Integration of Apache Hadoop

Possible use of Apache Hadoop for long-term storage of iba data

White Paper

Issue 1.0

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG

Koenigswarterstr. 44

90762 Fuerth

Germany

Contacts

Main office	+49 911 97282-0
Fax	+49 911 97282- 33
Support	+49 911 97282- 14
Engineering	+49 911 97282- 13
E-mail	iba@iba-ag.com
Web	www.iba-ag.com

© iba AG 2021, All rights reserved.

Authors

Fabian Mielke, Software Developer

Jonas Silvennoinen, Software Developer

Issue	Date	Author	Revision
1.0	10- 2021	FM	First edition

Content

1 Introduction

Industrial production generates a large amount of different technical operating data. This group also includes process data, measurement data and metadata. The process data of a plant are usually acquired cyclically, while measurement data can belong to a process as well as to a product or batch. Metadata, such as product numbers, customer numbers, or other additional information will also be added. Technical operating data can be recorded, stored, analyzed and visualized with the *iba* system. In general, detailed data recording is necessary for quality documentation. It is also useful to identify potential in process and quality improvement.

The amount of data generated in a company increases rapidly with its size and level of digitalization. The effort and complexity of data management also increase. Therefore, this white paper explores the question of how users of *iba* software can handle such requirements and what solutions are available to manage very large amounts of data. For this purpose, investigations were carried out at *iba* as part of the *NewTech4Steel [1]* research project. This white paper includes basic information about *Apache Hadoop*, the results of the project, and recommendations for *iba* users who have an enormous amount of data or plan to use *Apache Hadoop* to store and analyze measurement data. It also follows on from the previous white paper regarding the *ibaHD-Server benchmark*. The results of the *ibaHD-Server benchmark* have shown that *ibaHD-Server* is very well suited for high-frequency data recording due to its efficient storage algorithm [2]. The focus in this document is now on scalability and solutions for *big data*.

The following evaluations and investigations were carried out within the framework of the *NewTech4Steel [1]* research project at *iba*. It was examined whether and in which use cases it can be an advantage to use *Apache Hadoop* in addition to the *iba* system. The aim of the European research project *NewTech4Steel* is to improve steel production processes in terms of stability and product quality through disruptive, data-driven technologies and to communicate the benefits to the entire European steel industry. One of the technologies to be investigated is data collection and analysis using the *big data* concept. *NewTech4Steel* is part of the *Research for Coal and Steel Programme (RFCS-2017)* and will run for 42 months. The coordinator is the *VDEh-Betriebsforschungsinstitut GmbH* (BFI). *iba* AG is part of the project consortium and contributes to *NewTech4Steel* with its know-how on data acquisition and the *iba* system. A dedicated computer cluster was set up at *iba*, for example, enabling all project partners to implement *big data* applications. This cluster system was also used for the *Apache Hadoop* research. [1]

1.1 Big data and scalability

When the amount of data in a company reaches a particularly large volume, increases rapidly and includes diverse data formats, this is referred to as a *big data* scenario. There is no explicit definition, but the common criteria for *big data* are often the three properties *volume*, *variety* and *velocity*. Analyzing data with these properties in a conventional file or relational database system is complex to impossible. Conventional hardware may no longer be suitable for processing data on this scale. When the data volume achieves a magnitude of about 100 TB, specialized and optimized relational database systems reach their architectural and technical limits. As the volume of data increases, so does the effort required to keep the data operationally available and consistent. Relational databases of this size are customized and require cost-intensive hardware. If the existing performance is no longer sufficient, the hardware must be adapted. These adjustments through upgrading are called *vertical scaling*. [3]

1.2 Apache Hadoop

Apache Hadoop is a software designed for storing and analyzing enormous amounts of data. The software offers its own file system with the *Hadoop Distributed File System* (HDFS) as well as its own resource management (YARN).

Data can be distributed on a *computer cluster* and processed in parallel with *Hadoop*. The cluster consists of at least one *node* and can be scaled arbitrarily, for example by increasing or reducing the number of *nodes*. Each additional *node* added to the cluster increases computing power and storage space. The concept of using nodes also offers advantages in the distribution of computing power and can compensate failures. [4] Parallel processing of data on multiple cluster *nodes* enables *Hadoop* to be enormously powerful, and it becomes more efficient with each additional *node*.

Hadoop is *horizontally scalable*, meaning it is possible to expand the cluster on the fly. Conventional computers can be used as new *nodes* in the process. An upgrade is not mandatory. If necessary, the number of *nodes* can also be reduced. The performance and capacity of the overall system can be adjusted with comparatively little effort. [3]

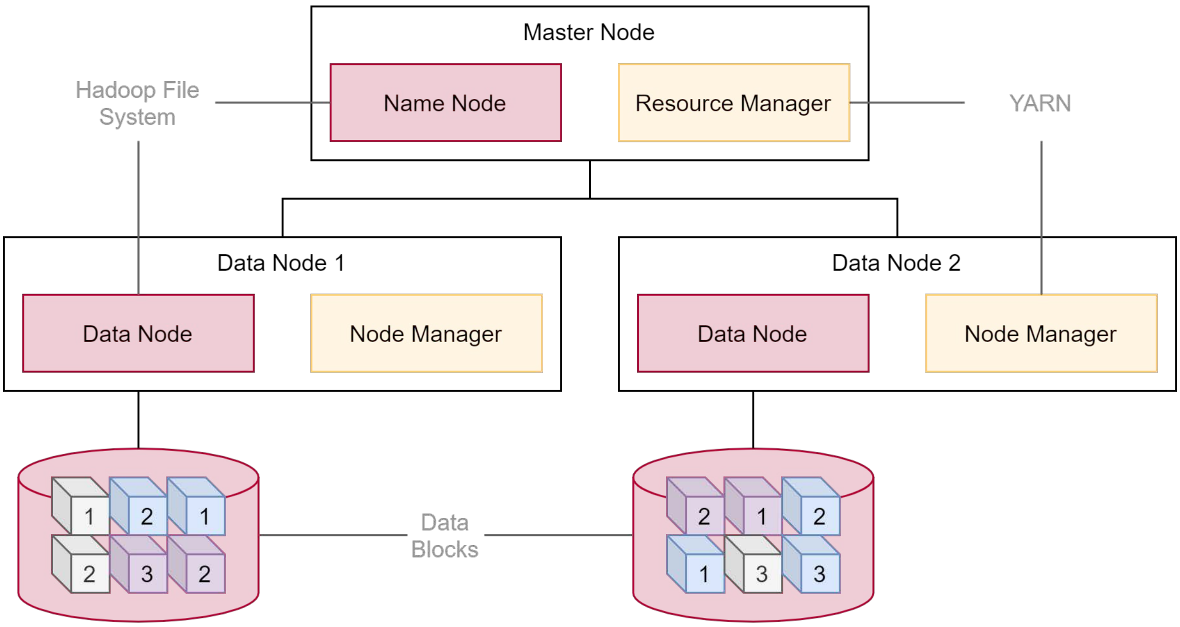

Fig. 1: Simplified structure of an Apache Hadoop cluster

To be able to process data in parallel, files in HDFS are divided into *data blocks* and replicated, as shown in the lower part of the Fig. 1, page 5 . *Replication* means that each block is duplicated and distributed to additional *nodes* to create redundancy. In principle, the HDFS supports all file types, but there are a number of particularly suitable formats. The difference is that these are divisible without loss of information and the individual blocks can be processed simultaneously by the *data nodes*. An example of a favorable format is *.csv*, since it is line-based and can be easily separated after each row of data.

There are also binary formats that are particularly suitable for the HDFS and *MapReduce*. Known representatives are *.avro* and *.parquet* files. *Apache* developed the *Parquet* format especially for fast, parallel processing with *Hadoop*. *Parquet* is a column-oriented format that can also be compressed and processed very efficiently. [17,21] All acquired data can be extracted into the *Parquet* format with the *iba* system. Unfavorable file formats for HDFS are those whose parts or blocks cannot be evaluated separately in a useful way. In this case, files must be joined together before processing using *SequenceFile*.

For some application scenarios, the use of *Hadoop* can be advantageous if *Hadoop* is used as a higher-level system and in addition to the *iba* system. In this combination, *ibaPDA* and *ibaHD-Server* carry out the recording and storage of measurement data. The *iba* data can then be extracted to *Hadoop* to perform joint analysis with other enterprise data. More information can be found in chapter ➚ *How can Hadoop be connected to the iba system?*, page 11.

It should also be pointed out at this point that it is not always sensible or associated with an increase in performance to introduce *Hadoop* for the analysis of company data. The reasons for this are listed in a separate section in chapter ➚ *When is Hadoop not suitable?*, page 13 .

2 For which use cases is Hadoop suitable?

In the following, the typical application areas of *iba* products are compared and classified with those of *Apache Hadoop*. We will then go into more detail about the use cases in which the use of *Apache Hadoop* can be useful as a supplement to the *iba* system.

In general, the domain of *iba AG* is the seamless and high-frequency recording of process and measurement data. The *iba* system for process data acquisition and analysis consists of perfectly-adjusted hardware and software components for acquiring, recording, analyzing and processing measurement data. The *iba* system can be adapted comfortably to various tasks due to the modular design and simple configuration. The recorded data can subsequently be used for quality documentation as well as for process analysis and troubleshooting with the help of the *iba* product portfolio. [5]

Data recorded with the *iba* system can be of high resolution and related to a plant or product. With the various *iba* software products, characteristic values and derived data can be calculated as well as extracted into a variety of external systems. The *.dat* file format and *ibaHD-Server* provide a consistent storage option within the *iba* system. Vice versa, the measurement results can be directly assigned to a process or a product and immediately queried again.

The domain of *Hadoop* is storing large amounts of data and analyzing it using *MapReduce jobs*. The *MapReduce jobs* are data processing tasks that can become arbitrarily complex and require a lot of computation time as they are executed over all data sets in HDFS. This computing time is expensive, especially when *Hadoop* resides in a managed cloud. *Hadoop* is not intended for time-critical messages or short-term decisions in industrial, production environments, but rather for medium-term analysis and statistics. The focus is not on exact documentation, but on bringing information together. In a sense, *Hadoop* relies on data sources and provides the ability to connect data to other arbitrary data sources if required. Some practical examples can be found on the *Apache* website. For example, *eBay* uses *Hadoop* to optimize product search, while *Facebook* analyzes user data and log files [6].

The specific advantages of HDFS over a local or network file system are particularly distinctive for large amounts of data and are listed below.

- Tolerance to system failures
- Parallel access to files through replication
- Higher performance when accessing files

The Fig. 2, page 8 compares the respective main features and strengths of the two systems. As a portfolio of comprehensive software solutions, the *iba* system offers a variety of analysis and reporting tools that the end user can use and operate directly after installation. The user also has the possibility to implement data evaluation and reporting according to his requirements with the *.dat* files and the *ibaHD-Server*. The strength of *Hadoop* lies mainly in its flexibility and scalability.

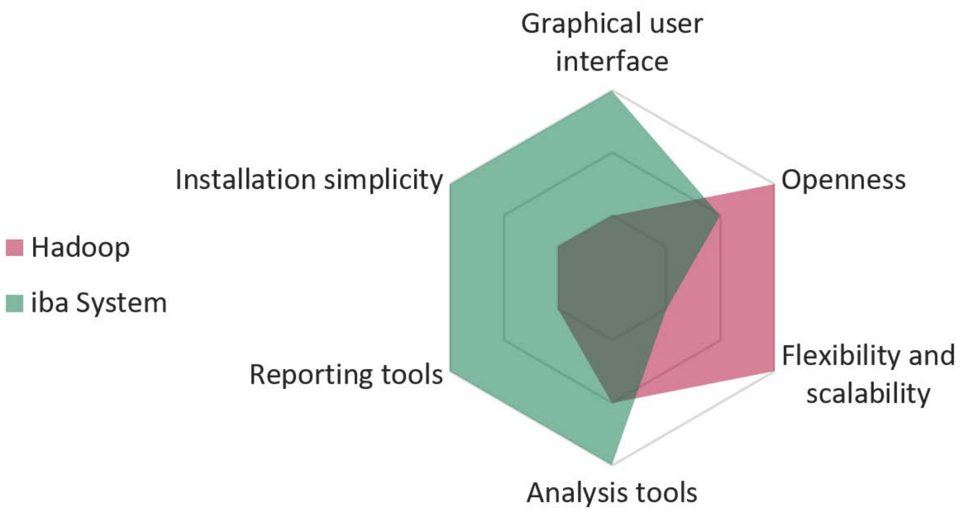

Fig. 2: Domains of the iba system and Apache Hadoop in comparison

2.1 Under what conditions does the combination of iba and Hadoop bring benefits?

In certain use cases, it may be beneficial to extract data from an *iba* system to *Hadoop*, thus combining both systems. The majority of the following requirements should be met in these use cases. The criteria refer to all company data, e.g. MES, log and *iba* data, which are to be evaluated with each other.

- There is enough expertise available on *Hadoop* or a corresponding cloud service.
- The data volumes are in the double-digit terabyte range.
- The company data are very heterogeneous.
- The company pursues an analysis strategy with target key figures.
- Production data is only recorded with the *iba* system and not analyzed.
- Data is to be stored centrally in a cloud or in a cluster and processed there.

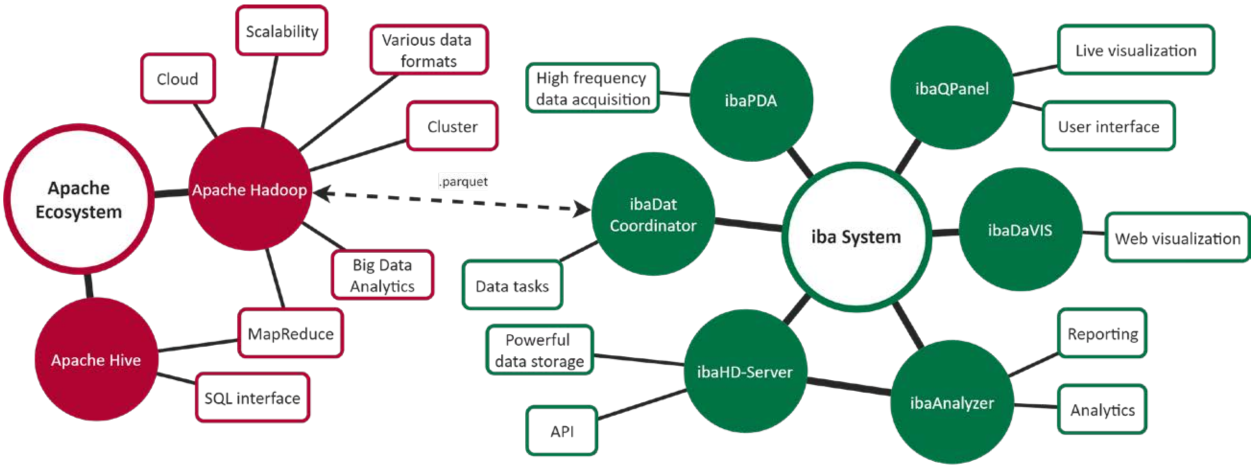

Fig. 3: Application areas of the iba system and Apache Hadoop with Apache Hive

It should be noted that the analysis process with the *iba* software toolchain is very different from that with *Hadoop*. The biggest difference between the *iba* software toolchain and *Hadoop* or even clusters in the cloud lies in the data preparation. *Hadoop* systems and cloud storage generally do not have a user interface where measurement values are displayed graphically. Thus, visual analysis by a user is not readily possible. With a pure *Hadoop* cluster, measurement values can only be processed further via *MapReduce jobs*. These must be designed and programmed in-house. There is an option to use other *Apache* tools, such as *Hive*, to facilitate the filtering and evaluation processes a bit for the user. Services can also be used for automated processing in a *cloud* environment. *Services* involve an implementation effort and must be set up through appropriate configuration.

Outside the *iba* system, the context of the information itself must also be provided, i.e. metadata must be subsequently assigned to the measurement data. Within the *iba* system, metadata is automatically recognized and can be associated with the correct machine or plant as data source.

The advantages of processing with *MapReduce* take effect when *iba* data in the company is only one of several data sources that are to be evaluated together. For example, the *iba* system may be responsible for fast processing of measurement data, while a *MapReduce job* evaluates a copy of the *iba* measurement data together with data from other systems in the company. The result is a so-called *lambda architecture*, as in Fig. 4, page 10, in which both slow and time-critical data can be processed. By developing *MapReduce jobs*, analysts have all the computational capabilities, but also the associated effort, in their own hands.

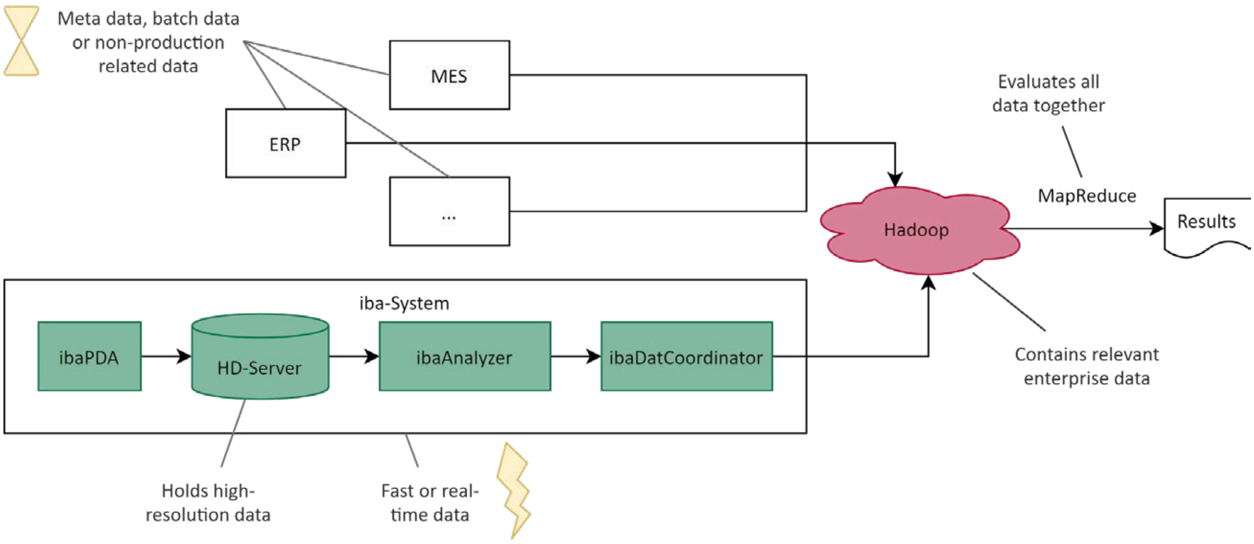

Fig. 4: Lambda architecture with the iba system as data source

2.2 How to set up a Hadoop system?

The three most common ways to use *Apache Hadoop* for data storage and analysis are described below.

1. Set up and run *Hadoop* yourself

   iba recommends this way only with restrictions or only for evaluation or research purposes. The commissioning of a productive system is probably too complex if there is no specific expertise or resources for this in the company. For example, a *Docker* environment may be suitable for getting to know *Hadoop* on a small scale. However, these systems often have low performance compared to a productive cluster.

2. Service provider

   To simplify the setup of a production system or a migration to a *big data* system, there is fee-based professional software based on *Apache* distributions available. *HortonWorks*, *IBM*, *Cloudera* or *Oracle* should be mentioned here, for example. Support services and easier administration are a big advantage with professional providers. In addition, some professional distributions also support *Windows*.

3. Cloud provider

   The third option is to use cloud providers such as *Amazon Web Services* (AWS) or *Microsoft Azure*. In general, setting up clusters in the cloud is very easy and there are a lot of interfaces to other *services*. Another advantage is the flexible scaling. It should be noted that the costs for cloud use also depend heavily on the use case and can be comparatively high.

2.3 How can data be managed in Hadoop?

There are many ways to manage data in *Hadoop*. These depend on whether there is access to an original HDFS or whether *Hadoop* is used within a managed cloud. If an original HDFS is used in the company, files can be fed into the system directly at the *name node* via scripts, for example. In addition, there is also a web service for YARN and HDFS that allows management via the *WebHDFS REST API* [7]. The same applies to the execution of self-developed *MapReduce jobs*. These can also be executed via scripts or by using other tools from the *Apache Ecosystem*. Here, for example, the software *Apache Hive* is suitable, which has an SQL interface. *Hive* automatically converts SQL queries into a *MapReduce job*. In addition, there are a variety of other additional tools and interaction possibilities with *Hadoop*.

Amazon Web Services (AWS) offers the ability to use a managed *Hadoop* system through Amazon EMR. Amazon EMR also enables an exchange to *Amazon Simple Storage Service (Amazon S3)*. As a consequence, the HDFS in AWS can also use the interfaces of S3. [8,9,10] At *Microsoft Azure*, a managed service *HDInsight* exists that can run *Hadoop*, *Spark*, *Kafka*, *HBase*, as well as other *Apache* tools in the *Azure Cloud*. When an *HDInsight* cluster has been created in *Azure*, it is possible to load data into the HDFS via a wide variety of interfaces. [11,12,13]

2.4 How can Hadoop be connected to the iba system?

The connection of *Apache Hadoop* to the *iba* system is possible with the currently available *iba* products. Only a certain procedure for uploading the data must be specified. Further management and analysis of the data is then possible via *Hadoop* and *MapReduce*. The Fig. 5, page 11 and the following description represent a proposed solution.

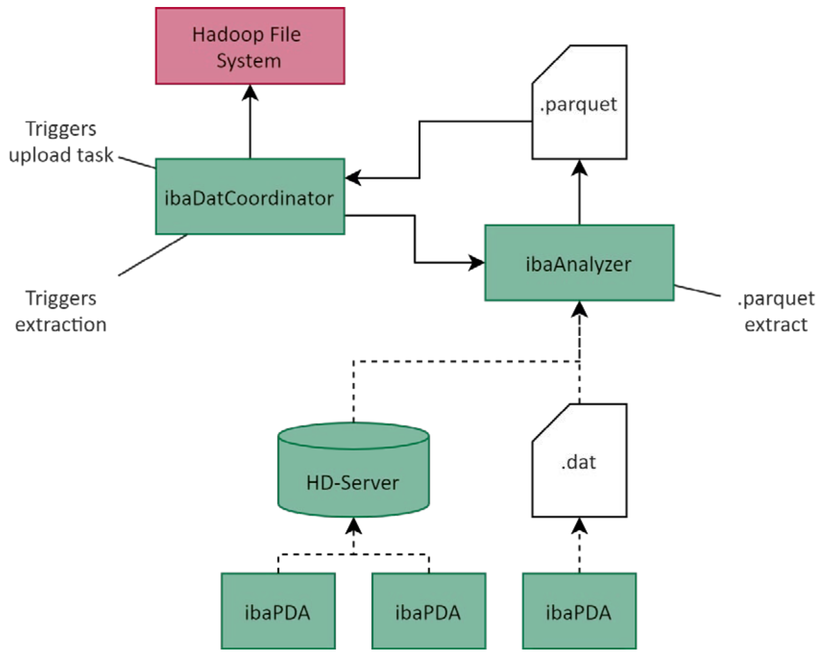

Fig. 5: Proposed solution for a connection of Hadoop to the iba system

The *iba* data, for example *.dat* files or data from the *ibaHD-Server*, have to be extracted into the *Parquet* format in the first step. The extraction is done with *ibaAnalyzer* and is automated by *ibaDatCoordinator* [14]. The *.parquet* files can then be loaded into the HDFS via a script. *AzCopy* is a useful solution [15] for the *Microsoft Azure Cloud*, for example. The export process from *ibaAnalyzer* and the subsequent upload to a *Hadoop* system can be automated with little effort using *ibaDatCoordinator* [16].

As already explained in chapter ↗ *Apache Hadoop*, page 5 it is an advantage to use the *Parquet* format. Due to the structure of the *.dat* format, these files cannot be processed in *Hadoop* without additional effort. *Parquet* is more suitable for parallel processing in HDFS. Nevertheless, it is basically possible to store *.dat* files in HDFS and use HDFS only as a file system.

2.5 Can data from Hadoop be reused in the iba system?

Parquet data can also be read by *ibaAnalyzer* [14]. The *.parquet* files that are located in the HDFS must first be saved back to the *Windows* file system so that they can be found and opened with *ibaAnalyzer.* With *ibaAnalyzer*, the data can also be converted into *.dat* format.

2.6 Application example

Assuming that parts are manufactured in a production facility, whose dimensions are monitored at the end with the *iba* system. Unfortunately, the products occasionally exceed the tolerance limit at the end of the line. It is assumed that certain machines used for forming the semi-finished products are responsible for this. The goal of the investigation is to identify the responsible machines. Measurement data from the *iba* system is regularly converted into *.parquet* files and stored in a *Hadoop* system. The machines regularly generate comprehensive status reports with many messages and current process data in form of *.csv* files, which are also transferred to *Hadoop*.

To investigate the correlation, a *MapReduce job* is developed that evaluates production data from previous months. A *MapReduce job* always consists of a *mapping* process, in which data is read, and a *reducing* process, in which the intermediate results from the *mapping* are merged. Two *mapping* processes are needed in this application example: one for reading the *.csv* files from the machines and one for determining the limit violations in the *.parquet* files coming from the *iba* system. The mapping processes can be executed in parallel on all *nodes* of the *Hadoop* system. After mapping the *.csv* files, for example, certain error messages with the specification of the machine as well as the associated time stamps are available as intermediate results. All the timestamps of a tolerance violation are obtained from the *mapper* of the *.parquet* files, for example. The two intermediate results are merged in an intermediate step. The *Reducing* process now determines whether there is a temporal match between deformation and the error messages and stores these occurrences as a result in a file. The process engineers identify the problematic machines and can examine them more closely to improve the process.

The example makes clear that *MapReduce* can involve a high development effort on the one hand, but on the other hand, highly complex, individual analyses can be performed. The development effort pays off if the analysis is also repeated frequently. Under certain conditions, *MapReduce jobs* can also be generated using other *Apache* tools and comparatively easily. As described in the following section, *MapReduce* is not suitable for every use case. Other systems may be sufficient for simple analyses.

3 When is Hadoop not suitable?

Hadoop is certainly not suitable for every application scenario. The following criteria are indications that *Hadoop* is not suitable for data processing or that the effort is not justified.

Level of digitalization

- There is no digitalization strategy.
- The relevant processes in the company are not yet digitalized.

Data infrastructure

- There are no dedicated data servers.
- Existing databases have gaps or are not consistent.

Data analytics

- There is no analysis strategy.
- There is no expertise in professional data analysis.
- There are no resources for the development of automatic analysis.

Many innovations and future-oriented technologies, including *Hadoop* as a *big data* application, have in common that they trigger trends and exaggerated expectations in a short period of time. These put themselves into perspective after a while and continue to develop until the technology reaches the production environment. [3,18] The scope of a technology also becomes sharper during these phases. *Apache Hadoop* is also not a universal solution for every use case where data is stored.

The favorable and less favorable file formats for the HDFS have already been mentioned in the introduction to *Hadoop* and the explanations of the *Parquet* format in chapter ➚ *Apache Hadoop*, page 5 In principle, all formats can be stored in HDFS, but this approach offers few or no advantages. A structured evaluation of data using a *MapReduce job* becomes increasingly complex when the files involved require preprocessing. Documents that are created and used manually by employees should not be stored in HDFS, but should still be stored in a file system that is easy to access by employees.

Relational databases in the enterprise also cannot be replaced by a *Hadoop* system in every case. A powerful database can return a result in a very short time by means of indexing. *Apache Hadoop*, however, stores data in files and does not index them. To perform a query, a *MapReduce job* must be run to search all the data. The *MapReduce* process will take a longer time in many cases. In general, *Hadoop* can and should not replace a relational database until its technical limits are reached. This is the case, for example, if the data volumes are so high that indexing becomes too time consuming. [19]

In addition to the technical aspects, there is also the economic perspective on *Apache Hadoop*, which is often difficult to quantify or estimate. The open source program library of *Apache Hadoop* can be used by companies without licensing costs [20]. Therefore, it might be assumed that *Hadoop* is more cost-effective than a professional, specialized database system. However, some circumstances must be taken into account. *Apache Hadoop* is a comprehensive software based on *Java* and developed for *Linux*. Planning, setting up, and administration of *Hadoop* as a production system requires advanced *Linux* skills and network configuration expertise. [19] In addition, productive use requires a configured cluster, which is usually very complex to configure. The steps mentioned must be taken into account when introducing *Hadoop*. *iba* recommends using the *Hadoop* distribution of a professional provider or having a service provider carry out the start-up if no dedicated IT resources are available in the company.

4 Lessons learned

The decision to use a higher-level *Hadoop* system must be well thought out. Setting up, configuring, and integrating a *Hadoop* cluster into the existing data infrastructure involves a corresponding effort and requires a clear strategy about what insights are to be gained from the *MapReduce jobs*. In any case, the requirements in the company for the data infrastructure should be compared with the properties of *Hadoop* beforehand. If the requirements do not align with these, another data storage solution may be more suitable.

If the introduction of a new *Hadoop* system is planned, *iba* recommends a professional distribution or the use of *Hadoop* in the context of a cloud. If both a *Hadoop* system and an *iba* system already exist in the company, the data can be extracted and moved to the HDFS using *ibaDatCoordinator.*

5 Glossary

Apache Hadoop	Software for a distributed file system on a *cluster* that can also be used to run *MapReduce jobs*.
Apache Hive	Extension to *Apache Hadoop* to include data query functionalities via SQL, for example.
Cluster	Networked association of computers.
.dat	File format of *iba* for measurement data recording, which is used in the iba system.
HDFS	*Hadoop Distributed File System*: File system of *Apache Hadoop* that stores and manages files within a cluster.
MapReduce	Algorithm for parallel data processing of large data sets on *clusters*.
Node	Single computer within a *cluster*.
.parquet	File format designed for use with *Apache* software tools.
SQL	*Structured Query Language*: Database language for creating and editing database structures.
YARN	*Yet another resource negotiator*: Resource manager in *Apache Hadoop*.

6 Sources and references

[1] NewTech4Steel (2021). *About*. NewTech4Steel Website.
URL: http://newtech4steel.eu/node/10 (last accessed 03.09.2021)

[2] Mielke, Fabian (2021). *ibaHD-Server Benchmark. Comparison between ibaHD-Server and time-based database systems.* White Paper. Issue 1.0. Iba AG (publisher) Fuerth, Germany.
URL: https://www.iba-ag.com/en/news/ibahd-server-benchmark (last accessed 07.10.2021)

[3] Fasel, Daniel (2014). *Big Data – Eine Einführung. HMD Praxis der Wirtschaftsinformatik.* Vol. 51, p. 386-400.

[4] Apache Software Foundation (2021). Apache Hadoop Homepage.
URL: https://hadoop.apache.org/ (last accessed 16.08.2021).

[5] iba AG (2021). *Areas of Application*. iba AG Website.
URL: https://www.iba-ag.com/en/areas-of-application (last accessed 16.08.2021).

[6] Apache Software Foundation (2020). *Powered by Apache Hadoop*. Apache Wiki.
URL:https://cwiki.apache.org/confluence/display/HADOOP2/PoweredBy (last accessed 16.08.2021).

[7] Apache Software Foundation (2020). *WebHDFS REST API*. Apache Hadoop Documentation.
URL: http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/WebHDFS.html (last accessed 16.08.2021)

[8] Amazon Web Services (2021). *What Is Amazon EMR?.* AWS Documentation.
URL: https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-what-is-emr.html (last accessed 16.08.2021)

[9] Amazon Web Services (2021). *Upload Data to Amazon S3*. AWS Documentation.
URL: https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-upload-s3.html (last accessed 16.08.2021)

[10] Amazon Web Services (2021). *What is Hadoop?.* Amazon EMR Details.
URL: https://aws.amazon.com/emr/details/hadoop/what-is-hadoop/ (last accessed 16.08.2021)

[11] Microsoft (2021). *Azure HDInsight documentation*. Microsoft Docs.
URL: https://docs.microsoft.com/en-us/azure/hdinsight/ (last accessed 16.08.2021)

[12] Microsoft (2020). *Quickstart: Create Apache Hadoop cluster in Azure HDInsight using Azure portal*. Microsoft Docs.
URL: https://docs.microsoft.com/en-us/azure/hdinsight/hadoop/apache-hadoop-linux-create-cluster-get-started-portal (last accessed 16.08.2021)

[13] Microsoft (2020). *Upload data for Apache Hadoop jobs in HDInsight.* Microsoft Docs.
URL: https://docs.microsoft.com/en-us/azure/hdinsight/hdinsight-upload-data#upload-data-to-azure-storage (last accessed 16.08.2021)

[14] iba AG (2021). ibaAnalyzer. Introduction and installation. Manual part 1. Issue 7.2, p. 12-13.

[15] Microsoft (2021). *Get started with AzCopy*. Microsoft Docs.
URL: https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10 (last accessed 16.08.2021)

[16] iba AG (2021). *Extraction task* and *Script task ibaDatCoordinator* manual Issue 2.3, p. 51-53.

[17] Apache Software Foundation and its Contributors (2021). *parquet-format*. GitHub Repository.
URL: https://github.com/apache/parquet-format (last accessed 16.08.2021)

[18] Gartner (2013). *Gartners Hype Cycle 2013*. In: Fasel, Daniel (2014). *Big Data – Eine Einführung. HMD Praxis der Wirtschaftsinformatik.*

[19] Apache Software Foundation (2019). *HadoopIsNot*. Apache Wiki.
URL: https://cwiki.apache.org/confluence/display/HADOOP2/HadoopIsNot (last accessed 16.08.2021)

[20] Apache Software Foundation (2004). *Apache License*. Version 2.0.
URL: https://www.apache.org/licenses/LICENSE-2.0.txt (last accessed 16.08.2021)

[21] Apache Software Foundation (2018). Apache Parquet Documentation.
URL: https://parquet.apache.org/documentation/latest/ (last accessed 02.09.2021)

7 Support and contact

Support

Phone: +49 911 97282-14

Fax: +49 911 97282-33

Email: support@iba-ag.com

Note

If you need support for software products, please state the license number or the CodeMeter container number (WIBU dongle). For hardware products, please have the serial number of the device ready.

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone: +49 911 97282-0

Fax: +49 911 97282-33

Email: iba@iba-ag.com

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

For contact data of your regional iba office or representative please refer to our web site

www.iba-ag.com.