

Re: optimize log parsing

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2005-10/msg00304.html>

- *From:* "it_says_BALLS_on_your forehead" <simon.chao@xxxxxxx>
 - *Date:* 5 Oct 2005 13:06:02 -0700
-

xhos...@xxxxxxxxxx wrote:

> "it_says_BALLS_on_your forehead" <simon.chao@xxxxxxx> wrote:

>> Hey Xho, I tried this:

>> ----

>> #!/apps/webstats/bin/perl

>>

>> use File::Copy;

>> use Parallel::ForkManager;

>>

>> my \$pm = Parallel::ForkManager->new(5);

>>

>> \$pm->run_on_start(

>> sub { my (\$pid,\$ident)=@_;

>> print "*** \$ident started, pid: \$pid\n";

>> }

>>);

>>

>> my @data = 1 ... shift;

>> for (@data) {

>> my \$pid = \$pm->start and next;

>> print "\$pid: \$_\n";

>> \$pm->finish;

>> }

>>

>> \$pm->wait_all_children;

>> -----

>> and got this:

>> #####

>> [smro180 123] ~/simon/1-perl > tryFork.pl 10

>> ** started, pid: 16208

>> 0: 1

>> ** started, pid: 16209

>> 0: 2

>> ** started, pid: 16210

> ...

>>

>> ...I read this:

>> start [\$process_identifier]

Re: optimize log parsing

>> This method does the fork. It returns the pid of the child process for
>> the parent, and 0 for the child process. If the \$processes parameter
>> for the constructor is 0 then, assuming you're in the child process,
>> \$pm->start simply returns 0.
>>
>> An optional \$process_identifier can be provided to this method... It is
>> used by the "run_on_finish" callback (see CALLBACKS) for identifying
>> the finished process.
>>
>> and this:
>> run_on_start \$code
>> You can define a subroutine which is called when a child is started. It
>> called after the successful startup of a child in the parent process.
>>
>> The parameters of the \$code are the following:
>>
>> - pid of the process which has been started
>> - identification of the process (if provi