

"Der neue Weg" macOS zu nutzen oder "The new way of building a Hackintosh"

Beitrag von „eldxmgw“ vom 6. Januar 2023, 21:06

Ich kann vermelden, dass ich die Monterey VM in ESXi 7.0U3g-20328353 nun vollkommen lauffähig eingerichtet bekommen habe.

- RX480 passthrough via HDMI wird nativ erkannt und ist voll Metal unterstützt. Es hängt mit nativer Auflösung und 60Hz an einem 34" 21:9 Monitor.
- VMware default VGA device habe ich im VMKernel lahm gelegt, die dann zwar die herkömmliche Bedienung der DCUI deaktiviert. Das ist aber nach initialer Einrichtung eh hinfällig, und wenn absolut notwendig schnell wieder rückgängig zu machen.
- Zwei RX480 passthrough configuration flags, insb. das für den path des VBIOS dumps habe ich der vmx hinzugefügt.
- RX480 HDMI sound passthrough funktioniert auch problemlos.
- Apple Keyboard (USB) und MX Master 3 Maus (USB Receiver) habe ich ebenso via passthrough über die vendor und device ID quirks in der boot.cfg, vmware config file und in der vmx eingetragen und der VM Konfiguration lauffähig hinzugefügt (btw. beide input devices hängen gar an einem manuellen USB Hub).
- USB 3.1 passthrough funktioniert auch anstandslos.
- Snapshots funktionieren entgegen so mancher Hinweise mit Passthrough VMs einwandfrei

Das Ganze ohne irgendwelche OC, extra kext oder sonstige OSX verarztungen. Vorhin habe ich auch das letzte aktuelle Overlay auf 12.6.2 drüber gebügelt. Die VM habe ich mir so eingerichtet dass ich sie direkt am Monitor über die Infrastruktur nutzen kann, oder via ARD. ESXi via WebUI. oder zentralisiert über das vCenter.

meld



*** VENTURA UPDATE ***, auch die neue Ventura VM läuft, wie zuvor die Monterey VM samt allen passthrough Geräte anstandslos. GPU mit Metal Unterstützung, USB 3.1, HDMI Audio, Maus + Keyboard passthrough alles out of the Box. An OSX wurde wie immer nach der Installation nichts verändert. Jetzt muss mal jemand erklären weshalb man sich noch mit OC & Co herumschlagen muss wenn das gar mit obsolete Skylake Architekturen klappt? Bilder als Proof -> siehe Bild 15 und 16. 😊

meld



Abschließend werde ich mir noch überlegen das Ganze mittels ESXi 8.0-20513097 nachzubauen. Ich bin mir im klaren darüber das hier seitens VMware in Gegensatz zu 7.x einige Hürden eingebaut wurden die MacOS erschweren generell laufen zu lassen, aber nicht unmöglich machen. Erfolg hatte ich schon, allerdings habe ich den ganzen Passthrough Umfang nicht aufgesetzt.

Jetzt könnte man noch darüber nachdenken den Hypervisor anzuweisen die OSX VM nach Systemstart -> Hypervisor wird initialisiert -> VM wird initialisiert -> passthrough ist aktiv - automagisch starten zu lassen, um dem Nutzer den Eindruck einer physischen Workstation mit single major release zu vermitteln. Was in den Layern drum herum läuft bekäme der in der Blase nicht mit. Im Prinzip wäre das wie es der Hackintosh klassisch vermittelte. Das ist alles in der ESXi bzw. VM Verwaltung einfach einzurichten. Man müsse sich nur die administrative Umsetzung durch den Kopf gehen lassen wie der Hypervisor reagiert, würde der Nutzer die OSX VM ausschalten und erwarten, dass die Workstation (eigentlich Host) sich ebenso wie traditionelles Blech verhält und sich ebenso ausschaltet. Ich denke das müsste per script geregelt werden (wenn man das will).

Im Prinzip sind die Schritte schnell und einfach zu erledigen.

(wer kein Passthrough anwenden will ist schon nach Punkt 7 fertig)

1. ESXi installieren und grundlegend einrichten
2. Passenden unlocker für die ESXi Version besorgen und anwenden (frei Verfügbar)
3. Gezogenes OSX Major release in ISO konvertieren
4. VMware Storage einrichten
5. VM erstellen und grundlegend einrichten. Nicht vergessen den NIC der VM als Adaptertyp mit VMXNET 3 zu versehen.
6. OSX Image ausrollen und grundlegend einrichten, und aktuelle VMware Tools manuell ziehen und unter OSX installieren.
7. ARD übers LAN aktivieren
8. Ich empfehle präventiv über Verwalten -> System -> Erweiterte Einstellungen über die Suche pcie zu bemühen und den Schlüssel VMkernel.Boot.disableACSCheck auf true zu setzen um die PCIe Funktionsprüfung bei Systemstart außer Kraft zu setzen. (Bild hängt anbei)
9. VMkernel VGA Treiber via ESXCLI command lahmlegen, damit die passthrough GPU devices nicht nach jedem Neustart wieder manuell aktiviert werden müssen Syntax:

- esxcli system settings kernel set -s vga -v FALSE (Vorsicht, ab dem Punkt quittiert die DCUI mit einem black screen s.o.) -> Neustart
10. Überprüfen ob in der Hypervisor Geräte Verwaltung die gewünschten passthrough Geräte aktiv sind, falls nicht manuell setzen -> Neustart
 11. In der VM Konfiguration die PCIe passthrough GPU als PCIe device hinzufügen. Nicht die dynamische PCIe Variante nutzen. Dies hat zumindest direkt über den Hypervisor nicht funktioniert.
 12. ****ANMERKUNG: SIEHE UNTEN**** svga.present in der vmx auf FALSE setzen (Vorsicht, ab hier funktioniert die VM Webkonsole nicht mehr. Ist aber auch nicht wichtig da ARD aktiv ist, und zusätzlich passthrough eingerichtet wird)
 13. Passenden VBIOS dump besorgen oder eigenen auslesen. Hier gibts passende, genau auf Spezifikation achten. Beispiel hier: <https://www.techpowerup.com/vg...ercolor-rx480-8192-160923>
 14. VBIOS dump am besten direkt auf dem VM Storage ablegen.
 15. Neuen flag pciPassthru0.opromEnabled in der vmx auf TRUE setzen
 16. Neuen flag pciPassthru0.filename in der vmx mit dem richtigen Pfad zum VBIOS dump auf dem VM Storage setzen. Beispiel: ../Powercolor.RX480.8192.160923.rom
 17. VM Starten, PCIe passthrough sollte nun funktionieren nachdem der Monitor auf den richtigen Port gesetzt wurde. Ich will es nur erwähnen, ich habe gelesen dass PCIe GPU passthrough angeblich nur via HDMI und nicht via DP möglich ist. Ich kann das aktuell nicht überprüfen.
 18. Ggf. weitere PCIe passthrough Geräte wie HDMI sound oder USB 3.1 (falls vorhanden) in der VM Konfiguration als PCIe device hinzufügen.

Das wärs eigentlich. Wer jetzt noch gewisse HID USB Geräte durchschleifen will macht hier weiter....

- Via SSH auf den Hypervisor einloggen und mittels `lsusb -v | grep -E '(^Bus|HID)'` die vendor und device ID für die gewünschten HID passthrough USB Geräte ermitteln. Bei der Lesart sind die Werte nach der ID entscheidend (vendorID:deviceID)

Beispiel:

Bus 001 Device 002: ID 1a40:0101 Terminus Technology Inc. Hub

- Wichtig: im folgenden bekommen die Werte ein 0x vorgeschoben. Z.B. 0xXXXX = vendorId und 0xYYYY = deviceId.
- /etc/vmware/config mit vi konfigurieren und die eigenen ermittelte Werte nach folgenden Schema eintragen und speichern...

Beispiel:

usb.quirks.device0 = "0x05ac:0x0250 allow"

usb.quirks.device1 = "0x046d:0xc52b allow"

- Nun muss dem VMKernel mitgeteilt werden nicht ständig die USB Eingabegeräte für sich zu proklamieren. Dazu wird ein vi auf /bootbank/boot.cfg gemacht und die kernelopt Zeile anvisiert.

Beispiel:

```
kernelopt=autoPartition=FALSE
```

Dort wird direkt dahinter mit einem space mit der eigenen ermittelten vendor und device ID folgende Lesart hinterlegt **vendorID:deviceID:minRevision:maxRevision:quirkName**

Beispiel für ein Gerät:

```
CONFIG./USB/quirks=0x05ac:0x0250:0x0000:0xffff:UQ_KBD_IGNORE
```

Beispiel für zwei Geräte:

```
CONFIG./USB/quirks=0x05ac:0x0250:0x0000:0xffff:UQ_KBD_IGNORE:0x046d:0xc52b:0x0000:0xffff:UQ_KBD_
```

Beispiel der ganzen kernelopt Zeile anhand zwei Geräte:

```
kernelopt=autoPartition=FALSE
```

```
CONFIG./USB/quirks=0x05ac:0x0250:0x0000:0xffff:UQ_KBD_IGNORE:0x046d:0xc52b:0x0000:0xffff:UQ_KBD_
```

- Der VM muss das passthrough der USB HID Geräte ebenso anhand der vendor und device ID mitgeteilt werden. Dies kann via ESXi WebUI oder shell gemacht werden. Hier können die Werte wie zuvor in der /etc/vmware/config eingetragen übernommen werden. Die vmx findet man in der shell unter /vmfs/volumes/VM Storage (share)/.....

Beispiel Auszug der vmx mit den neu eingetragenen Werte in der shell:

```
usb.generic.allowHID = "TRUE"
```

```
usb.quirks.device0 = "0x05ac:0x0250 allow"
```

```
usb.quirks.device1 = "0x046d:0xc52b allow"
```

- Hypervisor Neustart
- Zuletzt muss der VM Konfiguration noch die neuen USB HID Geräte in der ESXi WebUI hinzugefügt werden. Danach ist alles erledigt und sollte den ganz oben genannten Stand mit allen PCIe + USB HID passthrough Geräte entsprechen.

****ANMERKUNG****

Ab Punkt 11 erwähne ich stets die vmx. Damit ist u.a. die Konfiguration der VM gemeint. Es gibt prinzipiell zwei Vorgehensweisen diese zu editieren. Über die ESXi WebUI in der GUI -> VM -> bearbeiten -> VM Optionen -> Erweitert -> Konfigurationsparameter -> Konfiguration bearbeiten.

Oder in der shell im Verzeichnis der VM selber /vmfs/volumes/VM Storage (share)/...../vmx

Wichtig hierbei sind zwei Dinge. Egal ob in der GUI oder in der shell -> es sollte penibel auf die Syntax geachtet werden. Ein space zu viel, ein Schreibfehler ist schnell gemacht, man sieht den Wald vor lauter Bäumen nicht - es funktioniert nicht, oder die VM startet nicht oder verhält sich komisch. Auch sollte klar sein, dass wie die Parameter eingetragen werden sich von der GUI zur shell und anders herum, unterscheiden. Dass obwohl sie beide in der vmx landen.

Es gibt hier nicht "den richtigen Weg". Je nach situation ist mal der Eine, oder der Andere Weg besser.

Beispiel mit selben Parameter...

über die WebUI:

```
pciPassthru0.opromEnabled TRUE
```

```
pciPassthru0.filename ../Powercolor.RX480.8192.160923.rom
```

über die shell in der .vmx:

```
pciPassthru0.opromEnabled = "TRUE"
```

```
pciPassthru0.filename = "../Powercolor.RX480.8192.160923.rom"
```

Ich hoffe das hilft jemanden weiter. Und immer schön den Hypervisor in den Wartungsmodus setzen bevor Neu gestartet oder heruntergefahren wird! 😊

Bilder zum HowTo hängen anbei.

Bottom Line:

Das mittlerweile 5 Jahre alte Hackintosh System, nun Hypervisor Host basiert auf folgenden:

- GA-Z170X-Gaming 3
- Intel Skylake i7 4GHz 6700K
- 32GB RAM
- PowerColor Red Devil RX480 8GB VRAM
- diverser SSD und HDD Storage

Zuletzt wurde noch für 9€ und für die ESXi 7 und 8 Kompatibilität ein PCIe I210-T1 GbE NIC gekauft. Der on board NIC wurde disabled.

Das damalige System wurde angelehnt an einen iMac 17,1 aufgebaut und funktionierte auf herkömmlicher Hackintosh Art bis Mojave wie ich es vor ESXi verwendet habe größtenteils out of the box.