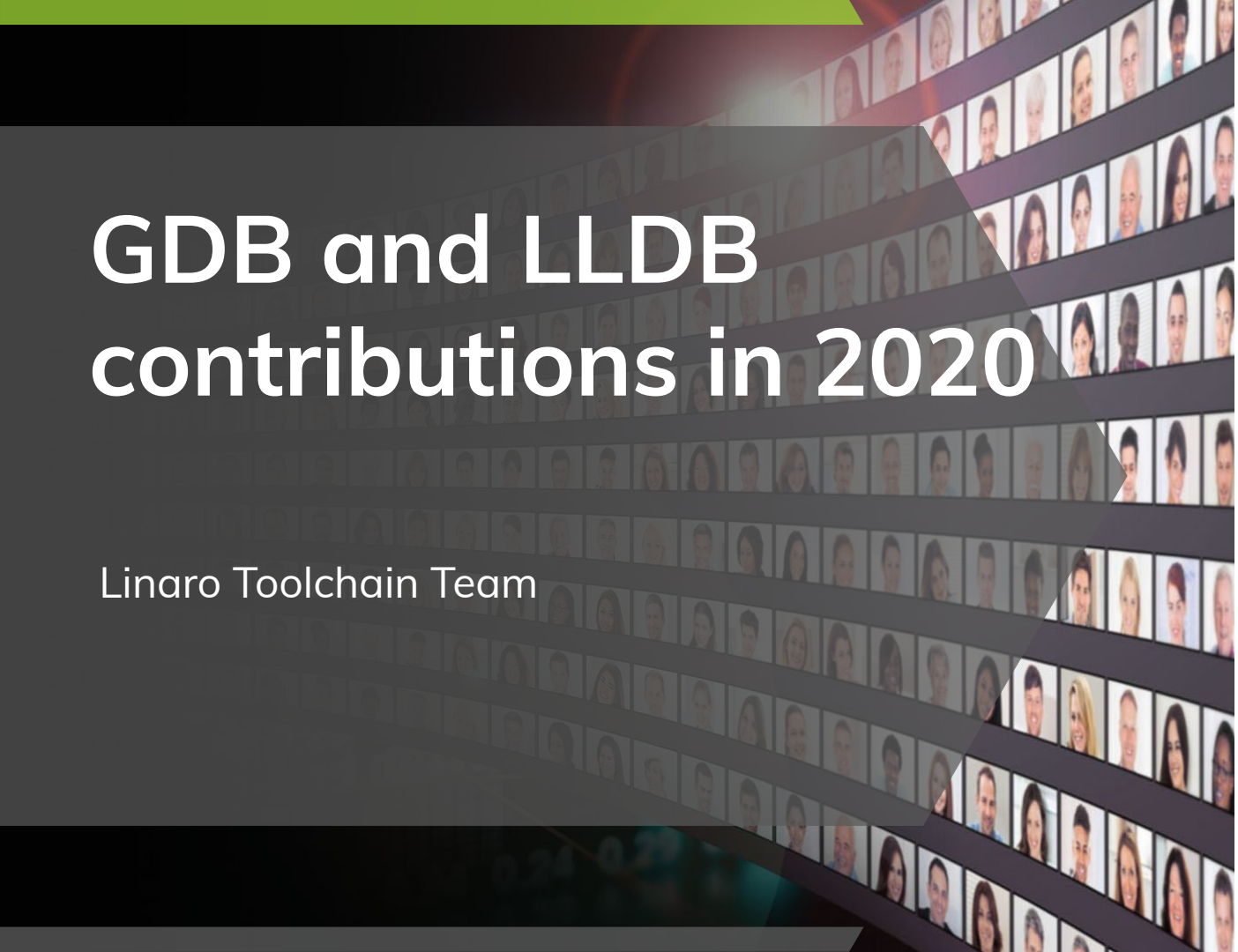
GDB and LLDB contributions in 2020

Linaro Toolchain Team



Linaro Connect
VIRTUAL • SPRING 2021

Introduction: Our Team

- David Spickett
- Diana Picus
- Luis Machado
- Omair Javaid

Introduction: LLDB Projects

- Upstream maintenance
- Scalable vector extension (SVE)
- Pointer authentication code (PAuth)
- Top byte ignore (TBI)
- Memory tagging extension (MTE)
- Morello Project

LLDB Maintenance

- Arm/AArch64 Code Ownership
- Arm and AArch64 Linux Buildbots
- Refactoring and Bug fixes

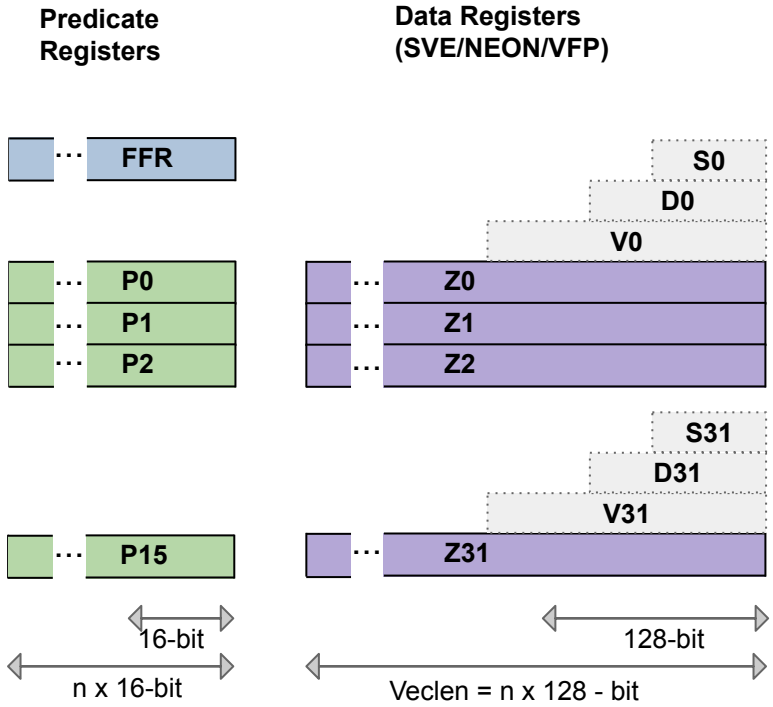
What is Scalable Vector Extension?

- SVE is an optional extension to the Armv8-A architecture (Armv8.2-A onwards).
- SVE is an add-on on top of AArch64 Advanced SIMD and floating-point unit.
- Finds its application in processing of large data sets in HPC workloads.
- SVE hardware aimed to complement auto vectorization capability of modern compilers.

Picture Source: wikipedia (Property of Fujitsu)

SVE Register Access

- Scalable data registers Z0-Z31
  - IEEE 754 compliant floating point
    - Double, single and half precision elements
  - Packed 64, 32, 16 & 8-bit integer
- Predicate registers (P0-P15) and first fault register (FFR)
- LLDB support merged upstream
- LLVM 12 release will have full support



Pointer Authentication Code

- An optional extension for Armv8.3 mandatory for Armv8.6 onwards.
- Mitigates ROP attacks
- Backward compatible utilizing NOP space
- Requires debugger's PAC awareness for unwinding stack.
- LLDB support under development

```
SUB sp, sp, #0x40
STP x29, x30, [sp,#0x30]
ADD x29, sp, #0x30
...

...
LDP x29,x30,[sp,#0x30]
ADD sp,sp,#0x40
RET
```

```
PACIASP
SUB sp, sp, #0x40
STP x29, x30, [sp,#0x30]
ADD x29, sp, #0x30
...

...
LDP x29,x30,[sp,#0x30]
ADD sp,sp,#0x40
AUTIASP
RET
```

Top Byte Ignore (TBI)

- Feature of Armv8 AArch64
- Known as “address tagging”
- Allows hardware to ignore the top byte of pointers. (max address size is currently 52 bits)
- Software can use this byte without manually removing it before every access.
- Debuggers must account for this when handling pointers. E.g. watchpoint locations

Equivalent pointers with TBI		
bits 63-56	**bits 55-52**	**bits 51-0**
0xEE	0x0	0x0111122223333
0xFF	0x0	0x0111122223333

Uses:

- Software memory tagging e.g.
  Hardware Address Sanitizer (HWASAN)
- Other object metadata

Supported as of GDB 8.1

LLDB support under development

Memory Tagging (MTE)

- Feature of Armv8.5-a for AArch64
- Uses pairs of 4 bit tags to verify memory accesses.
  - Logical tags (bits 59-56 of pointers)
  - Allocation tags (stored in hardware)

- Each allocation tag applies to a 16 byte “granule” of memory.
- A mismatched pair of tags generates an exception.
- Can detect memory safety issues. Buffer overflow, use after free, etc.

Pointer Layout		
bits 63-60	**bits 59-56**	**bits 55-0**
unused	4 bit tag	address

Buffer Overflow Example
(you have a pointer to Buffer A with a logical tag of 5)

	Buffer A		**Buffer B**
Location	granule 1	granule 2	granule 3
Allocation tag	5	5	6
Access	Ok	Ok	Exception

Memory Tagging Debugging

- MTE is supported in Linux userspace as of 5.10
- Linaro is enabling GDB and LLDB

Features:

- Reading and writing allocation tags
- Tag annotations for memory reads
- Reporting tag mismatches

- Complete GDB support is in review
- LLDB support is under development

```
(gdb) mtag check buf
Logical tag (0x9) does not match the allocation tag (0x0) for
address 0x900fffff7ffa000.
(gdb) x/4gm buf
<Allocation Tag 0x0 for range
[0x900fffff7ffa000,0x900fffff7ffa010)>
0x900fffff7ffa000:      0x0000000000000000      0x0000000000000000
<Allocation Tag 0x1 for range
[0x900fffff7ffa010,0x900fffff7ffa020)>
0x900fffff7ffa010:      0x0000000000000000      0x0000000000000000
(gdb) mtag setatag buf 1 09
Allocation tag(s) updated successfully.
(gdb) mtag check buf
Memory tags for address 0x900fffff7ffa000 match (0x9).
```

Morello Project

- Research program led by ARM
- UKRI funding for Digital Security by Design Programme
- Based on CHERI
  - Capability Hardware Enhanced RISC Instructions
- 129-bit capability types instead of pointers
  - Tag (1 bit)
  - Permissions, object type and bounds (64-bit)
  - Value and flags (64 bits)
- New C register set overlapping X registers
  - Registers are 129 bit in size (128 bits + tag bit)
- Enablement of LLVM/GNU Toolchains

arm
Morello Program

UK Research and Innovation

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

THE UNIVERSITY of EDINBURGH

Morello Project

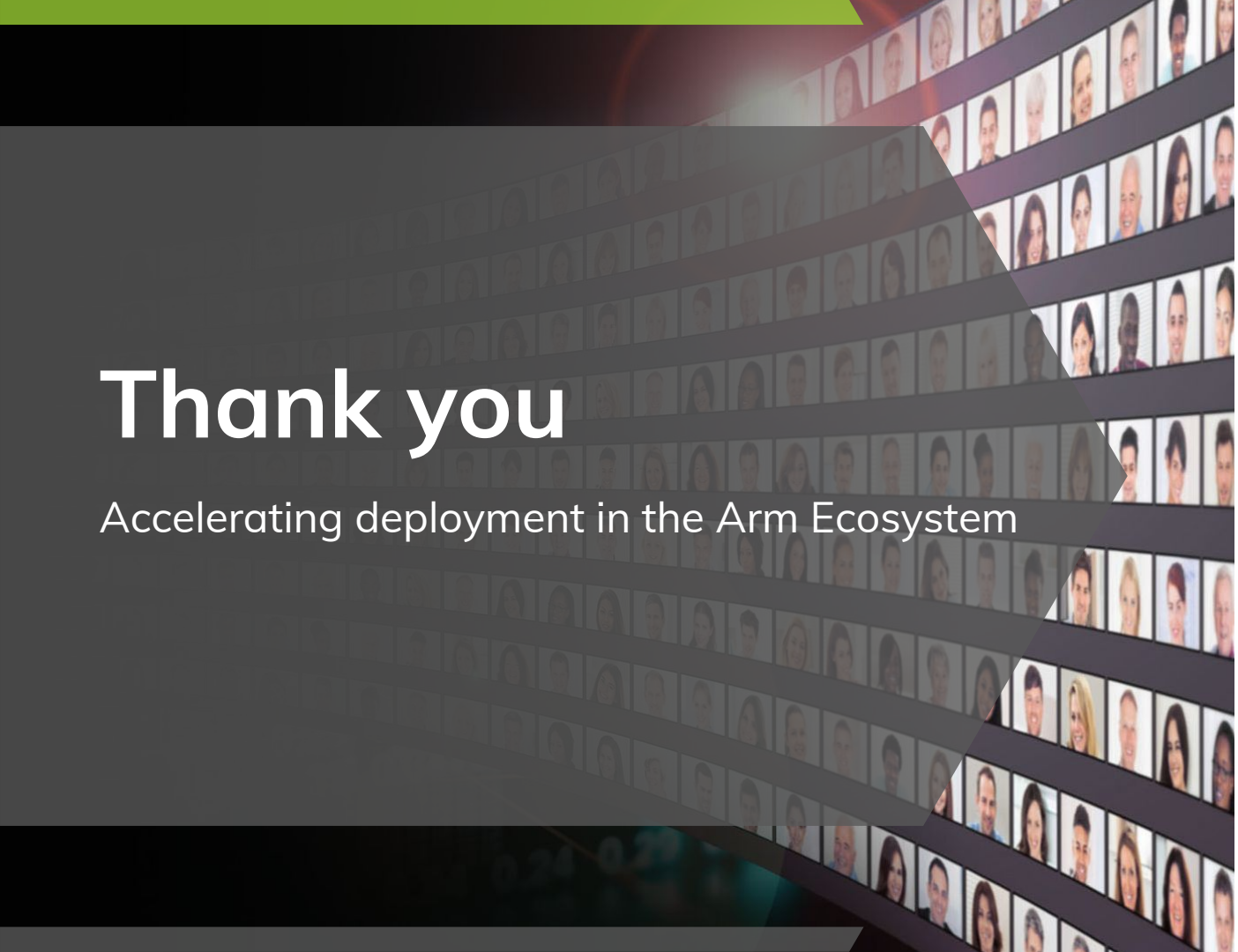
- Development platforms
  - Morello Platform Model
    - Open access Fixed Virtual Platform
  - Morello Hardware Development Platform
    - Expected in Q1 2022
    - Limited availability
- Enabled platforms
  - Android CHERI
  - Bare-Metal
  - AArch64 Linux
  - CHERIBSD
- Android Nano stack
- Linux Kernel (modified Android Common Kernel)

Morello Project

- Supported Morello features for LLDB and GDB
  - AAPCS64 (hybrid) and AAPCS64-CAP (pure capability)
  - C register set
  - DWARF capability support
  - PTRACE extensions
  - Capability reads/writes
  - Remote protocol extensions
  - Unwinding
  - Signal frame
  - Capability/Pointer conversions
- LLDB and GDB in alpha stage

GDB Status

- SVE
- PAC / TBI
- BFloat16
- MTE under review
- A number of bug fixes and improvements



Thank you

Accelerating deployment in the Arm Ecosystem

Linaro Connect
VIRTUAL • SPRING 2021