Subject: String should implement the Boyer Moore algo
Posted by victorb on Thu, 27 Feb 2014 15:05:05 GMT
View Forum Message <> Reply to Message

Hi all,

I used to use and contribute to U++ but it has been a long time since I haven't even started theIDE.

It's even more impressive that it was back then... Mirek, you're a genius!

I'm more developping web apps today and I have been looking at the
Boyer-Moore algo to speed up string search. It seems that U++ uses
a brute force algo, it should be able to do much better.

Please find some reference implementation below:
https://github.com/facebook/folly/blob/master/folly/FBString .h#L1796

http://doxygen.postgresql.org/varlena_8c_source.html

Keep up the good waork guys !

Subject: Re: String should implement the Boyer Moore algo
Posted by mirek on Fri, 28 Feb 2014 10:50:35 GMT
View Forum Message <> Reply to Message

Thing is, Find is meant for searching in relatively small Strings. IMO, in that case, the cost of precreating search tables in B-M algo outweights any possible gains. It also seams to me that for the same reason, such utility, which would definitely be useful in some cases, should be rather implemented as "search class" - because you can then "compile" the pattern once, then use for searching several times.

So for String::Find, let us consider the "brute force" a feature.

I would be nice to have BM as separate thing (class probably).

But another thing is that perhaps full PCRE has similar characteristics to BM but much more flexible use... (but I might be wrong about this one...)

Mirek

Subject: Re: String should implement the Boyer Moore algo
Posted by victorb on Fri, 28 Feb 2014 11:36:43 GMT

View Forum Message <> Reply to Message

Mirek,

The one implementation I've linked from FB is without pre-computations. It should be faster in any case - may be there should be a switch when looking one char with a tighter loop.

Folly (the lib from Facebook) is designed to be very fast, they claim a 30x speed increase for "casual cases" FWIW.

I'm really unsure if PCRE can perform as good as BM.

Victor

Subject: Re: String should implement the Boyer Moore algo
Posted by mirek on Fri, 28 Feb 2014 12:54:19 GMT
View Forum Message <> Reply to Message

Well, but that code is definitely not Boyer-Moore, but " Boyer-Moore-like trick " (it is in the comments

Albeit it looks like quite a clever trick, fixing the most profound cases where brute force fails.

Mirek

Subject: Re: String should implement the Boyer Moore algo
Posted by mirek on Sat, 01 Mar 2014 16:05:34 GMT
View Forum Message <> Reply to Message

After further analysing folly code, I am quite confused by where that 30x times speedup claim comes from.

If I can see the code right, the about only situation where any speedup compared to brute force happens is when the last character of string is matched, which is perhaps important for some corner cases, but for general case it is virtually useless...

(In contrast, _proper_ Boyer-Moore skips almost always, but well, that requires prebuilding those skip tables...)

But perhaps I am missing something?

Mirek

Subject: Re: String should implement the Boyer Moore algo

Well, I could not sleep about this, so I have decided to put it to test

I have created "proper" BM and BMH algorithm, then started benchamrking. Loaded 120MB XML file into String, appended "Hello world" (not present in the file otherwise) and started benchmarking.

At first, U++ brute force was 10 times slower than others. Analyzing it I have found that it is not particulary well optimized (as brute force), calling memcmp on each input byte. So I have added some microoptimizations and this is what I got then:

```
------------
Needle: Hello world
U++ Brute force: 127158988
  Time elapsed: 0.037
Folly: 127158988
  Time elapsed: 0.059
Boyer-Moore: 127158988
  Time elapsed: 0.082

  Time elapsed: 0.049
------------
Needle: Hel
U++ Brute force: 127158988
  Time elapsed: 0.036
Folly: 127158988
  Time elapsed: 0.063
Boyer-Moore: 127158988
  Time elapsed: 0.270

  Time elapsed: 0.154
```

First numbers are for the whole "Hello world" search, then i have searched only for "Hel", which accidentally is not in the file too.

I believe that the only real speed advantage the folly algorithm has lies in that simple loop

```
    while (i[nsize_1] != lastNeedle) {
      if (++i == iEnd) {
        // not found
        return -1;
      }
    }
```

implementing de-facto brutal force approach. Any "smart" skips are not frequent enough to change anything.

As for BM and BMH, it looks like the management of skips in reality and in modern CPU is too costly as compared to streamlined comparison loops.... and becomes really a burden when needle is short. Also interesting is that simpler, less clever variant BMH (BMH is in fact simplified BM) is faster... I guess complexity in the loop shows.

Well, at any case, the real result of this endeavour is optimized, albeit still brute-force, String::Find - IMO not bad at all

Mirek

Subject: Re: String should implement the Boyer Moore algo
Posted by zsolt on Sun, 02 Mar 2014 14:36:57 GMT
View Forum Message <> Reply to Message

Thanks

Subject: Re: String should implement the Boyer Moore algo
Posted by victorb on Sun, 02 Mar 2014 16:32:52 GMT
View Forum Message <> Reply to Message

Mirek,

Great work !

Subject: Re: String should implement the Boyer Moore algo
Posted by nlneilson on Sun, 02 Mar 2014 18:17:15 GMT
View Forum Message <> Reply to Message

Great!

It seemed to be fast enough before but should be even faster now.

Adding the option to not close the search window as often was great also.

Subject: Re: String should implement the Boyer Moore algo
Posted by mirek on Sun, 02 Mar 2014 18:22:41 GMT

Frankly, I am still wondering why "smart" Boyer-Moore is slower. It is possible that I have faulty implementation (I just adopted the code from wikipedia).

Anyone wishing to play with this is encouraged to try benchmarks/StringFind.

Mirek