Python Multiprocessing

Python Multiprocessing

- Why Python
- Why Multiprocessing

Python Multiprocessing

- Simple Monte Carlo – embarrassingly parallel

```
import random, math

def bsmcl (S, T, sig, r, div, numsteps, numpaths, payoff):
    dT = T/numsteps
    nudt = (r - div - 0.5 * sig**2)*dT
    sigsdt = sig * math.sqrt (dT)
    lnS = math.log (S)

    sumValues = 0

    for j in range (numpaths):
        # calculate one path
        lnST = lnS
        for i in range(numsteps):
            epsilon = random.gauss (0, 1)
            lnST = lnST + nudt + sigsdt * epsilon

        sumValues += payoff (math.exp (lnST))

    return sumValues / numpaths * math.exp (-r * T)

def EuroCallPayoff (K):
    return lambda S: max (S - K, 0)
```

Python Multiprocessing

```
# needs to be visible via an import of the module http://stackoverflow.com/questions/3288595/multiprocessin
def calcPath (par):
    lnST = par.lnS
    for i in range(par.numsteps):
        epsilon = random.gauss (0, 1)
        lnST = lnST + par.nudt + par.sigsdt * epsilon

    return par.payoff (math.exp (lnST))


def bsmc3p (S, T, sig, r, div, numsteps, numpaths, payoff, pool):
    dT = T/numsteps
    nudt = (r - div - 0.5 * sig**2)*dT
    sigsdt = sig * math.sqrt (dT)
    lnS = math.log (S)

    sumValues = 0

    par = AnyAttr (lnS = lnS, numsteps = numsteps, nudt = nudt, sigsdt = sigsdt, payoff = payoff)
    if pool == None:
        sumValues = sum (map (calcPath, [par] * numpaths))
    else:
        sumValues = sum (pool.map (calcPath, [par] * numpaths))

    return sumValues / numpaths * math.exp (-r * T)
```

Python Multiprocessing

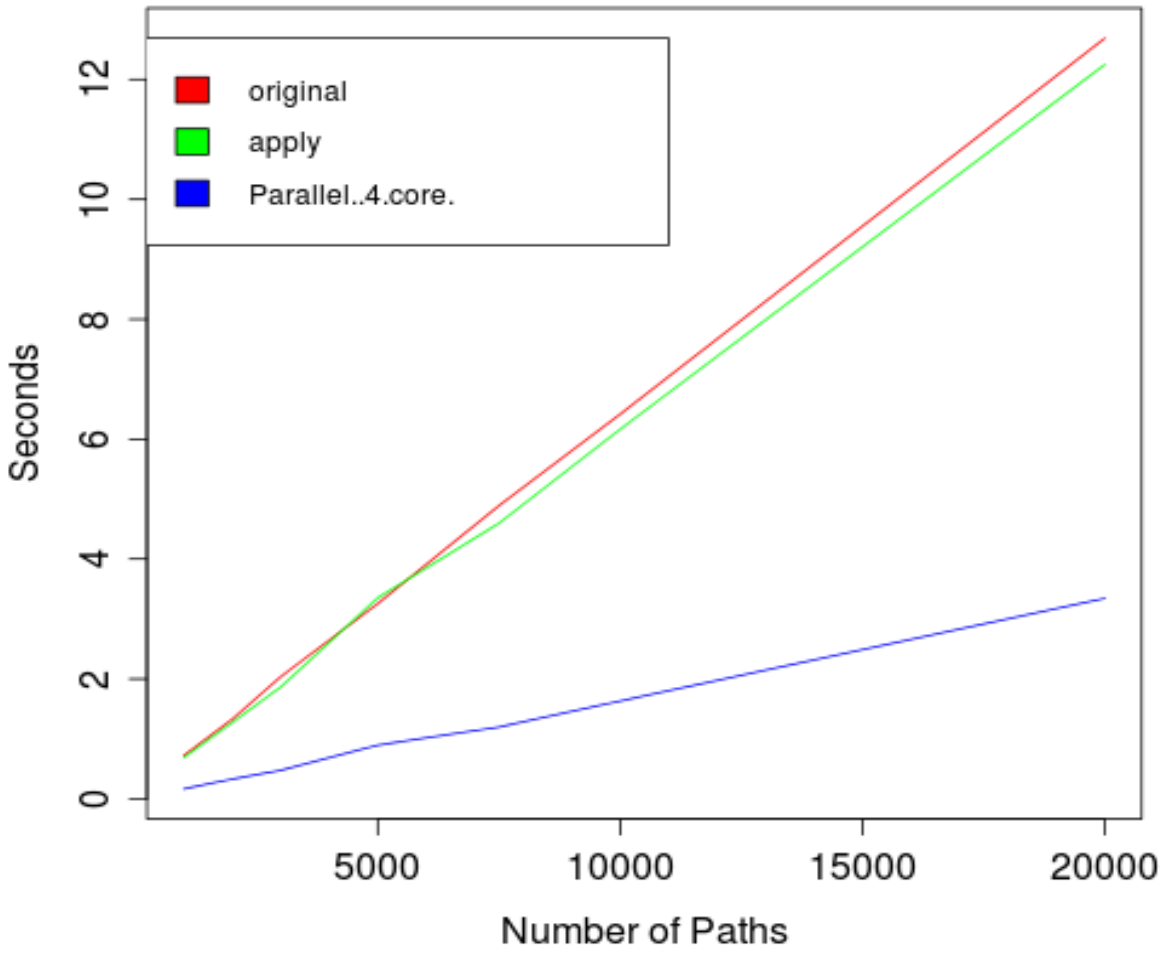
Snags...

```
# this doesn't work with multiprocessing because the lambda isn't visible
def EuroCallPayoff (K):
    return lambda S: max (S - K, 0)

# a class should work
class CallPayoff:
    def __init__ (self, K):
        self.K = K

    def __call__ (self, S):
        return max (S - self.K, 0)
```

Python Multiprocessing



Python Multiprocessing

- Other ways to speed things up
  - Psyco
  - Pypi?
  - Scipy - vectorisation