

# RAIDs – Redundant arrays of independent disks

More than just a bunch of disks

written by Hauke Stieler,  
supported by Konstantinos Chasapis

Issued on the 13<sup>th</sup> September, 2015



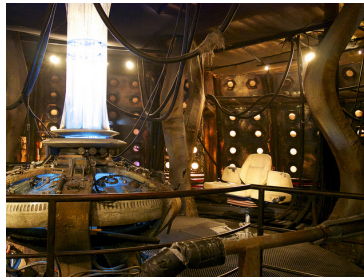
---

<sup>0</sup>UHH-Logo: [Sch]

## Abstract

This report is the result of the topic *RAID systems* in the Proseminar *Speicher- und Dateisysteme* (storage and file systems) of the Department of Informatics at the University of Hamburg during the summer semester 2015.

In this paper, you will find an overview of the RAID technology and a list of the most used RAID-levels including combinations of different levels. I will also describe how to set up a RAID in a software or hardware based environment.



”A hard drive is like a TARDIS. I like the bit when someone says it’s bigger on the inside.”  
– *unknown time traveler*

Picture by [jen].

Last changes made on the 13<sup>th</sup> September, 2015.

For further contact: [mail@hauke-stieler.de](mailto:mail@hauke-stieler.de)

# Contents

<b>1</b>	<b>Basics things about RAID</b>	<b>4</b>
1.1	Requirements . . . . .	4
1.2	Concepts of RAIDs . . . . .	4
1.2.1	Striping . . . . .	4
1.2.2	Mirroring . . . . .	4
1.2.3	Parity . . . . .	4
1.2.4	Hot-Spare . . . . .	5
<b>2</b>	<b>RAID Level</b>	<b>5</b>
2.1	Important properties . . . . .	5
2.2	Level 0 . . . . .	6
2.3	Level 1 . . . . .	6
2.4	Level 2 . . . . .	6
2.5	Level 3 . . . . .	7
2.6	Level 4 . . . . .	7
2.7	Level 5 . . . . .	7
2.7.1	The XOR operation . . . . .	8
2.7.2	Example: Calculate parity . . . . .	8
2.7.3	Data recovery . . . . .	8
2.8	Level 6 . . . . .	9
2.9	Level 10 & 01 . . . . .	9
2.10	Other level . . . . .	9
<b>3</b>	<b>Hard- and Software RAIDs</b>	<b>10</b>
3.1	Hardware RAIDs . . . . .	10
3.1.1	Setting up a hardware RAID . . . . .	10
3.2	Software RAIDs . . . . .	10
3.2.1	Linux (using mdadm) . . . . .	11
3.2.2	Windows . . . . .	13
3.3	Comparison . . . . .	13
<b>4</b>	<b>Usage</b>	<b>14</b>
4.1	Private . . . . .	14
4.2	Commercial . . . . .	14
<b>5</b>	<b>Appendix</b>	<b>15</b>
	List of Tables . . . . .	15
	List of Figures . . . . .	15
	References . . . . .	15

## 1 Basics things about RAID

The term *RAID* means *Redundant Array of Independent Disks* and summarizes concepts of connecting hard drives<sup>1</sup>.

A working RAID system builds a *logical unit* which means that the user does not see every hard drive as one drive, but all the connected drives. This logical unit provides more functions and features than one single hard drive and is therefore used by professional users (but more about this later in the section about the Usage).

The two main features that can be achieved with system are the prevention of data loss (increased data security) and a higher performance. Depending on the type of RAID (s. RAID Level), the system can achieve only one, both or even more functions.

In general a RAID provides *reliability* (be sure that no data gets lost), *availability* (always have access to the data even if one or more drive(s) fail(s)) and *capacity* (have all the features and get as much capacity as possible).

### 1.1 Requirements

As mentioned before, a RAID consist of many connected hard drives (at least 2, depending on the RAID level) and of course some cables and other hardware parts. In any case, you either need a special RAID controller or a special software to manage all these extra functions. Both ways are described later in section about Hard- and Software RAIDs.

Furthermore there must be - of course - enough ports on the main board or storage controller, power cables and power of the power supply. Possibly an extra card is necessary (for example a PCIe RAID controller, but more on this in the section about Hardware RAIDs).

### 1.2 Concepts of RAIDs

Next to the reliability, availability and capacity idea of a RAID, we will now discuss some technical aspects and concepts of RAIDs. These concepts are also part of a RAID Level so there might be some redundancy here.

#### 1.2.1 Striping

*Striping* means actually "cutting data into pieces". These pieces, also called *stripes* or *chunks*, are stored on different drives which allows the system to read and write faster, because the read and write operations can be parallelized.

The size of the chunks can be chosen at the set up. A very small size (like 16kb or less) causes many movements of the read-write-head of the disks and has therefore a negative impact on the performance. Very large chunks (like 512kb and more), can cause longer reading/writing times which does effect the performance in a negative way too.

#### 1.2.2 Mirroring

*Mirroring* is the opposite concept to striping, it saves the same data on all of the chosen hard drives. This creates redundancy and therefore security, meaning the data does not get lost so easy even if one or more drives fail (depending on the RAID Level).

#### 1.2.3 Parity

The concept of *parity* says, that there will be some extra information stored on the hard drives. Instead of a copy, the parity data is calculated via some special Algorithms (described in more details later) and

---

<sup>1</sup>WikiRAID

is not visible for the operating system and the user.

Storing these parity data creates also a kind of redundancy because it enables the restoration of lost drives/data by using these parity information.

However, there is no specific rule that says how to create or how to save these data. Some levels are storing the data chunks on separated disks, some levels put them on the same disks as the normal data and all ways have their own (dis)advantages.

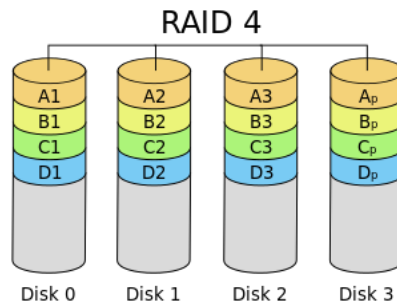


Figure 1: Disk 3 of this Level 4 RAID contains the parity data.[WCf]

### 1.2.4 Hot-Spare

An empty disk which is also included in the RAID is a so called *hot spare*. This stand-by will only be used when a normal disk fails and is also called a replacement-disk. An alternative to this disk is to replace the broken disk as fast as possible, which is not always possible.

## 2 RAID Level

This section deals with the main RAID *levels* and combinations that are pretty common. A RAID level is nothing less than the *type* of RAID and only says what this RAID can (not) do.

There are six standard levels (0, 1, . . . , 6) which are all different and have their own (dis)advantages. It is also possible to combine them and create new level with even more/other functions, but its diversity is limited by the possibilities of the controller.

Every RAID is visible as one logical unit for users and applications. Normal applications use the different drives as one disk to write on and read from them. Nevertheless it is possible to get information like temperature, amount of starts, power on hours count, etc. of every single disk.

For every RAID a specific amount of disks is necessary (at least 2) but there is no maximum limit to the amount of disks – in theory.

### 2.1 Important properties

Some properties will be discussed in the following sections, so here are some information about the terminology.

- The **space efficiency** is a relative value that indicates the amount of available space. A 1 would mean that all the space (100%) is available while a 0.5 means, that only half of the whole space (50%) is available.
- The **failure rate** is the chance that the *whole* RAID system fails and the data is not accessible any longer. A 0 means that the RAID will *never* fail (0% chance) and a high value indicates a more unsecure RAID.

- The **fault tolerance** is the amount of disks which can fail until data gets lost.

## 2.2 Level 0

The RAID 0 is a striping-only level which means that the data will be striped and then saved on different drives <sup>2</sup>. To achieve the best performance, the stripe size has – as mentioned earlier in section about Striping – to be about 128kb to 512kb.

A RAID 0 needs a minimum amount of 2 disks because there must be at least one other disk to store a part of the data on it. Because of non existing parity or redundancy, the relative space efficiency is 1 because RAID 0 can use 100% of its available space (as long as the drives have the same capacity). That is why the fault tolerance is 0 because there is no redundancy or parity. The failure rate increases with the number of disks and can be describes by the formula  $1 - (1 - r)^n$  ( $n$  =amount of disks,  $r$  =average failure rate of one disk).

## 2.3 Level 1

A RAID 1 is the opposite to the RAID 0, it is used to create pure *redundancy* for maximum security <sup>3</sup>. That security is ensured by saving the same data on every disk, wherefore some people also call this a "life-backup" because it is a backup which is always up to date.

Normally the performance is pretty much the same like the one of a single hard drive, but on some controller it is possible to read different parts from different disks (in figure 3 for instance A1 from disk 0 and A2 from disk 1). This allows pretty much the same reading speed as with a RAID 0.

The RAID 1 needs at least 2 disks to create one copy on each drive which results a space efficiency of  $1/n$ , so the available space is always equal to the size of one single disk. While the space efficiency decreases with the amount of disks, the failure rate with  $r^n$  increases. This is the key advantage of RAID 0 – its redundancy. One other remarkable point is the fault tolerance with  $n - 1$ , so every disk can fail but one, which is the highest of all RAID levels.

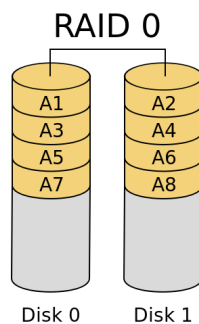


Figure 2: A RAID 0 system stripes the data and creates no redundancy.[WCb]

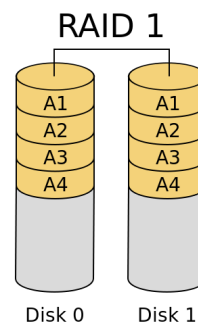


Figure 3: RAID 1 systems saves the same data on each drive.[WCd]

## 2.4 Level 2

The RAID 2 level is a striping level as well but it works on bit-level which means that each bit is evenly distributed <sup>4</sup>. Another key property is the error correction via Hamming-code which allows the controller to correct possible errors on-the-fly. The parity data of the Hamming-code is stored on separate drives.

<sup>2</sup>[WCa]

<sup>3</sup>[Leu]

<sup>4</sup>[Wik]

Nevertheless this level is not used any more because of the slightly lower space efficiency (compared to all other levels) which makes this level more expensive than other ones.

## 2.5 Level 3

RAID 3 is also a striping RAID but this time on byte-level<sup>5</sup>. It also has parity data on an extra disk (like Raid 2) but an on-the-fly correction is not natively provided.

Some people use RAID 3 to have a high performance when working with large files (like uncompressed videos or backup archives)<sup>6</sup>.

However there is no provable advantage over RAID 5 which has replaced the RAID 3 level and therefore RAID 3 is not/rarely used<sup>6</sup>.

## 2.6 Level 4

RAID 4 also stripes its data, but on block level (not on bit-/byte-level).

Like RAID 3, RAID 4 has also a separate parity disk, which is a huge disadvantage of RAID 4. When a system has lots of read and write cycles, the parity disk acts like a bottleneck, because the writing speed of the whole RAID is limited to the writing speed of this parity disk.

## 2.7 Level 5

RAID 5 is the most used RAID system. It has a high performance due to striped data and a high security because of parity information<sup>7</sup>.

The parity data is calculated via the logical XOR operation from the CPU or the RAID-Controller. Using XOR does slightly decrease the performance, but it is rather negligible.

To create a RAID 5, 3 or more disks are needed, whereas the space of one disk is used to store the parity data and all other disks storing the normal data. That is why the space efficiency is  $1 - 1/n$  (one parity disk minus the space efficiency of a RAID 1).

The failure rate is by far the most complicated formula of all and is describes by:

$$1 - (1 - r)^n - n \cdot r \cdot (1 - r)^{n-1}$$

The fault tolerance, however, is simply 1. One disk can fail without any data loss as we will see in the example when we calculate the parity data.

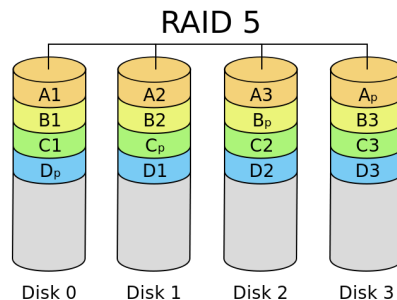


Figure 4: Party data is spread over all disks.[WCg]

<sup>5</sup>[Kum08, p.4, 3.3.]

<sup>6</sup>[WCh]

<sup>7</sup>[Kum08, p.5, 3.5]

### 2.7.1 The XOR operation

The XOR returns `true` (or 1) when *exactly* one input bit is `true` (or 1). Here is a small example of the functionality:

$$\begin{aligned}\text{XOR}(1,0) &= 1 = \text{XOR}(0,1) \\ \text{XOR}(1,1) &= 0 = \text{XOR}(0,0)\end{aligned}$$

The amount of input bits is variable because just one bit has to be 1 and it does not matter which one. Furthermore all the input bits can be treated in pairs:

$$\begin{aligned}\text{XOR}(0,1,1) &= \text{XOR}(\text{XOR}(0,1),1) \\ &= \text{XOR}(1,1) \\ &= 0\end{aligned}$$

### 2.7.2 Example: Calculate parity

Now we want to calculate some parity bits. It is not very hard, but need lot of writing space so this example will only calculate the parity for three bits for the RAID 5 with three disks.

The controller has now to calculate the parity  $A_p$  of the data  $A_1, A_2, A_3$ , which will be stored on every disk. Here are the original bits as vectors and every vector is stored on one disk:

$$A_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}, A_3 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}$$

Now the parity bit for each row of  $A_1, A_2, A_3$  can be calculated:

$$\left. \begin{aligned}\text{XOR}(1,0,1) &= 0 \\ \text{XOR}(0,1,0) &= 1 \\ \text{XOR}(0,1,1) &= 0\end{aligned} \right\} \Rightarrow A_p = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}$$

These bits must now be distributed over the three disks as you can see in figure 4.

### 2.7.3 Data recovery

The parity data of a RAID 5 is made to restore the content of one lost hard drive. So lets say disk two, with its data  $A_2$ , fails and has been replaced. Using the parity data, the lost data  $A_2$  can be restored by using the XOR operation again.

Here are the data blocks as vectors:

$$A_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, A_3 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, A_p = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, A_2 = ?$$

To calculate  $A_2$ , just XOR the rows of the given vectors and save the result on disk 2:

$$\left. \begin{aligned}\text{XOR}(1,1,0) &= 0 \\ \text{XOR}(0,0,1) &= 1 \\ \text{XOR}(0,1,0) &= 1\end{aligned} \right\} \Rightarrow A_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$



## 2.8 Level 6

The disadvantage of a RAID 5 is its relatively low fault tolerance. To solve this security "issue", the RAID 6 was invented and provides a higher fault tolerance of  $2^8$ .

To achieve this, a second parity block has to be calculated via gallois-fields or other methods like a Reed-Solomon-Code. Unfortunately this causes the disadvantage of RAID 6: The writing speed is a bit slower compared to the one of a RAID 5.

## 2.9 Level 10 & 01

A Level 10 (or 1+0) RAID and a 01 (or 0+1) RAID are pretty much the same. Both are combining the advantages of a RAID 0 (performance) and a RAID 1 (security)<sup>9</sup>.

The main difference is the fault tolerance of 10 and 01 which is less in the RAID 01. This is caused by the structure of RAID 01 (it is a mirrored RAID 0). For example: The chance of a total failure in a RAID 10 with 8 disks is  $1/7$  but in a RAID 01 it is  $4/7$ <sup>10</sup>.

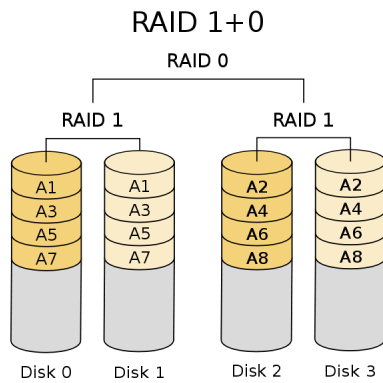


Figure 5: RAID 10: Striped mirrors.[WCe]

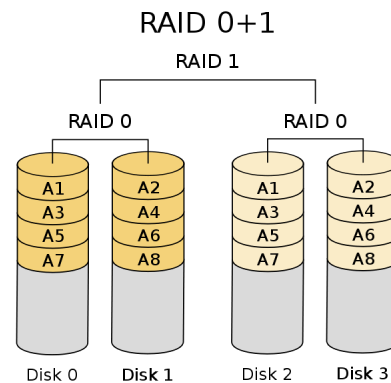


Figure 6: RAID 01: Mirrored stripes.[WCc]

The same effect can be described by the access limitations of a RAID: Imagine two disks of the RAID 01 (one in every RAID 0 component) fails, then there is the chance that e.g. Disk 0 and Disk 3 fails. This will cause the crash of the whole RAID 01. Unfortunately the controller can not copy data from disk 2 to the new disk 0 because it is in another RAID 0 section and normally it is not possible to read data from different RAID 0 components while the RAID is running. In a RAID 10 this scenario is not possible, that is why a RAID 01 is a bit less secure<sup>10</sup>.

Nevertheless is the performance the same and both RAID needs at least four disks (a RAID 01 is also possible with three disks, but most controllers implements it with a minimum amount of four disks<sup>11</sup>).

## 2.10 Other level

As you may already guess from section about Level 10 & 01, it is possible to combine several levels<sup>12</sup>. Some (more or less useful) RAID combinations are all 0x (striped levels) and 1x (mirrored levels) RAIDs and combinations with a RAID 5 (e.g. 65, 51, 52, 53, ...).

There are many other levels like 1E, 5E/5EE and combinations with hot-spare drives and so on. Most of them were invented many years ago and have never really been used.

<sup>8</sup>[Kum08, p.5, 3.6]

<sup>9</sup>[Leu]

<sup>10</sup>[JH]

<sup>11</sup>[Comc]

<sup>12</sup>[Wik]

### 3 Hard- and Software RAIDs

To set up a RAID, there are two different ways in doing this: Using either a *hardware* or a *software* based system. Both have advantages and disadvantages so here is a quick overview.

#### 3.1 Hardware RAIDs

To use a hardware RAID all disks have to be connected to a special controller, the RAID-controller. Most motherboards have such a controller out-of-the-box which can handle the basic levels like 0, 1, 10 and mostly 5 as well. Some more expensive boards may be able to manage other levels.

If the performance is too low or the functional range too small, there are dedicated controller for an PCI(e) port of the motherboard. Often these controllers have special hardware components for a faster parity calculation<sup>13</sup> which is good for systems with a Level 5 or Level 6 RAID.

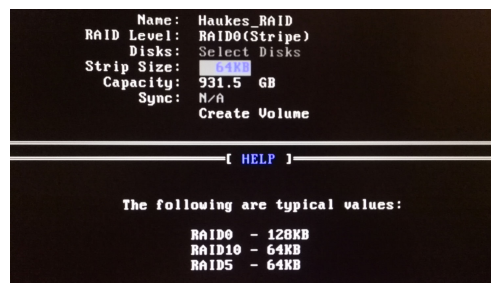


Figure 7: Most mainboards have an inbuilt RAID controller.[WCe]

The advantage of a hardware RAID is its high performance even if there are many disks in use. The disadvantage is its price, because those special hardware components are very expensive<sup>13</sup>.

##### 3.1.1 Setting up a hardware RAID

To create a hardware based RAID on a Computer, you have to enable the RAID-option in the BIOS/UEFI menu (normally the AHCI-mode is selected). After saving the settings and restarting the computer, a special function key (e.g. F10) will lead you into a set up menu (s. figure 7) where you can create, edit or delete a RAID.

While creating a new RAID, many options like Name, RAID Level and possibly a Strip-Size were shown. Some controller also have the feature to change the RAID, for example to add or remove disks or even migrate to another level. Normally all settings are very simple and straight forward so there should not be any problem in managing RAIDs.

All RAID setup menus are different so there is not one single manual on how to set up a hardware RAID, but normally there is an manual from your hardware manufacturer. Just check out your main board or controller manual for further information.

#### 3.2 Software RAIDs

To use a Software based RAID, the only requirement is a running computer and some special tools to create such a RAID. Some operating systems (like Windows) have those applications already preinstalled. On other systems (e.g. Linux distributions) they are available for free (some are even open-source).

Normally a software RAID can do as much as a hardware RAID can do, but more information about the (dis)advantages in the Comparison.

<sup>13</sup>[Kum08, p.1, 1.4]

### 3.2.1 Linux (using mdadm)

Linux has (normally) no software RAID applications installed, but there are the open-source tools mdadm (multiple **d**isk **a**dministration) and gparted, which allows you to create a software RAID.

In this part I will show you how to create your own software RAID with these tools <sup>14</sup> .

#### 1. Install mdadm and gparted

As mentioned before, you need mdadm to set-up a software RAID. To format the new drive and create partitions, you will also need a tool for that (gparted is a good option, it is open-source as well). To install this, use the following commands (for debian based distributions):

```
sudo apt-get install mdadm gparted
```

#### 2. Find all hard drives

To create a RAID you need a specific amount of drives. In Linux each disk and partition has an entry in the /dev/ directory and has a structure of /dev/sd<a-z><0-9> (e.g. /dev/sdb1). You have to find the path to all drives and remind them.

#### 3. Create empty partition table

Next step is the creation of a new partition table on every drive that should be in your RAID, this allows you to create an empty drive.

```
sudo parted /dev/sd... mklabel gpt
```

**Note:** If you have **not** an (U)EFI based system, use msdos instead of gpt.

#### 4. Mark partition as RAID-partition

After creating a new partition table, the system has to know which partition is a RAID partition. This can be done with parted as well by using this command:

```
sudo parted /dev/sd... set 1 raid on
```

The first partition of the drive /dev/sd... becomes a RAID partition.

**5. Finally create the RAID** After marking all partitions as RAID partition the system is ready for the most interesting step, the RAID creation. One last time we will use mdadm with some special parameters:

---

<sup>14</sup>[ubu]

```

sudo mdadm
  --create /dev/mdx
  --auto md
  --level=xy
  --raid-devices=n
    /dev/sd<a-z><0-9>
    /dev/sd<a-z><0-9>
    ...

```

Here are some information about the parameter used in the command above:

<code>/dev/mdx</code>	Defines the path of the new RAID (e.g. <code>/dev/md0</code> )
<code>level=xy</code>	Sets the level of the RAID (e.g. 10 or 5)
<code>/dev/sd&lt;a-z&gt;&lt;0-9&gt;</code>	A list if all drives that should be added to the new RAID

Table 1: Creating the RAID via mdadm with lots of parameters.

**6. Create new file system** Now we have an empty RAID without any file system on it. To be able to store data, we have to choose on a file-system (e.g. `ext4`). For this, Linux has the command `mkfs` (make file system).

```

sudo mkfs.ext4
  -b 4096
  -E stride=s, stripe-width=t
    /dev/mdx

```

Here is a description of all parameter used above <sup>15</sup> :

<code>mkfs.ext4</code>	The file system you want (e.g. <code>ext4</code> , <code>ntfs</code> , ...)
<code>-b 4096</code>	This specifies the block-size of the file system (don not become confused with chunks of the RAID; normally a block is 4096 byte large)
<code>stride=s</code>	The amount of blocks needed for one chunk. Calculated by $s = \frac{\text{chunk-size}}{\text{block-size}}$ To get the chunk-size, use <code>sudo mdadm -D /dev/mdx   grep "Chunk Size"</code> (the output is something like e.g. 512K). Example: A RAID 10 with 256kB chunk-size and 4kB block-size has $256/4 = 64$ strides.
<code>stripe-width=t</code>	The amount of used blocks when the RAID writes a whole chunk. The stripe-width is calculated by $t = s \cdot k$ with $k =$ amount of <i>usable</i> partitions and $s$ the stride parameter (from above). Example: In a RAID 5 with 4 drives is $k = 3$ because one fourth of the RAID contains parity data. With a stride amount of 64 (256kB chunk-size and 4kB block-size) follows the stripe-width with $t = 256/4 \cdot 3 = 64 \cdot 3 = 192$ .

<sup>15</sup>[ubu]

**Note:** A RAID 1 (and other non-striping levels) just needs the command `sudo mkfs.ext4 /dev/mdx` because there is no striping.

## 7. Mount the new RAID

Now you can mount the new RAID drive with the normal mount command via `sudo mount /dev/mdx /media/....`

### 3.2.2 Windows

Windows has an built-in tool to create software RAIDs. It's available via `start → manage` or via the command `win+R → diskmgmt.msc`.

Due to its GUI, it is very simple to create a new RAID volume or to delete one. You only have to be sure that the disks you want to use, are formatted. There are also some other labels in the context menu of the disk manager. Instead of *RAID 0* it says *Spanned Volume*, nevertheless the functionality is the same.

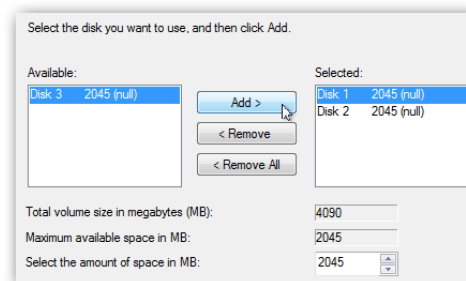


Figure 8: Software RAIDs are (mostly) easy to set-up.[WCC]

In the upcoming window (s. figure 8) you will have to choose some other hard drives that should be attached to the new RAID. After adding all drives the only thing to do is choosing a name and a drive-letter.

## 3.3 Comparison

Features	Software RAID	Hardware RAID Controller
Independent	No	Yes
RAID performance	Depending on Server(load)	High: Independent of server(load)
Functionality vulnerable to malware	Yes	No
Ability to migrate to other OS	Limited	Yes
Costs	Low costs	High costs
Typical RAID levels	RAID 0, 1	RAID 5, 6

Table 3: A short comparison of hard- and software RAID. Redrawn to [ada06].

## 4 Usage

A RAID is made to achieve high performance and/or for data security. Both aspects fit to companies with an own data center and for some private users with high requirements.

### 4.1 Private

The normal user of computer systems does not need a RAID for saving data. The normal HDD or SSD is absolutely sufficient. However, there are other people with more special requirements in their hardware.

One example are gamers. In order not to wait for saving and loading a game, level, map or anything else, some of them need/want the high reading and writing speed of a RAID. The performance can also be very convenient for backups because of their growth in size (about 40GB per game are pretty normal for recent AAA-games<sup>16</sup>).

Another system to store data in a private household are NAS-devices (**N**etwork **A**ttached **S**torage). Most of them use RAID systems to increase the security of their data<sup>17</sup>. These devices are used by e.g. families or small companies.

### 4.2 Commercial

As mentioned before, a normal data center uses RAID systems to increase their performance in terms of security and speed<sup>18</sup> (e.g. the DKRZ (Deutsches Klimarechenzentrum) uses RAID systems as well<sup>19</sup>).

One example are mail-server which have to handle many requests for a small amount of data. To process those requests a RAID with parallel drives (e.g. a RAID 10) is a good option<sup>20</sup>.

---

<sup>16</sup>[Sim]

<sup>17</sup>[Coma]

<sup>18</sup>[DCT]

<sup>19</sup>[Her97, p. 157]

<sup>20</sup>[Comb]

## 5 Appendix

### List of Tables

1	Creating the RAID via mdadm with lots of parameters. . . . .	12
3	A short comparison of hard- and software RAID. Redrawn to [ada06]. . . . .	13

### List of Figures

1	Disk 3 of this Level 4 RAID contains the parity data.[WCf] . . . . .	5
2	A RAID 0 system stripes the data and creates no redundancy.[WCb] . . . . .	6
3	RAID 1 systems saves the same data on each drive.[WCd] . . . . .	6
4	Parity data is spread over all disks.[WCG] . . . . .	7
5	RAID 10: Striped mirrors.[WCE] . . . . .	9
6	RAID 01: Mirrored stripes.[WCC] . . . . .	9
7	Most mainboards have an inbuilt RAID controller.[WCE] . . . . .	10
8	Software RAIDs are (mostly) easy to set-up.[WCC] . . . . .	13

### References

- [ada06] adaptec. Hardware raid vs. software raid: Which implementation is best for my application? [https://www.adaptec.com/nr/rdonlyres/14b2fd84-f7a0-4ac5-a07a-214123ea3dd6/0/4423\\_sw\\_hwraid\\_10.pdf](https://www.adaptec.com/nr/rdonlyres/14b2fd84-f7a0-4ac5-a07a-214123ea3dd6/0/4423_sw_hwraid_10.pdf), 2006. Last accessed: 18.04.2015.
- [Coma] Wikipedia Commons. Network attached storage. [https://en.wikipedia.org/wiki/Network-attached\\_storage](https://en.wikipedia.org/wiki/Network-attached_storage). Last accessed: 13.09.2015.
- [Comb] Wikipedia Commons. Raid - performance. <https://de.wikipedia.org/wiki/RAID#Performance>. Last accessed: 12.08.2015.
- [Comc] Wikipedia Commons. Raid 01. [https://de.wikipedia.org/wiki/RAID#RAID\\_01](https://de.wikipedia.org/wiki/RAID#RAID_01). Last accessed: 19.08.2015.
- [DCT] DCT\_Mark. The importance of raid storage in data centers. <http://www.datacentertalk.com/2011/12/the-importance-of-raid-storage-in-data-centers/>. Last accessed: 12.08.2015.
- [Her97] Axel Herbst. *Anwendungsorientiertes DB-Archivieren: Neue Konzepte zur Archivierung in Datenbanksystemen*. Springer-Verlag, 1997. ISBN: 978-3-642-60863-6.
- [jen] jenn2d2. Tardis interior. <https://www.flickr.com/photos/jenn2d2/15744546352>. Last accessed: 27.07.2015.
- [JH] Jason and Leah Heiss. Why is raid 1+0 better than raid 0+1? <http://www.aput.net/~jheiss/raid10/>. Last accessed: 29.08.2015.
- [Kum08] Arvind Kumar. Redundant array of inexpensive disks (raid) - technical paper. <http://www.finitysend.com:34999/get/pdf/93>, 2008. Last accessed: 18.04.2015.
- [Leu] Laurens Leurs. Raid. <http://www.prepressure.com/library/technology/raid>. Last accessed: 17.04.2015.

- [Sch] Universität Hamburg / Peter Schmidt. [http://upload.wikimedia.org/wikipedia/commons/thumb/9/94/UHH-Logo\\_2010\\_Farbe\\_CMYK.svg/500px-UHH-Logo\\_2010\\_Farbe\\_CMYK.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/9/94/UHH-Logo_2010_Farbe_CMYK.svg/500px-UHH-Logo_2010_Farbe_CMYK.svg.png). Last accessed: 10.05.2015.
- [Sim] Daniel Sims. Aaa game-file sizes are getting out of hand. <http://venturebeat.com/community/2014/10/17/aaa-game-file-sizes-are-getting-out-of-hand/>. Last accessed: 12.08.2015.
- [ubu] ubuntuusers. *Software-Raid*. <http://wiki.ubuntuusers.de/Software-RAID> Last accessed: 17.04.2015.
- [WCa] Wikimedia-Commons. Comparison. [http://en.wikipedia.org/wiki/Standard\\_RAID\\_levels#Comparison](http://en.wikipedia.org/wiki/Standard_RAID_levels#Comparison). Last accessed: 17.04.2015.
- [WCb] Wikimedia-Commons. Raid 0 with two disks (disk 0 and disk 1) over one logical volume a with odd blocks on disk 0 and even blocks on disk 1. [http://upload.wikimedia.org/wikipedia/commons/thumb/9/9b/RAID\\_0.svg/325px-RAID\\_0.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/9/9b/RAID_0.svg/325px-RAID_0.svg.png). Last accessed: 15.04.2015.
- [WCc] Wikimedia-Commons. Raid 0+1: striped sets in a mirrored set. [http://upload.wikimedia.org/wikipedia/commons/thumb/a/ad/RAID\\_01.svg/600px-RAID\\_01.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/a/ad/RAID_01.svg/600px-RAID_01.svg.png). Last accessed: 17.04.2015.
- [WCd] Wikimedia-Commons. Raid 1 with two disks (disk 0 and disk 1) over one logical volume a with all blocks replicated/mirrored from drive 0 to drive 1. [http://upload.wikimedia.org/wikipedia/commons/thumb/b/b7/RAID\\_1.svg/325px-RAID\\_1.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/b/b7/RAID_1.svg/325px-RAID_1.svg.png). Last accessed: 15.04.2015.
- [WCe] Wikimedia-Commons. Raid 1+0: mirrored sets in a striped set. [http://upload.wikimedia.org/wikipedia/commons/thumb/b/bb/RAID\\_10.svg/600px-RAID\\_10.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/b/bb/RAID_10.svg/600px-RAID_10.svg.png). Last accessed: 17.04.2015.
- [WCf] Wikimedia-Commons. Raid 4 with four disks (disk 0, 1, 2, and 3) and each group of blocks (orange, yellow, green, and blue) have a parity block on disk 3 (which is the dedicated parity block). [http://upload.wikimedia.org/wikipedia/commons/thumb/a/ad/RAID\\_4.svg/320px-RAID\\_4.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/a/ad/RAID_4.svg/320px-RAID_4.svg.png). Last accessed: 15.04.2015.
- [WCg] Wikimedia-Commons. Raid 5 with these four disks (disk 0, 1, 2, and 3) and each group of blocks (orange, yellow, green, and blue) have a distributed parity block that is distributed across the four disks. [http://upload.wikimedia.org/wikipedia/commons/thumb/6/64/RAID\\_5.svg/675px-RAID\\_5.svg.png](http://upload.wikimedia.org/wikipedia/commons/thumb/6/64/RAID_5.svg/675px-RAID_5.svg.png). Last accessed: 15.04.2015.
- [WCh] Wikipedia-Commons. Standard raid levels - raid 3. [https://en.wikipedia.org/wiki/Standard\\_RAID\\_levels#RAID\\_3](https://en.wikipedia.org/wiki/Standard_RAID_levels#RAID_3). Last accessed: 18.08.2015.
- [Wik] Wikipedia. Wikipedia - RAID. [de.wikipedia.org/wiki/RAID](http://de.wikipedia.org/wiki/RAID).