scaleup_parforMathwork.m

```
%{
In this example, you start on your local multicore desktop and measure the
time required to run a calculation, as a function of increasing numbers of
workers. The test is called a strong scaling test. It enables you to measure
the decrease in time required for the calculation if you add more workers.
This dependence is known as speedup, and allows you to estimate the parallel
scalability of your code. You can then decide whether it is useful to increase
the number of workers in your parallel pool, and scale up to cluster and cloud
computing.

%}
```

Triton

```
parpool(1);
```

```
Starting parallel pool (parpool) using the 'local' profile ...
connected to 1 workers.
```

```
a = MyRand(100);
```

```
Elapsed time is 1.228964 seconds.
```

```
%{
Elapsed time is 1.696408 seconds.
>> a = MyRand(100);
Elapsed time is 1.234502 seconds.
>> a = MyRand(100);
Elapsed time is 1.189794 seconds.
>>
%}
```

```
delete(gcp)
```

```
parpool(4)
```

```
Starting parallel pool (parpool) using the 'local' profile ...
connected to 4 workers.

ans =

 Pool with properties:

            Connected: true
           NumWorkers: 4
              Cluster: local
       AttachedFiles: {}
```

```
    AutoAddClientPath: true
          IdleTimeout: 30 minutes (30 minutes remaining)
          SpmdEnabled: true
```

```
a = MyRand(100);
```

```
Elapsed time is 0.597729 seconds.
```

```
%{
Elapsed time is 0.737882 seconds.
>> a = MyRand(100);
Elapsed time is 0.411070 seconds.
>> a = MyRand(100);
Elapsed time is 0.360522 seconds.
%}
```

```
delete(gcp)
```

```
parpool(8)
```

```
Starting parallel pool (parpool) using the 'local' profile ...
connected to 8 workers.

ans =

 Pool with properties:

            Connected: true
           NumWorkers: 8
              Cluster: local
        AttachedFiles: {}
    AutoAddClientPath: true
          IdleTimeout: 30 minutes (30 minutes remaining)
          SpmdEnabled: true
```

```
a = MyRand(100);
```

```
Elapsed time is 0.503713 seconds.
```

```
a = MyRand(100);
```

```
Elapsed time is 0.196310 seconds.
```

```
a = MyRand(100);
```

```
Elapsed time is 0.204232 seconds.
```

```
%{
Elapsed time is 0.499832 seconds.
Elapsed time is 0.199511 seconds.
Elapsed time is 0.199622 seconds.
```

```
%}
```

```
delete(gcp)
```

```
parpool(16)
```

```
Starting parallel pool (parpool) using the 'local' profile ...
connected to 16 workers.

ans =

 Pool with properties:

            Connected: true
           NumWorkers: 16
              Cluster: local
        AttachedFiles: {}
    AutoAddClientPath: true
          IdleTimeout: 30 minutes (30 minutes remaining)
          SpmdEnabled: true
```

```
a = MyRand(100);
```

```
Elapsed time is 0.526804 seconds.
```

```
a = MyRand(100);
```

```
Elapsed time is 0.171885 seconds.
```

```
a = MyRand(100);
```

```
Elapsed time is 0.162747 seconds.
```

```
%{
            Connected: true
            NumWorkers: 16
Elapsed time is 0.539691 seconds.
Elapsed time is 0.164308 seconds.
Elapsed time is 0.161784 seconds.
%}
```

```
delete(gcp)
```

```
parpool(20)
```

```
Starting parallel pool (parpool) using the 'local' profile ...
connected to 20 workers.

ans =
```

```
Pool with properties:

            Connected: true
           NumWorkers: 20
              Cluster: local
        AttachedFiles: {}
    AutoAddClientPath: true
          IdleTimeout: 30 minutes (30 minutes remaining)
          SpmdEnabled: true
```

```
a = MyRand(100);
```

```
Elapsed time is 0.587510 seconds.
```

```
a = MyRand(100);
```

```
Elapsed time is 0.179430 seconds.
```

```
a = MyRand(100);
```

```
Elapsed time is 0.180713 seconds.
```

```
%{
            Connected: true
            NumWorkers: 20
Elapsed time is 0.597396 seconds.
Elapsed time is 0.176818 seconds.
Elapsed time is 0.170725 seconds.
%}
% Increasing 16 -> 20 doesn't do any good.
```

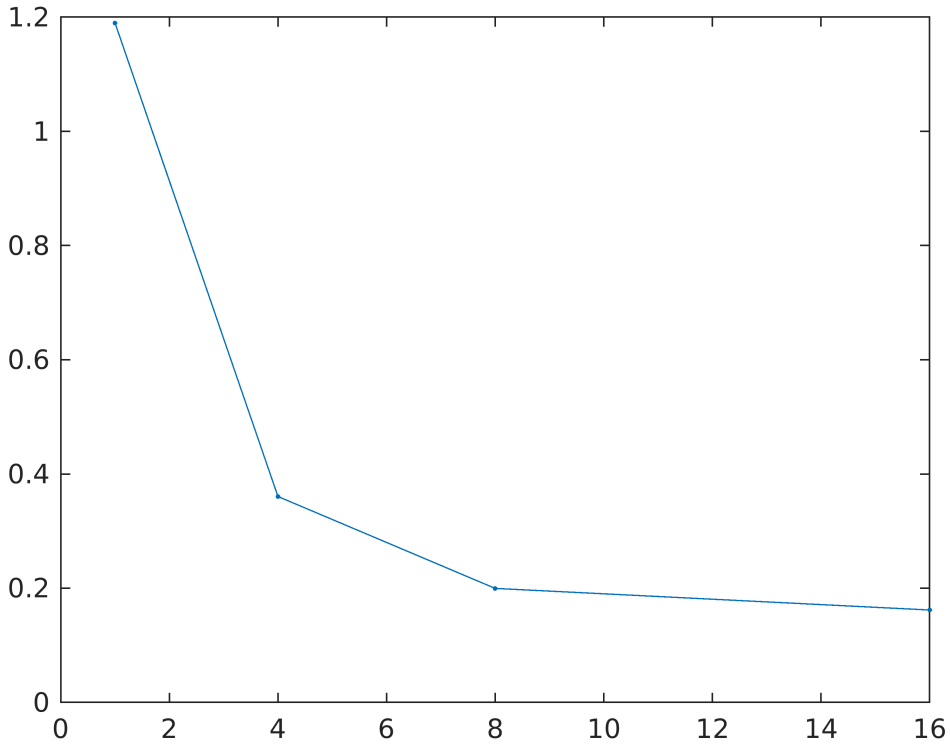
```
T=[1.189794 0.360522 0.199622 0.161784]
```

```
T = 1×4
    1.1898    0.3605    0.1996    0.1618
```

```
n=[1 4 8 16]
```

```
n = 1×4
     1     4     8    16
```

```
plot(n,T,'.-')
```




```
delete(gcp)
```

How about ordinary for-loop?

```
MyRandfor(100);
```

```
Elapsed time is 0.929337 seconds.
```

```
%{
Elapsed time is 1.065781 seconds.
Elapsed time is 1.040182 seconds.
%}
```

Dell:

myRand(1000)

```
%delete(gcp);
%{
ans =

 Pool with properties:
```

```
            Connected: true
           NumWorkers: 1
              Cluster: local
        AttachedFiles: {}
    AutoAddClientPath: true
          IdleTimeout: 30 minutes (30 minutes remaining)
          SpmdEnabled: true

>> a = MyRand(1000);
Elapsed time is 144.843540 seconds.
%}
```