

## Ruby master - Feature #13873

### Optimize Dir.glob with FNM\_EXTGLOB

09/05/2017 05:51 PM - naruse (Yui NARUSE)

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Target version:</b>	
<b>Description</b>	
Rails often queries Dir.glob with complex pattern.	
On accessing Rails site, it calls glob as follows on querying templates. Note that Dir[] insists File::FNM_EXTGLOB.	
<pre>def find_template_paths(query)   Dir[query].uniq.reject do  filename      File.directory?(filename)          # deals with case-insensitive file systems.       !File.fnmatch(query, filename, File::FNM_EXTGLOB)   end end</pre>	
<a href="https://github.com/rails/rails/blob/6f1c18308ebffc97d51440cdeed7be71de58f26a/actionview/lib/action_view/template/resolver.rb#L247">https://github.com/rails/rails/blob/6f1c18308ebffc97d51440cdeed7be71de58f26a/actionview/lib/action_view/template/resolver.rb#L247</a>	
For example on accessing index page of Post Model (/posts), it calls following glob: Dir["/home/naruse/foo/app/views/posts/index{.en,}.{.html,.text,.js,.css,.ics,.csv,.vcf,.png,.jpeg,.gif,.bmp,.tiff,.svg,.mpeg,.xml,.rss,.atom,.yaml,.multipart_form,.url_encoded_form,.json,.pdf,.zip,.gzip,}{{.raw,.erb,.html,.builder,.ruby,.coffee,.jbuilder,}}"]	
It calls many stat as follows:	
<pre>% sudo strace ./miniruby58882 -e'Dir["/home/naruse/foo/app/views/posts/index{.en,}.{.html,.text,.js,.css,.ics,.csv,.vcf,.png,.jpeg,.gif,.bmp,.tiff,.svg,.mpeg,.xml,.rss,.atom,.yaml,.multipart_form,.url_encoded_form,.json,.pdf,.zip,.gzip,}{{.raw,.erb,.html,.builder,.ruby,.coffee,.jbuilder,}}"]' &amp; grep newfstatat wc -l 400</pre>	
Because current glob implementation handles braces in early stage, then calls stat for each combination of brace patterns instead of using readdir results.	
Below is a patch to use readdir instead of hundreds of syscalls. Note that this patch changes the order of the result as follows:	
1) Failure: TestDir#test_glob [/home/naruse/ruby/test/ruby/test_dir.rb:157]: <["/tmp/test-all/___test_dir__20170906-8620-ka8yqu/}"}", "/tmp/test-all/___test_dir__20170906-8620-ka8yqu/}a"]> expected but was <["/tmp/test-all/___test_dir__20170906-8620-ka8yqu/}a", "/tmp/test-all/___test_dir__20170906-8620-ka8yqu/}"}"]>.	
diff --git a/dir.c b/dir.c index b7afaec4e0..7ca84fa8c5 100644 --- a/dir.c +++ b/dir.c @@ -290,6 +290,8 @@ bracket( #define UNESCAPE(p) (escape && *(p) == '\\\ ' ? (p) + 1 : (p)) #define ISEND(p) (!*(p)    (pathname && *(p) == '/')) #define RETURN(val) return *pcur = p, *scur = s, (val); +#define FNMATCH_ALLOC_N(type, n) ((type *)malloc(sizeof(type) * (n))) +#define FNMATCH_FREE(ptr) free(ptr)  static int fnmatch_helper( 	

```

@@ -348,6 +350,56 @@ fnmatch_helper(
    }
    goto failed;
}
+
+
+ case '{': {
+     size_t len = pend - p;
+     char *buf = FNMATCH_ALLOC_N(char, len);
+     const char *rbrace = NULL;
+     while (p < pend) {
+         const char *t = ++p;
+         int nest = 0;
+         while (p < pend && !(*p == ',' && nest == 0)) {
+             if (*p == '{') nest++;
+             if (*p == '}') {
+                 if (nest == 0) {
+                     if (!rbrace) rbrace = p;
+                     goto rest;
+                 }
+                 nest--;
+             }
+             if (*p == '\\') && escape) {
+                 if (++p >= pend) break;
+             }
+             Inc(p, pend, enc);
+         }
+         if (!rbrace) {
+             rbrace = p;
+             while (rbrace < pend && !(*rbrace == '}') && nest == 0) {
+                 if (*rbrace == '{') nest++;
+                 if (*rbrace == '}') nest--;
+                 if (*rbrace == '\\') && escape) {
+                     if (++p >= pend) break;
+                 }
+                 Inc(rbrace, pend, enc);
+             }
+         }
+rest:
+     memcpy(buf, t, p-t);
+     buf[p-t]=0;
+     strncpy(buf+(p-t), rbrace+1, len-(p-t));
+     {
+         const char *pp = buf, *ss = s;
+         r = fnmatch_helper((const char **)&pp, &ss, flags|FNM_DOTMATCH, enc);
+     }
+     if (r == 0) {
+         p = buf;
+         FNMATCH_FREE(buf);
+         RETURN(0);
+     }
+     if (p >= rbrace) break;
+     }
+     FNMATCH_FREE(buf);
+ }
+
+
+ /* ordinary */
@@ -1426,6 +1478,12 @@ has_magic(const char *p, const char *pend, int flags, rb_encoding *enc)
    case '[':
        return MAGICAL;

+     case '{':
+         if (flags & FNM_EXTGLOB) {
+             return MAGICAL;
+         }
+         break;
+

```

```

    case '\\':
        if (escape && p++ >= pend)
            continue;
@@ -2272,6 +2330,13 @@ push_pattern(const char *path, VALUE ary, void *enc)
    rb_ary_push(ary, name);
}

+struct push_glob_args {
+    struct glob_args glob;
+    int flags;
+    int fd;
+};
+static int push_caller(const char *path, VALUE val, void *enc);
+
static int
ruby_brace_expand(const char *str, int flags, ruby_glob_func *func, VALUE arg,
                  rb_encoding *enc, VALUE var)
@@ -2280,7 +2345,7 @@ ruby_brace_expand(const char *str, int flags, ruby_glob_func *func, VALUE ar
g,
    const char *p = str;
    const char *pend = p + strlen(p);
    const char *s = p;
-    const char *lbrace = 0, *rbrace = 0;
+    const char *lbrace = NULL, *rbrace = NULL;
    int nest = 0, status = 0;

    while (*p) {
@@ -2299,9 +2364,18 @@ ruby_brace_expand(const char *str, int flags, ruby_glob_func *func, VALUE a
rg,

        if (lbrace && rbrace) {
            size_t len = strlen(s) + 1;
-            char *buf = GLOB_ALLOC_N(char, len);
+            char *buf;
            long shift;

+            if (func == push_caller && !strchr(lbrace, '/')) {
+                /* Now it reaches file basename entry. */
+                /* Handle braces in glob_helper */
+                struct push_glob_args *a = (struct push_glob_args *)arg;
+                a->flags |= FNM_EXTGLOB;
+                return glob_call_func(func, s, arg, enc);
+            }
+
            buf = GLOB_ALLOC_N(char, len);
            if (!buf) return -1;
            memcpy(buf, s, lbrace-s);
            shift = (lbrace-s);
@@ -2365,12 +2439,6 @@ ruby_brace_glob(const char *str, int flags, ruby_glob_func *func, VALUE arg
)
    return ruby_brace_glob_with_enc(str, flags, func, arg, rb_ascii8bit_encoding());
}

-struct push_glob_args {
-    struct glob_args glob;
-    int flags;
-    int fd;
-};
-
static int
push_caller(const char *path, VALUE val, void *enc)
{

```

#### Related issues:

Related to Ruby master - Bug #13167: Dir.glob is 25x slower since Ruby 2.2

Closed

## Associated revisions

---

### Revision e2a92c54 - 10/21/2017 10:28 AM - naruse (Yui NARUSE)

Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]

The order of resulted array is changed in some cases.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60253 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

### Revision 60253 - 10/21/2017 10:28 AM - naruse (Yui NARUSE)

Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]

The order of resulted array is changed in some cases.

### Revision 60253 - 10/21/2017 10:28 AM - naruse (Yui NARUSE)

Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]

The order of resulted array is changed in some cases.

### Revision 60253 - 10/21/2017 10:28 AM - naruse (Yui NARUSE)

Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]

The order of resulted array is changed in some cases.

### Revision c1db20c3 - 10/21/2017 10:44 AM - naruse (Yui NARUSE)

Revert "Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]"

This reverts commit r60253 because it causes on error on Linux.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60255 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

### Revision 60255 - 10/21/2017 10:44 AM - naruse (Yui NARUSE)

Revert "Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]"

This reverts commit r60253 because it causes on error on Linux.

### Revision 60255 - 10/21/2017 10:44 AM - naruse (Yui NARUSE)

Revert "Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]"

This reverts commit r60253 because it causes on error on Linux.

### Revision 60255 - 10/21/2017 10:44 AM - naruse (Yui NARUSE)

Revert "Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]"

This reverts commit r60253 because it causes on error on Linux.

### Revision 38667153 - 10/22/2017 02:03 AM - naruse (Yui NARUSE)

Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]

The order of resulted array is changed in some cases.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60341 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

### Revision 60341 - 10/22/2017 02:03 AM - naruse (Yui NARUSE)

Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]

The order of resulted array is changed in some cases.

### Revision 60341 - 10/22/2017 02:03 AM - naruse (Yui NARUSE)

Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]

The order of resulted array is changed in some cases.

### Revision 60341 - 10/22/2017 02:03 AM - naruse (Yui NARUSE)

Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]

The order of resulted array is changed in some cases.

**Revision a6488f04 - 10/27/2017 08:40 AM - naruse (Yui NARUSE)**

Revert "Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]"

This reverts commit r60341,r60342,r60344,r60345.  
Breaking compability of the order of result breaks many tests.  
To avoid such effort to fix tests, the order should be kept.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@60478 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 60478 - 10/27/2017 08:40 AM - naruse (Yui NARUSE)**

Revert "Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]"

This reverts commit r60341,r60342,r60344,r60345.  
Breaking compability of the order of result breaks many tests.  
To avoid such effort to fix tests, the order should be kept.

**Revision 60478 - 10/27/2017 08:40 AM - naruse (Yui NARUSE)**

Revert "Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]"

This reverts commit r60341,r60342,r60344,r60345.  
Breaking compability of the order of result breaks many tests.  
To avoid such effort to fix tests, the order should be kept.

**Revision 60478 - 10/27/2017 08:40 AM - naruse (Yui NARUSE)**

Revert "Dir.glob with FNM\_EXTGLOB is optimized [Feature #13873]"

This reverts commit r60341,r60342,r60344,r60345.  
Breaking compability of the order of result breaks many tests.  
To avoid such effort to fix tests, the order should be kept.

## History

---

**#1 - 09/26/2017 07:24 PM - naruse (Yui NARUSE)**

- *Related to Bug #13167: Dir.glob is 25x slower since Ruby 2.2 added*

**#2 - 10/19/2017 08:30 AM - matz (Yukihiro Matsumoto)**

Agreed.

Matz

**#3 - 10/21/2017 10:28 AM - naruse (Yui NARUSE)**

- *Status changed from Open to Closed*

Applied in changeset [trunk|r60253](#).

---

Dir.glob with FNM\_EXTGLOB is optimized [Feature [#13873](#)]

The order of resulted array is changed in some cases.