

Arm64 Linux Kernel Architecture Update

Software enablement of Arm Architecture in Linux World

Akanksha Jain: Technology Management | Mark Rutland : Distinguished Engineer
Arm Central Engineering, Open-Source Software.



Agenda

Architectural Feature Landing Pages

Context and Scope

Arch Enablement Flowers

Kernel Landing

Architectural Feature Development Pipeline

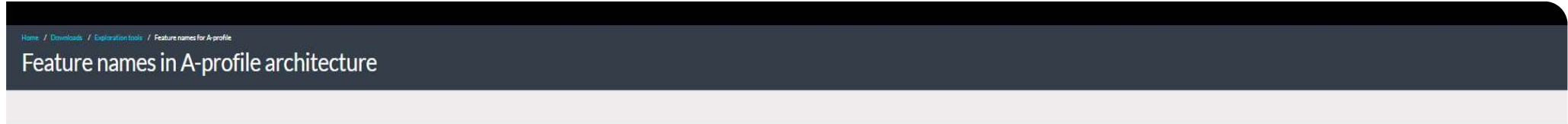
Ongoing activities

Other Highlights

Additional Reading Resources

A-profile features: Feature Names

<https://developer.arm.com/downloads/-/exploration-tools/feature-names-for-a-profile>



The tables on this page list all the features that are in A-profile architecture.

2022 Architecture Extensions

Note: Some 2022 Architecture features are not yet included.

Feature Name	Short description
FEAT_ABLE	Address Breakpoint Linking Extension
FEAT_ADERR	Asynchronous Device error exceptions
FEAT_AIE	Memory Attribute Index Enhancement
FEAT_ANERR	Asynchronous Normal error exception
FEAT_CLRBHB	Support for Clear Branch History instruction
FEAT_CHK	Check Feature Status
FEAT_CSSC	Common Short Sequence Compression instructions
FEAT_CSV2_3	Cache Speculation Variant 2 version 3

A-profile features: arm64 kernel support table

<https://developer.arm.com/Tools%20and%20Software/Linux%20Kernel#Components>

armDeveloperIP Explorer

Developing on ArmArchitecture and ProcessorsTools and Software

Technical SpecificationsResourcesEditionsComponentsSupport and Training

Table of Arm Architecture Features and Kernel Versions

View the following table of Arm architecture features against the merged Linux Kernel version.

See the [Feature names in A-profile architecture table](#) for a full list and comparison of old and new names.

New Feature Name	Old Feature Name	Kernel Version	Notes
FEAT_CRC32		3.14	HWCAP
FEAT_Debugv8p1	ARMv8.1-Debug	5.6	Debug with VHE
FEAT_LSE	ARMv8.1-LSE	4.3	Kernel atomics use and HWCAP
FEAT_RDM	ARMv8.1-RDMA	4.11	HWCAP
FEAT_HPDS	ARMv8.1-HPD	N/A	
		4.6	Virtualization Host Extensions
FEAT_VHE	ARMv8.1-VHE	4.17	KVM support
		4.3	Access flag & Dirty Bit Management (ARM64_HW_AFDDBM)
FEAT_HAFDBS	ARMv8.1-TTHM	4.7	KVM support
FEAT_PAN	ARMv8.1-PAN	4.3	

Architecture Enablement : Context and Scope

Recap | LHR23

2022 Extensions
[PIE, POE]

Confidential Compute
[kernel, kvm-tool, kvmunit-test, EDK2]

2020 extensions
[WFxT, LPA2 etc]

Advanced Arch Support

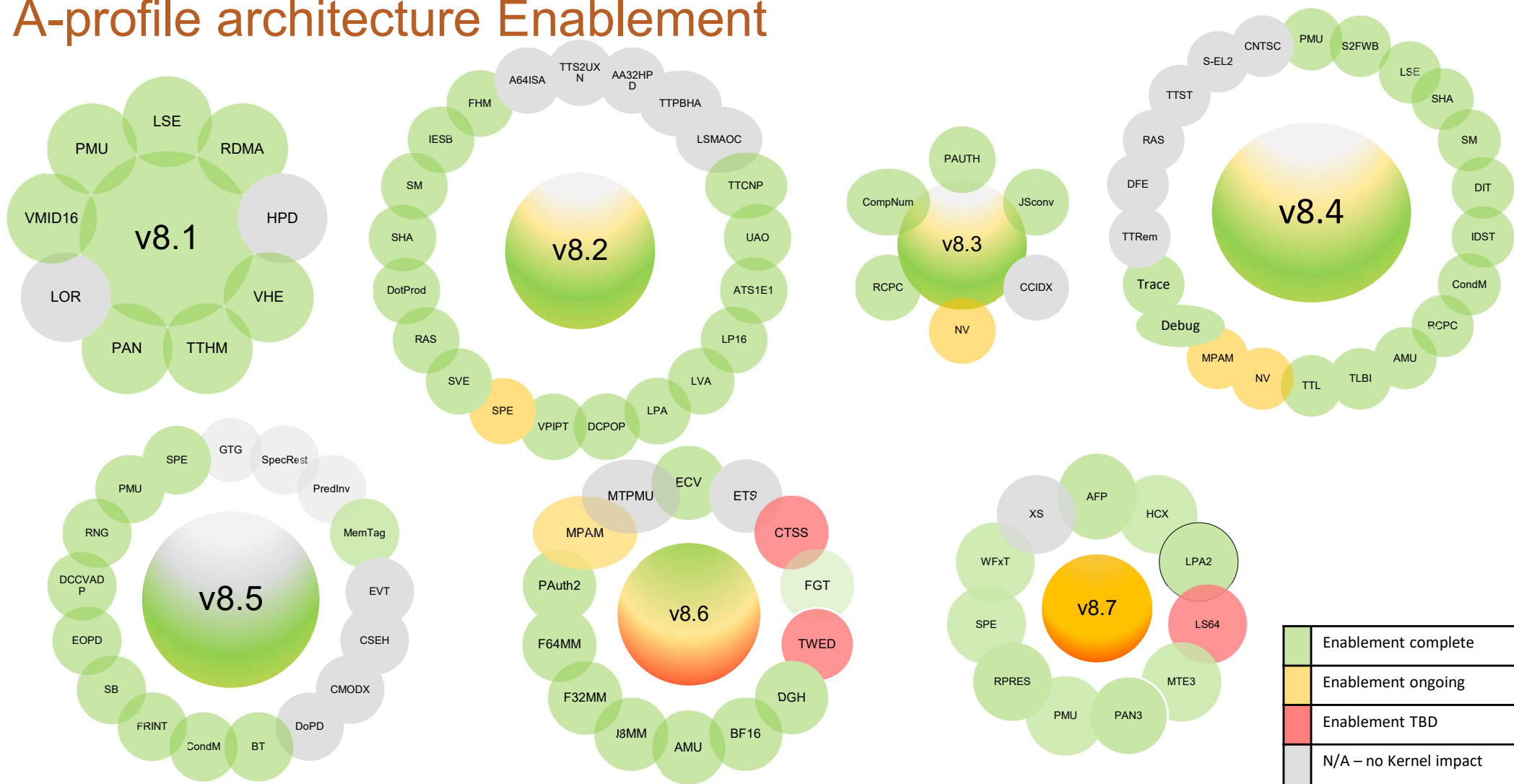
Linux Kernel Morello
[Morello SDK and Linux]

Armv8.8 | 2021 extensions
[FEAT_MOPS | NMI | PMUvTH]

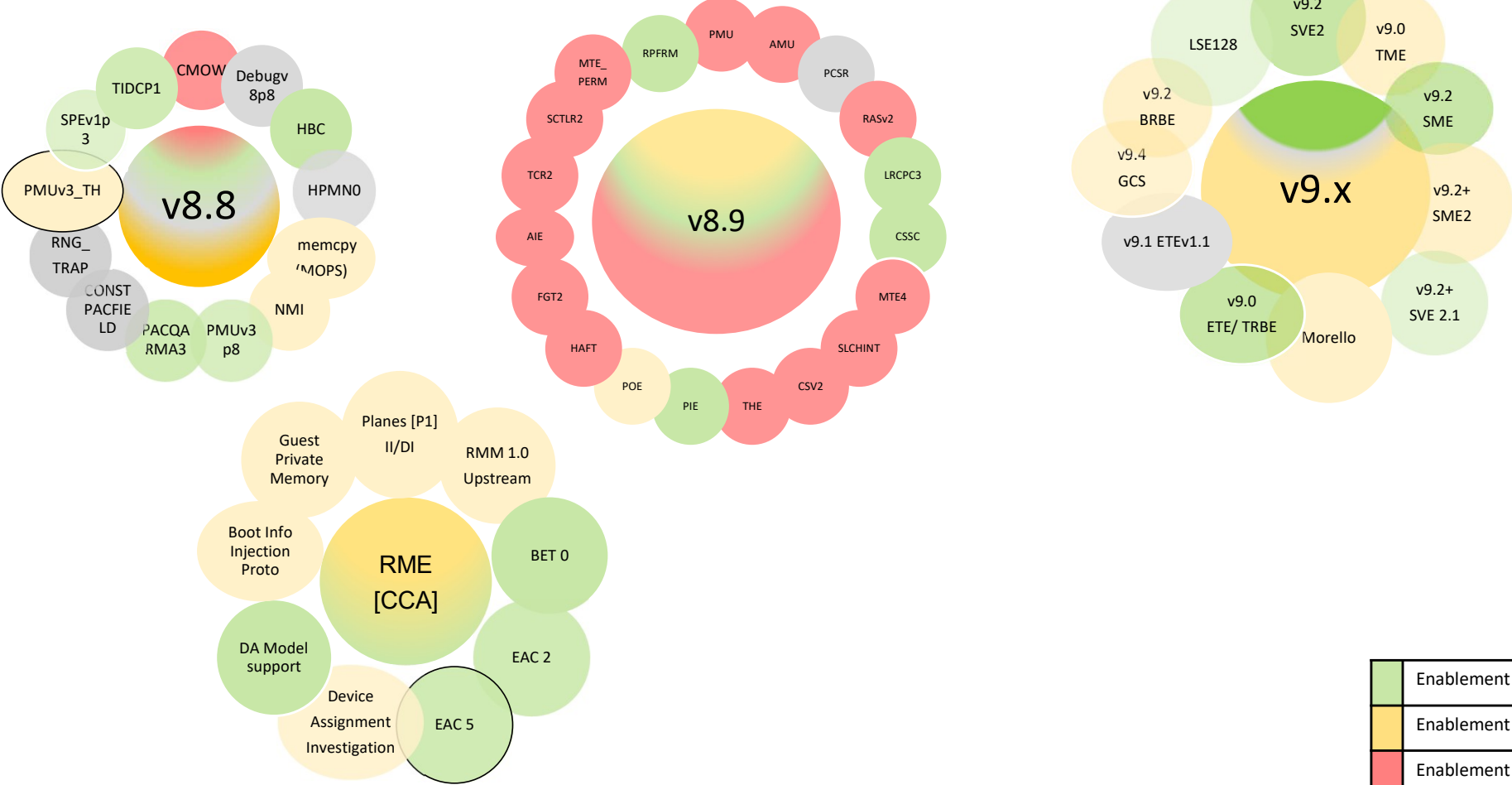
Kernel Landing

Scalable Matrix Extension

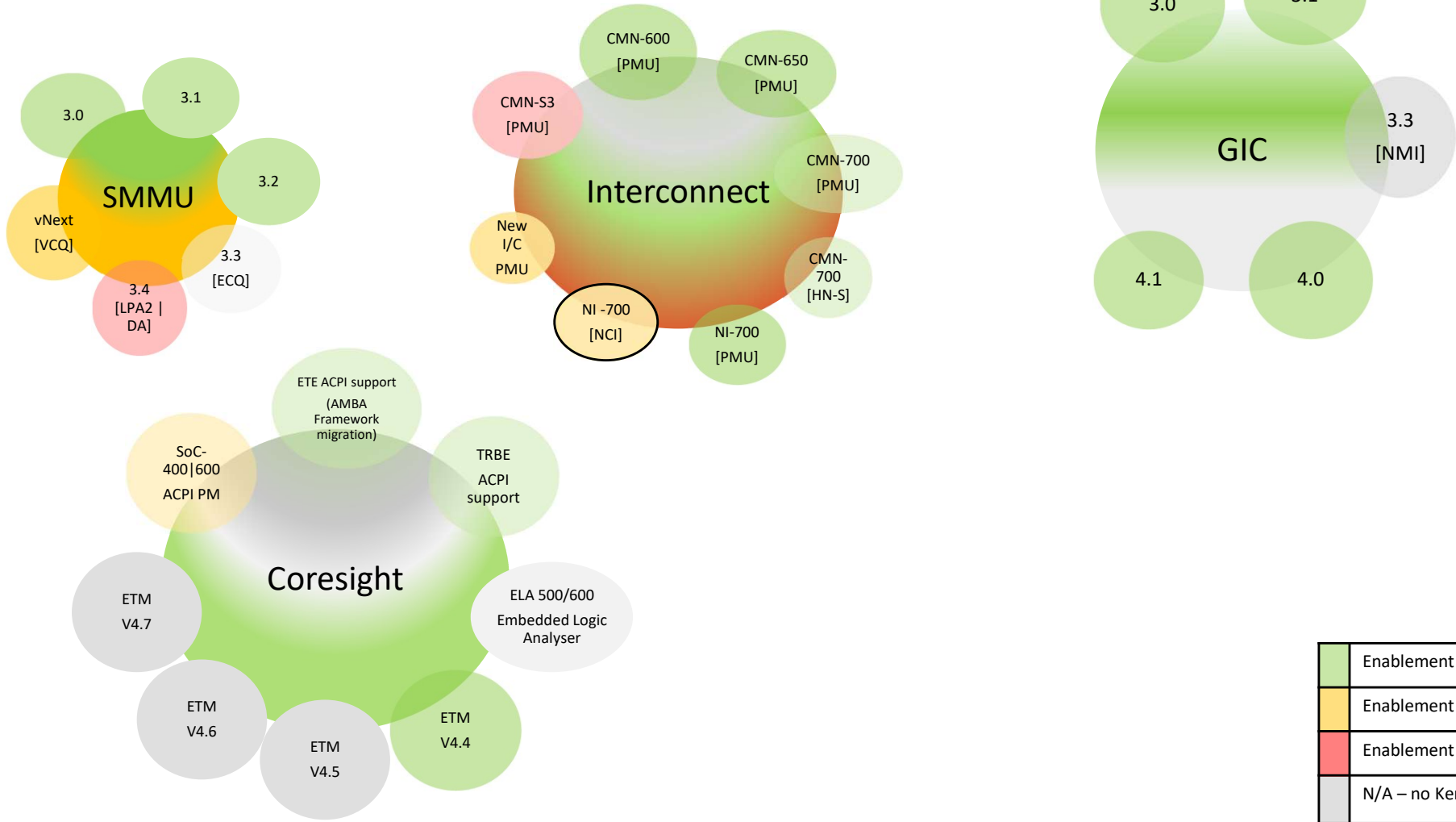
A-profile architecture Enablement



A-profile architecture Enablement



System IP Enablement



	Enablement complete
	Enablement ongoing
	Enablement TBD
	N/A – no Kernel impact

Arm Arch Features | Kernel Landing

Feature Name	Arch Version	Kernel Landing
MPAM [resctrl refactor- Part 1]	Arm v 8.3 8.4	6.9
FEAT_PMU	2021 8.8	6.8
NI-700	System IP	On-list [Aim 6.10]
FEAT_PIE	2022 8.9	6.5
FEAT_MOPS HWCAP	2021 8.8	6.5
FEAT_MOPS KVM	2021 8.8	6.7
FEAT_HBC HWCAP	2021 8.8	6.6
FEAT_LRCPC3 HWCAP	2022 8.9	6.7
FEAT_LSE_128 HWCAP	2022 9.x	6.7
FEAT_POE	2022 8.9	On-list
CMN 700 HN-S Support	System IP	6.6
ETE ACPI Support	CoreSight	6.6
TRBE ACPI Support	CoreSight	6.8

Arm Arch Features | Kernel Landing

Feature Name	Arch Version	Kernel Landing
Pointer Authentication	Arm 8.3 8.6	5.10
ETM 4.4	Arm 8.4	5.12
BTI User space Kernel	Arm 8.5	5.9
MTE User space Support	Arm 8.5	5.10
MTE In Kernel Support	Arm 8.5	5.11
MTE Async Support	Arm 8.5	5.13
MTE KVM Support	Arm 8.5	5.14
MTE core dump support	Arm 8.5	5.18
MTE Strcmp support	Arm 8.5	5.18
MTE3 user space	Arm 8.7	5.18
EPAN PAN3	Arm 8.7	5.13
ETE TRBE	Arm 8.7	5.13
AFP RPRES HWCAP support	Arm 8.7	5.17

Linux Kernel : Ongoing development

1

Permission Overlay (POE | S1 | S2)

- Allows constraining permissions of memory regions. Can be used from ELO
- Implemented using memory protection key syscall
- [v3](#) on-list; likely to get merged in the next few cycles
- Next steps: explore S2 use-case

4

BRBE : Branch Record Buffer Extension

- perf support for BRBE- records branches taken in an execution path
- [v17](#) on-list; awaits review from maintainers
- 1-2 more kernel iterations before a Kernel landing (tentatively 6.11)

2

FEAT_MOPS | memcpy

- Optimal memcpy achieved with single instruction sequence
- User space support -> Kernel 6.5 | KVM support -> 6.7
- In kernel support is WiP; tentatively 6.12

5

Scalable Matrix Extension | SME | SME2

- SME=> outer-product instructions to accelerate matrix operations
- SME 2 => add more instructions for multi-vector operations etc
- Everything minus KVM support merged in **Kernel 6.3**
- KVM support gated on some general upstream KVM changes; rebase the series soon

3

GICv3.3 + Arm v8.8 NMI

- Introduction to the Feature NMI in the [Arm arch](#)
- Gated on refactoring of previous pseudo-NMI patches
- Should be picked up in the upcoming Kernel cycles

6

Guarded Control Stack Enablement

- Provides mitigations against some forms of ROP attack. Return address is pushed onto the GCS and check with LR on return
- ELO patches [v8](#) on-list; should rebase the series post a few bug fixes

Arm 8.3 | 8.4 Nested Virtualisation

- ❑ Nested Virtualisation Extensions :
 - ❑ Development by the KVM arm64 community
 - ❑ Testing primarily for FEAT_NV2 (not FEAT_NV) as no hardware solely supporting FEAT_NV
 - ❑ (FEAT_NV = ARMv8.3 NV, FEAT_NV2 = ARMv8.4 NV2)
 - ❑ [v8 series](#) -> 19/69 => Kernel 6.3 ; [v9 series](#) -> 1st 7 patches => Kernel **6.5**
 - ❑ [v10 series](#) (50+9) + [30 ([NV Trap forwarding](#)) => Kernel **6.6**]
 - ❑ virtual EL2 handling merged upstream
 - ❑ [v11 series](#) (43 patches) on-list
 - ❑ 10/43 NV2 improvements merged in **Kernel 6.8**
 - ❑ 16/33 => shaving off the [shadow SMMU code](#) => patches on-list await review
 - ❑ Remaining work around GIC, timer and any follow up perf. Optimization => upcoming Kernel cycles
 - ❑ [Add PAC](#) support to the existing NV code (more details on linux-next) => aiming **6.10**

Arm 8.4 | 8.6 MPAM Enablement

- ❑ Working on enabling multi-arch support for the resctrl interface
 - https://www.kernel.org/doc/Documentation/x86/intel_rdt_ui.txt
 - <https://github.com/intel/intel-cmt-cat/wiki/resctrl>
- ❑ Public snapshot rebased against every kernel release; latest version available [here](#)
- ❑ Latest MPAM spec can be found [here](#)
- ❑ System characteristics to allow use of the resctrl interface :
 - Cache Portion bitmaps on L2 or L3, with up to 32 portions
 - Memory Bandwidth portion bitmaps on L3 with up to 32 portions

Arm MPAM controls	Resctrl interface
Cache portion bitmaps (L2 & L3 cache)	Available
Cache capacity	Not Available
Memory bandwidth portion bitmap	Available
Memory bandwidth min/max/stride	Not available
Raw priority	Not available

Arm 8.4 | 8.6 MPAM Enablement

x86 Refactoring – Near Completion

- ☐ Merge of the penultimate resctrl refactoring work in [kernel 6.9](#)
 - ☐ Most complex chunk of the resctrl refactoring work | **Monitoring and locking** series merged
 - ☐ Now, Last leg of the resctrl refactoring patches v2 [on-list](#); being actively reviewed
- ☐ Pivot to arm64 driver enablement – expect to be faster as no regression testing needed
- ☐ One-off **5.15 backport** available for partner ecosystem [here](#)

- ☐ ACPI specification is [now v2.0](#); (v1.0 has been deprecated)
- ☐ A draft Device Tree binding exists in the prototype tree; likely to change before upstream merge

- ☐ Ongoing development | [Projected landing estimates](#)
 - ☐ Arm64 driver prerequisite support – [6.11](#)
 - ☐ partID narrowing support – [6.12](#)
 - ☐ Firmware interface, PMU support
- ☐ KVM support
 - ☐ Intel RDT does not support virtualisation
 - ☐ Cost of rebasing KVM patches in parallel is high so aim to push them after a formative arm64 driver landing
- ☐ Refer to the latest branch on kernel.org with its list of known issues | [example](#)

Folios : Memory system optimisation

- ❑ Improving performance of Kernel [4K] by managing memory in variable size blocks instead of fixed block

[Also applicable for 16|64K Kernels that benefit from enablement of 2MB THP]

- ❑ Bigger block allocation on need basis; more efficient TLB usage; fewer misses and page faults
- ❑ Compatibility with existing 4K granularity user space ABI
- ❑ Kernel space still uses 4K so no extra mem. Cost
 - ❑ User space has Large Folios when they fit inside VMA [Fragmentation cost possible]
- ❑ Upstream Status + [Projected Landing](#)
 - ❑ **Allocating large Folios**
 - ❑ Multi-size THP : [6.8 -rc1](#)
 - ❑ File backed Memory gen. support since 6.1 | Client support [on-list](#)
 - ❑ [Large Folio Compaction](#) | [6.9](#)
 - ❑ [Swap-out](#) | [6.10](#) [Queued] ; [Swap-in](#) | [6.11](#)
 - ❑ **CONTPTTE Enablement**
 - ❑ Contpte Support : [6.9](#)
 - ❑ [ExeFolio](#) : [executable File backed mapping](#) : [6.11](#)
 - ❑ **Future**
 - ❑ [Frag. Analysis](#) | [Khugepaged Mechanism](#)
 - ❑ [Per VMA automatic Folio size determination](#) | [Trade off -> perf improv](#) | [Int fragmentation](#)

- ❑ Large scale deployments started

- ❑ Ubuntu reporting [%19 improvement](#) on kernel compile times
- ❑ Nvidia reporting [x10 improvement](#) on certain work loads
- ❑ Significant improvements [in boot time](#) [adjacent change to optimize barriers]

Arm Specification Support

<https://developer.arm.com/Architectures/Software%20Standards#Technical-Information>

- ☐ FF-A (SPCI)
 - ☐ FF-A 1.1 : Linux Kernel Notification | Memory Layout support merged in **Kernel 6.6**
 - ☐ FF-A User space [v4](#) : Aiming **6.10**
 - ☐ Active investigation cycles against FF-A 1.2 ALP1 spec | Development -> pick up in upcoming cycles

- ☐ SCMI
 - ☐ SCMI unidirectional mailbox, 3.2 power cap support merged upstream
 - ☐ Review support for vendor protocol | NXP | QC
 - ☐ For more info, tune in to these talks
 - ☐ [FF-A in KVM, May 16, 10:55](#) | [SCMI : Dev and Testing, May 16, 14:20](#)

- ☐ UEFI secure variable using standalone MM | FVP merged
- ☐ Capsule Update using '[Platform Security Firmware Update for the A-profile Arm Architecture](#)' | Internal Review
- ☐ Dynamic Tables Reorg : enable arch agnostic adoption | [Staging Branch](#) | progressing well

SPE | PMU Support

❑ Interconnect PMU Support

- ❑ CMN-700 Enhancement => HN-S Support merged in Kernel 6.6
- ❑ NI-700 support patches on-list; Tentative kernel Landing 6.10
- ❑ Neoverse - CMN S3 support being actively developed

❑ SPE | PMU

- ❑ 2020 Support => Merged in kernel 6.3 | 6.4
- ❑ 2021 SPE | PMU Support => SPE => 6.4 | PMU => 6.7 | PMU Threshold event – [6.8](#) ; KVM support – WiP
- ❑ 2022 PMU & SPE extensions
 - SPE additions => relatively small, require user-space work for new packets
 - PMU additions => possibly substantial kernel work required
 - SEBEP (roughly equivalent to x86 PEBS – [Begun Prototyping](#))
 - Rework of exception handling / masking
 - System PMU
 - firmware work for enumeration
 - PMU events

Other Highlights

- ❑ pKVM Support
 - ❑ pKVM hypervisor provides a separation of privileges between host and hypervisor parts of KVM
 - ❑ A new hypervisor driver for the Arm/Linaro SMMUv3 IOMMU introduced ([public tree](#))
 - ❑ Reference power management solution based on SCMI
 - ❑ Included in [TC22](#) solution stack
 - ❑ Merged in [Android 15](#) GKI
- ❑ Open-Source GPU development
 - ❑ Design goals: Kernel driver could support both DDK and Panfrost user-space drivers
 - ❑ “Panthor” Kernel driver for Mali CSF Valhall (Mali-G6x/Mali-Gxxx/Immortalis) GPUs ([queued for v6.10](#))
 - ❑ Collaboration with Collabora for the Kernel Driver and Mesa support

Confidential Compute : what, where and when!

- ❑ The Arm CCA allows the hypervisor to control the VM, but removes the right to access the code, register state or data that is used by the VM
- ❑ [Linux Kernel | KVM | EDK2](#) patches on-list against 6.9 -rc1 for [RMM 1.0 EAC5](#)
- ❑ [Arm Confidential Compute Architecture open-source enablement | Fri, 17 : 10:55-11:20 AM](#)

Resources and further reading

- ☐ [Linux Kernel Arm developer page](#)
- ☐ [Armv9-A Realm Management Extension](#)
- ☐ [ETE | TRBE Trace Guide](#)
- ☐ [Arm8.5-A Memory Tagging White Paper](#)
- ☐ [Arm Generic Interrupt Controller v3 and v4 - Virtualization](#)
- ☐ [Memory System Resource Partitioning and Monitoring \(MPAM\) Overview](#)
- ☐ [Aarch64 Exception Model](#)
- ☐ [Large Physical Address Extension](#)
- ☐ [GNU Toolchain Developer Page](#)
- ☐ [LLVM Toolchain Developer Page](#)
- ☐ [Architecture and Processor Community Blog Support](#)



Thank you

