乐加二进制Json方案

1. 意义

鉴于乐加目前的主要技术构成，服务器采用webserver技术，php为主，客户端使用u3d。通讯方式采用http，json大量使用。

有几个问题需要解决

1. 流量过大，json结构构成控制字符占了很大比重，需要减少传输容量
2. 安全性，json可读性非常高，明码传输不可取，需要处理。
3. 客户端采用了大量json配置文件，配置文件安全性需要提高。
4. 分析

对于流量过大的问题：

我们有很多方案：

1. 套压缩算法（php端可以GZIP），我们建立了一些测试json测试了GZIP的压缩率，普遍在5%到50%之间，压缩率还是比较好的。
2. 套用hpack之类的json压缩方案

对于表达表格数据的json，hpack的压缩率很理想，他抽出表头，换了一种方式构成json。而且可逆。

对于树形的数据hpack的压缩基本没有作用。

1. 抛弃json，使用protobuf之类的方式

对于安全性的问题

1. 可以加密，rsa算法之类的引入即可
2. 可以使用protobuf之类的二进制存储的方式，很难观察。
3. 可以使用BSON，二进制版本的json，增加不可读程度

对于配置文件安全性的问题

1. 考虑到可读性，配置文件的加密必须使用易于和可读的json相互转换的方式，仅有BSON符合这个前提

经过分析后，现存容易获取的方案没有一个能满足我们的所有诉求。

整合各种方案的优点后，我们需要一个具有如下特征的方案：

1. 有压缩功能
2. 有一定安全功能，不可肉眼直读
3. 和JSON可以相互转换
4. 设计

鉴于此，我们设计了《乐加二进制json方案》

此方案兼容四种json类型

Object 即{}

Array 即[]

String 即””

Number（包括整数，最大支持32bit整数，浮点数，最大支持32bit浮点数，bool型，null值）

为考虑存储的容量做了两个设计。

1. 是设计了字符串字典，除了一个字符的字符串直接存储，其他的字符串都放在字典中，这样相同字符串可以只存储一次（复杂json的key值会大量重复）

并且支持公共字典和本地字典的分离。

公共字典可以独立保存，本地字典必须保存在最终的二进制格式中。

如果没有公共字典，则不能完整的恢复整个json

1. 是采用二进制存储，并且采用bit级的协议，尽量减小存储容量，比如在整数的存储上采用了类似utf8的编码方式，可以弹性的使用1~9个字节存储

为考虑安全用二进制编码，数字和结构不能直接阅读，字符串暂采用utf8明码存储。

此方案为JSON的二进制版本，同BSON，可以和JSON方便的转换，但是存储容量比BSON更小。

提供c#和php两个语言的实现，协议公开。

1. 协议说明

本协议存储为二进制数据，以c#的观点称为byte[],以php的观点。

一份数据由两个部分组成，本地字符串字典、json节点，依次编码。

1. 本地字符串字典

变长整数(字符串总数)

Loop（字符串总数）

{

变长整数（字符串长度（utf8编码后））

Utf8编码字符串…

}

1. Json节点

Json节点编码方式为一个Tag字符+其他

根据Json节点类型编码方式不同

Tag字符的前两位用来标示Json节点类型

Json\_Array和Json\_Object的第3位用来标示子节点个数是否超过32

不超过32则用tag的后五位来表示节点个数

超过32则用Tag的后五位表示后续整数的字节数，然后在Tag后读取这么多字节作为一个整数。然后开始编码子节点。

Json\_Array编码

Tag

定长整数(子节点总数，可选，如果tag已经有长度，不需要这个整数)

Loop（子节点总数）

{

编码Json节点

}

Json\_Object编码

Tag

定长整数(子节点总数，可选，如果tag已经有长度，不需要这个整数)

Loop（子节点总数）//编码所有的key

{

变长整数（字符串长度（utf8编码后））

Utf8编码字符串…

}

Loop（子节点总数）//编码所有的value

{

编码Json节点

}

Json\_String编码

Tag的第三位第四位 表示String是否在字典中，在公共字典还是本地字典中

如果不再字典中，tag的后三位表示字符串长度，字符串将被直接编码在Tag之后。

否则tag表示之后用多少Byte表示一个定长整数，该整数为字典的索引。

Json\_Number编码

Tag的第三位和第四位第五位被用来表示 是否浮点 是否null，是否bool

如果null 和bool则没有后续编码

Bool的时候，是否null作为bool的值

否则后三位表示之后用多少Byte表示一个定长数，浮点或者整数。

目前浮点总是用四个字节编码