

ECS Dokumentation – Handl, Tributsch

Verwendete Hardware

Tributsch

CPU: AMD Ryzen 5 5600x

- Kerne: 6
- Threads: 12
- Basistakt: 3,7 GHz
- Turbotakt: 4,6 GHz
- L1 Cache: 6x 64KB
- L2 Cache: 6x 512KB
- L3 Cache: 32MB

RAM: G.Skill TridentZ RGB

- Technologie: DDR4 SDRAM
- Kapazität: 32 GB
- Module: 2
- Modulgröße: 16GB
- Geschwindigkeit: 3600 MHz
- Latenzzeiten: CL16 (16-19-19-39)
- Datenintegrität: Non-ECC

Handl

CPU: Intel Core i7-7700HQ

- Kerne: 4
- Threads: 8
- Basistakt: 2,8 GHz
- Turbotakt: 3,4 GHz
- L1I Cache: 4x32 KB
- L1D Cache: 4x32 KB
- L2 Cache: 4x256 KB
- L3 Cache: 4x1.5 MB

RAM:

- Technologie: DDR4 SDRAM
- Kapazität: 16 GB
- Module: 2
- Modulgröße: 8 GB
- Geschwindigkeit: 2400 MHz
- Datenintegrität: Non-ECC

Messungen

Mögliche numEntities vor dem Umbau auf ECS, um eine durchschnittliche Framerate von 33ms zu erreichen:

- Tributsch: numEntities = 170000
- Handl: numEntities = 52500

Gemessene durchschnittliche Framerate nach dem Umbau auf ECS (mit der gleichen Anzahl an Entities):

- Ø Framerate: 37ms (bei beiden Testgeräten)

Interpretation der Ergebnisse

Es konnte also kein eindeutiger Performancegewinn, sondern eher ein leichter Performanceverlust festgestellt werden. Ein großer Vorteil vom ECS Konzept ist die „cache-friendliness“, da die Daten hier dicht im Speicher nebeneinander liegen, sodass sie in einer gesamten Cache-Zeile auf einmal geladen werden können. Wenn das nächste Element durchiteriert wird, sind die Daten bereits vorhanden im Cache, was zu einer Performanceverbesserung führen sollte. Allerdings macht sich dies vor allem erst bei größeren und vor allem komplexeren Spielen bemerkbar, wo tausende komplexere Entities gleichzeitig am Screen sind. Daher ist bei diesem eher simpel gehaltenen Beispiel kein eindeutiger Performancegewinn erkennbar.