Paper Review: Parsing Gigabytes of JSON per Second

Paper authored by Geoff Langdale and Daniel Lemire

Review by Ishuah Kariuki

Introduction

This is a peer reviewed paper that lays out the implementation of the Open-Source JSON parser, simdjson (https://github.com/simdjson/simdjson)

What makes simdjson special? It's blazing fast (4x faster than RapidJSON and 25x faster than JSON for modern C++)

JSON

- Specified by Douglas Crockford in early 2000s
- RFC 7159 by Tim Brays in 2013
- Internationally accepted format for exchanging data

JSON Parsing

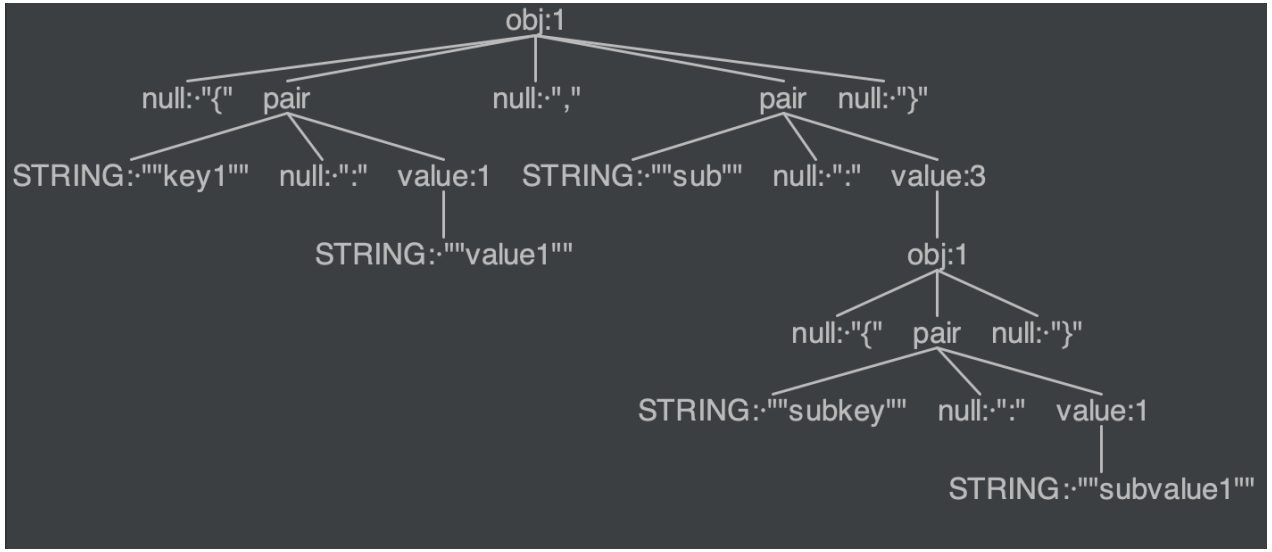
Transforming JSON text into a tree-like structure.

Steps:

- Read the whole document
- Validate JSON
- Check Unicode encoding
- Parse out numbers
- Build document-object-model

The Labor of Parsing JSON

Whenever the processor chooses between two code paths, there is a risk of incurring several cycles of penalty, due to a wrongly predicted branch on current pipelined processors.



Proposed solution

- Avoid hard-to-predict branches
- Take advantage of wide words
- Avoid memory/object allocation

Avoid hard-to-predict branches

Hard to predict branch

```
while (count != 0) {
  num = random();
  if ( num is odd) {
    output [index] = num;
    index += 1;
  }
  count--;
}
```

Go branchless

```
while (count != 0) {
  num = random();
  output [index] = num;
  index += (num bitand 1);
}
```

Wide words

- Avoid processing byte by byte and use SIMD when possible.
- SIMD is available on most commodity processors
- SIMD adds wider 128-bit, 256-bit and 512-bit registers.

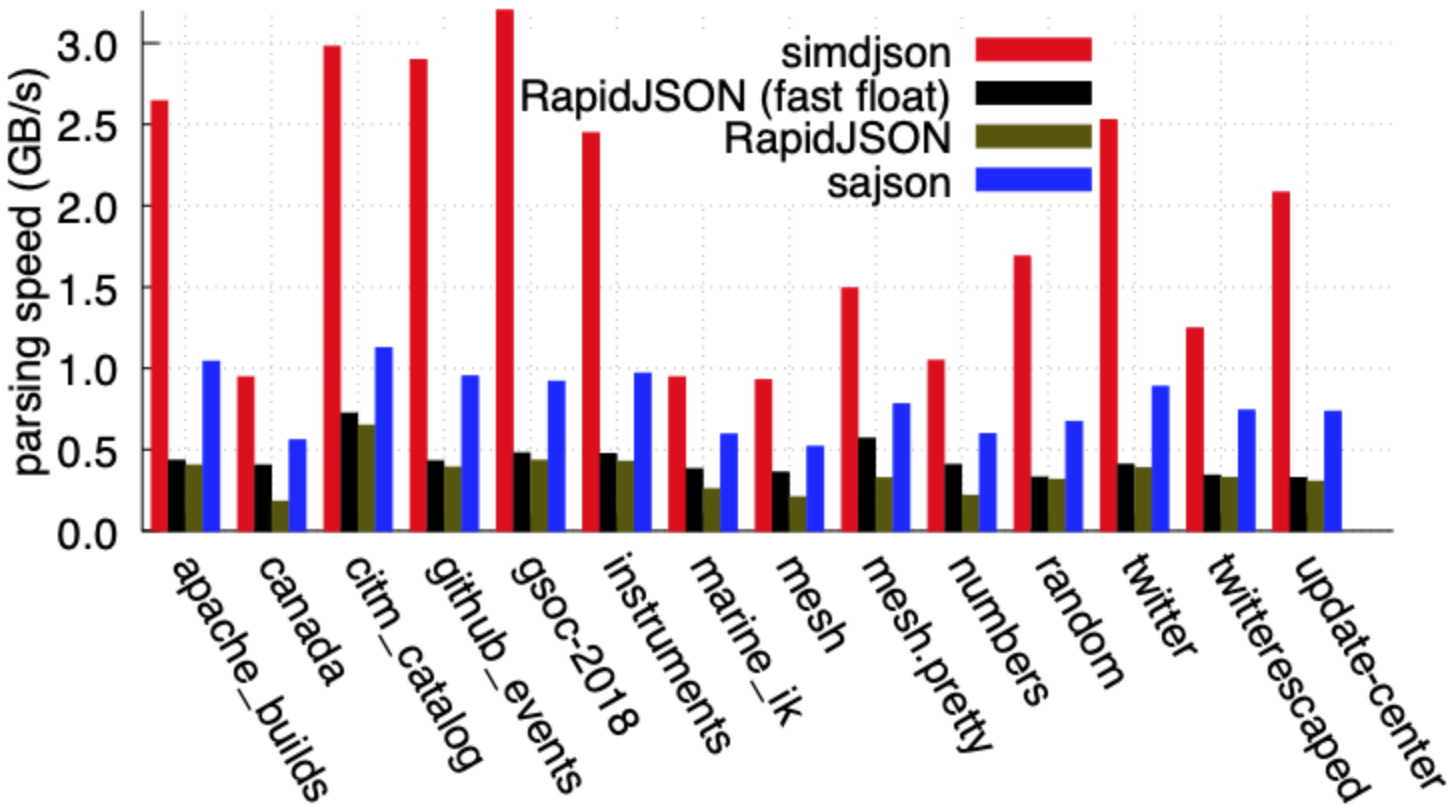
Avoid memory/object allocation

- Simdjson uses a goto-based state machine (the tape)

How fast is simdjson?

On a Skylake processor, the parsing speeds (in GB/s) of various processors on the twitter.json file are as follows.

parser	GB/s
simdjson	2.2
RapidJSON encoding-validation	0.51
RapidJSON encoding-validation, insitu	0.71
sajson (insitu, dynamic)	0.70
sajson (insitu, static)	0.97
dropbox	0.14
fastjson	0.26
gason	0.85
ultrajson	0.42
jsmn	0.28
cJSON	0.34


parsing speed (GB/s)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
simdjson
RapidJSON (fast float)
RapidJSON
sajson
apache_builds
canada
citm_catalog
github_events
gsoc-2018
instruments
marine_ik
mesh
mesh.pretty
numbers
random
twitter
twitterescaped
update-center

The End!