**Partition wisely**

https://www.talend.com/blog/2018/03/05/intro-apache-spark-partitioning-need-know/

Overhead (webpage)

Avoid skewed data, shuffle block.

(to manage memory; to ensure complete resource utilization on executor’s nodes)

Text

Description automatically generated

**“4 cores/8 thread processors”**

https://forums.tomshardware.com/threads/cores-vs-threads-explained.3460905/

Wordcount-improved (At home):

No partition: 5.7m

20 partitions: 1.7m

50 par: 1.9m

70 par: 2m. -> -ls output 70 partitions

90 par: 2m

100 par:

300 par: 5m

Euler - cluster:

20 par: 1.3m

50 par: 20s

100 par: 17s

Euler – local (10000000):

1 par: 15.16s / 13.85s

2 par: 14.7s

3 par: 13.39s

6 par: 14.23s

8 par: 14.53s

20 par: 15.7s

40 par: 17.82s

80 par: 17.07s

100 par: 19.04s

200 par: 23.94s

250 par: 24.51s

Euler – lab (100000000 samples):

2 par: 36.454s

3 par: 25.452s

5 par: 17.065s

6 par: 14.997s

7 par: 13.723s

8 par: 13.151s

20 par: 14.304s

200 par: 14.096s

600 par: 16.058s

1200 par: 20.38s

2400 par: 25.881s

3000 par: 31.163s

10

Sample Xn until sum > 1, then stop, get V(n);

Repeat many times;

Take average -> value for e.