Towards Benchmarking Modern Distributed Streaming Systems

Grace Huang (jie.huang@intel.com)

Jun 15, 2015

Acknowledgement

Dev team

- Shilei, Qian
- Qi, Lv
- Ruirui, Lu

Advisors

- Tathagata Das (Spark PMC member, committer)
- Saisai Shao
- Sean Zhong (Storm PMC member, committer)
- Tianlun Zhang

About US

Explore more
@ Intel Booth

Closely partnered with large web sites and ISVs on better user experiences

- Key contributions for better customer adoption. E.g., Usability, Scalability and Performance

- More utilities to improve the stability & scalability
  - HiBench: The Cross platforms micro-benchmark suite for big data (https://github.com/intel-hadoop/HiBench)
  - HiMeter: the light-weight workflow based big data performance analysis tool

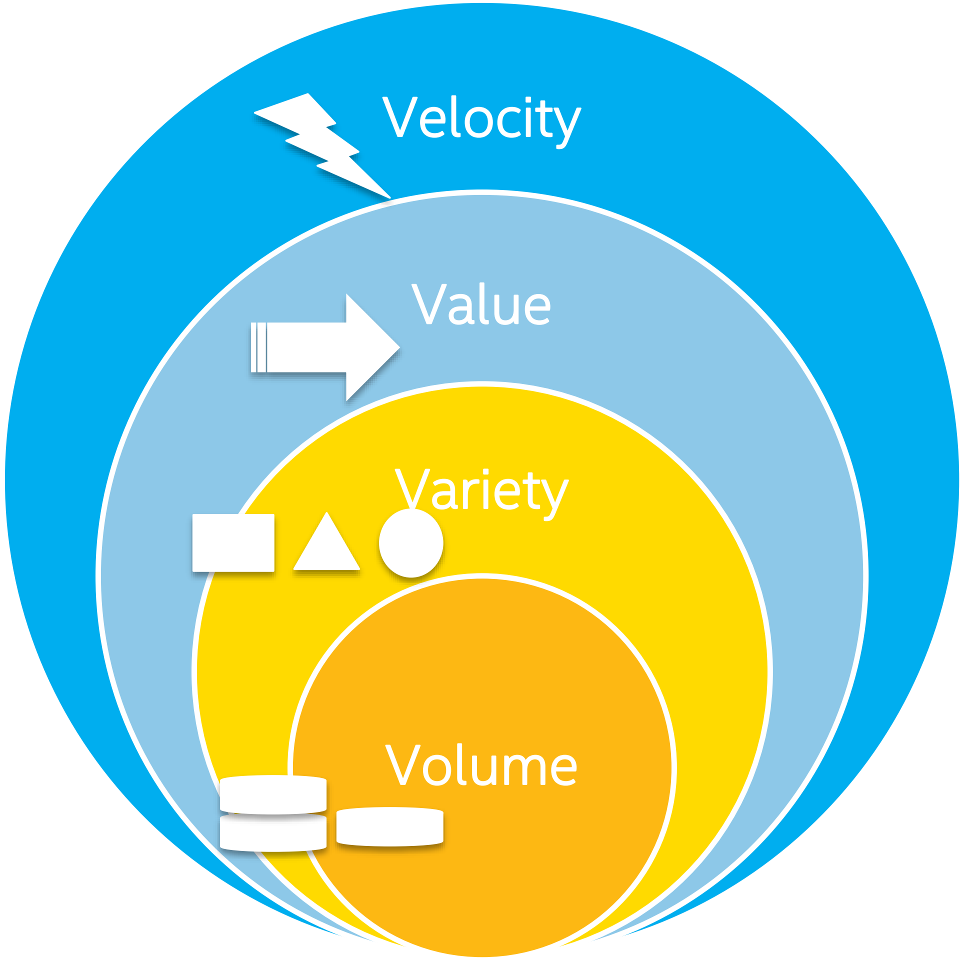
Agenda

- ❑ WHY we need *benchmarking for streaming system*?

- ❑ WHAT is *StreamingBench*?

- ❑ HOW to use it for Spark Streaming?

Time is Gold

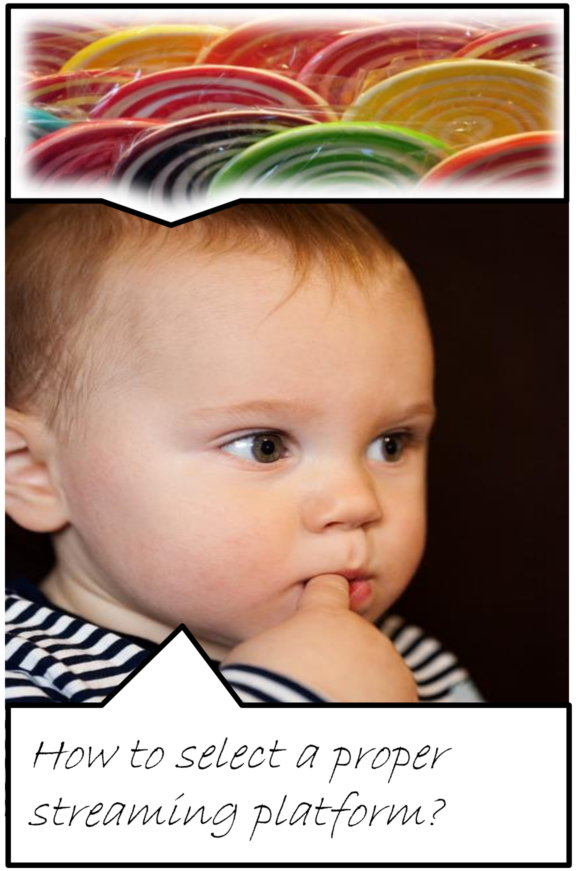


1. Big Data is flooding rather than streaming (FSI, IOT, …)

2. Dig out more profits from various streams in a real-time manner

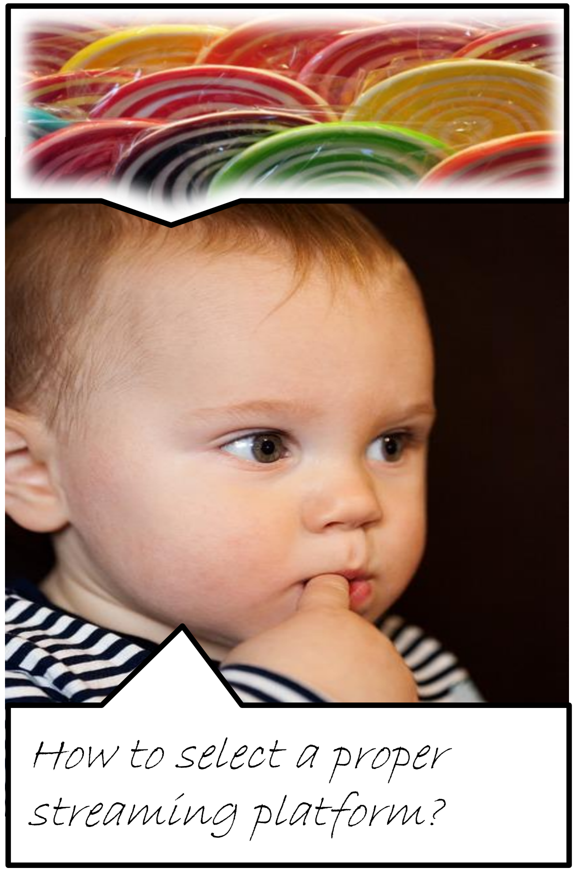
3. Streaming+X is blooming (W/ analytical query, graph, machine learning)

Frequent Questions from our Partners



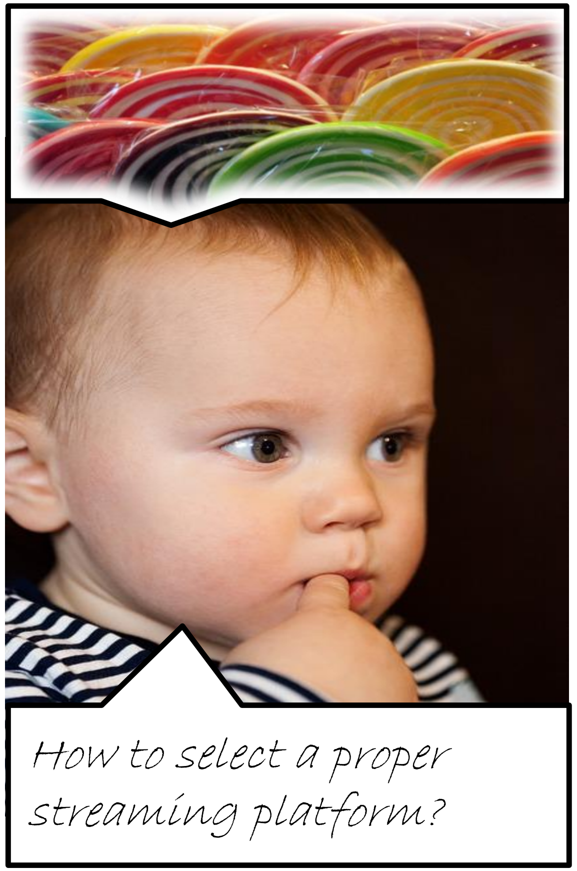
If you cannot measure it, you cannot improve it
---William Thomson

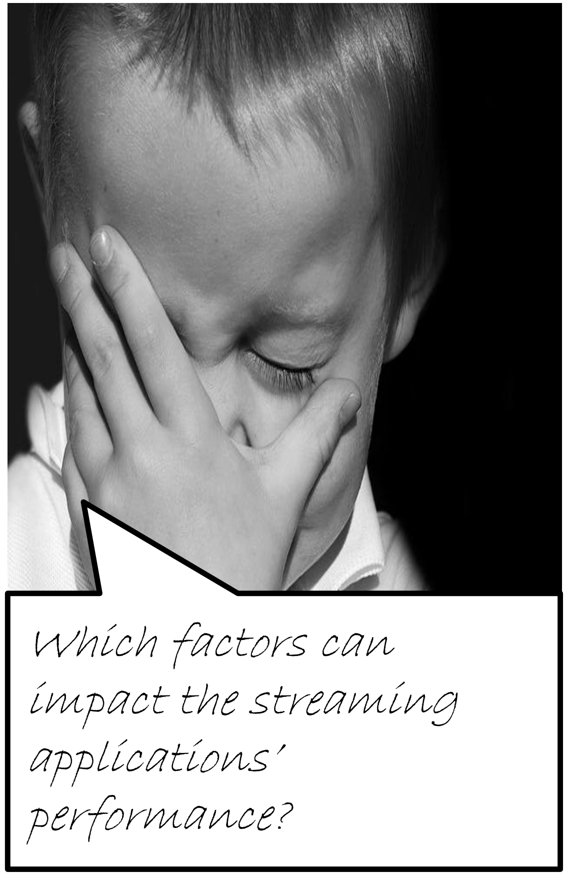
Frequent Questions from our Partners



If you cannot measure it, you cannot improve it
---William Thomson

Frequent Questions from our Partners




Which factors can impact the streaming applications' performance?

If you cannot measure it, you cannot improve it
---William Thomson

Frequent Questions from our Partners

How to select a proper streaming platform?
Is the streaming platform reliable and fault-tolerant?
Which factors can impact the streaming applications' performance?
How to properly select resources for the streaming applications?

If you cannot measure it, you cannot improve it
---William Thomson

Let’s meet StreamingBench

A streaming benchmark Utility consists of several micro workloads to

Users

1. Understand various streaming systems
2. Grasp tuning knobs
3. Allocate proper resources

Devs

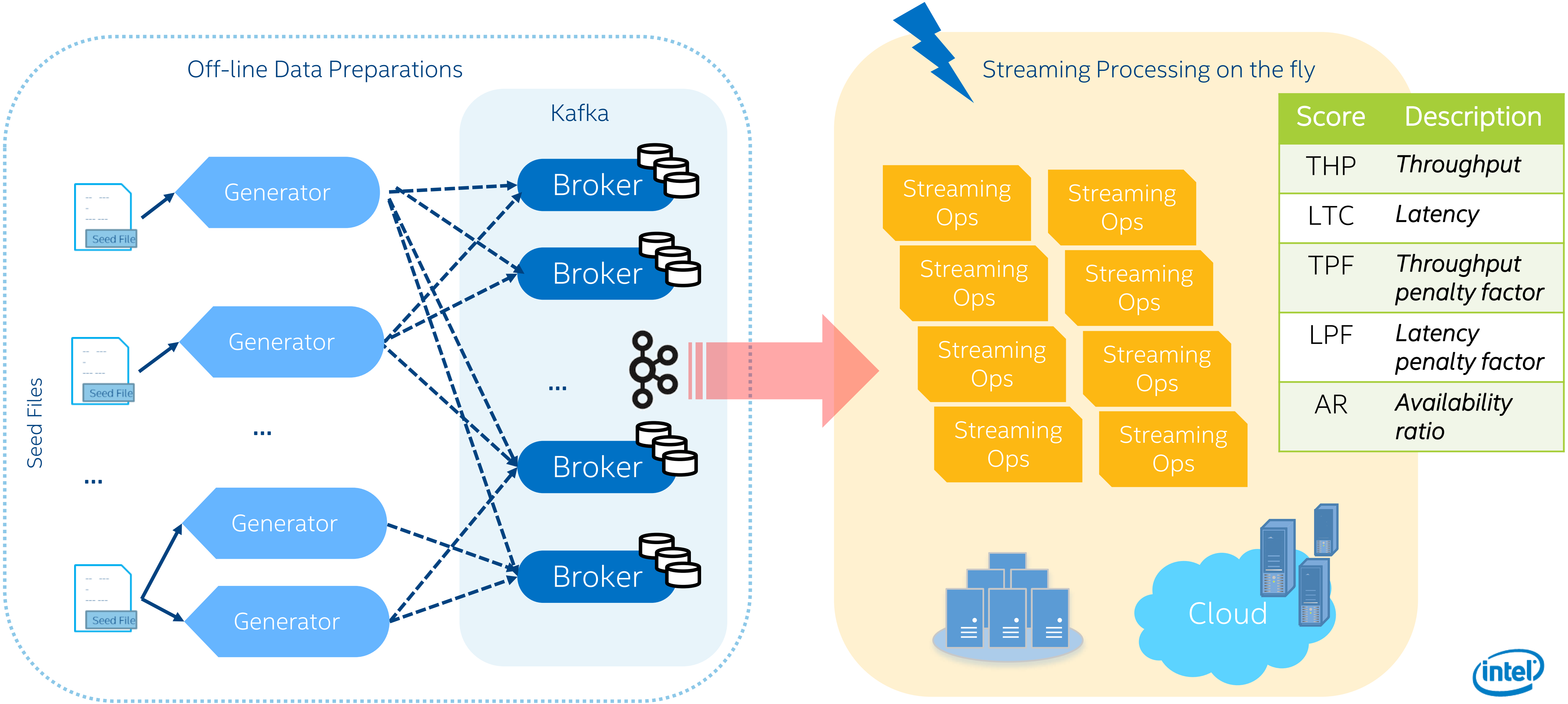
4. Improve streaming platforms further

Software

First Glance of StreamingBench

Workload	Rational	Complexity	Input data Seed
Identity	Directly emit the input record to output without any transformation	Low - Stateless computation	Average record size is 60 bytes in Text
Sampling	Select certain records from the input streams randomly.		
Projections	To collect a subset of columns for use in operations, i.e. a projection is the list of columns selected.		
Grep	To search streams for the occurrence of a string of characters that matches a specified pattern.		
Wordcount	Count the word occurrence number thru the entire stream	Medium – Stateful computation	
DistinctCount	Count the event number distinctly thru entire historical stream		
Statistics	The descriptive statistics including historical MIN, MAX and SUM based on the streaming data		Average record size is 200 bytes in numeric.

Architecture anatomy



Score	Description
THP	*Throughput*
LTC	*Latency*
TPF	*Throughput penalty factor*
LPF	*Latency penalty factor*
AR	*Availability ratio*

Important Statement:

The following part is not the answer to those questions, but just showcasing how to explore the answer.

Quick Guide

Systems Under Test*

Hardware

- 3-node Kafka cluster(48 partitions); 3-node streaming system(Spark Streaming, Storm, Trident and Samza);
- 108 cores, 384G Mem. 36x1T SATA HDDs per working cluster, 10G NIC per node

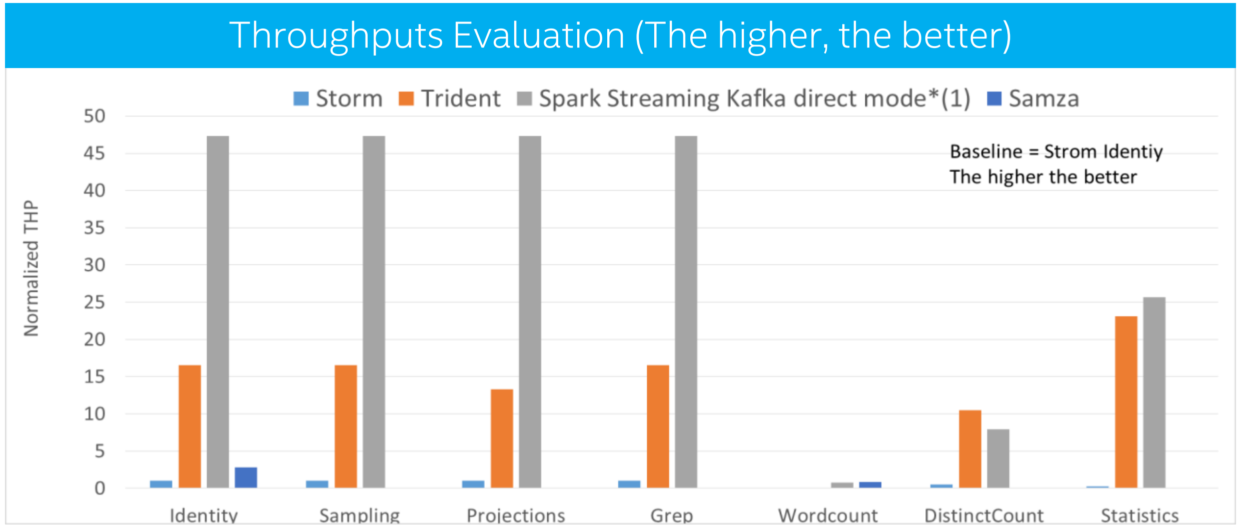
Software (more details in backup pages)

- Keep default configurations for all the test platforms(not tuned), except changing

1. Memory size, parallelism for each worker
2. Batch size, E.g.,
   - In Trident: 500MB fetchSizeBytes
   - In Spark Streaming: 1 second batch duration (with limited spark.streaming.kafka.maxRatePerPartition/spark.streaming.receiver.maxRate)
3. In Storm: TOPOLOGY_MAX_SPOUT_PENDING=1000
4. In Spark Streaming: Kafka direct mode;

1. How to select a proper streaming platform?

- Spark Streaming(Kafka direct mode):
  - ~2.5-3x Trident in “*stateless*” workloads(Network bound);
  - close to Trident in “*statefull*” workload (low parttion#)*
- Trident:
  - about x20 Storm (network bound with 500M fetchSizeByte);
- Storm
  - lowest among all for ACK ON for reliability
- Wordcount failed in Strorm/Trident
- Samza:
  - Performs 3x of Storm in Identity
  - Close to Spark streaming in Wordcount (low partition#)


Throughputs Evaluation (The higher, the better)

*See “*Which factors can impact the streaming applications’ performance*”
**with limited spark.streaming.kafka.maxRatePerPartition*

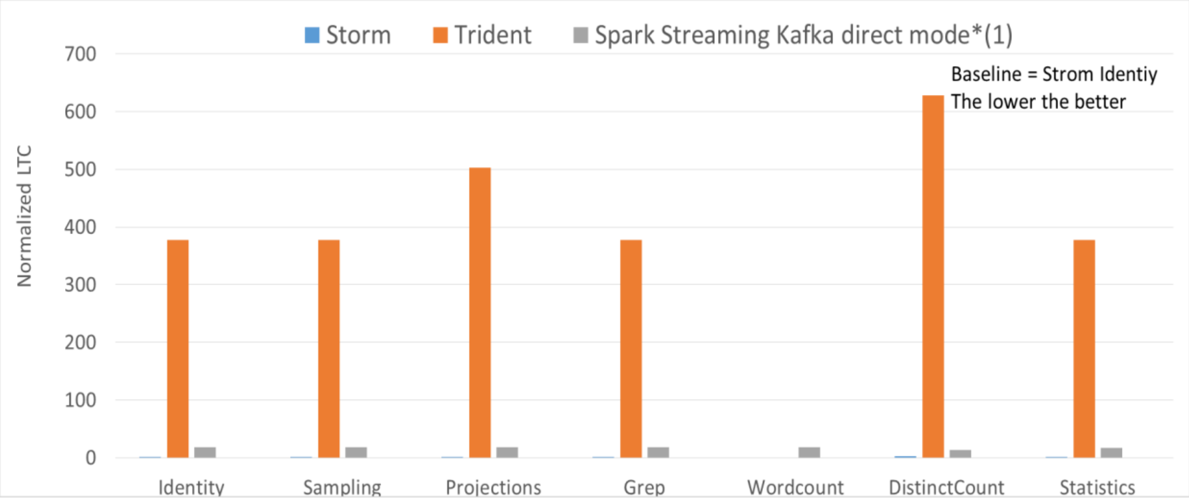
For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

1. How to select a proper streaming platform?

- Storms: the lowest response time (about 0.08-0.17 seconds)
- Spark Streaming: is controlled* around 1.5 second (batch duration), but with higher THP score.
- Trident: highest latency (about 30 seconds)
  - Since it fetches 500MB data in each partition.
  - Cutting down fetchSize leads to lower throughput, but better

  LTC score.

Latency Evaluation (The lower, the better)



$$\text{LTC} = \frac{1}{2}T_{data\ preparation} + T_{batch\ processing}$$

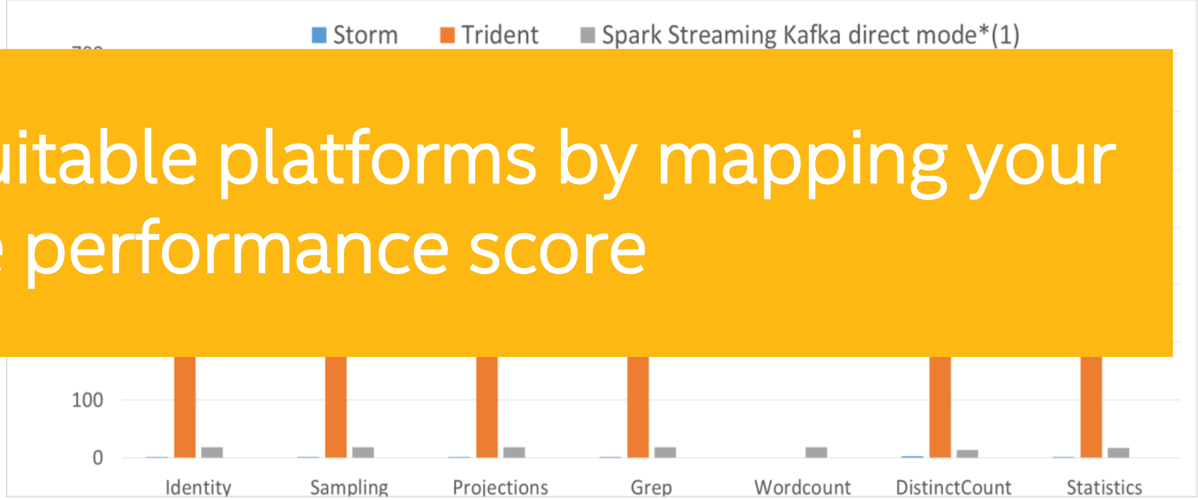
*with limited spark.streaming.kafka.maxRatePerPartition

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

1. How to select a proper streaming platform?

- Storms: the lowest response time (about 0.08-0.17 seconds)
- Spark Streaming: is controlled* around 1.5 second (batch duration), but with higher THP score.
- Trident: highest latency (about 30 seconds)
  - Since it fetches 500MB data in each partition.

Latency Evaluation (The lower, the better)



Key Learning: To find the most suitable platforms by mapping your requirement to the performance score

$$\mathrm{LTC} = \frac{1}{2} T_{data\ preparation} + T_{batch\ processing}$$

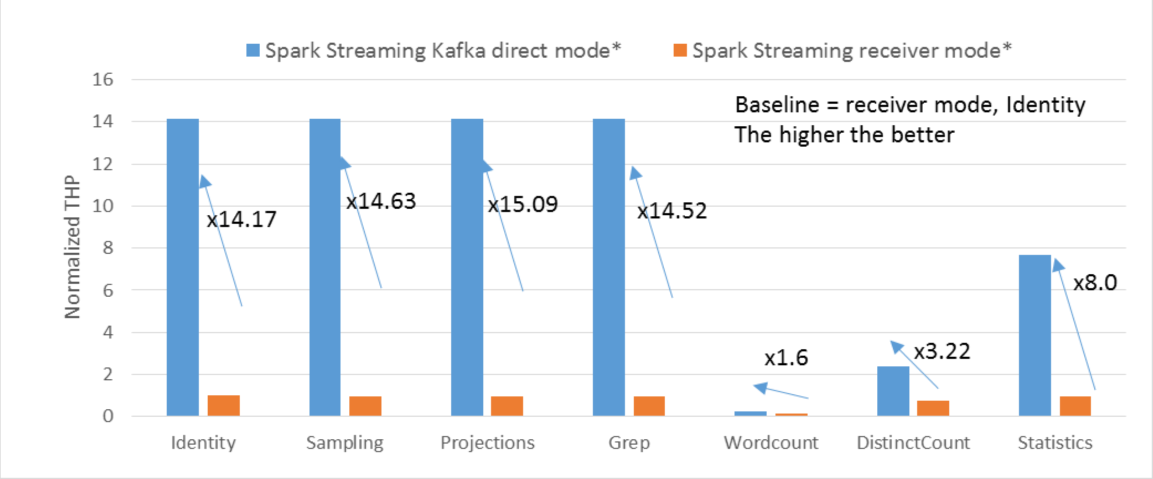
**with limited spark.streaming.kafka.maxRatePerPartition*

For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

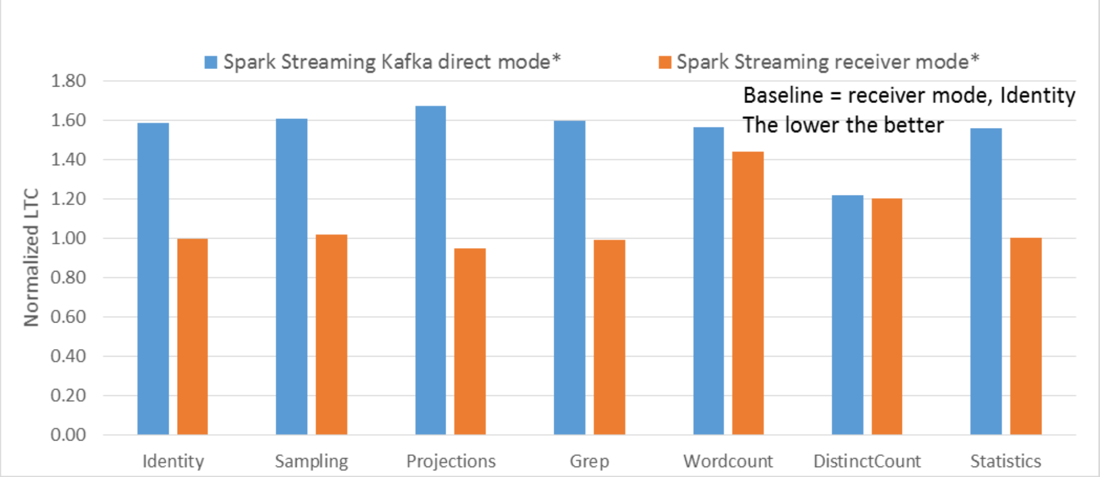
3. Which factors can impact the streaming applications' performance? (A)

1. The new experimental ***Kafka direct mode*** (1.3); while the old *receiver mode* is more general for all kinds of streaming sources.
2. It brings significant throughput improvements in Spark Streaming (WAL OFF)
3. Most of light-weight computation workloads in *direct mode* saturated 10G network bandwidth usage (100%).

THP score for various modes for Spark Streaming



LTC score for various modes for Spark Streaming



For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

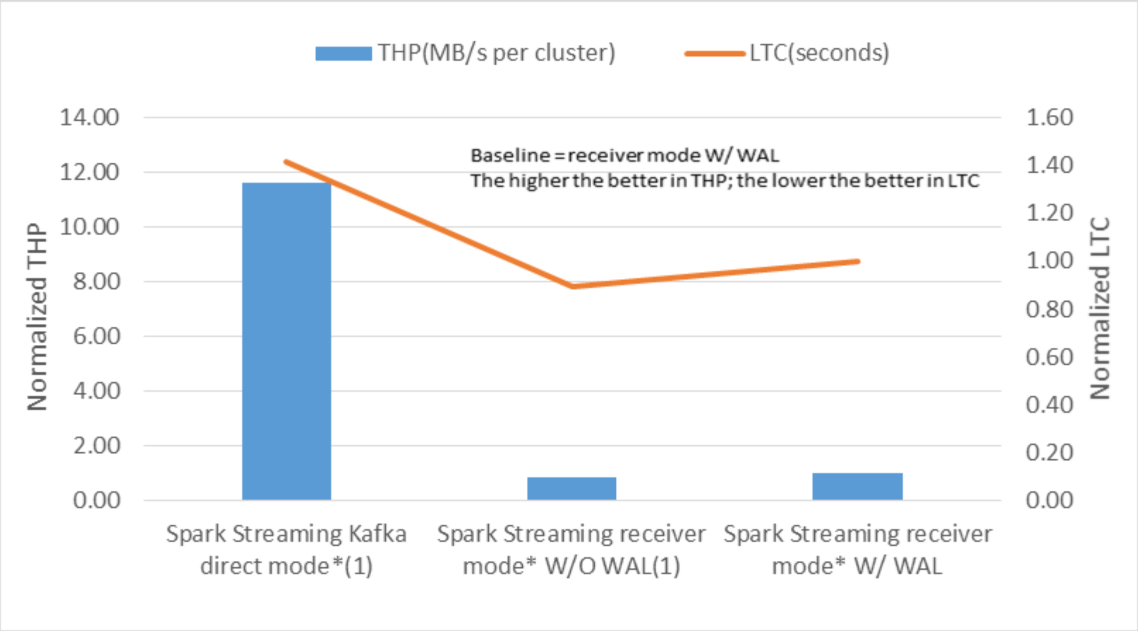
3. Which factors can impact the streaming applications' performance? (B)

- WAL doesn't bring any performance loss in *receive mode*(Identity)

  spark.streaming.receiver.writeAheadLog.enable = true

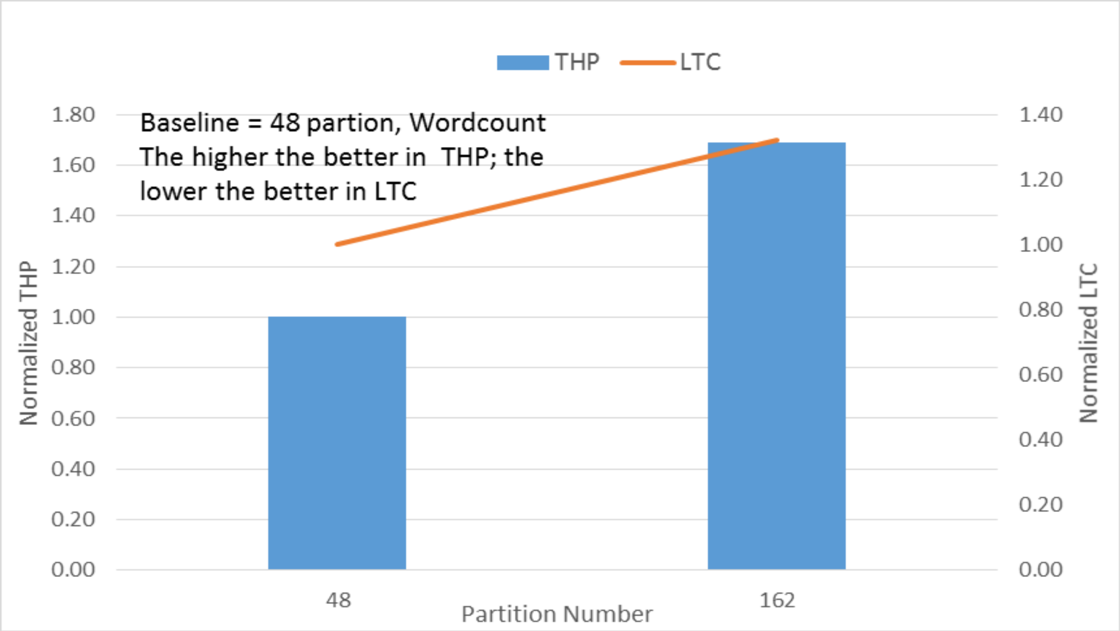
- Growing the partition number increases total throughput in Wordcount (*~1.67x*)

THP and LTC scores for Spark Streaming W/ and W/O WAL



Baseline = receiver mode W/ WAL
The higher the better in THP; the lower the better in LTC

THP and LTC scores in direct mode W/ different partition number



Baseline = 48 partion, Wordcount
The higher the better in THP; the lower the better in LTC

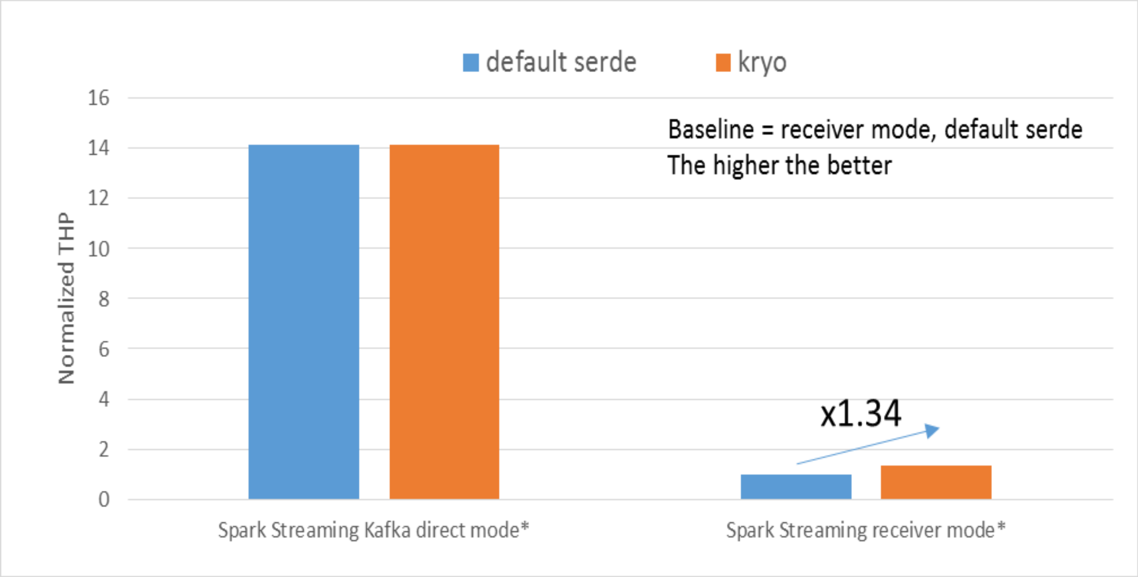
For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

3. Which factors can impact the streaming applications' performance? (C)

- Kryo serialization brings x1.34 throughput, x0.74 latency in *receiver mode*; (Identity)
- Since Kafka direct mode is network I/O bound, serialization doesn't change much.

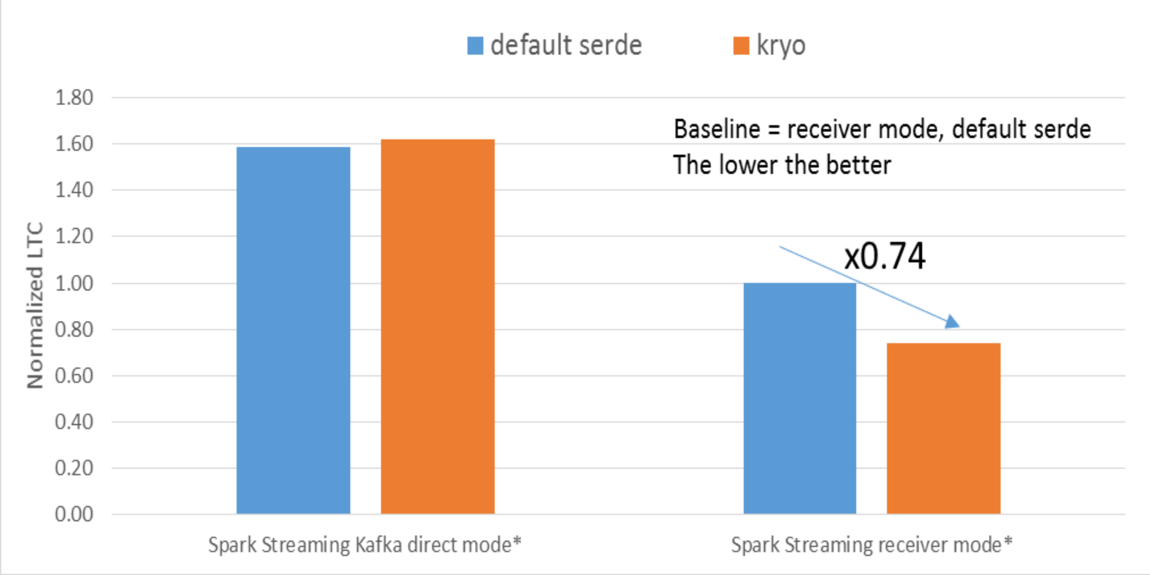
spark.serializer=org.apache.spark.serializer.KryoSerializer;
spark.kryo.referenceTracking=false

The total TPH score for various Serde in Identity



Baseline = receiver mode, default serde
The higher the better

The total LTC score for various Serde in Identity



Baseline = receiver mode, default serde
The lower the better

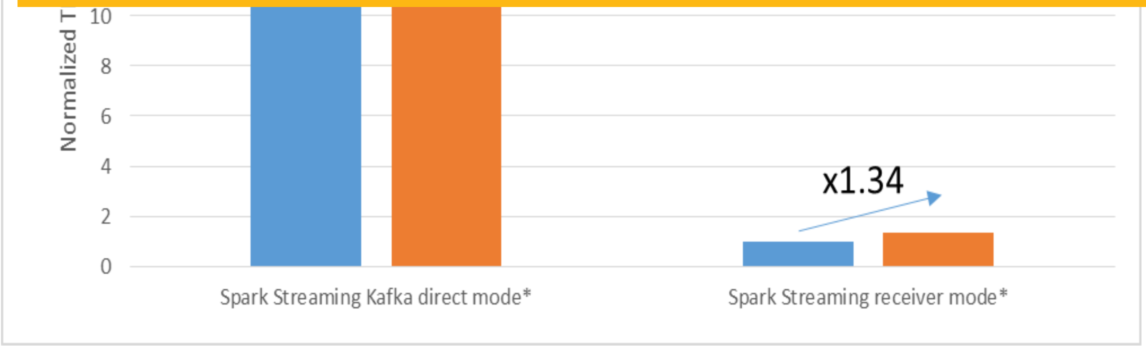
For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

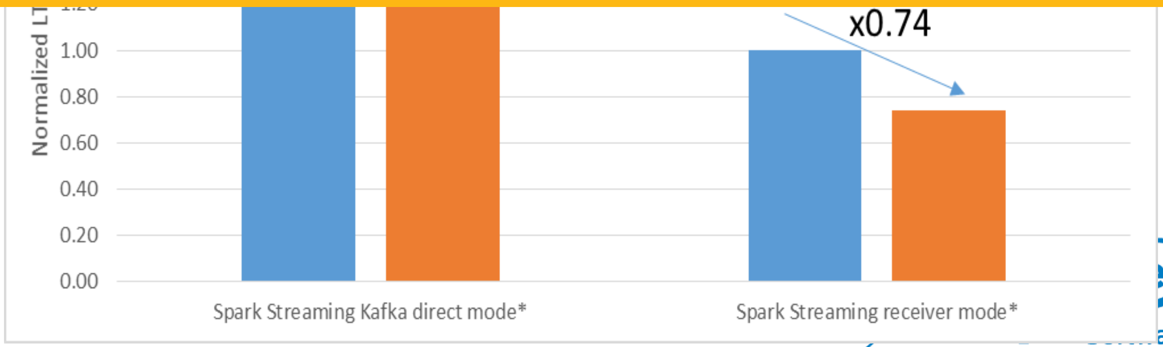
3. Which factors can impact the streaming applications' performance? (C)

- Kryo serialization brings x1.34 throughput, x0.74 latency in receiver mode; (Identity)
- Since Kafka direct mode is network I/O bound, serialization doesn't change much.

spark.serializer=org.apache.spark.serializer.KryoSerializer;
spark.kryo.referenceTracking=false

Key Learning: To simulate your workloads
and identify more key knobs





KEY TAKEAWAY

Public version is upcoming soon in

HiBench 5.0 release

http://github.com/intel-hadoop/HiBench

Stay tuned …

Call for more supports from YOU

More platforms

Enhance load generating

Better implementation

…

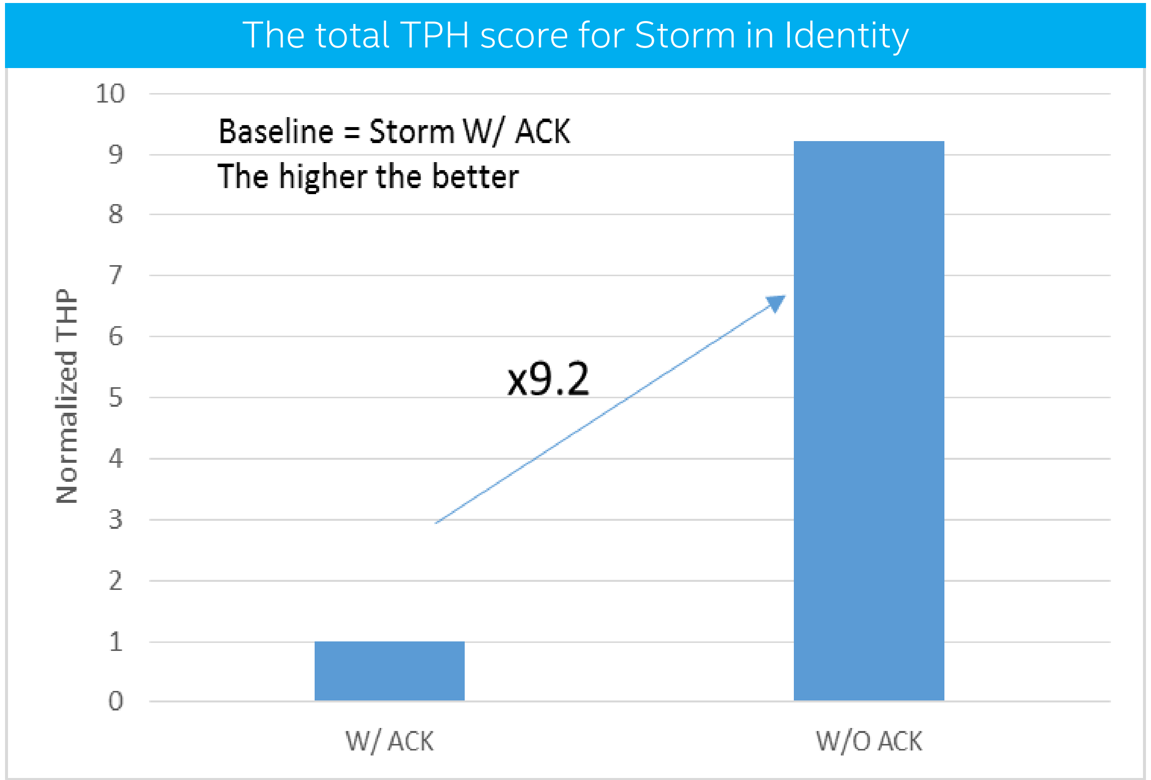
LISTEN to your voices and *LEARN* for more

BACKUP

3. Which factors can impact the streaming applications' performance? (D)

- Disabled ACK in Storm for Identity workload, brings x9.2 performance gain.
- To turn on ACK for better reliability. And Trident turns on ACK by default

To disable ACK by TOPOLOGY_ACKER_EXECUTORS=0



For more complete information about performance and benchmark results, visit www.intel.com/benchmarks.

Hardware/Software configurations

Node	3(data generators, i.e., Kafka) + 3(computing nodes) + 1(master) Dual Processor nodes
CPU	Xeon E5-2699 2.3GHz 18 physical cores
Memory	128GB
Hard disks	12 SATA HDDs 1T
NIC	10Gb

	Version	Other configurations

OS/kernel version	Ubuntu 14.04.2 LTS
	3.16.0-30-generic x86_64
		HugePage disabled;
		Ext4, noatime, nodiratime;
		ulimit –n 655360;

JDK version	Oracle jdk1.8.0_25	
Hadoop version	Hadoop-2.3.0-cdh5.1.3	

Streaming system configurations(1)

	Version	Other configurations
Storm, Trident	0.9.3	supervisor.slots.ports 6701 6702 6703 6704 nimbus.childopts -Xmx16g supervisor.childopts -Xmx16g worker.childopts -Xmx32g TOPOLOGY_ACKER_EXECUTORS*= 0 to disable ACK* TOPOLOGY_MAX_SPOUT_PENDING =1000
Spark	1.4 SNAPSHOT	Deployment: standalone mode SPARK_WORKER_CORES 72 SPARK_WORKER_MEMORY 100G SPARK_WORKER_INSTANCES 1 SPARK_EXECUTOR_MEMORY 100G SPARK_LOCAL_DIRS disk1-8 batch size: (changed that for healthy results) spark.streaming.kafka.maxRatePerPartition (direct-mode only) spark.streaming.receiver.maxRate (receiver-mode only) spark.serializer=org.apache.spark.serializer.KryoSerializer; spark.kryo.referenceTracking false spark.streaming.receiver.writeAheadLog.enable

Streaming system configurations(2)

	Version	Other configurations
Samza	0.8.0	yarn.nodemanager.aux-services: mapreduce_shuffle yarn.nodemanager.aux-services.mapreduce.shuffle.class: org.apache.hadoop.mapred.ShuffleHandler yarn.nodemanager.vmem-pmem-ratio: 3.1 yarn.nodemanager.vmem-check-enabled: false yarn.nodemanager.resource.memory-mb: 102400 yarn.scheduler.maximum-allocation-mb: 10240 yarn.scheduler.minimum-allocation-mb: 2048 yarn.nodemanager.resource.cpu-vcores: 40
Kafka	kafka_2.10-0.8.1	Each broker configuration: num.network.threads 4 num.id.threads 4 socket.send.buffer.bytes 629145600 socket.receive.buffer.bytes 629145600 socket.request.max.bytes 1048576000 log.dirs disk1-4 num.partitions 4 log.segment.bytes 536870912 log.retention.check.interval.ms 60000 zookeeper.connection.timeout.ms 1000000 replica.lag.max.messages 10000000

Software

Notices and Disclaimers

- INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.
  A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

- Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

- The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

- Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

- Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: http://www.intel.com/design/literature.htm

- Intel, the Intel logo, Intel Xeon, and Xeon logos are trademarks of Intel Corporation in the U.S. and/or other countries.

- Intel processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families: Go to: Learn About Intel® Processor Numbers http://www.intel.com/products/processor_number

- All the performance data are collected from our internal testing. Some results have been estimated based on internal Intel analysis and are provided for informational purposes only. Any difference in system hardware or software design or configuration may affect actual performance.

*Other names and brands may be claimed as the property of others.

intel®

Software