

erlang / otp Public

<> Code Issues 370 Pull requests 152 Actions Projects Wiki Se

Commit d49efa2

u3s committed on Sep 2, 2025 Verified

ssh: max_handles option added to ssh_sftpd
- add max_handles option and update tests (1000 by default)
- remove sshd_read_file redundant testcase

master · OTP-29.0-rc2 ... OTP-26.2.5.15
1 parent [c1ce39f](#) commit d49efa2

3 files changed +76 -58 lines changed

[↑ Top](#)

- ✓ lib/ssh
 - ✓ doc/src
 - ssh_sftpd.xml
 - ✓ src
 - ssh_sftpd.erl
 - ✓ test
 - ssh_sftpd_SUITE.erl

3 files changed +76 -58 lines changed

lib/ssh/doc/src/ssh_sftpd.xml

```

↑... @@ -65,6 +65,10 @@
65 65          If supplied, the number of filenames returned to the SFTP client per
      <C>READDIR</C>
66 66          request is limited to at most the given value.</p>
67 67      </item>

```

```

68 + <tag><c>max_handles</c></tag>
69 + <item>
70 + <p>The default value is <c>1000</c>. Positive integer value
    represents the maximum number of file handles allowed for a connection.</p>
71 + </item>
68 72 <tag><c>root</c></tag>
69 73 <item>
70 74 <p>Sets the SFTP root directory. Then the user cannot see any files

```

```

lib/ssh/src/ssh_sftpd.erl
@@ -52,6 +52,7 @@
52 52 file_handler, % atom() - callback module
53 53 file_state, % state for the file callback module
54 54 max_files, % integer >= 0 max no files sent during
    READDIR
55 + max_handles, % integer > 0 - max number of file handles
55 56 options, % from the subsystem declaration
56 57 handles % list of open handles
57 58 %% handle is either {<int>, directory, {Path, unread|eof}} or
@@ -65,6 +66,7 @@
65 66 Options :: [ {cwd, string()} |
66 67 {file_handler, CbMod | {CbMod, FileState}} |
67 68 {max_files, integer()} |
69 + {max_handles, integer()} |
68 70 {root, string()} |
69 71 {sftpd_vsn, integer()}
70 72 ],
@@ -115,8 +117,12 @@ init(Options) ->
115 117 {Root0, State0}
116 118 end,
117 119 MaxLength = proplists:get_value(max_files, Options, 0),
120 + MaxHandles = proplists:get_value(max_handles, Options, 1000),
118 121 Vsn = proplists:get_value(sftpd_vsn, Options, 5),
119 - {ok, State#state{cwd = CWD, root = Root, max_files = MaxLength,
122 + {ok, State#state{cwd = CWD,
123 + root = Root,
124 + max_files = MaxLength,
125 + max_handles = MaxHandles,

```

```

120 126         options = Options,
121 127         handles = [], pending = <<>>,
122 128         xf = #ssh_xfer{vsn = Vsn, ext = []}}}.
@@ -245,14 +251,16 @@ handle_op(?SSH_FXP_REALPATH, ReqId,
245 251         end;
246 252         handle_op(?SSH_FXP_OPENDIR, ReqId,
247 253         <<?UINT32(RLen), RPath:RLen/binary>>,
248 -         State0 = #state{xf = #ssh_xfer{vsn = Vsn},
249 -         file_handler = FileMod, file_state = FS0}) ->
254 +         State0 = #state{xf = #ssh_xfer{vsn = Vsn},
255 +         file_handler = FileMod, file_state = FS0,
256 +         max_handles = MaxHandles}) ->
250 257         RelPath = unicode:characters_to_list(RPath),
251 258         AbsPath = relate_file_name(RelPath, State0),
252 259
253 260         XF = State0#state.xf,
254 261         {IsDir, FS1} = FileMod:is_dir(AbsPath, FS0),
255 262         State1 = State0#state{file_state = FS1},
263 +         HandlesCnt = length(State0#state.handles),
256 264         case IsDir of
257 265         false when Vsn > 5 ->
258 266             ssh_xfer:xf_send_status(XF, ReqId, ?SSH_FX_NOT_A_DIRECTORY,
@@ -262,8 +270,12 @@ handle_op(?SSH_FXP_OPENDIR, ReqId,
262 270             ssh_xfer:xf_send_status(XF, ReqId, ?SSH_FX_FAILURE,
263 271                 "Not a directory"),
264 272             State1;
265 -         true ->
266 -         add_handle(State1, XF, ReqId, directory, {RelPath,unread})
273 +         true when HandlesCnt < MaxHandles ->
274 +         add_handle(State1, XF, ReqId, directory, {RelPath,unread});
275 +         true ->
276 +         ssh_xfer:xf_send_status(XF, ReqId, ?SSH_FX_FAILURE,
277 +             "max_handles limit reached"),
278 +         State1
267 279         end;
268 280         handle_op(?SSH_FXP_READDIR, ReqId,
269 281         <<?UINT32(HLen), BinHandle:HLen/binary>>,
@@ -720,7 +732,9 @@ open(Vsn, ReqId, Data, State) when Vsn >= 4 ->

```

```

720 732     do_open(ReqId, State, Path, Flags).
721 733
722 734     do_open(ReqId, State0, Path, Flags) ->
723 735     -   #state{file_handler = FileMod, file_state = FS0, xf = #ssh_xfer{vsn = Vsn}}
       = State0,
735 736     +   #state{file_handler = FileMod, file_state = FS0, xf = #ssh_xfer{vsn = Vsn},
736 737     +       max_handles = MaxHandles} = State0,
737 738     +   HandlesCnt = length(State0#state.handles),
724 738     AbsPath = relate_file_name(Path, State0),
725 739     {IsDir, _FS1} = FileMod:is_dir(AbsPath, FS0),
726 740     case IsDir of
↕     @@ -732,7 +746,7 @@ do_open(ReqId, State0, Path, Flags) ->
732 746         ssh_xfer:xf_send_status(State0#state.xf, ReqId,
733 747             ?SSH_FX_FAILURE, "File is a directory"),
734 748         State0;
735 749     -   false ->
749 750     +   false when HandlesCnt < MaxHandles ->
736 750         OpenFlags = [binary | Flags],
737 751         {Res, FS1} = FileMod:open(AbsPath, OpenFlags, FS0),
738 752         State1 = State0#state{file_state = FS1},
↕     @@ -743,7 +757,11 @@ do_open(ReqId, State0, Path, Flags) ->
743 757         ssh_xfer:xf_send_status(State1#state.xf, ReqId,
744 758             ssh_xfer:encode_erlang_status(Error)),
745 759         State1
746 760     -   end
760 761     +   end;
761 762     +   false ->
762 763     +   ssh_xfer:xf_send_status(State0#state.xf, ReqId,
763 764     +       ?SSH_FX_FAILURE, "max_handles limit reached"),
764 765     +   State0
747 765     end.
748 766
749 767     %% resolve all symlinks in a path
↕

```

lib/ssh/test/ssh_sftpd_SUITE.erl

...

```

↕     @@ -51,7 +51,6 @@
51 51     retrieve_attributes/1,
52 52     root_with_cwd/1,
53 53     set_attributes/1,

```

54	-	sshd_read_file/1,
55	54	ver3_open_flags/1,
56	55	ver3_rename/1,
57	56	ver6_basic/1,
↕		@@ -72,9 +71,8 @@
72	71	-define(SSH_TIMEOUT, 10000).
73	72	-define(REG_ATTRS, <<0,0,0,0,1>>).
74	73	-define(UNIX_EPOCH, 62167219200).
75	-	
76	-	-define(is_set(F, Bits),
77	-	((F) band (Bits)) == (F)).
	74	+ -define(MAX_HANDLES, 10).
	75	+ -define(is_set(F, Bits), ((F) band (Bits)) == (F)).
78	76	
79	77	%%-----
80	78	%% Common Test interface functions -----
↕		@@ -98,8 +96,7 @@ all() ->
98	96	links,
99	97	ver3_rename,
100	98	ver3_open_flags,
101	-	relpath,
102	-	sshd_read_file,
	99	+ relpath,
103	100	ver6_basic,
104	101	access_outside_root,
105	102	root_with_cwd,
↕		@@ -181,7 +178,7 @@ init_per_testcase(TestCase, Config) ->
181	178	{sftpd_vsn, 6}]]],
182	179	ssh:daemon(0, [{subsystems, SubSystems} Options]);
183	180	_ ->
184	-	SubSystems = [ssh_sftpd:subsystem_spec([])],
	181	+ SubSystems = [ssh_sftpd:subsystem_spec([max_handles, ?
		MAX_HANDLES}})],
185	182	ssh:daemon(0, [{subsystems, SubSystems} Options])
186	183	end,
187	184	
↕		@@ -317,33 +314,44 @@ open_close_dir(Config) when is_list(Config) ->
317	314	read_file(Config) when is_list(Config) ->

```

318 315 PrivDir = proplists:get_value(priv_dir, Config),
319 316 FileName = filename:join(PrivDir, "test.txt"),
317 +     {Cm, Channel} = proplists:get_value(sftp, Config),
318 +     [begin
319 +         R1 = req_id(),
320 +         {ok, <<?SSH_FXP_HANDLE, ?UINT32(R1), Handle/binary>>, _} =
321 +             open_file(FileName, Cm, Channel, R1, ?ACE4_READ_DATA bor ?
ACE4_READ_ATTRIBUTES,
322 +                 ?SSH_FXF_OPEN_EXISTING),
323 +         R2 = req_id(),
324 +         {ok, <<?SSH_FXP_DATA, ?UINT32(R2), ?UINT32(_Length), Data/binary>>,
_} =
325 +             read_file(Handle, 100, 0, Cm, Channel, R2),
326 +         {ok, Data} = file:read_file(FileName)
327 +     end || _I <- lists:seq(0, ?MAX_HANDLES-1)],
328 +     ReqId = req_id(),
329 +     {ok, <<?SSH_FXP_STATUS, ?UINT32(ReqId), ?UINT32(?SSH_FX_FAILURE),
330 +         ?UINT32(MsgLen), Msg:MsgLen/binary,
331 +         ?UINT32(LangTagLen), _LangTag:LangTagLen/binary>>, _} =
332 +         open_file(FileName, Cm, Channel, ReqId, ?ACE4_READ_DATA bor ?
ACE4_READ_ATTRIBUTES,
333 +             ?SSH_FXF_OPEN_EXISTING),
334 +         ct:log("Message: ~s", [Msg]),
335 +         ok.

```

```

320 336
321 -     ReqId = 0,
322 -     {Cm, Channel} = proplists:get_value(sftp, Config),
323 -
324 -     {ok, <<?SSH_FXP_HANDLE, ?UINT32(ReqId), Handle/binary>>, _} =
325 -         open_file(FileName, Cm, Channel, ReqId,
326 -             ?ACE4_READ_DATA bor ?ACE4_READ_ATTRIBUTES,
327 -             ?SSH_FXF_OPEN_EXISTING),
328 -
329 -     NewReqId = 1,
330 -
331 -     {ok, <<?SSH_FXP_DATA, ?UINT32(NewReqId), ?UINT32(_Length),
332 -         Data/binary>>, _} =
333 -         read_file(Handle, 100, 0, Cm, Channel, NewReqId),
334 -
335 -     {ok, Data} = file:read_file(FileName).

```

336	-	
337	-	%%-----
338	337	<code>read_dir(Config) when is_list(Config) -></code>
339	338	<code>PrivDir = proplists:get_value(priv_dir, Config),</code>
340	339	<code>{Cm, Channel} = proplists:get_value(sftp, Config),</code>
341	-	<code>ReqId = 0,</code>
342	-	<code>{ok, <<?SSH_FXP_HANDLE, ?UINT32(ReqId), Handle/binary>>, _} =</code>
343	-	<code>open_dir(PrivDir, Cm, Channel, ReqId),</code>
344	-	<code>ok = read_dir(Handle, Cm, Channel, ReqId).</code>
340	+	<code>[begin</code>
341	+	<code> R1 = req_id(),</code>
342	+	<code> {ok, <<?SSH_FXP_HANDLE, ?UINT32(R1), Handle/binary>>, _} =</code>
343	+	<code> open_dir(PrivDir, Cm, Channel, R1),</code>
344	+	<code> R2 = req_id(),</code>
345	+	<code> ok = read_dir(Handle, Cm, Channel, R2)</code>
346	+	<code>end _I <- lists:seq(0, ?MAX_HANDLES-1)],</code>
347	+	<code>ReqId = req_id(),</code>
348	+	<code>{ok, <<?SSH_FXP_STATUS, ?UINT32(ReqId), ?UINT32(?SSH_FX_FAILURE),</code>
349	+	<code> ?UINT32(MsgLen), Msg:MsgLen/binary,</code>
350	+	<code> ?UINT32(LangTagLen), _LangTag:LangTagLen/binary>>, _} =</code>
351	+	<code> open_dir(PrivDir, Cm, Channel, ReqId),</code>
352	+	<code> ct:log("Message: ~s", [Msg]),</code>
353	+	<code> ok.</code>
345	354	
346	-	%%-----
347	355	<code>write_file(Config) when is_list(Config) -></code>
348	356	<code>PrivDir = proplists:get_value(priv_dir, Config),</code>
349	357	<code>FileName = filename:join(PrivDir, "test.txt"),</code>
		<code>@@ -645,27 +653,6 @@ relpath(Config) when is_list(Config) -></code>
645	653	<code> Root = Path</code>
646	654	<code>end.</code>
647	655	
648	-	%%-----
649	-	<code>ssh_read_file(Config) when is_list(Config) -></code>
650	-	<code>PrivDir = proplists:get_value(priv_dir, Config),</code>
651	-	<code>FileName = filename:join(PrivDir, "test.txt"),</code>
652	-	
653	-	<code>ReqId = 0,</code>
654	-	<code>{Cm, Channel} = proplists:get_value(sftp, Config),</code>

```

655 -
656 -     {ok, <<?SSH_FXP_HANDLE, ?UINT32(ReqId), Handle/binary>>, _} =
657 -     open_file(FileName, Cm, Channel, ReqId,
658 -             ?ACE4_READ_DATA bor ?ACE4_READ_ATTRIBUTES,
659 -             ?SSH_FXF_OPEN_EXISTING),
660 -
661 -     NewReqId = 1,
662 -
663 -     {ok, <<?SSH_FXP_DATA, ?UINT32(NewReqId), ?UINT32(_Length),
664 -     Data/binary>>, _} =
665 -     read_file(Handle, 100, 0, Cm, Channel, NewReqId),
666 -
667 -     {ok, Data} = file:read_file(FileName).
668 -     %%-----
669 656     ver6_basic(Config) when is_list(Config) ->
670 657         PrivDir = proplists:get_value(priv_dir, Config),
671 658         %FileName = filename:join(PrivDir, "test.txt"),
672
673     @@ -1087,3 +1074,12 @@ encode_file_type(Type) ->
674
675 1087 1074
676 1088 1075     not_default_permissions() ->
677 1089 1076         8#600. %% User read-write-only
678
679 1077 +
680 1078 + req_id() ->
681 1079 +     ReqId =
682 1080 +         case get(req_id) of
683 1081 +             undefined -> 0;
684 1082 +             I -> I
685 1083 +         end,
686 1084 +         put(req_id, ReqId + 1),
687 1085 +         ReqId.

```

Comments 0



Please [sign in](#) to comment.