Does Hadoop need one-disk-per-core?

Can shared disks help reduce hadoop cluster cost?

Min Xu, Saisanthosh Balakrishnan, Gary Lauterbach, Sean Lie – SeaMicro
Kshitij Sudan – SeaMicro & University of Utah

SeaMICRO

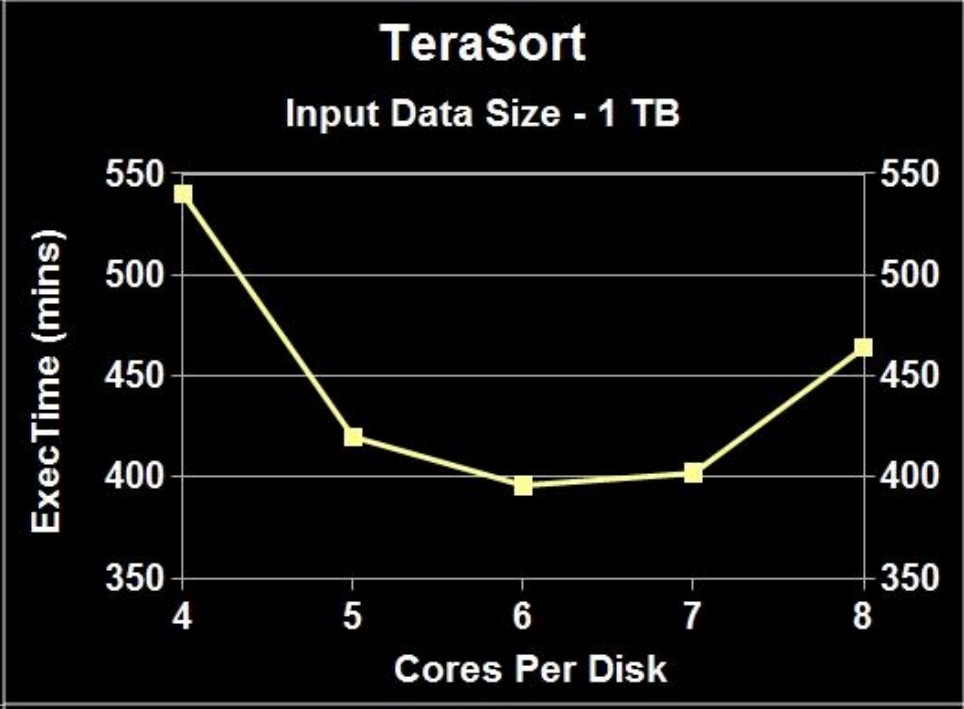
SeaMICRO

Traditional Hadoop Clusters

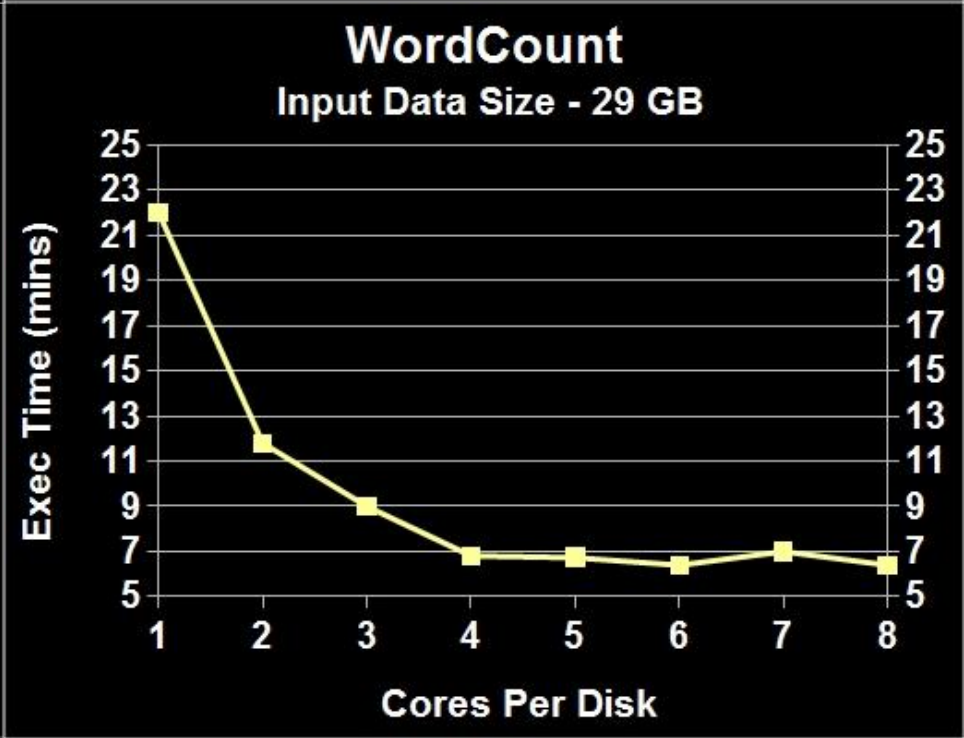
- Hadoop workloads => large data
- Large data => high disk bandwidth
- Server 1: 12 disks per 8 cores => great
- Server 2: 4 disks per 8 cores => good
- Server 3: 64 disk per 512 cores => not good?

Does hadoop really need high BW of “one disk per core”?!


Customer - 1
Input Data Size - 160 GB
Exec Time (mins)
120
100
80
60
40
20
1
2
3
4
5
6
7
8
Cores Per Disk


TeraSort
Input Data Size - 1 TB
Exec Time (mins)
550
500
450
400
350
4
5
6
7
8
Cores Per Disk


Customer - 2
Input Data Size - 100 GB
Exec Time (mins)
120
110
100
90
80
70
60
50
40
1
2
3
4
5
6
7
8
Cores Per Disk


WordCount
Input Data Size - 29 GB
Exec Time (mins)
25
23
21
19
17
15
13
11
9
7
5
1
2
3
4
5
6
7
8
Cores Per Disk

Why is 6 the magic number?

- Shouldn't the number be more workload dependent?
- Profiling individual nodes found
  - Server CPUs are busy with ctx switches during shuffle and sort
  - High CPU overhead in these ctx switches skewed the results
  - Limiting NIC interrupt rate significantly reduce the CPU overhead
- Lesson: due to all-to-all data transfers, the hadoop shuffle & sort stage has storm of ctx switches

Work-in-progress

- We continue to do experiments with varying core-per-disk ratios
  - Important to reduce the CPU overhead
  - Fix the total # of compute nodes and vary the total # of disks
- Does hadoop need one-disk-per-core?
  - Optimized results do not support it
  - Results will be workload dependent – Terasort is an extreme case where no map/reduce is done

Thank you!