Integration von Apache Hadoop

Möglicher Einsatz von Apache Hadoop zur Langzeitspeicherung von iba-Daten

White Paper
Ausgabe 1.0

Messsysteme für Industrie und Energie
www.iba-ag.com

Herausgeber

iba AG

Königswarterstr. 44

90762 Fürth

Deutschland

Kontakte

Zentrale	+49 911 97282-0
Telefax	+49 911 97282-33
Support	+49 911 97282-14
Technik	+49 911 97282-13
E-Mail	iba@iba-ag.com
Web	www.iba-ag.com

Autoren

Fabian Mielke, Software Developer

Jonas Silvennoinen, Software Developer

Ausgabe	Datum	Autor	Änderungen
1.0	10-2021	FM	Erstausgabe

Inhalt

1 Einleitung

In der industriellen Produktion fällt eine große Menge unterschiedlicher technischer Betriebsdaten an. Zu dieser Gruppe gehören auch Prozessdaten, Messdaten und Metadaten. Die Prozessdaten einer Anlage werden in der Regel zyklisch erfasst, während Messdaten sowohl zu einem Prozess als auch zu einem Produkt oder einer Charge gehören können. Hinzu kommen Metadaten, wie zum Beispiel Produktnummern, Kundennummern oder andere zusätzliche Informationen. Mit dem *iba*-System können technische Betriebsdaten aufgezeichnet, gespeichert, analysiert und visualisiert werden. Im Allgemeinen ist eine detaillierte Datenaufzeichnung notwendig zur Qualitätsdokumentation. Sie ist zudem nützlich, um Potenzial bei der Prozess- und Qualitätsverbesserung zu erkennen.

Die generierte Datenmenge eines Unternehmens nimmt mit dessen Größe und dem Digitalisierungsgrad schnell zu. Der Aufwand und die Komplexität der Datenverwaltung steigen ebenfalls. Daher wird in diesem White Paper der Fragestellung nachgegangen, wie Anwender von *iba*-Software mit solchen Anforderungen umgehen können und welche Lösungen es gibt, sehr große Datenmengen zu verwalten. Hierzu wurden Untersuchungen im Rahmen des Forschungsprojekts *NewTech4Steel [1]* bei *iba* durchgeführt. Dieses White Paper beinhaltet grundlegende Information zu *Apache Hadoop*, die Ergebnisse des Projekts sowie Empfehlungen für *iba*-Anwender, die ein enorm großes Datenaufkommen haben oder planen, *Apache Hadoop* zur Speicherung und Analyse der Messdaten zu verwenden. Es knüpft außerdem an das vorangegangene White Paper zum Thema *ibaHD-Server Benchmark* an. Die Ergebnisse des *ibaHD-Server Benchmark* haben gezeigt, dass *ibaHD-Server* aufgrund seines effizienten Speicheralgorithmus sehr gut für die hochfrequente Datenaufzeichnung geeignet ist [2]. In diesem Dokument liegt der Fokus nun auf der Skalierbarkeit und Lösungen für große Datenmengen.

Die nachfolgenden Auswertungen und Untersuchungen wurden im Rahmen des Forschungsprojekts *NewTech4Steel [1]* bei *iba* durchgeführt. Es wurde untersucht, ob und in welchen Anwendungsfällen es vorteilhaft sein kann, zusätzlich zum *iba*-System *Apache Hadoop* zu verwenden. Das Ziel des europäischen Forschungsprojekts *NewTech4Steel* ist es, die Prozesse in der Stahlproduktion durch disruptive, datengetriebene Technologien hinsichtlich ihrer Stabilität und der Produktqualität zu verbessern und den Nutzen der gesamten europäischen Stahlindustrie zu vermitteln. Eine der zu untersuchenden Technologien ist die Datensammlung und Auswertung mit dem *Big Data* Konzept. *NewTech4Steel* ist Teil des *Research for Coal and Steel Programme (RFCS-2017)* und hat eine Laufzeit von 42 Monaten. Der Koordinator ist die *VDEh-Betriebsforschungsinstitut GmbH* (BFI). Die *iba AG* ist Teil des Projektkonsortiums und trägt mit dem Knowhow zur Datenerfassung und dem *iba*-System zu *NewTech4Steel* bei. Bei *iba* wurde beispielsweise ein dediziertes *Computer-Cluster* aufgebaut, das allen Projektpartnern die Möglichkeit gibt, *Big Data* Anwendungen zu implementieren. Dieses *Cluster*-System wurde auch für die Untersuchungen von *Apache Hadoop* verwendet. [1]

1.1 Big Data und Skalierbarkeit

Wenn die Datenmenge im Unternehmen ein besonders großes Volumen erreicht, schnell zunimmt und vielfältige Datenformate beinhaltet, spricht man von einem *Big Data* Szenario. Es gibt keine eindeutige Definition, aber die gemeinsamen Kriterien für *Big Data* sind häufig die drei Eigenschaften *Volume*, *Variety* und *Velocity*. Eine Analyse von Daten mit diesen Eigenschaf-

ten in einem konventionellen Datei- oder relationalen Datenbanksystem ist komplex bis unmöglich. Für die Verarbeitung von Daten in dieser Größenordnung eignet sich unter Umständen keine konventionelle Hardware mehr. Wenn das Datenvolumen eine Größenordnung von etwa 100 TB erreicht, stoßen spezialisierte und optimierte relationale Datenbanksysteme architektonisch und technisch an ihre Grenzen. Denn mit zunehmendem Datenvolumen erhöht sich der Aufwand, die Daten operativ verfügbar und konsistent zu halten. Relationale Datenbanken dieser Größenordnung sind anwenderspezifisch angepasst und benötigen kostenintensive Hardware. Wenn die bestehende Leistung nicht mehr ausreicht, muss die Hardware angepasst werden. Diese Anpassungen durch Aufrüstung bezeichnet man als *vertikale Skalierung.* [3]

1.2 Apache Hadoop

Apache Hadoop ist eine Software, die für die Speicherung und Analyse enorm großer Datenmengen entwickelt wurde. Die Software bietet mit dem *Hadoop Distributed File System* (HDFS) ein eigenes Dateisystem sowie eine eigene Ressourcenverwaltung (YARN).

Mit *Hadoop* können Daten auf einem *Computer-Cluster* verteilt und parallel verarbeitet werden. Das Cluster besteht aus mindestens einem Knoten (engl. *Node*) und kann beliebig skaliert werden, indem zum Beispiel die Anzahl der *Nodes* erhöht oder reduziert wird. Jeder weitere *Node*, der dem Cluster hinzugefügt wird, erhöht die Rechenleistung und den Speicherplatz. Das Konzept der Verwendung von Knoten bietet außerdem Vorteile bei der Verteilung von Rechenleistung und kann Ausfälle kompensieren. [4] Die parallele Verarbeitung von Daten auf mehreren Cluster-*Nodes* macht *Hadoop* enorm leistungsfähig und sie wird mit jedem weiteren *Node* effizienter.

Hadoop ist *horizontal skalierbar*, d. h. es ist im laufenden Betrieb möglich, das Cluster zu erweitern. Dabei können konventionelle Computer als neue *Nodes* eingesetzt werden. Eine Aufrüstung ist nicht zwingend notwendig. Bei Bedarf kann die Zahl der *Nodes* auch reduziert werden. Die Leistung und die Kapazität des Gesamtsystems können mit vergleichsweise wenig Aufwand angepasst werden. [3]

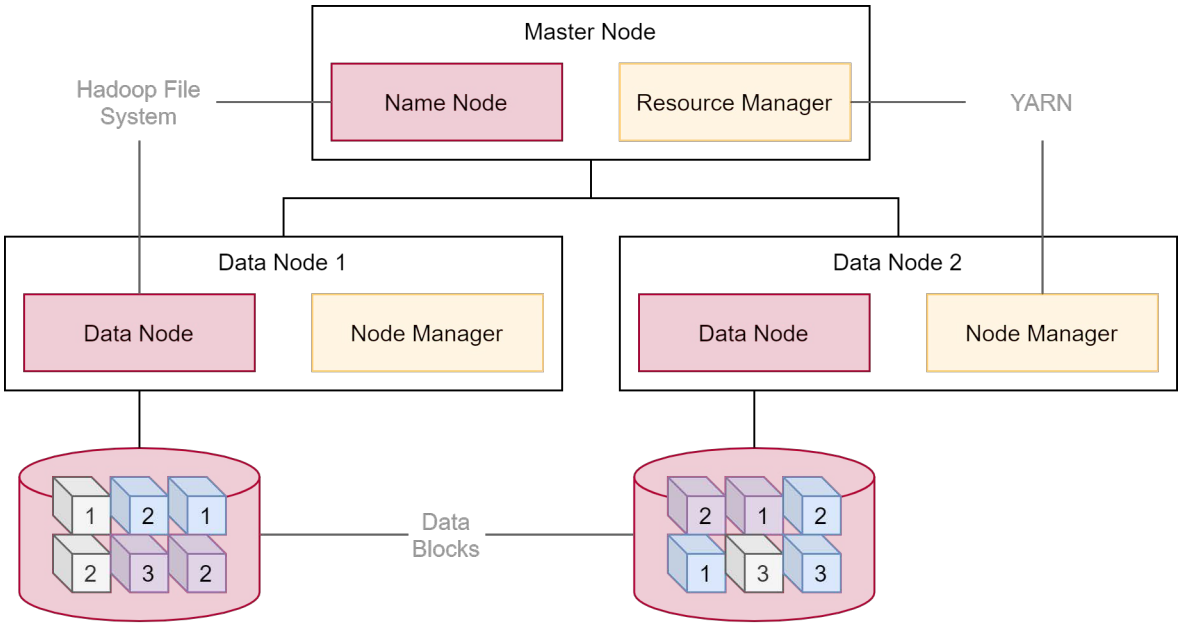

Abb. 1: Vereinfachter Aufbau eines Apache Hadoop Clusters

Um Daten parallel verarbeiten zu können, werden Dateien im HDFS in Datenblöcke (engl. *Data Blocks*) zerteilt und repliziert, wie im unteren Teil der Abb. 1, Seite 6 dargestellt. *Replikation* bedeutet, dass jeder Block vervielfältigt und auf weiteren *Nodes* verteilt wird, um eine Redundanz herzustellen. Grundsätzlich unterstützt das HDFS alle Dateitypen, es gibt jedoch eine Reihe besonders geeigneter Formate. Der Unterschied ist, dass diese ohne Verlust von Informationen teilbar sind und die einzelnen Blöcke gleichzeitig von den *Data Nodes* verarbeitet werden können. Ein Beispiel für ein günstiges Format ist zum Beispiel *.csv*, da es zeilenbasiert ist und nach jeder Datenreihe problemlos getrennt werden kann.

Es gibt auch binäre Formate, die für das HDFS und *MapReduce* besonders geeignet sind. Bekannte Vertreter sind *.avro*- und *.parquet*-Dateien. *Apache* hat das *Parquet*-Format eigens für die schnelle, parallele Verarbeitung mit *Hadoop* entwickelt. *Parquet* ist ein spaltenorientiertes Format, das zudem sehr effizient komprimiert und verarbeitet werden kann. [17,21] Mit dem *iba*-System können alle erfassten Daten in das *Parquet*-Format extrahiert werden. Ungünstige Dateiformate für HDFS sind diejenigen, deren Teile bzw. Blöcke alleine nicht sinnvoll auswertbar sind. In diesem Fall müssen Dateien vor der Verarbeitung mithilfe von *SequenceFile* zusammengefügt werden.

Für einige Anwendungsszenarien kann der Einsatz von *Hadoop* vorteilhaft sein, wenn *Hadoop* als übergeordnetes System und zusätzlich zum *iba*-System verwendet wird. In dieser Kombination übernehmen *ibaPDA* und *ibaHD-Server* die Aufzeichnung und Speicherung von Messdaten. Die *iba*-Daten können anschließend nach *Hadoop* extrahiert werden, um dort eine gemeinsame Analyse mit anderen Unternehmensdaten vornehmen zu können. Mehr Informationen dazu befinden sich in Kapitel ↗ *Wie kann Hadoop an das iba-System angebunden werden?*, Seite 11.

An dieser Stelle sei auch darauf hingewiesen, dass es nicht in jedem Fall sinnvoll oder mit einem Leistungszuwachs verbunden ist, *Hadoop* für die Auswertung von Unternehmensdaten einzuführen. Die Gründe hierfür werden in einem eigenen Abschnitt in Kapitel ↗ *Wann eignet sich Hadoop nicht?*, Seite 14 aufgeführt.

2 Für welche Anwendungsfälle eignet sich Hadoop?

Im Folgenden werden die typischen Anwendungsgebiete von *iba*-Produkten mit denen von *Apache Hadoop* verglichen und eingeordnet. Anschließend wird näher auf die Anwendungsfälle eingegangen, in denen der Einsatz von *Apache Hadoop* als Ergänzung zum *iba*-System sinnvoll sein kann.

Die Domäne der *iba AG* ist im Allgemeinen die lückenlose und hochfrequente Aufzeichnung von Prozess- und Messdaten. Das *iba*-System zur Messdatenerfassung und -analyse besteht aus aufeinander abgestimmten Hard- und Softwarekomponenten, mit denen sowohl die Aufzeichnung, als auch die Auswertung und Weiterverarbeitung von Messdaten vorgenommen werden kann. Durch die Konfigurierbarkeit und die modulare Architektur kann das *iba*-System an vielfältige Aufgabenstellungen angepasst werden. Die aufgezeichneten Daten können mithilfe des *iba*-Produktportfolios anschließend zur Qualitätsdokumentation sowie zur Prozessanalyse und Fehlersuche verwendet werden. [5]

Daten, die mit dem *iba*-System aufgezeichnet werden, können hoch aufgelöst sein und hängen mit einer Anlage oder einem Produkt zusammen. Mit den verschiedenen *iba*-Softwareprodukten können Kennwerte und abgeleitete Daten kalkuliert sowie in eine Vielzahl externer Systeme extrahiert werden. Das *.dat*-Dateiformat und *ibaHD-Server* bieten innerhalb des *iba*-Systems eine konsistente Speichermöglichkeit. Umgekehrt können die Messergebnisse direkt einem Prozess oder einem Produkt zugeordnet und unmittelbar wieder abgefragt werden.

Die Domäne von *Hadoop* ist das Speichern von großen Datenmengen und die Auswertung mithilfe von *MapReduce Jobs*. Die *MapReduce Jobs* sind Datenverarbeitungsaufgaben, die beliebig komplex werden und viel Rechenzeit in Anspruch nehmen können, da sie über alle Datensätze im HDFS ausgeführt werden. Diese Rechenzeit ist kostenintensiv, besonders wenn sich *Hadoop* in einer verwalteten Cloud befindet. *Hadoop* ist nicht für zeitkritische Meldungen oder kurzfristige Entscheidungen im industriellen, produktiven Umfeld gedacht, sondern vielmehr für mittelfristige Analysen und Statistiken. Es steht nicht die exakte Dokumentation, sondern das Zusammenführen von Informationen im Vordergrund. *Hadoop* ist gewissermaßen auf Datenquellen angewiesen und bietet die Möglichkeit, Daten bei Bedarf mit anderen beliebigen Datenquellen in Verbindung zu bringen. Einige Praxisbeispiele befinden sich auf der *Apache* Website. Beispielsweise nutzt *eBay Hadoop*, um die Suche nach Produkten zu optimieren, während *Facebook* Nutzerdaten und Log-Dateien analysiert [6].

Die konkreten Vorteile des HDFS gegenüber einem lokalen oder Netzwerk-Dateisystem sind besonders bei großen Datenmengen ausgeprägt und werden im Folgenden genannt.

- Toleranz gegenüber Ausfällen des Systems
- Paralleler Zugriff auf Dateien durch die Replikation
- Höhere Performance beim Zugriff auf Dateien

Die Abb. 2, Seite 8 stellt die jeweiligen Haupteigenschaften und Stärken der beiden Systeme gegenüber. Das *iba*-System bietet als ein Portfolio umfangreicher Softwarelösungen eine Vielzahl von Analyse- und Reporting-Tools, die der Endnutzer nach der Installation direkt verwenden und bedienen kann. Mit den *.dat*-Dateien und dem *ibaHD-Server* hat der Anwender außerdem die Möglichkeit, die Datenauswertung und das Reporting nach seinen Anforderungen umzusetzen. Die Stärke von *Hadoop* liegt vor allem in der Flexibilität und Skalierbarkeit.

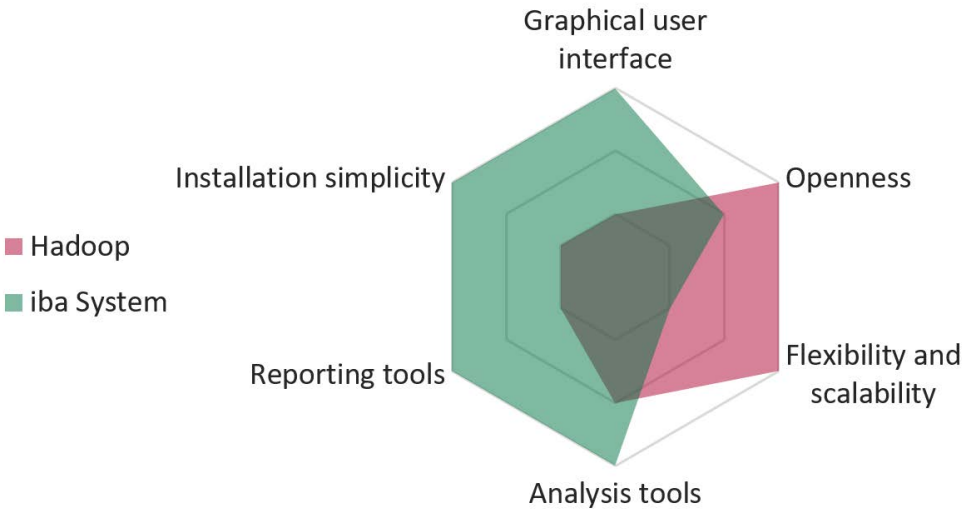

Abb. 2: Domänen des iba-Systems und von Apache Hadoop im Vergleich

2.1 Unter welchen Voraussetzungen bringt die Kombination von iba und Hadoop Vorteile?

In bestimmten Anwendungsfällen kann es von Vorteil sein, Daten aus einem *iba*-System nach *Hadoop* zu extrahieren und somit beide Systeme zu kombinieren. In diesen Use Cases sollte die Mehrheit der nachfolgenden Voraussetzungen erfüllt sein. Die Kriterien beziehen sich auf alle Unternehmensdaten, z. B. MES-, Log- und *iba*-Daten, die miteinander ausgewertet werden sollen.

- Es gibt genug Fachwissen über Hadoop oder einen entsprechenden Cloud-Service
- Die Datenmengen befinden sich im zweistelligen Terabyte-Bereich
- Die Unternehmensdaten sind sehr heterogen
- Das Unternehmen verfolgt eine Analysestrategie mit Ziel-Kennwerten
- Produktionsdaten werden mit dem *iba*-System lediglich aufgezeichnet und nicht analysiert
- Daten sollen zentral in einer Cloud oder in einem Cluster abgelegt und dort verarbeitet werden

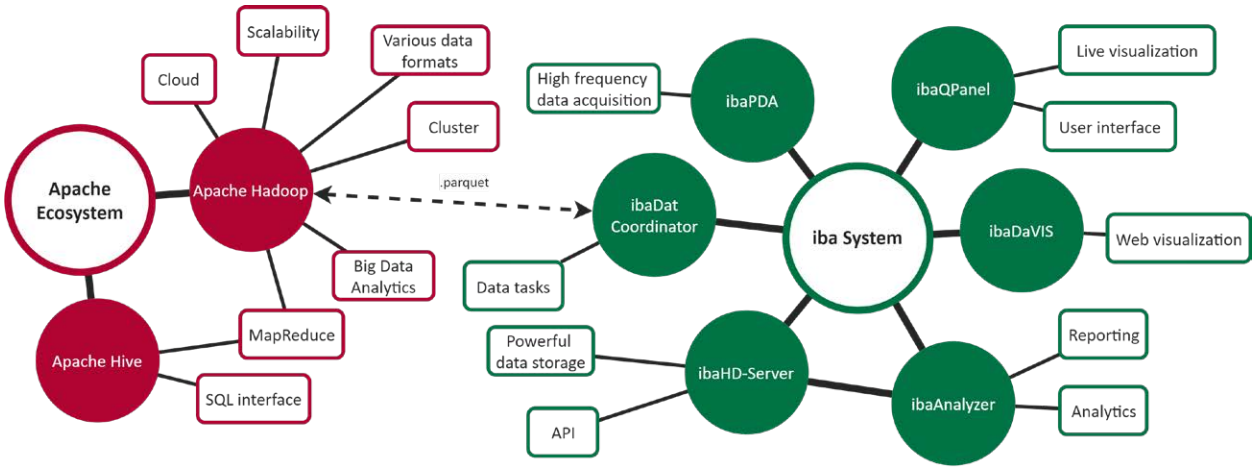

Abb. 3: Anwendungsbereiche des iba-Systems und von Apache Hadoop mit Apache Hive

Es muss beachtet werden, dass sich der Analyseprozess mit der *iba*-Software-Toolchain stark von dem mit *Hadoop* unterscheidet. Der größte Unterschied zwischen der *iba*-Software-Toolchain zu *Hadoop* oder auch zu Clustern in der Cloud liegt in der Datenaufbereitung. *Hadoop*-Systeme und Cloudspeicher haben in der Regel keine Benutzeroberfläche, in denen sich Messwerte grafisch darstellen lassen. Damit sind visuelle Analysen durch einen Benutzer nicht ohne Weiteres möglich. Mit einem reinen *Hadoop*-Cluster können Messwerte lediglich über *MapReduce* Jobs weiterverarbeitet werden. Diese müssen selbst konzipiert und programmiert werden. Es gibt die Möglichkeit andere *Apache* Tools, wie zum Beispiel *Hive,* zu verwenden, um die Filter- und Evaluationsprozesse für den Benutzer etwas zu vereinfachen. In einer *Cloud*-Umgebung können auch Dienste (engl. *Services*) für die automatische Verarbeitung benutzt werden. *Services* bringen einen Implementierungsaufwand mit sich und müssen durch eine entsprechende Konfiguration eingerichtet werden.

Außerhalb des *iba*-Systems muss zudem der Kontext der Informationen selbst hergestellt werden, d. h. Metadaten müssen den Messdaten nachträglich zugeordnet werden. Innerhalb des *iba*-Systems werden Metadaten automatisch erkannt und können mit der richtigen Maschine oder Anlage als Datenquelle in Verbindung gebracht werden.

Die Vorteile der Verarbeitung mit *MapReduce* kommen zum Tragen, wenn *iba*-Daten im Unternehmen lediglich eine von mehreren Datenquellen sind, die gemeinsam ausgewertet werden sollen. Das *iba*-System kann beispielsweise für die schnelle Verarbeitung von Messdaten zuständig sein, während ein *MapReduce Job* eine Kopie der *iba*-Messdaten gemeinsam mit Daten aus anderen Systemen des Unternehmens auswertet. Es entsteht eine so genannte *Lambda-Architektur*,wie in Abb. 4, Seite 10, in der sowohl langsame als auch zeitkritische Daten verarbeitet werden können. Durch die Entwicklung von *MapReduce Jobs* haben Analysten alle Möglichkeiten zur Berechnung, aber auch den damit verbundenen Aufwand, selbst in der Hand.

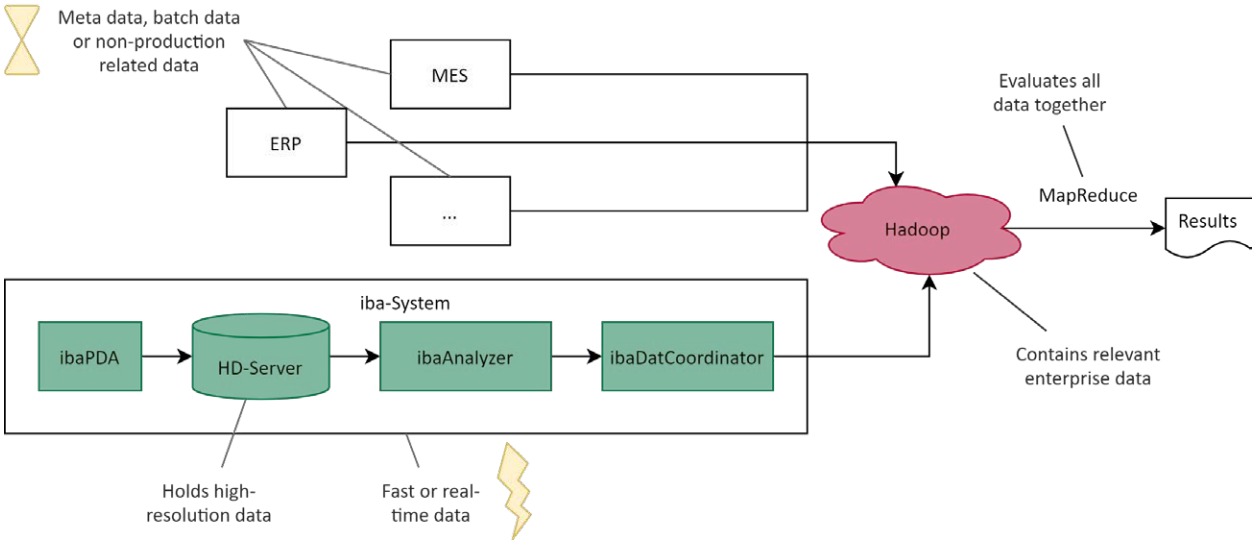

Abb. 4: Lambda-Architektur mit dem iba-System als Datenquelle

2.2 Wie setzt man ein Hadoop-System auf?

Die drei gängigsten Wege *Apache Hadoop* zur Datenspeicherung und -analyse zu verwenden werden im Folgenden beschrieben.

1. *Hadoop* selbst aufsetzen und in Betrieb nehmen

   Diesen Weg empfiehlt *iba* nur eingeschränkt bzw. lediglich für Evaluations- oder Forschungszwecke. Die Inbetriebnahme eines Produktivsystems ist vermutlich zu komplex, wenn im Unternehmen kein einschlägiges Fachwissen oder Ressourcen dafür vorhanden sind. Für das Kennenlernen von *Hadoop* im kleinen Umfang kann sich zum Beispiel eine *Docker*-Umgebung eignen. Allerdings haben diese Systeme im Vergleich zu einem produktiven Cluster oft nur eine geringe Leistungsfähigkeit.

2. Serviceanbieter

   Um das Aufsetzen eines Produktivsystems oder eine Migration zu einem *Big Data* System zu vereinfachen gibt es professionelle, kostenpflichtige Software, die auf *Apache* Distributionen basieren. Hier sind beispielsweise *HortonWorks*, *IBM*, *Cloudera* oder *Oracle* zu nennen. Ein großer Vorteil bei professionellen Anbietern sind die Supportdienstleistungen und die erleichterte Administration. Zudem unterstützen einige professionelle Distributionen auch *Windows*.

3. Cloud-Provider

   Die dritte Möglichkeit ist das Nutzen von Cloud-Providern wie beispielsweise *Amazon Web Services* (AWS) oder *Microsoft Azure*. Im Allgemeinen ist das Aufsetzen von Clustern in der Cloud sehr einfach und es gibt eine Vielzahl von Schnittstellen zu anderen *Services*. Ein weiterer Vorteil ist die flexible Skalierung. Es gilt zu beachten, dass die Kosten für die Cloud-Nutzung ebenfalls stark vom Anwendungsfall abhängen und vergleichsweise hoch sein können.

2.3 Wie können Daten in Hadoop verwaltet werden?

Es gibt viele Möglichkeiten Daten in *Hadoop* zu verwalten. Diese hängen davon ab, ob man Zugriff zu einem originären HDFS hat oder ob *Hadoop* innerhalb einer verwalteten Cloud benutzt wird. Wenn ein originäres HDFS im Unternehmen genutzt wird, können Dateien zum Beispiel über Skripte direkt am *Name Node* in das System eingespeist werden. Zusätzlich gibt es für YARN and HDFS auch noch einen Webservice, der eine Verwaltung über die *WebHDFS REST API* [7] zulässt. Ähnliches gilt für die Ausführung von selbst entwickelten *MapReduce Jobs*. Diese können ebenfalls über Skripte oder mithilfe von anderen Tools aus dem *Apache Ecosystem* ausgeführt werden. Hier bietet sich zum Beispiel die Software *Apache Hive* an, die über ein SQL-Interface verfügt. *Hive* wandelt SQL-Abfragen automatisch in einen *MapReduce Job* um. Darüber hinaus existiert eine Vielzahl weiterer Zusatztools und Interaktionsmöglichkeiten mit *Hadoop*.

Amazon Web Services (AWS) bietet die Möglichkeit, über Amazon EMR ein verwaltetes *Hadoop*-System zu nutzen. Amazon EMR ermöglicht außerdem einen Austausch zum *Amazon Simple Storage Service (Amazon S3)*. Folglich können für das HDFS in AWS auch die Schnittstellen von S3 verwendet werden. [8,9,10] Bei *Microsoft Azure* existiert ein verwalteter Dienst *HDInsight*, der *Hadoop*, *Spark*, *Kafka*, *HBase* sowie andere Apache Tools in der *Azure* Cloud ausführen kann. Wenn ein *HDInsight*-Cluster in *Azure* angelegt wurde, kann man über verschiedenste Schnittstellen Daten in das HDFS laden. [11,12,13]

2.4 Wie kann Hadoop an das iba-System angebunden werden?

Die Anbindung von *Apache Hadoop* an das *iba*-System ist mit den aktuell verfügbaren *iba*-Produkten möglich. Es muss lediglich eine bestimmte Vorgehensweise für den Upload der Daten festgelegt werden. Die weitere Verwaltung und Analyse der Daten ist anschließend über *Hadoop* und *MapReduce* möglich. Die Abb. 5, Seite 12 und die nachfolgende Beschreibung stellen einen Lösungsvorschlag dar.

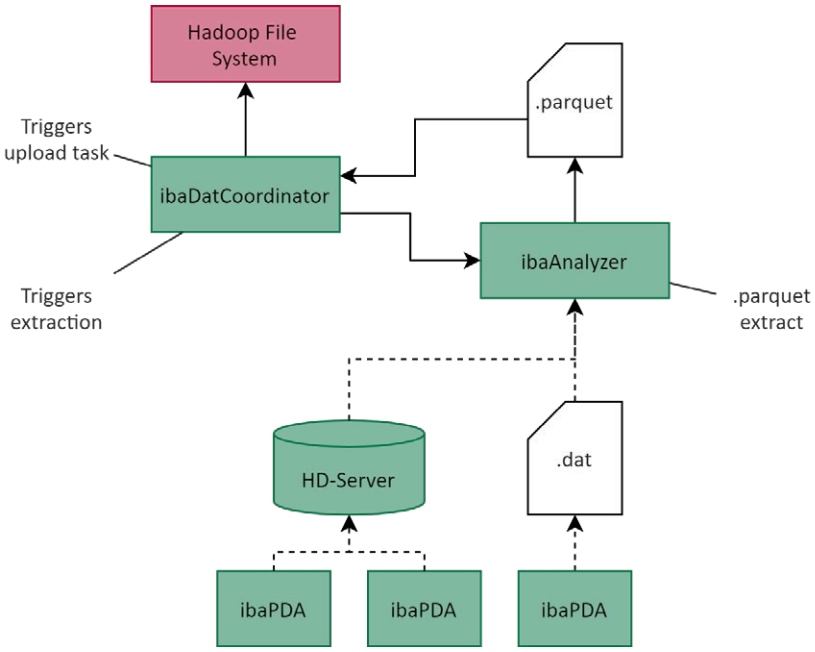

Abb. 5: Lösungsvorschlag für eine Anbindung von Hadoop an das iba-System

Im ersten Schritt müssen die *iba*-Daten, zum Beispiel *.dat*-Dateien oder Daten aus dem *ibaHD-Server*, in das *Parquet*-Format extrahiert werden. Die Extraktion erfolgt mit *ibaAnalyzer* und wird von *ibaDatCoordinator* automatisiert [14]. Anschließend können die *.parquet*-Dateien über ein Skript in das HDFS geladen werden. Für die *Microsoft Azure Cloud* bietet sich zum Beispiel *AzCopy* an [15]. Der Exportprozess aus *ibaAnalyzer* und der anschließende Upload in ein *Hadoop*-System lässt sich mit geringem Aufwand über *ibaDatCoordinator* automatisieren [16].

Wie bereits in Kapitel ↗ *Apache Hadoop*, Seite 5 erklärt, ist es von Vorteil, das *Parquet*-Format zu verwenden. Aufgrund des Aufbaus des *.dat*-Formats können diese Dateien nicht ohne zusätzlichen Aufwand in *Hadoop* verarbeitet werden. Für die parallele Verarbeitung im HDFS ist *Parquet* besser geeignet. Dennoch ist es grundsätzlich möglich *.dat*-Dateien im HDFS abzulegen und HDFS lediglich als Dateisystem zu verwenden.

2.5 Können Daten aus Hadoop wieder im iba-System verwendet werden?

Parquet-Daten können mit *ibaAnalyzer* auch wieder eingelesen werden [14]. Die *.parquet*-Dateien, die sich im HDFS befinden, müssen zuerst wieder im *Windows*-Dateisystem gespeichert werden, damit man sie mit *ibaAnalyzer* finden und öffnen kann. Mit *ibaAnalyzer* können die Daten auch in das *.dat*-Format konvertiert werden.

2.6 Anwendungsbeispiel

Angenommen, in einer Produktionsstätte werden Teile gefertigt, deren Maße am Ende mit dem *iba*-System überwacht werden. Unglücklicherweise überschreiten die Produkte am Ende der Anlage gelegentlich die Toleranzgrenze. Es wird vermutet, dass bestimmte Maschinen zur Umformung der Halbfabrikate dafür verantwortlich sind. Das Ziel der Untersuchung ist es, die verantwortlichen Maschinen zu identifizieren. Die Messdaten aus dem *iba*-System werden regelmäßig in *.parquet*-Files konvertiert und in einem *Hadoop*-System abgelegt. Die Maschinen erzeugen regelmäßig umfangreiche Statusberichte mit vielen Meldungen und aktuellen Prozessdaten in Form von *.csv*-Dateien, die ebenfalls in *Hadoop* transferiert werden.

Um den Zusammenhang zu untersuchen, wird ein *MapReduce Job* entwickelt, der die Produktionsdaten der vergangenen Monate auswertet. Ein *MapReduce Job* besteht immer aus einem *Mapping*-Prozess, bei dem Daten eingelesen werden, und einem *Reducing*-Prozess, bei dem die Zwischenergebnisse aus dem *Mapping* zusammengeführt werden. In diesem Anwendungsbeispiel werden zwei *Mapping*-Prozesse benötigt: einer zum Einlesen der *.csv*-Dateien aus den Maschinen und einer für das Ermitteln der Grenzwertüberschreitungen in den *.parquet*-Dateien, die aus dem *iba*-System kommen. Die *Mapping*-Prozesse können parallel auf allen *Nodes* des *Hadoop*-Systems ausgeführt werden. Nach dem *Mapping* der *.csv*-Dateien liegen als Zwischenergebnisse zum Beispiel bestimmte Fehlermeldungen mit der Angabe der Maschine sowie den dazugehörigen Zeitstempeln vor. Aus dem *Mapper* der *.parquet*-Dateien gehen zum Beispiel alle Zeitstempel einer Toleranzüberschreitung hervor. Die beiden Zwischenergebnisse werden in einem Zwischenschritt zusammengeführt. Der *Reducing*-Prozess ermittelt nun, ob es eine zeitliche Übereinstimmung zwischen Verformung und den Fehlermeldungen gibt und speichert diese Vorkommnisse als Ergebnis in eine Datei. Die Prozessingenieure kennen nun die problematischen Maschinen und können diese näher untersuchen, um den Prozess zu verbessern.

Aus dem Beispiel wird ersichtlich, dass *MapReduce* einerseits einen hohen Entwicklungsaufwand mit sich bringen kann, aber andererseits auch hochkomplexe, individuelle Analysen vorgenommen werden können. Der Entwicklungsaufwand rentiert sich, wenn die Analyse zudem häufig wiederholt wird. Unter bestimmten Bedingungen können *MapReduce Jobs* auch mithilfe anderer *Apache*-Tools und vergleichsweise einfach generiert werden. Wie im nachfolgenden Abschnitt beschrieben wird, ist *MapReduce* nicht für jeden Anwendungsfall geeignet. Für einfache Analysen sind möglicherweise andere Systeme ausreichend.

3 Wann eignet sich Hadoop nicht?

Hadoop ist keinesfalls für jedes Anwendungsszenario geeignet. Die folgenden Kriterien sind ein Indiz dafür, dass sich *Hadoop* nicht für die Datenverarbeitung eignet oder sich der Aufwand nicht rechtfertigt.

Digitalisierungsgrad

- Es existiert keine Digitalisierungsstrategie.
- Die relevanten Prozesse im Unternehmen sind noch nicht digitalisiert.

Dateninfrastruktur

- Es gibt keine dedizierten Datenserver.
- Bestehende Datenbanken weisen Lücken auf oder sind nicht konsistent.

Datenanalyse

- Es gibt keine Analyse-Strategie.
- Es gibt kein Fachwissen über professionelle Datenanalyse.
- Es gibt keine Ressourcen für die Entwicklung der automatischen Analyse.

Viele Innovationen und zukunftsorientierte Technologien, darunter auch *Hadoop* als *Big Data* Anwendung haben gemeinsam, dass sie in kurzer Zeit Trends und übertriebene Erwartungen auslösen. Diese relativieren sich nach einer Zeit und entwickeln sich weiter, bis die Technologie im Produktivumfeld ankommt. [3,18] Im Laufe dieser Phasen schärft sich ebenfalls der Anwendungsbereich einer Technologie. Auch *Apache Hadoop* ist keine universelle Lösung für jeden Anwendungsfall, bei dem Daten gespeichert werden.

In der Einführung zu *Hadoop* und den Erklärungen zum *Parquet*-Format in Kapitel ↗ *Apache Hadoop*, Seite 5 wurde bereits auf die günstigen und weniger günstigen Dateiformate für das HDFS hingewiesen. Zwar können grundsätzlich alle Formate im HDFS abgelegt werden, allerdings bietet dieses Vorgehen nur wenige oder gar keine Vorteile. Eine strukturierte Auswertung von Daten mithilfe eines *MapReduce Jobs* wird zunehmend komplexer, wenn es sich um Dateien handelt, die eine Vorverarbeitung benötigen. Dokumente, die händisch von Mitarbeitern erstellt und verwendet werden, sollten nicht im HDFS abgelegt werden, sondern nach wie vor in einem für die Mitarbeiter zugänglicheren Dateisystem.

Relationale Datenbanken im Unternehmen können ebenfalls nicht in jedem Fall durch ein *Hadoop*-System abgelöst werden. Eine leistungsfähige Datenbank kann mittels der Indizierung in sehr kurzer Zeit ein Ergebnis zurückliefern. *Apache Hadoop* allerdings speichert Daten in Dateien und indiziert sie nicht. Um eine Abfrage durchzuführen, muss ein *MapReduce Job* ausgeführt werden, der alle Daten durchsucht. Der *MapReduce*-Prozess wird in vielen Fällen eine längere Zeit in Anspruch nehmen. Im Allgemeinen gilt, dass *Hadoop* erst dann eine relationale Datenbank ersetzen kann und sollte, wenn deren technische Grenzen erreicht sind. Das ist beispielsweise der Fall, wenn die Datenmengen so hoch sind, dass die Indizierung zu aufwendig wird. [19]

Neben den technischen Aspekten gibt es noch die wirtschaftliche Perspektive auf *Apache Hadoop*, die oft schwer zu beziffern oder einzuschätzen ist. Die quelloffene Programmbibliothek

von *Apache Hadoop* können Unternehmen ohne Lizenzkosten verwenden [20]. Man könnte deswegen annehmen, dass *Hadoop* kostengünstiger als ein professionelles, spezialisiertes Datenbanksystem ist. Allerdings müssen einige Gegebenheiten berücksichtigt werden. *Apache Hadoop* ist eine umfangreiche Software, die auf *Java* basiert und für *Linux* entwickelt wurde. Das Planen, Aufsetzen und Administrieren von *Hadoop* als Produktivsystem erfordert fortgeschrittene Linux-Kenntnisse und Fachwissen zum Thema Netzwerkkonfiguration. [19] Zudem wird für den Produktiveinsatz ein konfiguriertes Cluster benötigt, das in der Regel sehr komplex zu konfigurieren ist. Die genannten Schritte müssen bei der Einführung von *Hadoop* mit einkalkuliert werden. *iba* empfiehlt, die *Hadoop* Distribution eines professionellen Anbieters zu verwenden oder die Inbetriebnahme durch einen Dienstleister vornehmen zu lassen, falls im Unternehmen keine dedizierten IT-Ressourcen zur Verfügung stehen.

4 Lessons Learned

Die Entscheidung für ein übergeordnetes *Hadoop*-System muss gut durchdacht sein. Das Aufsetzen, Konfigurieren und Integrieren eines *Hadoop Clusters* in die bisherige Dateninfrastruktur bedeutet einen entsprechenden Aufwand und setzt eine klare Strategie darüber voraus, welche Erkenntnisse aus den *MapReduce Jobs* gewonnen werden sollen. In jedem Fall sollten vorher die Anforderungen im Unternehmen an die Dateninfrastruktur mit den Eigenschaften von *Hadoop* abgeglichen werden. Wenn sich die Anforderungen nicht mit diesen decken, eignet sich womöglich eine andere Datenspeicherungslösung besser.

Wenn die Einführung eines neuen *Hadoop*-Systems geplant ist, empfiehlt *iba* eine professionelle Distribution oder die Verwendung von *Hadoop* im Rahmen einer Cloud. Besteht im Unternehmen bereits sowohl ein *Hadoop*-System als auch ein *iba*-System, können die Daten mithilfe von *ibaDatCoordinator* extrahiert und in das HDFS verschoben werden.

5 Glossar

Apache Hadoop	Software für ein verteiltes Dateisystem auf einem *Cluster*, mit der auch *MapReduce Jobs* durchgeführt werden können.
Apache Hive	Erweiterung zu *Apache Hadoop* um Datenabfrage-Funktionalitäten zum Beispiel über SQL.
Cluster	Vernetzter Verbund von Computern.
.dat	Dateiformat von *iba* für die Messdatenaufzeichnung, das im *iba*-System zum Einsatz kommt.
HDFS	*Hadoop Distributed File System*: Dateisystem von *Apache Hadoop*, das Dateien innerhalb eines Clusters speichert und verwaltet.
MapReduce	Algorithmus für die parallele Datenverarbeitung von großen Datenmengen auf *Clustern*.
Node	Einzelner Computer innerhalb eines *Clusters*.
.parquet	Dateiformat, das für die Verwendung mit *Apache* Softwaretools entwickelt wurde.
SQL	*Structured Query Language*: Datenbanksprache zum Erstellen und Bearbeiten von Datenbankstrukturen.
YARN	*Yet another resource negotiator*: Ressourcenmanager in *Apache Hadoop*.

6 Quellen und Verweise

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

[2] Mielke, Fabian (2021). *ibaHD-Server Benchmark. Vergleich zwischen ibaHD-Server und zeitbasierten Datenbanksystemen.* White Paper. Ausgabe 1.0. iba AG (Hrsg.). Fürth, Deutschland.
URL: https://www.iba-ag.com/de/news/ibahd-server-benchmark (aufgerufen am 07.10.2021)

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

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

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

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

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

[14] iba AG (2021). ibaAnalyzer. Einführung und Installation. Handbuch Teil 1. Ausgabe 7.2, S. 13-14.

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

[16] iba AG (2021). *Extraktionsaufgabe* und *Skriptaufgabe*. *ibaDatCoordinator* Handbuch. Ausgabe 2.3, S. 53-56.

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

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

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

7 Support und Kontakt

Support

Tel.: +49 911 97282-14

Fax: +49 911 97282-33

E-Mail: support@iba-ag.com

Hinweis

Wenn Sie Support benötigen, dann geben Sie bitte bei Softwareprodukten die Lizenznummer bzw. die CodeMeter-Containernummer (WIBU-Dongle) an. Bei Hardwareprodukten halten Sie bitte ggf. die Seriennummer des Geräts bereit.

Kontakt

Hausanschrift

iba AG
Königswarterstraße 44
90762 Fürth
Deutschland

Tel.: +49 911 97282-0

Fax: +49 911 97282-33

E-Mail: iba@iba-ag.com

Postanschrift

iba AG
Postfach 1828
90708 Fürth

Warenanlieferung, Retouren

iba AG
Gebhardtstraße 10
90762 Fürth

Regional und weltweit

Weitere Kontaktadressen unserer regionalen Niederlassungen oder Vertretungen finden Sie auf unserer Webseite

www.iba-ag.com.