FIT3140 Assignment 4

Report

Dekel Pilli & James Pickering

2017

Table of Contents

[The data formats 2](#_Toc494118830)

[JSON 2](#_Toc494118831)

[Introduction 2](#_Toc494118832)

[MessagePack 2](#_Toc494118833)

[Introduction 2](#_Toc494118834)

[Protocol Buffer 2](#_Toc494118835)

[Introduction 2](#_Toc494118836)

[Measurement process 2](#_Toc494118837)

[Results 2](#_Toc494118838)

[Spikes 2](#_Toc494118839)

[Conclusion 2](#_Toc494118840)

## The data formats

Our work consisted of two separate spikes, as described below.

## JSON

JSON, or Javascript Object Notation, is a potentially human-readable data format. The structure and syntax of its data is similar to that of hash maps in Python (and other languages), with potential for nested ‘hash maps’:

*{*

*fieldname: content,*

*fieldname2: {*

*subfieldname: content,*

*subfieldname2: content*

*},*

*fieldname3: content*

*}*

Despite being inefficient in terms of the storage space it requires, its human readability makes it a popular choice for data storage, including [Google’s Firebase](https://firebase.google.com/docs/database/web/structure-data#how_data_is_structured_its_a_json_tree).

## MessagePack

MessagePack uses JSON for its storage structure, but then encodes the JSON data to achieve smaller file sizes. The downside to this approach is the need to encode any data before showing it to users. So, while transferring and storing information becomes easier, displaying it becomes a more computationally complicated task.

## Protocol Buffer

Protocol Buffer is a data serialisation method developed by Google. While, like JSON, it is possible to store objects within objects, Protocol Buffer’s syntax for doing this is very different:

*message Object {*

*string content = 1;*

*int32 contentTwo = 2;*

*}*

*//A set of Objects*

*message Set {*

*repeated Object objects = 1;*

*}*

Unlike JSON, Protocol Buffer uses immutable types, reducing the flexibility of data entry. This means that the desired data structure needs to be defined before any data is entered. The downside of this is, of course, the additional setup time and reduced flexibility, but it enforces a consistent data structure. Additionally, once data has been entered, it is serialised before it is written, meaning the data itself is not human readable until it is decoded.

## Measurement process

To benchmark the efficiency of these protocols, we generated small (10,000 entries), medium (500,000), and large (1,000,000) version of each of these.

## Results

a

## Spikes

## Conclusion