

# Paravirtualized time support for arm64

Arm specification DEN0057/A defines a standard for paravirtualised time support for AArch64 guests:

<https://developer.arm.com/docs/den0057/a>

KVM/arm64 implements the stolen time part of this specification by providing some hypervisor service calls to support a paravirtualized guest obtaining a view of the amount of time stolen from its execution.

Two new SMCCC compatible hypercalls are defined:

- PV\_TIME\_FEATURES: 0xC5000020
- PV\_TIME\_ST: 0xC5000021

These are only available in the SMC64/HVC64 calling convention as paravirtualized time is not available to 32 bit Arm guests. The existence of the PV\_TIME\_FEATURES hypercall should be probed using the SMCCC 1.1 ARCH\_FEATURES mechanism before calling it.

## PV\_TIME\_FEATURES

Function ID:	(uint32)	0xC5000020
PV_call_id:	(uint32)	The function to query for support. Currently only PV_TIME_ST is supported.
Return value:	(int64)	NOT_SUPPORTED (-1) or SUCCESS (0) if the relevant PV-time feature is supported by the hypervisor.

## PV\_TIME\_ST

Function ID:	(uint32)	0xC5000021
Return value:	(int64)	IPA of the stolen time data structure for this VCPU. On failure: NOT_SUPPORTED (-1)

The IPA returned by PV\_TIME\_ST should be mapped by the guest as normal memory with inner and outer write back caching attributes, in the inner shareable domain. A total of 16 bytes from the IPA returned are guaranteed to be meaningfully filled by the hypervisor (see structure below).

PV\_TIME\_ST returns the structure for the calling VCPU.

## Stolen Time

The structure pointed to by the PV\_TIME\_ST hypercall is as follows:

Field	Byte Length	Byte Offset	Description
Revision	4	0	Must be 0 for version 1.0
Attributes	4	4	Must be 0
Stolen time	8	8	Stolen time in unsigned nanoseconds indicating how much time this VCPU thread was involuntarily not running on a physical CPU.

All values in the structure are stored little-endian.

The structure will be updated by the hypervisor prior to scheduling a VCPU. It will be present within a reserved region of the normal memory given to the guest. The guest should not attempt to write into this memory. There is a structure per VCPU of the guest.

It is advisable that one or more 64k pages are set aside for the purpose of these structures and not used for other purposes, this enables the guest to map the region using 64k pages and avoids conflicting attributes with other memory.

For the user space interface see Documentation/virt/kvm/devices/vcpu.rst section "3. GROUP: KVM\_ARM\_VCPU\_PVTIME\_CTRL".