

Re: question about forked processes writing to the same file

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2005-10/msg01374.html>

- *From:* "it_says_BALLS_on_your forehead" <simon.chao@xxxxxxx>
 - *Date:* 23 Oct 2005 12:43:43 -0700
-

xhoster@xxxxxxxxx wrote:

```
> "it_says_BALLS_on_your forehead" <simon.chao@xxxxxxx> wrote:
>>
>> still, i have the code written in such a way that i'm opening, locking,
>> writing to, and closing a file handle for every record.
>
> From the code you posted before, you are forking a process for every
> record. Given that, it is hard to believe that opening and closing a file
> is significant.
>
>> i wish there
>> were a way that lock and release without having to open/close every
>> time.
>
> There is. Have you read the perldoc -q append entry someone pointed out
> earlier?
>
>> i don't know if that would save much time or not. maybe the extra
>> time is not from the extra opening/closing, but from the locking and
>> having to wait to write. i will do more research and share if i find a
>> solution.
>
> I'm still incredulous. If any of this has a meaningful impact on
> performance, you are doing something wrong at a more fundamental level.
```

hey Xho, i tried to trim down the code as much as possible while maintaining the exact same functionality of the code. in this simplified version, there are 2 scripts: betaProcess.pl and betaParse.pl

```
betaProcess.pl
#-----
```

```
use strict; use warnings;
use Net::FTP;
use File::Copy;
use Cwd;
```

Re: question about forked processes writing to the same file

```
use Parallel::ForkManager;

my ($area, $processDate) = @ARGV;

my %fetch;
my $files_are_missing;
my $count = 1;
do {
    &get_fetch(\%fetch, $area, $processDate);

    $files_are_missing = 0;

    if ( ($count % 30) == 0 ) {
        doMail("$0 – slept $count times...each for a minute. maybe some
files are missing.", "", $toList);
    }

    print LOGS localtime() . " – about to start the cycle $count.\n";
    my $redo = &startCycle( $area, 16, \%fetch );
    print LOGS localtime() . " – about to end the cycle $count.\n";

    %fetch = ();

    if ( $redo ) {
        print LOGS localtime() . " – will sleep for a minute because at
least one file was missing.\n";
        $fetch{redo}++;
        sleep 60;
        $files_are_missing = 1;
    }
    $count++;
}
while ($files_are_missing);

print LOGS "Ending $0 at –" . UnixDate('now', '%m/%d/%Y %H:%M:%S') .
"–\n";
close LOGS;

my $time_end = localtime();
print "START: $time_start\n";
print " END: $time_end\n";
doMail("$time_start: $0 – Start => " . localtime() . ": $0 – Done", "",
$toList);

#--- subs ---

sub get_fetch {
my ($fetch_ref, $area, $processDate) = @_ ;
my ($command, @result);
```

Re: question about forked processes writing to the same file

```
if (! keys( %{ $fetch_ref } )) {
print "in get_fetch there were no keys. so NO MISSING FILES\n";
$command = "ssh -2 -l <user> <server>
/export/home/<user>/getFileSize.pl $area $processDate";
}
else {
# process missing files

print "in get_fetch there were keys. so files were missing!\n";
delete $fetch_ref->{redo};
$command = "ssh -2 -l <user> <server>
/export/home/<user>/getFileSize.pl $area.missing $processDate";
}

@result = ` $command`;
for (@result) {
chomp;
my ($file, $size, $time) = split('=>');
$fetch_ref->{$file} = {
time => $time,
size => $size,
};
}
}

sub startCycle {
my ($area, $num_groups, $data_ref) = @_ ;
my %data = %{ $data_ref };

my $server = 'server';
my ($userName, $password) = split /\|/, $mValues{ $server };

my %missing;

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

for my $file ( sort { $data{$b}->{size} <=> $data{$a}->{size} } keys
%data ) {

$pm->start and next;

if ( $data{ $file }->{size} ) {
print "size is TRUE: $file has size - $data{ $file }->{size} - \n";
# fetch and parse

my ($server, $dir, $file_type, $slice, $controlNo, $isCritical) =
split(/\|/, $file );

my $remoteFile = "$dir/$processDate.gz";
my $localFile =
```

Re: question about forked processes writing to the same file

```
"$cycleType{$area}->{rawLogs}/$file_type.$processDate$slice";
#

# Establish FTP Connection:

#

my $ftp;
unless ($ftp = Net::FTP->new($server)) {
doMail("Problem with $0", "Can't connect to $server with ftp,
$@\n");
die;
}
unless ($ftp->login($userName, $password)) {
doMail("Problem with $0", "Can't login to $server with ftp
using -$userName- and -$password- $@\n");
die;
}
$ftp->binary();

if ($ftp->get($remoteFile, $localFile)) {
# print "got $remoteFile to $localFile\n";

my $doneFile = $localFile;
$doneFile =~ s/^.*\//g;
$doneFile =~ s/\.gz$/;
$doneFile = "$cycleType{$area}->{work}/$doneFile";
# Kick Off Parsing for this file:

my $command = "betaParse.pl $processDate $localFile $doneFile
$hash{$area}->{parseMethod}";
system($command);
}
else {
print localtime() . " - FTP MESSAGE: $ftp->message: $@\n";
open( my $fh_missing, '>>', "$area.missing.fetch" ) or die
"can't open $area.missing.fetch: $!\n";
my $missingFile = $file;
$missingFile =~ s/|\//g;
my $controlNo = 1;
my $isCritical = 'Y';
print $fh_missing "$missingFile $controlNo $isCritical\n";
close $fh_missing;
}
$ftp->quit();
}
else {
# Capture missing logs to deal with later

print "size is FALSE: $file has size -$data{$file}->{size}-\n";
```

Re: question about forked processes writing to the same file

Re: question about forked processes writing to the same file

```
$missing{$file} = {
time => scalar(localtime()),
size => $data{$file}->{size},
};
print "$file: $missing{$file}->{time} and
-$missing{$file}->{size}-\n";
open( my $fh_missing, '>>', "$area.missing.fetch" ) or die "can't
open $area.missing.fetch: $!\n";
while( my ($missingFile, $attr) = each %missing ) {
# my ($server, $path, $frontName, $backName) = split(/\//,
$missingFile);
$missingFile =~ s/^\/\//g;
my $controlNo = 1;
my $isCritical = 'Y';
print $fh_missing "$missingFile $controlNo $isCritical\n";
}
close $fh_missing;

}
$pm->finish;
}
$pm->wait_all_children;

my $redo = 0;
if ( -e "$area.missing.fetch" ) {
my $command = "scp -oProtocol=2 $area.missing.fetch
<user>\@<server>:/export/home/<user>/data";
my $src = system($command);

unlink "$area.missing.fetch";
$redo = 1;
}

return $redo;
}

##
# betaParse.pl
##
use strict;
use Cwd;
use Date::Manip;
use File::Basename;
use File::Copy;

my $rawCounts = 0;

my $numOutputFiles = 10;

open(my $fh_in, "gzip -dc $inputFile|") || dieWithMail("Can't open
```

Re: question about forked processes writing to the same file

Re: question about forked processes writing to the same file

```
$inputFile $!\n");

$status = &reformatLogs($fh_in);

close $fh_in;

if ($status == 1) {

system("touch $inputFile.DONE");
}

#---- subs ----

sub reformatLogs {
my ($fh_in) = @_ ;

while( <$fh_in> ) {
$rawCounts++;
chomp;

# process $_

# evenly distribute data to output files
my $section = $rawCounts % $numOutputFiles;

open( my $fh, ">>log$section" ) || die "can't open log$section:
$!\n";
flock( $fh, 2 );
print $fh "$_\n";
close( $fh );
}
return 1;
}
```

....is there a problem because of the mix of forked processes and system calls? perhaps i should change the system call to a function call (after making the necessary code changes)

previously, the open filehandles were at the top of betaParse.pl and there was no locking. this appeared to cause the record splicing, although the processing was about twice as fast. can you shed some light on this phenommenon? you said before that the string length at which writing would start going crazy was around maybe 4096. the records in these weblogs are perhaps a maximum of 10 lines (this is a conservative estimate) on a 1024 x 768 res screen maximized.

.

• *Follow-Ups:*

Re: question about forked processes writing to the same file

Re: question about forked processes writing to the same file

- ◆ **Re: question about forked processes writing to the same file**
◇ From: xhoster

• **References:**

- ◆ **question about forked processes writing to the same file**
◇ From: it_says_BALLS_on_your_forehead
- ◆ **Re: question about forked processes writing to the same file**
◇ From: Gunnar Hjalmarsson
- ◆ **Re: question about forked processes writing to the same file**
◇ From: it_says_BALLS_on_your_forehead
- ◆ **Re: question about forked processes writing to the same file**
◇ From: Gunnar Hjalmarsson
- ◆ **Re: question about forked processes writing to the same file**
◇ From: A. Sinan Unur
- ◆ **Re: question about forked processes writing to the same file**
◇ From: it_says_BALLS_on_your_forehead
- ◆ **Re: question about forked processes writing to the same file**
◇ From: Gunnar Hjalmarsson
- ◆ **Re: question about forked processes writing to the same file**
◇ From: it_says_BALLS_on_your_forehead
- ◆ **Re: question about forked processes writing to the same file**
◇ From: Gunnar Hjalmarsson
- ◆ **Re: question about forked processes writing to the same file**
◇ From: it_says_BALLS_on_your_forehead

- Prev by Date: **Re: replacing backslash**
- Next by Date: **Re: why the perl docs suck**
- Previous by thread: **Re: question about forked processes writing to the same file**
- Next by thread: **Re: question about forked processes writing to the same file**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**