

Licenses and Interpreted Languages for DHTC

Thursday morning, 10:45 am

Christina Koch (ckoch5@wisc.edu)
Research Computing Facilitator
University of Wisconsin - Madison

Expanding Our Horizons

- Our previous examples had two things in common:
 - Software was unlicensed
 - Code could be compiled or otherwise installed and then run
- This presentation discusses:
 - Licensed software
 - Running interpreted languages (Matlab, Python) on a DHTC system



Open Science Grid

LICENSING

Licensing

- Many scientific softwares are licensed.
- Licenses are restrictive, particularly for high-throughput computing

License Variations

- Per machine or 'single-install'
- Per *running* instance of the software (per “job”)
- Per username / user
- Via a license server
 - can support 1 - 1000s of concurrently running processes (“seats”)

Licensing implications for DHTC

- Per machine or 'single-install': can't be used for DHTC
- Per job: restrictive, limits the number of jobs you can have running, how do you access licenses from execute servers?
- Username: restrictive, could only run jobs on one system where your jobs run as *your username*

Approaches

- Seek out open source alternatives
 - Python or R packages that emulate specific software behavior
 - If you can't replace entire workflow, substitute free software where you can
- License-free workarounds (Matlab)
- Choose the least restrictive license possible



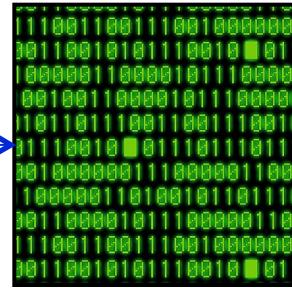
Open Science Grid

INTERPRETED LANGUAGES

Interpreted code

- Instead of being compiled and then run...

```
1 //pass
2 // - file("login.dat")
3 $i = 0; $i < count($users); $i++
4 $line = $users[$i];
5 if (eregi("username(.*)", $line))
6 {
7     // User gevonden, password is nu
8     // opgeslagen in $pass
9     $pass = $reg[$i];
10    break; // stop met de 'for'-loop
11 }
12 return $pass;
13
14 function IsLoggedIn()
15 {
16     Global $username, $password;
17     if ($username == $password)
18         $pass = md5($password);
19     else
20         $pass = FALSE;
21 }
```

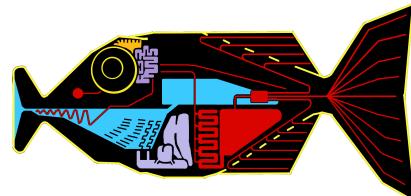


Binary code representation of the interpreted script.



- ...interpreted languages are translated into binary code “on the fly”

```
1 //pass
2 // - file("login.dat")
3 $i = 0; $i < count($users); $i++
4 $line = $users[$i];
5 if (eregi("username(.*)", $line))
6 {
7     // User gevonden, password is nu
8     // opgeslagen in $pass
9     $pass = $reg[$i];
10    break; // stop met de 'for'-loop
11 }
12 return $pass;
13
14 function IsLoggedIn()
15 {
16     Global $username, $password;
17     if ($username == $password)
18         $pass = md5($password);
19     else
20         $pass = FALSE;
21 }
```





Open Science Grid

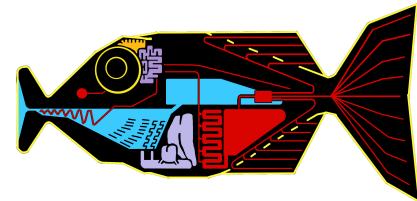
Interpretation

Script

```
$tPass = null;
$users = file("login.dat");
for ($i = 0; $i < count($users); $i++) {
    $line = $users[$i];
    if (ereg("^\$username(.*)", trim($line))) {
        // User gevonden, Password is nu
        $pass = $regs[1];
        break; // Stop met de 'for'-loop
    }
}
return $pass;

function IsLoggedIn() {
    global $username, $password;
    if ($username && $password)
        return ($password == md5(GetPassword($username)));
    return FALSE;
}
```

Interpreter

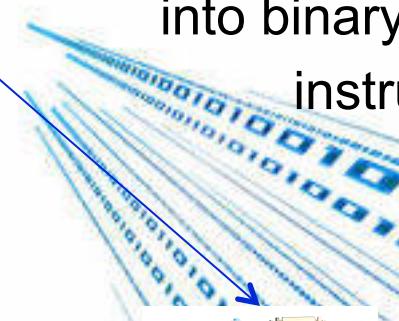


uses



Libraries

text turns
into binary
instructions





Open Science Grid

On the command line

A screenshot of a macOS terminal window titled "ckoch — bash — 53x14". The window has three tabs: "bash", "ckoch5@submit-5:~" (selected), "ckoch5@os...ster/osg-ss ...", and "ckoch5@os.../osg/python ...". The terminal content shows the creation and execution of a Python script named "hello.py".

```
[~]$ cat hello.py
import sys

name = sys.argv[1]
print "Hello", name
[~]$ python hello.py "Open Science Grid"
Hello Open Science Grid
[~]$
```

Common interpreted languages*

- Python
- R
- Julia
- Ruby
- Matlab
- Perl
- Javascript

The rest of this talk will cover running Matlab and Python jobs on a DHTC system.

*Note: the line between interpreted/compiled languages can be fuzzy. Many languages support both options, with one method being more common.

Running interpreted code in jobs

General procedure

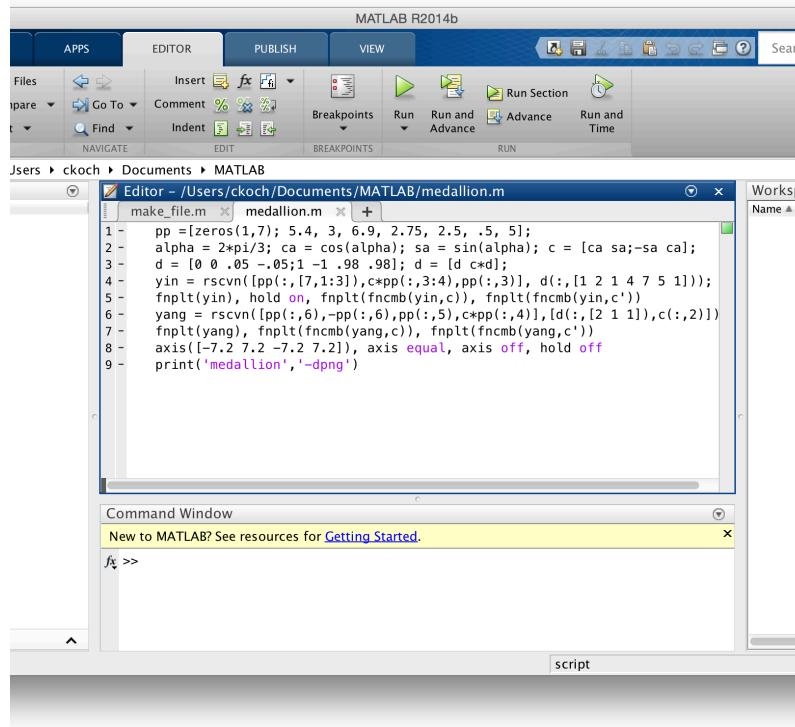
- Need to bring along interpreter and script
- Use a wrapper script as the executable
- Wrapper script will:
 - “Install” the interpreter
 - Run the script using the local installation

Matlab

- Wait a minute...isn't Matlab licensed?
- Yes, when interpreted on your computer using a normal Matlab installation.
- However, Matlab code can also be compiled.
- Once compiled, the code can be run without a license using a (free) set of files called the Matlab runtime.

Matlab contrast

Running Matlab on your computer
Uses license per instance



Running Matlab on DHTC
Uses license once, runs
many instances for free



Matlab script(s)
compiled w/ Matlab
compiler (uses license)

Compiled binary
interpreted by



Matlab Runtime (free)



Matlab on DHTC

1. Compile Matlab code using the Matlab compiler (mcc)
 - requires a license
2. Prepare a copy of the Matlab runtime
 - download for free from Mathworks
3. Write a script that “installs” the runtime
 - The Matlab compiler actually writes most of this script for you
4. Use the runtime install to run the compiled Matlab code

Interpreted Languages

- Matlab is still being compiled before running
- What about programs like Python or R that aren't usually compiled?

Python

- Common language used in research computing
- Can incorporate external modules for extra functionality
- Usually interpreted, not easy to compile
- Free!

Python on DHTC

1. Bring along:
 - pre-built installation OR Python source code
 - your Python code
2. Use a wrapper script to:
 - unpack pre-built install OR install from source
 - run your Python script

(Similar to Exercise 1.3 this morning, will also work for R)

Exercises

- Running Matlab Jobs
 - Exercise 1.4
- Running Python Jobs
 - Exercise 1.5: Pre-building Python and using that installation
 - Exercise 1.6: Writing a script that installs Python with every job
- Half of the room should start with Matlab, the other with Python

Questions?

- Feel free to contact me:
 - ckoch5@wisc.edu
- Now: Hands-on Exercises
 - 11:00am-12:15pm
- Next:
 - 12:15-1:15pm: Lunch
 - 1:15 onward: free time