Improving Scalability of Apache Spark-based Scale-up Server through Docker Container-based Partitioning

Joohyun Kyong and Sung-Soo Lim

School of Computer Science
Kookmin University

February 25, 2017

Outline

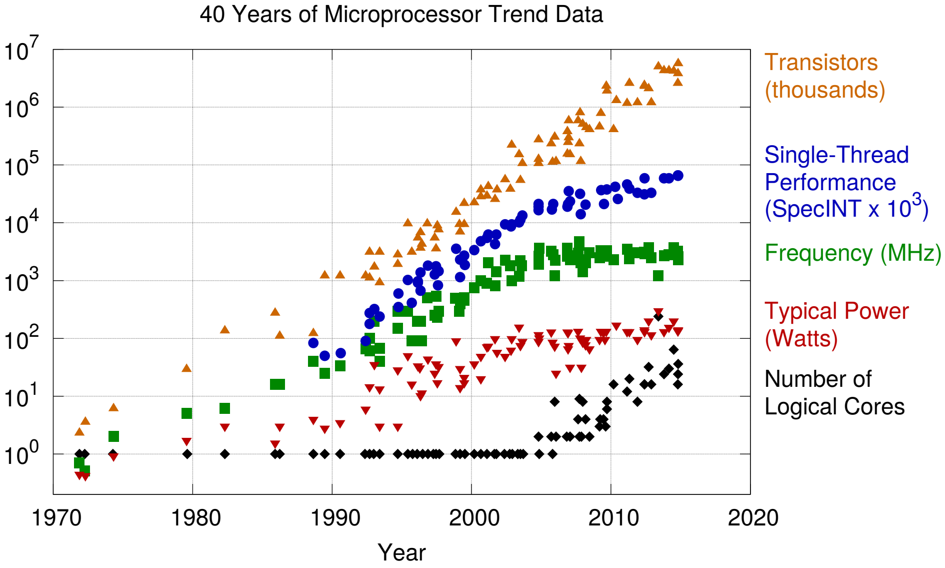
- Background of research and History of the Scale-up server scalability
- Scale-up Server Scalability Problems
- Our method and Evaluation
- Our architecture

Big Data Market Driving Factors

More than 65 billion devices were connected to the Internet by 2010, and this number will go up to 230 billion by 2020.

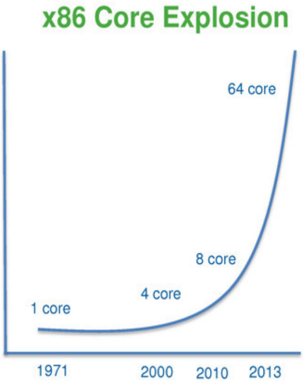
∗ "The Internet of Things Is Coming" [John Mahoney et al., 2013]

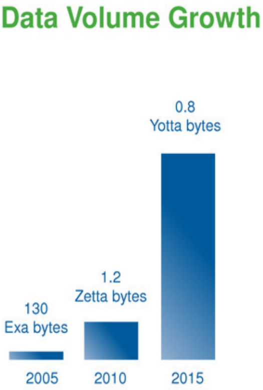
And 40 Years of Microprocessor Trend Data



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2015 by K. Rupp

CPU Trend and Data Trend

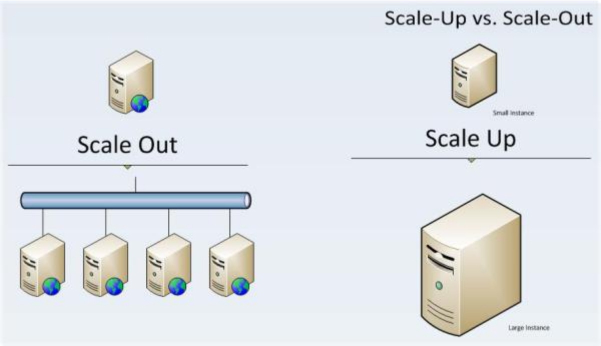

x86 Core Explosion


Data Volume Growth

Source:SGI

Scale Out Vs. Scale Up

- Scale out or scale horizontally: adding more nodes to a system.
- Scale up or scale vertically: adding resources to a single node in a system.



History of the Scale-out Scalability Research

- Big Data frameworks focus has been on Scale-out.
  - For example, Hadoop, Spark
- M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica. Resilient Distributed Datasets: A Fault-tolerant Abstraction for In-memory Cluster Computing. In Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, NSDI'12, 2012.
- X. Ren, G. Ananthanarayanan, A. Wierman, and M. Yu. Hopper: Decentralized Speculation-aware Cluster Scheduling at Scale. In Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, SIGCOMM '15, pages 379-392, 2015.
- K. Ousterhout, R. Rasti, S. Ratnasamy, S. Shenker, and B.-G.Chun. Making Sense of Performance in Data Analytics Frameworks. In Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation, NSDI'15, 2015.
- M. Maas, K. Asanović, T. Harris, and J. Kubiatowicz. Taurus: A Holistic Language Runtime System for Coordinating Distributed Managed-Language Applications. In Proceedings of the Twenty-First International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS '16, pages 457-471, 2016.

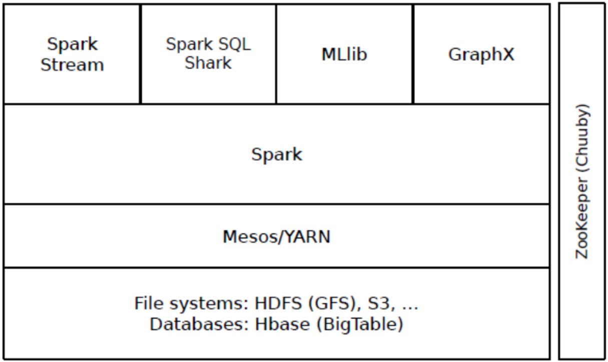
But, what about Scale-up server?

Why scale-up when you can scale-out?

- If data fits in memory of multicore then often order of magnitude better performance
  - GraphLab1 (multicore) is 1000x faster than Hadoop (cluster)
  - Multicores now have 1-12 TB memory: most graph analytics problems fit!
- And scale-up servers are mostly used in scientific analytics areas.

BigData framework – Apache Spark

Spark Big Data Analytics Stack



Spark Stream

Spark SQL Shark

MLlib

GraphX

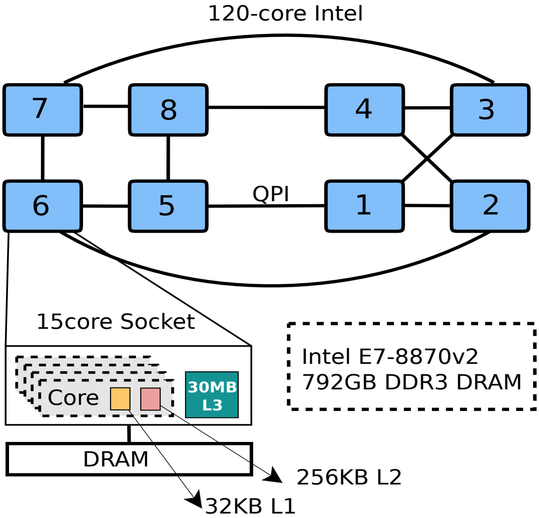
Spark

Mesos/YARN

File systems: HDFS (GFS), S3, ...
Databases: Hbase (BigTable)

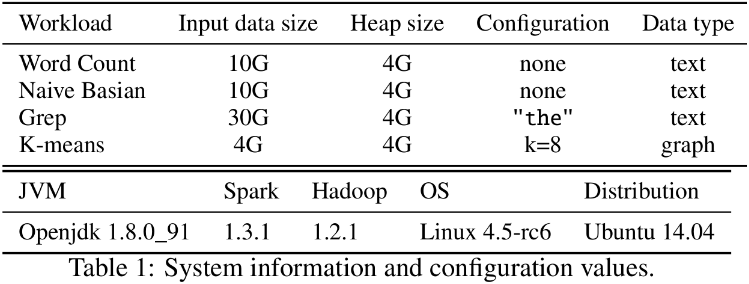
ZooKeeper (Chuuby)

Test-bed : our scale-up server



Intel E7-8870v2
792GB DDR3 DRAM

Benchmark – BigDataBench

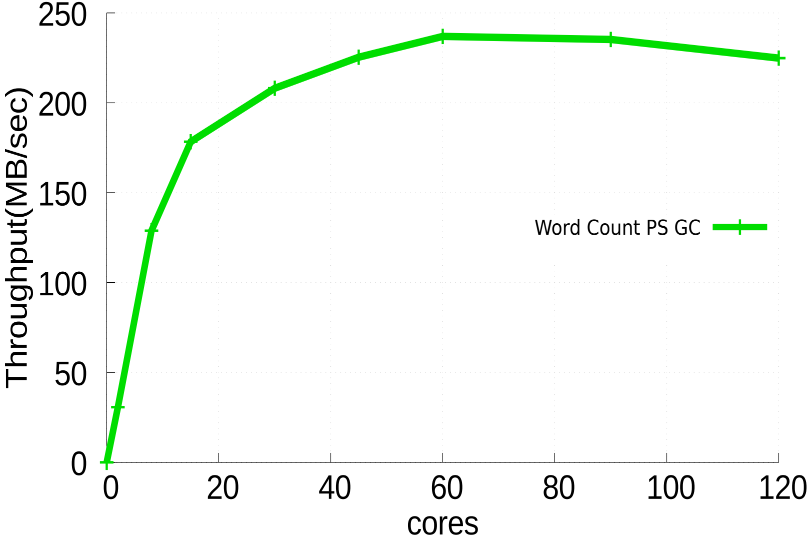


Workload	Input data size	Heap size	Configuration	Data type
Word Count	10G	4G	none	text
Naive Basian	10G	4G	none	text
Grep	30G	4G	"the"	text
K-means	4G	4G	k=8	graph

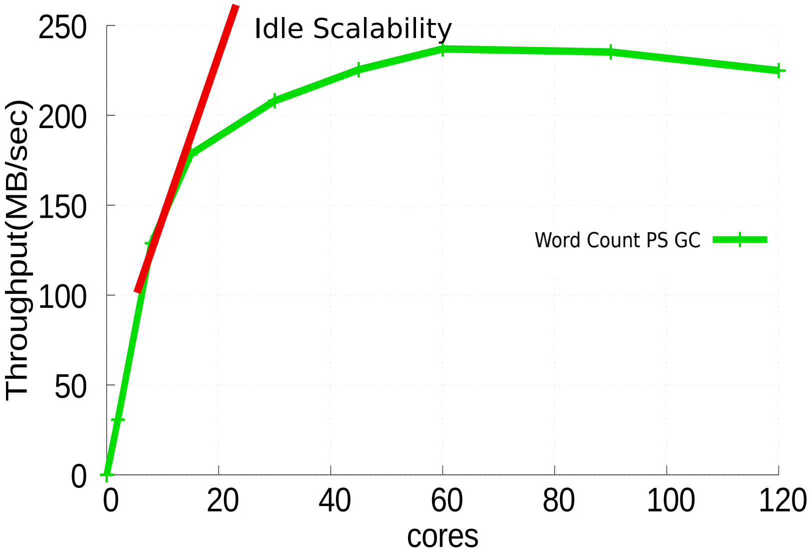
JVM	Spark	Hadoop	OS	Distribution
Openjdk 1.8.0_91	1.3.1	1.2.1	Linux 4.5-rc6	Ubuntu 14.04

Table 1: System information and configuration values.

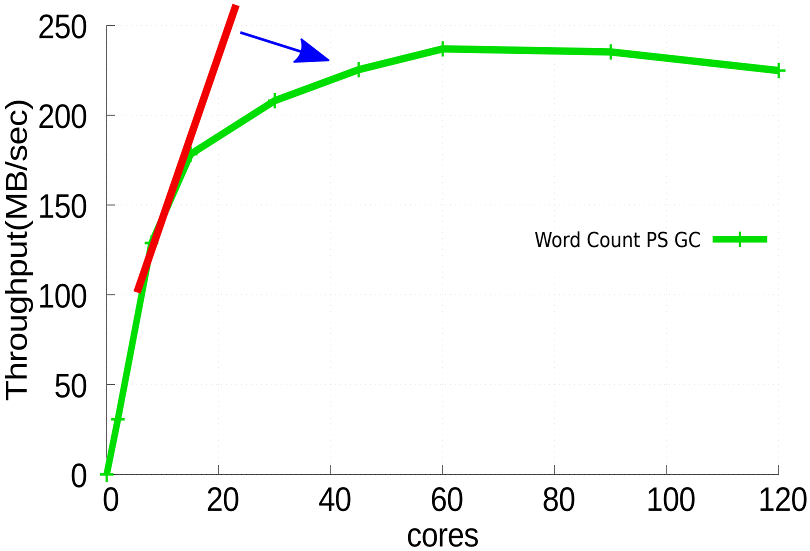
Spark Scale-up Server Scalability Problems



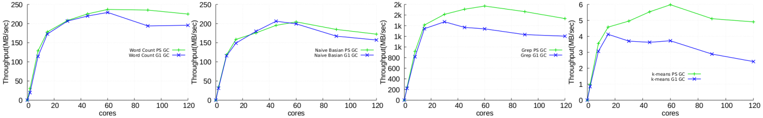
Spark Scale-up Server Scalability Problems



Spark Scale-up Server Scalability Problems

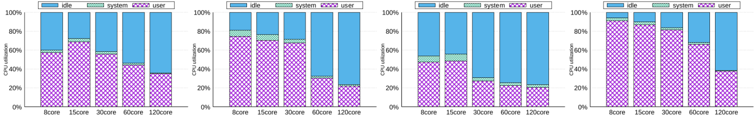


Spark Scale-up Server Scalability Problems



(a) Word Count (b) Naive Basian (c) Grep (d) K-means

Figure 1: Performance scalability.



(a) Word Count (b) Naive Basian (c) Grep (d) K-means

Figure 2: CPU utilization.

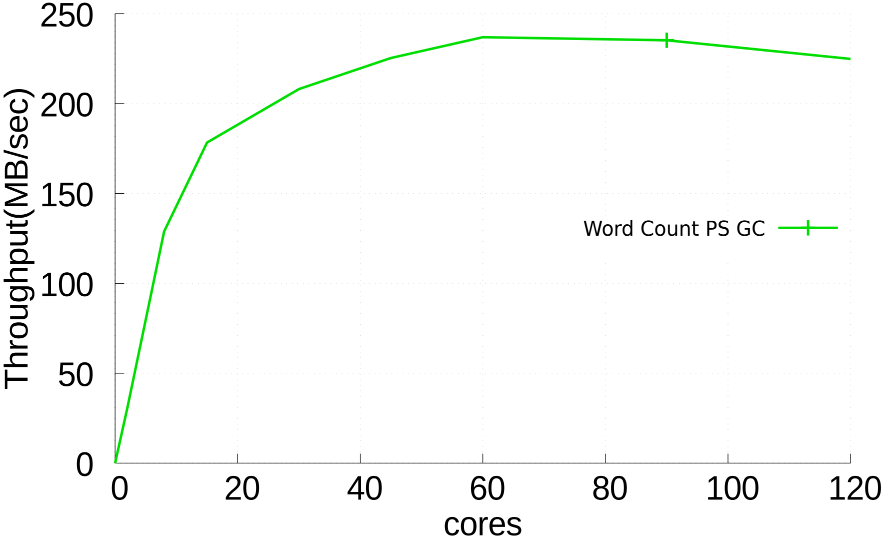
The General Scalability Problems – Previous Research

- Garbage Collection(GC) overheads.
- Locality of memory accesses on Non-Uniform Memory Access(NUMA) architecture.

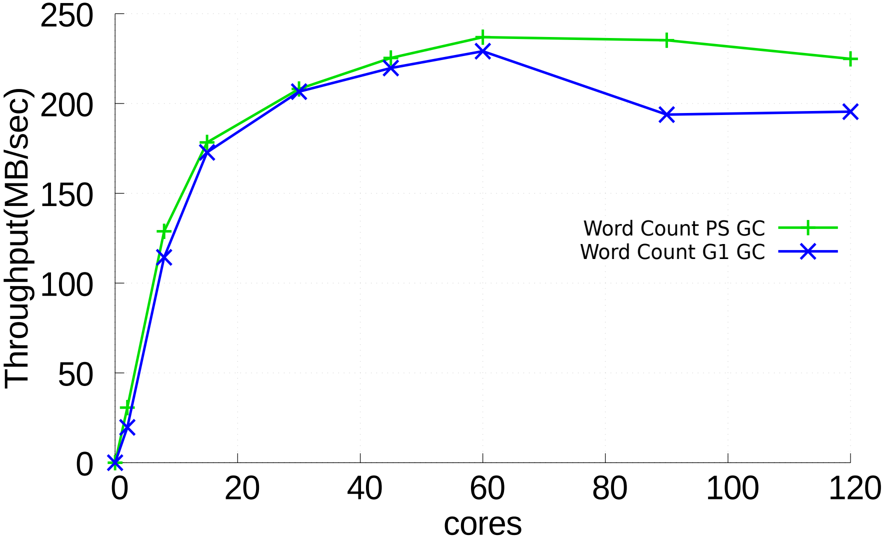
The General Scalability Problems – Previous Research

- Garbage Collection(GC) overheads.
- Locality of memory accesses on Non-Uniform Memory Access(NUMA) architecture.

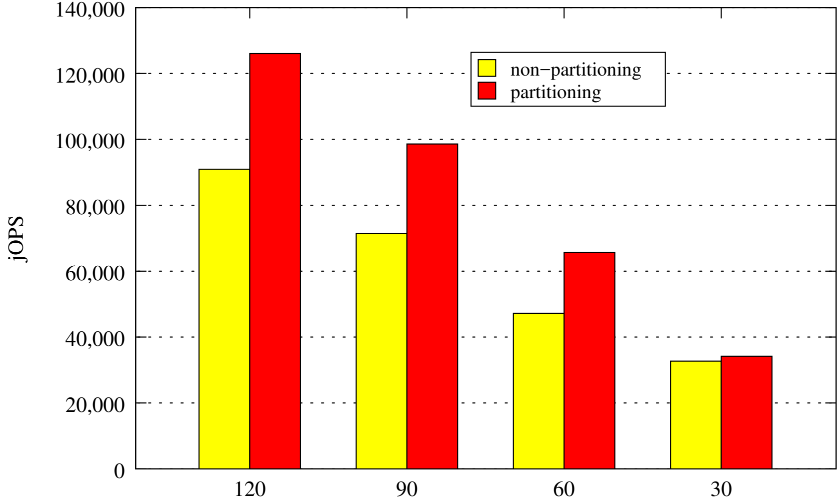
Effect of Garbage Collection



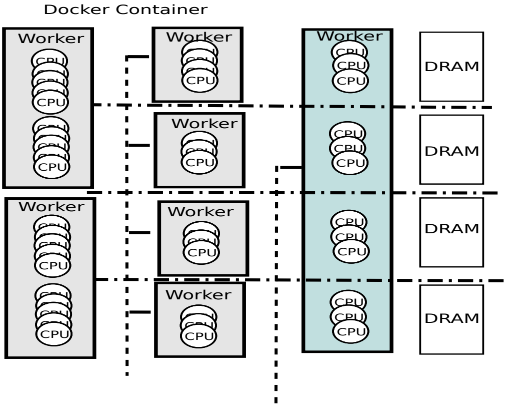
Effect of Garbage Collection



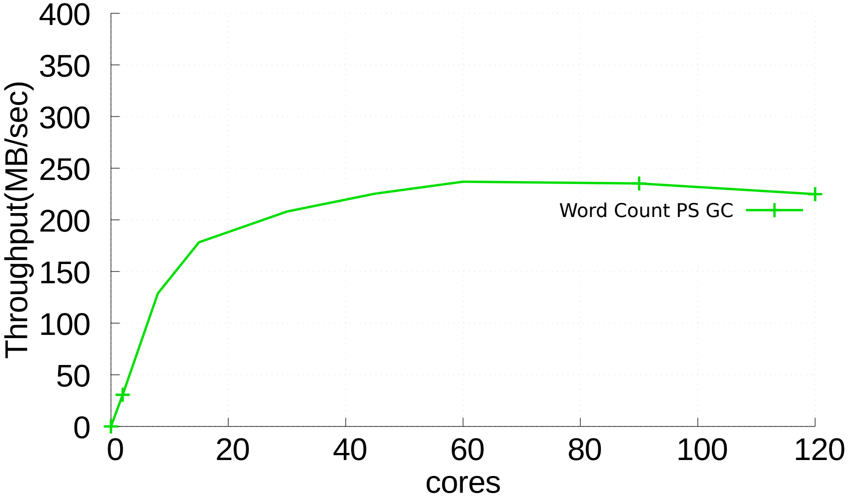
Benefit of JVM Partitioning



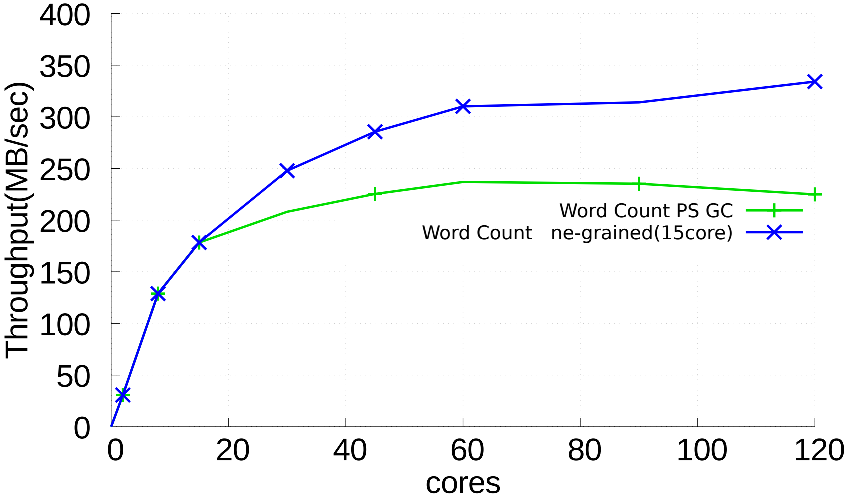
Proposed Solution - Docker container-based partitioning



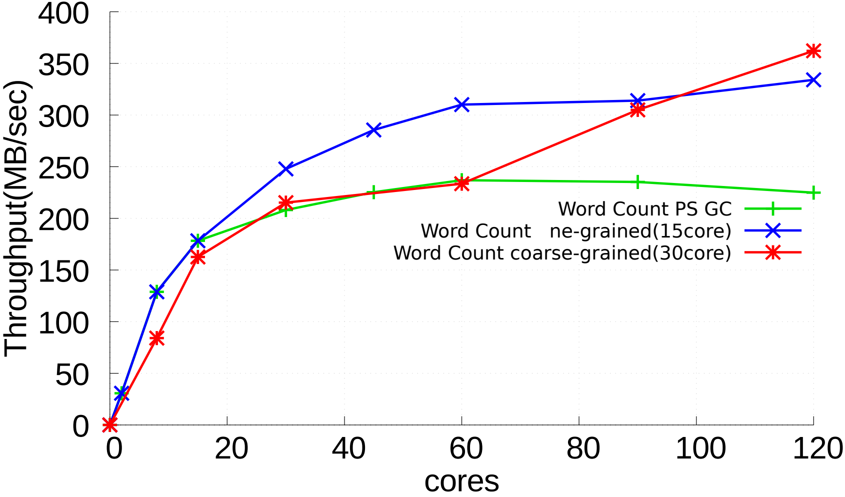
Benefit of Container-based Partitioning - WC



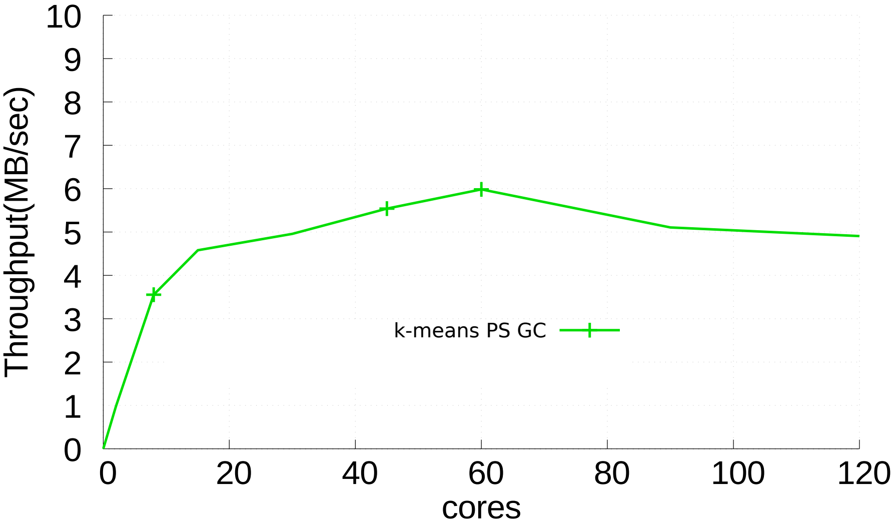
Benefit of Container-based Partitioning - WC



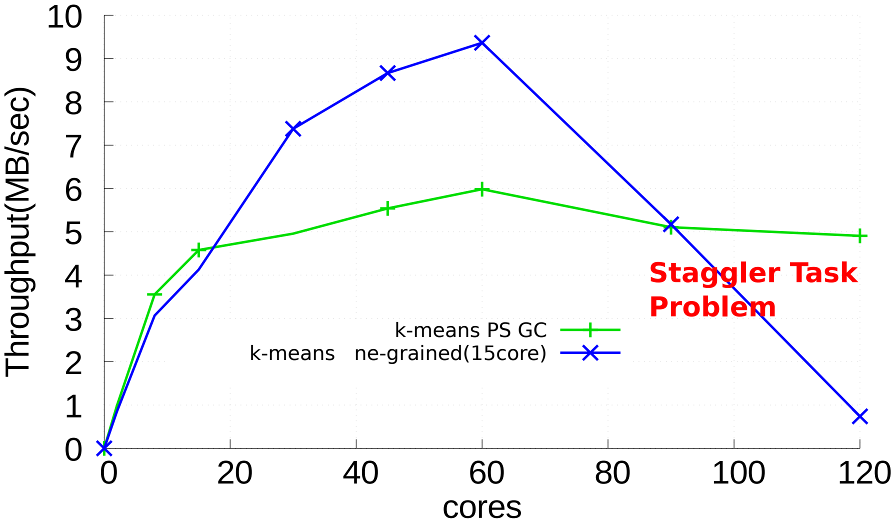
Benefit of Container-based Partitioning - WC



Benefit of Container-based Partitioning - K-means

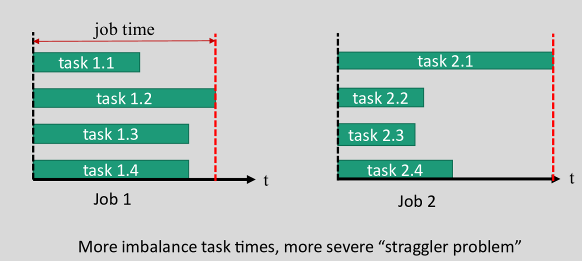


Benefit of Container-based Partitioning - K-means



Staggler Task Problem

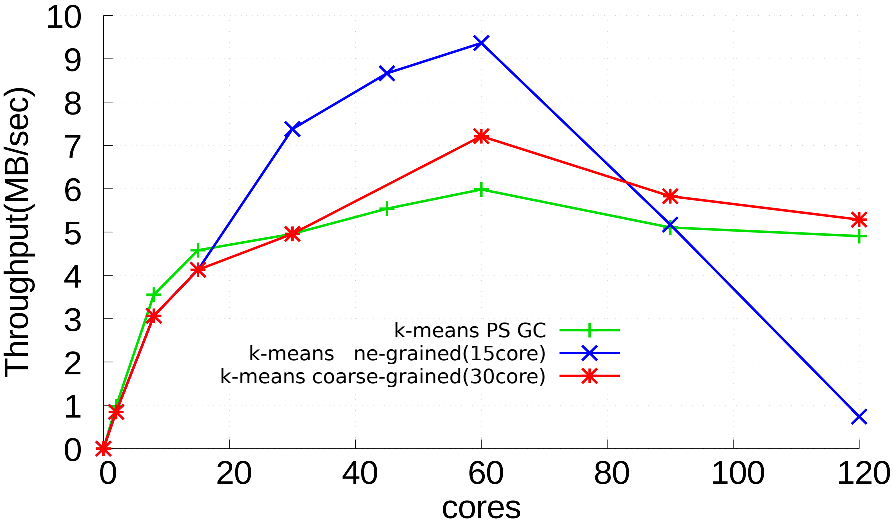
What is Straggler Tasks Problem?



More imbalance task times, more severe “straggler problem”

Source: http://slideplayer.com/slide/3315804/

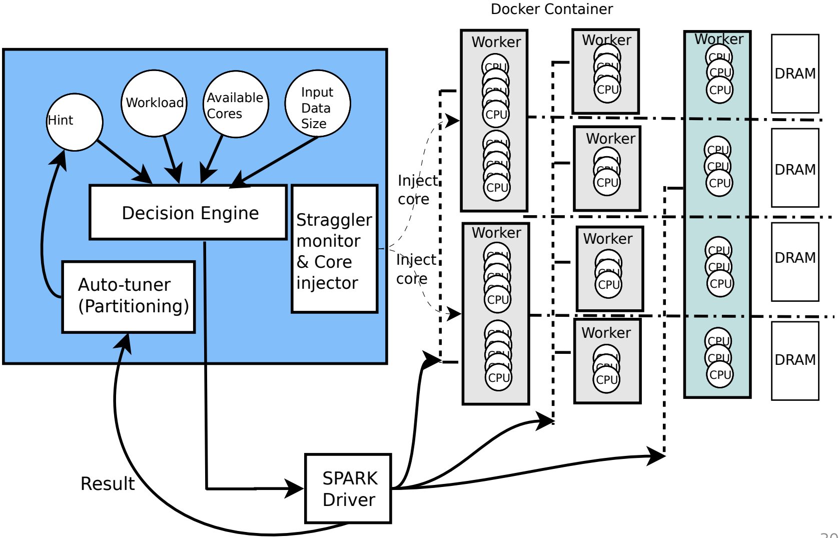
Benefit of Container-based Partitioning - K-means



Proposed architecture – Design Consideration

- Finding best-fit CPU counts.
- Solving the straggler tasks(i.e, tasks take significantly longer than expected to complete) problem.
- Improving the NUMA locality.
- Avoiding operating systems noise.

Proof-of-concept architecture



Future Directions

- Implementing the proof-of-concept architecture.
- Solving the straggler tasks problem.

QnA

Q & A