**WAS IST EIGENTLICH DIE BLOCKGRÖßE?**

Eine wichtige Kenngröße in einem RAID-Verbund ist die *block size* / *Blockgröße*, die manchmal auch als *stripe size* bezeichnet wird. Obwohl sie selten explizit auftaucht, ist sie ein wichtiger Parameter für die Performance, die der RAID-Verbund erreichen kann. Die Blockgröße beschreibt eine zusammenhängende Größe an Daten, die jeweils *gemeinsam* verarbeitet werden. Bei RAID-Systemen wird meistens eine Blockgröße im Bereich von 64 kiByte bis 512 kiByte verwendet, je nach Hersteller. Die Größe kann manchmal auch vor dem Erstellen des RAID-Verbundes konfiguriert werden.

**Was bedeutet das nun konkret?** In folgendem Beispiel soll eine Datei *Bericht.docx* in einem RAID0Verbund mit zwei Datenträgern abgelegt werden. Die ist 200 KiByte groß, und die Blockgröße beträgt 64 kiByte. Anhand der Blockgröße wird nun die Aufteilung der Daten bestimmt und die Datei in die vier Blöcke A, B, C und D geteilt, die jeweils eine Größe von 64 kiByte haben:

Bericht.docx

Bericht.docx

50

kiByte

100

kiByte

150

kiByte

200

kiByte

0

kiByte

*Blockgröße 64 kiByte*

A

B

D

C

*Blockgröße 64 kiByte*

*Blockgröße 64 kiByte*

*Aufteilen*

*in Blöcke*

*Verteilen auf*

*Datenträger*

Auf den beiden Datenträgern werden die Daten dann folgendermaßen abgelegt:

D

C

A

B

**RAID 0**

*Blockgröße*

*64*

*kiByte*

*Blockgröße*

*64*

*kiByte*

*Blockgröße*

*64*

*kiByte*

Es werden also immer Abschnitte in der Größe von 64 kiByte gemeinsam als Block behandelt und zwischen den Datenträgern verteilt. Diese Daten werden auch immer gemeinsam geschrieben und gelesen. Dabei können sich auch mehrere Dateien in einem Block befinden, und umgekehrt kann auch eine kleinere Datei zufällig in zwei Blöcken landen.

In unserem Beispiel wurden in Block D erst 8 kiByte verwendet, es sind also 56 kiByte frei. Wird nun eine weitere Datei in der Größe von 60 kiByte gespeichert, landen die ersten 56 kiByte ebenfalls in Block D, und die verbleibenden 4 kiByte in einem neuen Block E.

Die Blockgröße beeinflusst also, ab welcher Datenmenge die Daten auf mehrere Festplatten gespeichert werden.

# RAID-SIMULATION

In dieser Übung werden Sie die verschiedenen RAID-Modi simulieren, in dem Sie sich anschauen, wie genau die Daten auf den einzelnen Datenträgern verteilt werden und wie die Daten bei einer defekten Festplatte wieder gelesen werden können.

**AUFGABE 1 – RAID-VERBÜNDE OHNE PARITÄT**

In Ihrem Analyse-Labor finden Sie verschiedene RAID-Verbünde vor, die analysiert werden müssen. Die Blockgröße beträgt in unserer Simulation immer zwei Bytes. Dabei entspricht ein Buchstabe immer genau einem Byte, das Symbol ␣ steht für ein Leerzeichen.

1. **Bestimmen** **Sie** jeweils den RAID-Modus, der verwendet wurden
2. **Ergänzen Sie** die weitere Verteilung der Daten auf die Datenträger.
3. Einige Datenträger sind leider defekt, sie sind mit markiert. **Geben** Sie an, wie die Daten ausgelesen aussehen, **setzen** Sie für verlorene Daten das Symbol ⍰ **ein.**

Inhalt a): [Ha|ll|o␣|We|lt|!␣|Ic|h␣|bi|n␣|ei|ne|␣D|at|ei|!]

RAID: 0

Datenträger 1 Datenträger 2 Datenträger 3 Datenträger 4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Block 1 | | Ha | ll | o␣ | We |
| Block 2 | | lt | !␣ | Ic | h␣ |
| Block 3 | | bi | n␣ | ei | ne |
| Block 4 | | |␣D | at | |ei | ! |
| Inhalt b): | | [Ei|ne|␣w|ic|ht|ig|e␣|In|fo|rm|at|io|n.] | | | | | |

RAID: 5

1 2 3 4 5 6

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Block 1 | | Ei | | Ei | ne | ne | ␣w | ␣w |
| Block 2 | | ic | | ic | ht | ht | ig | ig |
| Block 3 | | e␣ | | e␣ | In | In | fo | fo |
| Block 4 | | rm | | rm | at | at | io | io |
| Block 5 | | n. | | n. |  |  |  |  |
| Inhalt c): | |  | | [Hi|er|␣s|te|ht|␣w|as.] | | | | | | |

RAID: 1

Datenträger 1 Datenträger 2 Datenträger 3 Datenträger 4

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Block 1 | | Hi | | | er | Hi | er |
| Block 2 | | ␣s | | | te | ␣s | te |
| Block 3 | | ht | | | ␣w | ht | ␣w |
| Block 4 | | as | | | . | as | . |
| Inhalt d): | |  | |  | [Gu|te|n␣|Ta|g.] | | | | |

RAID:

Datenträger 1 Datenträger 2 Datenträger 3 Datenträger 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Block 1 | Gu | Gu | Gu | Gu |
| Block 2 | te | te | te | te |
| Block 3 | n␣ | n␣ | n␣ | n␣ |
| Block 4 | Ta | Ta | Ta | Ta |
| Block 5 | g. | g. | g. | g. |

Inhalt e): [Ha|ll|o␣|We|lt|!␣|Ic|h␣|bi|n␣|ei|ne|␣D|at|ei|!]

RAID:

Datenträger 1 Datenträger 2 Datenträger 3 Datenträger 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Block 1 | Ha | !␣ | ei | ! |
| Block 2 | ll | lt | n␣ | ei |
| Block 3 | o␣ | We| | bi | at |
| Block 4 |  | Ic | h␣| |  |
| Block 5 |  |  |  |  |

**AUFGABE 2 – RAID-VERBÜNDE MIT PARITÄT**

Für die Berechnung der Prüfsumme müssen Sie die ASCII-Zeichen in Binärzahlen umrechnen. Die Daten sind im ISO-8859-1-Format kodiert worden, eine entsprechende ASCII-Tabelle finden Sie in Ihrem ITHandbuch oder im Anhang zu dieser Aufgabe.

Bei **RAID 5** werden für die Paritätsdaten jeweils die anderen n – 1 Datenblöcke mit der XOR-Funktion verknüpft. Um die Parität zu berechnen, wandeln Sie also zuerst die ASCII-Zeichen entsprechend der ASCII-Tabelle in Binärschreibweise um und verknüpfen die Binärzahlen dann bitweise mit der XORFunktion. Um einen fehlenden Datenblock zu berechnen, verknüpfen Sie die verbliebenen Datenblöcke und die Parität ebenfalls mit der XOR-Funktion – dann ergibt sich der fehlende Datenblock.

Bei **RAID 6** werden die Paritätsdaten p genauso wie bei RAID 5 berechnet. Die Paritätsdaten q werden mit einem Reed-Solomon-Code berechnet. Reed-Solomon-Codes werden zum Beispiel auch bei CD’s und DVD’s zur Fehlerkorrektur verwendet, z.B. um einfache Kratzer „auszugleichen“. Die Berechnung ist mathematisch aufwändiger, und zum Beispiel im Tutorial „*A Tutorial on Reed-Solomon Coding for Fault-Tolerance in RAID-like Systems*“ [[http://web.eecs.utk.edu/~jplank/plank/papers/CS-96-332.pdf]](http://web.eecs.utk.edu/~jplank/plank/papers/CS-96-332.pdf) beschrieben.

1. **Bestimmen** **Sie** jeweils den RAID-Modus, der verwendet wurde.
2. **Ergänzen Sie** die weitere Verteilung der Daten auf die Datenträger und **berechnen Sie** die fehlenden Paritäten.

|  |  |  |
| --- | --- | --- |
| Inhalt 2a): |  | [Hi|er|␣s|te|ht|␣e|tw|as|.] |

RAID: 5

Datenträger 1 Datenträger 2 Datenträger 3 Datenträger 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Block 1 | Hi | er | ␣s | [Parität] |
| Hex: | 0x4869 | 0x6572 | 0x2073 | 0x0D68 |
| Binär | 0100 1000  0110 1001 | 0110 0101  0111 0010 | 0010 0000  0111 0011 | 0000 1101  0110 0000 |
| Block 2 Hex:  Binär | ll  0x6C6C  01101100  01101100 | 0x0020  00100000  00000000 | [Parität]  0x5376  01010011  01110110 | u  0x0D75  00001101  01110101 |
| Block 3 Hex:  Binär |  | [Parität] |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Inhalt 2b): |  |  | [Ha|ll|o␣|Du] |

RAID: 6

Datenträger 1 Datenträger 2 Datenträger 3 Datenträger 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Block 1 | Ha | ll | [Parität p] | [Parität q] |
| Hex: | 0x4861 | 0x6C6C | 0x240D | [Reed-Solomon] |
| Binär | 0100 1000  0110 0001 | 0110 1100  0110 1100 | 0010 0100  0000 0000 | [Reed-Solomon] |
| Block 2 | Du | [Parität p] | [Parität q] |  |
| Hex: | 0x6F20 |  | [Reed-Solomon] | 0x4475 |
| Binär | 0110 1111  0010 0000 | 01010011  01110110 | [Reed-Solomon] | 0100 0100  0111 0101 |

1. Einige Datenträger sind leider defekt, sie sind mit markiert. **Stellen** Sie die Daten (wenn möglich) **wieder her**.

**Setzen** Sie für verlorene Daten das Symbol ⍰ **ein.**

|  |  |  |
| --- | --- | --- |
| Inhalt 2c): |  | [ | | | | | ] |

RAID: 6

Datenträger 1 Datenträger 2 Datenträger 3 Datenträger 4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Block 1 |  |  |  | [Parität] |
| Hex:  Binär | 0x20202020  0101 0011  0111 0101 | 0x20202020  0110 0101  0110 1001 | 0x20202020  0111 0010  0110 0111 | 0101 0001  0111 0000 |
| Block 2 |  |  | [Parität] |  |
| Hex:  Binär | 0x20202020  0110 0101  0110 1001 | 0x20202020  0110 0101  0110 1001 | 0110 1101  0010 0000 | 0x20202020  0110 1101  00100001 |

# AUSZUG AUS DER ASCII-TABELLE / ISO-8859-1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **Dez** | **Hex** |  | | … | … | … | | 32 | 20 | ␣ | | 33 | 21 | ! | | … | … | … | | 46 | 2E | . | | … | … | … | | 48 | 30 | 0 | | 49 | 31 | 1 | | 50 | 32 | 2 | | 51 | 33 | 3 | | 52 | 34 | 4 | | 53 | 35 | 5 | | 54 | 36 | 6 | | 55 | 37 | 7 | | 56 | 38 | 8 | | 57 | 39 | 9 | | 58 | 3A | : | | … | … | … | | 63 | 3F | ? | | … | … | … | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **Dez** | **Hex** |  |  | **Dez** | **Hex** |  | | … | … | … | … | … | … | | 65 | 41 | A | 97 | 61 | a | | 66 | 42 | B | 98 | 62 | b | | 67 | 43 | C | 99 | 63 | c | | 68 | 44 | D | 100 | 64 | d | | 69 | 45 | E | 101 | 65 | e | | 70 | 46 | F | 102 | 66 | f | | 71 | 47 | G | 103 | 67 | g | | 72 | 48 | H | 104 | 68 | h | | 73 | 49 | I | 105 | 69 | i | | 74 | 4A | J | 106 | 6A | j | | 75 | 4B | K | 107 | 6B | k | | 76 | 4C | L | 108 | 6C | l | | 77 | 4D | M | 109 | 6D | m | | 78 | 4E | N | 110 | 6E | n | | 79 | 4F | O | 111 | 6F | o | | 80 | 50 | P | 112 | 70 | p | | 81 | 51 | Q | 113 | 71 | q | | 82 | 52 | R | 114 | 72 | r | | 83 | 53 | S | 115 | 73 | s | | 84 | 54 | T | 116 | 74 | t | | 85 | 55 | U | 117 | 75 | u | | 86 | 56 | V | 118 | 76 | v | | 87 | 57 | W | 119 | 77 | w | | 88 | 58 | X | 120 | 78 | x | | 89 | 59 | Y | 121 | 79 | y | | 90 | 5A | Z | 122 | 7A | z | | |  |  |  | | --- | --- | --- | | **Dez** | **Hex** |  | | … | … | … | | 196 | C4 | Ä | | … | … | … | | 214 | D6 | Ö | | … | … | … | | 220 | DC | Ü | | … | … | … | | 228 | E4 | ä | | … | … | … | | 246 | F6 | ö | | … | … | … | | 252 | FC | ü | | … | … | … | |