Hubble constant change with time?

LAEs: ca. 800 000

OII emitters: 1–2 million



Lyman-α emitters

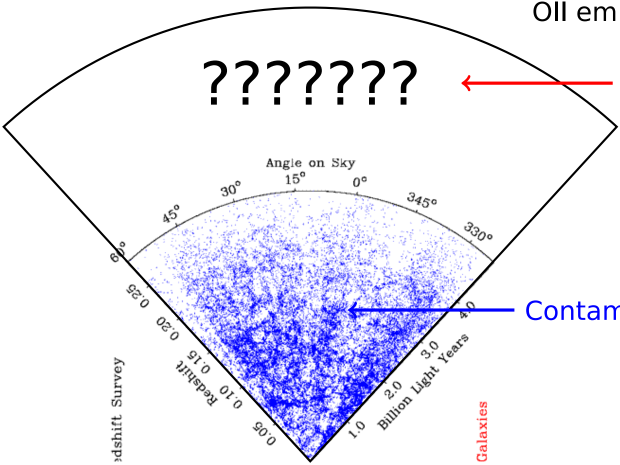
LAEs trace the matter distribution at high redshift

How does the view of the yardstick change as a function of redshift?

HETDEX: How does the Hubble constant change with time?

LAEs: ca. 800 000

OII emitters: 1–2 million



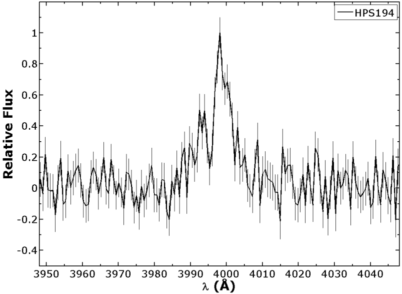
Lyman-α emitters

LAEs trace the matter distribution at high redshift

Contaminants: OII emitters

How does the view of the yardstick change as a function of redshift?

How to distinguish between LAEs and OII emitters



HETDEX will get something like 36 000 spectra per exposure.
Additionally, we will use broadband survey to get continuum flux.

$\rightarrow$ Equivalent width cut:

- LAEs > 20 Å
- OII < 20 Å

Do Bayesian classification in the future.
In any case, we need the continuum flux.

Project: Simulate a bunch of LAEs and OII emitters, and see how well we can distinguish between them

- Draw RA, Dec, z randomly from power spectrum (done by theorists)
- Draw flux randomly from luminosity function
- Plop down fake galaxies onto real images (this takes longest!)

Project: Simulate a bunch of LAEs and OII emitters, and see how well we can distinguish between them

- Draw RA, Dec, z randomly from power spectrum (done by theorists)
- Draw flux randomly from luminosity function
- Plop down fake galaxies onto real images (this takes longest!)

- Determine flux (using RA, Dec, z given by HETDEX)
- YODA (Yet another Object Detection Application)
- inherited python code (the horror!)
- uses files as communication between methods

Problem size

- $\sim 800\,000$ LAEs
- ~ 2 million OII emitters
- HETDEX will obtain spectra for all of them
- still fits in memory (ca. 500 MB)
- should scale mostly linearly

`mpi4py` for embarrassingly easy parallelization

```python
from mpi4py import MPI

def main():
    comm = MPI.COMM_WORLD
    size = comm.Get_size()
    rank = comm.Get_rank()

    if rank == 0:
        config, correction, OII = yoda.common_init()

    # broadcast these to everybody:
    config = comm.bcast(config, root=0)
    correction = comm.bcast(correction, root=0)
    OII = comm.bcast(OII, root=0)

    # do stuff
    out = numpy.empty(100000)
    ...
```

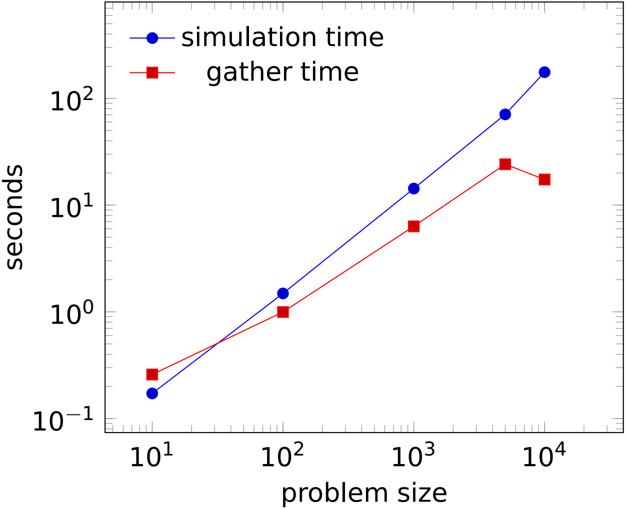
```python
# did stuff
out = numpy.empty(100000)
...

# gather the results in a list of arrays
arraylist = comm.gather(out, root=0)

# only the rank=0 thread gets the list
if arraylist is not None:
    out = numpy.concatenate(arraylist)
else:
    out = None
```

Run program with

```
mpirun yoda.py
```

Simulation time, Gather time

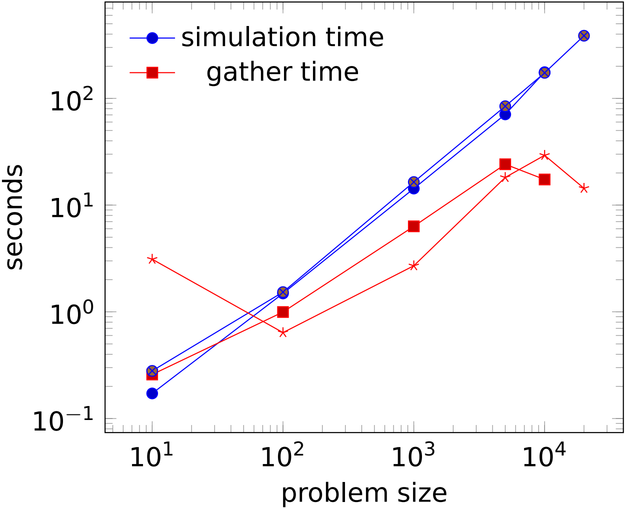
Each thread does its own thing, until we need to gather the results together.



$\rightarrow$ Astronomical scatter when the network is involved.

Simulation time, Gather time

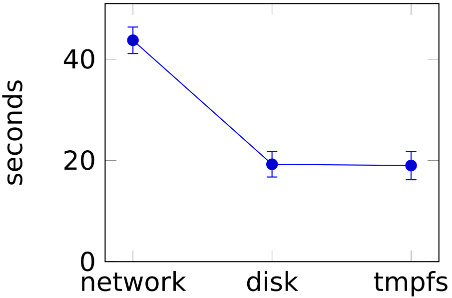
Each thread does its own thing, until we need to gather the results together.



$\rightarrow$ Astronomical scatter when the network is involved.

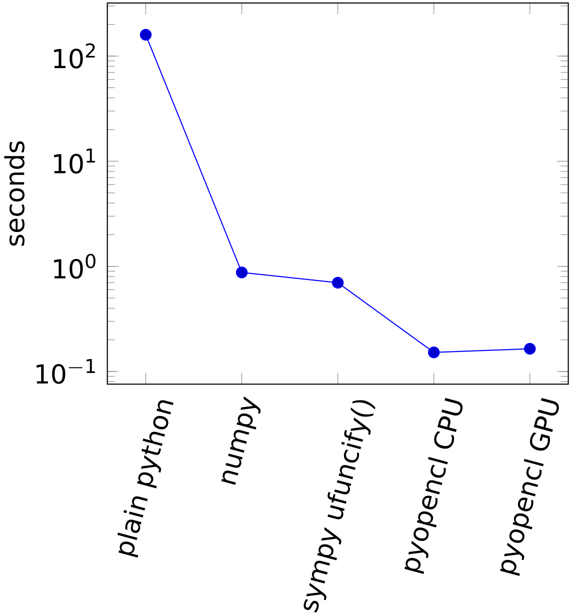
Many small files

- IO on `lionxv.rcc.psu.edu`:
  - network ($HOME, Infiniband)
  - disk (`/tmp/`)
  - tmpfs (`/dev/shm/`, no execute permissions)



- print() statements are slow, especially over the network

Plain Python, numpy, ufuncify(), and opencl



`drake.astro.`
`psu.edu`:

Intel Core
i5-3330S
2.7 GHz
Quad-core

NVidia GeForce
GT 640M