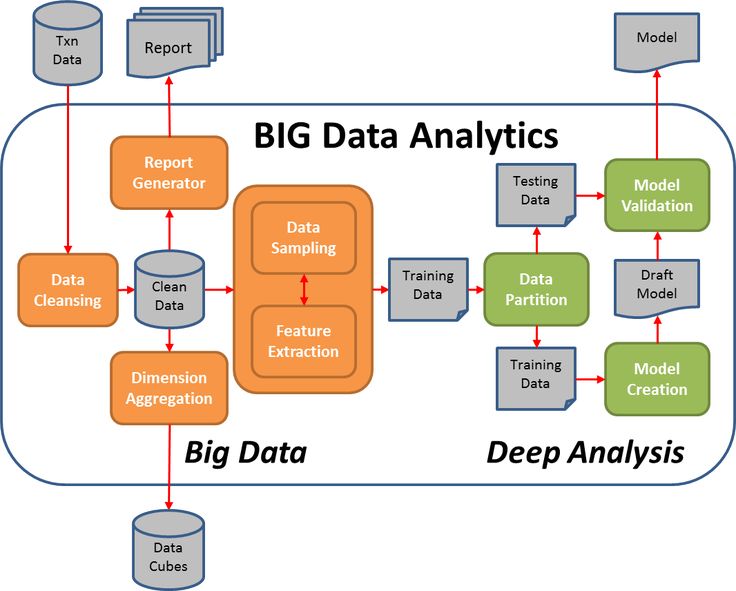
Big data analysis with IBM

cloud databases

Design:



Handle the ingestion, processing, and analysis of data that is too large or complex for A big data architecture is designed to traditional database systems. Big data solutions typically involve one or more of the following types of workload: Batch processing of big data sources at rest.

Big data analysis compression

data compression

In this section we will review a few compression techniques and demonstrate them through example. The primary insight that we will reach is that the best results are received by adapting the compression algorithm to the specific properties of the data at hand. A pinnacle example of this can be found in the distinction between lossy and lossless compression schemes.

Compression ratio:

This measures the reduction in the size of data as a result of compression.

Loss of information:

the degree to which the loss of information resulting from the compress Only relevant in the case of lossy compression, this measures ion impacts the quality of our data. There are many different ways of measuring this loss of quality, depending on the type of data, the domain, what the data will be used for, and more.

Overhead of compression:

Using a compression scheme implies the need to compress and uncompress the data at different stages of the pipeline. Both activities require a certain amount of compute resources and might imply a certain degree of latency. The precise amounts of required compute and latency can vary considerably based on the compression strategy .

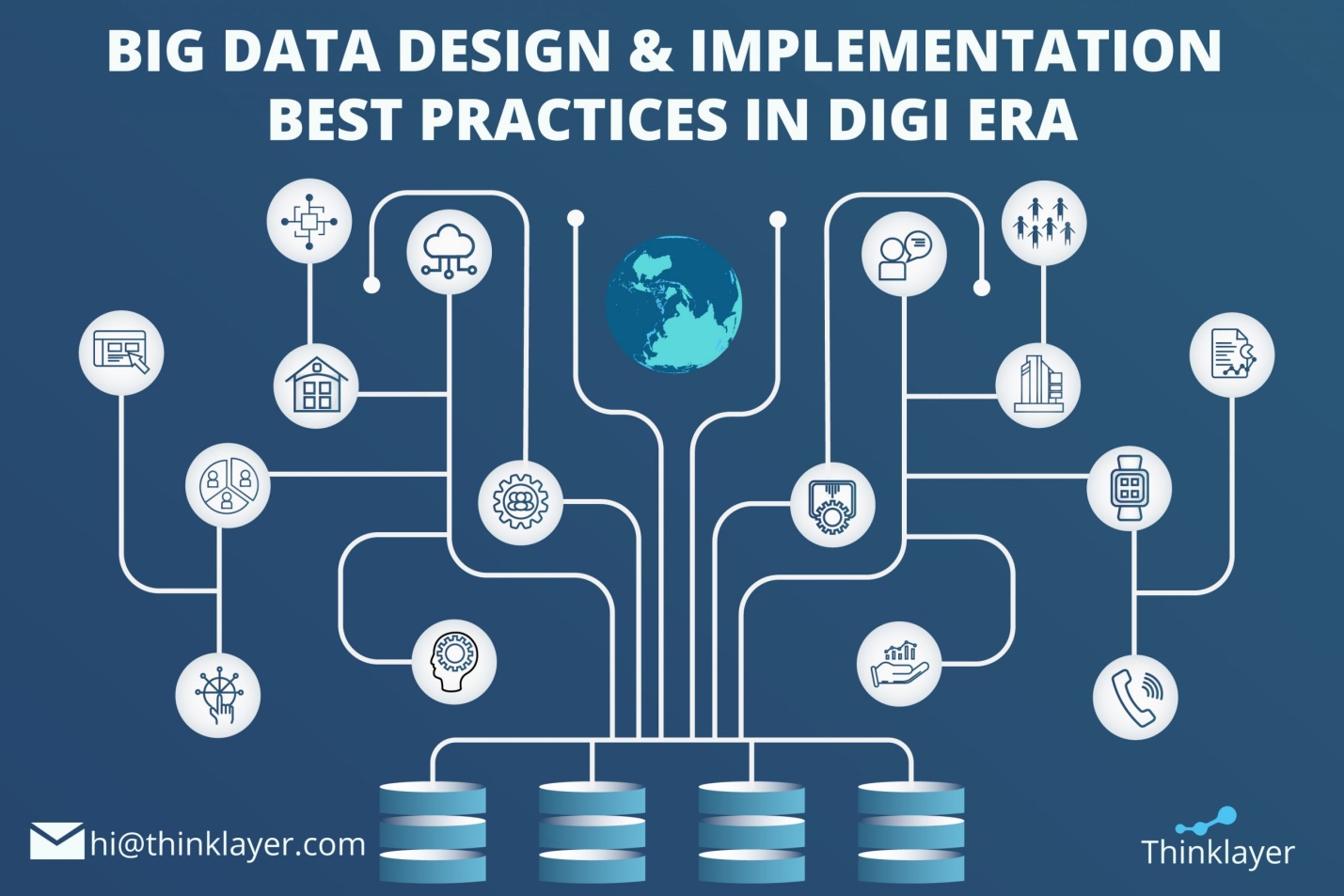
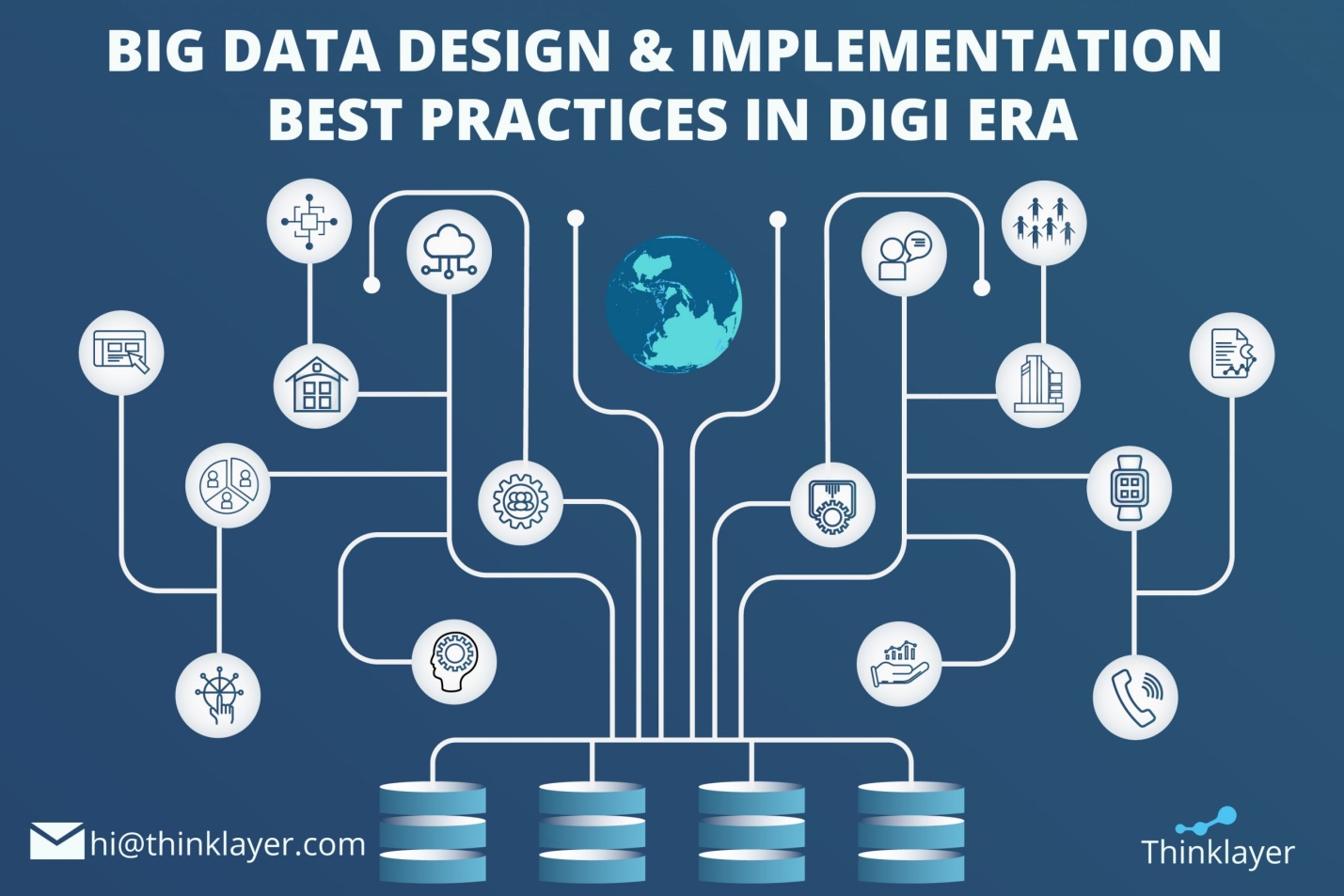
Infrastructure Dependencies:

Different compression schemes will vary based on their infrastructure dependencies. These can be HW dependencies and/or SW dependencies.

Characterstic of bigdata :

The first three characteristics of big data are volume, velocity, and variety. Additional characteristics of big data are variability, veracity, visualization, and value. Understanding the characteristics of Big Data is the key to learning its usage and application properly.

Implementation OF big data:



Key steps of big data implementation

\* Formulate and plan the concept.

\* Design the architecture.

\* Develop and perform QA.

\* Deploy the solution.

\* Provide support and maintenance.

\* Forming a high-level vision of the future big data solution, outlining:

\* Data processing specifics (batch, real-time, or both).

\* Required storage capabilities (data availability, data retention period, etc.).

\* Integrations with the existing IT infrastructure components (if applicable).

\* The number of potential users.

\* Security and compliance (e.g., HIPAA, PCI DSS, GDPR) requirements.

\* Analytics processes to be introduced to the solution (e.g., data mining, ML-powered predictive analytics).

\* Choosing a deployment model: on-premises vs. cloud (public, private) vs. hybrid.

\*Selecting an optimal technology stack.

program:

from PIL import Image

import numpy as np

np.random.seed(0)

im = Image.open('image.jpeg', mode='r')

image = np.array(im)

H,W,C = image.shape

# create artificial labels from image color channels

label1 = image[:,:,0].astype(np.int32)//16

label2 = image[:,:,1].astype(np.int32)//16

depth = (image[:,:,2]+np.random.normal(size=(H,W))).astype(np.float32)

# write all data sample elements to file

with open('image.bin','wb') as f:

f.write(image.tobytes())

with open('label2.bin','wb') as f:

f.write(label2.tobytes())

with open('label1.bin','wb') as f:

f.write(label1.tobytes())

with open('depth.bin','wb') as f:

f.write(depth.tobytes())

Big data analysis with IBM

cloud databases

Design:

Handle the ingestion, processing, and analysis of data that is too large or complex for A big data architecture is designed to traditional database systems. Big data solutions typically involve one or more of the following types of workload: Batch processing of big data sources at rest.

Big data analysis compression

data compression

In this section we will review a few compression techniques and demonstrate them through example. The primary insight that we will reach is that the best results are received by adapting the compression algorithm to the specific properties of the data at hand. A pinnacle example of this can be found in the distinction between lossy and lossless compression schemes.

Compression ratio:

This measures the reduction in the size of data as a result of compression.

Loss of information:

the degree to which the loss of information resulting from the compress Only relevant in the case of lossy compression, this measures ion impacts the quality of our data. There are many different ways of measuring this loss of quality, depending on the type of data, the domain, what the data will be used for, and more.

Overhead of compression:

Using a compression scheme implies the need to compress and uncompress the data at different stages of the pipeline. Both activities require a certain amount of compute resources and might imply a certain degree of latency. The precise amounts of required compute and latency can vary considerably based on the compression strategy .

Infrastructure Dependencies:

Different compression schemes will vary based on their infrastructure dependencies. These can be HW dependencies and/or SW dependencies.

Characterstic of bigdata :

The first three characteristics of big data are volume, velocity, and variety. Additional characteristics of big data are variability, veracity, visualization, and value. Understanding the characteristics of Big Data is the key to learning its usage and application properly.

Implementation OF big data:

Key steps of big data implementation

\* Formulate and plan the concept.

\* Design the architecture.

\* Develop and perform QA.

\* Deploy the solution.

\* Provide support and maintenance.

\* Forming a high-level vision of the future big data solution, outlining:

\* Data processing specifics (batch, real-time, or both).

\* Required storage capabilities (data availability, data retention period, etc.).

\* Integrations with the existing IT infrastructure components (if applicable).

\* The number of potential users.

\* Security and compliance (e.g., HIPAA, PCI DSS, GDPR) requirements.

\* Analytics processes to be introduced to the solution (e.g., data mining, ML-powered predictive analytics).

\* Choosing a deployment model: on-premises vs. cloud (public, private) vs. hybrid.

\*Selecting an optimal technology stack.

program:

from PIL import Image

import numpy as np

np.random.seed(0)

im = Image.open('image.jpeg', mode='r')

image = np.array(im)

H,W,C = image.shape

# create artificial labels from image color channels

label1 = image[:,:,0].astype(np.int32)//16

label2 = image[:,:,1].astype(np.int32)//16

depth = (image[:,:,2]+np.random.normal(size=(H,W))).astype(np.float32)

# write all data sample elements to file

with open('image.bin','wb') as f:

f.write(image.tobytes())

with open('label2.bin','wb') as f:

f.write(label2.tobytes())

with open('label1.bin','wb') as f:

f.write(label1.tobytes())

with open('depth.bin','wb') as f:

f.write(depth.tobytes())

THANK YOU

v