Ruby master - Misc #15800

Reduce ONIG_NREGION from 10 to 4: power of 2 and testing revealed most pattern matches are less than or equal to 4 results

04/27/2019 01:00 PM - methodmissing (Lourens Naudé)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
</tbody>
</table>

Description

References PR [https://github.com/ruby/ruby/pull/2135](https://github.com/ruby/ruby/pull/2135) - it's a very small change, but runnin due diligence past the list too for discussion.

I noticed onig_region_resize (called from onig_region_copy) would default to allocating a 10 * 8 bytes block on 64bit for both the beg and end members of OnigRegion.

Preliminary testing with Rails and the benchmark suite suggests that most pattern matches are <= 4 results.

Due diligence with debug counters

Few requests on a blank redmine instance:

```
[RUBY_DEBUG_COUNTER]    obj_match_under4   10650  <<<<<<<<<<<<
[RUBY_DEBUG_COUNTER]    obj_match_ge4     1589   <<<<<<<<<<<<
[RUBY_DEBUG_COUNTER]    obj_match_ge8     66     <<<<<<<<<<<<
[RUBY_DEBUG_COUNTER]    obj_match_ptr    12305  <<<<<<<<<<<<
```

single match 1000000.times { 'haystack'.match(/hay/) }:

```
[RUBY_DEBUG_COUNTER]    obj_match_under4   999366 <<<<<<<<<<<<
[RUBY_DEBUG_COUNTER]    obj_match_ge4     473    <<<<<<<<<<<<
[RUBY_DEBUG_COUNTER]    obj_match_ge8     0      <<<<<<<<<<<<
[RUBY_DEBUG_COUNTER]    obj_match_ptr    999839            
```

multiple matches > 4 1000000.times { /(.)(.)(\d+)(\d)/.match("THX1138.") }:

```
[RUBY_DEBUG_COUNTER]    obj_match_under4   353    <<<<<<<<<<<<
[RUBY_DEBUG_COUNTER]    obj_match_ge4     997579 <<<<<<<<<<<<
[RUBY_DEBUG_COUNTER]    obj_match_ge8     0      <<<<<<<<<<<<
[RUBY_DEBUG_COUNTER]    obj_match_ptr    997932            
```

Memory and ips benchmarks, MatchData specific

```
```

```
compare-ruby: ruby 2.7.0dev (2019-04-19 trunk 67619) [x86_64-linux]
built-ruby: ruby 2.7.0dev (2019-04-19 reduce-onig-de.. 67619) [x86_64-linux]
```

```
last_commit=Reduce ONIG_NREGION from 10 to 4: power of 2 and testing revealed most pattern matches are <= 4 results
Calculating -------------------------------------
```
```
<table>
<thead>
<tr>
<th></th>
<th>compare-ruby</th>
<th>built-ruby</th>
</tr>
</thead>
<tbody>
<tr>
<td>match_gt4</td>
<td>11.936M</td>
<td>11.600M</td>
</tr>
<tr>
<td>match_small</td>
<td>11.848M</td>
<td>11.608M</td>
</tr>
</tbody>
</table>
```

Comparison:

```
<table>
<thead>
<tr>
<th></th>
<th>compare-ruby</th>
<th>built-ruby</th>
</tr>
</thead>
<tbody>
<tr>
<td>match_gt4</td>
<td>11936000.0 bytes</td>
<td>11600000.0 bytes</td>
</tr>
</tbody>
</table>
```

```
match_small
built-ruby: 11608000.0 bytes
```

03/19/2022
compare-ruby: 11848000.0 bytes - 1.02x larger


compare-ruby: ruby 2.7.0dev (2019-04-19 trunk 67619) [x86_64-linux]
built-ruby: ruby 2.7.0dev (2019-04-19 reduce-onig-de.. 67619) [x86_64-linux]

I am fine with removing the debug counters and committed them for now as it's easier for reviewers to also reproduce locally.

For additional context I noticed that character offsets are bounded by the num_regs member as per https://github.com/ruby/ruby/blob/trunk/re.c#L989-L1005 and therefore investigated converging allocated and num_regs to be less divergent for the common cases

And some more of the 80 byte allocs from strscan with only the first chunk referenced:

```
==24182==  283 of 1000
==24182==  max-live: 19,520 in 244 blocks
==24182==  tot-alloc: 30,480 in 381 blocks (avg size 80.00)
==24182==  deaths: 381, at avg age 423,950,747 (3.96% of prog lifetime)
==24182==  acc-ratios: 1.95 rd, 4.98 wr (59,728 b-read, 151,920 b-written)
==24182==  at 0x4C2DECF: malloc (in /usr/lib/valgrind/vgpreload_exp-dhat-amd64-linux.so)
==24182==  by 0x2561E6: onig_region_resize (regexec.c:260)
==24182==  by 0x2561E6: onig_region_resize_clear (regexec.c:298)
==24182==  by 0x2561E6: onig_match (regexec.c:3882)
==24182==  by 0xA4C376B: strscan_do_scan (strscan.c:472)
==24182==  by 0xA4C376B: strscan_skip (strscan.c:570)
==24182==  by 0xE2E5B4E: vm_call_cfunc_with_frame (vm_insnshelper.c:2207)
==24182==  by 0xE2E5B4E: vm_call_cfunc (vm_insnshelper.c:2225)
==24182==
==24182== Aggregated access counts by offset:
==24182==
==24182==   [ 0] 26456 26456 26456 26456 26456 26456 26456 26456 0 0 0 0 0 0 0 0
==24182==   [ 16] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
==24182==   [ 32] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
==24182==   [ 48] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
==24182==   [ 64] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

History

#1 - 05/11/2019 01:42 PM - nobu (Nobuyoshi Nakada)
- Status changed from Open to Closed

#2 - 07/29/2019 06:28 AM - ko1 (Koichi Sasada)

https://github.com/ruby/ruby/commit/a47f598d77ac97f9fe89fe16aa8bcab4fd262c16