AMD64 Specific Boot Options

There are many others (usually documented in driver documentation), but only the AMD64 specific ones are listed here.

Machine check

Please see Documentation/x86/x86_64/machinecheck.rst for sysfs runtime tunables.

mce=off
: Disable machine check

mce=no_cmci
: Disable CMCI(Corrected Machine Check Interrupt) that Intel processor supports. Usually this disablement is not recommended, but it might be handy if your hardware is misbehaving. Note that you'll get more problems without CMCI than with due to the shared banks, i.e. you might get duplicated error logs.

mce=dont_log_ce
: Don't make logs for corrected errors. All events reported as corrected are silently cleared by OS. This option will be useful if you have no interest in any of corrected errors.

mce=ignore_ce
: Disable features for corrected errors, e.g. polling timer and CMCI. All events reported as corrected are not cleared by OS and remained in its error banks. Usually this disablement is not recommended, however if there is an agent checking/clearing corrected errors (e.g. BIOS or hardware monitoring applications), conflicting with OS's error handling, and you cannot deactivate the agent, then this option will be a help.

mce=no_lmce
: Do not opt-in to Local MCE delivery. Use legacy method to broadcast MCEs.

mce=bootlog
: Enable logging of machine checks left over from booting. Disabled by default on AMD Fam10h and older because some BIOS leave bogus ones. If your BIOS doesn't do that it's a good idea to enable though to make sure you log even machine check events that result in a reboot. On Intel systems it is enabled by default.

mce=nobootlog
: Disable boot machine check logging.

mce=monarchtimeout (number)
: monarchtimeout: Sets the time in us to wait for other CPUs on machine checks. 0 to disable.

mce=bios_cmci_threshold
: Don't overwrite the bios-set CMCI threshold. This boot option prevents Linux from overwriting the CMCI threshold set by the bios. Without this option, Linux always sets the CMCI threshold to 1. Enabling this may make memory predictive failure analysis less effective if the bios sets thresholds for memory errors since we will not see details for all errors.

mce=recovery
: Force-enable recoverable machine check code paths

nomce (for compatibility with i386)
: same as mce=off

Everything else is in sysfs now.

APICs

apic
: Use IO-APIC. Default

noapic
: Don't use the IO-APIC.

disableapic
: Don't use the local APIC

nolapic
: Don't use the local APIC (alias for i386 compatibility)

pirq=...
: See Documentation/x86/i386/IO-APIC.rst

noapictimer
: Don't set up the APIC timer

no_timer_check
: Don't check the IO-APIC timer. This can work around problems with incorrect timer initialization on some boards.

apicpmtimer
: Do APIC timer calibration using the pmtimer. Implies apicmaintimer. Useful when your PIT timer is totally

broken.

Timing

notsc
: Deprecated, use tsc=unstable instead.

nohpet
: Don't use the HPET timer.

Idle loop

idle=poll
: Don't do power saving in the idle loop using HLT, but poll for rescheduling event. This will make the CPUs eat a lot more power, but may be useful to get slightly better performance in multiprocessor benchmarks. It also makes some profiling using performance counters more accurate. Please note that on systems with MONITOR/MWAIT support (like Intel EM64T CPUs) this option has no performance advantage over the normal idle loop. It may also interact badly with hyperthreading.

Rebooting

reboot=b[ios] | t[riple] | k[bd] | a[cpi] | e[fi] | p[ci] [, [w]arm | [c]old]

bios
: Use the CPU reboot vector for warm reset

warm
: Don't set the cold reboot flag

cold
: Set the cold reboot flag

triple
: Force a triple fault (init)

kbd
: Use the keyboard controller. cold reset (default)

acpi
: Use the ACPI RESET_REG in the FADT. If ACPI is not configured or the ACPI reset does not work, the reboot path attempts the reset using the keyboard controller.

efi
: Use efi reset_system runtime service. If EFI is not configured or the EFI reset does not work, the reboot path attempts the reset using the keyboard controller.

pci
: Use a write to the PCI config space register 0xcf9 to trigger reboot.

Using warm reset will be much faster especially on big memory systems because the BIOS will not go through the memory check. Disadvantage is that not all hardware will be completely reinitialized on reboot so there may be boot problems on some systems.

reboot=force
: Don't stop other CPUs on reboot. This can make reboot more reliable in some cases.

reboot=default
: There are some built-in platform specific "quirks" - you may see: "reboot: <name> series board detected. Selecting <type> for reboots." In the case where you think the quirk is in error (e.g. you have newer BIOS, or newer board) using this option will ignore the built-in quirk table, and use the generic default reboot actions.

Non Executable Mappings

noexec=on|off

on
: Enable(default)

off
: Disable

NUMA

numa=off
: Only set up a single NUMA node spanning all memory.

numa=noacpi

Don't parse the SRAT table for NUMA setup

numa=nohmat
: Don't parse the HMAT table for NUMA setup, or soft-reserved memory partitioning.

numa=fake=<size>[MG]
: If given as a memory unit, fills all system RAM with nodes of size interleaved over physical nodes.

numa=fake=<N>
: If given as an integer, fills all system RAM with N fake nodes interleaved over physical nodes.

numa=fake=<N>U
: If given as an integer followed by 'U', it will divide each physical node into N emulated nodes.

ACPI

acpi=off
: Don't enable ACPI

acpi=ht
: Use ACPI boot table parsing, but don't enable ACPI interpreter

acpi=force
: Force ACPI on (currently not needed)

acpi=strict
: Disable out of spec ACPI workarounds.

acpi_sci={edge,level,high,low}
: Set up ACPI SCI interrupt.

acpi=noirq
: Don't route interrupts

acpi=nocmcff
: Disable firmware first mode for corrected errors. This disables parsing the HEST CMC error source to check if firmware has set the FF flag. This may result in duplicate corrected error reports.

PCI

pci=off
: Don't use PCI

pci=conf1
: Use conf1 access.

pci=conf2
: Use conf2 access.

pci=rom
: Assign ROMs.

pci=assign-busses
: Assign busses

pci=irqmask=MASK
: Set PCI interrupt mask to MASK

pci=lastbus=NUMBER
: Scan up to NUMBER busses, no matter what the mptable says.

pci=noacpi
: Don't use ACPI to set up PCI interrupt routing.

IOMMU (input/output memory management unit)

Multiple x86-64 PCI-DMA mapping implementations exist, for example:

1. <kernel/dma/direct.c>: use no hardware/software IOMMU at all (e.g. because you have < 3 GB memory). Kernel boot message: "PCI-DMA: Disabling IOMMU"
2. <arch/x86/kernel/amd_gart_64.c>: AMD GART based hardware IOMMU. Kernel boot message: "PCI-DMA: using GART IOMMU"
3. <arch/x86_64/kernel/pci-swiotlb.c> : Software IOMMU implementation. Used e.g. if there is no hardware IOMMU in the system and it is need because you have >3GB memory or told the kernel to us it (iommu=soft)) Kernel boot message: "PCI-DMA: Using software bounce buffering for IO (SWIOTLB)"

```
iommu=[<size>][,noagp][,off][,force][,noforce]
[,memaper[=<order>]][,merge][,fullflush][,nomerge]
[,noaperture]
```

General iommu options:

off

Don't initialize and use any kind of IOMMU.

noforce
: Don't force hardware IOMMU usage when it is not needed. (default).

force
: Force the use of the hardware IOMMU even when it is not actually needed (e.g. because < 3 GB memory).

soft
: Use software bounce buffering (SWIOTLB) (default for Intel machines). This can be used to prevent the usage of an available hardware IOMMU.

iommu options only relevant to the AMD GART hardware IOMMU:

<size>
: Set the size of the remapping area in bytes.

allowed
: Overwrite iommu off workarounds for specific chipsets.

fullflush
: Flush IOMMU on each allocation (default).

nofullflush
: Don't use IOMMU fullflush.

memaper[=<order>]
: Allocate an own aperture over RAM with size 32MB<<order. (default: order=1, i.e. 64MB)

merge
: Do scatter-gather (SG) merging. Implies "force" (experimental).

nomerge
: Don't do scatter-gather (SG) merging.

noaperture
: Ask the IOMMU not to touch the aperture for AGP.

noagp
: Don't initialize the AGP driver and use full aperture.

panic
: Always panic when IOMMU overflows.

iommu options only relevant to the software bounce buffering (SWIOTLB) IOMMU implementation:

swiotlb=<pages>[,force]

<pages>
: Prereserve that many 128K pages for the software IO bounce buffering.

force
: Force all IO through the software TLB.

Miscellaneous

nogbpages
: Do not use GB pages for kernel direct mappings.

gbpages
: Use GB pages for kernel direct mappings.