

VIDYABHARTI TRUST COLLEGE OF BUSINESS, COMPUTER-SCIENCE AND RESEARCH, UMRAKH (Affiliated with Veer Narmad South Gujarat University, Surat)

Department of Computer Application Newsletter

GET TO KNOW MORE ABOUT

 $\left[\right]$

VOLUME 05 ISSUE 04 APRIL 2025

Vidyabharti Trust, Umrakh, At.&Po. Baben, Ta. Bardoli, Dist. Surat. 394601



Contents

Facebook Thrift	3
What is Thrift?	3
History and Development by Facebook	3
Supported Languages, Protocols, transports and servers:	4
Evolution of Facebook Thrift	5
The Real-World Example	7
Real-World Example of FBThrift :- How Facebook Uses It	7
Conclusion	9



Facebook Thrift

Facebook Thrift, commonly known as Apache Thrift, is an open-source software framework developed by Facebook to facilitate efficient communication across different programming languages in distributed systems. Initially designed for internal use, Facebook later contributed Thrift to the open-source community, and it is now maintained by the Apache Software Foundation.

Thrift enables developers to define services using a simple Interface Definition Language (IDL) and then generate code in multiple languages such as C++, Java, Python, JavaScript, and more. It provides a flexible, high-performance framework for Remote Procedure Call (RPC) communication, making it a preferred choice for companies handling large-scale distributed systems.

What is Thrift?

Thrift is an open-source **Remote Procedure Call (RPC) framework** designed for efficient and scalable cross-language communication in distributed systems. It allows developers to define data structures and services using an **Interface Definition Language (IDL)**, which is then compiled into multiple programming languages.

History and Development by Facebook

Thrift was developed by Facebook in 2007 to solve communication challenges within its large- scale distributed infrastructure. It was open-sourced in 2008 and later became an Apache Software Foundation project, enabling widespread adoption in various industries.



Supported Languages, Protocols, transports and servers: -

1. Thrift supports a wide range of programming languages including:-

- C++, Java, Python, JavaScript, PHP, Ruby, Go, and more.
- It also supports multiple protocols, transport and servers mechanisms

2. Protocols: -

- **TBinaryProtocol** A straight forward binary format encoding numeric values to binary,rather than to text. Simple, but not optimized for space efficiency. Faster to process thanthe text protocol but more difficult to debug.
- **TCompactProtocol** More compact binary format and most efficient.
- **TDebugProtocol** A human readable text format and easy to debug.
- **TDenseProtocol** Similar to TCompactProtocol but strips off the meta information from what is transmitted, and adds it back at the receiver.
- **TDenseProtocol** -is still experimental and not yet available in the Java implementation.
- **TJSONProtocol** Uses JSON for encoding of data.
- **TSimpleJSONProtocol** A write only protocol that cannot be parsed by Thrift because it drops metadata using JSON. Suitable for parsing by scripting languages.

3. Transport:-

- **TfileTransport** This transport writes to a file. This transport is not included with the Java implementation, but simple to implement.
- **TFramedtransport** Sends data in frames, where each frame is preceded by a length. This transport is required when using a non-blocking server.
- **TMemoryTransport** Uses memory for I/O.
- **Socket** Uses blocking socket I/O for transport.
- 4. Servers :-
 - **TNonblockingServer** A multi-threaded server using non-blocking I/O (Javaimplementation uses NIO channels). TFramedTransport must be used with this server.
 - **TSimpleServer** A single-threaded server using standard blocking I/O. Useful for testing.
 - **TThreadPoolServer** A multi-threaded server using standard blocking I/O.



Evolution of Facebook Thrift

Evolution of Facebook Thrift (FBThrift) – Explained Simply

Facebook Thrift (FBThrift) is an improved version of Apache Thrift, designed for highspeed communication between services in a distributed system. It evolved to meet Facebook's growing needs for scalability, efficiency, and performance. Let's go step by step through its evolution.

1. The Birth of Thrift (2007) – A Facebook Creation

- Facebook initially developed Thrift as an internal tool to improve communication between its services.
- Before Thrift, Facebook used traditional methods like REST APIs, but theywere slow and inefficient for large-scale systems.
- Thrift introduced Remote Procedure Calls (RPCs), which allowed different services to communicate as if they were calling functions within the same system.
- It supported multiple programming languages (C++, Java, Python, PHP, etc.), making it easier for developers to work with different technologies.

2. Open-Sourcing Thrift (2008) – Joining the Apache Community

- Facebook realized that Thrift could benefit other companies, so they donated it to the Apache Software Foundation in 2008.
- It became Apache Thrift, a widely used open-source RPC framework.
- Many companies adopted Apache Thrift for its cross-language support and efficient serialization.

3. The Need for a More Advanced Thrift (2010s) – Facebook's Growing Challenges

- Facebook's services kept growing, handling billions of requests per second.
- Apache Thrift was not optimized for massive scalability and low-latency communication.
- Facebook needed improvements in performance, security, and asynchronous processing.



4. FBThrift is Born (2013) – A Faster and More Scalable Thrift

- Better Performance Optimized for speed and lower memory usage.
- Asynchronous Processing Services could handle multiple requests at once.
- Improved Serialization Faster and more efficient data transfer.
- Security Enhancements–Added support for SSL/TLS encryption for secure communication.
- Multiplexed Services–Allowed multiple services to run on the same network connection.
- Facebook did not contribute FBThrift back to Apache because it was customized for Facebook's internal systems.

5. FBThrift Today – Powering Facebook's Infrastructure

- FBThrift is used for high-performance micro services at Facebook, WhatsApp, and Instagram.
- It is optimized for Facebook's massive infrastructure, handling real-time communication, data fetching, and AI model execution.
- While Apache Thrift is still popular, FBThrift remains a private, internal tool used by Facebook.

Summary of FBThrift's Evolution:-

Year	Event	Key Improvements
2007	Facebook develops Thrift	Faster communication between services
2008	Facebook donates Thrift to Apache	Becomes open-sourceas Apache Thrift
2010s	Facebook faces scaling issues	Apache Thrift lacks speed & scalability
2013	Facebook creates FBThrift	Faster, more scalable, async processing
Today	FBThrift powers Facebook's services	Used internally, not open- source



The Real-World Example

Real-World Example of FBThrift :- How Facebook Uses It

FBThrift is primarily used within Facebook's infrastructure to power communication between different services. Here's a real-world example of how Facebook leverages FBThrift:

Example: Facebook's News Feed Service

Imagine you open Facebook, and your News Feed loads instantly with posts, images, and videos. This happens because multiple backend services communicate efficiently using FBThrift. Let's break it down:

1. User Opens the Facebook App

- When you open Facebook, your app requests data from the backend to load the News Feed.
- This request is sent to a Front-End API Service.

2. FBThrift Handles Communication Between Services

The API Service does not store the News Feed itself. Instead, it uses FBThrift to communicate with multiple backend services:

- **Posts Service** \rightarrow Fetches new posts from your friends.
- Ads Service \rightarrow Retrieves personalized ads for your feed.
- Media Service \rightarrow Loads images and videos efficiently.
- **Ranking Service** \rightarrow Decides which posts to show first based on AI algorithms.
- Since Facebook handles billions of requests per second, FBThrift ensures: Fast data fetching (low latency)
 Efficient serialization (optimized data transfer)
 Asynchronous processing (multiple services work in parallel)



3. FBThrift Optimizes Large-Scale Communication

- Instead of sending separate requests for posts, ads, and media, FBThrift bundles them into a single efficient RPC call.
- This reduces network traffic and speeds up response times.
- Facebook's News Feed loads in milliseconds instead of seconds.

4. Secure and Reliable Communication

- FBThrift encrypts data using SSL/TLS to protect user information.
- It ensures fault tolerance, meaning if one service fails, the system can recover quickly without crashing the entire app.

Why Facebook Uses FBThrift Instead of gRPC?

- **Extreme scalability** \rightarrow Handles billions of users efficiently.
- **Custom optimizations** \rightarrow Tailored for Facebook's needs.
- Low-latency RPC → Ensures real-time performance for features like News Feed, Messenger, and Instagram Stories.

Other Real-World Use Cases of FBThrift

Besides News Feed, Facebook also uses FBThrift for:

- Messenger & WhatsApp \rightarrow Fast, encrypted messaging between servers.
- **Instagram Stories** \rightarrow Fetches and ranks stories in real time.
- AI & Machine Learning \rightarrow Helps Facebook train and deploy AI models efficiently.



Conclusion

Thrift is a powerful library for creating high-performance services that can be calledfrom multiple languages. Thrift will be a good choice for an application where there is need for multiple languages to communicate where speed is a concern and the clients and servers areca-located. Thrift might also make a good choice for IPC on a single machine where speed and/or interoperability are a concern.

Thrift is already used in wide variety of applications at Facebook. So many developers are also contributing at Apache to make the Thrift a scalable, efficient and system reliable technology. Thrift is a forthcoming technology for software paradigm.

- PROF. NIDHI PATEL







(A huge feat achieved by humanity!)