Chapter No. 10
"HBase Use Cases"
In this package, you will find:
The author’s biography
A preview chapter from the book, Chapter no.10 "HBase Use Cases"
A synopsis of the book’s content
Information on where to buy this book

About the Author
Shashwat Shriparv was born in Muzaffarpur, Bihar. He did his schooling from Muzaffarpur and Shillong, Meghalaya. He received his BCA degree from IGNOU, Delhi and his MCA degree from Cochin University of Science and Technology, Kerala (C-DAC Trivandrum).

He was introduced to Big Data technologies in early 2010 when he was asked to perform a proof of concept (POC) on Big Data technologies in storing and processing logs. He was also given another project, where he was required to store huge binary files with variable headers and process them. At this time, he started configuring, setting up, and testing Hadoop HBase clusters and writing sample code for them. After performing a successful POC, he initiated serious development using Java REST and SOAP web services, building a system to store and process logs to Hadoop using web services, and then storing these logs in HBase using homemade schema and reading data using HBase APIs and HBase-Hive mapped queries. Shashwat successfully implemented the project, and then moved on to work on huge binary files of size 1 to 3 TB, processing the header and storing metadata to HBase and files on HDFS.

For More Information:  
Shashwat started his career as a software developer at C-DAC Cyber Forensics, Trivandrum, building mobile-related software for forensics analysis. Then, he moved to Genilok Computer Solutions, where he worked on cluster computing, HPC technologies, and web technologies. After this, he moved to Bangalore from Trivandrum and joined PointCross, where he started working with Big Data technologies, developing software using Java, web services, and platform as Big Data. He worked on many projects revolving around Big Data technologies, such as Hadoop, HBase, Hive, Pig, Sqoop, Flume, and so on at PointCross. From here, he moved to HCL Infosystems Ltd. to work on the UIDAI project, which is one of the most prestigious projects in India, providing a unique identification number to every resident of India. Here, he worked on technologies such as HBase, Hive, Hadoop, Pig, and Linux, scripting, managing HBase Hadoop clusters, writing scripts, automating tasks and processes, and building dashboards for monitoring clusters.

Currently, he is working with Cognilytics, Inc. on Big Data technologies, HANA, and other high-performance technologies.

You can find out more about him at https://github.com/shripav and http://helpmetocode.blogspot.com. You can connect with him on LinkedIn at http://www.linkedin.com/pub/shashwat-shripav/19/214/2a9. You can also e-mail him at dwivedishashwat@gmail.com.

Shashwat has worked as a reviewer on the book *Pig Design Pattern*, Pradeep Pasupuleti, Packt Publishing. He also contributed to his college magazine, *InfinityTech*, as an editor.

Acknowledgments

First, I would like to thank a few people from Packt Publishing: Kevin for encouraging me to write this book, Prachi for assisting and guiding me throughout the writing process, Pankaj for helping me out in technical editing, and all other contributors to this book.

I would like to thank all the developers, contributors, and forums of Hadoop, HBase, and Big Data technologies for giving the industry such awesome technologies and contributing to it continuously. Thanks to Lars and Noll for their contribution towards HBase and Hadoop, respectively.
I would like to thank some people who helped me to learn from life, including teachers at my college—Roshani ma'am (Principal), Namboothari sir, Santosh sir, Manjush ma'am, Hudlin Leo ma'am, and my seniors Jitesh sir, Nilanchal sir, Vaidhath sir, Jwala sir, Ashutosh sir, Anzar sir, Kishor sir, and all my friends in Batch 6. I dedicate this book to my friend, Nikhil, who is not in this world now. Special thanks to Ratnakar Mishra and Chandan Jha for always being with me and believing in me. Thanks also go out to Vineet, Shashi bhai, Shailesh, Rajeev, Pintu, Darshna, Priya, Amit, Manzar, Sunil, Ashok bhai, Pradeep, Arshad, Sujith, Vinay, Rachana, Ashwathi, Rinku, Pheona, Lizbeth, Arun, Kalesh, Chitra, Fatima, Rajesh, Jasmin, and all my friends from C-DAC Trivendrum college. I thank all my juniors, seniors, and friends in college. Thanks to all my colleagues at C-DAC Cyber Forensic: Sateesh sir, my project manager; Anwer Reyaz. J, an enthusiast who is always encouraging; Bibin bhai sahab; Ramani sir; Potty sir; Bhadran sir; Thomas sir; Satish sir; Nabeel sir; Balan sir; Abhin sir; and others. I would also like to thank Mani sir; Raja sir; my friends and teammates: Maruthi Kiran, Chethan, Alok, Tariq, Sujatha, Bhagya, and Mukesh; Sri Gopal sir, my team leader; and all my other colleagues from PointCross. I thank Ramesh Krishnan sir, Manoj sir, Vinod sir, Nand Kishor sir, and my teammates Varun bhai sahab, Preeti Gupta, Kuldeep bhai sahab, and all my colleagues at HCL Infosystems Ltd. and UIDAI. I would also like to thank Satish sir; Sudipta sir; my manager, Atul sir; Pradeep; Nikhil; Mohit; Brijesh; Kranth; Ashish Chopara; Sudhir; and all my colleagues at Cognilytics, Inc.

Last but not the least, I would like to thank papa, Dr. Rameshwar Dwivedi; mummy, Smt. Rewa Dwivedi; bhai, Vishwas Priambud; sister-in-law, Ragini Dwivedi; sweet sister, Bhumika; brother-in-law, Chandramauli Dwivedi; and new members of my family, Vasu and Atmana.

If I missed any names, it does not mean that I am not thankful to them, they all are in my heart and I am thankful to everyone who has come in my life and left their mark. Also, thanking is not in any order.

For More Information:
Learning HBase

This book will provide a top-down approach to learning HBase, which will be useful for both novices and experts. You will start learning configuration, code to maintenance, and troubleshooting—a kind of all-in-one HBase knowledge bank. This will be a step-by-step guide, which will help you work on HBase. The book will include day-to-day activities using HBase administration, and the implementation of Hadoop, plus HBase cluster setup from ground approach. The book will cover a complete list of use cases and explanations to implement HBase as an effective Big Data tool. It will also help you understand the layout and structure of HBase. There are lots of books available on the market on HBase, but they lack something in them; some of them focuses more on configuration and some on coding, but this book will provide a kind of start-to-end approach, which will be useful for a person with zero knowledge in HBase to the person proficient in HBase. This book is a complete guide to HBase administration and development with real-time scenarios and an operation guide.

This book will provide an understanding of what HBase is like, where it came from, who all are involved, why should we consider using it, why people are using it, when to use it, and how to use it. This book will give overall information about the HBase ecosystem. It's more like an HBase-confusion-buster book, a book to read and implement in real life. The book has in-depth theory and practical examples on HBase features. This theoretical and practical approach clears doubts on Hadoop and HBase. It provides complete guidance on configuration/management/troubleshooting of HBase clusters and their operations. The book is targeted at administration and development aspects of HBase; administration with troubleshooting, setup, and development with client and server APIs. This book also enables you to design schema, code in Java, and write shell scripts to work with HBase.

What This Book Covers

Chapter 1, Understanding the HBase Ecosystem, introduces HBase in detail, and discusses its features, its evolution, and its architecture. We will compare HBase with traditional databases and look at add-on features and the various underlying components, and its uses in the industry.

Chapter 2, Let's Begin with HBase, deals with the HBase components in detail, their internal architecture, communication between different components, how it provides scalability, as well as the HBase reading and writing cycle process, HBase housekeeping tasks, region-related operations, the different components needed for a HBase cluster configuration, and some basic OS tuning.

For More Information:

Chapter 3, *Let's Start Building It*, lets us proceed ahead with building an HBase cluster. In this chapter, you will find information on the various components and the places we can get it from. We will start configuring the cluster and consider all the parameters and optimization tweaks while building the Hadoop and HBase cluster. One section in the chapter will focus on the various component-level and OS-level parameters for an optimized cluster.

Chapter 4, *Optimizing the HBase/Hadoop Cluster*, teaches us to optimize the HBase cluster according to the production environment and running cluster troubleshooting tasks. We will look at optimization on hardware, OS, software, and network parameters. This chapter will also teach us how we can optimize Hadoop for a better HBase.

Chapter 5, *The Storage, Structure Layout, and Data Model of HBase*, discusses HBase's data model and its various data model operations for fetching and writing data in HBase tables. We will also consider some use cases in order to design schema in HBase.

Chapter 6, *HBase Cluster Maintenance and Troubleshooting*, covers all the aspects of HBase cluster management, operation, and maintenance. Once a cluster is built and in operation, we need to look after it, continuously tune it up, and troubleshoot in order to have a healthy HBase cluster. We will also study the commands available with HBase and Hadoop shell.

Chapter 7, *Scripting in HBase*, explains an automation process using HBase and shell scripts. We will learn to write scripts as an administrator or developer to automate various data-model-related tasks. We will also read about various backup and restore options available in HBase and how to perform them.

Chapter 8, *Coding HBase in Java*, teaches Java coding in HBase. We will start with basic Java coding in HBase and learn about Java APIs available for client requests. You will also learn to build a basic client in Java, which can be used to contact an HBase cluster for various operations using Java code.

Chapter 9, *Advance Coding in Java for HBase*, focuses more on Java coding in HBase. It is a more detailed learning about all the different kind of APIs, classes, methods, and interfaces available in Java for HBase. You will also see the different kind of web services or thrift services, which you can use to ease up the coding and using the inbuilt service and not implementing the entire architecture code in Java. This chapter has a section that includes a discussion of some special features of HBase and some open source projects available, which can be used in coordination with HBase for a production cluster and a project.

Chapter 10, *HBase Use Cases*, discusses the use cases in the industry, which are being used with HBase as their underlying technology.

For More Information:  
HBase Use Cases

In this chapter, we will walk through some HBase user cases, zoom in the project layout and design to understand how to utilize HBase for different business usages, and provide some references to you for further development.

We will see the use case and architecture used at:

- Facebook
- Groupon
- Pinterest
- LongTail Video

This chapter will also show how these leading entities are using HBase for their project and the architecture of the project.

HBase in industry today

With the increase in demand for Big Data day by day in order to solve the problems comprising planet-size data, HBase is one of the players due to its characteristic of persistent read performance. Using these systems, one can store and analyze data in coordination with various other tools.

Also features such as HBase running on cloud and expanding and reducing on demand will enable even small-scale companies to use its services. With the increasing demand of data storage space, the need to process it and represent unstructured data into a structured format can be achieved by HBase.

The new sub and open projects that are being created, which enables HBase to support SQL queries (Phoenix), will surely boost and prepare conventional SQL developers to perform queries on HBase. This provides us with a facility to query faster than any other relational database system with the data of terabytes and petabytes' size.

For More Information:
The future of HBase against relational databases

It would be wrong to say that NoSQL, a column-based database, will replace the RDBMS. HBase is still evolving and it cannot be used for all the use cases. There will always be a need for different types of database to work in coordination with each other satisfying different use cases to build a complete production environment.

However, the new features of high availability, almost consistent reads, ability to store and retrieve petabytes of data, and support of various evolving open source tools to fill up the existing gaps will take HBase and other NoSQL databases very far for sure. We have just started looking at the power of these systems and are yet to understand and implement real-life use cases with enormous amount of data.

So, for sure, Hadoop plus HBase and other subprojects are the future of data warehousing and massive processing.

Some real-world project examples' use cases

In this section, we will list out use cases of HBase being used in the industry today. References and more details can be found at links provided in the Useful links and references section at the end of the chapter.

HBase at Facebook

Facebook, as you all know, is a social utility that connects people with friends and others who work, study, and live around them. Facebook uses HBase mainly to power their messages infrastructure. The following are the services where Facebook uses HBase:

- Messages between users
- Chats
- E-mails
- SMS

Choosing HBase

The following are the reasons why Facebook chooses HBase:

- Provides high write throughput
- Good random read performance compared to other DBs

For More Information:

• Horizontal scalability
• Automatic failover
• Strong consistency
• Benefits of HDFS such as fault tolerant, scalable, checksums, and MapReduce

HBase acts like a caching layer on top of Hadoop when bundled these two together, faster data process compare with other NoSQL + Hadoop.

**Storing in HBase**

Facebook uses HBase to store data. The following are the types of data that are stored in HBase:

• Small messages
• Message metadata (thread/message indices)
• Search index

Facebook also use Haystack (a software that runs on a single machine and stores data without replication; it only cares about local aggregated blob storage, concentrating upon reducing disk seeks, and speeding up file retrieval) to store attachments and large messages.

**The architecture of a Facebook message**

The following is a diagrammatical representation of how things work:

For More Information:
In this process flow, a client asks the user directory service for the user details, then after getting the detail of the user, the client sends the request to the application server, which may be a Tomcat machine, running a custom application.

The application server using a custom service HBase index is searched for the user-related cell from where a message can be written, appended, or retrieved, and for the attachments the Haystack is searched.

Facts and figures
Let's have a look at some of the eye-opening facts and figures:

- Talking about Facebook message statistics, number of messages sent and received per day is more than 6 billion
- Talking about traffic to HBase, have a look at the following points:
  - Almost 100 billion read/writes per day
  - At peak time, more or less 1.5 million operations per second
  - About 55 percent reads and 45 percent writes
  - More than 2 petabytes of data and with replication, it's more than 6 petabytes
  - Data is compressed using Lempel–Ziv–Oberhumer (LZO) compression
  - And the data is growing around 300 terabytes per month
- Schema changes during production
- Heavy use of bulk import

Let's now see some other use cases in the industry.

HBase at Pinterest
Pinterest is deployed on Amazon Elastic Compute Cloud (EC2). Pinterest uses a follow model in which the user follows other users. It needs to update the feed data for every user as soon as a follower makes changes in a pin or updates a pin. This is the most classic social-media kind of application.

This happens for hundreds of millions of pins per month and about a billion writes per day. So the following are the specification of implementation:

- They choose a wide schema where each user's following feed is a single row in HBase
- This exploits the sorting order within columns as each user wants to see the latest in his/her feed

For More Information:
• They have increased the per region MemStore size to 512 MB MemStore which leads to 40 MB HFile instead of the smaller 8 MB file for default MemStore; this leads to less-frequent compactions
• Maintains mean time to recovery (MTTR) of less than 2 minutes by reducing various timeout settings such as socket, connect, and stale node.

The layout architecture
Let's now see the basic layout architecture. The following layout represents the follower and the followee relationship:

![Diagram showing follower and followee relationship]

And according to this structure, it forms a kind of graph inside. As you can see in the following diagram, it shows the path or writes in the system. Internally, there is a single row for each user in HBase and all the followers' and followees' information is stored in it.

![Diagram showing PinStore, FrontEnd, MessageBus, Workers, Thrift, HBase, FollowStore, and New Pin]

This is the HBase architecture that Pinterest uses to store hundreds and thousands of pins per month.

For More Information:
HBase at Groupon

Groupon is a deal-of-the-day website that features discounted gift certificates usable at local or national companies. Groupon was launched in November 2008. At Groupon, there are two requirements:

- Notify users about the deals via an e-mail
- Provide good user experience on the website

When they started off, they used Hadoop MapReduce jobs for e-mail deal delivery and used MySQL for their online application. Now, they have started relevance and a personalization system, both on HBase. They use a very wide schema in HBase as one column family for user history and profile and another for e-mail history.

The layout architecture

Now, let's see their read/write and architecture flow:

The data pipeline collects the data and keeps on writing to HBase, which is used for offline purposes such as sending e-mails from where the data is replicated to the HBase cluster, which is there for online relevance suggestions on the website.
The following are some of the optimization they performed:

- Presplit tables
- Increased lease timeout
- Increased scanner timeout
- Increased region size to 10 GB
- Keep less number of regions per region servers
- For heavy write jobs, they set the following values:
  - `hbase.hregion.memstore.block.multiplier` to 4
  - `hbase.hregion.memstore.flush.size` to 134217728
  - `hbase.hstore.blockingstorefiles` to 100
  - `hbase.zookeeper.useMulti` to false (for stable replication)

They performed all these changes in order to optimize and make HBase suit their requirement and this was done after evaluating the productivity and performance of the cluster.

**HBase at LongTail Video**

LongTail Video company provides JW Player, which is an online video player used by nearly 2 million websites. They are completely deployed on AWS and as such use HBase and EMR from Amazon.

Their requirements are:

- Very fast queries across datasets
- There should be a support for date-range queries
- Ability to store a huge amount of aggregated data
- There should be flexibility in dimensions used for rollup tables

HBase fits very well on these requirements. They use full-fledged Python shop so use HappyBase and have Thrift running on all the nodes of the HBase cluster for reading and writing.

Some statistics about them are as follows:

- 156 million unique viewers
- 1.04 billion video stream
- 3 TB compressed data per month and 12-15 TB uncompressed data per month

For More Information:

The layout architecture

This is a diagrammatical representation of the architecture:

HBase at Aadhaar (UIDAI)

UIDAI is a government organization of India that provides unique identification to Indian nationals. At UIDAI, all the open source software such as Hadoop, HBase, and Hive are being used. HBase is used to store data about the residents.

The layout architecture

Let’s see the architecture of the project that is available on the Internet:

For More Information:
There are some use cases and the architecture flow of the use cases that are being used in different companies in the industry today. This trend is increasing day by day with more people coming and including this technology and implementing it in their project requirement.

Hadoop and HBase and other Big Data components today are providing a complete infrastructure for a production system with data migration tools, data mapping tools, and some add-on open source projects.

Likewise, according to our requirements, we can first design the architecture of our project based on the components and the type of functionalities required, and we can choose from a wide variety of Hadoop and its subproject components.

**Useful links and references**

You can always use these links for reference.

The following are the use-cases-related links:

- [http://hbasecon.com/archive.html](http://hbasecon.com/archive.html)
- [https://www.facebook.com/UsingHBase](https://www.facebook.com/UsingHBase)
- [http://www.slideshare.net/brizzdotcom/facebook-messages-hbase](http://www.slideshare.net/brizzdotcom/facebook-messages-hbase)
- [http://www.slideshare.net/cloudera/case-studies-session-3b](http://www.slideshare.net/cloudera/case-studies-session-3b)
- [http://www.slideshare.net/cloudera/operations-session-1](http://www.slideshare.net/cloudera/operations-session-1)

These are some other references:

- [https://github.com/larsgeorge/hbase-schema-manager/tree/master](https://github.com/larsgeorge/hbase-schema-manager/tree/master)
- [http://www.jnbridge.com/jn/blog/tag/hbase/](http://www.jnbridge.com/jn/blog/tag/hbase/)
- [http://www.slideshare.net/cloudera/5-h-base-schemahbasecon2012](http://www.slideshare.net/cloudera/5-h-base-schemahbasecon2012)

For More Information:  
HBase Use Cases

The following links will be useful for schema designing:

- http://www.slideshare.net/amansk/hbase-schema-design-big-data-techcon-boston
- http://0b4af6cdc2f0c5998459-c0245c5c937c5dedcca3f1764ecc9b2f.r43.cf2.rackcdn.com/9353-login1210_khurana.pdf
- http://www.slideshare.net/amansk/hbase-schema-design-big-data-techcon-boston
- http://www.slideshare.net/cloudera/5-h-base-schemahbasecon2012
- https://www.youtube.com/watch?v=_HLoH_PgrLk

The following are the links to HBase books and references:

- https://www.youtube.com/results?search_query=hbase
- https://www.youtube.com/results?search_query=hadoop

Summary

In this chapter, we discussed the future aspect of HBase and the different use cases being implemented in the industry using HBase, its process flow, and architecture.

Today, the need of HBase is growing rapidly and we can get the list of companies whose projects are powered by HBase at http://wiki.apache.org/hadoop/Hbase/PoweredBy.

After reading this book, you should be able to move forward and design the use cases, performing administrative tasks and writing codes for HBase. Furthermore, the reader can always visit HBase Wiki, the HBase Apache website, and HBase source site for more updated and recent information.

HBase use has grown a lot but it still has a long way to go.
Where to buy this book

You can buy Learning HBase from the Packt Publishing website:

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.