3. Distributed Arrays and Map-Reduce

February 13, 2014

```
         addprocs(8)
In [1]:  8-element Array{Any,1}:
Out [1]:  2
          3
          4
          5
          6
          7
          8
          9
```

```
         nprocs()
In [2]:  9
Out [2]:
```

```
         dat=map(x->integer(x*10),rand(10000,10000))
In [3]:  10000x10000 Array{Int64,2}:
Out [3]:  7   9   3  6   0  3  3   3   0  9   6  ...   0   7   7   9  10  2  1
         3   3
          8  10   6  1   5  2  5   7   4  9   3        6   7   2  10   5  6  2
         2   9
          6   5   8  8   3  6  1   9   5  6   4        4   7   9   6   1  1  4
         9   5
          2   3   5  8  10  1  7   7   9  7   5        4   7   1   1  10  9  4
         6   7
          0   0   1  3   2  8  9  10   6  7   1        5   2   3   3   1  8  6
         1   8
          7   7   6  2   5  9  9   9   7  3   0  ...   3  10   3   2  10  7  4
         3   7
          4   3   4  5   7  8  8   9   8  0   5       10   6   9   4   6  7  9
         2   7
          1   2  10  0   4  6  1   2   3  1   7        1   0   1   3   7  5  2
         4   8
          9   4   4  0   8  3  8   7   9  1   5        7   6   8   5   1  7  5
         2   3
          8   9   8  8   0  8  8   9   4  0   3        1   1   2   8   9  3  0
         9   8
          5   0  10  9   9  6  8   8  10  7   6  ...   7  10   6   7   8  4  1
         4   4
          8   6  10  0   6  4  5   7   9  6   0        9   0   4   4   4  3  9
         5   9
          0   6   9  3   2  9  6   7   1  9   3       10   2   1   7   5  2  5
         6   4
          ⋮                  ⋮                  ⋮  ⋱                         ⋮
```

```
 6   8   6  8   2  6  9   7   3  3   5      5   2   2   1   6  2  8
6   2
 6   7   5  1   6  7  3   4   2  6   7      8   3   7   4   1  2  8
3   1
 7   3   3  5   7  9  7   8   0  1   7  ...   3   7   7   2   0  4  3
5   9
 7   2   3  0   5  6  2   3   1  9   6      0   4   8   9   4  9  6
5  10
 5   8   2  7   1  1  2   5   5  2   1      6   4   3   8   1  1  6
9   5
 8   1   5  3   3  3  0   6   6  9   8      1   5   7   2   6  6  8
7   6
 0   4   0  1   2  7  7   4   9  7   2      5   2   8   1   9  8  7
10   9
 6   3   2  2   8  4  8   8   8  4   8  ...   7   4  10   5   6  5  8
2   8
 1   3   0  1   1  9  0   6  10  4   8      8   6   5   2   0  4  1
9   2
 5   1   4  8   9  4  1   1   3  7  10      5  10   7   3   7  6  3
2   1
 6   7   8  7   6  4  4   7   5  6   4      0   4   9   3   6  8  5
7   9
 8   2   8  6  10  9  6  10   4  6   8      2   4   1   4   3  6  2
1   4
```

In [4]:

```
ddat=distribute(dat)
```

Out [4]:

```
10000x10000 DArray{Int64,2,Array{Int64,2}}:
 7   9   3  6   0  3  3   3   0  9   6  ...   0   7   7   9  10  2  1
3   3
 8  10   6  1   5  2  5   7   4  9   3      6   7   2  10   5  6  2
2   9
 6   5   8  8   3  6  1   9   5  6   4      4   7   9   6   1  1  4
9   5
 2   3   5  8  10  1  7   7   9  7   5      4   7   1   1  10  9  4
6   7
 0   0   1  3   2  8  9  10   6  7   1      5   2   3   3   1  8  6
1   8
 7   7   6  2   5  9  9   9   7  3   0  ...   3  10   3   2  10  7  4
3   7
 4   3   4  5   7  8  8   9   8  0   5     10   6   9   4   6  7  9
2   7
 1   2  10  0   4  6  1   2   3  1   7      1   0   1   3   7  5  2
4   8
 9   4   4  0   8  3  8   7   9  1   5      7   6   8   5   1  7  5
2   3
 8   9   8  8   0  8  8   9   4  0   3      1   1   2   8   9  3  0
9   8
 5   0  10  9   9  6  8   8  10  7   6  ...   7  10   6   7   8  4  1
4   4
 8   6  10  0   6  4  5   7   9  6   0      9   0   4   4   4  3  9
5   9
 0   6   9  3   2  9  6   7   1  9   3     10   2   1   7   5  2  5
6   4
 ⋮                 ⋮                 ⋮  ⋱                      ⋮
```

```
  6   8   6  8   2  6  9   7   3  3   5       5   2   2   1   6  2  8
 6   2
  6   7   5  1   6  7  3   4   2  6   7       8   3   7   4   1  2  8
 3   1
  7   3   3  5   7  9  7   8   0  1   7  ...   3   7   7   2   0  4  3
 5   9
  7   2   3  0   5  6  2   3   1  9   6       0   4   8   9   4  9  6
 5  10
  5   8   2  7   1  1  2   5   5  2   1       6   4   3   8   1  1  6
 9   5
  8   1   5  3   3  3  0   6   6  9   8       1   5   7   2   6  6  8
 7   6
  0   4   0  1   2  7  7   4   9  7   2       5   2   8   1   9  8  7
 10   9
  6   3   2  2   8  4  8   8   8  4   8  ...   7   4  10   5   6  5  8
 2   8
  1   3   0  1   1  9  0   6  10  4   8       8   6   5   2   0  4  1
 9   2
  5   1   4  8   9  4  1   1   3  7  10       5  10   7   3   7  6  3
 2   1
  6   7   8  7   6  4  4   7   5  6   4       0   4   9   3   6  8  5
 7   9
  8   2   8  6  10  9  6  10   4  6   8       2   4   1   4   3  6  2
 1   4
```

```
map(x->myid(),dat)
```

In [5]:

```
10000x10000 Array{Int64,2}:
```

Out [5]:

```
 1  1  1  1  1  1  1  1  1  1  1  1  1  ...  1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1     1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1     1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1     1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1     1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  ...  1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1     1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1     1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1     1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1     1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  ...  1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1     1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1     1  1  1  1  1  1  1  1  1
1  1  1
 ⋮              ⋮              ⋮        ⋱        ⋮              ⋮
 1  1  1  1  1  1  1  1  1  1  1  1  1      1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1      1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  ...  1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1      1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1      1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1      1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1      1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1  ...  1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1      1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1      1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1      1  1  1  1  1  1  1  1  1
1  1  1
 1  1  1  1  1  1  1  1  1  1  1  1  1      1  1  1  1  1  1  1  1  1
1  1  1
```

In [6]:

```
map(x->myid(),ddat)
```

Out [6]:

```
10000x10000 DArray{Int64,2,Array{Int64,2}}:
 2  2  2  2  2  2  2  2  2  2  2  2  2  ...  8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2      8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2      8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2      8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2      8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2  ...  8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2      8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2      8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2      8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2      8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2  ...  8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2      8  8  8  8  8  8  8  8  8
8  8  8
 2  2  2  2  2  2  2  2  2  2  2  2  2      8  8  8  8  8  8  8  8  8
8  8  8
 ⋮              ⋮              ⋮         ⋱         ⋮              ⋮
 3  3  3  3  3  3  3  3  3  3  3  3  3     9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3     9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3  …  9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3     9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3     9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3     9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3     9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3  …  9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3     9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3     9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3     9  9  9  9  9  9  9  9  9
9  9  9
 3  3  3  3  3  3  3  3  3  3  3  3  3     9  9  9  9  9  9  9  9  9
9  9  9
```

```
@time map(sin,dat);
```

In [6]:

```
elapsed time: 18.495849369 seconds (4000237720 bytes allocated)
```

```
@time map(sin,ddat);
```

In [6]:

```
elapsed time: 3.801187871 seconds (1143612 bytes allocated)
```

In [7]:

```
function intsum(x)
  if x==0
    0
  else
    x+intsum(x-1)
  end
end
```

```
intsum (generic function with 1 method)
```

Out [7]:

```
@time reduce(+,map(intsum,dat))
```

In [8]:

```
elapsed time: 13.26386314 seconds (3573678796 bytes allocated)
1925337529
```

Out [8]:

In [9]:

```
@everywhere function intsum(x)
  if x==0
    0
  else
    x+intsum(x-1)
  end
end
```

```
@time reduce(+,map(intsum,ddat))
```

In [10]:

```
elapsed time: 6.24797383 seconds (1743872 bytes allocated)
1925337529
```

Out [10]:

In [11]:

```
@everywhere function rsumtail(x,n)
  if x==0
    n
  else
```

```
    rsumtail(x-1,n+x)
  end
end

@everywhere function rsumtail(x)
  rsumtail(x,0)
end

@everywhere function rsum(x)
  if x==0
    0
  else
    x+rsum(x-1)
  end
end
```

In [12]:

```
@time rsumtail(100000)
```

```
elapsed time: 0.00278938 seconds (42804 bytes allocated)
5000050000
```

Out [12]:

In [13]:

```
@time rsum(100000)
```

```
elapsed time: 0.002768959 seconds (23496 bytes allocated)
5000050000
```

Out [13]:

In [14]:

```
@time (s=0;
for i in dat
  s+=intsum(i)
end;
s)
```

```
elapsed time: 26.858735368 seconds (6400048908 bytes allocated)
1925337529
```

Out [14]:

In [21]:

```
@time @parallel (+) for i in dat
  intsum(i)
end
```

```
elapsed time: 30.597089961 seconds (8590437940 bytes allocated)
1925337529
```

Out [21]:

In [16]:

```
@time reduce(+,map(intsum,ddat))
```

```
elapsed time: 7.447626368 seconds (1453340 bytes allocated)
1925337529
```

Out [16]:

In [22]:

```
ddat.indexes
```

Out [22]:

```
2x4 Array{(Range1{Int64},Range1{Int64}),2}:
 (1:5000,1:2500)      (1:5000,2501:5000)      ...  (1:5000,7501:10000)
 (5001:10000,1:2500)  (5001:10000,2501:5000)
(5001:10000,7501:10000)
```

In [23]:

```
ddat.chunks
```

Out [23]:

```
2x4 Array{RemoteRef,2}:
 RemoteRef(2,1,5)  RemoteRef(4,1,7)  RemoteRef(6,1,9)
RemoteRef(8,1,11)
 RemoteRef(3,1,6)  RemoteRef(5,1,8)  RemoteRef(7,1,10)
RemoteRef(9,1,12)
```

In [24]:

```
ddat.pmap
```

Out [24]:

```
8-element Array{Int64,1}:
 2
 3
 4
 5
 6
 7
 8
 9
```

```
fetch(@spawnat 2 sum(x->x*x,localpart(ddat)))
```

In [27]:

```
418800280
```

Out [27]:

```
map(fetch,{@spawnat p sum(map(x->x*x,localpart(ddat))) for p in ddat.pmap})
```

In [28]:

```
8-element Array{Any,1}:
 418800280
 418938675
 418841631
 418547283
 419008882
 418728812
 418788879
 418975117
```

Out [28]:

```
@time reduce(+,map(fetch,{@spawnat p sum(map(x->x*x,localpart(ddat))) for p in procs(d
```

In [29]:

```
elapsed time: 2.15132193 seconds (663104 bytes allocated)
3350629559
```

Out [29]:

```
@time reduce(+,map(x->x*x,ddat))
```

In [39]:

```
elapsed time: 3.018259173 seconds (1472940 bytes allocated)
3350629559
```

Out [39]:

```
@time sum(map(x->x*x,dat))
```

In [42]:

```
elapsed time: 8.165484936 seconds (2399993056 bytes allocated)
3350629559
```

Out [42]: