Learning Compilation Results Of Linux Kernel Configurations

Encadrants

Mathieu Acher (Contact)
Mail : mathieu.acher@irisa.fr, http://www.mathieuacher.com/

Jean-Marc Jézéquel (Contact)
Mail : jean-marc.jezequel@irisa.fr

Structure d'accueil

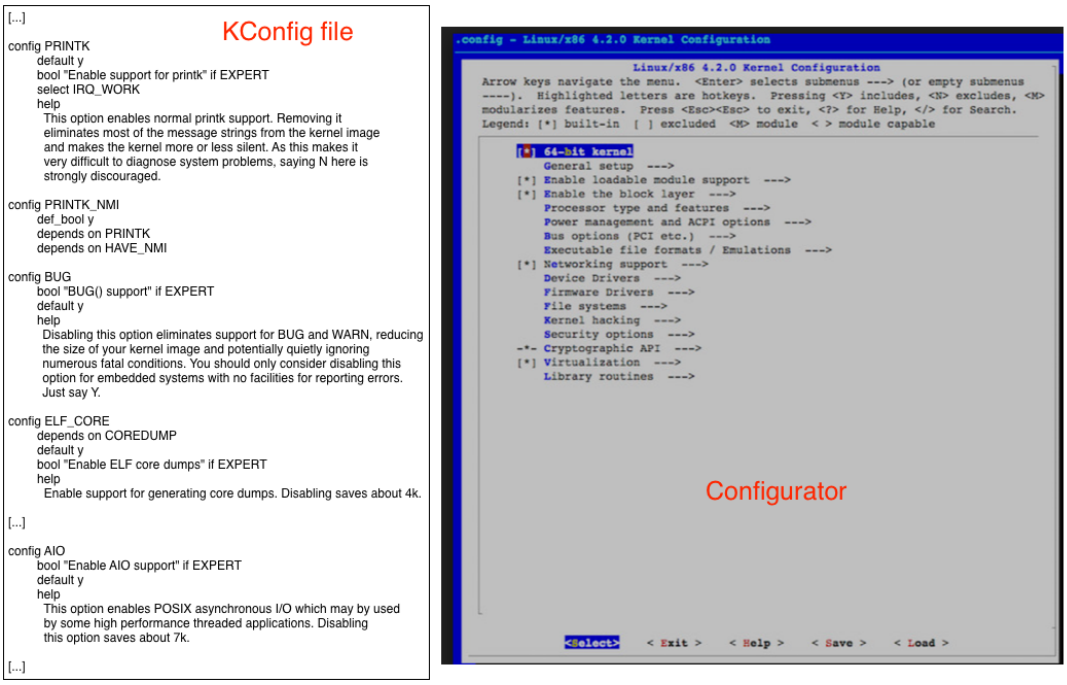
Ville : Rennes
Désignation de l'établissement : Laboratoire
Nom de l'établissement : Inria / IRISA
Équipe : DiverSE

Mots-clés :

- Artifical intelligence
- Machine Learning
- Software engineering
- Operating system

Description :

With more than 15000 configuration options, Linux is certainly the most complex software configurable system ever developed. The Linux kernel offers an almost infinite number of different kernels for fitting the needs of users. Among the resulting many billions possible configurations, some of them may not compile, produce large kernels (around 2 gigabytes!) or small kernels (less than 1 megabyte!), etc.



```
[...]

config PRINTK
        default y
        bool "Enable support for printk" if EXPERT
        select IRQ_WORK
        help
          This option enables normal printk support. Removing it
          eliminates most of the message strings from the kernel image
          and makes the kernel more or less silent. As this makes it
          very difficult to diagnose system problems, saying N here is
          strongly discouraged.

config PRINTK_NMI
        def_bool y
        depends on PRINTK
        depends on HAVE_NMI

config BUG
        bool "BUG() support" if EXPERT
        default y
        help
          Disabling this option eliminates support for BUG and WARN, reducing
          the size of your kernel image and potentially quietly ignoring
          numerous fatal conditions. You should only consider disabling this
          option for embedded systems with no facilities for reporting errors.
          Just say Y.

config ELF_CORE
        depends on COREDUMP
        default y
        bool "Enable ELF core dumps" if EXPERT
        help
          Enable support for generating core dumps. Disabling saves about 4k.

[...]

config AIO
        bool "Enable AIO support" if EXPERT
        default y
        help
          This option enables POSIX asynchronous I/O which may by used
          by some high performance threaded applications. Disabling
          this option saves about 7k.

[...]
```

```
.config - Linux/x86 4.2.0 Kernel Configuration
                       Linux/x86 4.2.0 Kernel Configuration
  Arrow keys navigate the menu.  <Enter> selects submenus ---> (or empty submenus
  ----).  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M>
  modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.
  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

        [*] 64-bit kernel
            General setup  --->
        [*] Enable loadable module support  --->
        [*] Enable the block layer  --->
            Processor type and features  --->
            Power management and ACPI options  --->
            Bus options (PCI etc.)  --->
            Executable file formats / Emulations  --->
        [*] Networking support  --->
            Device Drivers  --->
            Firmware Drivers  --->
            File systems  --->
            Kernel hacking  --->
            Security options  --->
        -*- Cryptographic API  --->
        [*] Virtualization  --->
            Library routines  --->
```

Figure 1 In the left-hand side, an KConfig file excerpt of the Linux kernel. At the right hand-side, the configurator of Linux. Challenge: how to select options that will lead to a small kernel (eg less than 1 megabyte) ?

The goal of this project is to predict, in advance, what are the compilation results of a given Linux kernel configuration. We have already gathered the largest dataset with compilation results of 120000 configurations.

The intern will explore machine learning-based techniques (decision trees, random forests, boosting trees, neural networks, etc.) to effectively learn compilation results. We are targetting different use cases :

- Understanding important options (eg that have a strong impact on the size of the kernel)
- Predicting with a high accuracy some outcomes of the compilation process (eg does this configuration compile ?)
- Narrowing the space of configurations to reach certain objectives (eg kernel less than 1Mo)

Large experiments as well as the development of innovative statistical machine learning techniques are needed to answer these questions. Another possible, related direction is to perform contributions for computing at very large scale Linux kernel configurations.

Skills: We plan to use either R or scikit-learn for performing machine learning experiments. It will be an opportunity to use and apply popular tools in machine learning.

This work is part of an ANR research project called VaryVary. There are 3 open positions (1 Engineer, 1 PhD student, 1 Postdoc)
https://varyvary.github.io/
It is a good starting point for discovering the project and applying to these positions.

Bibliographie :

1. F. Hutter, L. Xu, H. H. Hoos, and K. Leyton-Brown, "Algorithm runtime prediction: Methods and evaluation," Artificial Intelligence, vol. 206, pp. 79 – 111, 2014.

2. Mathieu Acher, Paul Temple, Jean-Marc Jézéquel, José Angel Galindo Duarte, Jabier Martinez, and Tewfik Ziadi. VaryLaTeX: Learning Paper Variants That Meet Constraints (2018). In 12th International Workshop on Variability Modelling of Software-intensive Systems VaryLaTeX

3. Axel Halin, Alexandre Nuttinck, Mathieu Acher, Xavier Devroey, Gilles Perrouin, Benoit Baudry. Test them all, is it worth it? Assessing configuration sampling on the JHipster Web development stack (2018) arxiv

4. Paul Temple, Mathieu Acher, Jean-Marc A Jézéquel, and Olivier Barais. Learning Contextual Variability Models. IEEE Software open access

5. Paul Temple, José Angel Galindo Duarte, Mathieu Acher, and Jean-Marc Jézéquel. Using Machine Learning to Infer Constraints for Product Lines. In Software Product Line Conference (SPLC'16), Beijing, China, September 2016 open access

6. Paul Temple, Mathieu Acher, Jean-Marc A Jézéquel, Léo A Noel-Baron, and José A Galindo. Learning-Based Performance Specialization of Configurable Systems. Research report, IRISA, Inria Rennes ; University of Rennes 1, Feb 2017. open acess