

Erledigt

14 Core Hackintosh auf El Capitan, Postinstall & Optimization Hilfe

Beitrag von „griven“ vom 5. November 2016, 02:12

Und das macht so auch Sinn.

Hyperthreading ist ja nichts anderes als die Möglichkeit auf einem CPU Kern einen 2. Thread laufen zu lassen wenn der erste Thread gerade pausiert also zum Beispiel auf eine EA Operation wartet etc. damit das möglichst effizient funktioniert muss die Software dafür optimiert sein was im Falle von FinalCut nur bedingt der Fall ist. Jetzt ist es natürlich ein erheblicher Unterschied ob man ein Benchmark laufen lässt das genau darauf optimiert ist und so die Leistung optimal ausreizt oder ob man Videos rendern möchte. Im Hyperthreading Umfeld kommt die erste Prio immer den physikalischen Kernen zu sind diese ausgelastet wird eben auch kein weiterer Thread geöffnet selbst wenn die CPU es könnte.

Cinebench berechnet ein einziges statisches Bild das Endergebnis ist ein einzelnes Bild sprich hier macht es Sinn die Berechnungsschritte auf alle verfügbaren Cores (auch die virtuellen) zu verteilen es spielt bei dem zu erzielenden Ergebnis keine Rolle in welcher Reihenfolge die Ergebnisse vorliegen sprich die Aufgabe lässt sich prima aufteilen und verteilt rechnen. Geekbench stellt theoretische Berechnungen an auch hier geht es nicht darum am Ende ein Ergebnis zu erhalten das die Leistung aller Kerne in einem definierten Ergebnis vereint sondern vielmehr misst Geekbench wie viele einzelne Berechnungen in einem definierten Zeitfenster mit einem definierten Algorithmus gleichzeitig möglich sind wobei das Ergebnis jeder Berechnung für sich steht. Software zum encodieren oder umcodieren von Videos profitiert davon aber nicht denn in einer linearen Abfolge (Video) lassen sich die Aufgaben nicht oder nur schlecht verteilen. Ein Video setzt sich aus vielen Einzelbildern zusammen die in einer definierten Abfolge aneinander gereiht werden müssen zudem muss obendrein noch der Ton berechnet bzw. umgerechnet werden.

Was also tun? Ich kann entweder die Berechnung eines einzelnen Frames auf alle verfügbaren Kerne verteilen sprich jeder Kern berechnet einen Bruchteil des einzelnen Frames und erst wenn alle Fertig sind ist der Frame berechnet und kann gespeichert werden (HT schließt sich hier aus denn das Ergebnis muss ja vorgehalten werden bis alle fertig sind) oder aber ich lasse jeden verfügbaren Kern je einen Frame aus einer Abfolge berechnen (bei 12 Kernen 12 Frames, bei 14 Kernen 14 Frames) aber auch hier muss die Reihenfolge eingehalten werden sprich auch hier schließt sich HT aus denn es gibt schlicht keine Möglichkeit einen sinnvollen 2 Task

zu eröffnen weil vollkommen unklar ist was als nächstes zu Berechnen ist bis nicht das Ergebnis aller gestarteten Berechnungen vorliegt.

Das alles ist jetzt natürlich sehr vereinfacht dargestellt denn in der Praxis spielt beim rendern von Bewegtbildern die GPU auch eine gewisse Rolle. FinalCut setzt bisher auf OpenGL bzw. OpenCL sprich lagert einen nicht unerheblichen Teil der nötigen Berechnungen auf die GPU aus sofern sich diese auf OpenCL versteht sprich dann spielt die CPU sogar eine untergeordnete Rolle weil sich die nötigen Berechnungen viel effizienter auf der GPU erledigen lassen. Premiere geht einen ähnlichen Weg setzt aber eher auf CUDA sprich auch hier werden, kompatible Karte vorausgesetzt und CUDA Treiber installiert nötige Berechnungen nicht auf der CPU durchgeführt sondern auf der für diese Zwecke wesentlich besser geeigneten GPU. Um das Ganze noch verwirrender zu gestalten bedienen sich moderne Videoschnitt Lösungen auch Hardwareseitig vorhandener Encoder bzw. Decoder so besitzen I5 oder I7 CPU's zum Beispiel eine eingebaute Beschleunigung für den H.264 Codec welcher den XEONS fehlt usw.