

projekt kivitendo - Fehler #96

Template file templates/webpages/menu/*.html not found

05.10.2015 10:07 - Jan Büren

Status:	Neu	Beginn:	05.10.2015
Priorität:	Normal	Abgabedatum:	
Zugewiesen an:		% erledigt:	0%
Kategorie:		Geschätzter Aufwand:	0:00 Stunde
Zielversion:		Aufgewendete Zeit:	0:00 Stunde
Beschreibung			
Unregelmässig kommt es im fcgi-Modus zu dieser Fehlermeldung:			
<pre>[Mon Oct 05 09:47:40 2015] [warn] [client 87.193.197.72] mod_fcgid: stderr: Template file templates/webpages/menu/header.html not found at /usr/local/src/lx-office-erp/SL/Layout/Top.pm line 9, referer: e/lx-erp/do.pl?action=edit&type=sales_delivery_order&vc=customer&id=388434&callback=do.pl%3faction%3dorders%26l_transdate%3dY%26l_reqdate%3dY%26l_donumber%3dY%26l_ordnumber%3dY%26l_cusordnumber%3dY%26l_name%3dY%26l_employee%3dY%26l_delivered%3dY%26open%3d1%26delivered%3d1%26notdelivered%3d1%26transdatefrom%3d05.10.2015%26type%3dsales_delivery_order%26vc%3dcustomer%26sort%3dtransdate</pre>			
Als erste Idee, hab ich das ulimit höher gesetzt, dass ist aber mehr eine Aktion auf Verdacht:			
<pre>1. vim security/limits.conf + www-data - nofile 20000 1. su - www-data \$ ulimit -n 20000</pre>			
Das hat in einem anderen fcgi Perl Projekt entsprechend geholfen, allerdings war dort die Fehlermeldung auch eindeutiger. Die Frage ist, warum kann diese Datei von Perl nicht mehr geöffnet werden, müsste eigentlich mit einem FileDescriptor-Limit, o.ä. zu tun haben.			

Zugehörige Revisionen

Revision 108753a7 - 22.11.2019 22:45 - Bernd Bleßmann

WebDav: Fehler beim Kopieren anzeigen / Verzeichnis zurück wechseln

Wenn in SL::Form->parse_template bei Common::copy_file_to_webdav_folder etwas schief ging, wurde dort ein "die" oder "Form->error" aufgerufen. Allerdings wird in parse_template vorher das Arbeitsverzeichnis gewechselt, so dass die web-templates zum Anzeigen des Fehlers nicht mehr gefunden werden.

Dies ist nur ein schlechter Fix. In #96 (redmine) sind einige bessere Lösungen erwähnt, die aber etwas mehr Aufwand und vor allem Testen verlangen.

Bezieht sich auch auf #96 (redmine)
Refs #96

Revision da1f7513 - 26.11.2019 10:22 - Bernd Bleßmann

WebDav: Fehler beim Kopieren anzeigen / Verzeichnis zurück wechseln (2)

Der erste commit 108753a78b203dbe0ccb6438cc16c8df33c04d3 hat das Drucken ohne Fehler beim Ins-Webdav-Kopieren kaputt gemacht. Probleme waren:

- ein return vergessen
- chdir zurück auch ohne Fehler

Diese commit fixt das.

Bezieht sich auch auf #96 (redmine)
Refs #96

Historie

#1 - 06.10.2015 09:37 - Moritz Bunkus

Wenn das das nächste Mal passiert, wäre es sinnvoll, dass du zuerst mal nachschaust, was gerade überhaupt offen ist. Dazu kannst du z.B. »lsof« nutzen. Interessant wäre zum Einen, wie viele Dateien offen sind, und zum Anderen welche.

Ich rate auch, die Anpassung vom File-Limit erst mal rückgängig zu machen, da blindes Stochern das Debuggen meist erschwert.

#2 - 06.10.2015 10:12 - Jan Büren

Moritz Bunkus schrieb:

Wenn das das nächste Mal passiert, wäre es sinnvoll, dass du zuerst mal nachschaust, was gerade überhaupt offen ist. Dazu kannst du z.B. »lsof« nutzen. Interessant wäre zum Einen, wie viele Dateien offen sind, und zum Anderen welche.

Danke für den Hinweis. Wir haben noch ein anderes System wo dies unregelmässig auftritt, daran hab ich total nicht mehr gedacht.

```
$ lsof | grep www-data | wc -l
```

Ich hab kurz was geskriptet, dass sollte ausreichen, für die Analyse:

```
#!/bin/bash

a=`lsof | grep www-data | wc -l`

if [ $a > 1100 ]
then
    lsof | grep www-data > /usr/local/sbin/ulimit-watch.log
fi
```

#3 - 13.10.2015 10:21 - Bernd Bleßmann

Es ist wieder passiert. Die Liste der offenen Dateien offenbart (mir) nichts ungewöhnliches. Heute war allerdings ein dispatcher.pl-Prozess da, der 75% des Hauptspeichers belegte und es wurde auch schon ausgelagert.

#4 - 13.10.2015 10:30 - Moritz Bunkus

Klingt nach einem Speicherleck. Entweder, du untersuchst das weiter, oder du implementierst einen Workaround.

Weiter untersuchen bedeutet: regelmäßiges Loggen des vom dispatcher.pl benötigten Speichers (z.B. einmal pro Minute das aktuelle Datum und die Ausgabe von »ps auxw|grep dispatcher.pl« oder so in eine Datei schreiben); dann untersuchen, ob der Speicherbedarf langsam und stetig wächst oder ob es große Sprünge gibt.

Workaround: einmal täglich (z.B. früh morgens) den Webserverprozess neu starten.

So ein Restart ist eh sinnvoll, weil ein bekanntes Problem von Perl ist, dass einmal vom Betriebssystem allozierter Speicher nicht wieder zurückgegeben wird, selbst wenn Perl intern danach nur noch deutlich weniger Speicher benötigen würde.

#5 - 13.10.2015 10:52 - Sven Schöling

Aus genau dem Grund kann der Apache auch in regelmässigen Abständen fcgid Prozesse für euch neu starten. Die entsprechende Konfigoption ist FcgidMaxRequestsPerProcess (default 0 = disabled).

Zum eigentlichen Problem kann ich nicht viel sagen ausser was Mosu auch schon gesagt hat. Cargo Cult Administration ist murks, lass das. Ich hab den Fehler noch nie gesehen. Ich habe kurz meine Kunden durchgeschaut und finde auch bei keinem von denen so eine Fehlermeldung.

Mein Debugging Ansatz wäre:

- Da steht file not found. Wenn der die Datei nicht findet, hat sich das cwd geändert? Liegt die Datei im NFS und das ist kurz weg?
- Ist das ne Standardversion? Benutzen die irgendwelche esoterischen Mandantenconfigs? Ist der Server ne VM? Ist das ein debianoid oder nicht? suexec? vhost? ssl?
- Wenn ihr [debug] global = REQUEST anmacht, was waren die letzten Requests die vor der Fehlermeldung reingekommen sind? Muster erkennbar?
- wenn ihr wissen wollt, wie sich die Prozessgrösse entwickelt wäre es cool, wenn Ihr in das `$::lxdebug->end_request` einbaut, dass der optional die Prozessgrösse loggen kann, dann kann man sehen wo das alles hingeht.

#6 - 13.10.2015 10:55 - Moritz Bunkus

cwd = working directory geändert ist eine gute Frage.

Wenn es das nächste Mal auftritt, dann schaut bitte mal nach, wohin der Symlink `/proc/<PIDvonDispatcher.pl>/cwd` zeigt.

#7 - 13.10.2015 12:38 - Bernd Bleßmann

Erstmal Danke für die Hinweise.

Den Workaround mit dem Neustart machen wir ohnehin schon einmal früh morgens. Ich werde mir dann auch mal FcgidMaxRequestsPerProcess ansehen.

Ich habe jetzt zuerst versucht, das irgendwie nachstellen zu können. Das Problem tritt nicht nur in der Kundeninstallation auf, sondern ich kann es auch auf unserem Entwicklungs-Rechner mit der unstable reproduzieren:

- apache neu gestartet
- einloggen
- sehr lange Liste mit Aufträgen anzeigen lassen (hier ca. 9500)
- einen Auftrag in neuem Tab öffnen
- Drucken -> der Fehler kommt

/proc/xxx/cwd zeigt auf das Installations-Verzeichnis, aber eine Debug-Ausgabe in SL::Presenter::render vor der Stelle mit, an der die Fehlermeldung ausgegeben wird, zeigt, dass das CWD auf ../users steht.

Mir scheint, dass das PDF-Erzeugen / Drucken ohne Fehler abbricht und das working dir nicht zurücksetzt.
Ich schaue mal weiter.

#8 - 13.10.2015 13:03 - Moritz Bunkus

Interessant. Ein Backtrace von der Stelle würde helfen. Mach doch bitte in der Konfiguration die Option backtrace_on die an, Apache neu starten, Fehler erzwingen. Dann sollte ein Backtrace bei der Exception im kivitendo-Log auftauchen (oder Error-Log? Bin mir gerade nicht sicher).

Alternativ die folgende Zeile in SL/Layout/Top.pm in sub pre_content vor dem \$self->render(...) einfügen:

```
$::lxdebug->show_backtrace(1) if ! -f 'templates/webpages/menu/header.html';
```

#9 - 13.10.2015 15:33 - Bernd Bleßmann

Moritz Bunkus schrieb:

Interessant. Ein Backtrace von der Stelle würde helfen. Mach doch bitte in der Konfiguration die Option backtrace_on die an, Apache neu starten, Fehler erzwingen. Dann sollte ein Backtrace bei der Exception im kivitendo-Log auftauchen (oder Error-Log? Bin mir gerade nicht sicher).

Ja, das gibt weiteren Aufschluss. Log erscheint im error-log und im Browser.

```
Template file templates/webpages/menu/header.html not found at /usr/local/src/kivitendo-bernd/SL/Layout/Top.pm line 9.
at SL/Dispatcher.pm line 168.
  SL::Dispatcher::__ANON__('Template file templates/webpages/menu/header.html not found a...') called at /usr/share/perl/5.18/Carp.pm line 100
  Carp::croak('Template file templates/webpages/menu/header.html not found') called at /usr/local/src/kivitendo-bernd/SL/Presenter.pm line 66
  SL::Presenter::render('SL::Presenter=HASH(0x9cf2d40)', 'menu/header', 'now', 'DateTime=HASH(0x9566560)', 'is_fastcgi', 1, 'is_links', '') called at /usr/local/src/kivitendo-bernd/SL/Layout/Top.pm line 9
  SL::Layout::Top::pre_content('SL::Layout::Top=HASH(0x9cee00)') called at /usr/local/src/kivitendo-bernd/SL/Layout/Base.pm line 45
  SL::Layout::Base::pre_content('SL::Layout::V3=HASH(0x8dda3e0)') called at /usr/local/src/kivitendo-bernd/SL/Form.pm line 527
  Form::header('Form=HASH(0x8d20b48)') called at /usr/local/src/kivitendo-bernd/SL/Form.pm line 745
  Form::show_generic_error('Form=HASH(0x8d20b48)', 'Kopieren der Datei von /usr/local/src/kivitendo-bernd/users/k...') called at /usr/local/src/kivitendo-bernd/SL/Form.pm line 312
  Form::error('Form=HASH(0x8d20b48)', 'Kopieren der Datei von /usr/local/src/kivitendo-bernd/users/k...') called at /usr/local/src/kivitendo-bernd/SL/Common.pm line 667
  Common::copy_file_to_webdav_folder('Form=HASH(0x8d20b48)') called at /usr/local/src/kivitendo-bernd/SL/Form.pm line 1138
  Form::parse_template('Form=HASH(0x8d20b48)', 'HASH(0x8d10e28)') called at /usr/local/src/kivitendo-bernd/bin/mozilla/io.pl line 1650
  main::print_form(undef) called at /usr/local/src/kivitendo-bernd/bin/mozilla/io.pl line 1261
  main::print() called at /usr/local/src/kivitendo-bernd/bin/mozilla/common.pl line 431
  main::call_sub('print') called at /usr/local/src/kivitendo-bernd/bin/mozilla/oe.pl line 2149
  main::dispatcher() called at /usr/local/src/kivitendo-bernd/bin/mozilla/common.pl line 431
  main::call_sub('::dispatcher') called at SL/Dispatcher.pm line 295
  eval {...} called at SL/Dispatcher.pm line 304
  SL::Dispatcher::handle_request('SL::Dispatcher=HASH(0x21c7cb8)', 'FCGI=SCALAR(0x7f94758)') called at /usr/local/src/kivitendo-bernd/dispatcher.fpl line 15
```

#10 - 13.10.2015 15:57 - Moritz Bunkus

Ach euer komischer Mechanismus zum Kopieren ins WebDav. Das dürft ihr gerne selber fixen :)

Kopieren schlägt fehl (nicht gefunden weil aktuelles Verzeichnis falsch? keine Berechtigungen für Webserveruser?), dann soll dazu ein Fehler ausgegeben werden, aber das Verzeichnis stimmt noch nicht.

Richtig gefixt wäre das IMHO so, dass direkt nach dem Aufruf des Template-Parsers bereits das Verzeichnis wieder auf das Installationsverzeichnis geändert wird und all nachfolgender Code in `Form::parse_template` dann entsprechend davon ausgeht, im Installationsverzeichnis zu sein und nicht in `users`.

#11 - 13.10.2015 16:13 - Bernd Bleßmann

Moritz Bunkus schrieb:

Ach euer komischer Mechanismus zum Kopieren ins WebDav. Das dürft ihr gerne selber fixen :)

Kopieren schlägt fehl (nicht gefunden weil aktuelles Verzeichnis falsch? keine Berechtigungen für Webserveruser?), dann soll dazu ein Fehler ausgegeben werden, aber das Verzeichnis stimmt noch nicht.

Nee. Die pdf-Datei existiert gar nicht (oder nicht mehr). Weiß noch nicht, warum.

Richtig gefixt wäre das IMHO so, dass direkt nach dem Aufruf des Template-Parsers bereits das Verzeichnis wieder auf das Installationsverzeichnis geändert wird und all nachfolgender Code in `Form::parse_template` dann entsprechend davon ausgeht, im Installationsverzeichnis zu sein und nicht in `users`.

Sollte nicht eher der Template-Parser selber ins ursprüngliche Verzeichnis zurückwechseln, wenn er schon dahin wechselt?

#12 - 13.10.2015 16:19 - Moritz Bunkus

Prinzipiell sollten das alle tun, aber momentan tut es keiner. Warum nicht? Weil alles Uralt-Code ist, der immer nur notdürftig refactort wurde.

Wenn du es also über-korrekt fixen willst:

- In jedem `SL::Template::*` die `parse`-Funktion so anpassen, dass sie: * das Arbeitsverzeichnis speichert, * Exceptions abfängt * Im Falle einer Exception das Arbeitsverzeichnis wiederherstellt und die gleiche Exception erneut wirft, * im Falle keiner Exception einfach nur das Arbeitsverzeichnis wiederherstellt
- `Form::parse_template` so anpassen, dass sie gar nichts mit dem Verzeichnis macht

Bei solchen Dingen vermisste ich Programmiersprachen mit *richtigen* Objekten und richtigen Exceptions. In C++ wäre so etwas absolut trivial und vor allem korrekt umsetzbar **seuz**

#13 - 13.10.2015 17:24 - Bernd Bleßmann

Nochmal zurück zum ursprünglichen Problem: auch wenn ich webdav abschalte, kommt es zu einem Fehler, nämlich, dass die pdf-Datei nicht gefunden wird. Das Problem mit dem falschen working dir hat das nur verschleiert.

Anscheinend geht der system-Aufruf für `pdflatex` schief. Liefert -1 zurück, was besagt, dass der Prozess nicht gestartet werden konnte. Mehr weiß ich noch nicht

Das hier nur zur Info - demnächst hoffentlich mehr.

#14 - 15.10.2015 12:07 - Jan Büren

Ok. Das Verhalten lässt sich reproduzieren, wenn der RAM insgesamt knapp wird, bzw. auslagert.

Es kommt dann aber auch schon an anderen Stellen im Log / Programm zu `internal server errors`, z.B.

```
[Thu Oct 15 08:43:12.910442 2015] [fcgid:warn] [pid 15889:tid 140513145816960] mod_fcgid: cleanup zombie process 15955
[Thu Oct 15 08:43:26.129464 2015] [cgid:error] [pid 15890:tid 140513034905344] [client 130.180.56.146:33357] End of script output before headers: oe.pl, referer: https://kiebitz.kivitendo.de/kivitendo-jan/oe.pl?action=search&type=sales_order
```

Parallel auf der Konsole:

```
kiebitz@kiebitz:~$ free -m
-bash: fork: Cannot allocate memory
```

Oder sowas:

```
[pid 15887:tid 140513145816960] (12)Cannot allocate memory: AH01252: couldn't create child process: 12: oe.pl
[pid 15891:tid 140512900130560] [client 130.180.56.146:33556] End of script output before headers: oe.pl, referer: https://kiebitz.kivitendo.de/kivitendo-jan/oe.pl?action=search&type=sales_order
[pid 15891:tid 140512900130560] [client 130.180.56.146:33556] AH01261: daemon couldn't find CGI process for connection 72, referer: https://kiebitz.kivitendo.de/kivitendo-jan/oe.pl?action=search&type=sales_order
[pid 15884:tid 140513145816960] AH00052: child pid 15890 exit signal Segmentation fault (11)
```

```
Can't load '/usr/lib/perl5/auto/Bit/Vector/Vector.so' for module Bit::Vector: /usr/lib/perl5/auto/Bit/Vector/V
ector.so: failed to map segment from shared object: Cannot allocate memory at /usr/lib/perl/5.18/DynaLoader.pm
 line 184.
```

Mit folgender Ergänzung:

```
system("${latex} --interaction=nonstopmode $form->{tmpfile} > $form->{tmpfile}.err");
if ($ret != 0) {
    croak "buggy pdf call mit: $!, $?, $ret";
}
```

... und einem Treffer genau beim Ausdruck, wird in etwa folgende Fehlermeldung geworfen:

```
buggy pdf call mit: Inappropriate ioctl for device at /usr/local/src/kivitendo-jan/SL/Form.pm line 1117.
```

Wenn ich den Aufruf eine Ebene tiefer analysiere, und ein strace für pdflatex setze, sieht man, dass im Fehlerfall der Befehl (pdflatex) erst gar nicht ausgeführt wird, bzw. das strace keine Ausgabe mehr zurückliefert:

Normalfall:

```
close(4)                                = 0
munmap(0x7fa126ac8000, 4096)             = 0
write(1, "\nOutput written on kivitendo-pri...", 118) = 118
exit_group(0)                           = ?
+++ exited with 0 +++
i di
```

Hat hier jmd. noch eine andere (nicht bessere) Idee als den RAM zu erhöhen und den Systemaufruf wie folgt etwas sinnvoller im Fehlerfall zu behandeln:

```
for (my $run = 1; $run <= 2; $run++) {
+   my $ret = system("${latex} --interaction=nonstopmode $form->{tmpfile} " .
+               " > $form->{tmpfile}.err");
+   #my $ret = system("strace -f ${latex} --interaction=nonstopmode $form->{tmpfile} > $form->{tmpfile}.err");
+   if ($ret != 0) {
+       croak "buggy pdf call mit: $!, $?, $ret";
+   }
+   if ($?) {
+       $ENV{HOME}      = $old_home;
+       $ENV{openin_any} = $old_openin_any;
+       $self->{error} = $form->cleanup($latex);
+       return 0;
+   }
+ }
```

Wäre ich dankbar. Falls nicht, bin ich auch zufrieden nichts wesentliches übersehen zu haben.

#15 - 15.10.2015 12:23 - Moritz Bunkus

Den Rückgabewert von system zu prüfen ist durchaus sinnvoll. Aber bitte mit richtiger Einrückung, ordentlicher Fehlermeldung und vor allem mit die und nicht mit croak. croak ist für Fälle sinnvoll, wo signalisiert werden soll, dass die Benutzung der aktuellen Funktion falsch ist, z.B. wenn du eine Funktion aufrufst und vergisst, essenzielle Parameter zu übergeben. Ansonsten ist die immer die richtige Variante.

#16 - 15.10.2015 12:24 - Jan Büren

Hi,
yo. die, die, die. Die Einrückung ist nur hier durchs manuelle Einfügen der Pluszeichen doof.

Danke für die schnelle Rückmeldung

#17 - 15.10.2015 14:25 - Bernd Bleßmann

Moritz Bunkus schrieb:

Den Rückgabewert von system zu prüfen ist durchaus sinnvoll. Aber bitte mit richtiger Einrückung, ordentlicher Fehlermeldung und vor allem mit die und nicht mit croak. croak ist für Fälle sinnvoll, wo signalisiert werden soll, dass die Benutzung der aktuellen Funktion falsch ist, z.B. wenn du eine Funktion aufrufst und vergisst, essenzielle Parameter zu übergeben. Ansonsten ist die immer die richtige Variante.

Ich habe die Überprüfung des Rückgabewerts in [232d78687663884df38c106f9089f637509722fd](#) eingebaut.

Das Fixen des Ich-bin-im-falschen-Verzeichnis-Webdav-Bugs steht noch aus.

#18 - 21.10.2015 13:32 - Jan Büren

Moritz Bunkus schrieb:

Ach euer komischer Mechanismus zum Kopieren ins WebDav. Das dürft ihr gerne selber fixen :)

Kopieren schlägt fehl (nicht gefunden weil aktuelles Verzeichnis falsch? keine Berechtigungen für Webserveruser?), dann soll dazu ein Fehler ausgegeben werden, aber das Verzeichnis stimmt noch nicht.

Richtig gefixt wäre das IMHO so, dass direkt nach dem Aufruf des Template-Parsers bereits das Verzeichnis wieder auf das Installationsverzeichnis geändert wird und all nachfolgender Code in Form::parse_template dann entsprechend davon ausgeht, im Installationsverzeichnis zu sein und nicht in users.

Hmm. Die parse_template Funktion überblick ich nicht direkt.

Ok, self->cleanup wechselt nach tmpdir und springt dann wieder nach cwd.
Von daher fühlt sich das safe an:

```
if (!$template->parse(*OUT)) {
    $self->cleanup();
    $self->error("$self->{IN} : " . $template->get_error());
}
```

```
close OUT if $self->{OUT};
+ chdir("$self->{cwd}");
```

Alle nachfolgenden chdir können dann raus in dieser Methode. In der Tat müsste dann copy_file_to_webdav_folder nochmal gecheckt werden, ob ich hier dieses Verzeichnis auch erwarte.

Das hier sieht dann stark überflüssig/optimierbar aus:

```
copy_file_to_webdav_folder

# maybe the path does not exist (automatic printing), see #2446
if (!-d $complete_path) {
    # we need a chdir and restore old dir
    my $current_dir = POSIX::getcwd();
    chdir("$form->{cwd}");
    mkdir_with_parents($webdav_folder);
    chdir($current_dir);
}
```

Noch besser fühlt sich dann aber eine Überarbeitung an, die SL/Webdav/File.pm verwendet.

#19 - 22.10.2015 09:49 - Jan Büren

Aktuell ist es wieder zu dieser Fehlermeldung gekommen:

Top sagt:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
3410	www-data	20	0	1955m	1.7g	7296	S	0	58.2	1:41.33	dispatcher.fcgi

```
[Thu Oct 22 09:18:02 2015] [warn] [client 130.180.124.54] mod_fcgid: stderr: system call to /usr/bin/pdflatex failed: Cannot allocate memory at /usr/local/src/lx-office-erp/SL/Template/LaTeX.pm line 539.,
[Thu Oct 22 09:21:53 2015] [warn] [client 130.180.124.54] mod_fcgid: stderr: system call to /usr/bin/pdflatex failed: Cannot allocate memory at /usr/local/src/lx-office-erp/SL/Template/LaTeX.pm line 539.,
[Thu Oct 22 09:22:49 2015] [warn] [client 130.180.124.54] mod_fcgid: stderr: system call to /usr/bin/pdflatex failed: Cannot allocate memory at /usr/local/src/lx-office-erp/SL/Template/LaTeX.pm line 539., referer: /lx-erp/oe.pl?action=edit&type=sales_order&id=394287
[Thu Oct 22 09:24:09 2015] [warn] [client 130.180.124.54] mod_fcgid: stderr: system call to /usr/bin/pdflatex failed: Cannot allocate memory at /usr/local/src/lx-office-erp/SL/Template/LaTeX.pm line 539., referer: /lx-erp/oe.pl?action=edit&type=sales_order&vc=customer&id=394287&callback=oe.pl%3faction%3dorders%26l_transdate%3dY%26l_reqdate%3dY%26l_ordnumber%3dY%26l_cusordnumber%3dY%26l_name%3dY%26l_netamount%3dY%26l_amount%3dY%26l_employee%3dY%26l_globalprojectnumber%3dY%26l_open%3dY%26l_delivered%3dY%26l_payment_terms%3dY%26l_exported_sp_bekuplast%3dY%26l_cvar_NK_Termin%3dY%26l_closed%3dY%26open%3d1%26closed%3d1%26delivered%3d1%26notdelivered%3d1%26ordnumber%3d29716%26type%3dsales_order%26vc%3dcustomer%26sort%3dtransdate
```

Der Physikalische RAM liegt bei 3GB. Gibt es hier vielleicht ein 2GB-Limit für fcgi oder sowas?

```
# free -m
```

	total	used	free	shared	buffers	cached
Mem:	3003	2910	92	0	136	381
-/+ buffers/cache:		2392	610			
Swap:	1021	2	1019			

In syslog ist nichts

#20 - 22.10.2015 13:22 - Jan Büren

Ich hab jetzt noch ein strace an den dispatcher.fcgi drangehängen, auch hier ist nichts weiter zu erkennen:

```
write(3, "\\1\\7\\0\\1\\0\\177\\1\\0system call to /usr/bin/"..., 144) = 144
write(3, "\\1\\6\\0\\1\\5@\\0\\0\\", "\\controller.pl?action="..., 1376) = 1376
shutdown(3, 1 /* send */) = 0
select(4, [3], NULL, NULL, {2, 0}) = 1 (in [3], left {1, 999999})
read(3, "", 1024) = 0
close(3) = 0
```

Weise Perlmönche berichten, dass (v)forks davon ausgehen, dass der Kind-Prozess denselben Speicherbedarf hat, wie der Vater-Prozess, dass könnte eine plausible Erklärung sein. [1[http://www.perlmonks.org/?node_id=861672]]

Ich hab den Swap jetzt doppelt so hoch wie den RAM, dass ist wie Linux im Jahre 95 bis 2005 ...

```
free -m
```

	total	used	free	shared	buffers	cached
Mem:	3003	2785	217	0	17	558
-/+ buffers/cache:		2209	793			
Swap:	6141	26	6115			

Wenn die Vermutung stimmt, dann denkt das System jetzt:

Parent RAM 1.7gb

-> system (1.7gb * 2 < mem + swap) == true => i.O.

to be continued ...

#21 - 23.10.2015 11:12 - Bernd Bleßmann

Jan Büren schrieb:

Weise Perlmönche berichten, dass (v)forks davon ausgehen, dass der Kind-Prozess denselben Speicherbedarf hat, wie der Vater-Prozess, dass könnte eine plausible Erklärung sein. [1[http://www.perlmonks.org/?node_id=861672]]

Prinzipiell stimmt das, da der Prozess kopiert wird. Allerdings wird dabei versucht, mit "copy on write" und anderen Techniken Speicherplatz zu sparen.

Ich hab den Swap jetzt doppelt so hoch wie den RAM, dass ist wie Linux im Jahre 95 bis 2005 ...

Da sehe ich eher ein Problem - liegt evtl. an der Virtualisierung mit KVM. Denn aus Deinen letzten beiden Posts geht hervor, das gar nicht ausgelagert wird. Ich habe hier das Problem mal mit einer Installation in VirtualBox nachgestellt. Da kommt kein Fehler und der (große) Swap-Bereich wird kräftig benutzt.

Evtl. sollten wir erstmal Debug-Statements bzw. Logging einschalten, um herauszubekommen, was vorher genau gemacht wird.

to be continued ...

Meiner Meinung nach ist dieses Ticket aber gelöst, wenn der webdav-Verzeichnis-wechsel-dich-Bug behoben ist. Mit dem Speicherverbrauch oder KVM oder sonst was sollten wir evtl. ein neues aufmachen.