Parallel XSTAR

Mike Noble & Li Ji

Kavli Institute for Astrophysics and Space Research
Massachusetts Institute of Technology
November 7, 2007

What Is XSTAR?

A computer program for calculating the physical conditions and emission spectra of photoionized gases.

Written by Tim Kallman

http://heasarc.gsfc.nasa.gov/docs/software/xstar/xstar.html

Available as executable tool (e.g. in HEADAS)

or

In warmabs and photemis XSPEC local models

Why are we tinkering with XSTAR?

To extend Ji collisional non-equilibrium ionization code to photoionized plasmas, by taking advantage of the atomic data and physics from the equilibrium photoionization model of Xstar.

The expanded model will allow us to probe various physical processes in a wider range of astrophysical objects, such as colliding winds in X-ray binaries, outflows in AGNs and shock flows in the IGM.

Ji et al, Chandra Theory Proposal
Awarded Funding for A09

Compute Intensive

- Partly involves numerous XSTAR simulations
- Over wide range of physical parameters
- To generate table model for spectroscopic analysis
- Can be very time consuming

Example: 600 XSTAR jobs > 26 hours

Classic Approach – Serial Execution

xstar2xspec

xstinitable

↓

xstar N times

↓

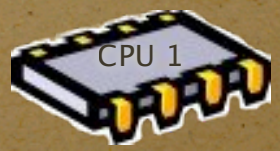
xstar2table

Perl script

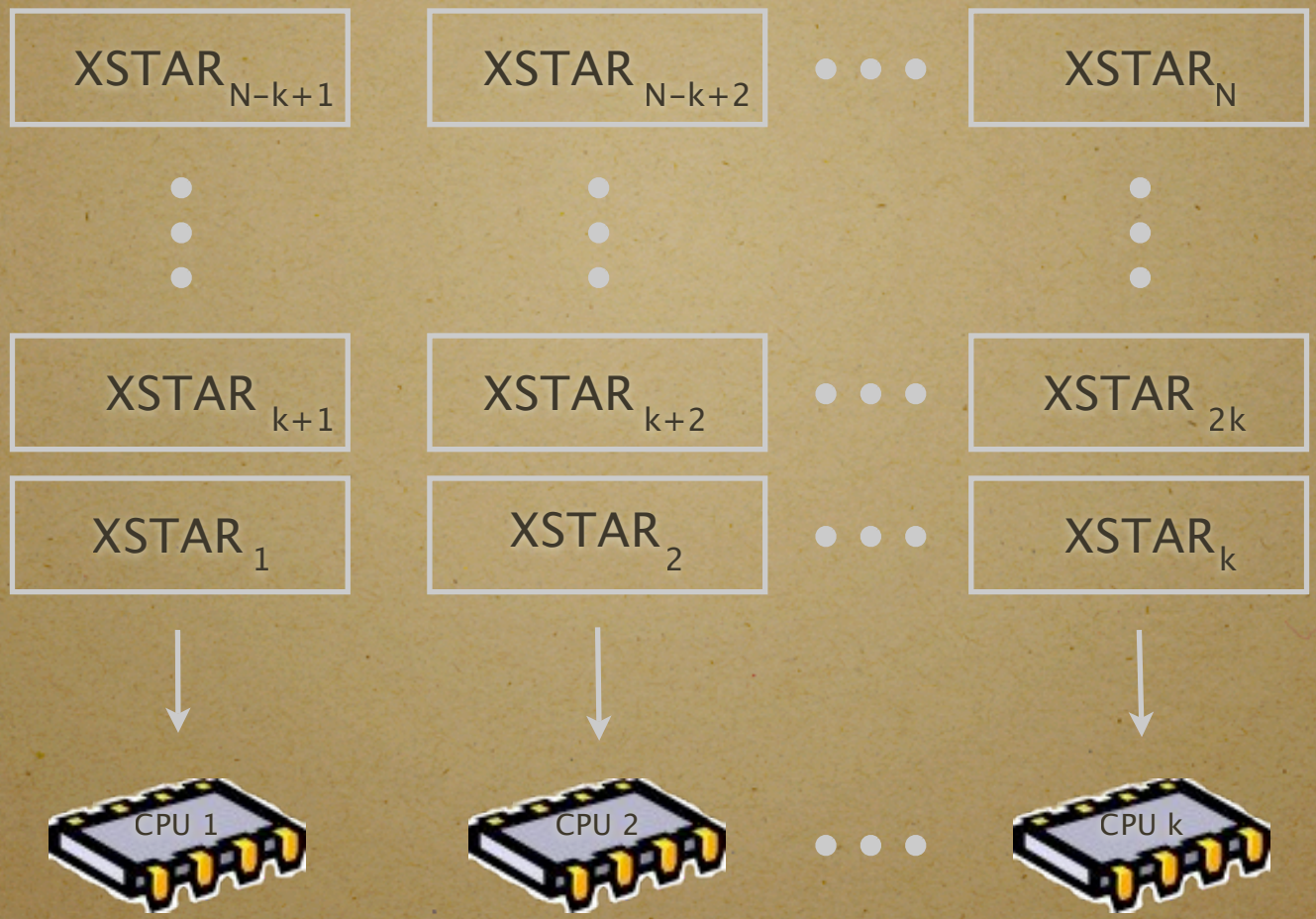
Generate list of N jobs

Job_i = XSTAR with i-th set of params

Collate spectra into table model (FITS file)



Each Job Independent → Easy Parallelism

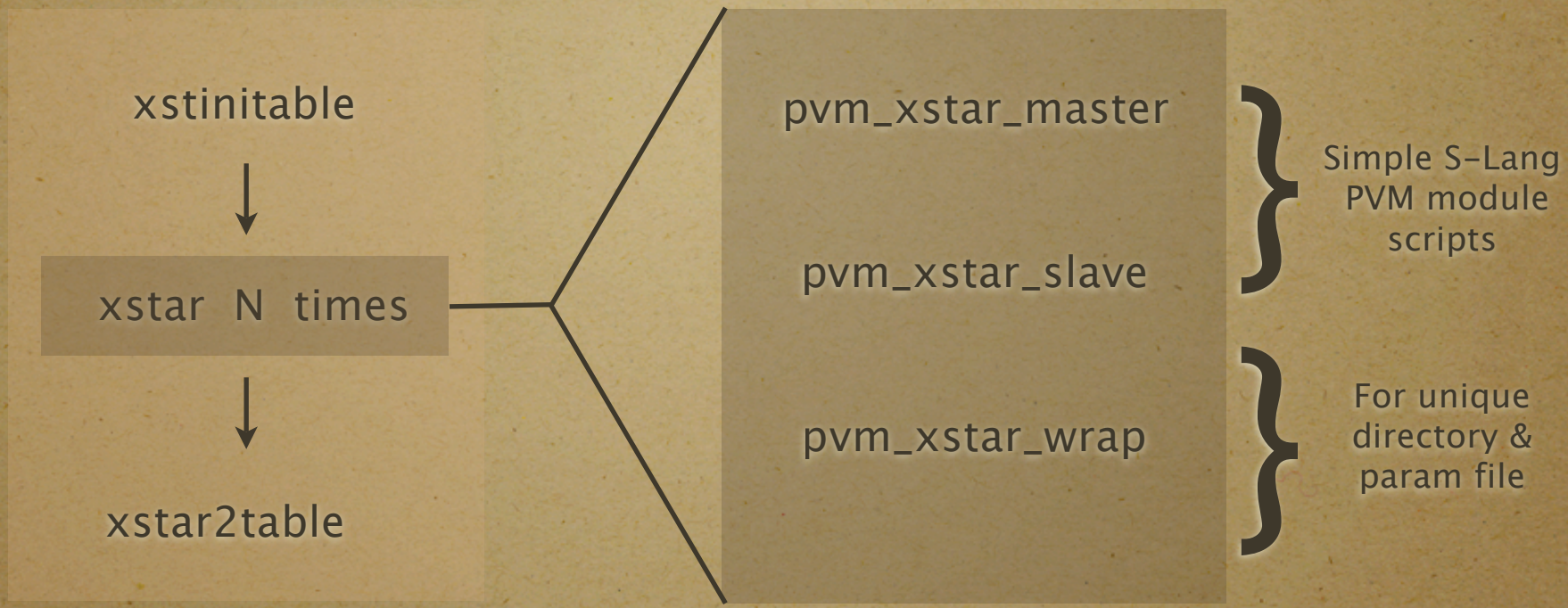


k = ~25 on NE80 network

k = 52 on HYDRA beowulf cluster

pvm_xstar

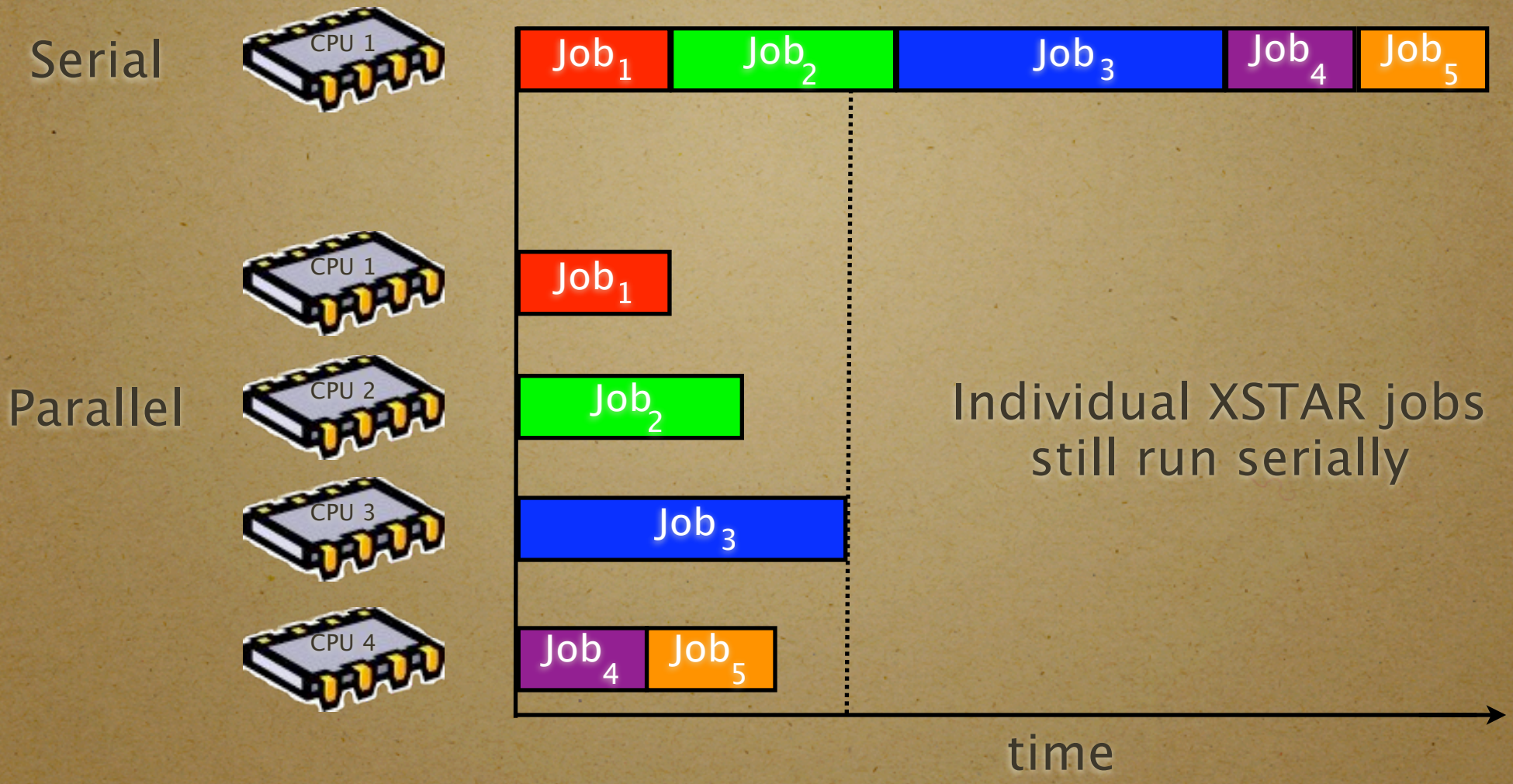
Bourne shell script – same form as xstar2xspec



NET EFFECT

system("xstar ...") → pvm_spawn("xstar ...")

Performance Expectations



Amdahl's Law Restated : can't go faster than slowest part

Experimental Results

Serial
(NE80 Linux)

600 XSTAR jobs ~ 26 hours

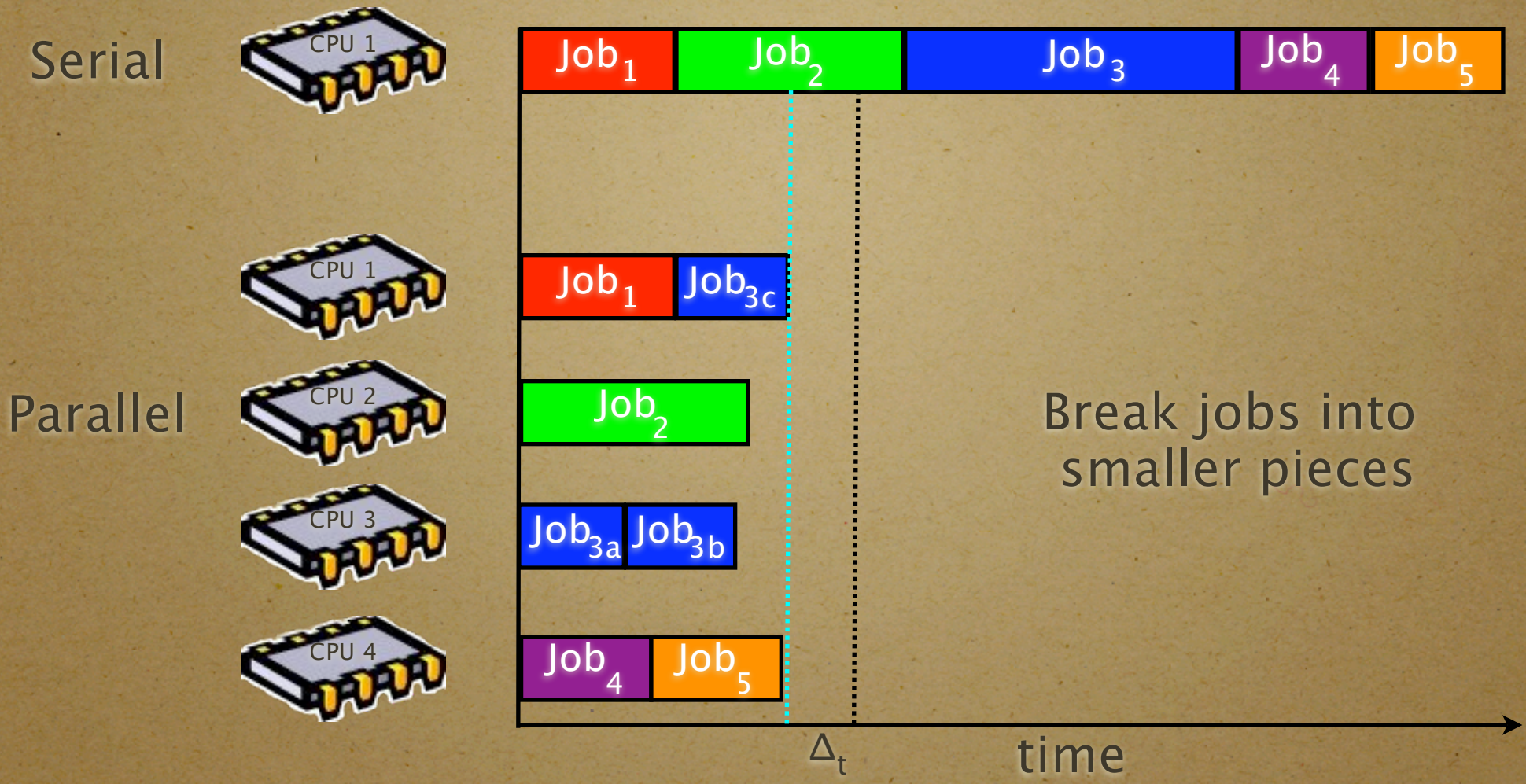
4200 XSTAR jobs ~ 7.5 days
(extrapolation)

Parallel
(HYDRA Beowulf)

4200 XSTAR jobs ~ 110 minutes
(Li's run from last night)

Compute time no longer inhibits asking of "interesting" questions, or answering them!

Increasing Parallelism



By parallelizing XSTAR Internals

Wouldn't That Be Hard?

Maybe Not

- Profile code to see hot spots
- Insert OpenMP statements around key loops

```
!$OMP PARALLEL
!$OMP DO
      do i=1,n
        x = w * (i - 0.5)
        sum = sum + f(x)
      end do
```

- OpenMP in GCC 4.2 : now widely applicable
- Ignored by non-OpenMP compilers (comments)