NAME
qemu−cpu−models − QEMU / KVM CPU model configuration

SYNOPSIS
QEMU / KVM CPU model configuration

DESCRIPTION
QEMU / KVM virtualization supports two ways to configure CPU models

Host passthrough
This passes the host CPU model features, model, stepping, exactly to the guest. Note that KVM may filter out some host CPU model features if they cannot be supported with virtualization. Live migration is unsafe when this mode is used as libvirt / QEMU cannot guarantee a stable CPU is exposed to the guest across hosts. This is the recommended CPU to use, provided live migration is not required.

Named model
QEMU comes with a number of predefined named CPU models, that typically refer to specific generations of hardware released by Intel and AMD. These allow the guest VMs to have a degree of isolation from the host CPU, allowing greater flexibility in live migrating between hosts with differing hardware.

In both cases, it is possible to optionally add or remove individual CPU features, to alter what is presented to the guest by default.

Libvirt supports a third way to configure CPU models known as ''Host model''. This uses the QEMU ''Named model'' feature, automatically picking a CPU model that is similar the host CPU, and then adding extra features to approximate the host model as closely as possible. This does not guarantee the CPU family, stepping, etc will precisely match the host CPU, as they would with ''Host passthrough'', but gives much of the benefit of passthrough, while making live migration safe.

Recommendations for KVM CPU model configuration on x86 hosts

The information that follows provides recommendations for configuring CPU models on x86 hosts. The goals are to maximise performance, while protecting guest OS against various CPU hardware flaws, and optionally enabling live migration between hosts with heterogeneous CPU models.

Preferred CPU models for Intel x86 hosts

The following CPU models are preferred for use on Intel hosts. Administrators / applications are recommended to use the CPU model that matches the generation of the host CPUs in use. In a deployment with a mixture of host CPU models between machines, if live migration compatibility is required, use the newest CPU model that is compatible across all desired hosts.

Skylake−Server
Skylake−Server−IBRS
Intel Xeon Processor (Skylake, 2016)

Skylake−Client
Skylake−Client−IBRS
Intel Core Processor (Skylake, 2015)

Broadwell
Broadwell−IBRS
Broadwell−noTSX
Broadwell−noTSX−IBRS
Intel Core Processor (Broadwell, 2014)

Haswell
Haswell−IBRS
Haswell−noTSX
Haswell−noTSX−IBRS
Intel Core Processor (Haswell, 2013)

IvyBridge
IvyBridge−IBRS
> Intel Xeon E3−12xx v2 (Ivy Bridge, 2012)

SandyBridge
SandyBridge−IBRS
> Intel Xeon E312xx (Sandy Bridge, 2011)

Westmere
Westmere−IBRS
> Westmere E56xx/L56xx/X56xx (Nehalem-C, 2010)

Nehalem
Nehalem−IBRS
> Intel Core i7 9xx (Nehalem Class Core i7, 2008)

Penryn
> Intel Core 2 Duo P9xxx (Penryn Class Core 2, 2007)

Conroe
> Intel Celeron_4x0 (Conroe/Merom Class Core 2, 2006)

Important CPU features for Intel x86 hosts

The following are important CPU features that should be used on Intel x86 hosts, when available in the host CPU. Some of them require explicit configuration to enable, as they are not included by default in some, or all, of the named CPU models listed above. In general all of these features are included if using ''Host passthrough'' or ''Host model''.

pcid
> Recommended to mitigate the cost of the Meltdown (CVE−2017−5754) fix
>
> Included by default in Haswell, Broadwell & Skylake Intel CPU models.
>
> Should be explicitly turned on for Westmere, SandyBridge, and IvyBridge Intel CPU models. Note that some desktop/mobile Westmere CPUs cannot support this feature.

spec−ctrl
> Required to enable the Spectre v2 (CVE−2017−5715) fix.
>
> Included by default in Intel CPU models with −IBRS suffix.
>
> Must be explicitly turned on for Intel CPU models without −IBRS suffix.
>
> Requires the host CPU microcode to support this feature before it can be used for guest CPUs.

stibp
> Required to enable stronger Spectre v2 (CVE−2017−5715) fixes in some operating systems.
>
> Must be explicitly turned on for all Intel CPU models.
>
> Requires the host CPU microcode to support this feature before it can be used for guest CPUs.

ssbd
> Required to enable the CVE−2018−3639 fix
>
> Not included by default in any Intel CPU model.
>
> Must be explicitly turned on for all Intel CPU models.
>
> Requires the host CPU microcode to support this feature before it can be used for guest CPUs.

pdpe1gb
> Recommended to allow guest OS to use 1GB size pages
>
> Not included by default in any Intel CPU model.
>
> Should be explicitly turned on for all Intel CPU models.

Note that not all CPU hardware will support this feature.

md-clear
Required to confirm the MDS (CVE−2018−12126, CVE−2018−12127, CVE−2018−12130, CVE−2019−11091) fixes.

Not included by default in any Intel CPU model.

Must be explicitly turned on for all Intel CPU models.

Requires the host CPU microcode to support this feature before it can be used for guest CPUs.

Preferred CPU models for AMD x86 hosts

The following CPU models are preferred for use on Intel hosts. Administrators / applications are recommended to use the CPU model that matches the generation of the host CPUs in use. In a deployment with a mixture of host CPU models between machines, if live migration compatibility is required, use the newest CPU model that is compatible across all desired hosts.

EPYC
EPYC−IBPB
AMD EPYC Processor (2017)

Opteron_G5
AMD Opteron 63xx class CPU (2012)

Opteron_G4
AMD Opteron 62xx class CPU (2011)

Opteron_G3
AMD Opteron 23xx (Gen 3 Class Opteron, 2009)

Opteron_G2
AMD Opteron 22xx (Gen 2 Class Opteron, 2006)

Opteron_G1
AMD Opteron 240 (Gen 1 Class Opteron, 2004)

Important CPU features for AMD x86 hosts

The following are important CPU features that should be used on AMD x86 hosts, when available in the host CPU. Some of them require explicit configuration to enable, as they are not included by default in some, or all, of the named CPU models listed above. In general all of these features are included if using ''Host passthrough'' or ''Host model''.

ibpb
Required to enable the Spectre v2 (CVE−2017−5715) fix.

Included by default in AMD CPU models with −IBPB suffix.

Must be explicitly turned on for AMD CPU models without −IBPB suffix.

Requires the host CPU microcode to support this feature before it can be used for guest CPUs.

stibp
Required to enable stronger Spectre v2 (CVE−2017−5715) fixes in some operating systems.

Must be explicitly turned on for all AMD CPU models.

Requires the host CPU microcode to support this feature before it can be used for guest CPUs.

virt−ssbd
Required to enable the CVE−2018−3639 fix

Not included by default in any AMD CPU model.

Must be explicitly turned on for all AMD CPU models.

This should be provided to guests, even if amd-ssbd is also provided, for maximum guest compatibility.

Note for some QEMU / libvirt versions, this must be force enabled when when using ''Host model'', because this is a virtual feature that doesn't exist in the physical host CPUs.

amd−ssbd
Required to enable the CVE−2018−3639 fix

Not included by default in any AMD CPU model.

Must be explicitly turned on for all AMD CPU models.

This provides higher performance than virt-ssbd so should be exposed to guests whenever available in the host. virt-ssbd should none the less also be exposed for maximum guest compatibility as some kernels only know about virt-ssbd.

amd−no−ssb
Recommended to indicate the host is not vulnerable CVE−2018−3639

Not included by default in any AMD CPU model.

Future hardware generations of CPU will not be vulnerable to CVE−2018−3639, and thus the guest should be told not to enable its mitigations, by exposing amd-no-ssb. This is mutually exclusive with virt-ssbd and amd-ssbd.

pdpe1gb
Recommended to allow guest OS to use 1GB size pages

Not included by default in any AMD CPU model.

Should be explicitly turned on for all AMD CPU models.

Note that not all CPU hardware will support this feature.

Default x86 CPU models

The default QEMU CPU models are designed such that they can run on all hosts. If an application does not wish to do perform any host compatibility checks before launching guests, the default is guaranteed to work.

The default CPU models will, however, leave the guest OS vulnerable to various CPU hardware flaws, so their use is strongly discouraged. Applications should follow the earlier guidance to setup a better CPU configuration, with host passthrough recommended if live migration is not needed.

qemu32
qemu64
QEMU Virtual CPU version 2.5+ (32 & 64 bit variants)

qemu64 is used for x86_64 guests and qemu32 is used for i686 guests, when no −cpu argument is given to QEMU, or no <cpu> is provided in libvirt XML.

Other non-recommended x86 CPUs

The following CPUs models are compatible with most AMD and Intel x86 hosts, but their usage is discouraged, as they expose a very limited featureset, which prevents guests having optimal performance.

kvm32
kvm64
Common KVM processor (32 & 64 bit variants)

Legacy models just for historical compatibility with ancient QEMU versions.

486
athlon

phenom
coreduo
core2duo
n270
pentium
pentium2
pentium3
> Various very old x86 CPU models, mostly predating the introduction of hardware assisted virtualization, that should thus not be required for running virtual machines.

Supported CPU model configurations on MIPS hosts

QEMU supports variety of MIPS CPU models:

Supported CPU models for MIPS32 hosts

The following CPU models are supported for use on MIPS32 hosts. Administrators / applications are recommended to use the CPU model that matches the generation of the host CPUs in use. In a deployment with a mixture of host CPU models between machines, if live migration compatibility is required, use the newest CPU model that is compatible across all desired hosts.

mips32r6-generic
> MIPS32 Processor (Release 6, 2015)

P5600
> MIPS32 Processor (P5600, 2014)

M14K
M14Kc
> MIPS32 Processor (M14K, 2009)

74Kf
> MIPS32 Processor (74K, 2007)

34Kf
> MIPS32 Processor (34K, 2006)

24Kc
24KEc
24Kf
> MIPS32 Processor (24K, 2003)

4Kc
4Km
4KEcR1
4KEmR1
4KEc
4KEm
> MIPS32 Processor (4K, 1999)

Supported CPU models for MIPS64 hosts

The following CPU models are supported for use on MIPS64 hosts. Administrators / applications are recommended to use the CPU model that matches the generation of the host CPUs in use. In a deployment with a mixture of host CPU models between machines, if live migration compatibility is required, use the newest CPU model that is compatible across all desired hosts.

I6400
> MIPS64 Processor (Release 6, 2014)

Loongson-2F
> MIPS64 Processor (Loongson 2, 2008)

Loongson-2E
> MIPS64 Processor (Loongson 2, 2006)

mips64dspr2
> MIPS64 Processor (Release 2, 2006)

MIPS64R2-generic
5KEc
5KEf
> MIPS64 Processor (Release 2, 2002)

20Kc
> MIPS64 Processor (20K, 2000)

5Kc
5Kf
> MIPS64 Processor (5K, 1999)

VR5432
> MIPS64 Processor (VR, 1998)

R4000
> MIPS64 Processor (MIPS III, 1991)

Supported CPU models for nanoMIPS hosts

The following CPU models are supported for use on nanoMIPS hosts. Administrators / applications are recommended to use the CPU model that matches the generation of the host CPUs in use. In a deployment with a mixture of host CPU models between machines, if live migration compatibility is required, use the newest CPU model that is compatible across all desired hosts.

I7200
> MIPS I7200 (nanoMIPS, 2018)

Preferred CPU models for MIPS hosts

The following CPU models are preferred for use on different MIPS hosts:

MIPS III
> R4000

MIPS32R2
> 34Kf

MIPS64R6
> I6400

nanoMIPS
> I7200

Syntax for configuring CPU models

The example below illustrate the approach to configuring the various CPU models / features in QEMU and libvirt

QEMU command line

Host passthrough

```
$ qemu-system-x86_64 -cpu host
```

With feature customization:

```
$ qemu-system-x86_64 -cpu host,-vmx,...
```

Named CPU models

```
$ qemu-system-x86_64 -cpu Westmere
```

With feature customization:

```
$ qemu-system-x86_64 -cpu Westmere,+pcid,...
```

Libvirt guest XML

Host passthrough

```
<cpu mode='host-passthrough'/>
```

With feature customization:

```
<cpu mode='host-passthrough'>
    <feature name="vmx" policy="disable"/>
    ...
</cpu>
```

Host model

```
<cpu mode='host-model'/>
```

With feature customization:

```
<cpu mode='host-model'>
    <feature name="vmx" policy="disable"/>
    ...
</cpu>
```

Named model

```
<cpu mode='custom'>
    <model name="Westmere"/>
</cpu>
```

With feature customization:

```
<cpu mode='custom'>
    <model name="Westmere"/>
    <feature name="pcid" policy="require"/>
    ...
</cpu>
```

SEE ALSO

The HTML documentation of QEMU for more precise information and Linux user mode emulator invocation.

AUTHOR

Daniel P. Berrange