HTMap and High Throughput Computing Notebooks

European HTCondor Workshop 2019

Todd Tannenbaum
Center for High Throughput Computing
Goal: Bring distributed High Throughput Computing into the scientific Python environment
HTMap
# Describe work
def double(x):
    return 2 * x

# Do work
doubled = map(double, range(10))

# Use results!
print(list(doubled))# [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
# Describe work

def double(x):
    return 2 * x

# Do work

doubled = map(double,range(10))

# Use results!

print(list(doubled))
# [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
Using HTMap

```python
import htmapi

# Describe work
def double(x):
    return 2 * x

# Do work
doubled = htmapi.map(double,range(10))

# Use results!
print(list(doubled))
# [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
```
import htmapi

# Describe work
def double(x):
    return 2 * x

# Do work
doubled = htmapi.map(double, range(10))

# Use results!
print(list(doubled))
# [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
Using HTMap Tags

import htmapped

# Describe work
def double(x):
    return 2 * x

# Do work
doubled = htmapped.map(double, range(10), tag="Simulation1")

# Use results!
print(list(doubled))
# [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]
HTMap
Who is HTMap ideal for?

- Has a working computation written in **Python**
- Knows Scientific Python, and want to live in that environment
- **Wants** to use high-throughput computing... but is **very busy**
Who is HTMap NOT ideal for?

“The output of my analysis isn't a Python object”

“My function takes a millisecond to run”

“But I don’t use Python…”
Long-Running

Independent Execution

multiprocessing

concurrent.futures

htmap

condor_submit

condor_submit_dag

Interdependent Execution

Short-Running
Jupyter Notebook

- Open source browser-based application to create and share *interactive documents* that contain
  - Live code
    - Python, R, Julia, Scala, Bash, …
  - Equations
  - Visualizations
  - Narrative Text

- Also has a console window and file mover
Exploring the Lorenz System

In this Notebook we explore the Lorenz system of differential equations:

\[ \begin{align*}
\dot{x} &= \sigma(y - x) \\
\dot{y} &= \rho x - y - xz \\
\dot{z} &= -\beta z + xy
\end{align*} \]

This is one of the classic systems in non-linear differential equations. It exhibits a range of complex behaviors as the parameters (\(\sigma, \beta, \rho\)) are varied, including what are known as chaotic solutions. The system was originally developed as a simplified mathematical model for atmospheric convection in 1963.

In [7]: interact(Lorenz, N=fixed(10), angle=(0.,360.),
\sigma=(0.0,50.0),\beta=(0.,5), \rho=(0.0,50.0))

\[
\begin{align*}
\text{angle} & \quad 308.2 \\
\text{max\_time} & \quad 12 \\
\sigma & \quad 10 \\
\beta & \quad 2.6 \\
\rho & \quad 28
\end{align*}
\]
Can start a Jupyter instance on your laptop

› Install Jupyter via Anaconda or PIP, e.g.
  % pip install jupyter

› And fire it up

  % jupyter notebook

› This command will start a small embedded web server on your laptop; point your browser at http://localhost:8888 and go.
Can start a Jupyter instance on a remote server

- Point your browser at a URL where a JupyterHub server is listening
Goal: Bring distributed High Throughput Computing into the scientific Python environment

Allow users to easily *develop/test* using a small/responsive pool (eg their laptop!), and then easily *run* using all the cores in an HTCondor cluster
High Throughput Computing Notebook

Docker container with

• Python 3
• Jupyter
• Popular Python science packages
• HTCondor Python bindings
• HTMap
• Personal HTCondor pool
  • Started if no _condor_SCHEDD_HOST environment variable set
High Throughput Computing Notebook

› Run it on your laptop
  • Container will start up a personal HTCondor pool, and then Jupyter
  • HTMap uses the personal pool

› Run it on a server that has both JupyterHub and a HTCondor Schedd connected to your site's pool
  • Container will start Jupyter
  • HTMap uses the entire site pool
Run htc-notebook on your laptop with a personal pool

• Container will start up a personal HTCondor pool, and then Jupyter

• HTMap uses the personal pool
Run htc-notebook on a *remote server* with a *personal pool*.
Run htc-notebook on a *remote* server with a *personal pool*
Run on a remote server using entire site pool
Run on a remote server using entire site pool
Run on a remote server using entire site pool
Docker Hub (images) / GitHub Repos (src)
https://hub.docker.com/u/htcondor
https://github.com/htcondor/htc-notebook

Run it on your laptop

docker run -p 8888:8888 htcondor/htc-base-notebook
(then open your web browser as instructed)

Pick a software environment!

- htcondor/htc-scipy-notebook
- htcondor/htc-tensorflow-notebook
- htcondor/htc-r-notebook
- htcondor/htc-pyspark-notebook
- htcondor/htc-datascience-notebook
Thank You

Interested? Talk to us!

https://github.com/htcondor/htmap
https://github.com/htcondor/htc-notebook