

Ruby trunk - Feature #13896

Find.find -> Use Dir.children instead of Dir.entries

09/14/2017 06:32 AM - esparta (Espartaco Palma)

Status:	Closed
Priority:	Normal
Assignee:	
Target version:	

Description

Dir.children is available since Feature [#11302](#). Find.find can use of the new list (having no '.' neither '..' entries), making now superflous an if statement.

This change can improve the performance of Find.find when the path has lots of entries (thousands?).

Some profiling I did using 50,000 files on a given folder, using this code:

```
total_size = 0

Find.find(ENV["HOME"]) do |path|
  if FileTest.directory?(path)
    if File.basename(path)[0] == ?.
      Find.prune # Don't look any further into this directory.
    else
      next
    end
  else
    total_size += FileTest.size(path)
  end
end
```

Before the patch

```
~/ruby -rprofile before.rb
% cumulative self self total
time seconds seconds calls ms/call ms/call name
48.37 9.24 9.24 100014 0.09 0.60 Find.find
13.52 11.82 2.58 50005 0.05 0.07 nil#
5.17 12.81 0.99 50005 0.02 0.35 Kernel#catch
5.05 13.77 0.96 50006 0.02 0.04 Kernel#dup
4.96 14.72 0.95 50006 0.02 0.02 Kernel#initialize_dup
3.87 15.46 0.74 1 738.89 5952.54 Array#reverse_each
2.39 15.92 0.46 100012 0.00 0.00 String#==
1.98 16.29 0.38 50004 0.01 0.01 File.join
1.93 16.66 0.37 50004 0.01 0.01 FileTest.size
1.87 17.02 0.36 50005 0.01 0.01 File.lstat
1.76 17.36 0.34 50005 0.01 0.01 FileTest.directory?
1.34 17.61 0.26 50004 0.01 0.01 Integer#+
1.30 17.86 0.25 50005 0.00 0.00 File::Stat#directory?
1.29 18.11 0.25 50006 0.00 0.00 String#initialize_copy
1.29 18.35 0.25 50004 0.00 0.00 Array#unshift
1.28 18.60 0.24 50006 0.00 0.00 Array#shift
1.25 18.84 0.24 50004 0.00 0.00 Kernel#untaint
1.25 19.08 0.24 50005 0.00 0.00 Kernel#taint
0.10 19.09 0.02 1 19.53 19.55 Dir.entries
0.02 19.10 0.00 1 4.10 4.10 Array#sort!
0.00 19.10 0.00 2 0.47 1.12 Kernel#require
0.00 19.10 0.00 4 0.02 0.04 Gem.find_unresolved_default_spec
0.00 19.10 0.00 2 0.04 9549.52 Array#each
0.00 19.10 0.00 1 0.05 0.07 MonitorMixin#mon_enter
0.00 19.10 0.00 1 0.04 0.07 MonitorMixin#mon_exit
```

0.00	19.10	0.00	1	0.04	0.05	Module#module_function
0.00	19.10	0.00	4	0.01	0.01	IO#set_encoding
0.00	19.10	0.00	1	0.02	0.02	Dir.open
0.00	19.10	0.00	1	0.02	0.11	Array#collect!
0.00	19.10	0.00	1	0.02	0.02	MonitorMixin#mon_check_owner
0.00	19.10	0.00	2	0.01	0.01	Module#method_added
0.00	19.10	0.00	3	0.00	0.00	Thread.current
0.00	19.10	0.00	2	0.01	0.01	String#encoding
0.00	19.10	0.00	2	0.01	0.01	BasicObject#singleton_method_added
0.00	19.10	0.00	2	0.00	0.00	Kernel#respond_to?
0.00	19.10	0.00	1	0.01	0.01	TracePoint#enable
0.00	19.10	0.00	1	0.01	0.01	Gem::Specification.unresolved_deps
0.00	19.10	0.00	1	0.01	0.01	File.exist?
0.00	19.10	0.00	1	0.01	0.01	File.basename
0.00	19.10	0.00	1	0.01	0.01	Thread::Mutex#lock
0.00	19.10	0.00	1	0.01	0.01	String#[]
0.00	19.10	0.00	1	0.01	0.01	Gem.suffixes
0.00	19.10	0.00	1	0.01	0.01	Thread::Mutex#unlock
0.00	19.10	0.00	1	0.01	0.01	Encoding.find
0.00	19.10	0.00	1	0.01	0.01	BasicObject#==
0.00	19.10	0.00	1	0.00	0.00	TracePoint#disable
0.00	19.10	0.00	1	0.00	0.00	Kernel#block_given?
0.00	19.10	0.00	1	0.00	19100.95	#toplevel

After the patch

%	cumulative	self	self	self	total	name
time	seconds	seconds	calls	ms/call	ms/call	
45.15	7.70	7.70	100012	0.08	0.52	Find.find
15.12	10.27	2.58	50005	0.05	0.07	nil#
5.76	11.25	0.98	50005	0.02	0.31	Kernel#catch
5.66	12.22	0.96	50006	0.02	0.04	Kernel#dup
5.59	13.17	0.95	50006	0.02	0.02	Kernel#initialize_dup
4.30	13.91	0.73	1	733.77	3952.79	Array#reverse_each
2.12	14.27	0.36	50005	0.01	0.01	File.lstat
2.08	14.62	0.35	50004	0.01	0.01	FileTest.size
2.06	14.97	0.35	50004	0.01	0.01	File.join
1.93	15.30	0.33	50005	0.01	0.01	FileTest.directory?
1.47	15.55	0.25	50005	0.00	0.00	File::Stat#directory?
1.46	15.80	0.25	50006	0.00	0.00	String#initialize_copy
1.46	16.05	0.25	50004	0.00	0.00	Integer##
1.44	16.30	0.25	50004	0.00	0.00	Array#unshift
1.44	16.54	0.24	50006	0.00	0.00	Array#shift
1.41	16.78	0.24	50004	0.00	0.00	Kernel#untaint
1.40	17.02	0.24	50005	0.00	0.00	Kernel#taint
0.11	17.04	0.02	1	19.16	19.18	Dir.children
0.02	17.04	0.00	1	4.14	4.14	Array#sort!
0.00	17.04	0.00	1	0.61	0.69	Kernel#require_relative
0.00	17.04	0.00	1	0.04	0.05	Module#module_function
0.00	17.04	0.00	4	0.01	0.01	IO#set_encoding
0.00	17.04	0.00	1	0.02	17043.77	Array#each
0.00	17.04	0.00	1	0.02	0.11	Array#collect!
0.00	17.04	0.00	1	0.02	0.02	Dir.open
0.00	17.04	0.00	2	0.01	0.01	Module#method_added
0.00	17.04	0.00	1	0.01	0.01	TracePoint#enable
0.00	17.04	0.00	2	0.00	0.00	String#encoding
0.00	17.04	0.00	2	0.00	0.00	BasicObject#singleton_method_added
0.00	17.04	0.00	1	0.01	0.01	File.basename
0.00	17.04	0.00	1	0.01	0.01	File.exist?
0.00	17.04	0.00	1	0.01	0.01	String#[]
0.00	17.04	0.00	1	0.01	0.01	Encoding.find
0.00	17.04	0.00	1	0.01	0.01	String#==
0.00	17.04	0.00	1	0.01	0.01	BasicObject#==
0.00	17.04	0.00	1	0.00	0.00	TracePoint#disable
0.00	17.04	0.00	1	0.00	0.00	Kernel#block_given?
0.00	17.04	0.00	1	0.00	0.00	Kernel#respond_to?
0.00	17.05	0.00	1	0.00	17045.11	#toplevel

Associated revisions

Revision b2996b30 - 09/15/2017 05:00 PM - naruse (Yui NARUSE)

Find.find -> Use Dir.children instead of Dir.entries

Dir.children is available since Feature #11302.
Find.find can use of the new list (having no '.' neither '..' entries),
making now superflous an if statement.

This change can improve the performance of Find.find when the path
has lots of entries (thousands?).

<https://bugs.ruby-lang.org/issues/11302>
patched by Espartaco Palma esparta@gmail.com
<https://github.com/ruby/ruby/pull/1697> fix GH-1697
[Feature #13896]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@59926 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 59926 - 09/15/2017 05:00 PM - naruse (Yui NARUSE)

Find.find -> Use Dir.children instead of Dir.entries

Dir.children is available since Feature #11302.
Find.find can use of the new list (having no '.' neither '..' entries),
making now superflous an if statement.

This change can improve the performance of Find.find when the path
has lots of entries (thousands?).

<https://bugs.ruby-lang.org/issues/11302>
patched by Espartaco Palma esparta@gmail.com
<https://github.com/ruby/ruby/pull/1697> fix GH-1697
[Feature #13896]

Revision 59926 - 09/15/2017 05:00 PM - naruse (Yui NARUSE)

Find.find -> Use Dir.children instead of Dir.entries

Dir.children is available since Feature #11302.
Find.find can use of the new list (having no '.' neither '..' entries),
making now superflous an if statement.

This change can improve the performance of Find.find when the path
has lots of entries (thousands?).

<https://bugs.ruby-lang.org/issues/11302>
patched by Espartaco Palma esparta@gmail.com
<https://github.com/ruby/ruby/pull/1697> fix GH-1697
[Feature #13896]

Revision 59926 - 09/15/2017 05:00 PM - naruse (Yui NARUSE)

Find.find -> Use Dir.children instead of Dir.entries

Dir.children is available since Feature #11302.
Find.find can use of the new list (having no '.' neither '..' entries),
making now superflous an if statement.

This change can improve the performance of Find.find when the path
has lots of entries (thousands?).

<https://bugs.ruby-lang.org/issues/11302>
patched by Espartaco Palma esparta@gmail.com
<https://github.com/ruby/ruby/pull/1697> fix GH-1697
[Feature #13896]

Revision 37c08fad - 11/07/2018 03:55 PM - hsbt (Hiroshi SHIBATA)

Dir.children is available since Feature #11302. FileUtils uses
Dir.each on an internal method encapsulated on a private class
Entry_#entry, having no '.' neither '..' entries would make

now superfluous a chained reject filtering.

This change can improve the performance of these FileUtils methods when the provided path covers thousands of files or directories:

- chmod_R
- chown_R
- remove_entry
- remove_entry_secure
- rm_r
- remove_dir
- copy_entry

Related: Feature #13896 <https://bugs.ruby-lang.org/issues/13896>

[Feature #14109][Fix GH-1754]

Co-Authored-By: esparta esparta@gmail.com

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@65610 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 65610 - 11/07/2018 03:55 PM - hsbt (Hiroshi SHIBATA)

Dir.children is available since Feature #11302. FileUtils uses Dir.each on an internal method encapsulated on a private class Entry_#entry, having no '.' neither '..' entries would make now superfluous a chained reject filtering.

This change can improve the performance of these FileUtils methods when the provided path covers thousands of files or directories:

- chmod_R
- chown_R
- remove_entry
- remove_entry_secure
- rm_r
- remove_dir
- copy_entry

Related: Feature #13896 <https://bugs.ruby-lang.org/issues/13896>

[Feature #14109][Fix GH-1754]

Co-Authored-By: esparta esparta@gmail.com

Revision 65610 - 11/07/2018 03:55 PM - hsbt (Hiroshi SHIBATA)

Dir.children is available since Feature #11302. FileUtils uses Dir.each on an internal method encapsulated on a private class Entry_#entry, having no '.' neither '..' entries would make now superfluous a chained reject filtering.

This change can improve the performance of these FileUtils methods when the provided path covers thousands of files or directories:

- chmod_R
- chown_R
- remove_entry
- remove_entry_secure
- rm_r
- remove_dir
- copy_entry

Related: Feature #13896 <https://bugs.ruby-lang.org/issues/13896>

[Feature #14109][Fix GH-1754]

Co-Authored-By: esparta esparta@gmail.com

History

#1 - 09/14/2017 06:33 AM - esparta (Espartaco Palma)

- Backport deleted (2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN)

- Tracker changed from Bug to Misc

#2 - 09/15/2017 04:58 PM - naruse (Yui NARUSE)

- Tracker changed from Misc to Feature

#3 - 09/15/2017 05:00 PM - naruse (Yui NARUSE)

- Status changed from Open to Closed

Applied in changeset [trunk|r59926](#).

Find.find -> Use Dir.children instead of Dir.entries

Dir.children is available since Feature [#11302](#).

Find.find can use of the new list (having no '.' neither '..' entries), making now superflous an if statement.

This change can improve the performance of Find.find when the path has lots of entries (thousands?).

<https://bugs.ruby-lang.org/issues/11302>

patched by Espartaco Palma esparta@gmail.com

<https://github.com/ruby/ruby/pull/1697> fix GH-1697

[Feature [#13896](#)]