an analysis platform
an analysis platform

(...that works on multiple types of data)

Matthew Turk (UCSD)
The Problem:
The Problem:

In your dataset you have Things. Things you want to study. But you have to find these Things.
The Problem:

But the way you find a Thing isn’t necessarily the same way I’d find a Thing.
The Problem:

So how do we really know we’re making fair comparisons?
And more to the point, if I get a postdoc somewhere else, should I really have to re-learn how to make a slice through a simulation?
yt is an astrophysical simulation analysis package.
yt is an astrophysical simulation analysis package.

It helps you ask questions of your data.
yt is an astrophysical simulation analysis package.

It helps you ask questions of your data.

I’d like for it to work with lots of types of data.
<table>
<thead>
<tr>
<th>yt 1.0</th>
<th>yt 1.5</th>
<th>yt 1.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enzo</td>
<td>Enzo</td>
<td>Enzo</td>
</tr>
<tr>
<td></td>
<td>Orion</td>
<td>Orion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chombo</td>
</tr>
<tr>
<td>yt 1.8</td>
<td>yt 1.9</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>Enzo</td>
<td>Enzo</td>
<td></td>
</tr>
<tr>
<td>Orion</td>
<td>Orion</td>
<td></td>
</tr>
<tr>
<td>Chombo</td>
<td>Chombo</td>
<td></td>
</tr>
<tr>
<td>Flash</td>
<td>Flash</td>
<td></td>
</tr>
<tr>
<td>Ramses</td>
<td>Ramses</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Gadget</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ART</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZEUS-MP</td>
<td></td>
</tr>
</tbody>
</table>
So how does one add support for a new code?
Simulation Output

MetaData

Data

Internal yt structures
Most importantly, grids!
requires

- Understanding parameters and units
- Understanding spatial relationships of data
- Understanding disk IO for data
- Understanding data locality on disk
FLASH is a code developed by the ASC FLASH center to study supernovae. Lately it has been put to good use for cosmology.
Block structured AMR.

HDF5 output.

FLASH is an ideal code to put into yt.
Movie from Sean Couch.
(Please forgive the artifacts, yt doesn’t handle FLASH boundary conditions very well just yet.)
RAMSES is an octree, cell-based code.

Because yt was designed to operate on patches, it's optimized for few operations on large chunks of data.
process
Phase One:

Read in the data, very flatly.

Each oct grid was a yt grid.

This was prohibitively slow.
Phase Two:

Read in the data, very flatly.

Each oct grid was a proto grid.

Identify clusters.
Recursively split until efficiency requirement is met.

This worked pretty well, but for big datasets it didn’t converge.
By applying a series of cuts and subdomains, this process became reasonably speedy.
yt can now do cross-code comparisons of RAMSES, FLASH and Enzo.

(modulo a few bugs with large datasets.)
going forward:

A format for grid and gridded astrophysical data.

We will implement and support this in yt 2.0.
Identical scripts, different data, reliable cross-code comparisons.
A brief soliloquy about my research efforts!
A brief soliloquy about my research efforts!

(I do more than just yt and enzo development)
My thesis work was on adding a new chemical model to Enzo.
This can be tedious work.
This can be tedious work.

Balancing reactions, coding update steps, testing results, trying new algorithms, threading reaction rate coefficients...
This can be tedious work.

Balancing reactions, coding update steps, testing results, trying new algorithms, threading reaction rate coefficients...

But in truth, it’s pretty fun.
dengo

A new package for programmatically describing chemical rate equations and generating solvers to be used in simulation codes.

It's a meta-solver.
HI = Species("HI", weight=1)
H2I = Species("H2I", weight=2)
r13 = Reaction("r13", [(1,H2I), (2,HI)], [(3,HI)])
r22 = Reaction("r22", [(3,HI)], [(1,H2I), (1,HI)])

This is then used to generate a solver for use in a simulation code, including Jacobian calculation, EOS and cooling.

First implementation: CVODE in Enzo.
released soon...
Thank you.

yt team:

Jeff Oishi
Britton Smith
Stephen Skory
Sam Skillman
John Wise
Oliver Hahn
John ZuHone
Chris Moody

http://yt.enzotools.org/
yt-users@lists.spacepope.org
matthewturk@gmail.com