strong scalability

January 17, 2018

```python
In [2]: %matplotlib inline

In [18]: import matplotlib.pyplot as plt
         import numpy as np
         from pylab import *
         import matplotlib as mpl
         mpl.rcParams['figure.dpi']= 200


         data10M = []
         with open('srisultati10M.txt') as datafile10M:
                 for line in datafile10M:
                     data10M.append(float(line))

         data1G = []
         with open('srisultati1G.txt') as datafile1G:
                 for line in datafile1G:
                     data1G.append(float(line))

         data100M = []
         with open('srisultati100M.txt') as datafile100M:
                 for line in datafile100M:
                     data100M.append(float(line))


         x = np.array([1,2,4,8,16,20])
         y = x
         data10M

Out[18]: [1.0,
          1.04640633843,
          1.05778032037,
          1.00762942779,
          0.933838383838,
          0.913086419753]

In [37]: #errors
```

```python
err10M = []
with open('serr10M.txt') as datafile10M:
        for line in datafile10M:
                err10M.append(float(line))

err1G = []
with open('serrors1G.txt') as datafile1G:
        for line in datafile1G:
                err1G.append(float(line))

err100M = []
with open('serr100M.txt') as datafile100M:
        for line in datafile100M:
                err100M.append(float(line))

err100M
```

Out[37]: [0.250031022723,
 0.117924426642,
 0.0410291749347,
 0.0103242493746,
 0.0125045005443,
 0.019262056204]

```python
In [31]: plt.legend(bbox_to_anchor=(0.5, 1.02, 1., .102), loc=3,
                   ncol=2, mode="expand", borderaxespad=0.)
plt.ylabel('Speedup')
plt.xlabel('Number of processors')
#plt.plot(t,y)
plt.plot(x,y,'r', label='bisector line')
plt.plot(x,data1G, 'g',label='10^7')
plt.errorbar(x,data1G, yerr = err1G)
plt.plot(x,data10M, 'y', label='10^8')
plt.errorbar(x,data10M, yerr = err10M)
plt.plot(x,data100M, 'b', label='10^9')
plt.errorbar(x,data100M, yerr = err100M)
#plt.plot(x,data1G, 'go', 'g')
#plt.plot(x,data10M, 'yo', 'y')
#plt.plot(x,data100M, 'bo', 'b')
plt.xticks(x,[1,2,4,8,16,20])
plt.title('STRONG SCALABILITY')
plt.legend()
plt.show()
```

STRONG SCALABILITY