- AEP -
Automatic Exchange of Embedded Software Parameters

Rita de Cássia Cazu Soldi
rita@lisha.ufsc.br

Antônio Augusto Medeiros Fröhlich
guto@lisha.ufsc.br

Agenda

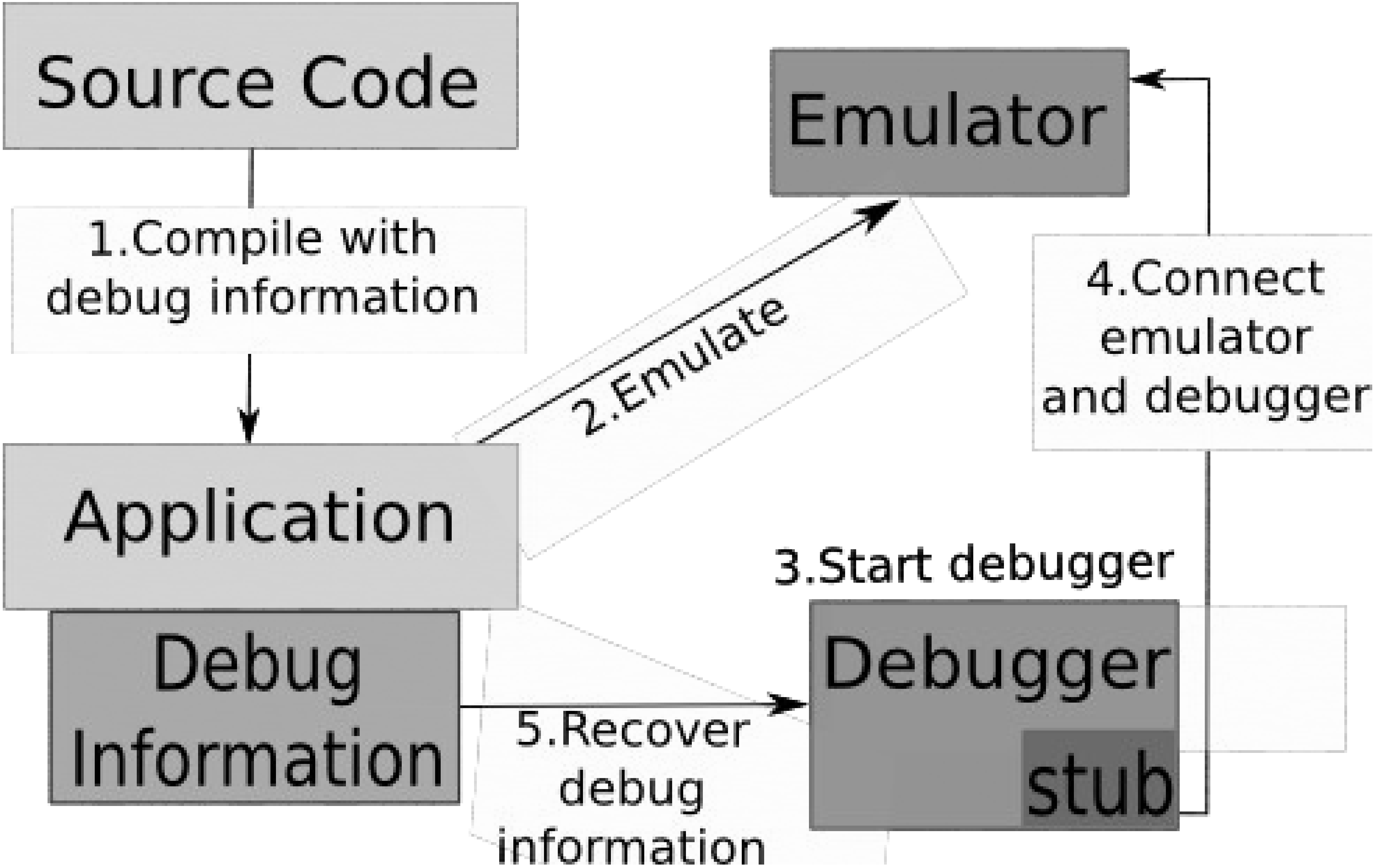
- Introduction
  - The problem of testing
- Embedded systems environment
  - Support for testing and debugging
- Automatic exchange of embedded system software parameters
  - General algorithm and implementation details
- Results
  - Classification of configurations, execution time and size of additional information
- Conclusion and Future work

Introduction

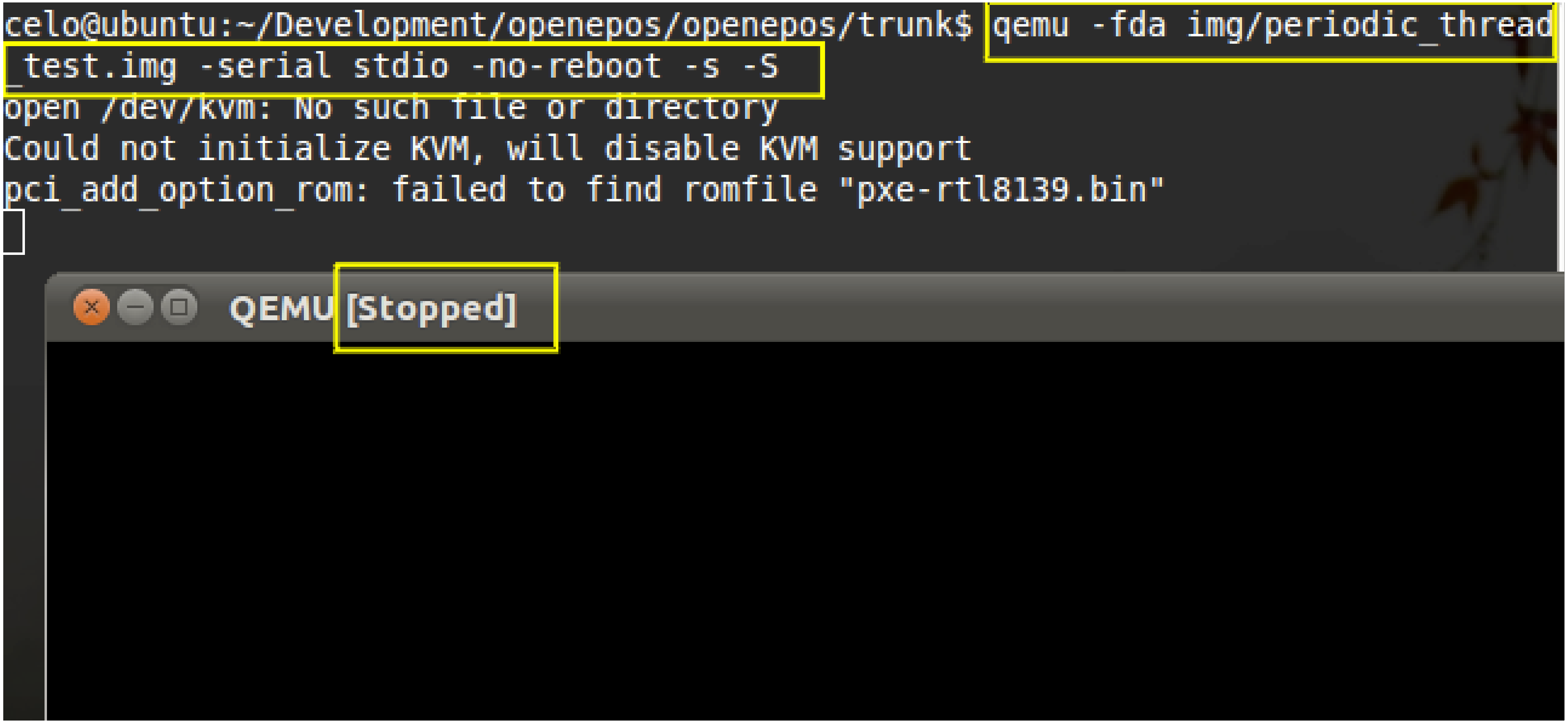
- Embedded system
  - Combination of software and hardware designed to perform a specific task
  - Popular and intrinsic to humans daily life

- Software testing
  - Analyse and detect the differences between existing and required conditions
  - Non-trivial and time-consuming
  - Embedded software testing should also consider software/hardware integration
  - Once a test fail, debugging can locate/identify bugs and then correct the software.

Embedded systems debugging environment

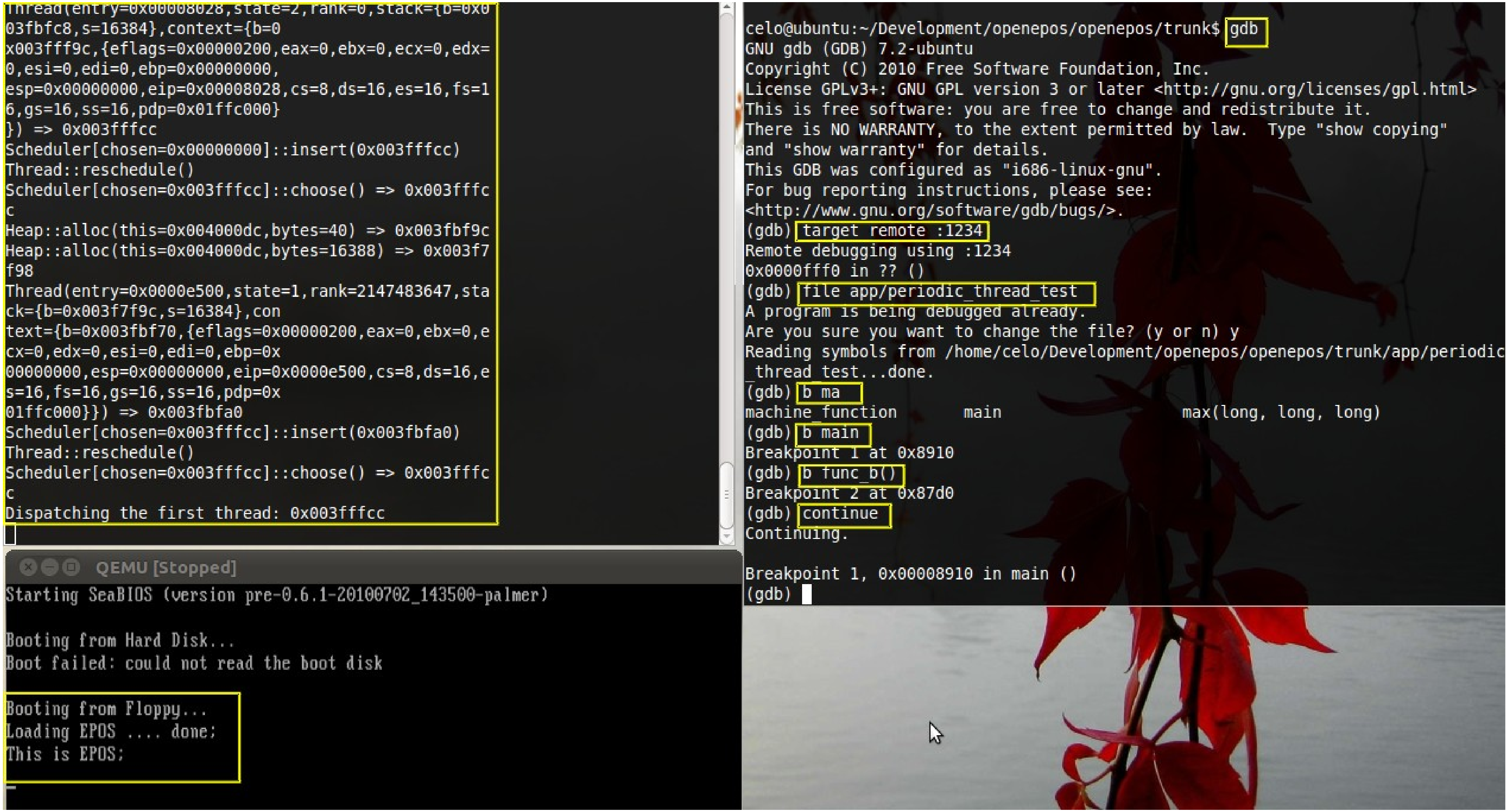

Source Code
1.Compile with debug information
Application
Debug Information
2.Emulate
Emulator
3.Start debugger
Debugger
stub
4.Connect emulator and debugger
5.Recover debug information

Embedded systems debugging environment



```
celo@ubuntu:~/Development/openepos/openepos/trunk$ qemu -fda img/periodic_thread
_test.img -serial stdio -no-reboot -s -S
open /dev/kvm: No such file or directory
Could not initialize KVM, will disable KVM support
pci_add_option_rom: failed to find romfile "pxe-rtl8139.bin"
```

Embedded systems debugging environment



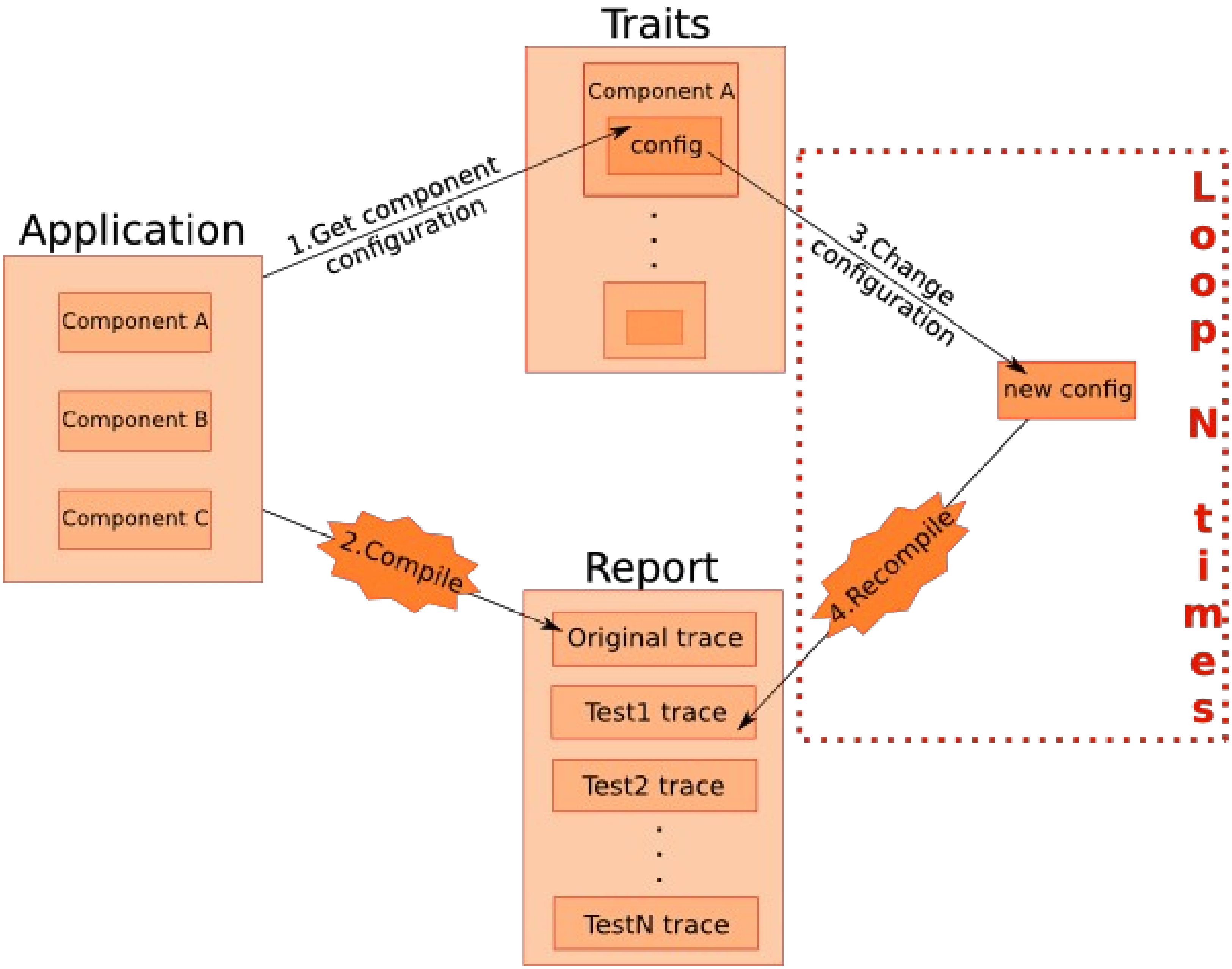
```
Thread(entry=0x00008028,state=2,rank=0,stack={b=0x0
03fbfc8,s=16384},context={b=0
x003fff9c,{eflags=0x00000200,eax=0,ebx=0,ecx=0,edx=
0,esi=0,edi=0,ebp=0x00000000,
esp=0x00000000,eip=0x00008028,cs=8,ds=16,es=16,fs=1
6,gs=16,ss=16,pdp=0x01ffc000}
}) => 0x003fffcc
Scheduler[chosen=0x00000000]::insert(0x003fffcc)
Thread::reschedule()
Scheduler[chosen=0x003fffcc]::choose() => 0x003fffc
c
Heap::alloc(this=0x004000dc,bytes=40) => 0x003fbf9c
Heap::alloc(this=0x004000dc,bytes=16388) => 0x003f7
f98
Thread(entry=0x0000e500,state=1,rank=2147483647,sta
ck={b=0x003f7f9c,s=16384},con
text={b=0x003fbf70,{eflags=0x00000200,eax=0,ebx=0,e
cx=0,edx=0,esi=0,edi=0,ebp=0x
00000000,esp=0x00000000,eip=0x0000e500,cs=8,ds=16,e
s=16,fs=16,gs=16,ss=16,pdp=0x
01ffc000}}) => 0x003fbfa0
Scheduler[chosen=0x003fffcc]::insert(0x003fbfa0)
Thread::reschedule()
Scheduler[chosen=0x003fffcc]::choose() => 0x003fffc
c
Dispatching the first thread: 0x003fffcc
```

QEMU [Stopped]

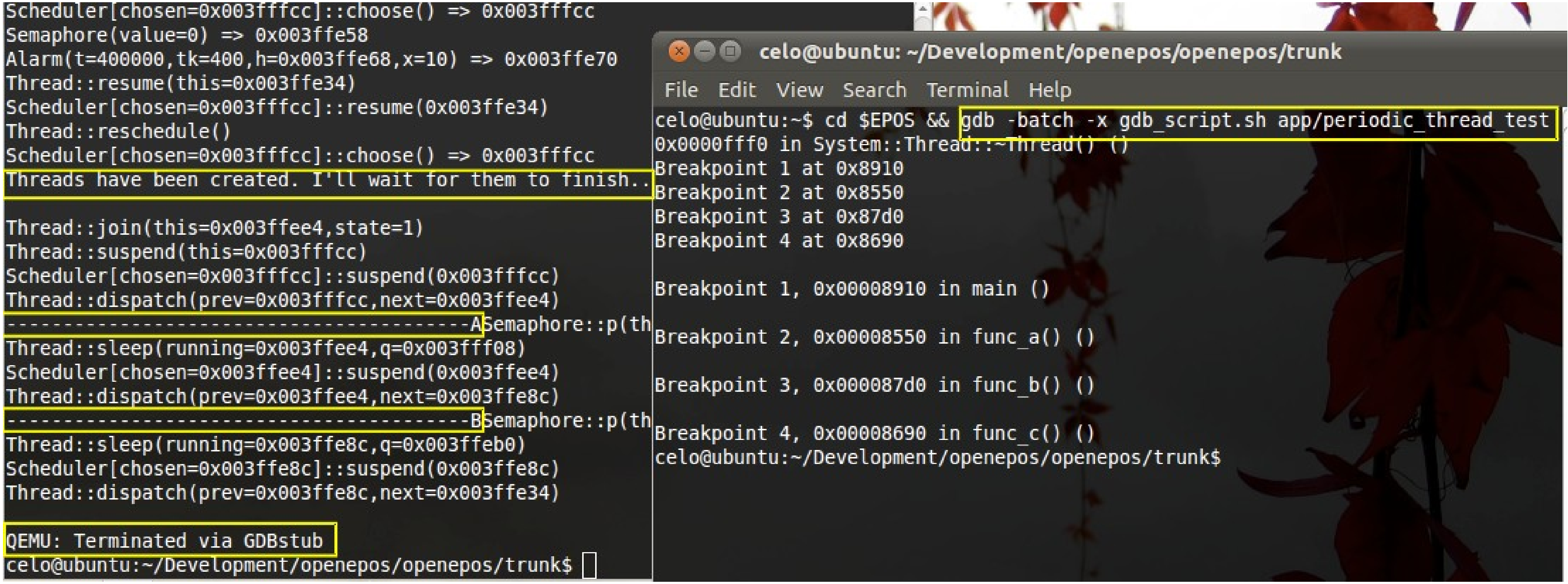
```
Starting SeaBIOS (version pre-0.6.1-20100702_143500-palmer)

Booting from Hard Disk...
Boot failed: could not read the boot disk

Booting from Floppy...
Loading EPOS .... done;
This is EPOS;
_
```

```
celo@ubuntu:~/Development/openepos/openepos/trunk$ gdb
GNU gdb (GDB) 7.2-ubuntu
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
(gdb) target remote :1234
Remote debugging using :1234
0x0000fff0 in ?? ()
(gdb) file app/periodic_thread_test
A program is being debugged already.
Are you sure you want to change the file? (y or n) y
Reading symbols from /home/celo/Development/openepos/openepos/trunk/app/periodic
_thread_test...done.
(gdb) b ma
machine_function       main                     max(long, long, long)
(gdb) b main
Breakpoint 1 at 0x8910
(gdb) b func_b()
Breakpoint 2 at 0x87d0
(gdb) continue
Continuing.

Breakpoint 1, 0x00008910 in main ()
(gdb) 
```

Automatic exchange of software parameters



Automatic exchange of software parameters



```
Scheduler[chosen=0x003fffcc]::choose() => 0x003fffcc
Semaphore(value=0) => 0x003ffe58
Alarm(t=400000,tk=400,h=0x003ffe68,x=10) => 0x003ffe70
Thread::resume(this=0x003ffe34)
Scheduler[chosen=0x003fffcc]::resume(0x003ffe34)
Thread::reschedule()
Scheduler[chosen=0x003fffcc]::choose() => 0x003fffcc
Threads have been created. I'll wait for them to finish..

Thread::join(this=0x003ffee4,state=1)
Thread::suspend(this=0x003fffcc)
Scheduler[chosen=0x003fffcc]::suspend(0x003fffcc)
Thread::dispatch(prev=0x003fffcc,next=0x003ffee4)
-----------------------------------------ASemaphore::p(th
Thread::sleep(running=0x003ffee4,q=0x003fff08)
Scheduler[chosen=0x003ffee4]::suspend(0x003ffee4)
Thread::dispatch(prev=0x003ffee4,next=0x003ffe8c)
-----------------------------------------BSemaphore::p(th
Thread::sleep(running=0x003ffe8c,q=0x003ffeb0)
Scheduler[chosen=0x003ffe8c]::suspend(0x003ffe8c)
Thread::dispatch(prev=0x003ffe8c,next=0x003ffe34)

QEMU: Terminated via GDBstub
celo@ubuntu:~/Development/openepos/openepos/trunk$
```

celo@ubuntu: ~/Development/openepos/openepos/trunk
File Edit View Search Terminal Help

```
celo@ubuntu:~$ cd $EPOS && gdb -batch -x gdb_script.sh app/periodic_thread_test
0x0000fff0 in System::Thread::~Thread() ()
Breakpoint 1 at 0x8910
Breakpoint 2 at 0x8550
Breakpoint 3 at 0x87d0
Breakpoint 4 at 0x8690

Breakpoint 1, 0x00008910 in main ()

Breakpoint 2, 0x00008550 in func_a() ()

Breakpoint 3, 0x000087d0 in func_b() ()

Breakpoint 4, 0x00008690 in func_c() ()
celo@ubuntu:~/Development/openepos/openepos/trunk$
```

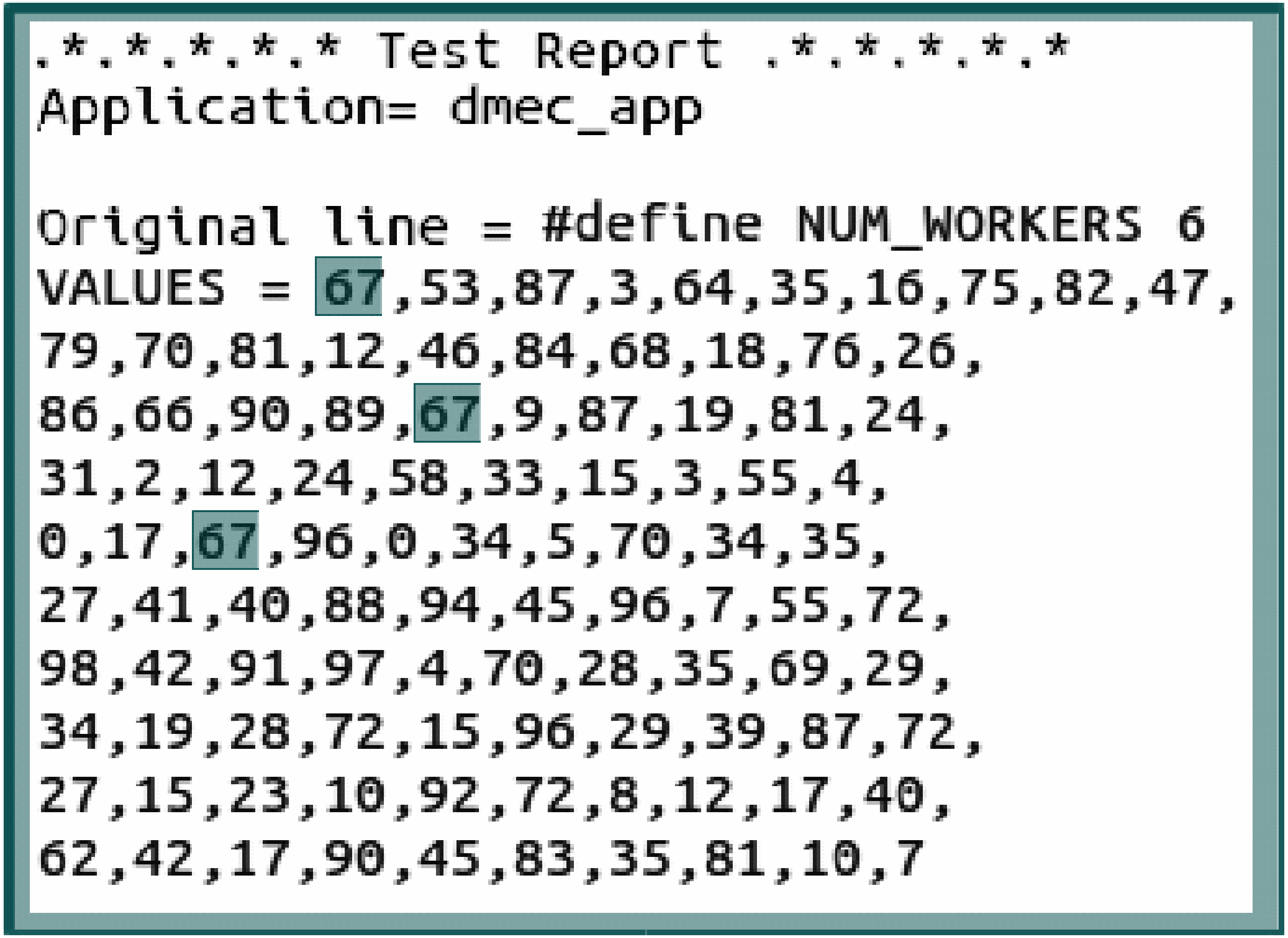
AEP - Automatic exchange of embedded system software parameters

- Requirements
  - Application-Oriented System Design
  - Feature Based Model
  - Parametric Model

- Granularity of configuration
  - Random (worst-case)
  - Partially random
  - Determined

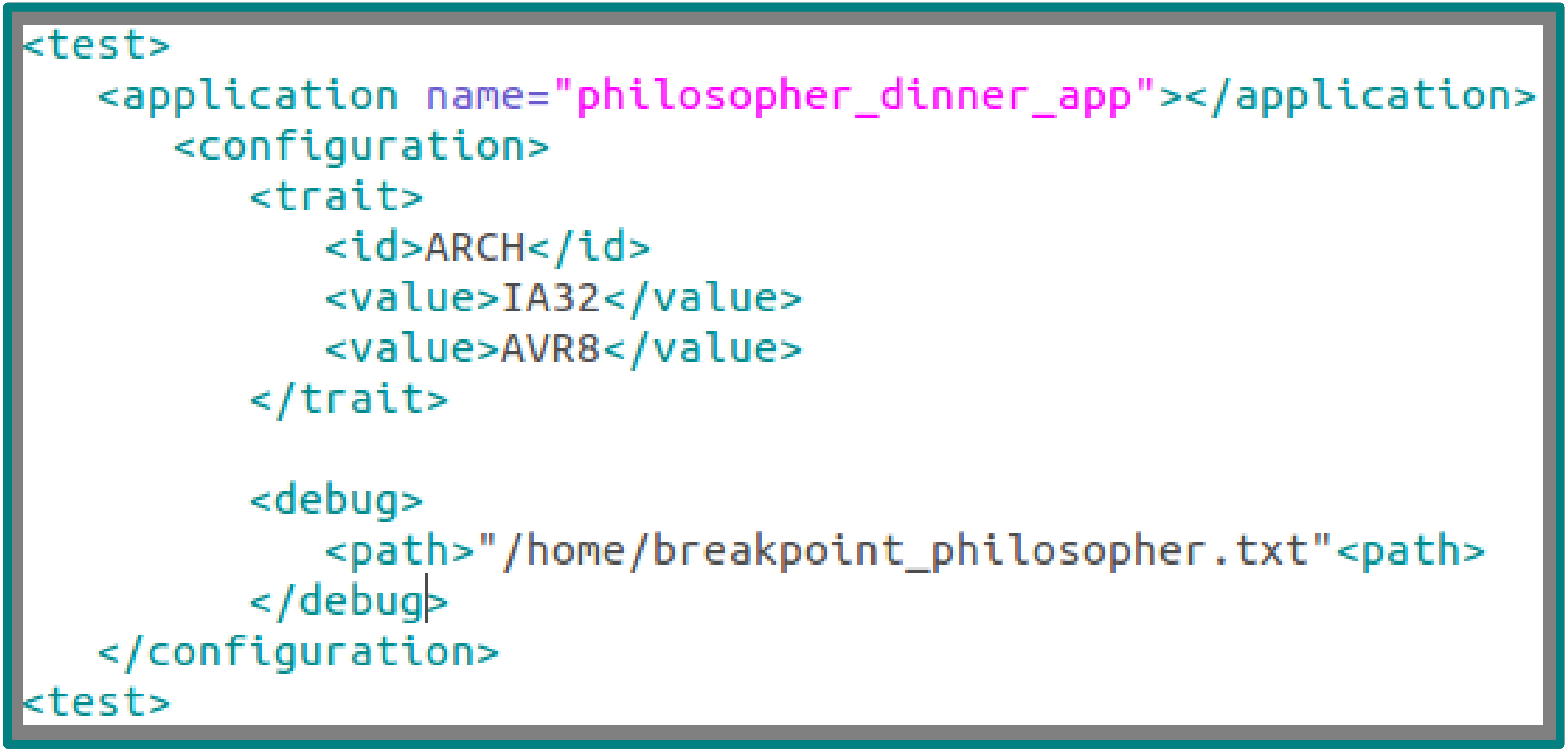
- Definition of exchange success
  - No compilation errors
  - Is there a register on the exchange on the report
  - No difference between traces logs

AEP - Automatic exchange of embedded system software parameters



```
.*.*.*.*.* Test Report .*.*.*.*.*
Application= dmec_app

Original line = #define NUM_WORKERS 6
VALUES = 67,53,87,3,64,35,16,75,82,47,
79,70,81,12,46,84,68,18,76,26,
86,66,90,89,67,9,87,19,81,24,
31,2,12,24,58,33,15,3,55,4,
0,17,67,96,0,34,5,70,34,35,
27,41,40,88,94,45,96,7,55,72,
98,42,91,97,4,70,28,35,69,29,
34,19,28,72,15,96,29,39,87,72,
27,15,23,10,92,72,8,12,17,40,
62,42,17,90,45,83,35,81,10,7
```

AEP - Automatic exchange of embedded system software parameters



```
<test>
   <application name="philosopher_dinner_app"></application>
      <configuration>
         <trait>
            <id>ARCH</id>
            <value>IA32</value>
            <value>AVR8</value>
         </trait>

         <debug>
            <path>"/home/breakpoint_philosopher.txt"<path>
         </debug>
   </configuration>
<test>
```

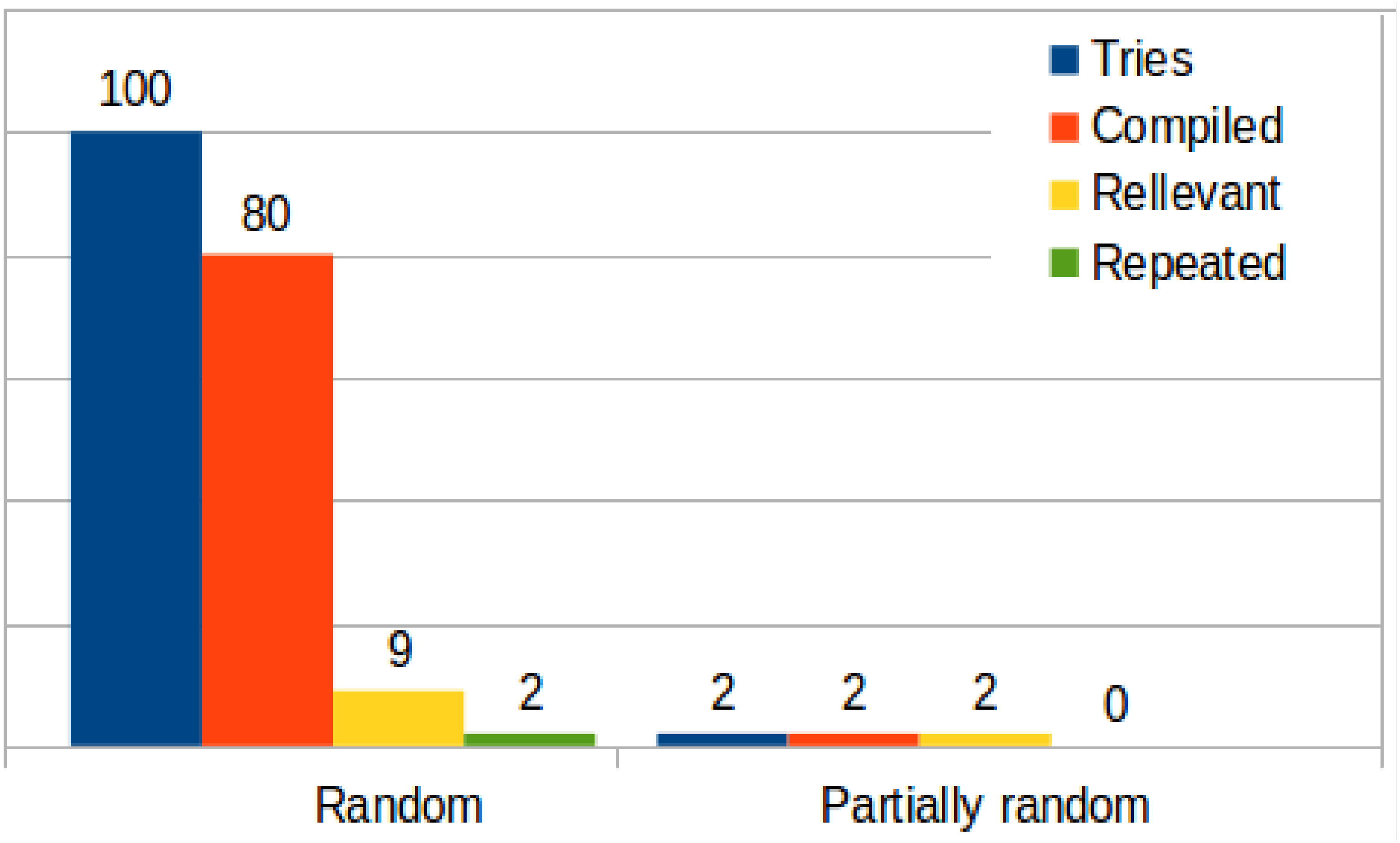
AEP - Automatic exchange of embedded system software parameters

- XML configuration file:
  - Application
    - Name
  - Exchange of parameters
    - Traits
    - Values
    - Interval
    - Number of tries
  - Debugging
    - File with debugging commands
    - Compare traces

Results

Philosopher's Dinner

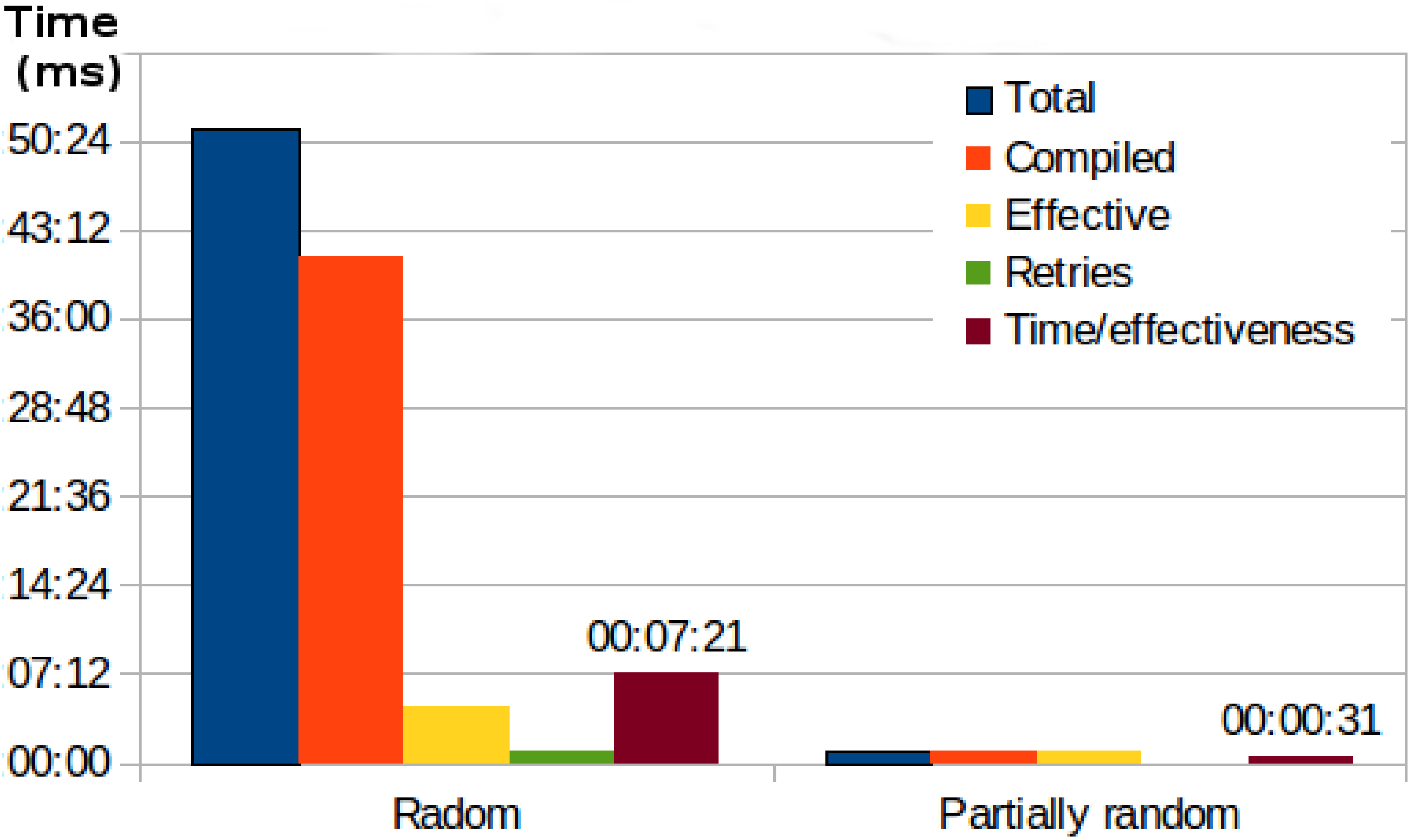
Classification generated configurations



Results

Philosopher's Dinner

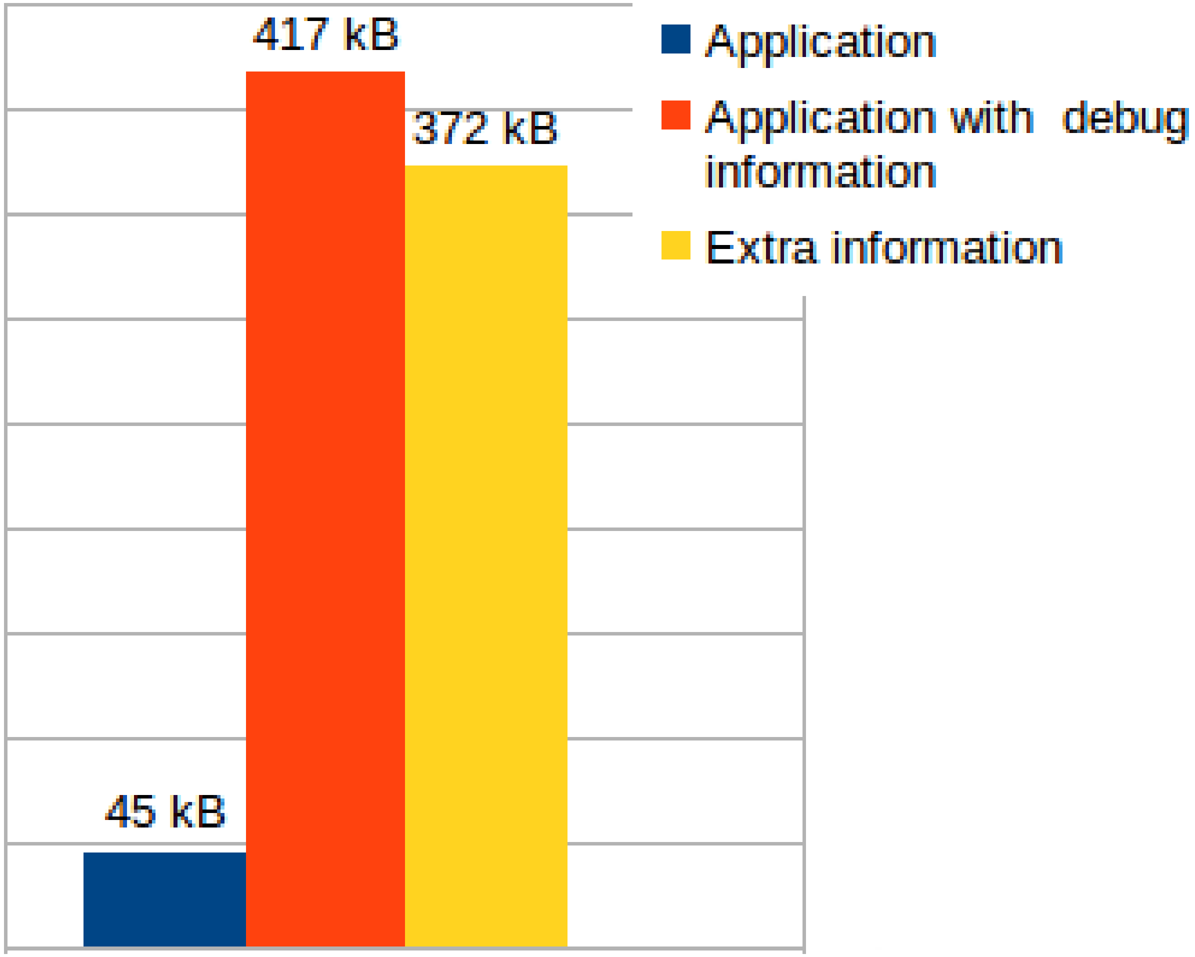
Classification of execution time



Results

Philosopher's Dinner

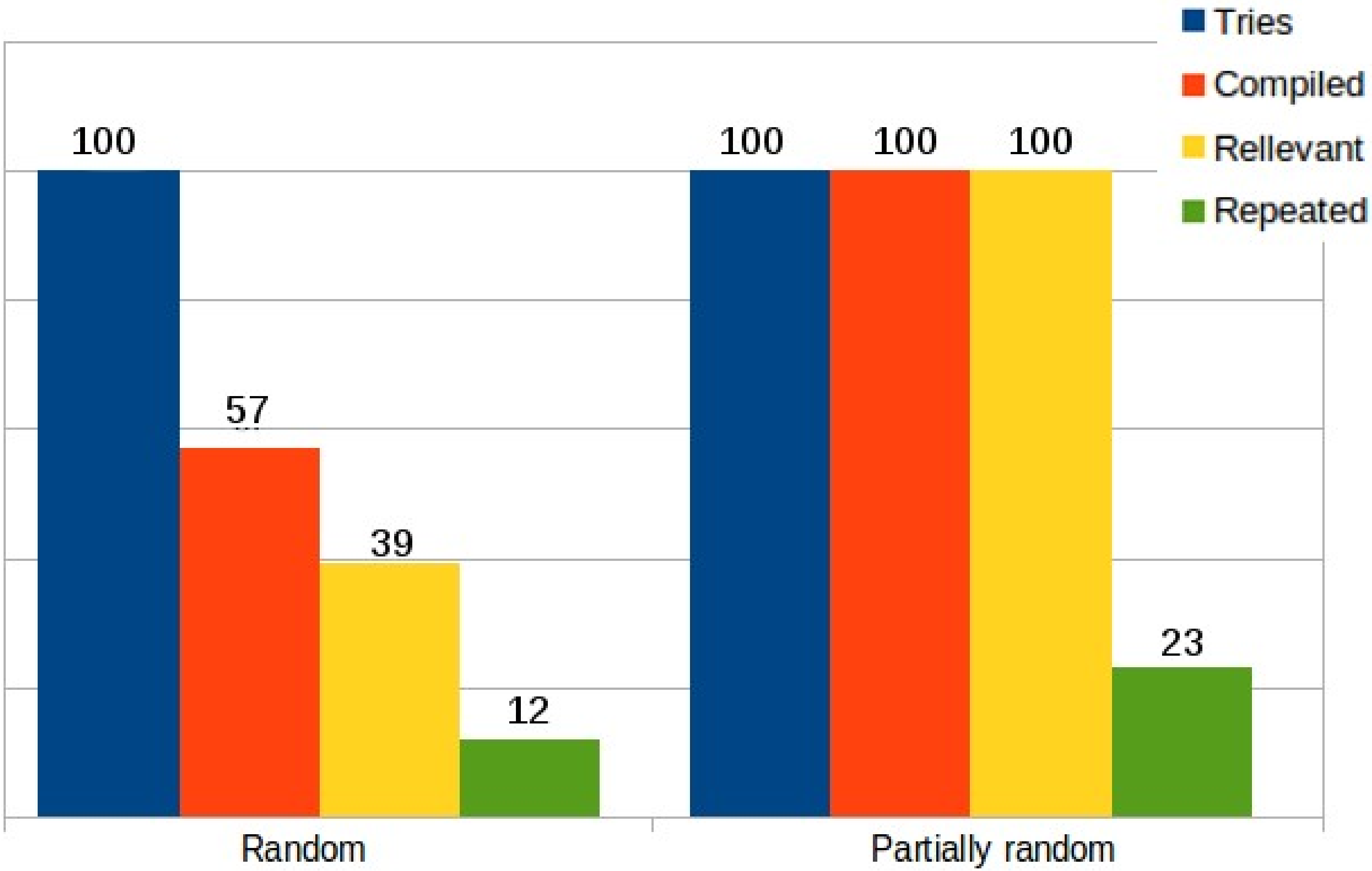
Size of information



Results

DMEC

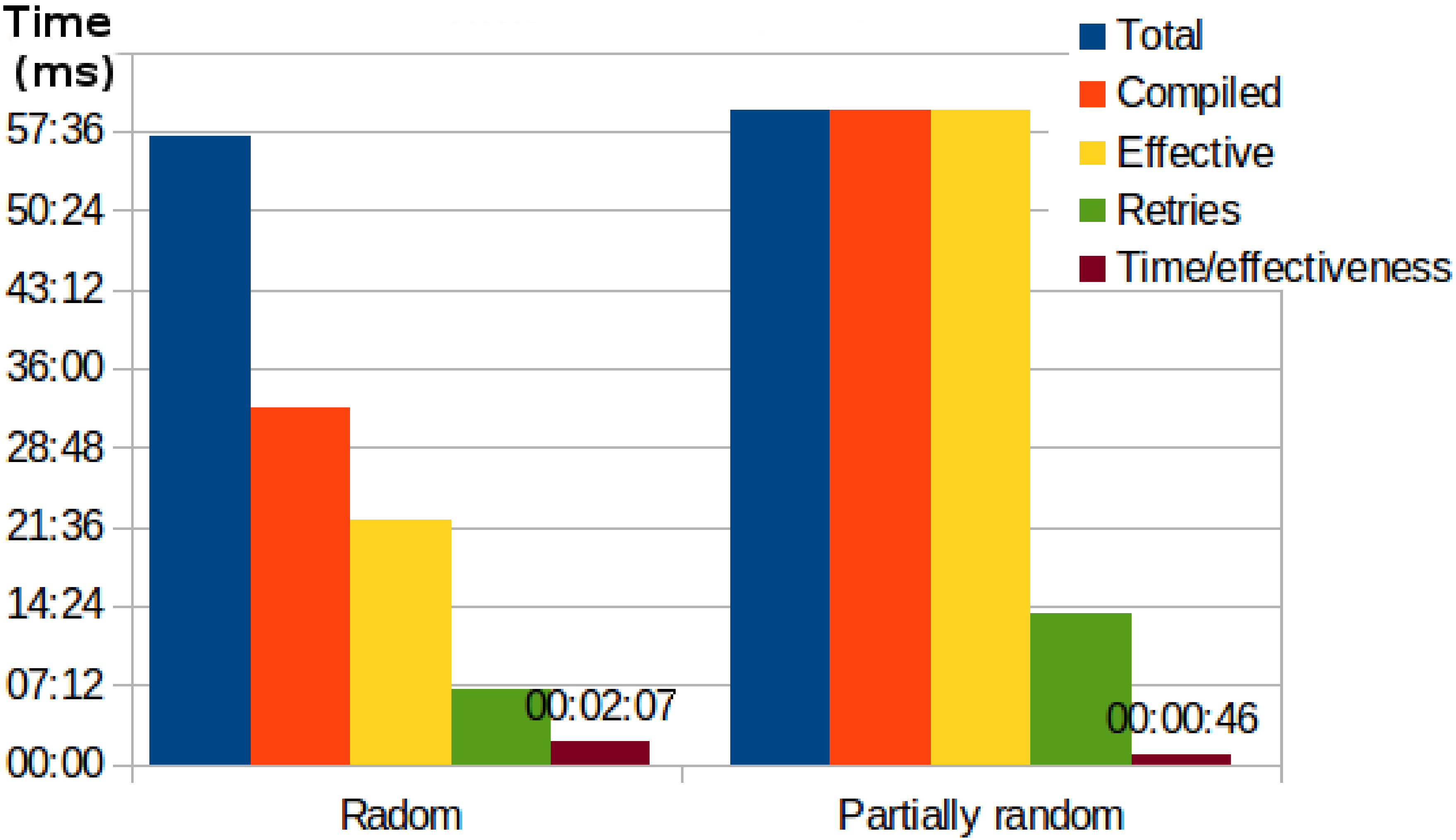
Classification generated configurations



Results

DMEC

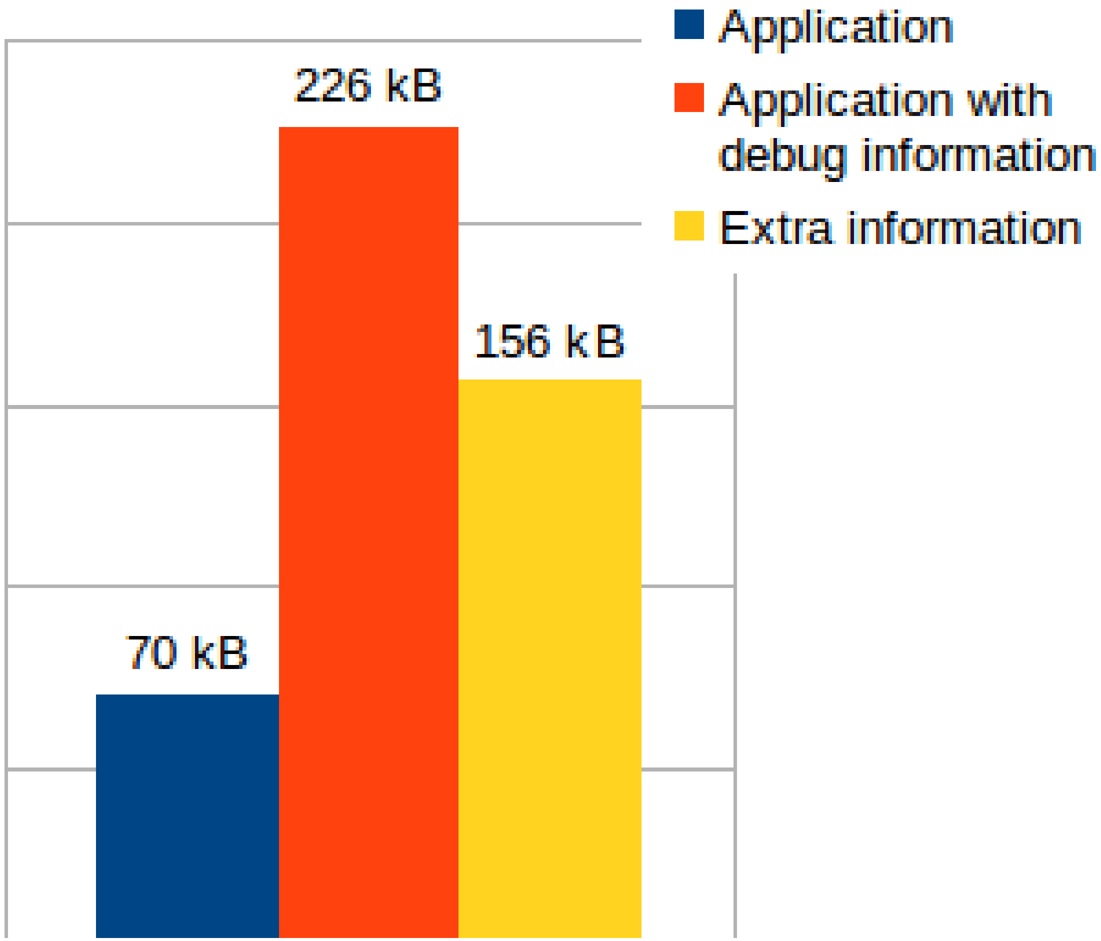
Classification of execution time



Results

DMEC

Size of information



Conclusion and Future work

- This work show that it is possible to create an stable development environment for embedded systems

- The activity of verify this become simpler, since the testing and debugging can share the same environment
  - If necessary a manual debugging, the AEP report can provide useful information

- As future work, there are some improvements, such as:
  - Extract input data directly from application
  - Support new types of testing
  - Deepen the level of information in the report, also returning an analysis of data collected from the application