Software Model Checking and Debian - better together

Michael Tautschnig (mt)
Introducing myself as this is my first Debian (Mini-)conf

- DD since 05/12/2007
- Packages (co-)maintained mainly relate to my academic activities: BrickOS & friends (Lego Mindstorms), SAT solvers, CBMC, and some others
- When time permitted: debian-mentors/sponsorship

- Main interests: software quality and automation
  - Passion for quality is main driver of this work
Writing correct code is easy: Bubble Sort

```c
void bubble(int a[], int N) {
    int i, j, t;

    for (i = N; i >= 0; i--) {
        for (j = 2; j <= 1; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}
```
Writing correct code is easy: Bubble Sort

```c
void bubble(int a[], int N) {
    int i, j, t;
    for (i = N; i >= 0; i--) {
        for (j = 2; j <= 1; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}
```
Writing correct code is easy: Bubble Sort

```c
void bubble(int a[], int N) {
    int i, j, t;

    for (i = N; i >= 0; i--) {
        for (j = 2; j <= N; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}
```
Writing correct code is easy: Bubble Sort

```c
void bubble(int a[], int N) {
    int i, j, t;
    for (i = N; i >= 0; i--) {
        for (j = 2; j <= 1; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}
```
void bubble(int a[], int N) {
    int i, j, t;
    for (i = N; i >= 0; i--) {
        for (j = 2; j <= 1; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}

int main(int argc, char* argv[]) {
    int a[5];
    int i;
    printf("to sort:");
    for (i = 0; i < 5; ++i)
        printf(" %d", a[i]);
    printf("\n");
    bubble(a, 5);
    printf("sorted:");
    for (i = 0; i < 5; ++i)
        printf(" %d", a[i]);
    printf("\n");
    return 0;
}
void bubble(int a[], int N) {
    int i, j, t;
    for (i = N; i >= 0; i--) {
        for (j = 2; j <= 1; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}

int main(int argc, char* argv[]) {
    int a[5] = { 1, 5, 3, 5, 2 };
    int i;

    printf("to sort:");
    for (i = 0; i < 5; ++i)
        printf(" %d", a[i]);
    printf("\n");

    bubble(a, 5);

    printf("sorted:");
    for (i = 0; i < 5; ++i)
        printf(" %d", a[i]);
    printf("\n");

    return 0;
}
Manual Testing

```c
34 int main(int argc, char* argv[]) {
35     int a[SIZE] = { 1, 5, 3, 5, 2 };  
36     int i;
37
38     printf("to sort:");
39     for (i = 0; i < SIZE; ++i)
40         printf(" %d", a[i]);
41     printf("\n");
42
43     bubble(a, SIZE);
44
45     printf("sorted:");
46     for (i = 0; i < SIZE; ++i)
47         printf(" %d", a[i]);
48     printf("\n");
49
50     assert(isSorted(a, SIZE));
51
52     return 0;
53 }
54 ```
Bubble Sort - Fixing one bug

```c
void bubble(int a[], int N) {
    int i, j, t;

    for (i = N; i >= 0; i--) {
        for (j = 2; j <= 1; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}
```
void bubble(int a[], int N) {
    int i, j, t;

    for (i = N; i >= 0; i--) {
        for (j = 2; j <= i; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}
Automated Test-input Generation

```
$ ./fshell_bubble.c --tco-location --verbosity 0 | tee /dev/stderr | ./C-Unit_Generator.pl
```
void bubble(int a[], int N) {
    int i, j, t;

    for (i = N; i >= 0; i--) {
        for (j = 2; j <= i; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}
Bubble Sort - Another bug fixed

```c
void bubble(int a[], int N) {
    int i, j, t;

    for (i = N; i >= 0; i--) {
        for (j = 1; j <= i; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}
```
Model Checking

```c
#include <stdlib.h>
#include <stdio.h>
#include <assert.h>

#ifndef SIZE
#define SIZE 5
#endif

int isSorted(int a[], int N);

void bubble(int a[], int N) {
    int i, j, t;
    for (i = N; i >= 0; i--) {
        for (j = i + 1; j < N; j++) {
            if (a[j - 1] > a[j]) {
                t = a[j - 1];
                a[j - 1] = a[j];
                a[j] = t;
            }
        }
    }
}

int isSorted(int a[], int N) {
    int i;
    for (i = 0; i < N - 1; i++) {
        if (a[i] > a[i + 1]) {
            return 0;
        }
    }
    return 1;
}
```

When testing isn’t useful anymore ...

```c
volatile unsigned x = 0, y = 0;
volatile unsigned r1 = 0, r2 = 0;

void* A(void* arg) {
    x = 1;
    r1 = y + 1;
}

void* B(void* arg) {
    y = 1;
    r2 = x + 1;
}

void main() {
    pthread_create(0, 0, A, 0);
    pthread_create(0, 0, B, 0);
    assert(!(r1 == 1 && r2 == 1));
}
```
When testing isn’t useful anymore ...

```c
volatile unsigned x = 0, y = 0;
volatile unsigned r1 = 0, r2 = 0;

void* A(void* arg) {
    x = 1;
    r1 = y + 1;
}

void* B(void* arg) {
    y = 1;
    r2 = x + 1;
}

void main(){
    pthread_create(0, 0, A, 0);
    pthread_create(0, 0, B, 0);
    assert(!(r1 == 1 && r2 == 1));
}
```

Assertion `!(r1 == 1 && r2 == 1)` failed.
When testing isn’t useful anymore ...

```c
volatile unsigned x = 0, y = 0;
volatile unsigned r1 = 0, r2 = 0;

void* A(void* arg) {
    x = 1;
    r1 = y + 1;
}

void* B(void* arg) {
    y = 1;
    r2 = x + 1;
}

void main()
{
    pthread_create(0, 0, A, 0);
    pthread_create(0, 0, B, 0);
    assert(!(r1 == 1 && r2 == 1));
}
```

Assertion `!(r1 == 1 && r2 == 1)' failed. ... in 0.1-10% of all test runs
What is Software Model Checking

• Fully automatic method
• May provide proofs of correctness
• Input:
  • Specification: `assert(x!=0);`
  • Model: source code
• Output:
  • “yes” (specification always holds)
  • “no” + counterexample (specification can be violated)

• Main academic problem: scalability
• Practical problem: making tools work (on real code)
Applying CBMC to this piece of code

```
$ cbmc/cbmc --mm sc mp.c
```
Applying CBMC to this piece of code

```
$ cbmc/cbmc --mm sc mp.c
```
Applying CBMC to this piece of code

```bash
$ cbmc/cbmc --mm tso mp.c
```
What could Software Model Checking do for us?

Re: max_wal_senders must die

From: Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
To: Robert Haas <robertmhaas(at)gmail(dot)com>
Cc: Bruce Momjian <bruce(at)momjian(dot)us>, Josh Berkus <josh(at)agliodbs(dot)com>, hackers(at)postgresql(dot)org
Subject: Re: max_wal_senders must die
Date: 2010-11-13 15:07:21
Message-ID: 24987.1289660841@sss.pgh.pa.us (view raw or flat)
Thread: 2010-11-13 15:07:21 from Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
Lists: postgresql-hackers

> Come to think of it, I'm not really sure I understand what protects
> SetLatch() against memory ordering hazards. Is that actually safe?

Hmm ... that's a good question. It certainly *looks* like it could
malfunction on machines with weak memory ordering.

regards, tom lane
What could Software Model Checking do for us?

Yes, WaitLatch is vulnerable to weak-memory-ordering bugs

From: Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
To: postgresql-hackers(at)postgresql(dot)org
Subject: Yes, WaitLatch is vulnerable to weak-memory-ordering bugs
Date: 2011-08-07 17:47:49
Message-ID: 24241.1312739269@sss.pgh.pa.us (view raw or flat)
Thread: 2011-08-07 17:47:49 from Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
Lists: postgresql-hackers

I suspected $SUBJECT from the beginning, and I've now put in enough work to be able to prove it. The attached test program reliably fails within a few minutes of being started, when run with 8 worker processes on an 8-core PPC machine. It's a pretty simple "token passing ring" protocol, and at some point one of the processes sees its latch set without seeing its flag set, so it goes back to sleep and the token stops getting passed.

regards, tom lane
What could Software Model Checking do for us?

Yes, WaitLatch is vulnerable to weak-memory-ordering bugs

```c
void ResetLatch(volatile Latch *latch)
{
    /* Only the owner should reset the latch */
    Assert(latch->owner_pid == MyProcPid);
    latch->is_set = false;

    /* XXX there really ought to be a memory barrier operation right here, to
    * ensure that the write to is_set gets flushed to main memory before we
    * examine any flag variables. Otherwise a concurrent SetLatch might
    * falsely conclude that it needn't signal us, even though we have missed
    * seeing some flag updates that SetLatch was supposed to inform us of.
    * For the moment, callers must supply their own synchronization of flag
    * variables (see latch.h).
    */
}
```

and at some point one of the processes sees its latch set without seeing its flag set, so it goes back to sleep and the token stops getting passed.

regards, tom lane
What could Software Model Checking do for us?

---

Re: Weak-memory specific problem in ResetLatch/WaitLatch (follow-up analysis)

From: Michael Tautschnig <mt(at)debian(dot)org>
To: postgresql-hackers(at)postgresql(dot)org
Cc: Jade Alglave <jade(dot)alglave(at)cs(dot)ox(dot)ac(dot)uk>, Vincent Nimal <vincent(dot)nimal(at)balliol(dot)ox(dot)ac(dot)uk>, Daniel Kroening <kroening(at)cs(dot)ox(dot)ac(dot)uk>

Subject: Re: Weak-memory specific problem in ResetLatch/WaitLatch (follow-up analysis)
Date: 2012-03-24 17:01:32
Message-ID: 20120324170131.GB8779@l04.local (view raw or flat)

In summary, we were thus able to show that both points marked with "XXX there really ought to be a memory barrier" in

http://git.postgresql.org/gitweb/?p=postgresql.git;a=commitdiff;h=4e15a4db5e65e43271f8d20750d6500ab12632d0

are the appropriate points to place memory synchronisation primitives, and picking an lwsync-equivalent in both cases is sound and does not require any other modifications.

Best,
Michael

Software Model Checking and Debian - better together | Michael Tautschnig (mt)
What works at this stage?

- Other medium-scale experiments: proving the need of a barrier in read-copy-update in the Linux kernel

- More experiments required...

- In general: first step is successful compilation (and linking) in a way suitable for the tools

- How to automate experiments at large scale?
What can Debian do for Software Model Checking

- Linux distributions enable experiments at large scale
  - Wheezy has more than 400 million LOC
    - [http://blog.james.rcpt.to/2012/02/13/debian-wheezy-us19-billion-your-price-free/](http://blog.james.rcpt.to/2012/02/13/debian-wheezy-us19-billion-your-price-free/)
  - Broad range of ports makes Debian even more interesting

- In particular: uniform build system

- Great infrastructure such as sources.debian.net
For example: analysing 200 million LOC for potential weak memory susceptibility
For example: analysing 200 million LOC for potential weak memory susceptibility
First steps

• Compiling packages using goto-cc
  • goto-cc builds intermediate-representation object files for CBMC/CProver tools
  • goto-cc accepts (most of) gcc’s options
• Sanity check: dumping intermediate representation back as C code (using goto-instrument)

• Both goto-cc and goto-instrument are part of the cbmc package
Initial Experiments

- Following http://www.hermann-uwe.de/blog/rebuilding-the-whole-debian-archive-using-the-open64-compiler

- Using cowbuilder/pbuilder

- gcc and ld in chroot replaced by bash script

- Running (multiple) buildall instances (pbuilder package)

- Mostly works using sudo

- Debugging sometimes requires root access

- Scripts, notes: https://github.com/tautschnig/cprover-debian
Package builds using goto-cc

- goto-cc builds intermediate representation, not executable
- Link-time type checking
- Scripts replacing link dest of /usr/bin/gcc, /usr/bin/ld

1. Run real gcc/ld
2. Parse selected options (e.g. find output file name)
3. Compile/link using goto-cc and add result as additional ELF section

- Resulting file remains executable
- Stable under file renaming
Some future Improvements

- Currently only C front-end sufficiently complete
  - No support for nested functions (like Clang)

- ~700 packages FTBFS

- Should install build depends from local build
  - goto-cc sections only present in package-local libraries, not in build deps
goto-instrument --dump-c

- Intermediate representation only has goto for control flow
- Maintains variable names, source code locations

- Human-readable C code constructed
  - Loops, if/else, switch restored
  - Formatted

- Tested for compilation and convergence

- Many failures caused by missing function declarations
Jenkins Setup
Jenkins Setup

• One job per tool and package, generated using job-dsl plug-in
• debile didn’t yet exist 1 year ago ...

• Pros:
  • Easy to set up
  • Rich collection of plug-ins (job-dsl, bulk builder, claims)
  • Master/slave support
  • Usable by non-tech users

• Cons:
  • Limited scalability (~36000 jobs!)
  • Java web application uses ~17GB RAM
  • Some pages put heavy strain on browser
Observations

• Verbose build logs would be valuable for debugging

• Reporting bugs:
  • What should be considered a bug?
  • (Almost) all bugs are upstream errors
    • Need account with all upstream BTS’
    • Upstream dead/no BTS
  • [Link](http://bugs.debian.org/cgi-bin/pkgreport.cgi?users=mt@debian.org&tag=goto-cc&archive=both)
Reporting bugs

Debian Archived Bug report logs: Bugs tagged goto-cc

- Outstanding bugs -- Critical bugs; Unclassified (1 bug)
  - #702089 [CLI] Passes literal struct instead of pointer-to-struct
- Outstanding bugs -- Important bugs; Unclassified (2 bugs)
- Outstanding bugs -- Normal bugs: Patch Available (1 bug)
- Outstanding bugs -- Normal bugs: Normal (78 bugs)
- Outstanding bugs -- Normal bugs: Minor bugs: Unclassified (6 bugs)
- Outstanding bugs -- Wihlist items: Unclassified (1 bug)
- Forwarded bugs -- Normal bugs (8 bugs)
- Forwarded bugs -- Minor bugs (1 bug)
- Resolved bugs -- Grave functionality bugs (1 bug)
- Resolved bugs -- Serious (policy violations or makes package unfit for release) (9 bugs)
- Resolved bugs -- Normal bugs (25 bugs)
- Resolved bugs -- Minor bugs (3 bugs)

Outstanding bugs-- Critical bugs; Unclassified (1 bug)

- #702089 [CLI] Passes literal struct instead of pointer-to-struct

Outstanding bugs-- Important bugs; Unclassified (2 bugs)

- #688785 [xbmc] xbm: Fatal: can’t open /dev/random: Bad address
- #722511 [lilo] lilo: FTBFS: maximum path length limited to 65 chars

Outstanding bugs-- Normal bugs: Patch Available (1 bug)

- #697851 [lilo] Use of nested functions in config check

Outstanding bugs-- Normal bugs: Unclassified (78 bugs)

- #684508 [lilo] Use of nested functions in config check
- #684509 [am-utils] Configure check uses single-argument main function
- #688381 [lilo] Lilo bug: Wrong order of arguments to gdp_unit_conversion
- #688386 [lilo] Porting declaration of rl last_func
- #688386 [lilo] [lilo] Use-read-gnu-perl] Confliting declaration of rl last_func
- #688387 [lilo] Use-nameset] Missing entries in Unicoded Block
Examples of Errors: rsync

file main.c line 58: error: conflicting types for variable `c::curr_dir_len'
old definition in module exclude file exclude.c line 41
  unsigned int
new definition in module main file main.c line 58
  signed int
Examples of Errors: yp-svipc

file ywrap.c line 81: error: conflicting types for variable `c::svipc_debug'
old definition in module yorick_svipc file../common/svipc_misc.h line 52
  signed int
new definition in module ywrap file ywrap.c line 81
  char [41]
Examples of Errors: xtux

file menu.c line 28: error: conflicting types for variable `c::num_entity_types'
old definition in module main file main.c line 23
    unsigned char
new definition in module menu file menu.c line 28
    signed int
Examples of Errors: simh

file PDP18B/pdp18b_fpp.c line 146: error: conflicting types for variable `c::pcq'
old definition in module pdp18b_cpu file PDP18B/pdp18b_cpu.c line 374
  **signed short int** [641]
new definition in module pdp18b_fpp file PDP18B/pdp18b_fpp.c line 146
  **signed int** [641]
Examples of Errors: xrdp

/bin/bash ../../../libtool --tag=CC --mode=link gcc -DXRDP_CFG_PATH="/etc/xrdp" -DXRDP_SBIN_PATH="/usr/sbin" -DXRDP_SHARE_PATH="/usr/share/xrdp" -DXRDP_PID_PATH="/var/run/xrdp" -g -O2 -o xrdp-sesrun sesrun.o tcp.o config.o ../../../common/libcommon.la
libtool: link: gcc -DXRDP_CFG_PATH="/etc/xrdp" -DXRDP_SBIN_PATH="/usr/sbin" -DXRDP_SHARE_PATH="/usr/share/xrdp" -DXRDP_PID_PATH="/var/run/xrdp" -g -O2 -o ../../../libs/xrdp-sesrun sesrun.o tcp.o config.o ../../../common/libs/libcommon.so -Wl,-rpath -Wl,../../../usr/lib/xrdp

file config.c line 33: error: conflicting types for variable `c::g_cfg'
old definition in module sesrun file sesrun.c line 33
struct config_sesman {
  char [32l] listen_address;
  char [16l] listen_port;
  signed int enable_user_wm;
  char [32l] default_wm;
  char [32l] user_wm;
  unsigned int $pad0;
  char * auth_file_path;
  struct list * vnc_params;
  struct list * rdp_params;
  struct log_config log;
  struct config_security sec;
  struct config_sessions sess;
}

new definition in module config file config.c line 33
struct config_sesman {
  char [32l] listen_address;
  char [16l] listen_port;
  signed int enable_user_wm;
  char [32l] default_wm;
  char [32l] user_wm;
  unsigned int $pad0;
  char * auth_file_path;
  struct list * vnc_params;
  struct list * rdp_params;
  struct log_config log;
  struct config_security sec;
  struct config_sessions sess;
} *
Probably not an error: tiemu

Old definition in module calc file gui/calc/calc.c line 81

```c
struct #anon#ST[S32'x'||S32'y'||S32'w'||S32'h'|] {
    signed int x;
    signed int y;
    signed int w;
    signed int h;
}
```

New definition in module screen file gui/calc/screen.c line 77

```c
union #anon#UN[SYM:c::tag-_anon#ST[S32'x'||S32'y'||S32'w'||S32'h'|]|'wr'||SYM:c::tag-_GdkRectangle'|'gr'|] {
    struct #anon#ST[S32'x'||S32'y'||S32'w'||S32'h'|] wr;
    struct _GdkRectangle gr;
}
```
Questions to you

- Where can I find the most formal specification of linking?
- How to make diagnosing more efficient?
- Is this considered useful?