Machine Learning and Configurable Systems (Tutorial at SPLC'20: Part 4)

Juliana Alves Pereira, Hugo Martin,
Paul Temple, Mathieu Acher

https://github.com/VaryVary/

Learning with Linux Kernels

Enormous configurations space eg Linux has 15K+ options, tri-state values {y, n, m}. A build takes 15 minutes on average on a recent machine

KConfig file

```
[...]

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.

[...]
```

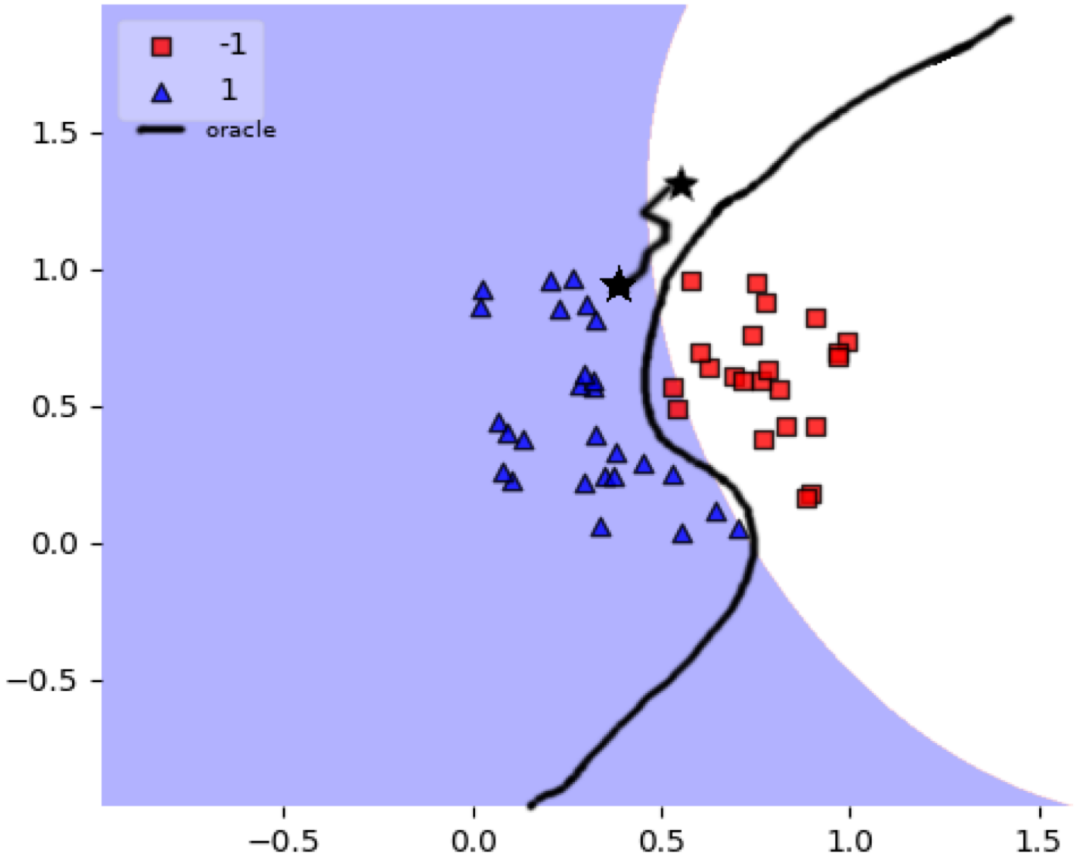
Configurator

```
.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  --->

    <Select>    < Exit >    < Help >    < Save >    < Load >
```

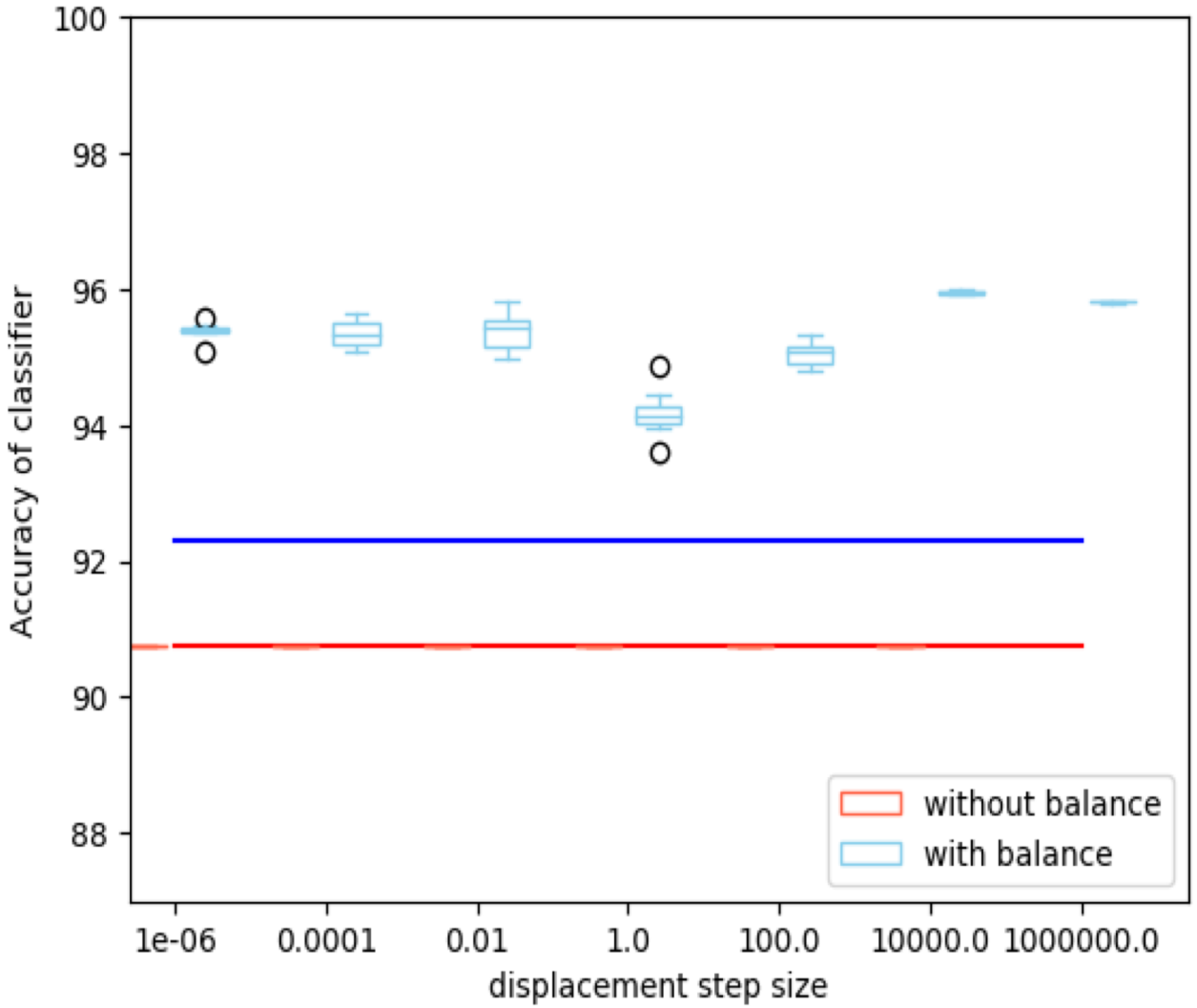
Adversarial Configurations



- Challenge ML model and try to assess weakness
- Adversarial retraining
- Generate new interesting configurations

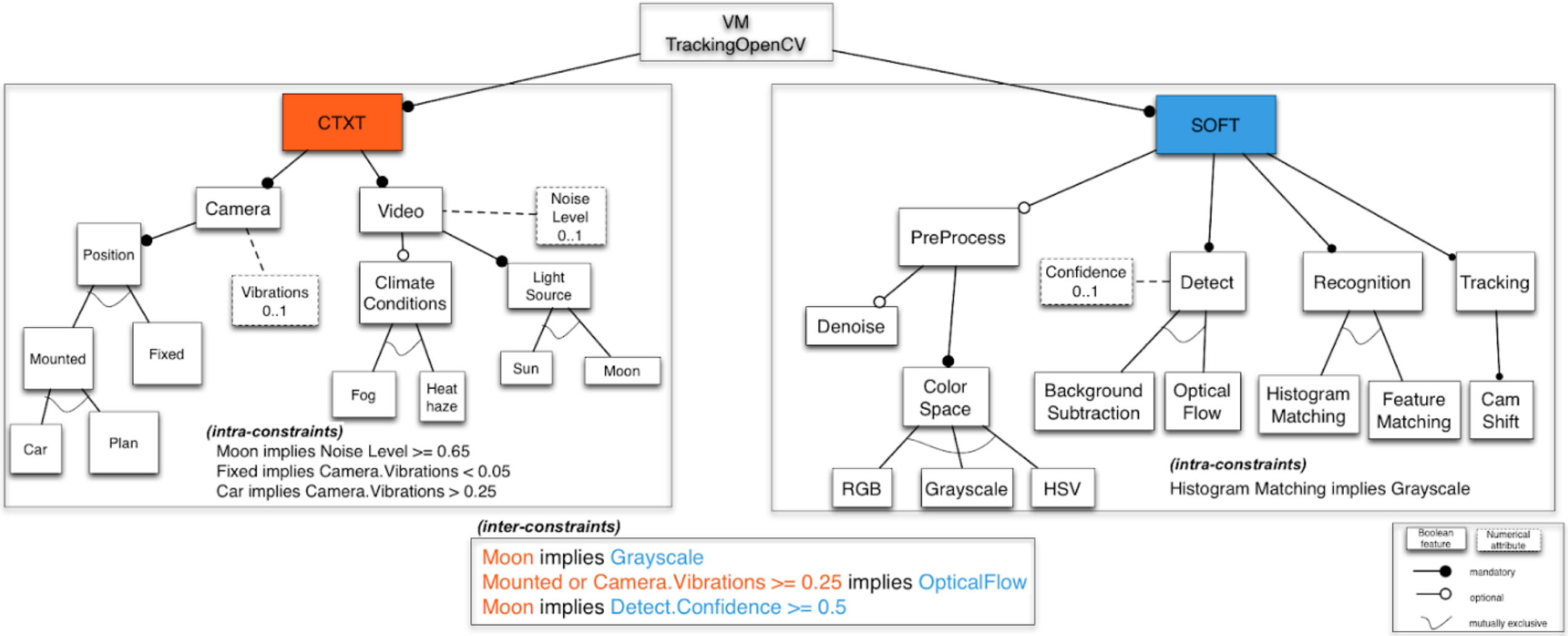
SPLC 2019, Towards Quality Assurance of Software Product Lines with Adversarial Configurations, EMSE follow-up, under review

Adversarial Configurations

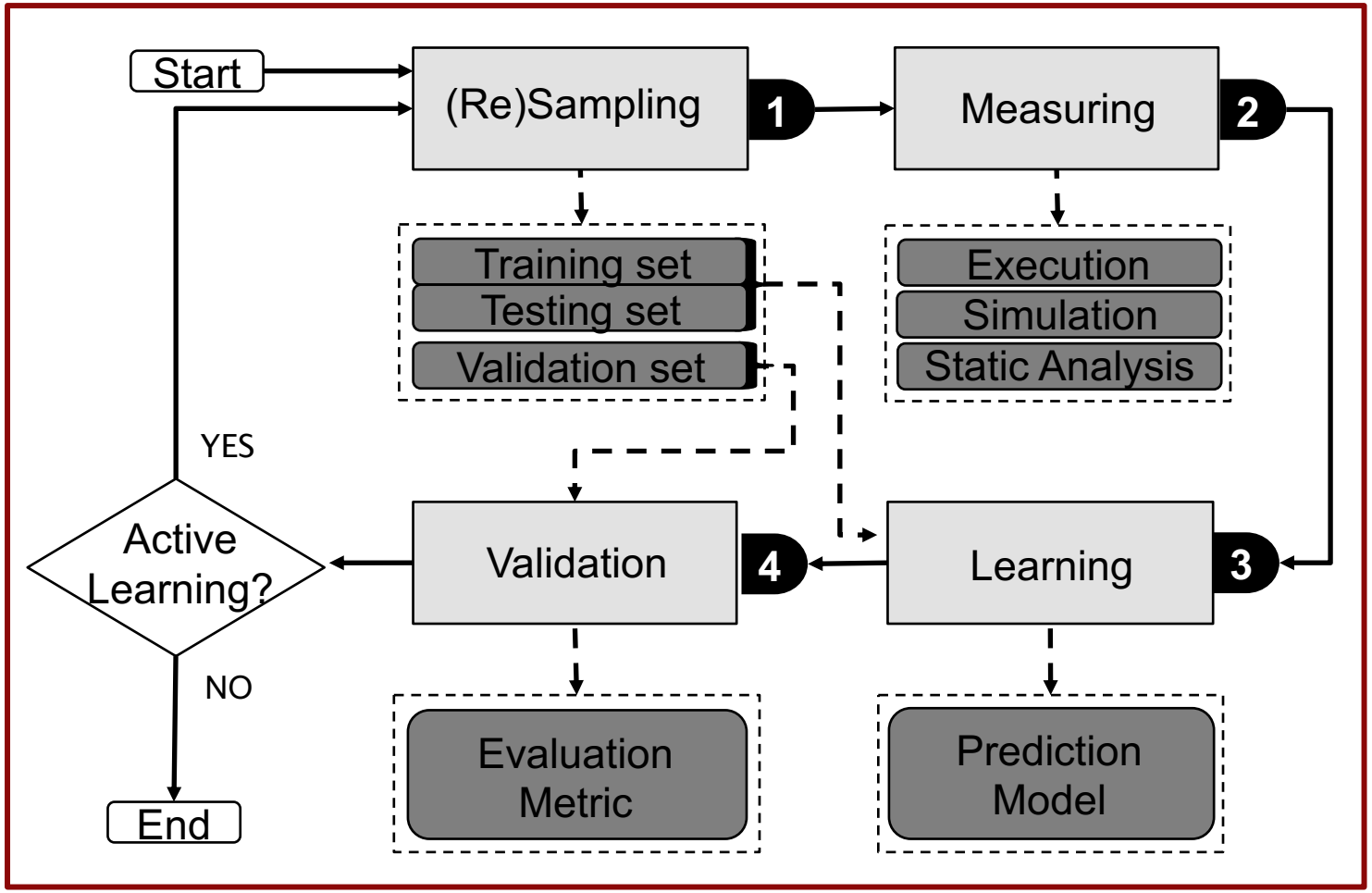


- AdvML generate valid and misclassified configurations
- Adversarial retraining (25 adversarial configurations)
- Promising results

Learning Context



Open Directions



- **Research Questions:**

RQ1. What are the **applications** addressed for learning techniques to explore configuration spaces?

RQ2. Which **sampling** methods are used by these techniques?

RQ3. How do the proposed techniques deal with **measurements** of non-functional properties?

RQ4. Which **learning** techniques have been proposed in the literature?

RQ5. How are these techniques **validated**?

RQ6. What are the **limitations** faced by the current techniques and **open challenges** that need attention in the future?

Open Directions

- Sampling
  - Explore the use of open-box approaches
- Measuring
  - No consensus about now many times we need to repeat measurements
- Learning
  - A few approaches use parameter tuning
- Validation
  - No qualitative studies

Aren't we there yet?

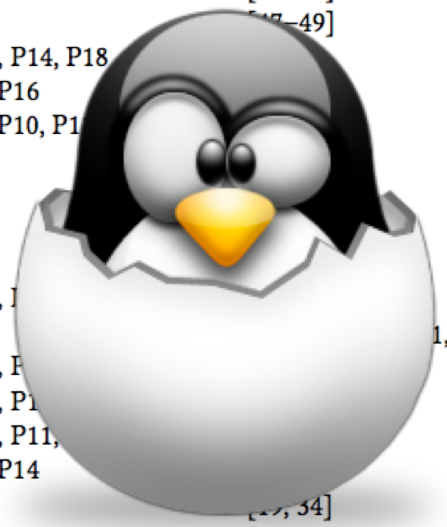
To what extent learning approaches are effective for real-world software systems?

Conclusion

- End-users fear variability and stick to default configurations that may be sub-optimal.
- There are not concrete evidence about the use of learning techniques in practice by users of the evaluated systems.

References	Name	Domain	NFP
[12, 45, 51]	*HIPA^{cc}*	Video processing	P14
[12, 22, 45]	AJStats	Code analyzer	P14
[10, 12, 22, 33, 36, 41, 45, 46, 51]	Apache	Web server	P13, P14, P18
[33]	Apache Storm	Stream processing	P9, P16
[10, 12, 22, 22, 33, 36, 41, 45–49, 51]	Berkeley DB	Database system	P8, P10, P1
[12, 22, 45]	Clasp	ASP solver	P14
[19]	CoBot System	Robotic system	P4
[48]	Curl	Data transfer	P19
[22, 33, 45, 51]	DUNE	Stencil code	P14
[22, 33, 45, 51]	HSMGP	Stencil code	P14, I
[45, 51]	JavaGC	Runtime environment	P14
[47–49]	LinkedList	Data structures	P10, I
[48? –50]	Linux kernel	Operating system	P10, P1
[10, 12, 22, 34, 36, 41, 45, 46, 48, 51]	LLVM	Compiler	P10, P11,
[12, 22, 33, 45]	LRZIP	Compression library	P2, P14
[51, 52]	MOTIV	Video encoder	P17
[34]	noc-CM-log	FPGA	P3, P15
[19]	NoSQL	Database system	P9
[51]	OpenCV	Video tracking	P11

References	Name	Domain	NFP
[47–49]	PKJab	Instant messenger	P10, P11
[47–49]	Prevayler	Database	P10, P11
	RAR	Compression	P19
	RollingSort	Stream processing	P9, P16
	SaaS system	Cloud computing	P14
	SaC	Code analyzer	P8, P14
	SensorNetwork	Simulation	P10, P11
	Sol	Stream processing	P9
	sort-256	FPGA	P1, P16
	SPEAR	SAT solver	P14
1, 46–49, 51]	SQLite	Database system	P10, P11, P14, P19
	StockOnline	Database system	P14
	Trimesh	Mesh solver	P7, P14
	Violet	UML editor	P10, P11
	Wget	Data transfer	P10, P19
34]	WordCount	Stream processing	P9, P16
[32]	WordPress	Content management	P3
[10, 12, 18, 22, 33, 34, 36, 41, 45, 46, 48, 51]	x264	Video encoder	P3, P5, P14, P19
[47–49]	ZipMe	Compression	P10, P11, P19



Thanks to our Sponsors!

Bronze Level

pure·systems

Supporters

acm Association for Computing Machinery

SIGSOFT

ÉTS
Le génie pour l'industrie

MetaCase

Montréal 2020

24th ACM International Systems and Software Product Line Conference