Handling V's of Big Data

Velocity, Volume and Variety

Eka Kurniawan

V's of Big Data

The FOUR V's of Big Data

From traffic patterns and music downloads to web history and medical records, data is recorded, stored, and analyzed to enable the technology and services that the world relies on every day. But what exactly is big data, and how can these massive amounts of data be used?

As a leader in the sector, IBM data scientists break big data into four dimensions: **Volume, Velocity, Variety and Veracity**

Depending on the industry and organization, big data encompasses information from multiple internal and external sources such as transactions, social media, enterprise content, sensors and mobile devices. Companies can leverage data to adapt their products and services to better meet customer needs, optimize operations and infrastructure, and find new sources of revenue.

By 2015
4.4 MILLION IT JOBS
will be created globally to support big data, with 1.9 million in the United States

Volume
SCALE OF DATA

40 ZETTABYTES
[43 TRILLION GIGABYTES]
of data will be created by 2020, an increase of 300 times from 2005

It's estimated that
2.5 QUINTILLION BYTES
[2.3 TRILLION GIGABYTES]
of data are created each day

6 BILLION PEOPLE
have cell phones

WORLD POPULATION: 7 BILLION

Most companies in the U.S. have at least
100 TERABYTES
[100,000 GIGABYTES]
of data stored

Variety
DIFFERENT FORMS OF DATA

As of 2011, the global size of data in healthcare was estimated to be
150 EXABYTES
[161 BILLION GIGABYTES]

By 2014, it's anticipated there will be
420 MILLION WEARABLE, WIRELESS HEALTH MONITORS

4 BILLION+ HOURS OF VIDEO
are watched on YouTube each month

30 BILLION PIECES OF CONTENT
are shared on Facebook every month

400 MILLION TWEETS
are sent per day by about 200 million monthly active users

Velocity
ANALYSIS OF STREAMING DATA

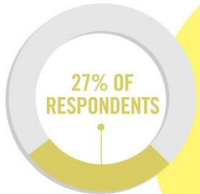
The New York Stock Exchange captures
1 TB OF TRADE INFORMATION
during each trading session

Modern cars have close to
100 SENSORS
that monitor items such as fuel level and tire pressure

By 2016, it is projected there will be
18.9 BILLION NETWORK CONNECTIONS
– almost 2.5 connections per person on earth

Veracity
UNCERTAINTY OF DATA

1 IN 3 BUSINESS LEADERS
don't trust the information they use to make decisions



in one survey were unsure of how much of their data was inaccurate

Poor data quality costs the US economy around
$3.1 TRILLION A YEAR

Sources: McKinsey Global Institute, Twitter, Cisco, Gartner, EMC, SAS, IBM, MEPTEC, QAS

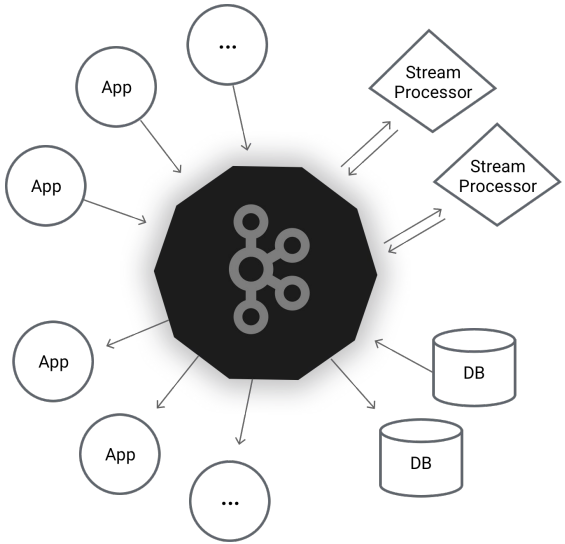
https://www.ibmbigdatahub.com/infographic/four-vs-big-data

IBM

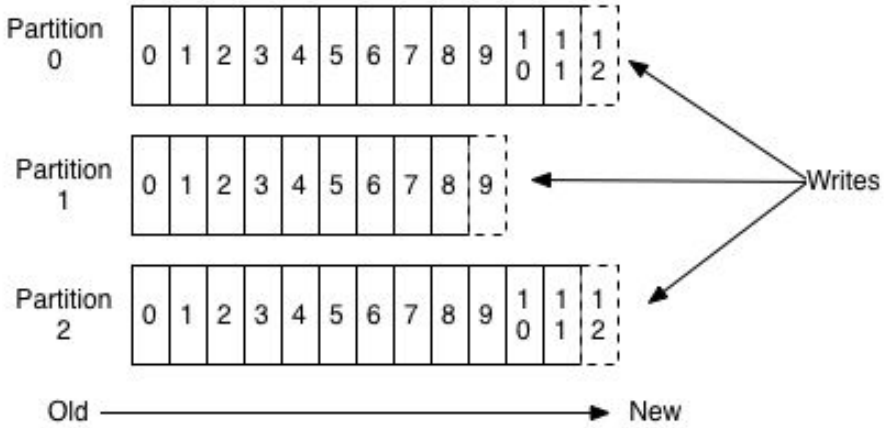
Handling Velocity

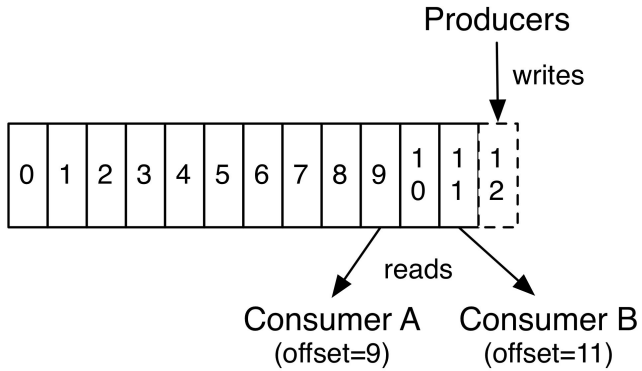
Apache Kafka

distributed streaming platform



https://kafka.apache.org/documentation/#gettingStarted
https://kafka.apache.org





Anatomy of a Topic

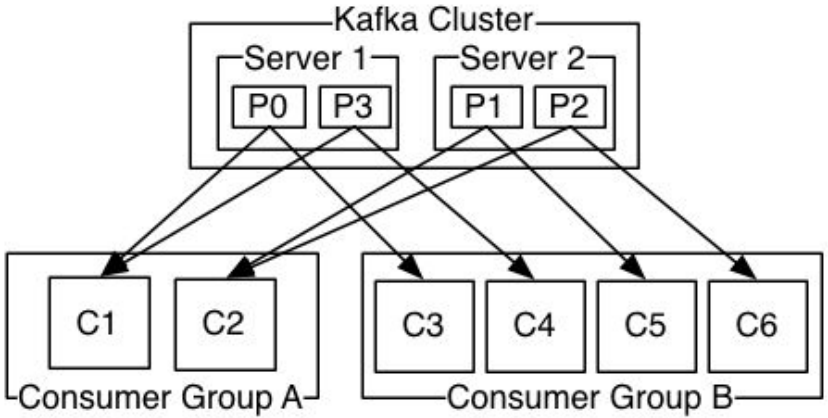
Topic

"category or feed name to which records are published"

Partition

"ordered, immutable sequence of records that is continually appended to"

https://kafka.apache.org/intro



Consumer Groups

https://kafka.apache.org/intro

Apache Kafka
Demo

https://github.com/bukalapak/BD-Handling-Vs#velocity-using-apache-kafka

Apache Kafka Benchmark Result

Produce and Consume Runtimes for Different Number of Messages using One Kafka Partition

Number of Messages	Total Messages Size (MB)	Produce Runtime (ms)	Consume Runtime (ms)
1	0.003	0.1	0.004
10	0.03	0.4	0.04
100	0.31	4	0.2
1000	3.13	46	3

https://github.com/bukalapak/O2O-bridge

Apache Kafka Benchmark CPU Utilization

CPU Utilization for Each Process during Producing and Consuming Messages

Process	CPU Utilization	
	Produce	Consume
Java	120%	77%
Python	48%	60%
docker-proxy	29%	25%

https://github.com/bukalapak/O2O-bridge

Apache Kafka Benchmark Environment

Hardware Specification and Software Version

Hardware	Specification
Processor	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
CPU Cores	6
CPU Threads	12
Memory	16 GiB
Storage Device	SSD

Software	Version
Python	3.6.6
confluent_kafka	0.11.6

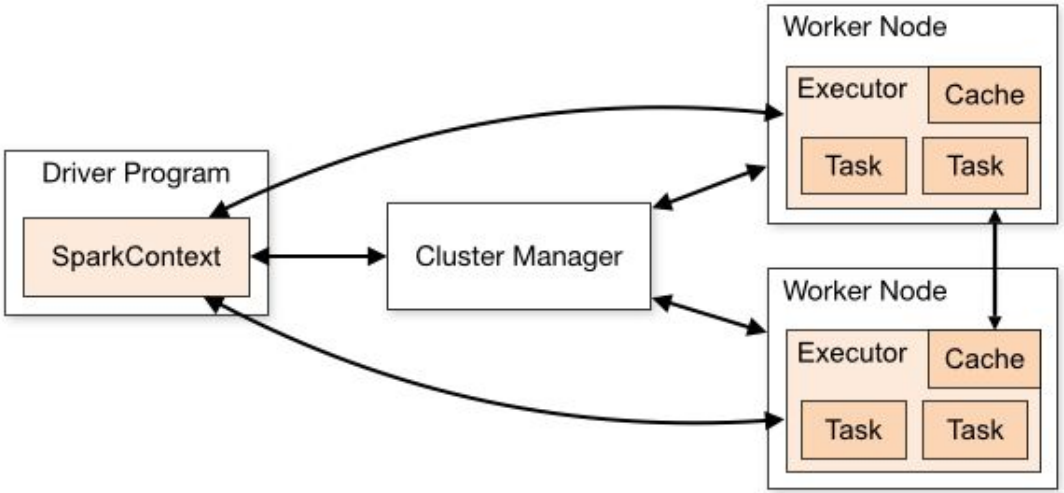
https://github.com/bukalapak/O2O-bridge

Handling Volume

Apache Spark

unified analytics engine for large-scale data processing

https://spark.apache.org



Cluster Overview

SparkContext

"coordinates Spark applications run as independent sets of processes on a cluster"

Executors

"processes that run computations and store data for your application"

https://spark.apache.org/docs/latest/cluster-overview.html

Apache Spark
Demo

https://github.com/bukalapak/BD-Handling-Vs#volume-using-apache-spark

Apache Spark Benchmark Result

Total Product Names to Process

Methods	Implementation	Total Product Names per Thread per Second
Hashing	Python Dictionary	25
MapReduce	Apache Spark	**191,167**

https://github.com/bukalapak/DL-Training-Rathirdtouille

Apache Spark Benchmark Environment

Hardware Specification and Software Version

Hardware	Specification	
	Hashing	MapReduce
Processor	Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz
Memory	256	16
Storage Device	HDD	HDD

Software	Version	
	Hashing	MapReduce
Python	3.5.3	3.6.3
TensorFlow	1.9.0	1.11.0
PySpark	n/a	2.3.1

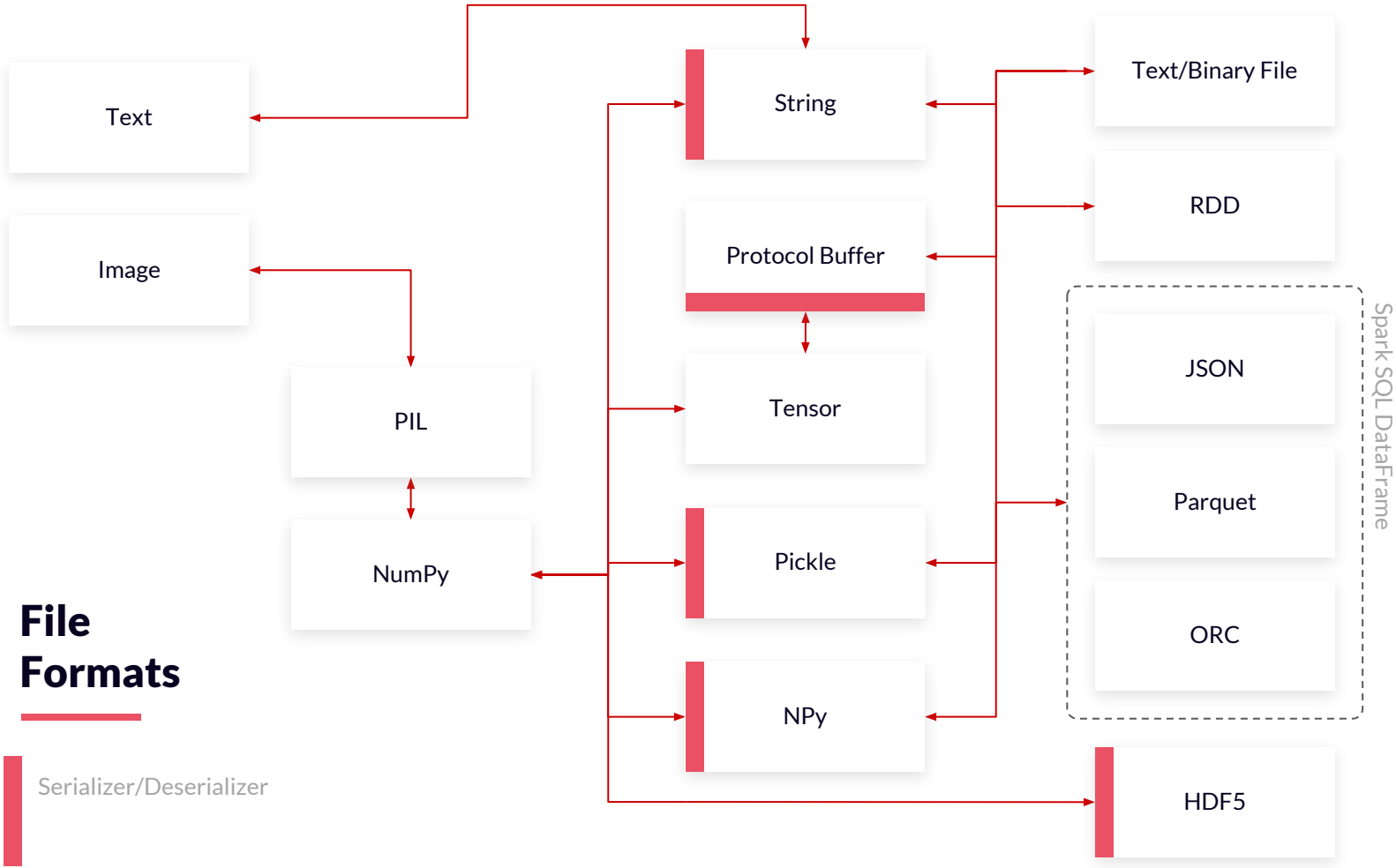
https://github.com/bukalapak/DL-Training-Rathirdtouille

Handling Variety

Apache Spark SQL

Apache Spark's module for working with structured data

https://spark.apache.org/sql/



File Formats

Serializer/Deserializer

Serializers
Demo

https://github.com/bukalapak/BD-Handling-Vs#variety-of-serializers

Serializers Benchmark Result

Serializing and Deserializing 224x224 Matrix of Integers

Serializers	Serializing Runtime (ms)	Deserializing Runtime (ms)	Storage Space (KB)
Protocol Buffer*	2.51	5.02	**151**
NPy*	0.787	0.557	269
Pickle*	**0.103**	**0.076**	**151**
HDF5**	3.78	1.87	153

* Write/read to/from memory
** Write/read to/from file

https://github.com/bukalapak/BD-Benchmark-Matrix-Serialization

Serializers
Benchmark Environment

Hardware Specification and Software Version

Hardware	Specification
Processor	Intel(R) Core(TM) i7-3615QM CPU @ 2.30GHz
CPU Cores	4
CPU Threads	8
Memory	8 GB 1600 MHz DDR3
Storage Device	SSD

Software	Version
Python	3.6.6
NumPy	1.14.5
TensorFlow	1.10.1

https://github.com/bukalapak/BD-Benchmark-Matrix-Serialization

File Formats
Demo

https://github.com/bukalapak/BD-Handling-Vs#variety-of-file-formats

File Formats Benchmark Result

Image Preprocessing of 1000 same images with 4 executors, 4 cores/executor and Pickle as the serializer

File Formats	Runtime/Image (ms)	Number of Partitions	Non-blocked Storage Space (MB)
JSON	**22**	20	809.4
Parquet	815	20	**2.84**
ORC	41	16	6.84

https://github.com/bukalapak/BD-Benchmark-Image-Scaling

File Formats Benchmark Environment

Hardware Specification and Software Version

Hardware	Specification
Processor	Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz

Software	Version
Python	3.6.3
PySpark	2.3.0
NumPy	1.14.0
PIL	4.2.1

https://github.com/bukalapak/BD-Benchmark-Image-Scaling

Conclusions

01 Big Data is here.

02 There are horizontally scalable solutions to handle the V's of Big Data.

03 Kafka produce can handle ~11 thousand messages/thread/second.
Kafka consume can handle ~205 thousand messages/thread/second.
Spark can handle ~191 thousand product names/thread/second.

04 No Silver Bullet. Data-mart stores convenient but redundant data.

05 Combination of Spark and TensorFlow is the foundation for Big Data and Big Compute.

Thank You

Eka Kurniawan

AI Engineer