

I Delimit from each other the following SSD parts: Cells, Pages and Blocks.

Cells: -Speichern einzelne Bits (Mehrere Arten von 1-4 Bits)

-Zustand wird durch Elektronenschwellenwert bestimmt

Pages:-Page = Gruppe aus Cells

-kleinste Lese-/Speichereinheit, die read und write unterstützt. D.h. wenn wir eine Zelle lesen wollten, dann müssen wir die gesamte zugehörige page laden

-keine inplace updates also: jedes write erstellt eine neue page, auch wenn wir nur 1 Bit ändern wollen

Blocks: -Ein Block ist die kleinste Löscheinheit, bestehend aus mehreren Pages

➔ Wenn wir eine Page Löschen wollen, müssen wir den ganzen Block zu dem sie gehört löschen (damit eingehend copy paste von validen Pages auf neuen Block; Stichwort garbage collection)

I What is the purpose of garbage collection in SSDs?

Garbage collection bestimmt, wann eine Löschoperation ausgeführt wird. Dies passiert, wenn die Anzahl an invaliden pages innerhalb eines Blockes einen Schwellwert überschreitet. Dann werden die noch validen pages in einen neuen Block kopiert und der alte Block wird gelöscht.

Ein weiterer Nutzen eines garbage collection Algorithmus' ist auch, die write amplification so niedrig zu halten wie möglich.

Garbage collection bewegt Datensegmente, die zusammengehörig sind, näher aneinander (falls möglich).

I What is the purpose of wear leveling in SSDs?

Wear leveling versucht die Abnutzung der pages/ Blöcke auf die gesamte SSD zu verteilen, damit nicht einige Stellen signifikant schneller kaputt gehen als andere.

Dies geschieht, indem die Schreiboperationen über die ganze SSD verteilt werden.

I Tell some interesting things about SSDs with an M.2 form factor.

- M.2 Formfaktor gibt es für PCIe und SATA
 - o D.h. auch keine Garantie für NVMe Protokoll, könnte auch AHCI sein
- Es gibt auch verschiedene „Key“ Größen, also Anschlüsse, mit unterschiedlicher Pin Anzahl
-

I What influence do garbage collection and wear leveling have on write amplification of an SSD?

Wear leveling erhöht write amplification, da selten genutzte Daten auf oft genutzte pages geschrieben werden können, um die abnutzung der SSD zu verringern, dies benötigt jedoch mehr writes.

Garbage collection erhöht write amplification, da wir mit jeder Löschoption den validen Rest des Blockes auf einen neuen Block schreiben. Deswegen sind effiziente garbage collection Algorithmen wichtig.

I Discuss three different recommendations for writing code for SSDs.

Avoid scattered updates → So wenige Dateien wie möglich benutzen, um unsere Daten zu speichern (z.B. 1 log-file). Dadurch erreichen wir schnellere performance, nutzen die SSD weniger ab, und brauchen weniger Operationen auf der SSD, was bedeutet dass der Controller der SSD weniger ausgelastet ist.

Avoid full SSDs → Die SSD wird signifikant langsamer, wenn sie fast voll ist, da der garbage collector mehr Blöcke verschieben muss, um freien Speicher zu schaffen. Lösung dazu könnte manuelle Partitionierung sein, damit garantiert leerer Speicher vorhanden ist.

Use multiple threads to do small IO → ein einziger IO Thread kann nicht die gesamten Parallelisierungsmöglichkeiten der SSD ausnutzen. Um die SSD optimal zu nutzen (wenn wir viele kleinen Dateien lesen wollen), sollten wir daher mehrere Threads nutzen.

Use Few Threads to do big IO → Wenn wir sowieso schon große Daten lesen, dann kann die SSD selbst optimal arbeiten. Wir müssen also nicht nach helfen.

I How could the CPU load for IO be reduced?

Lieber asynchrone IO calls benutzen, als blocking.

Heißt, dass bei einer read Operation nicht gewartet wird, bis der Lesevorgang vorbei ist (blocking), sondern etwas anderes gemacht wird und man erst wieder weiterarbeitet, wenn die Daten bereit sind.

Es wird auch empfohlen, das eigene OS buffering auszuschalten, damit die Applikation eigenes, effizienteres buffering definieren kann.

I How could you solve problems that do not fit in DRAM without major code adjustments?

SSD als DRAM-Ersatz verwenden.

Geschicktes Pipelining, um Latenzen zu verstecken.

Im Code durch: memory mapped files, die dann vom Programm als Arrays behandelt werden.