

Presentation on Problem statement

Shubhendra

Directory Structure

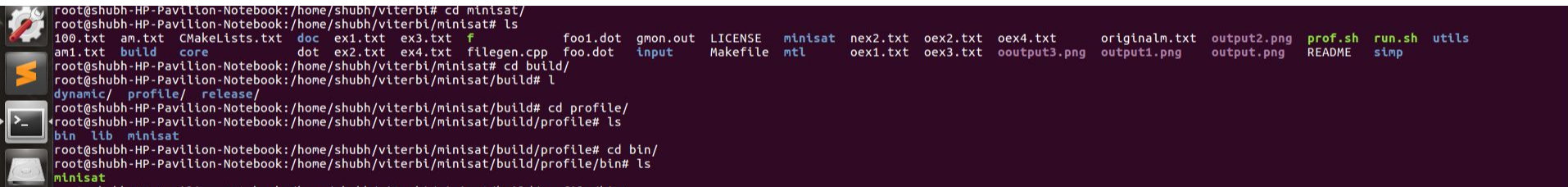
shubh-HP-Pavilion-Notebook: /home/shubh/viterbi/binutils-2.32/gprof

En (97%) 6:54 AM

```
shubh@shubh-HP-Pavilion-Notebook:~$ sudo su
[sudo] password for shubh:
root@shubh-HP-Pavilion-Notebook:/home/shubh# cd m
bash: cd: m: No such file or directory
root@shubh-HP-Pavilion-Notebook:/home/shubh# pwd
/home/shubh
root@shubh-HP-Pavilion-Notebook:/home/shubh# cd viterbi/
root@shubh-HP-Pavilion-Notebook:/home/shubh/viterbi# ls
a.txt  binutils-2.32  gmon.out  minisat  nex2.txt  original.txt  sample  sample.c
root@shubh-HP-Pavilion-Notebook:/home/shubh/viterbi# cd minisat/
root@shubh-HP-Pavilion-Notebook:/home/shubh/viterbi/minisat# ls
100.txt  am.txt  CMakeLists.txt  doc  ex1.txt  ex3.txt  f  foo1.dot  gmon.out  LICENSE  minisat  nex2.txt  oex2.txt  oex4.txt  originalm.txt  output2.png  prof.sh  run.sh  utils
am1.txt  build  core  dot  ex2.txt  ex4.txt  filegen.cpp  foo.dot  input  Makefile  mtl  oex1.txt  oex3.txt  ooutput3.png  output1.png  output.png  README  simp
root@shubh-HP-Pavilion-Notebook:/home/shubh/viterbi/minisat# cd ..
root@shubh-HP-Pavilion-Notebook:/home/shubh/viterbi# cd binutils-2.32/gprof/
root@shubh-HP-Pavilion-Notebook:/home/shubh/viterbi/binutils-2.32/gprof# ls
aarch64.c  bconv.pl  cg_arcs.h  ChangeLog-2004  ChangeLog-2013  config.status  fdl.texi  gen-c-prog.awk  gprof.h  hist.o  mips.o  sparc.c  symtab.o
aarch64.o  bb_exit_func.c  cg_arcs.o  ChangeLog-2005  ChangeLog-2014  config.texi  flat_bl.c  gmon.h  gprof.info  i386.c  po  sparc.o  TEST
aclocal.m4  bsd_callg_bl.c  cg_dfn.c  ChangeLog-2006  ChangeLog-2015  configure  flat_bl.m  gmon_io.c  gprof.o  i386.o  README  stamp-h1  TODO
alpha.c  bsd_callg_bl.m  cg_dfn.h  ChangeLog-2007  ChangeLog-2016  configure.ac  flat_bl.o  gmon_io.h  gprof.texi  libtool  search_list.c  stamp-h.in  utils.c
alpha.o  bsd_callg_bl.o  cg_dfn.o  ChangeLog-2008  ChangeLog-2017  corefile.c  fsf_callg_bl.c  gmon_io.o  hertz.c  MAINTAINERS  search_list.h  sym_ids.c  utils.h
a.txt  call_graph.c  cg_print.c  ChangeLog-2009  ChangeLog-2018  corefile.h  fsf_callg_bl.m  gmon_out.h  hertz.h  Makefile  search_list.o  sym_ids.h  utils.o
basic_blocks.c  call_graph.h  cg_print.h  ChangeLog-2010  ChangeLog-9203  corefile.o  fsf_callg_bl.o  gprof  hertz.o  Makefile.am  source.c  sym_ids.o  vax.c
basic_blocks.h  call_graph.o  cg_print.o  ChangeLog-2011  config.cache  corefile.o  gconfig.h  gprof.1  hist.c  Makefile.in  source.h  symtab.c  vax.o
basic_blocks.o  cg_arcs.c  ChangeLog  ChangeLog-2012  config.log  ex4.txt  gconfig.in  gprof.c  hist.h  mips.c  source.o  symtab.h
root@shubh-HP-Pavilion-Notebook:/home/shubh/viterbi/binutils-2.32/gprof#
```

Command running and explanation

- 1.) `sudo su` - Enables to get access for the directory that needs permissions.
- 2.) Minisat that has been edited has to be configured..meaning that I have to create a minisat exec file from the code written.
 - a.) Make clean
 - b.) Make
 - c.) This will allow to create a build directory with profile folder.

A terminal window with a dark background and light-colored text. The prompt is 'root@shubh-HP-Pavillion-Notebook:'. The user enters 'cd minisat/' and 'ls', showing a directory listing of various files. Then they enter 'cd build/' and 'ls', showing another directory listing. Next, they enter 'cd profile/' and 'ls', showing a third directory listing. Finally, they enter 'cd bin/' and 'ls', showing a fourth directory listing. The terminal output is as follows:

```
root@shubh-HP-Pavillion-Notebook:/home/shubh/viterbi# cd minisat/
root@shubh-HP-Pavillion-Notebook:/home/shubh/viterbi/minisat# ls
100.txt  am.txt  CMakeLists.txt  doc  ex1.txt  ex3.txt  f  foo1.dot  gmon.out  LICENSE  minisat  nex2.txt  oex2.txt  oex4.txt  originalm.txt  output2.png  prof.sh  run.sh  utils
am1.txt  build  core  dot  ex2.txt  ex4.txt  filegen.cpp  foo.dot  input  Makefile  mtl  oex1.txt  oex3.txt  ooutput3.png  output1.png  output.png  README  simp
root@shubh-HP-Pavillion-Notebook:/home/shubh/viterbi/minisat# cd build/
root@shubh-HP-Pavillion-Notebook:/home/shubh/viterbi/minisat/build# ls
dynamic/  profile/  release/
root@shubh-HP-Pavillion-Notebook:/home/shubh/viterbi/minisat/build# cd profile/
root@shubh-HP-Pavillion-Notebook:/home/shubh/viterbi/minisat/build/profile# ls
bin  lib  minisat
root@shubh-HP-Pavillion-Notebook:/home/shubh/viterbi/minisat/build/profile# cd bin/
root@shubh-HP-Pavillion-Notebook:/home/shubh/viterbi/minisat/build/profile/bin# ls
minisat
```

- 3.) The minisat exe is created. Now the minisat along with the input case stored in directory `/home/shubh/` will be executed. So this helps in generating the `gmon.out` file which is the call graph file generated by the execution of the functions performed.
- 4.) The `gmon.out` [1] thus produced can be used to use `gprof`. It needs two files - `gmon.out` and `minisat.exe`

Contd.

5.) Gprof that was edited by us will be ready when we use commands :

Make

Make install

6.) Use this gprof to find the text file which contains the information of what profiling is?

```
root@shubh-HP-Pavilion-Notebook:/home/shubh/viterbi# cd minisat/
```

```
root@shubh-HP-Pavilion-Notebook:/home/shubh/viterbi/minisat# /home/shubh/viterbi/binutils-2.32/gprof/gprof ./build/profile/bin/minisat -d gmon.out > ex4.txt
```

```
root@shubh-HP-Pavilion-Notebook:/home/shubh/viterbi/minisat# subl ex4.txt
```

Gmon.out[1]

The profile data is taken from the call graph profile file (**gmon.out** by default) created by programs that are compiled with the **cc** command by using the **-pg** option. The **-pg** option also links in versions of library routines that are compiled for profiling, and reads the symbol table in the named object file (**a.out** by default), correlating it with the call graph profile file. If more than one profile file is specified, the **gprof** command output shows the sum of the profile information in the specified profile files.

The **-pg** option causes the compiler to insert a call to the **mcount** subroutine into the object code that is generated for each recompiled function of your program. During program execution, each time a parent calls a child function the child calls the **mcount** subroutine to increment a distinct counter for that parent-child pair. Programs that are not recompiled with the **-pg** option do not have the **mcount** subroutine, and therefore keep no record of who called them [1].

Three kinds of gprof

Version 0 : The original version.

Version 1 : The version containing the print statements for analyzing but not changing any of the bin_count i.e. time.

Version 2 : The version containing the change of time.

Series of commands used

oex4 represents the version1 gprof result and **ex4** represents the version2 gprof result.

FOR OEX4.TXT[3] : `cd binutils-2.32/gprof/`

`make`

`make install`

Inside the minisat directory - `./build/profile/bin/minisat /home/shubh/ex4.cnf`

`/home/shubh/viterbi/binutils-2.32/gprof/gprof ./build/profile/bin/minisat -d gmon.out > oex4.txt`

FOR EX4.TXT[3] : `cd ../binutils-2.32/gprof/`

`make`

`make install`

`/home/shubh/viterbi/binutils-2.32/gprof/gprof ./build/profile/bin/minisat -d gmon.out > ex4.txt`

Example[3]

Taken from SAT 2011 competition. The following competition was divided into industrial and manual one. One example is from manual and one is from industrial. I will present the industrial one here in the presentation. I wish to change the branch of addclause()-----> propagate() bcz we have to change the self time of propagate().

Assume bin_count assigned is 1.

Gprof results (version 1) [3]:

Id = 133 to 251....so the total time taken is : 11.49 sec.

Prop() time is 8.82 taking 76%.

Gprof results (version 2) [3]:

Total time taken is : 3.9 sec.

Prop() time is 1.24 taking 31.59%. The total bin_count was 880 from 133 to 251.

Analysis of the Result

Explanation behind the example is :

Total time changed is = (total time in between bin 133-251) - (bin_count)*(251-133+1) = 880 - 1*(252-133) = 880-119 = 761 i.e. 7.61 sec difference. So seeing the propagate time should be 8.82 - 7.61 = 1.21 sec.

Total time should be 11.49 - 7.61 i.e. 3.88 sec i.e. the % time contributed is : 31%

CLOSE ENOUGH!!!

Hence profiling is correct if you identify the bins correctly.

References

- 1.) https://www.ibm.com/support/knowledgecenter/en/ssw_aix_72/com.ibm.aix.cmds2/gprof.htm
- 2.) <http://www.satcompetition.org/>
- 3.) https://github.com/singhalshubh/SAT_Parallelism