



Protocol Buffer 在 Android 企业云通讯录中的应用^{*}

殷 昊^{1,2}, 沈奇威^{1,2}, 王 纯^{1,2}

(1. 北京邮电大学网络与交换技术国家重点实验室 北京 100876;

2. 东信北邮信息技术有限公司 北京 100191)

摘要

本文介绍了在 Android 企业云通讯录中,应用 Protocol Buffer 进行数据更新的使用方法,并通过实验测试数据,对 XML(extensible markup language)、JSON(JavaScript object notation)、Protocol Buffer 进行时间和空间性能比较。

关键词 Protocol Buffer; Android; 性能

1 引言

2007 年,由 Google 公司推出的 Android 操作系统,由于具有良好的开放性和平台开发的便捷性等特点,迅速得到许多厂商的支持,同时也受到广大应用开发者的追捧,市场份额迅速攀升^[1],已跃居第一大手机操作系统。并且,每个成熟的企业都会维护一个员工通讯录,而让员工把企业通讯录添加到手机通讯录中会造成其通讯录超量,故而可查询、直接拨打、来电提示的 Android 企业通讯录应用便有了其实用价值,这些功能在 Android 应用中实现都比较容易。又因为员工离职、新员工加入、员工信息变化等,企业通讯录会经常发生变化,因此采用在云端维护数据、应用端更新的模式很为适合,服务器与终端之间更新的模式以及方式是整个应用的核心环节,需要满足传输数据量小、时间耗费短、安全性高等条件。

2 相关背景

目前大多数 Android 网络应用采用 XML(extensible markup language)或者 JSON(JavaScript object notation)等在传统互联网领域已经被广泛应用的数据通信格式,其优点是技术已经被大多数开发者掌握,学习成本低,开发时不需要额外安装编译器,并且明文方式方便扩展与调试。但是,这二者有着信息冗余度大、解析慢、明文传输等缺点,解决这些问题需要额外的压缩与解压缩、加密与解密等方式,增加工作量的同时,又使得编码、解析的过程更慢。而相对于 XML 和 JSON,Google 的另外一项技术 Protobuf(Protocol Buffer)有着数据量小、解析快、安全性较高等优点,这些优点都是移动互联网现在非常需要的特点,也是企业云通讯录进行同步更新环节所需要的特点。

Protobuf 是 Google 公司开发的一种数据描述语言,广泛地应用于 Google 内部,用于结构化数据的描述、传输和存储^[2],类似于 XML、JSON 这样的数据表示语言,其最大的特点是基于二进制^[3],因此比传统的数据表示语言 XML、

^{*} 国家自然科学基金资助项目 (No. 61072057, No. 61101119, No. 61121001, No. 60902051), 长江学者和创新团队发展计划基金资助项目 (No. IRT1049), 国家科技重大专项“移动互联网总体架构研究”基金资助项目 (No. 2011ZX03002-001-01)

JSON 高效、短小得多。虽然是二进制数据格式,但并没有因此而让使用变得复杂,开发人员通过按照一定的语法定义结构化的消息格式(.proto 文件),然后使用命令行工具中简单的命令语句,就可以自动生成相关的代码文件,可以支持 Java、C++、python 等语言环境,另外第三方扩展支持更多语言。通过将生成的代码文件包含在项目中,可以很轻松地调用相关方法来完成业务消息的编码与解析工作。Protobuf 库在效率、数据大小、易用性上有着很好的性能,故而能在 Google 公司得到广泛应用。Google 已经将这样一套框架开源发布,Protobuf 能够在更多的方面得到更广泛的应用,目前已经正在使用的有超过 48 162 种报文格式定义和超过 12 183 个 .proto 文件^[4]。Protobuf 的这些优点使得其不仅在服务器通信以及存储上有用武之地,同时在 Android 终端网络应用中,尤其是云端应用中,使用也非常适合,减小数据传输流量的同时又能提高解析效率。

3 Protobuf 在 Android 企业云通讯录中使用

利用时间戳的方式,在云端记录所有数据发生变化的时间戳,终端应用发起更新时,提交上次更新的时间戳,云端服务器便可提取出所有发生变化的数据,只要注意时间戳要统一使用系统的时间戳,这种变量更新的模式就不难实现。数据一旦更新到客户端应用中,在应用实现查找、呼叫、发邮件、来电监听也都不难实现。目前很多个人通讯录都有相关功能。整个企业云通讯录应用的核心在于如何将更新数据既快速又省流量地传输到应用终端,以给用户最好的使用体验。相对于目前普遍采用的 JSON 和 XML 数据格式,Protobuf 数据格式具有数据小、解析快的特点,将大大节省流量、节约同步耗时,同时,其非明文传输的特点,也一定程度上起到加密作用。下面介绍 Protobuf 在 Android 企业云通讯录中的应用方法。

3.1 编写结构化的消息格式(.proto 文件)

按照官方文档的编写规则以及员工的具体属性,编写数据消息格式文件(ebbook.proto)。

3.2 生成自动存取类

首先把前面编译生成的 protobuf-java-2.4.1.jar 和编写的 .proto 文件复制到同一个文件夹中,便于引用。使用 Protobuf 编译器编译 ebbook.proto 文件。打开命令行窗口,并定位到该文件夹中,运行命令 `protoc --java_out=. ebbook.proto`,执行这个命令后会在当前目录下根据

ebbook.proto 定义的分组名路径生成一个 java 文件 Ebbook.java,提供数据的编码、解码以及其他很多相关的能力。

3.3 Jar 分组引入

创建相关的工程,并将 protobuf-java-2.4.1.jar 引入到工程的引用库中。

3.4 生成的文件引入

将 Ebbook.java 文件添加到 Android 应用以及网络服务器端工程的代码文件中,注意分组名的一致。

3.5 使用 Protobuf 提供的编解码能力

在本例中,在服务器端实现对更新的员工信息的编码,而在终端应用中接收编码的信息并解析,然后添加到手机端的数据库中。

3.5.1 服务器端

服务器端,从数据库中查询所有手机端上次更新时间戳之后的发生变化的数据项,进行 Protobuf 方式的编码,具体代码如下。

```
ResultSet rs = st.executeQuery ("SELECT * FROM
ebstaff where Timestamp>" + timestamp);

List<EBer> ebers = new ArrayList<EBer>();
while (rs.next() ){
    EBer e = EBer.newBuilder( )
        .setName(rs.getString("Name"))
        .setUid(rs.getString("Uid"))
        .setMail(rs.getString("Mail"))
        .setTel(rs.getString("Tel"))
        .setMobile(rs.getString("Mobile"))
        //此处省略了其他属性
        .build( );
    ebers.add(e);
}

ebContact = EBContact.newBuilder( ).addAllEber(ebers).
build( );
```

从代码中可以看到,每一个员工的信息,会构建一个 EBer 对象,所有 EBer 对象汇总到一个链表中,然后将该链表编码到 EBContact 对象中。而 EBer 和 EBContact 类都是自动生成的 Ebbook 的内部类,实现 Protobuf 的编码能力非常简单,并不会给使用者带来麻烦。

生成的 ebContact 对象,采用如下方式发送到手机端。

```
response.setContentType("application/octet-stream");
```



```
response.setCharacterEncoding("utf-8");
response.setContentLength(ebContact.getSerializedSize( ));
ebContact.writeTo(response.getOutputStream( ));
response.flushBuffer( );
```

3.5.2 客户端

在客户端,对获取到的 ebContact 对象进行解析也很方便,所用到的能力,也都是自动生成的代码中提供的,具体代码如下。

```
EBContact ebcontact = EBContact.parseFrom(getData( ));
int length = ebcontact.getEberCount( );
persons = new Person[length];
for(int i=0;i<length;i++){
    persons[i] = Person.fromEBer(ebcontact.getEber(i));
}
```

其中的 getData() 方法可获取 HTTP(hyper text transport protocol,超文本传输协议)响应的内容,利用自动生成的代码提供的能力很容易就可以解析响应的内容。

在这个实例中,可以验证,Protobuf 的确可以在 Android 网络通信应用中使用,并且使用很方便,但是具体性能如何,还需要进一步比较测试。

4 Protobuf、XML、JSON 性能测试比较

在考虑一项技术的应用时,通常关注两点:使用的便捷性和使用的性能指标。通过上面的介绍,笔者发现,Google 已经为用户做了足够的工作,让 Protobuf 的使用者可以很方便地引用其能力来实现相关的功能。但是,相对于大多数开发者非常熟悉的 XML 和 JSON 而言,使用 Protobuf 需要付出一点学习成本。

性能方面,官方文档给出的之所以不只是用 XML 的理由为:Protobuf 更简单,文件大小比 XML 小 3~10 倍,快 20~100 倍,更少的歧义,可以方便地生成数据存取类^[5]。

为了更好地测试 Protobuf 的性能,笔者选择目前在 Android 网络应用开发中最常用的两种数据信息格式 XML、JSON 做对比,进行测试。在性能测试中,关注两点,一是数据大小,直接关系到数据传输所耗费的数据流量;二是从发起请求开始到结果呈现,整个过程中耗费的时间,这直接关系到用户的体验。

实验所用的工程,沿用上面描述的企业通讯录的例子,把 XML、JSON 方式的编解码能力也扩充到工程中,在手机端根据按键的不同,选择发送不同的请求,服务器根

据手机端应用的请求进行不同方式的编码,以 20 组员工信息作为数据进行时间与空间的测试。

4.1 空间测试

3 种方式编码结果的文件大小如图 1 所示。

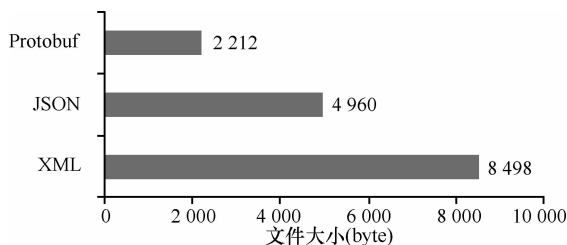


图 1 空间比较结果

从图 1 中可以看出,Protobuf 方式的文件大小接近 XML 方式文件大小的 1/4,符合官方文档里面小 3~10 倍的说法,JSON 较 XML 在文件大小方面也有明显优势,但是与 Protobuf 相比,还相差较多。同时,这也意味着,在 Android 客户端网络应用中使用 Protobuf 进行数据交互,将为用户节省将近 3/4 的流量。

4.2 时间测试

在客户端点击按键时,记录起始时间戳;在信息呈现出来时,记录结束时间戳,二者之差就是整体耗时,这段时间也是用户的等待时间,是影响用户体验的重要因素,因此时间性能非常重要,比空间性能更易影响用户体验。由于这段时间有多种因素影响,故而测试 50 组数据,求平均值来做比较。使用 galaxy nexus 手机通过 Wi-Fi 连接到与服务器同一局域网内,测试结果如图 2 所示;通过 EDGE (enhanced data rate for GSM evolution) 网络连接到公网服务器,测试结果如图 3 所示。

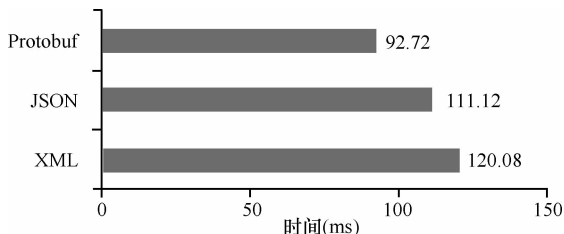


图 2 内网 Wi-Fi 条件下用户等待时间比较结果

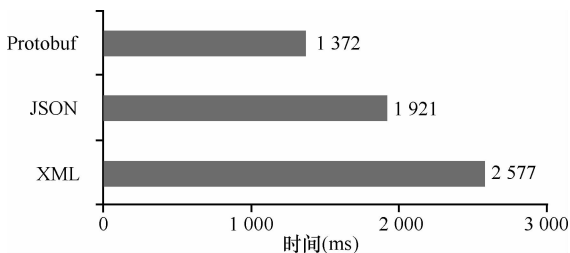


图 3 公网 EDGE 条件下用户等待时间比较结果

显然,Protobuf 的时间性能依然是最好的,其次是 JSON,最差的还是 XML,但是测试结果却没有官方文档里面表述的快 20~100 倍那么多,差距比较小,这是因为这个时间里包含着网络传输的时间,并且网络传输时间在其中占有相当大的比重。不过,笔者所关心的性能就是用户感知到的时间,而不是纯编码和解析时间上究竟相差多少倍。从中也不难看出,其实,不管是快多少倍,毕竟解析速度都是很快的,差距都是毫秒级的,也就是说,在解析时间上用户难以感知到差距。另外,在 EDGE 网络中,传输的速度相对局域网 Wi-Fi 环境下要慢很多,传输速度决定整个更新操作的用户感知时间,而 Protobuf 在空间上的优势,大大节约传输时间,时间性能优势更明显,故而用户体验更好。

另外,安全性也是一个很重要的指标,Protobuf 非明文传输,一定程度上,比直接使用 XML 和 JSON 安全性要高一些,但是,Protobuf 并不是非常安全的数据传输格式,如保密级别要求较高,仍需要进一步的加密措施。

5 结束语

Protobuf 作为一种数据信息格式,在 Google 公司内部已经得到广泛使用,其在空间性能、时间性能、易用

性、安全性上有着明显的优势,而这些性能也是 Android 企业云通讯录所需要的。经过测试比较,将 Protobuf 技术引入到 Android 企业云通讯录中,可以降低应用流量耗费、提高应用信息呈现速度,同时基本不会增加开发者的代码工作量。

参考文献

- 1 李林涛,石庆民. Android 智能手机操作系统的研究. 科技信息, 2011(25):80
- 2 宫唐小恒,李旭伟. Protocol Buffers——比 XML 快近 100 倍. 电脑与信息技术, 2009, 17(1):65~68
- 3 史栋杰. 五种快速序列化框架的性能比较. 电脑知识与技术, 2010, 6(34):9710~9722
- 4 Google Protocol Buffer 的使用和原理. <http://www.ibm.com/developerworks/cn/linux/l-cn-gpb/>, 2012
- 5 Protocol Buffer 官方主页. <http://code.google.com/intl/zh-CN/apis/protocolbuffers/docs/overview.html>, 2012
- 6 Jianhua Feng, Jinhong Li. Google Protocol Buffers research and application in online game. Proceedings of 2011 13th IEEE Joint International Computer Science and Information Technology Conference, Chongqing, China, 2011
- 7 杨丰盛. Android 应用开发揭秘. 北京:机械工业出版社, 2010
- 8 朱丹丹,王纯,郭盛兴. 基于 TD-SCDMA 电子相框的照片分享方案. 计算机系统应用, 2012, 21(5):12~16

Application of Protocol Buffer in Enterprise Cloud Contacts

Yin Hao^{1,2}, Shen Qiwei^{1,2}, Wang Chun^{1,2}

(1. State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; 2. EB Information Technology Co. Ltd., Beijing 100191, China)

Abstract This paper introduces the method to use Protocol Buffer to update data in enterprise cloud contacts, and compares the time and size of XML(extensible markup language), JSON(JavaScript object notation) and Protocol Buffer, through the test data.

Key words Protocol Buffer, Android, performance

(收稿日期:2012-08-06)