ScopingExperiment

September 11, 2018

1 Scoping Experiment

This experiment is designed to show a pecularity in Julia'd scoping. Julia uses a layered scoping where the scope of the inner function has access to the values of the outer function. For example:

```
In [1]: x=5; y=7; #Defined globally
        function scopeTest(z)
          x += z #Changes global value
          y = Vector{Float64}(1) #Declares a variable, local scope
          y[1] = 2
          return x + y + z
        end
```

```
Out[1]: scopeTest (generic function with 1 method)
```

However, what is happening here, and why?

```
In [6]: using Distributed
        addprocs(1)
        function f1()
          @distributed for i = 1:100
            x = 10
            if x < 100
              x = x + 1
            end
          end
          x = x + 100 + 10
          return x
        end
        f1()
```

```
        UndefVarError: x not defined



        Stacktrace:
```

```
[1] macro expansion at /buildworker/worker/package_linux64/build/usr/share/julia/stdli

[2] f1() at ./In[6]:4

[3] top-level scope at In[6]:13
```

```
In [7]: function f2()
          @distributed for i = 1:100
            x = 10
            if x < 100
              x = x + 1
            end
          end
          return x
        end
        f2()
```

```
Out[7]: 5
```