服务器RPC机制protobuf

CareDear Service Team

November 4, 2014

Contents

Abstract

本文档介绍Google的服务器间RPC机制－protobuf

1 简介

这里暂时以Google的protobuf-2.5.0版本为基准，protobuf的完整代码包，可以在Caredear服务器的backend/3rd_party/protobuf-2.5.0/目录下找到。

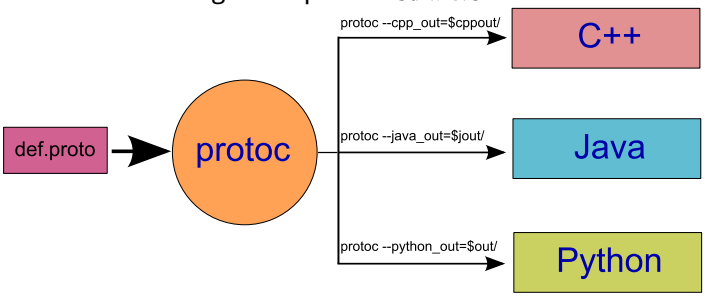
代码编译出的结果是一个编译解析器(protoc)，目前支持下面三种语言的绑定：

1. C++
2. Python
3. Java

Figure-1是protobuf处理流程，其编译器(protoc)将会把交互消息定义.proto文件(IDL)，按照用户指定语言绑定来自动生成相对应代码。

各个语言间可以无缝的交互数据，而不必再自己手动地编写代码去解析，或者组装数据包。

Figure 1: protoc的使用原理



比如Figure-2是一个更详细的例子，通过protoc生成的代码文件中，已经自动包含了数据的组包和解包的逻辑，用户不应该修改这些自动生成的源代码文件，而应该在自己的源文件中调用自动生成的这些代码接口。

从Figure-2可看出，当在.proto文件中定义完数据域后，protoc会在各自语言绑定中完成setXXX和getXXX接口与实现。

.proto文件中的package命令转换到C++代码将是namespace，而在Java代码中则就是Java程序的package。

2 Java的绑定

目前默认下，protoc只会生成Java代码，而Java运行需要的库，则要求使用Apache Maven包(mvn命令行程序)来编译。

在Ubuntu系统上，使用apt-get install mvn既可得到mvn程序，再使用下面命令：

```
protobuf/java $ mvn install
protobuf/java $ mvn package
```

这将会在java/目录下生成target/protobuf-java-2.5.0.jar包，Java语言程序编译和运行，都需要依赖该包。

和C++绑定不同，Java生成代码中的set()/get()并不是直接定义的，它是通过Builder()来间接实现的：

```
TokenMessage.TokenRequest.Builder b =
    TokenMessage.TokenRequest.newBuilder();
b.setUid("");
b.setAppid("");
...
TokenMessage.TokenRequest req = b.build();
```

命令行运行Java程序的方法：

```
$ java -classpath protobuf-java-2.5.0.jar:. Main
```

3 使用示例

在caredear服务器代码的backend/docs/samplecode/目录下，是一个演示Java前端和C++后端交换数据的流程。

Figure-2演示了两个模块间消息数据交换的定义，以及由TokenMessage.proto的IDL转换成各自语言绑定的过程。

由于protobuf只是负责数据封装和解析，如何传输并没有做具体要求。

所以对于跨机器情况下，我们通常可以将数据对象序列化后，转换成Byte类型，再加上leading length的字段，表明整个数据的有效长度。

这样接收终端会根据这个leading length知道整个数据长度，然后按需读取数据，在使用protobuf提供的接口进行数据解析。

Figure-3演示了两端数据交换时需要的一些代码调用。

4 IDL语法

目前protobuf支持的数据类型定义有：

.ptoro数据类型	C++类型	Java类型	说明
int32	int	int	根据实际大小变长实现
fixed32	int	int	定长的4字节大小
string	std::string	Java.lang.String	字符串
bool	bool	Boolean	布尔值

复杂.proto文件数据定义时，若需要引用其它的.proto文件，则可以使用import关键词：

```
import "other/other_def.proto"
```

5 关于本文档

介绍protobuf话题的文档置于backend/docs/目录下。

Figure-2和Figure-3描述的代码，位于backend/docs/samplecode/目录下。

References

[1] memcached: http://memcached.org, a high-performance cache system.

[2] Libevent: http://libevent.org, an open source event notification library.

[3] Protobuf: http://code.google.com/p/protobuf/, Google protobuf官方网址.

Figure 2: 基于protobuf协议的跨语言消息数据定义流程

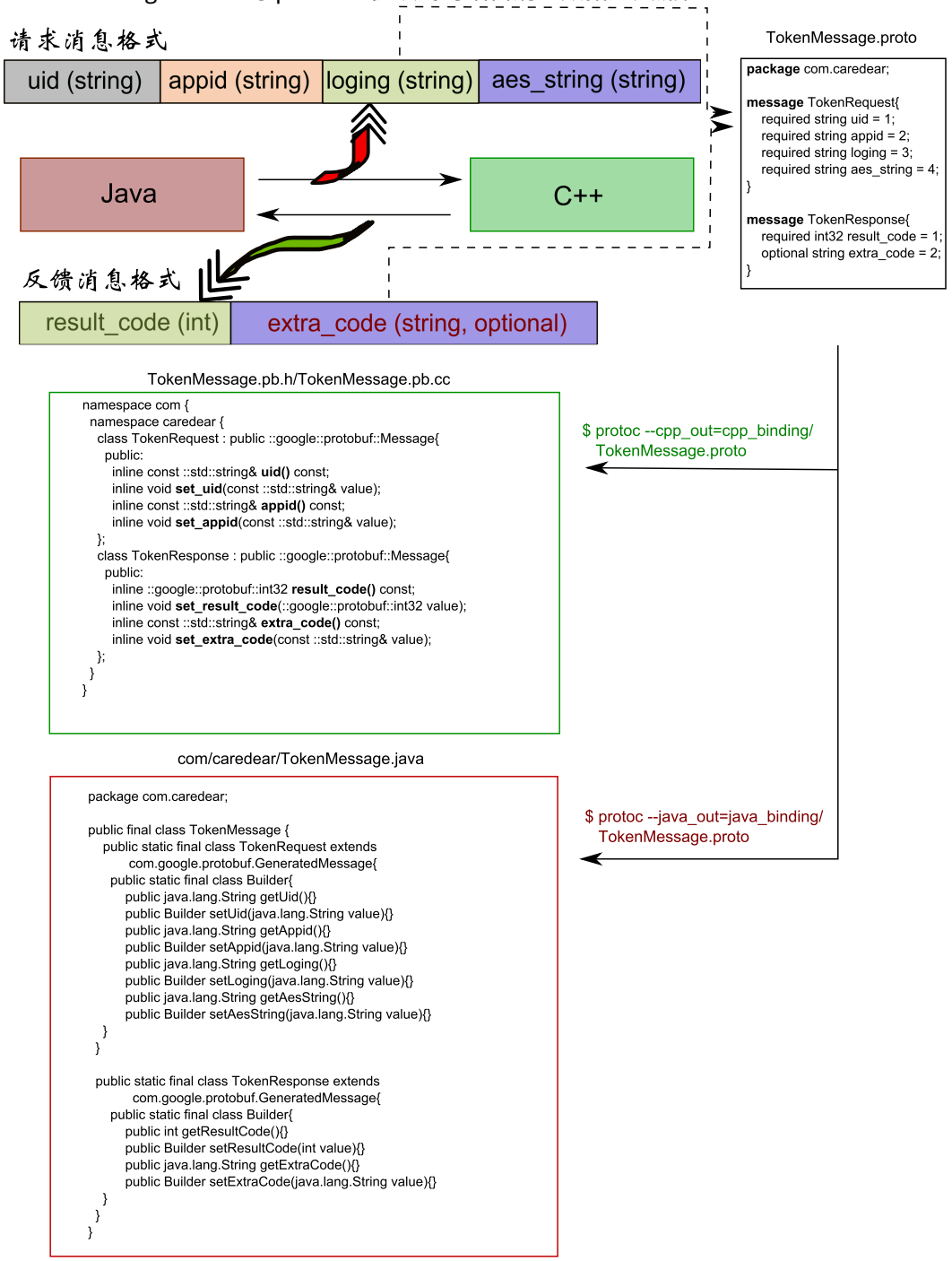


Figure 3: C++和Java数据交换代码示例