EFI_MEMORY_ACCEPTANCE_PROTOCOL

Author: dionnaglaze@google.com

Date created: 2022-10-11

Document UEFI Specification 2.10

Submitter TianoCore Community

Summary of the change

Specifies expected UEFI memory acceptance behavior to accept all memory eagerly before ExitBootServices returns. Also adds a new protocol to disable the eager acceptance behavior to “allow” unaccepted memory to pass through to the loaded image.

Benefits of the change

We solve the following problem:
The memory type introduced in UEFI specification v2.9, EfiUnacceptedMemory, does not have a specified way for operating systems to safely upgrade from not supporting the type to supporting the type. The UEFI can choose to either pose a memory map including the type or not. There is no way for an OS loader to inform the UEFI that it supports the type or not.

Memory acceptance for the entire virtual RAM space is not required for booting a system. Support for a Confidential compute (CC) technology does not require supporting unaccepted memory, so the UEFI cannot simply choose to use the unaccepted memory type when it detects that it is running on a CC technology that requires memory acceptance.

The behavior of the UEFI for CC technologies before supporting the unaccepted memory type is to eagerly accept all memory. This behavior is required since otherwise there was no way to indicate down the boot chain which regions of memory have not yet been accepted. We take eager acceptance as the expected and correct behavior for the UEFI when booting the next boot stage without being told otherwise.

Operating systems that support the EfiUnacceptedMemory memory type are able to dynamically accept memory that was not pre-accepted by the UEFI. Without the EfiUnacceptedMemory support in the guest OS kernel, the boot time will take longer due to:

1. UEFI must accept the whole memory allocated for the guest images before transferring control to the guest OS kernel
2. While accepting the pages, because of confidential computing requirements, the host has to allocate the backing pages and pin them to prevent guest memory to be migrated to new host physical addresses.

Currently UEFI cannot know if EfiUnacceptedMemory is supported in the guest OS. This document proposes a protocol to tell the UEFI not to accept all memory when EfiUnacceptedMemory is supported in the guest OS kernel.

Impact of the change

This is an optional protocol for OS loaders to opt into getting the unaccepted memory type in their memory map.

Code first

The latest version of this proposal is at https://github.com/deeglaze/edk2/tree/enable_umv8. Following code-first policy, the names are prefixed with the bugzilla issue number for discussing the specification change, Bz3987.

Detailed description of the change

Changes to section 8.5.4: Exchanging information between the OS and Firmware

> *The firmware and an Operating System may exchange information through the OsIndicationsSupported and the OSIndications variables as follows:*

to become

> The firmware and an Operating System may exchange information through protocols defined in [~~Protocols - OS to Firmware Communication~~Confidential Computing](#chapter~~21~~38) and the OsIndicationsSupported and the OSIndications variables. The OsIndicationsSupported and OSIndications variable usage is as follows:

Addition of section to Chapter 38 - Confidential Computing~~chapter between Chapter 20 Protocols - ACPI Protocols~~ and ~~Chapter 21 String Services~~
~~Text follows~~

~~Chapter 21~~38.5 ~~Protocols - OS to Firmware communication~~

~~There are situations such as a multi-boot system where a single system-wide bit in OsIndications is not suitable, and persistence is not necessary. In those cases, we specify protocols to communicate support for UEFI features that cannot be known ahead of time.~~

~~Section 21.1 –~~ EFI_MEMORY_ACCEPTANCE_PROTOCOL

There are situations such as a multi-boot system where a single system-wide bit in OsIndications is not suitable, and persistence is not necessary. In those cases, we specify protocols to communicate support for UEFI features that cannot be known ahead of time.

38.5.1 - EFI_MEMORY_ACCEPTANCE_PROTOCOL

Summary

This protocol solves the problem of allowing EfiUnacceptedMemoryType in the memory map for an OS that understands the type, and disallowing it for an OS that doesn't understand it. This upgrade problem exists since EfiUnacceptedMemoryType was not available in UEFI version 1.0.

A UEFI that provides this protocol allows an OS to have access to EfiUnacceptedMemoryType entries in its memory map. The OS has until it calls ExitBootServices to call EFI_MEMORY_ACCEPTANCE_PROTOCOL.AllowUnacceptedMemory() to indicate that not all memory should be accepted.

If EFI_MEMORY_ACCEPTANCE_PROTOCOL is provided, the UEFI shall not accept all memory until ExitBootServices() is called. Otherwise, the UEFI must accept all memory by or during OnReadyToBoot.

A UEFI that provides this protocol shall not have any unaccepted memory below 4GB by or during OnReadyToBoot. This ensures that an OS that doesn't support unaccepted memory can run to ExitBootServices() without an unaccepted memory access error.

~~The UEFI shall accept all memory any time before ExitBootServices returns if it does not provide this protocol.~~ Accepted EfiUnacceptedMemoryType regions shall be reclassified as EfiConventionalMemory. ~~The UEFI shall not accept all memory until ExitBootServices is called if the protocol is provided.~~

~~This protocol and acceptance behavior ensures that the UEFI safely supports OSes that don't interpret the new EfiUnacceptedMemory type and those that do.~~

GUID

```
#define EFI_MEMORY_ACCEPTANCE_PROTOCOL_GUID \
  {0xc5a010fe,0x38a7,0x4531, \
    {0x8a,0x4a,0x05,0x00,0xd2,0xfd,0x16,0x49}}
```

Protocol Interface Structure

```
typedef struct _EFI_MEMORY_ACCEPTANCE_PROTOCOL {
  EFI_ALLOW_UNACCEPTED_MEMORY  AllowUnacceptedMemory;
} EFI_MEMORY_ACCEPTANCE_PROTOCOL;
```

Parameters

AllowUnacceptedMemory Disables behavior of accepting all memory before ExitBootServices returns. See the [EFI_MEMORY_ACCEPTANCE_PROTOCOL.~~Disable~~AllowUnacceptedMemory()](#section~~21~~38.~~15~~.~~12~~) function description.

Description

This interface does not provide any way to re-enable the behavior of accepting all memory.

~~Section 21.1.1~~38.5.2 EFI_MEMORY_ACCEPTANCE_PROTOCOL.AllowUnacceptedMemory()

Summary

Disables behavior of accepting all memory before ExitBootServices returns.

Prototype

```
typedef
EFI_STATUS
(EFIAPI *EFI_ALLOW_UNACCEPTED_MEMORY)(
  IN  EFI_MEMORY_ACCEPTANCE_PROTOCOL  *This
  );
```

Parameters

This A pointer to EFI_MEMORY_ACCEPTANCE_PROTOCOL instance. Type EFI_MEMORY_ACCEPTANCE_PROTOCOL is defined in [EFI_MEMORY_ACCEPTANCE_PROTOCOL](#section~~21~~38.5.1)

Description

Meant to be called by OS loader before ExitBootServices if the OS supports EfiUnacceptedMemoryType and means to assume the role of accepting memory before its use.