The Future of IPython: Interactive Parallel and Distributed Computing

Brian E. Granger
Tech-X Corporation

Fernando Perez
University of Colorado

Benjamin Ragan-Kelly (student)
Santa Clara University

SciPy'06
Caltech

IPython: An Enhanced Interactive Python Shell

- IPython is an enhanced interactive Python shell
- It is becoming the de facto shell for scientific computing in Python
- Included with most major Linux distributions
- Capabilities:
  - User extensible syntax
  - GUI integration (wx, Qt, GTK, etc.)
  - Seamless system shell access
  - Dynamic object/namespace introspection: docstrings, attributes, methods, source code
  - Numbered input/output prompts with command history
  - Session logging and restoring
  - Embeddable

http://ipython.scipy.org

Goals

Create an open source architecture that enables parallel and distributed programs to be developed, monitored, executed and debugged interactively and collaboratively.

- All of IPython's capabilities will be available over the wire.
- Easy things should be easy.
- It should integrate well with existing C/C++/Fortran/MPI code.
- It should support many different styles of parallelism: message passing, task farming, distributed memory.
- Fully interactive work with up to 256 processors (latency < 0.1 sec).
- Wide range of hardware: laptops to NERSC supercomputers

Realities of Distributed Scientific Computing

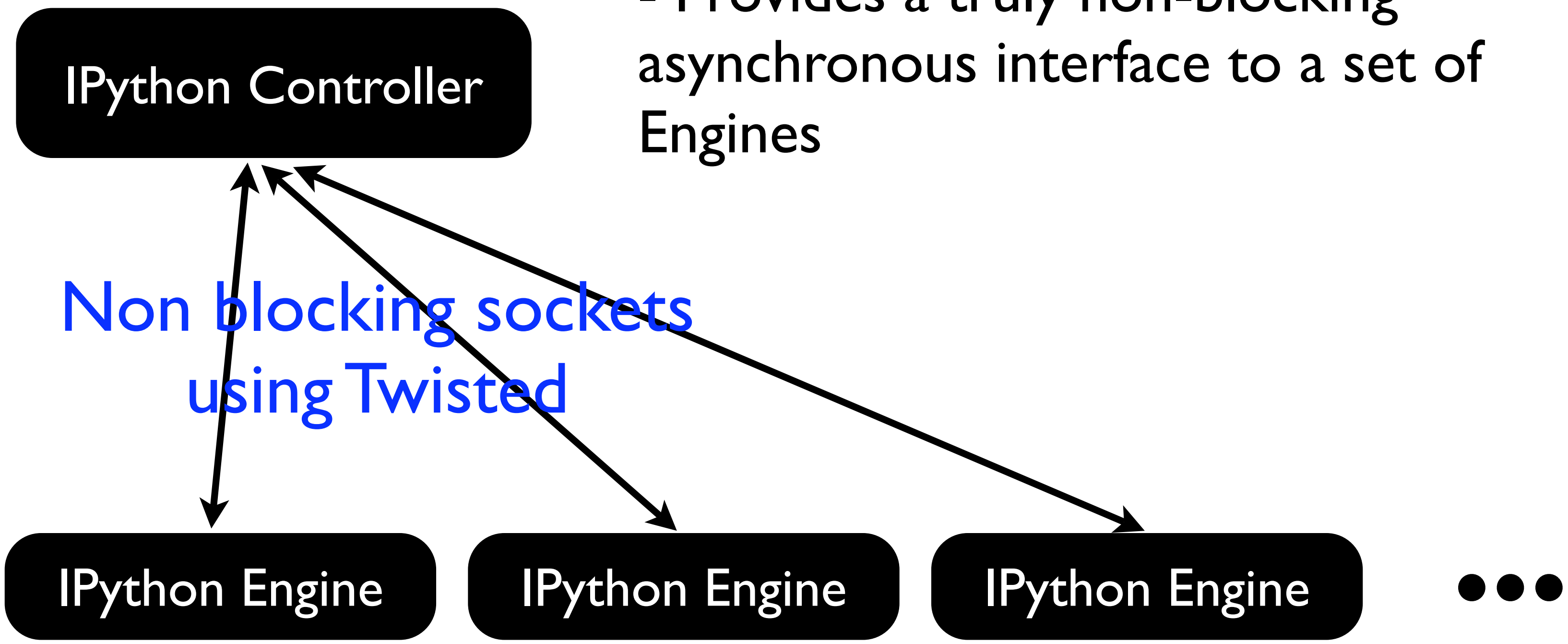
- Scientific code is often written in in compiled langauges (C/C++/Fortran).
- This code takes a long time to execute.
- While this code is running, nothing else (such as network communications) can happen in process.
- This is completely orthogonal to the asynchronous nature of distributed, network based computing.
- Because of the Global Interpreter Lock (GIL) threads don’t provide a general solution to this problem.

Minimal Requirements for Good Parallelism in Python

- Multiple processes.
- Non-blocking sockets.
- Asynchronous error/fault handling.

Architecture

- Manages the Engines
- Queue for each Engine
- Provides a truly non-blocking asynchronous interface to a set of Engines



IPython Controller

Non blocking sockets
using Twisted

IPython Engine

IPython Engine

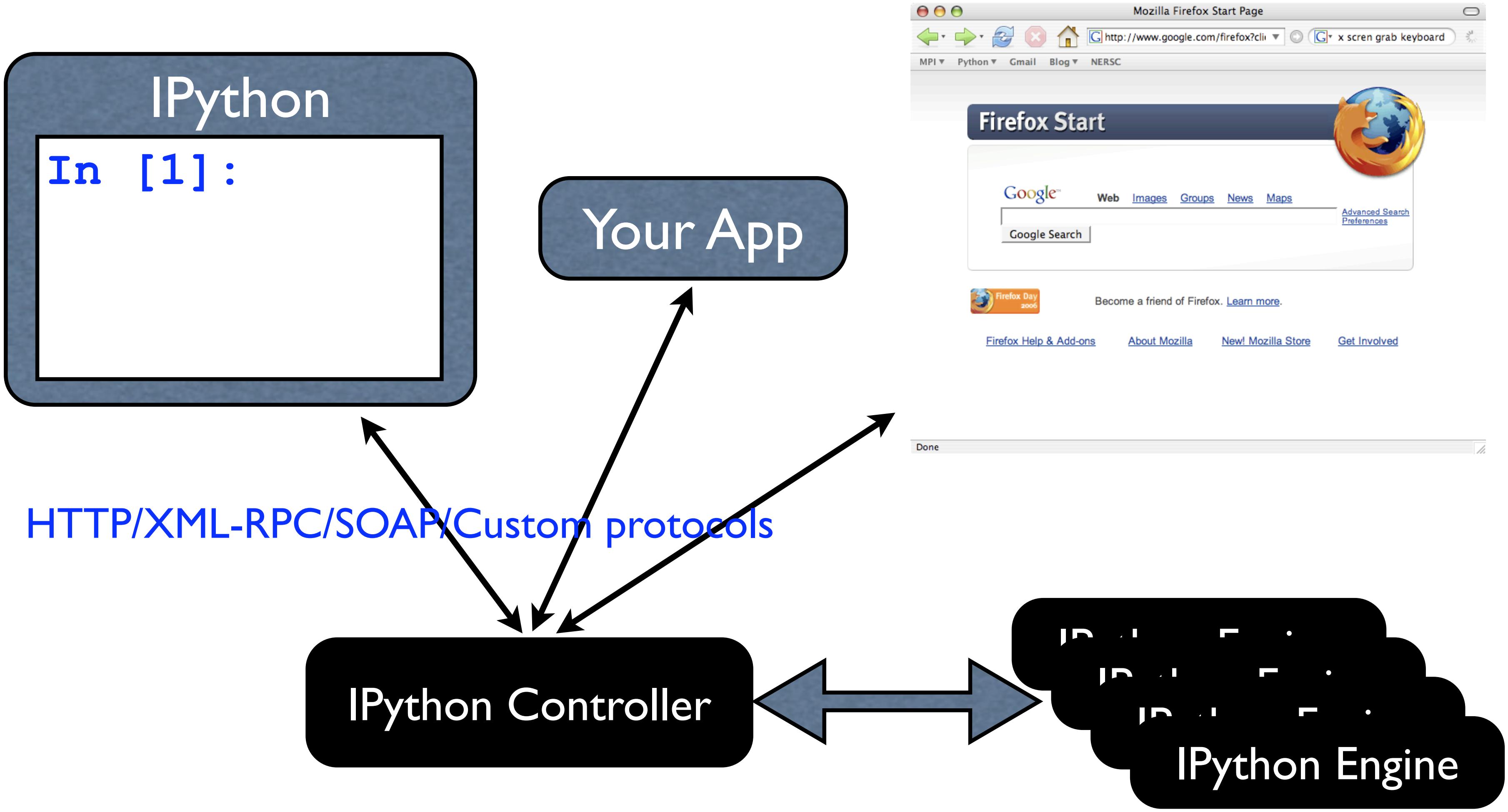
IPython Engine

•••

(I)Python virtual machines exposed to the network. These can and will block!

Architecture

Multiple/simultaneous users/apps can connect to the Controller = collaboration



IPython

```
In [1]:
```

Your App

Mozilla Firefox Start Page

http://www.google.com/firefox?cli

x scren grab keyboard

MPI ▾ Python ▾ Gmail Blog ▾ NERSC

Firefox Start

Google

Web Images Groups News Maps

Advanced Search

Preferences

Google Search

Firefox Day 2006

Become a friend of Firefox. Learn more.

Firefox Help & Add-ons About Mozilla New! Mozilla Store Get Involved

Done

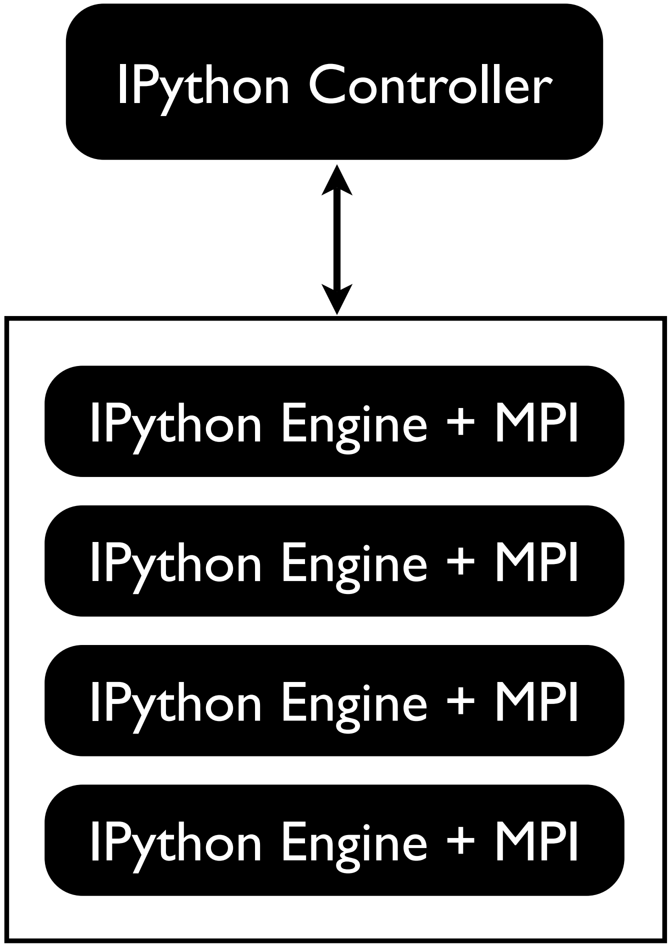
HTTP/XML-RPC/SOAP/Custom protocols

IPython Controller

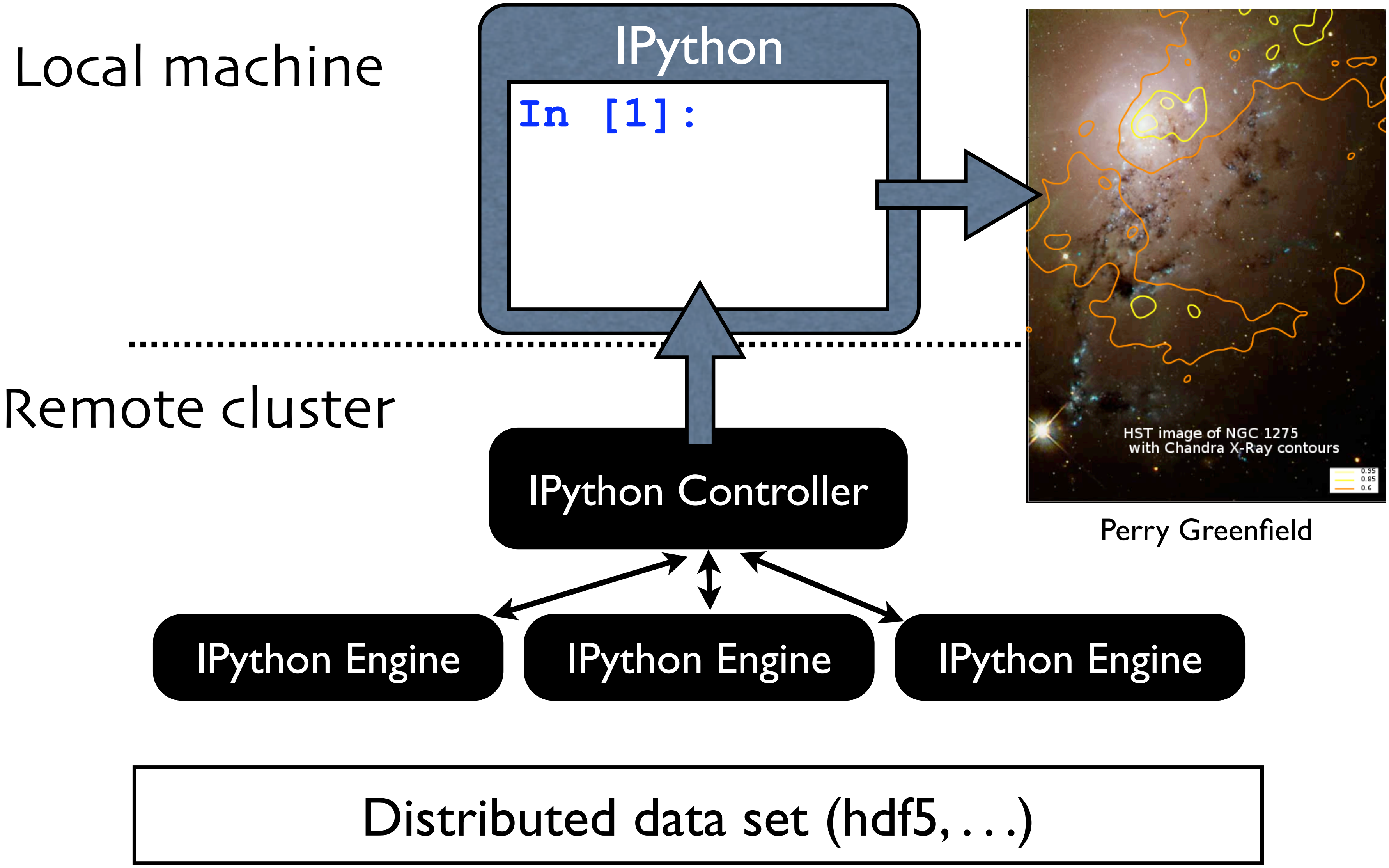
IPython Engine

Interactive Steering of Traditional Parallel Codes

- The IPython Engine can optionally initialize MPI at startup.
- User code can call wrapped C/C++/Fortran code that makes calls to MPI.
- We also support many of the Python/MPI bindings.
- The Controller and Frontend/ Clients don't use MPI.
- Full interactive control of a traditional parallel app.



Interactive and Parallel Data Analysis of Massive Distributed Data Sets



Local machine

IPython

```
In [1]:
```

Remote cluster

IPython Controller

IPython Engine

IPython Engine

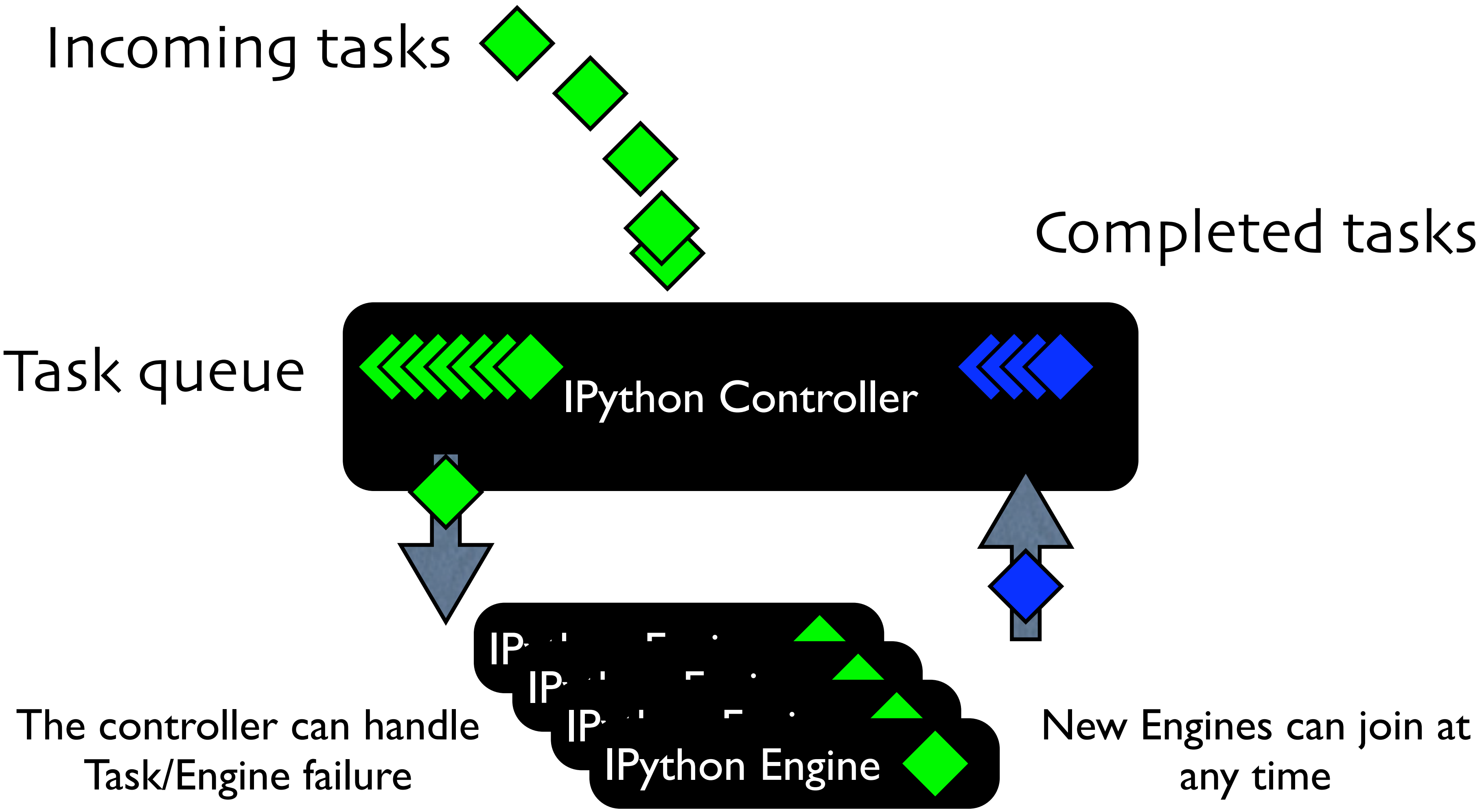
IPython Engine

Distributed data set (hdf5, . . .)

HST image of NGC 1275 with Chandra X-Ray contours

Perry Greenfield

Load Balanced, Fault Tolerant Task Farming



Incoming tasks

Completed tasks

Task queue

IPython Controller

IPython Engine

The controller can handle Task/Engine failure

New Engines can join at any time

Status

- Most of the infrastructure for interactive parallel and distributed is done.
- Working on testing/optimization/documentation.
- Beginning to implement the “IPythonic Features.”
- Watch ipython.scipy.org for updated info.
- Or check it out:

```
svn co http://ipython.scipy.org/svn/ipython/
ipython/branches/chainsaw ipython1
```

http://ipython.scipy.org/moin/Design