White Paper

Think big

How big data will change your approach to financial risk management

Watson Financial Services

IBM

Highlights

- Adoption of big data by financial institutions is continuing to gain momentum
- Financial risk management is a good use case for big data
- Performance improvements enable new operating models
- Built-in redundancy and fault tolerance prevent failure points

Introduction

When people talk about big data today, they mean more than just Hadoop and the ability to manage lots of files. Today, big data refers to the entire collection of components that surround Hadoop and make it possible to run complete business processes. Historically, big data has been applied to business processes centered around the core areas of search, social and e-commerce. A quick internet search of “big data” reveals many examples of use cases that focus on these three areas. This observation makes sense when you think about the growth of internet-related companies over the past 15 years and then realize that these companies are some of the biggest contributors to big data technology.

Today the story is different because big data has matured to the point where it’s possible to apply the technology to sophisticated business processes outside of search, social and e-commerce. Many of these applications don’t qualify as “big” when simply looking at the amount of data involved. However, even with “small” data, big data technology can have tremendous benefits, ranging from improved performance, better resiliency and improved operating models.

For several years, IBM has been working with big data and testing its applicability to the business processes that support financial risk management. The initial tests exceeded expectations and IBM moved forward to replatform its core simulation framework to run on big data. But what is big data? What is it used for? And why is it such a good fit for financial risk management?

What is big data?

The problem of managing an ever-increasing amount of data goes back well over 300 years. From John Graunt's 1663 experiments in statistical data analysis on mortality, to the 1881 invention of the Hollerith Tabulation Machine to dramatically reduce US census data analysis efforts, to Codd's 1970 framework for relational databases, people have been designing solutions to manage and analyze increasing amounts of data. More recently, NASA researchers Michael Cox and David Ellsworth are commonly cited as the first people to use the term "big data" in their 1997 NASA paper, Application- Controlled Demand Paging for Out-of-Core Visualization,[1] The paper describes the challenge of visualizing large scientific data sets, which "can surpass 100 Gbytes." Although 100 Gbytes is not a particularly large amount of data by today's standards, the concept is the same. When "data sets are generally quite large, taxing the capacities of main memory, local disk, and even remote disk," Cox and Ellsworth go on to point out, "we call this the problem of big data."

The current big data technology stack has its origins in two research papers published in 2003 and 2004: The Google File System[2] and MapReduce: Simplified Data Processing on Large Clusters.[3] These papers define a distributed files system and the methods for managing tasks across a cluster of nodes, which is important for accessing large data sets. New data is being created at an ever-increasing rate. According to IDC, the total amount of data in the world in 2013 was 4.4 zettabytes.* This figure is expected to grow by a factor of 10 to reach 44 zettabytes by 2020,[4] and by a factor of 40 to reach 180 zettabytes by 2025.[5] To deal with this rapid growth, companies, such as Google, Yahoo, Facebook, Hortonworks, Cloudera, LinkedIn, eBay, IBM and others, have contributed to the development of Hadoop and the rest of the big data technology stack.

What is big data?

Founded in 2006, Hadoop is an open source Apache project that implements a distributed file system, and supports data processing engines, such as MapReduce and Spark. A typical Hadoop cluster includes multiple servers, or nodes, each with multiple disks. The core framework includes:

- Hadoop Distributed File System (HDFS)[6]: a system that manages the files across multiple nodes and disks on the cluster
- Hadoop YARN[7]: a scheduler that manages resources, tasks and jobs running on the cluster
- Hadoop MapReduce[8]: an implementation of the MapReduce programming library

Other important components that are part of the Apache big data technology stack include:

Ambari: Created by Hortonworks, Ambari is a web-based administrative tool for provisioning, configuring and monitoring a Hadoop cluster.

Hive: First developed by Facebook, Hive is a data warehousing framework that implements an SQL-like query language named Hive Query Language (HQL). Hive is similar to a relational database management system (RDBMS) in which users create a schema of tables, but the data resides in text files on HDFS. HQL statements are compiled into MapReduce statements for execution.

Tez: Created by Hortonworks, Tez is a replacement MapReduce engine that attempts to speed up certain components, such as Hive, by employing in-memory caching and introducing the execution of graphs of tasks.

HBase: A distributed NoSQL database, built to the design described in Google's Bigtable research paper,[9] that uses HDFS as a data store. It's designed for large-scale read and write record access, containing hundreds of millions of records, and can handle varying workload styles, from batch processing to real-time data access. And unlike Cassandra, HBase has built-in support for data versioning.

Cassandra: Developed by Facebook, Cassandra is a distributed NoSQL database that's similar to HBase in performance and design heritage, with a notable exception of having an SQL-like language, Cassandra Query Language (CQL). Another major difference is Cassandra's tunable "eventual" data consistency compared to HBase's support of atomicity, consistency, isolation, and durability (ACID) semantics. Cassandra's lower initial consistency among nodes results in higher write performance, helping to ensure updates are cluster-wide at the time of completion at the cost of data synchronization overhead.

Spark: Originally developed by UC Berkeley's AMPLab, Spark is a data processing engine that addresses some of the limitations of the MapReduce engine. It works as a read-only distributed in-memory data set that allows iterative algorithms to loop over the data set, unlike MapReduce, which only allows a single pass over the data set.

A brief history of big data

Some of the major milestones in timeline for the evolution of the Hadoop and big data technology stack include:

- 2003 Google File System (GFS)[10]
- 2004 Google MapReduce[11]
- 2006 Apache project begins at Yahoo
- 2008 Facebook develops Cassandra as an open source Google project
- 2008 Cloudera founded to manage Hadoop clusters
- 2009 Facebook spins out Hive as an Apache project
- 2009 Spark development begins at UC Berkeley
- 2009 Cassandra becomes an Apache incubator project
- 2010 Apache HBase graduates from a Hadoop subproject to an Apache top-level project
- 2011 Yahoo spins out Hortonworks
- 2011 McKinsey issues its initial expectations around big data[12]
- 2011 IBM launches IBM® BigInsights®[13]
- 2012 Apache Hadoop 1.0 is released
- 2012 Yarn becomes an Apache subproject
- 2013 Ambari becomes an Apache incubator project
- 2013 "Big data" officially defined in the Oxford English Dictionary[14]
- 2013 Tez becomes an Apache incubator project
- 2014 Ambari added to the Apache/Hortonworks toolkit
- 2015 IBM releases SystemML
- 2015 Impala becomes an Apache incubator project
- 2015 IBM joins forces with Spark[15]

Open source and Hadoop distributions

Because the Apache big data projects are all open source, anyone can download and use them. However, the lack of official support can be a deterrent to commercial use. Similar to Linux, there are several companies that address this deficiency by selling support packages around the open source components. These companies create big data “distributions” by downloading specific versions of the various components, providing additional interoperability testing, adding proprietary configuration and management modules, and then selling support. The major companies providing big data distributions are Cloudera, Hortonworks, MapR and IBM.

Cloudera:

- Founded in 2008 by former employees of Google, Yahoo, Facebook and Oracle
- Known for conservatively releasing upgrades
- Created the open source Impala low-latency SQL engine, which can directly read data from HDFS and HBase storage; shares the same metadata and SQL syntax as Hive, and is intended to be a high-performance replacement for Hive
- Has a proprietary deployment, configuration and monitoring tool, Cloudera Manager, that’s a replacement for the open source Ambari

Hortonworks:

- Originally maintained by Yahoo, and spun out as an Apache incubator project and independent company in 2011
- Known for employing large numbers of contributors to the Apache Foundation

MapR:

- Founded in 2009 by former employees of Google and EMC
- Known for its fault tolerance and performance improvements to the file system and HBase
- Created a proprietary version of HBase that interfaces directly in the file system layer to improve performance, as well as serve as a replacement for HDFS, called MapR FS

IBM:

- Released IBM InfoSphere® BigInsights in 2011
- Known for its strong support of Spark and machine learning extensions, as well as proprietary extensions to Hadoop
- Provides a unified SQL interface to Hive, HBase and Spark with IBM Big SQL using a single database connection, as well as providing federated access to non-big data databases, such as IBM DB2® and Oracle
- Enables R to be used as a query language for data sources exposed by Big SQL using the IBM InfoSphere BigInsights Big R extension; also enables pushing down R functions to each node in the cluster to execute user-written code across the cluster on localized data

Why big data for financial risk management?

Although the amount of data required to run a bank's traditional end-of-day risk management system is large, it's not "big" when compared to the data sets used in other domains, such as search, social or e-commerce. However, risk management shares many common data patterns with these other areas. As a result, big data technology is ideal for delivering performance improvements to risk calculations.

Risk management systems require a wide range of data from different source systems. Using a traditional RDBMS to store, map and partition the data doesn't scale well. As the instrument and position volumes increase, an ever-increasing percentage of the overall time is spent managing the data rather than running the risk calculations and generating reports. With big data, Cassandra, MapReduce and Spark all provide scalable alternatives for implementing extract, transform, load (ETL) logic, partitioning instruments into small "units of work," and generating reports.

In general, many risk management jobs are trivially parallel. By partitioning the instruments to be analyzed, simulation jobs can be run in parallel across all available nodes. However, running many more smaller jobs increases the total number of database operations when compared to running fewer larger jobs. By utilizing the big data approach of moving the calculation to the data rather than moving the data to the calculation, overall run times can be reduced because multiple database read and writes are eliminated. Another challenge with running lots of small jobs is scheduling them efficiently across the available nodes. This issue involves solving the classical bin packing problem, which is NP-complete; the only way to find the best solution is to try them all. As a result, schedulers use algorithms and heuristics to quickly find a good, but not necessarily the best solution. On big data, if a job cannot be scheduled on a node with the data because the node is busy, the data must be copied to a node that is available. In IBM's initial work, the big data scheduler, Yarn, proved to be very efficient at scheduling; less than two percent of the jobs required data to be copied.

Risk management is often considered a mission-critical application and most financial instructions invest to help prevent failure points. Because Hadoop makes multiple copies of each file, and spreads the copies across all the drives and nodes in the cluster, redundancy and fault tolerance are built into the system. A single drive or node on the cluster can fail but, because there are two other copies of every file available, there is not a single point of failure. If a disk crashes or a node dies, the job is restarted on another node.

By eliminating the data movements, speeding up the ETL, and efficiently parallelizing the calculations, overall runtimes on big data are measured in tens of minutes rather than hours. As a result, financial institutions can eliminate complex processes required to deal with exception handling when something goes wrong. With these performance improvements, you no longer need special error correction runs that work on small subsets of the data. You can simply fix the problem and rerun the entire job. Furthermore, multiple intraday runs are now possible. In other words, the operating model for the risk system can be moved from a single static overnight batch run to a new dynamic environment that allows on-demand runs to handle virtually any number of situations. Situations include correct bad input data, run a special analysis after a market event, satisfy a regulator request to rerun for a specific historical date, or create a new ad hoc report for senior management. These use cases are possible on big data because the run times are so short.

The IBM story

In the summer of 2015, IBM decided to perform a short proof-of-concept (POC) test using big data to see if it addressed a performance issue that one of its clients was experiencing running a liquidity risk system.

The existing system was a traditional batch process where large numbers of files were prepared in an ETL phase and passed to a scheduler. The scheduler then ran the simulations in parallel across the available CPUs, and the output was loaded into a database for final reporting.

The ETL phase for the risk system was IBM Algo® Datamart, which is an RDBMS where input data records from various source systems are loaded using proprietary software into the database. The system then performed the following operations:

- Executed referential checks on input record dependencies
- Ran business logic to compress the input records, pooling the positions
- Transformed the compressed records into the IBM RiskWatch® format
- Created a series of input files to allow the simulations to run in parallel across the available CPUs

In the simulation phase, RiskWatch processed each input file created by the ETL stage and calculated the scenario-depending cash flows and other analytics for each position in the input file. For each input, RiskWatch performed the following operations:

- Loaded the input file, plus auxiliary files, such as scenarios
- Ran the simulation in memory, for example, generated cash flows, valuations and so forth
- Stored simulation results, for example cash flows and values, in proprietary binary files

The reporting phase required loading the simulation results into an RDBMS, and then creating a business intelligence (BI) cube:

- Data was exported from the RiskWatch cubes to comma separated values (CSV) format
- This CSV data was loaded into a staging table in the database
- The staging table was loaded into a dimensional table
- The BI tool created a cube from the dimensional table

Proof of concept

The POC that ran from July 2015 through September 2015 used big data technology to improve the performance and operability of the current system.

ETL phase

The ETL modeling subsystem in the existing setup performed several rules-based transformations and referential integrity checks that would have required substantial effort to replicate with a MapReduce equivalent. As a result, the POC used only a subset of positions comprising about 60 percent of the portfolio. The positions selected included both the highest volume instruments and those that were the most complicated to model. For these positions, a MapReduce job was created to perform the ETL with the goal to determine both the feasibility of using MapReduce as well as the effort required to create an ETL phase using this technique.
As expected, a record-by-record transformation fit the MapReduce technique very well. Static data required for translation and reference chasing was loaded in memory from the distributed cache, and multiple map phases were used for the complex positions, as they required more than one pass through the data. The performance improvement was substantial using the MapReduce implementation over the existing system. The big data version took approximately two minutes to model the two position types, which was 60 percent of the portfolio, compared to over 60 minutes using the traditional approach.

Simulation phase

There were three different approaches considered for running RiskWatch on the Hadoop cluster: Yarn distributed shell, Java MapReduce job and Hadoop Streaming. In each case, the positions to be simulated were divided into approximately 2,000 files of equal size. While all three methods were attempted, ultimately, Hadoop Streaming† was the method adopted, as the other two methods had issues that required unwieldly workarounds.

Hadoop Streaming met the design goals because it was easy to pass the input files to RiskWatch. It was also easy to develop a small amount of Java code to inform the resource manager about some extra files containing static data that were also required for the simulation. This approach was successful because:

- Hadoop task and resource managers understood the number of map tasks needed to be spawned
- Each task had a very high hit rate of data affinity, for example, typically over 98 percent of the tasks were run on a data node where the position file was located
- The job management tools understood the tasks were all related because they were all part of a single job
- Efficient use of the Hadoop distributed cache was simple to achieve on the command line because a tar file containing scenarios and market data was supplied with the job submission and made available to each map task
- All process and file management within the task were easy to maintain because the streaming job was written as a shell script

† Hadoop Streaming is a utility that allows users to create MapReduce jobs that can run any executable or script as the mapper or reducer.

Reporting phase

In keeping with the “move computing to the data, not data to the computing” philosophy of big data, the computed cash flows were left on HDFS, and exposed by way of a Hive schema. To perform the aggregations, the BI tool was configured with the Hive ODBC driver, and a connection was made to the Hive service on the cluster. In addition, rather than computing haircuts at simulation time, they were applied on the fly by way of SQL.

In the original system, the time involved setting up the first query was substantial. The two most time-consuming steps were copying the cash flows to a staging table in the RDBMS, and then transforming them to a dimensional model. Subsequently, a cube was created in the BI tool to prepare for the first report. Directly querying Hive views from the BI tool effectively removed this overhead, and replaced it with a series of MapReduce jobs run under Tez. While this method also had overhead, these views were materialized as optimized row columnar (ORC) tables within Hive, allowing very good interactive performance.

Performance results

The cluster used to run these results was a 12-node cluster where each node had 48 cores, totaling 576 cores. Twenty percent of the cluster’s capacity was allocated per risk run.

The ETL modeling results, as discussed previously, were extremely promising. Extrapolating from the earlier results, the runtime of the ETL for the entire data set, using the big data technique, is estimated to be under four minutes, reducing the total batch time by almost an hour.

The simulation phase, however, was not expected to yield much in the way of performance improvements. The bulk of the time spent during this phase is typically spent within RiskWatch running CPU-intensive calculations. However, the existing system relies on a client server model to obtain positions, which, in turn, relies on retrieving data from an RDBMS. Once the number of simultaneous requests hits a high enough number, the system rapidly degrades. Using HDFS and Yarn to distribute the data and processing, and relying on data locality, the technique explored here has no scalability issues, and is, in fact, remarkably linear as the following chart displays.

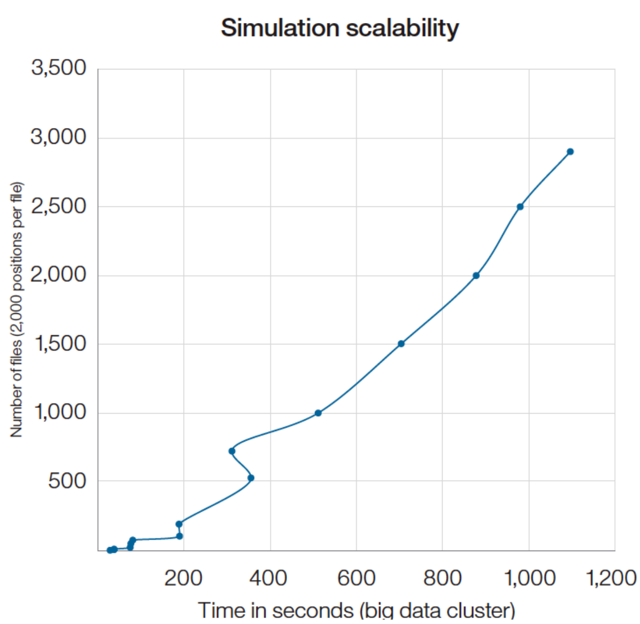

Figure 1: Scalability of simulation

For the reporting phase, the results also improved dramatically. In the original system, the time to render the first cash flow report was 280 minutes, including staging cash flow data, transforming to a dimensional model and creating a BI cube. Furthermore, the reports used **compressed** data: 350 thousand positions after **compression** from the original 6 million positions. Accessing the data through a Hive schema-on-read on the uncompressed data, the first report was available in under one minute.

The performance improvements changed the operational characteristics of the system. In the original system, given typical operational delays, at best the system could be run once per day. Using a big data approach, the system can be run with full data granularity over 10 times a day. The result is more accurate reporting, both by running uncompressed data and providing the capability to have rerun the job after making corrects to the input data. Furthermore, the shorter run times provide a more flexible operating model that supports rerun for other ad hoc analysis.

A first set of solutions on big data

Given the performance enhancements gained by adopting big data design concepts, IBM refactored the IBM Algo One® Foundation to exploit the strengths of the platform. The initial version of the Algo One Big Data Foundation was released in December 2016. It included enhancements to key legacy components: the RiskWatch simulation engine, the Riskflow scheduler and Datamart data loading and mapping service as follows:

- Datamart was enhanced to run as a distributed process and can run on multiple nodes
- Datamart was integrated with Apache Cassandra for data storage and handling
- Datamart was integrated with Apache Spark to provide faster data loading and mapping when compared to MapReduce
- Simulation results written to HDFS are accessible using Hive schema-on-read data structures for ease of reporting
- Riskflow, which schedules the data loading and risk simulation processes, was integrated with MapReduce and Yarn

With the creation of the Algo One Big Data Foundation, IBM has shifted from a three-tier legacy architecture to big data as the base platform. This foundation is designed to underpin many of IBM's risk analytics solutions, such as asset liability management (ALM), liquidity risk, market risk and so forth.

IBM's new balance sheet risk management solution, which is focused on ALM and liquidity risk, is being built to run on top of the Algo One Big Data Foundation. Because these risk solutions require large volumes of instrument data and granular cash flows, ALM and liquidity risk is well suited to take advantage of many of the key benefits provided by big data.

In addition to the performance benefits gained by using big data technology at its core, IBM's new ALM and liquidity risk solution also introduced a new user experience (UX). Creation of a new user interface (UI) enables users to manage and configure virtually all aspects of the ALM and liquidity risk solution, including hierarchy structures, reporting time steps and modelling approaches. Following IBM Design Thinking, the system provides a "one UI" approach that simplifies tasks and streamlines workflow. A new sandboxing capability allows changes to batch and ad hoc runs, with multiple intraday runs now possible due to big data. Front-end interactive reporting is provided to structure reports on high-volume results.

Following the launch of ALM and liquidity risk on big data, other risk solutions are being refactored to run on the big data platform. POCs with clients recently confirmed the similar performance and scalability for market risk and stress testing, demonstrating these benefits along both the calculation and aggregation dimensions.

Beyond this, IBM positions itself to offer additional risk solutions, pulling the benefits of big data as the core platform and enabling capability across virtually all risk solutions.

Conclusion

Big data technology has already arrived at financial institutions. Many firms are either starting to experiment or plan to experiment with big data. However, the tests so far have only scratched the surface of the platform's capabilities. Risk management systems are a good use case because they take advantage of many of the benefits provided by big data. These benefits include; faster ETL, highly parallelizable and scalable simulations, improved fault tolerance and redundancy, quicker reporting, shorter run times, and more flexible operating models. These benefits mean more accurate and timely reporting at a more granular level. Over the last several years, IBM has worked with multiple clients to validate these benefits and offers its leading risk solutions on big data.

About the authors

Nadia Abuseif joined Algorithmics® in June 2011, shortly prior to its acquisition by IBM in 2012, and currently manages strategic projects for the Research and Innovations team in the financial risk segment of IBM Watson® Financial Services. This group is responsible for delivering cutting-edge technology in the risk space through multiple client-engaged POCs and initiatives. Additionally, Nadia spearheads the offering management project management office (PMO), which is responsible for enabling decision forums, as well as creating and communicating the overall portfolio strategy in the financial risk segment of IBM Watson Financial Services

Nadia draws strength from over 15 years of experience at various Canadian banks in areas including payments, trade systems and regulatory compliance. She graduated from McGill University in 2000 with a BA of Commerce Degree, majoring in information technology.

Curt Burmeister leads the Research and Innovations team, as well as the Quantitative Research and Financial Engineering team for the financial risk segment of IBM Watson Financial Services. The Innovations team incubates ideas that apply new technologies and methodologies to risk management. Most recently, the group led the effort to replatform IBM risk analytics' core simulation framework on big data. The Quantitative Research and Financial Engineering team is responsible for the research, design and validation of the financial models and other quantitative methods use by IBM's financial risk solutions. In addition to model enhancements, the team works on topics such as risk factor back testing, credit valuation adjustment (CVA) sensitivities and wrong way risk

Curt holds an MBA in Financial Engineering from MIT and a BA in Computer Science and Mathematics from Cornell University. He has a patent in compiler technology for register allocation and has cofounded three companies.

Peter Timofejew joined Algorithmics two years before its acquisition by IBM in 2012. Peter is an architect in the Research and Innovations team in the financial risk segment of IBM Watson Financial Services. As part of the Innovations team, Peter uses his industry experience, combined with research into cutting-edge technologies, to explore new approaches to solving financial risk management problems. Most recently, he has driven the adoption of big data as the new foundation for financial risk solutions.

Prior to joining IBM, Peter spent 14 years at TD Securities designing and building their market and counterparty credit systems, as well as a credit derivatives trading system. Before this position, he was part of the team at the Toronto Stock Exchange that converted the equity trading floor to be fully electronic, and was the author of STAMP, the electronic exchange traded equity and equity derivative messaging protocol.

About IBM Watson® Financial Services

IBM is working with organizations across the financial services industry to use IBM Cloud, cognitive, big data, RegTech and blockchain technology to address their business challenges. Watson Financial Services merges the cognitive capabilities of Watson and the expertise of Promontory Financial Group to help risk and compliance professionals make better informed decisions to manage risk and compliance processes. These processes range from regulatory change management to specific compliance processes, such as anti-money laundering, know your customer, conduct surveillance and stress testing.

For more information

To learn more about IBM financial risk and regulatory compliance solutions, visit ibm.com/RegTech and follow us on Twitter @IBMFintech

Footnotes

1. https://www.nas.nasa.gov/assets/pdf/techreports/1997/nas-97-010.pdf

2.https://research.google.com/archive/gfs.html

3.https://research.google.com/archive/mapreduce.html

4.https://www.emc.com/leadership/digital-universe/2014iview/index.htm

5.https://www.forbes.com/sites/michaelkanellos/2016/03/03/152000-smart-devices-every-minute-in-2025-idc-outlines-the-future-of-smart-things/

6.https://www-01.ibm.com/software/data/infosphere/hadoop/hdfs/

7.https://www.ibm.com/developerworks/library/bd-yarn-intro/

8.https://www-01.ibm.com/software/data/infosphere/hadoop/mapreduce/

9.https://research.google.com/archive/bigtable.html

10.http://research.google.com/archive/gfs.html

11.http://research.google.com/archive/mapreduce.html

12.http://www.mckinsey.com/business-functions/digital-mckinsey/our-insights/ big-data-the-next-frontier-for-innovation

13.http://www.prnewswire.com/news-releases/ibm-unveils-breakthrough-software-and-new-services-to-exploit-big-data-122345833.html

14.https://whatsthebigdata.com/2013/06/15/big-data-arrives-at-the-oxford- english-dictionary/

15.http://www-03.ibm.com/press/us/en/pressrelease/47107.wss

IBM Corporation
Route 100
Somers, NY 10589

Produced in the United States of America, April 2019

FDW12349-USEN-01

Watson Financial Services