Thrift Evaluation

Content

[1 Web page 1](#_Toc329687581)

[2 Documentation 2](#_Toc329687582)

[2.1 Web’s documentation 2](#_Toc329687583)

[2.2 Manuals 2](#_Toc329687584)

[2.3 Installation 2](#_Toc329687585)

[3 Software 2](#_Toc329687586)

[3.1 Paradigm 2](#_Toc329687587)

[3.2 API 3](#_Toc329687588)

[3.3 Behavior 3](#_Toc329687589)

[3.4 Supported programming languages 3](#_Toc329687590)

[3.5 Platforms 3](#_Toc329687591)

[3.6 Tipos soportados 3](#_Toc329687592)

[3.7 Programming model 4](#_Toc329687593)

[3.8 Installation 4](#_Toc329687594)

[3.9 External libraries 4](#_Toc329687595)

[3.10 Making an example 4](#_Toc329687596)

[4 Tecnic details 5](#_Toc329687597)

[4.1 Performance 5](#_Toc329687598)

[5 Modelo de negocio 5](#_Toc329687599)

[5.1 License 5](#_Toc329687600)

[5.2 Project that use it 5](#_Toc329687601)

[6 Notes 5](#_Toc329687602)

# Web page

Web page’s design in vertical format with a top menu.

**Advantages:**

* Home page has a directly access to a *download* link and an example.
* Home page has a resume centered in the programming languages that are supported, and emphasize use it inmediatly.

**Disadvantages:**

* In comparison with ICE web, thrift web’s design is too simple. Its appearance is less commercial, as it was a little open source project.

# Documentation

## Web’s documentation

**Disadvantages:**

* There is very little documentation. There is only a whitepaper document that details at high level the product.

Highlights about Thrift that are exposed in the *whitepaper* document are:

* Thrift is divided in modules: transports, protocols, etc…
* Thrift supports a set of common types in large collection of different programming languages.
* Thrift provides a set of transports. Transports are modular and user can select which will be used.
* Thrift uses the term *Protocol* to refer to a type of serialization/deserialization. Protocols are also modular and user can select which will be used.
* Thrift provides a mechanism of versioning. User can change a function definition adding parameters and the communication between the client and the server still Works.

## Manuals

**Disadvantages:**

* Thrift’s package doesn’t offer manual.

## Installation

The installation’s format in Linux is the common way. There is information for some Linux distributions about what dependencies must to be installed and information about how compile the code. All this information is via web.

# Software

Thrift is a software framework , for scalable cross-language services development, combines a software stack with a code generation engine to build services.

## Paradigm

The paradigm is RPC.

## API

API is written in C++ and it’s oriented to objects.

**Advantages:**

* Object’s design is simple. For each functional there is an object. For example, for each transport exists a simple object that user can create and use.
* Thrift offers a *validdata* attribute for each field of the structure and a *isset* attribute that checks whether the field was set.

**Disadvantages:**

* The user has to managed too objects because there isn’t a dynamic mechanism to configure Thrift.

## Behavior

**Advantages:**

* Thrift exposes the TBufferedTransport object. It seems to work like a flow controller. Our product could expose something similar.
* Thrift exposes some strategies for the server in form of objects: TSimpleServer, TThreadedServer and TThreadPoolServer.
* The serialization doesn’t need to know about the length of the structure. Its binary protocol serializes in run-time and inserting identifiers for each field. Deserialization doesn’t need to know the initial IDL.

## Supported programming languages

**Advantage:**

* Thrift lets develop distributed applications in C++, Java, C#, Python, Ruby, PHP, Erlang, Perl, Haskel, Cocoa, JavaScript, Node.js, Smalltalk, OCaml and Delphy.

## Platforms

Thrift is supported in Windows 32-bits, Windows 64-bits, Linux 32-bits and Linux 64-bits.

**Disadvantages:**

* Cygwin is needed for Thrift compilation in Windows.

## Supported types

Types supported by Thrift: basic types, enumerations, structures, sequences of bytes and several containers (lists, maps, dictionaries). Also it is supported services (equal to interfaces in other middlewares), operations and exceptions.

**Advantages:**

* Thrift uses its own IDL, in opposition to WSDL.
* Set of simplified types.
* There are exceptions.
* The word *async* is used in de IDL.
* Thrift supports sparse structures.

## Programming model

Programming with Thrift consists in four steps:

1. Define types and services with Thrift IDL language.
2. Generates code in the desired programming language using the previous definitions.
3. Write the client application and compile it.
4. Write the server application and compile it.

An example of how to create a server:

shared\_ptr handler(new UserStorageHandler());

shared\_ptr processor(new UserStorageProcessor(handler));

shared\_ptr serverTransport(new TServerSocket(port));

shared\_ptr transportFactory(new TBufferedTransportFactory());

shared\_ptr protocolFactory(new TBinaryProtocolFactory());

TSimpleServer server(processor, serverTransport, transportFactory, protocolFactory);

**Disadvantages:**

* If user wants to make a change like change the transport, user has to rewrite the code and compile the application again. Thrift is not dynamic.

## Installation

In Windows there is a simple installer but it has to be compiled. Also in Linux Thrift has to be compiled and installed from the source code.

**Disadvantages:**

* Thrift always has to be compiled.
* Jaime had issues compiling the Windows version. The problem was some headers.

## External libraries

Thrift uses boost library’s header for smart pointers and for thread system. Its IDL compiler uses lex and yaac.

## Making an example

**Disadvantages:**

* Jaime had to apply two patches to make work the server.
* Ricardo had to include some header’s includes in code because the generated code doesn’t include them.

# Technical details

## Performance

In the next web page:

<http://www.eishay.com/2009/03/more-on-benchmarking-java-serialization.html>

Thrift has worst performance tan protocol buffers, but it beats others with better performance.

In a discussion a user points out that Thrift is good with little-size data, but its performance decreases with structures or complex types larger than 1Kb.

# Modelo de negocio

## License

Apache’s license.

## Project that use it

# Notes