

Erledigt

Hackintosh für Videobearbeitung mit externer SSD

Beitrag von „armut“ vom 10. Januar 2018, 22:52

Hallo Leute,

bitte korrigiert den Threat, fals es am falschen Ort geöffnet wurde.

Ich produziere seit einigen Jahren Videoclips mit FCPX. Anfangs besaß ich noch einen MacPro 2013, da Apple aber etwas träge mit dem Hardware-Update ist, stieg ich letztendlich zu Hackintosh um.

In meinem jetzigen Rechner, bearbeite ich Videos auf der SSD mit SATA Verbindung. Die Geschwindigkeit reicht allemale.

Mit dem letzten Sicherheitsupdate (Update-Risiko) kam mir der Gedanke, die Bearbeitung auf eine externe SSD auszulagern via USB 3.0.

Mein einziges Problem ist jedoch, dass externe SSDs über USB 3.0 kein TRIM unterstützt.

Ist TRIM wirklich sooo wichtig, dass ich diese Funktion unbedingt einschalten muss oder könnte ich TRIM auf einer externen SSD vernachlässigen?

Mich würde auch interessieren, ob es möglich ist (in Zukunft, wenn ich meine Hardware aktualisiere), SSDs via USB 3.1 Gen 2, sprich Thunderbolt im vollen Umfang ohne Probleme betreiben kann?

Vielen Dank im Voraus...:)

Beitrag von „griven“ vom 18. Januar 2018, 22:36

Trim ist bei aktuellen SSD Laufwerken nicht mehr so wichtig denn die Controller dieser

Laufwerke besitzen in der Regel eine eigene GarbageCollection die sich darum kümmert gelöschte Bereiche wieder als beschreibbar zu markieren von daher Du kannst es vermutlich vernachlässigen.

Zu Trim im allgemeinen es handelt sich dabei um ein Kommando aus dem SATA Befehlssatz welches von der IOAHCIFamily.kext verwaltet wird womit sich auch erschließt warum das Kommando bei Laufwerken die via USB angebunden sind nicht zur Verfügung steht 😄

Beitrag von „armut“ vom 19. Januar 2018, 08:34

Vielen Dank für die Antwort.

Ich habe mir bereits eine WD Blue 3D Nand SSD 512GB (m.2) vorerst gekauft und diese in einen externen m.2 RAID Gehäuse eingebaut, welches derzeit ohne RAID-Modi benutzt wird.

Die Geschwindigkeiten über USB 3.0 liegen beim Lesen/Schreiben etwa bei 420MB.

Später will das Ganze in Raid 0 (da USB 3.1 ext. Gehäuse) betreiben, um die maximale Geschwindigkeit ausschöpfen zu können.

Beitrag von „al6042“ vom 19. Januar 2018, 10:32

Die maximale Geschwindigkeit wird aber nicht vom USB3.1 kommen, sondern vom Lesen/Schreiben auf den RAID0 Verbund...

Dieser ist an der Stelle auch gleich mal "nur" SATA mit SATA 6 Gb/s und einer theoretischen Geschwindigkeit von sequenziell Lesen = bis zu 560 MB/s und sequenziell Schreiben = bis zu 530 MB/s ansprechbar.

Dagegen wagen wir mal einen Blick auf die USB-Geschwindigkeiten:

Vergleich: USB

Schnittstelle	USB 2.0	USB 3.0	USB 3.1	USB 3.2
Transferrate (theoretisch)	480 MBit/s	5 GBit/s	10 GBit/s	20 GBit/s
	bis 60 MByte/s	bis 600 MByte/s	bis 1.200 MByte/s	
Transferrate (praktisch)	ca 30 MByte/s	ca. 450 MByte/s	ca. 800 MByte/s	ca. 1.600 MByte/s

Aus meiner Sicht wird dir das Tempo von USB3.1 wahrscheinlich nicht viel bringen...

Vielleicht gibt es auch ein m.2 USB3.1 Gehäuse mit PCIe/NVMe Unterstützung, um dann noch mal etwas mehr Performance zu erhalten, aber mit deinem Konstrukt sehe ich keinen allzu großen Performance-Schub.

Beitrag von „Brumbaer“ vom 19. Januar 2018, 13:35

USB 3.0 ist von der Übertragungsrate ausreichend für eine SSD.

Ein Gerät, das einen Raid 0 Controller hat (im Gegensatz zu 2 USB Geräten, die als Raid 0 betrieben werden), benötigt USB 3.1 für volle Geschwindigkeit

Allerdings haben Geräte, die über USB angeschlossen werden, geringere IOPS als Geräte, die über SATA angeschlossen werden.

SSDs unterscheiden benutzte und unbenutzte Blöcke und unbenutzte können gelöscht oder nicht gelöscht sein.

Ein Block in einer SSD kann nicht überschrieben werden - ohne ihn vorher zu löschen. Das dauert relative lange, deshalb wird der Inhalt in einen unbenutzten und schon gelöschten Block geschrieben und der alte Block als unbenutzt und nicht gelöscht gekennzeichnet. Das funktioniert, weil es eine Tabelle gibt in der steht wo der Inhalt eines Blockes im Speicher untergebracht ist. Der Inhalt muss nicht immer an der selben Stelle stehen, solange man dafür sorgt, dass die Tabelle auf die richtige Stelle zeigt.

Wenn die SSD gerade nichts macht, wird die Garbage Collection gestartet und die löscht nun die unbenutzten, aber nicht gelöschten Blöcke, so dass sie für die nächste Schreiboperation zur Verfügung stehen.

Unbenutzt bedeutet hier unbenutzt aus Sicht der SSD. Ein Block der aus Benutzer/Betriebssystem Sicht gelöscht wird, ist aus Sicht der SSD immer noch benutzt, denn die SSD weiß nicht, dass das OS den Block als gelöscht ansieht.

Dafür gibt es das Trim Kommando. Das sagt der SSD, dass ein Block vom OS nicht mehr benötigt wird und er deshalb gelöscht und zu den anderen fürs Schreiben verfügbaren Blöcken hinzugefügt werden kann.

In der Praxis, wird der Block vermutlich nicht sofort gelöscht, sondern er wird nur als unbenutzt gekennzeichnet und bei der nächsten Garbage Collection gelöscht.

Trim ist ein SATA Kommando.

NVMe ist nicht SATA und kennt deshalb keinen Trim-Befehl, aber es kennt Deallocate. Deallocate markiert einen Block als unbenutzt und dann kann er bei der

Wir schauen uns mal 5 Speicherbereiche(SSD-Blöcke) in der SSD an.

Block 77 Daten für den OS-Block 48

Block 78 unbenutzt, nicht gelöscht

Block 105 Daten für den OS-Block 44

Block 111 Daten für den OS-Block 45

Block 115 unbenutzt, gelöscht

Nun schreibt das System Block 45 neu, es sucht sich einen unbenutzten gelöschten Block und verwendet den

Block 77 Daten für den OS-Block 48

Block 78 unbenutzt, nicht gelöscht

Block 105 Daten für den OS-Block 44

Block 111 unbenutzt, nicht gelöscht

Block 115 Daten für den OS-Block 45

Morgens um halb zehn in Deutschland, der Anwender(in) ißt sein K..., und die SSD hat Zeit die Garbage Collection laufen zu lassen

Block 77 Daten für den OS-Block 48

Block 78 unbenutzt, gelöscht

Block 105 Daten für den OS-Block 44

Block 111 unbenutzt, gelöscht

Block 115 Daten für den OS-Block 45

Jetzt löscht das System die Datei, die Block 44 enthält.

System ohne Trim/Deallocate

Block 77 Daten für den OS-Block 48

Block 78 unbenutzt, gelöscht

Block 105 Daten für den OS-Block 44

Block 111 unbenutzt, gelöscht

Block 115 Daten für den OS-Block 45

Wie sie sehen, sehen sie nichts, denn die SSD weiß nicht, dass der Block nicht mehr gebraucht wird.

System mit Trim/Deallocate

Block 77 Daten für den OS-Block 48

Block 78 unbenutzt, gelöscht

Block 105 unbenutzt, nicht gelöscht

Block 111 unbenutzt, gelöscht

Block 115 Daten für den OS-Block 45

Jetzt weiß es die SSD und gibt den Block frei

Garbage Collection und Trim/Deallocate haben unterschiedliche Aufgaben und ersetzen sich nicht gegenseitig.

NVMe benötigt keinen Trim Befehl, weil deallocate schon Teil des "normalen" NVMe Verhalten ist, während Trim "ein neues (relativ zu Ur-SATA) Feature" ist.

Beitrag von „Doctor Plagiat“ vom 19. Januar 2018, 15:14

Wie immer, wie in der "Sendung mit der Maus" klar, deutlich und gut erklärt. 👍

Beitrag von „armut“ vom 19. Januar 2018, 19:21

Vielen Dank für die ausführliche Erklärung der TRIM, Garbage Collection etc. Funktionen. Sehr informativ und hilfreich...:)

Das externe Gehäuse hat einen USB 3.1 Gen 2 Typ A Connector. Laut Datenblatt sind bei einer Verbindung mit diesem Port 10Gbit im RAID 0 Modus möglich, welches eine Geschwindigkeit von etwa 700-800MB ergeben müsste.