AFRAID? v2

A RAID primer for beginners and non-techies

wolfcrow

Written by **Sareesh Sudhakaran**

AFRAID?

Disclosure: *Links in this ebook may be to our affiliates; sales through affiliate links may benefit this site. Please help support this website by patronizing Amazon.* ***It won't cost you anything extra.***

A RAID system can boost your productivity, eliminate downtime and provide redundancy (not backup, we'll get to it) for your valuable data, even if you are one-man-one-computer shop.

You might have a few 'how' questions lined up:

- How does it work?
- How will it help *me*?
- How do I set it up quickly and painlessly?
- How do I learn about these things without taking a PhD?

This primer will explain:

- How computers work,
- How RAID works,
- Why it's different from backup,
- How it can help you as a video editor or media professional,
- How DAS, NAS or SAN works, and when to use what,
- How to set up a SAN for video production and how to incorporate RAID into it,
- What the best RAID level for editing is, and
- A comparison of ten Thunderbolt RAID storages for video editing.

What's new in Version 2.0?

Version 1.0, released in October 2012, focused mainly on the technicalities of RAID, without going into detail about how it affects a video editor or media professional.

This version takes it to a whole new level. The goal is to help you use RAID for your practical and commercial benefit.

How to use this eBook

This ebook is split into chapters covering only enough material to chew on for one day. It is written for the absolute novice who might have no clue about the intricacies of computer technology. You don’t have to know *anything*, trust me.

Start at the beginning. Don’t skip anything, even if the words sound ominous. I’ve taken great pains to simplify everything. And, I guarantee it’ll be an entertaining read! Why?

Because it is written in the form of a semi-fictional murder mystery!

You read that right. I’ve always hated textbooks. Anyway, it has been statistically proven that people learn better when there’s a crime involved (don’t quote me on that!).

Enjoy, and please give feedback!

CONTENTS

Chapter 1:

Data, File, Code and Software

Deep in the heart of town, hidden away in a shady street, is a RAID 5-star hotel called Hotel Network. Very few understand how this place runs or makes money.

A few curious detectives looked for clues in documents, receipts, or any pieces of paper that managed to leave the hotel. Every single one of them contained lines of 0s and 1s, nothing else! What did these numbers mean?

Everything in computing is 0s and 1s. Imagine a soda stream flowing through a pipe – a mix of water (0) and carbon dioxide (1). This is data.

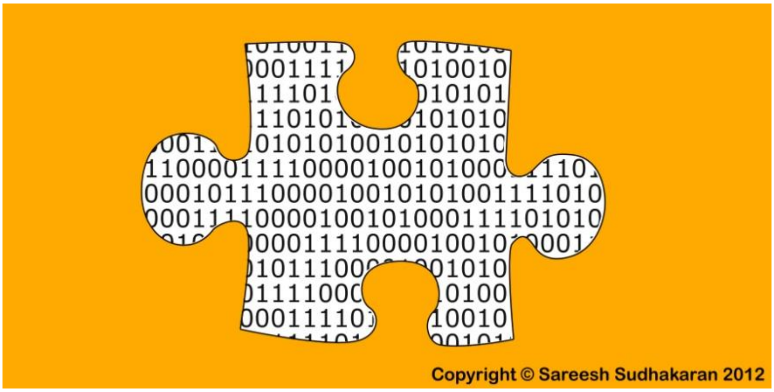
It can stay in one place, like a soda lake, or be moving, like a soda river.

The detectives assumed that the 0s and 1s were information of some kind. Technically, they were incorrect.

The 0s and 1s is just data. When it makes sense, it will become information. That's the difference between raw data and information.

Hotel Network was notorious for not using phones or email. Everything was on paper, in 1s and 0s. Sometimes the 0s and 1s were on small scraps of paper. Sometimes the detectives found large sheets of paper with 0s and 1s written from end to end.

It was evident that the size of the paper corresponded to the information written on it.



This paper is called a File in computer terminology. A file can contain raw data. If you know what that data means, the file then contains information. Whatever floats your boat.

It was clear to the dumbest detective in town that there must be a system or set of rules to govern the way in which 0s and 1s were written, quite similar to the rules of grammar and spelling that govern language.

Depending on what the information is, the way in which it is written might change. E.g., one file might be a phone number in lipstick, another might be pink slip, or a bill for an extravagant meal, and yet another might be a realty brochure promising clean blue water in dirt creek.

It's almost like using the alphabet from one language, like English, e.g., to write in many languages. This is called a File System.

Without knowing the rules that govern each file system, it is impossible to gather what the 0s and 1s mean. Only those who know the rules can make sense of it. If a file falls in the wrong hands, its information is safe.

A data file is an organized collection of data. It could be a shopping list, a telephone directory, a photograph, an encyclopaedia or even the whole internet downloaded into a .txt file.

The size of the paper is the file size. To calculate the file size, detectives found the best way was to count the number of bits. What's a 'bit'?

Each 0 or 1 is a bit. It is represented by 'b' (Always Lower case).

Eight bits make a Byte. If bits are letters then bytes are words. It is represented by 'B' (Always Upper case).

The size of a file was traditionally measured in bytes. As file sizes grew, it became important to find new units:

- Kilobyte – 1024 bytes, written as KB (Upper case)
- Megabyte – 1024 kilobytes, written as MB (Upper case)
- Gigabyte – 1024 megabytes, written as GB (Upper case)

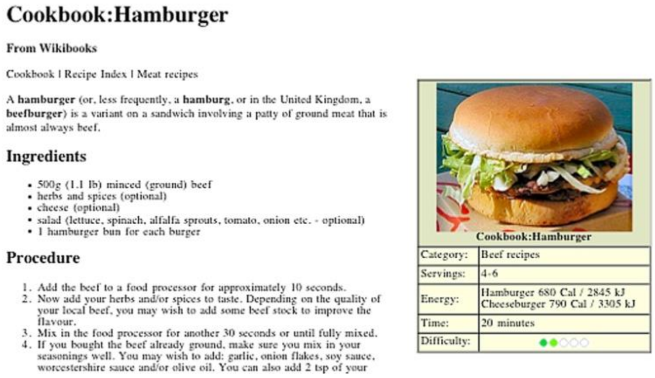
I'm sure you were irrirated by the word 'Upper case'. There's a reason I keep mentioning it: Sometimes, you get words like 'Kb' or 'Mb' or 'Gb'. Do they mean anything at all? Yes, they do:

- Kilobit – 1024 bits, written as Kb (Lower case)
- Megabit – 1024 kilobits, written as Mb (Lower case)
- Gigabit – 1024 megabits, written as Gb (Lower case)

As you can see, the 'case' means everything.

What if the file contains data that is not just information, but a set of instructions to do something?

On one scrap of paper, investigators found a set of instructions on how to prepare a hamburger – a recipe!



Cookbook:Hamburger

From Wikibooks

Cookbook | Recipe Index | Meat recipes

A **hamburger** (or, less frequently, a **hamburg**, or in the United Kingdom, a **beefburger**) is a variant on a sandwich involving a patty of ground meat that is almost always beef.

Ingredients

- 500g (1.1 lb) minced (ground) beef
- herbs and spices (optional)
- cheese (optional)
- salad (lettuce, spinach, alfalfa sprouts, tomato, onion etc. - optional)
- 1 hamburger bun for each burger

Procedure

1. Add the beef to a food processor for approximately 10 seconds.
2. Now add your herbs and/or spices to taste. Depending on the quality of your local beef, you may wish to add some beef stock to improve the flavour.
3. Mix in the food processor for another 30 seconds or until fully mixed.
4. If you bought the beef already ground, make sure you mix in your seasonings well. You may wish to add: garlic, onion flakes, soy sauce, worcestershire sauce and/or olive oil. You can also add 2 tsp of your

Cookbook:Hamburger

Category:	Beef recipes
Servings:	4-6
Energy:	Hamburger 680 Cal / 2845 kJ Cheeseburger 790 Cal / 3305 kJ
Time:	20 minutes
Difficulty:	●●○○○

Similar scraps were found in various paper sizes – depending on the complexity of the instructions.

A file ‘scrap’ with a set of instructions is called Code. A code is meant to be executed – no, not killed, but *done*. In other words, it’s a to-do list.

The person reading the file that contains code is meant to follow the instructions in the file.

Investigators also found code in gold-embossed paper in wooden handmade frames behind protective glass. To the casual observer they automatically screamed: “I’m precious! Look at me! I’m important!”

Obviously, they paid more attention to code that called attention to itself. This kind of code is called Software. Wares are what you sell in a marketplace. Code that is intended to be sold is called software.

With so many files flying around Hotel Network, it was a miracle they got anything done at all. The detectives were stumped until someone pointed out one crucial fact:

Every piece of code had the same person’s fingerprints! This person was obviously very important, and he or she probably knew everything there was to know about the hotel.

If only they could find this person!

Takeaways:

- **Data** – 0s and 1s.
- **Information** – 0s and 1s that make sense.
- **Code** – 0s and 1s that do something (like a magic spell).
- **File** – a wrapper that contains data, information or code.
- **Bit** – A 0 or 1 is a bit. E.g., 0100 is four bits (digits). It is represented by ‘b’ (Lower case).
- **Byte –** 8 bits is a Byte, just as 12 inches is a foot. It is represented by ‘B’ (Upper case).
- **Kilobyte –** 1024 bytes is a kilobyte. It is represented by ‘KB’.
- **Megabyte –** 1024 kilobytes is a megabyte. It is represented by ‘MB’.
- **Gigabyte –** 1024 Megabytes is a Gigabyte. It is represented by ‘GB’.

Chapter 2:

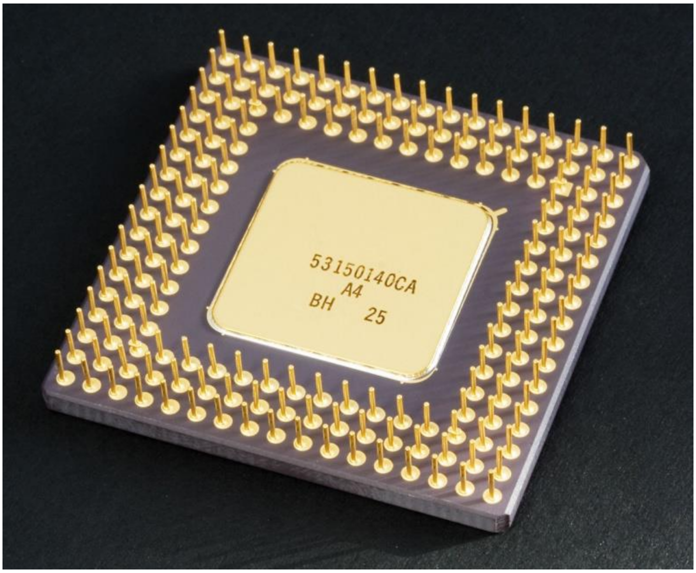
CPU, RAM, Motherboard and OS

It was the Concierge!

The poor guy had to receive code and data files from guests, staff and his bosses. It was his job to execute instructions. The hotel ran on this man's clock. They called him a Processing Unit (Heartless creeps!).

If he sneezed, somebody screamed: "Off with his Bottleneck!"

He couldn't stop. As long as Hotel Network runs, he must run faster. No breaks, no delays, and a whole lot of heat.



Photograph © Andrew Dunn, 9 November 2005. Website: http://www.andrewdunnphoto.com/

The processing unit 'processes' or thinks or calculates or executes – whatever the job description calls for. Since our concierge is the heart of the hotel, he is the Central Processing Unit, or CPU.

Of course, the dogs never let him know he was 'central'. His is a coming of age story that won't be told here.

There's a girl he likes. She works in reception. Her name is Miss Memory. She collects files and holds on to them until they are required.

The CPU dances to the tune of the guests, staff and his boss. Nobody knows where the next bit of instruction is going to come from, or what it'll ask the poor guy to do. If it's a large set of instructions, everyone else's demands will have to wait.

Miss Memory has to keep in step. She just accepts the CPU's random behaviour and writes it off to the demands of the job.

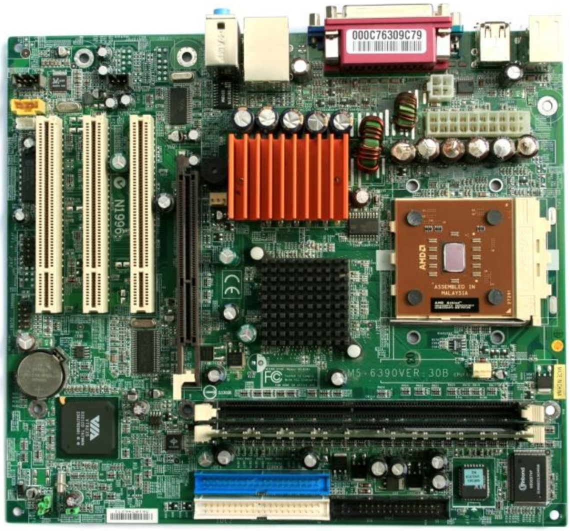
Of course, our shy PU doesn't have the guts, or the time, to ask her out. He doesn't even know her full name. He calls her Mem. Officially, she goes by the name: Random Access Memory, or RAM. That would have scared him off forever.

The lobby floor, the place where data physically moves, and where all the other components are connected, is called the Motherboard. If the motherboard is slippery, then somebody carrying the data might slip and fall.

"Bottleneck!"

If the motherboard isn't well designed, somebody might lose their way.

“Bottleneck!”



Attribution: I, Jonathan Zander

Who the hell is this, screaming "Bottleneck" all the time?

High above the main hotel lobby is a glass wall, through which the hotel manager keeps a tab on the CPU and RAM. He’s the dope who screams “bottleneck!” whenever a guest complains.

Sometimes he makes a fuss for no reason whatsoever.

Obviously, he is the law. Nothing happens that is not under his control. If he doesn’t like the look of a file, he labels it corrupt. That’s a death sentence in these parts.

He is the Operating System, or OS.

If a guest complains about a bill, or a leaking ceiling, or poor Wi-fi, the instructions are studied by the OS, and hotel rules apply strictly.

When making calculations, the CPU must follow the rules set out by the OS on how to deal with files.

Sometimes, if the bits of code are scamsters or goons, like viruses or trolls, then the hotel could be compromised. Nobody likes to be made a fool of. The OS keeps a strict vigil. If he fails it is the Blue Screen of Death (BSOD). When he blows his top, something or somebody will hang.

To keep one step ahead of everyone else, the OS must keep his information secure. Evidently there is a repository for data somewhere – a storeroom if you will, where the darkest secrets are saved.

Where is it?

Takeaways:

- **CPU -** The device that calculates and makes sense of information stored in files.
- **RAM** – Memory, much faster than any other kind of storage, designed to keep pace with the blistering fast CPU – like a nurse who assists a doctor during an operation.

- **Motherboard** – a circuit board the above components are connected to – very important part of any computer!
- **Hardware** – Together, the CPU, RAM, Hard Disks and Motherboard (and every other component) is what is called the hardware – the physical ‘hard’ parts of any computer.
- **Software** – The programs that are essentially just 0s and 1s.
- **Operating System** – a large software that controls every other file and software. It stands between files and the hardware.

Chapter 3:

Storage and Servers

Every file has to be stored somewhere. In the case of Hotel Network, these files were stored in special rooms called Hard Disks.

When a file is required, a Bellboy removes the appropriate file and transfers it to the RAM for processing.

There are many storage rooms in Hotel Network. When one hard disk fills up, another room is used. A bunch of rooms is called JBOD – or Just a Bunch of Disks. They have no relation to each other, except they happen to be placed near one another.

The Bellboy is given the keys by the OS. The detectives assumed that information contained in simple hard disks couldn't be that important, otherwise why rely on a mere Bellboy? There had to be some place else.

They found an underground vault with a guard sitting in front – an obnoxious tough guy who is lord of the vault.

This guy calls himself a Server. His job is to serve files that are asked for, by whoever needs them, as long as it is approved by the OS. Yeah, he's scared of the OS, but he likes to believe he has a mind of his own.

Author: Jonathunder under license: http://creativecommons.org/licenses/by-sa/3.0/deed.en

A Server is something that serves files when needed.

Because the Server has been sitting there for so long, everybody calls the underground vault a Server, too. The Server has no business outside the vault anyway.

Well, if it walks like a computer and quacks like a computer, then it must be one, too, right? Typically a server is just a computer, with the key difference that a server is optimised to be connected to a network, and is forever ready to send and receive files over it. It is built to perform only this function, day in and day out, without stopping for a tea break even.

We all recognize a personal computer when we see one:

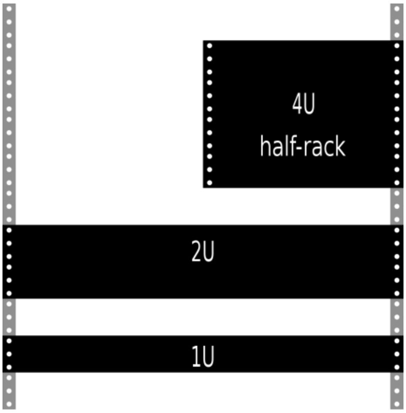
A server looks like this:

If both are computers why do they look different?

Take a look at this comparison:

The key is organization and storage space. A typical home or office puts a PC on a table. However, servers are usually used in tandem with other servers or hardware, and have to fit in places you wouldn't believe.

The height of each drawer is actually a standard, so that one can always plan one’s space nicely. This standard is designated by the letter U, called a Rack Unit, and it is 1.75 inches (44.45 mm) high. There are also standards for the width, mounting, etc.



When we need a drawer that can’t be the size of 1U (one unit), we can always design a box that is 2U, 3U, 4U and so on – as long as they are standard multiples of U.

Finally, it ends up looking like this:

Author: Jfreyre

See how the whole shebang is tucked neatly into a corner of the office? Here are some reasons to have a computer in server configuration:

- Wheels for portability
- Racks to save space
- Ease of maintenance
- Easy to manage cabling
- Better heat management

The OS is a crazy freak. In addition to the vault and JBODs he also has a locker in his office – for the most critical of files. It is connected only to his office. The rest of the hotel isn’t privy to it.

Wow! That’s a lot of information for the lowly paid detectives who weren't looking for trouble. The hotel was more sinister than it first looked.

What the heck is Hotel Network, really?

Takeaways:

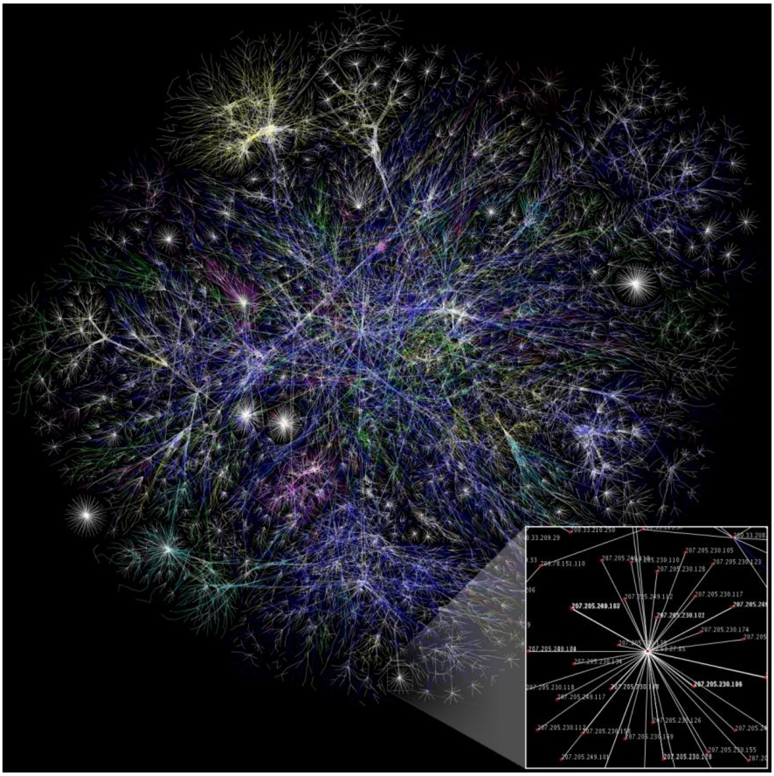
- **Hard Disk** – a physical device capable of storing large quantities of data.
- **JBOD** – Just a Bunch of Disks – Hard disks sitting next to each other but not related in any way.
- **Computer** – A box that contains a CPU, RAM, Motherboard and Hard Disks – created for the purpose to calculate or store something.
- **Server** – A term that has two meanings:
  - Software that controls who gets what files – it does the serving.
  - The computer (hardware) that is specifically designed to just serve files (and not play video games or watch movies).

Chapter 4:

A Computer Network

The detectives had deceived themselves! Initially, since they could only get a good look at the lobby and reception, they thought the Concierge was king.

The concierge, lobby and reception was just one computer. There were a lot more!



The Internet map 2005 Author: The Opte Project

Hotel Network had many departments – housekeeping, the kitchen, accounts, management, transportation, etc. Each of these were computers by themselves – each with a CPU at the head, with their own storage needs, data files and software - even their own operating systems.

The hotel has many floors, many ‘motherboards’. In server language, these boards are called Backplanes. Backplanes are designed to hold many types of hardware – Expansion slots, hard drives, motherboards, etc.

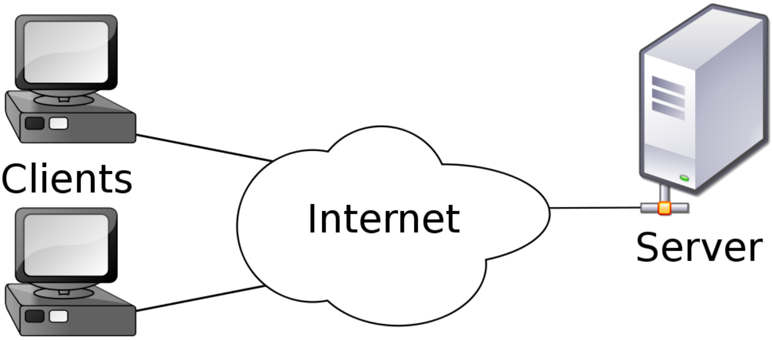
The OS’s office was called the Mainframe. The self-preening prick! Real life mainframes are important systems, mainly doing the most critical of applications. He was just a server with loads of self-importance in his software.

Each room or device is called a Node. Hotel Network is basically many nodes connected together. The guest computers are called Clients. They had to take whatever the servers wanted to dish out. Occasionally they requested for something or complained, and they got exactly what they paid for.

As you can imagine, no one tipped at this hotel.

Clients like to believe they worked independently, but they are always under the strict supervision of servers. Each department needs access to centralized data, for which the vault was created. Everybody bowed down to the mainframe. This network, in general, is called a client-server model.

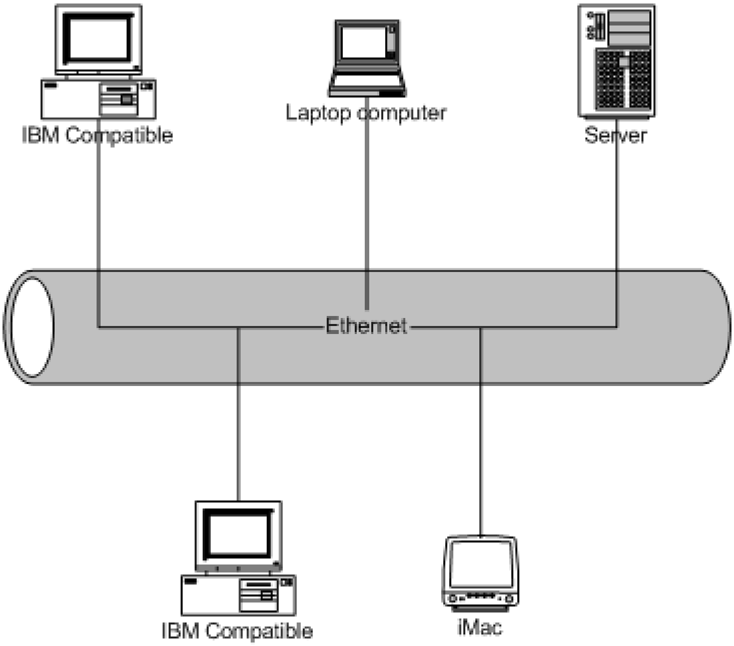
Client asks, server serves.



What an operation! This is what is known as a Computer Network – a linkage of computers with the single purpose of sharing and working together.

If you don't want to share, you don't need to be on a network. If you don't want to pass files around, you don't want a server. Want to keep everything personal? Use a Personal Computer (PC).

Hotel Network was a LAN – a Local Area Network. This is what a LAN looks like:



Author: Ilario

LANs can be strung together in many ways. It can get fairly complex, depending on how many devices you want to join together, how many *kinds of devices* you want to join together, how far away these devices are with respect to one another, and how much data has to be passed around.

Network devices use a special kind of language to talk to each other. These are called Protocols. It is similar to the political protocols followed internationally for acceptable behavior. If a node is not following a protocol correctly, it is not heeded. The most common protocol we know is the Internet Protocol, or IP. There are many others.

The lines between each node is either a wireless connection or a cable. The most common cable connection is the Ethernet connection. For professional applications, the most standard link is the Gigabit Ethernet standard, with a promised speed of 1 Gbps (Gigabits per second).

Computers connected thus, in infinite permutations and combinations, each serving one another as long as permissions are granted and they speak the same language, and sharing information on a global scale, is what we call the Internet.

At this point the detectives were awed. Hotel Network was an operation all right – but not a criminal one!

They had assumed it was bad just because it was complicated. It is actually a well organized, delicate, stable and robust system of information sharing. Nothing like this has been seen in the course of human history. Information can seem to travel at the speed of thought.

Actually, it travels faster.

The detectives had soda in the main lobby and marvelled at the Concierge’s efficiency at work, but stupidity in recognizing RAM's love. They appreciated the backplane architecture of the lobby and the millions of pieces of data flying around. They still cringed when the OS shouted: "Bottleneck!"

For the first time in the history of the hotel a waiter got tipped. The detectives left Hotel Network with smiles on their faces.

Little did they know what was in store for them.

Takeaways:

- **Backplane** – a motherboard for servers.
- **Mainframe** – the 'God' computer in a network of computers. Not all networks need mainframes. There can be a democracy, you know.
- **Node** – Every physical device connected to a network and counted on its own. Servers and computers are the most important kinds of nodes.
- **Network** – a connection of computers to each other so they can pass information and directives around.
- **Ethernet** – the name of the most widely used connection between computers.
- **Internet Protocol** – the language computers use to communicate with one another.
- **Local Area Network (LAN)** – a group of computers in a facility, office or home.
- **The Internet** – Billions of computers connected to each other around the world – sharing, talking, doing, faster than was ever possible before.

Chapter 5:

Data Flow

The key task of any network is data transfer. A network that can't get data around in time is a failed network.

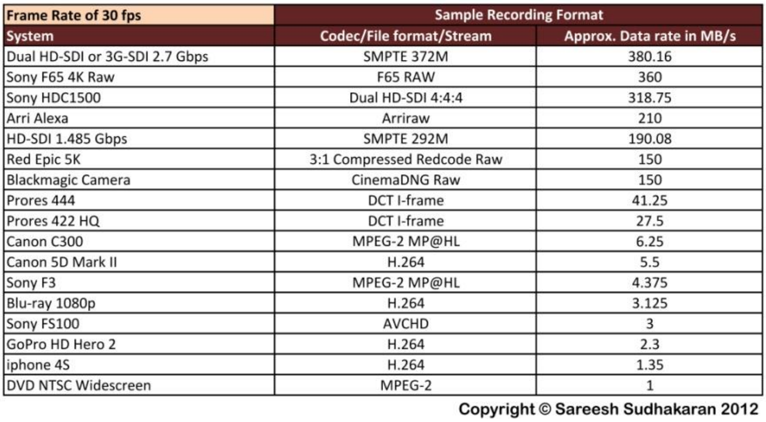
It's like a friend who learns about a special sale offer, and by the time her text message reaches you the sale is over. Phooey.

We, as filmmakers, editors or media professionals, are in the business of transferring images over a network. The first thing we should know is how much data needs to flow through the network. We can't order a truck until we know the size of the elephant we want to transport.

To help you do this, I have written a series of articles:

- Bit rate vs Data rate
- Understand the Sizes and Costs of Working with Uncompressed Footage
- The Limits of Data Transmission in our World
- A modern trend is to record in RAW files, which typically, when in uncompressed mode, is 1/3rd the size of full raster images. Read about that in Deconstructing RAW.

Now, quickly, let's take a look at the data rates of certain standards:



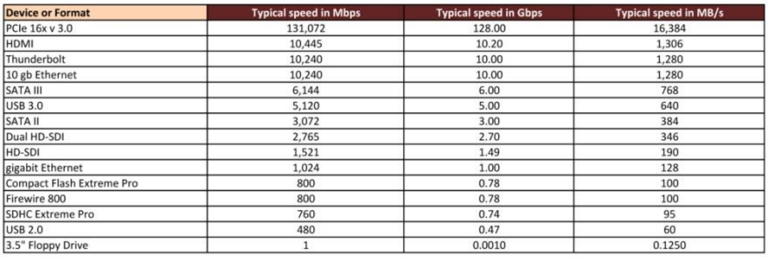
Frame Rate of 30 fps	Sample Recording Format	
System	Codec/File format/Stream	Approx. Data rate in MB/s
Dual HD-SDI or 3G-SDI 2.7 Gbps	SMPTE 372M	380.16
Sony F65 4K Raw	F65 RAW	360
Sony HDC1500	Dual HD-SDI 4:4:4	318.75
Arri Alexa	Arriraw	210
HD-SDI 1.485 Gbps	SMPTE 292M	190.08
Red Epic 5K	3:1 Compressed Redcode Raw	150
Blackmagic Camera	CinemaDNG Raw	150
Prores 444	DCT I-frame	41.25
Prores 422 HQ	DCT I-frame	27.5
Canon C300	MPEG-2 MP@HL	6.25
Canon 5D Mark II	H.264	5.5
Sony F3	MPEG-2 MP@HL	4.375
Blu-ray 1080p	H.264	3.125
Sony FS100	AVCHD	3
GoPro HD Hero 2	H.264	2.3
iphone 4S	H.264	1.35
DVD NTSC Widescreen	MPEG-2	1

Copyright © Sareesh Sudhakaran 2012

What are some of the scenarios in which uncompressed footage is transported in real-time (or faster!)?

- Playback and Dailies
- Grading
- Visual Effects Facility
- Fast Copying or Backup

Now let’s have a reality check:



Device or Format	Typical speed in Mbps	Typical speed in Gbps	Typical speed in MB/s
PCIe 16x v 3.0	131,072	128.00	16,384
HDMI	10,445	10.20	1,306
Thunderbolt	10,240	10.00	1,280
10 gb Ethernet	10,240	10.00	1,280
SATA III	6,144	6.00	768
USB 3.0	5,120	5.00	640
SATA II	3,072	3.00	384
Dual HD-SDI	2,765	2.70	346
HD-SDI	1,521	1.49	190
gigabit Ethernet	1,024	1.00	128
Compact Flash Extreme Pro	800	0.78	100
Firewire 800	800	0.78	100
SDHC Extreme Pro	760	0.74	95
USB 2.0	480	0.47	60
3.5" Floppy Drive	1	0.0010	0.1250

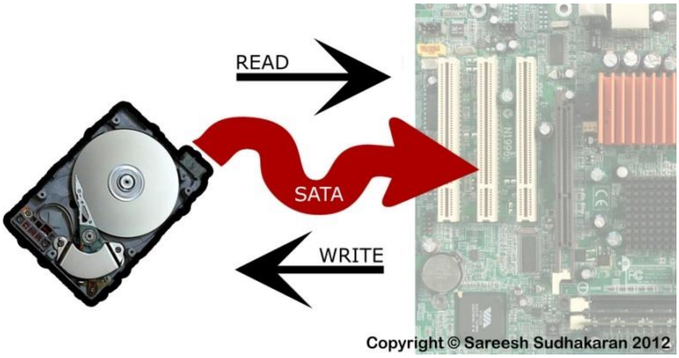
Copyright © Sareesh Sudhakaran 2012

***Note:** Thunderbolt 2 is capable of speeds of 20 Gbps, while the newer version of USB 3.0 is capable of 10 Gbps.You can see why Thunderbolt is popular!*

You're probably thinking: there are some numbers there that are way above uncompressed footage rates. So what's the problem?

The problem is: a chain is only as strong as its weakest link. Data transfer through a computer or network is bits flowing through a chain of devices and events, any one of which can lag behind - it's like a relay race.

Let's see how:



In its simplest sense, data is read from or written to a hard disk. A hard disk can only do one thing at a time.

One of the important parameters that affect hard drive transfer rates is the Average Access Time. The average access time is the time a drive takes to get its act together to begin transferring data. Imagine someone asking you your name. Even in your prime, it takes a few milliseconds for the request to reach your brain and result in an answer. Hard drives are no different.
One of the methods in which hard disk transfer efficiency is increased is by means of a Disk Buffer or Cache. On good hard drives, this size is about 64 MB, and it acts like an overflow outlet on a plumbing line.

In any case, a typical SATA drive only gives out half to two-thirds of its rated speed. This means for SATA II you'll get about 150-250 MB/s and for SATA III you'll get about 500-600 MB/s tops - at least with today's technology.

It’s not just hard drives. Even Ethernet, rated at 1 Gbps, is hindered by the speed of modems (100 Mbps). The examples are endless.

These hindrences, that limit the potential data transfer rate of any network or system, are called a Bottleneck.

If you are out in the market to buy or build a storage system for your filmmaking or editing needs, your goal is to eliminate as many bottlenecks as you can, while keeping your whole operation profitable.

E.g., SSDs (Solid State Drives) are faster than typical Platter Drives (Spinning Hard Disks), but at a price. We have already seen that consumer hard disks max out at about $0.12 per GB. For SSDs, this number is about $0.75 per GB - that's 6 times the price. It's one of the several factors you must consider when putting together a storage system.

Everything starts with an estimation of the total amount of footage you are going to use on any project. When working with RAW or uncompressed files, the footage time is in the hours, and the data capacities required for storage are in the Terabytes (TB). On top of that, you also need to back up data to at least one other drive. That easily doubles the price, minimum.

The writing is on the wall - the bottleneck today for data transfer is the hard drive (disk). See where this is going?

It is a juggling act not for the faint hearted.

Out of this chaos, one name stands tall – a hero that gets all the bad press but saves the day.

It is called RAID.

Takeaways:

- **Data Rate** – the number of MB that you can transfer per second over a connection, written as MB/s. The same thing can be expressed as Mbps, but that means Mega bits per second, and they are related like this: 1 MB/s = 8 Mbps.

- **USB** – the most popular and widely used connection standard that lets you connect hard drives and other devices to your computer.
- **Thunderbolt** – a proprietary standard developed by Intel and Apple that doubles the speed of USB and allows long cable lengths.
- **Platter Drive** – a spinning disk, typically most hard disks.
- **SSD** – Solid State Drive, a hard drive technology that does not need spinning disks. All pen drives, thumb drives, camera and mobile cards, etc., are solid state drives in the strict sense.
- **SATA** – the connection between the hard drive and the motherboard, which must be fast enough to take full advantage of any drive.
- **Terabyte (TB)** – 1024 Gigabytes

Chapter 6:

What is RAID?

Before the sodas were digested our detectives got called back to Hotel Network.

A brutal crime had taken place. A hard drive was found dead in one of the storage bays. Foul play was suspected. Was it murder or suicide? Or was the death caused by 'natural causes'?

Preliminary reports indicated a clear case of overwork - the poor guy's spindle had burst, stopping data flow and causing instant death.

Curiously, Hotel Network wasn't affected. It continued to function. How did it manage that?

"Hard drives have two major drawbacks. They're slow and they fail", declared the OS. "I would be stupid to rely completely on them."

"You obviously use backups."

"Obviously, detective. I don't need to be told how to run my shop. But one doesn't have time to run around looking for hard drives in Hotel Network. *Data must always be present and ready for use*."

"You mean to say, even if drives are slow and unreliable, they still have to be fast and reliable?"

"Well put, detective. That's what we do here. Ever heard of the story of the father who taught his sons that one twig can break easily, but twigs tied together are harder to break?"

"How on earth do you manage that?"

"Easy. I create an army of hard drives working together. By taking away their independence, I make them stronger and faster."

The slave driver!

Imagine watching a movie with your friends, from a file stored on a hard drive. Just when you reach your favorite scene, the drive fails. Your friends pass disgusted looks all around, except in your face.

You mumble a few apologies, curse the drive manufacturer, and walk to (or drive or fly or log on to) your backup area, take out the backup disk – and connect the backup. Then, you connect another backup drive (you no longer have a backup of the backup, remember?) to copy this disk. After the copying is done you disconnect the new backup and take it back (or drive or fly or upload to) to your backup area.

Then, you get back to your movie and hit the play button. Your favourite scene isn't your favourite anymore. Your friends have already updated their social network statuses.

Studies have shown that the typical drive failure rate is about 2% to 5% on average. This kind of performance hit is acceptable to most consumers – that's why computers and laptops come with single drives. But it's not acceptable for mission critical work where your life, or your pay check, depends on it.

In video production, time is money, and it is critical to have your expensive color grading artist work on the grade rather than wait for an hour or more for data to be replaced.

It is also important for a DIT on a set where everyone needs real-time feedback of what is being shot. Nobody has time or the money to wait around while you hunt for that backup disk.

One needs a system of backup that is 'real-time'. If one drive fails, another drive must take over and the system must continue to function. What kind of system does this?

Enter RAID, or Redundant Array of Inexpensive Disks (Some substitute Independent for Inexpensive, but I'll stick to the original acronym).

A bunch of disks in RAID is called an Array, so we often refer to them as RAID arrays. In simple terms, if JBOD is a bunch of goons then a RAID array is a special forces unit - highly organized and prepared – each soldier watches the other’s back. A drive in a RAID array is expected to fail at any time. Nobody is indispensable.

Let’s get one thing clear – RAID is not intended to be backup. You still need to copy your data to other drives or the Cloud or whatever to keep them really safe.

The commander of the RAID array is a slave master. He is called the RAID Controller. RAID controllers can be hardware or software based. In the case of Hotel Network, the OS was controlling the RAID array directly.

Could this have been the reason for the drive's untimely 'failure'?

"You can't suspect me!" remonstrated the OS. "I'm the Operating System! Drives fail. That's the reality of it."

"We'll see."

The detectives felt it was necessary to investigate Hotel Network's RAID system in greater detail. Which RAID was it?

This was going to be a whole new level of weird.

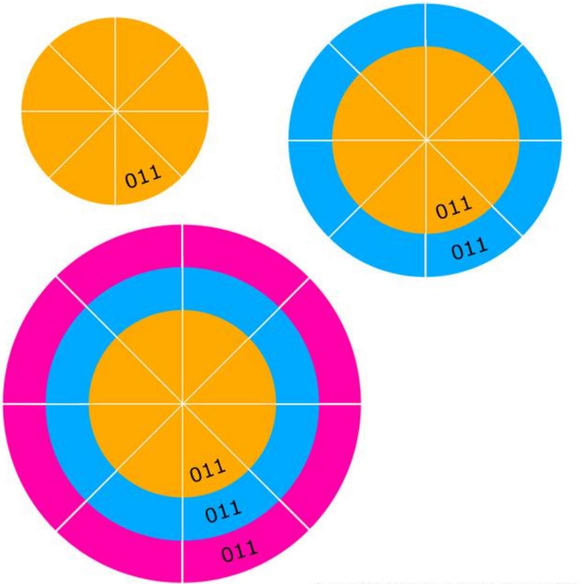
Takeaways:

- **Backup** – a copy of your data. Data gets corrupted, overwritten or lost. Backups are a way of life.
- **Redundancy** – a philosophy that ensures your work doesn't stop if one drive fails.
- **RAID** – the technology that utilizes the concept of redundancy to hard drives.
- **RAID Array** – a group of drives in RAID, which forms a separate unit and acts like one super hero hard drive.
- **RAID Controller** – a hardware or software that controls the RAID Array.

Chapter 7:
Understanding RAID

Before they started investigations, the detectives huddled together to mull over how to tackle the insane RAID problem, and to have some wonderful pizza served fresh from Hotel Network's famous kitchen.

The pizza came in three sizes, and it was difficult estimating whether an increase in size (which also increased the price) gave a proportionate amount of toppings.



Copyright © Sareesh Sudhakaran 2012

Each pizza slice is called a Block. A block is just data grouped together so it can be read or written easily. Think shoveling. It would take an insane amount of time to move sand if we had to do it one particle at a time, but thanks to a shovel it can be done faster.

Data is bits of 0s and 1s, as we have seen, and a block is a group of 0s and 1s. The size of the block is decided by the file system.

The pizzas were delicious. Some detectives could swallow an entire slice in one go. Others had to take tiny nibbles. It was a constant battle between saving time vs -

"Bottleneck!"

One detective almost choked. No, it wasn't the guy who swallowed a whole slice. Not this time, anyway.

With so many pizzas leaving the kitchen how did the chefs keep track of the toppings? How did they ensure each slice had the right number of toppings?

The answer to error detection is an elegant solution called a Parity Check. How does it work?

Simply put, a bit (0 or 1) is added at the end of each block, to either make it even or odd. The parity system is fixed to either even or odd, not both.

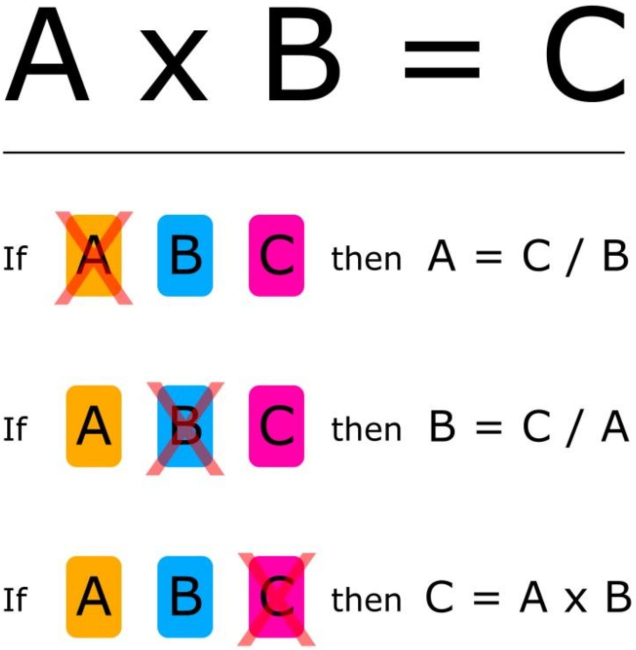
One of the detectives screamed: "Are you friggin' crazy? There are enough 0s and 1s flying around anyway, and you want to *add* one more?" This detective was promptly calmed down with a can of beer, with compliments from the chef.

Initially, things were a mess, the chef explained. Pizzas roll out of the kitchen like an assembly line. The chef responsible to add toppings just looked at the pizza already done, and repeated the same on the next pizza, and so on.

Very soon they realized all it took was one mistake, and every pizza henceforth would have that error. If the chef added an extra slice of pepperoni on one pizza, then every subsequent pizza had the extra pepperoni.

The head chef corrected this problem by asking the chef to look at the last pizza and compare it to a standard pizza chart. From then on, things went perfectly.

Here's an example of how a parity check works in general:



Copyright © Sareesh Sudhakaran 2012

If we have two drives A and B that need to be filled with information, we add a third drive C as a parity drive. Its data is calculated using the formula predetermined as A x B = C.

Now, if one drive fails or has an error, its information can be recreated from the other two drives by using the same formula.

What if we just backed up the data in the traditional way? If we had to backup the information in both drives A and B, we'd need two more drives C and D. By using the parity method, we have a way to keep data 'backed up' with only 3 drives. Neat.

This, in theory, is a simple process. In reality, though, it's anything but. Like all solutions, it has its pros and cons. However, it is one of the bedrocks of many RAID systems.

All the math behind a RAID system is governed by the RAID Controller. With some controllers, once a RAID system is built, if the controller fails, then one might have to purchase the same model, or at least another similar controller from the same manufacturer.

It's like having a librarian for a large library organizing books in alphabetical order according to the last name of the author. When he moves on, the guy coming after cannot find anything if his system is to sort books by the first letter of the title, and so on. There are infinite ways in which data can be organized.

Which means, there is more than one kind of RAID system. It's a whodunit for most professionals. Probably the most frequently asked question about RAID is not "What is RAID", but "Which RAID should I use?". Each system has its own unique properties - just like the army, navy and the air force. They are generally classified as the same thing, but are built and organized separately, with different goals in mind.

For better or for worse RAID systems are designated with numbers:

- RAID 0
- RAID 1
- RAID 2
- RAID 3
- RAID 4
- RAID 5
- RAID 6

RAID 0 isn't even a true RAID, because there's no redundancy. But it has an ace up its sleeve that no other RAID has. We'll take a look at each one soon.

There are many situations in which one objective might be better served with two kinds of RAIDs. Can we have RAIDs of RAIDs? Yes, and they are designated thus:

- RAID 0+1
- RAID 10
- RAID 50
- RAID 53

- RAID 100

And so on. You are only limited by your imagination (and some technology). This kind of structure is called a Nested RAID.

The key thing is to remember that once a bunch of disks are configured in a RAID system, the RAID system behaves like a single disk to an outsider. It's exactly like a hard drive, except with super powers.

RAID numbers are written in the format: RAID XYZ. How do we make sense of the format RAID XYZ? Visualize it in this way:

- RAID X is a box of drives. If you're having trouble imagining, use DVDs instead. RAID X is a box of DVDs.

- RAID XY is a RAID Y truck filled with RAID Z boxes.

- RAID XYZ is a RAID Z ship filled with RAID Y trucks, each truck filled with RAID X boxes.

So if you're faced with a dilmena of remembering the difference between, say RAID 01 (RAID 0+1) and RAID 10 (RAID 1+0), then just use the above visualization. The left-most number is the lowest level.

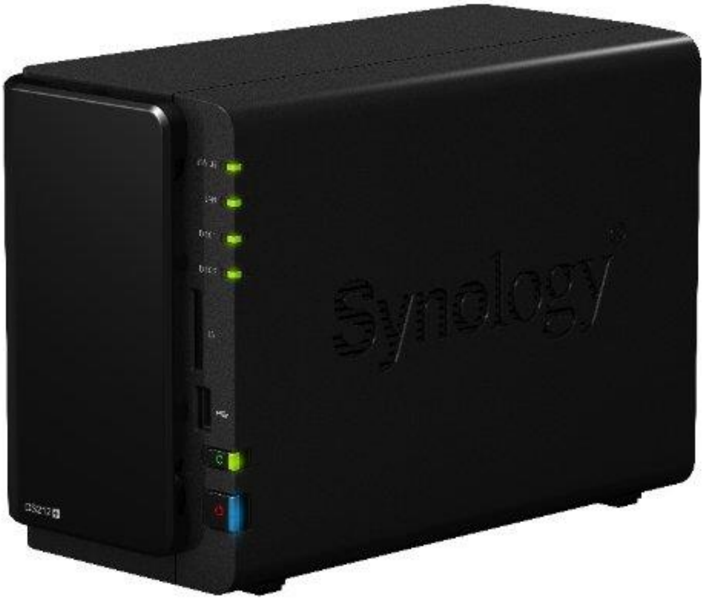
In the next chapter, we'll take a look at the two gods of the RAID mythology - RAID 0 and RAID 1.

Takeaways:

- **Block** – how a hard drive groups data so they can be tranferred faster – like shovelling sand instead of transferring one particle at a time.
- **Parity** – the method of using two data sets to calculate a third value. This is used to save space and bandwidth.
- **RAID XYZ** – a numbering system that tells you what RAID technology is being used.

Chapter 8:

RAID 0 and RAID 1



“This is getting out of hand.” declared the detective in charge. “If we have to get a handle on RAID levels, we’ll need something to guide us.”

Right then, one of the housekeeping staff – a busty lady – handed him a note.

Had he scored? With a twinkle in his eye, he opened the note.

On it was scrawled:

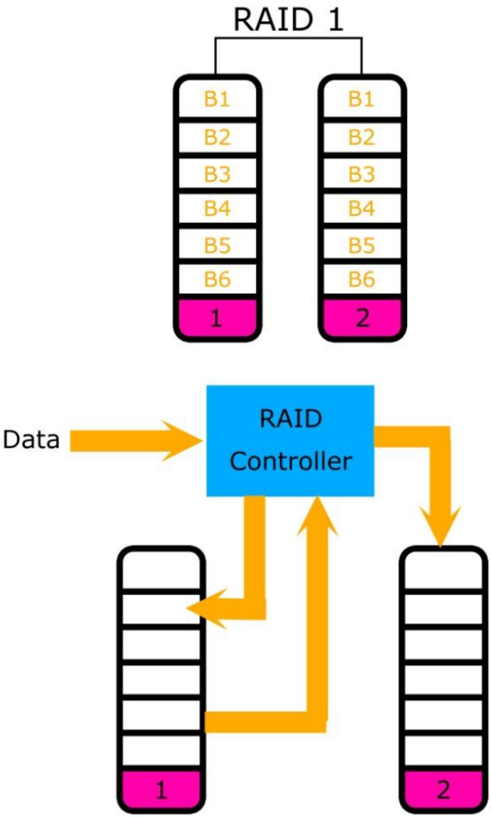
To understand RAID, you’ll need to look at
stability, speed and capacity.

Stability tells us how well the RAID will remain active, and how well it can heal (rebuild data) itself once a drive fails. Speed tells us how fast the RAID array will read and write data. Capacity tells us how much of the total capacity needs to be sacrificed to fulfil a particular RAID architecture.

A 1 GB drive in RAID does not always give 1 GB.

“Boys,” the detective announced, “we’ve hit the mother lode.”

Confidently, they walked into that part of the hotel known in hushed tones as RAID 1, commonly drawn as a pair of hanging nunchucks:



Copyright © Sareesh Sudhakaran 2012

RAID 1 needs at least two drives. If drive one holds X amount of data, drive two will have the same data copied to it – a duplication.

This duplication is called Mirroring.

If one drive fails, the other has all the information, and the system continues functioning.

RAID 1 is the simplest RAID architecture.

What's the difference between two drives in RAID 1 and two drives having the same data but are JBODs?

By assigning the RAID 1 architecture to the two drives, you're telling your computer that these two drives will *behave like one drive*. How does that help?

When one drive fails, the RAID Controller will ensure the data is read from the second drive and all is normal. The Computer will never know what took place inside the RAID array. All it cares about is its data getting read or written.

In JBODs, on the other hand, when one drive fails, you'll physically have to notify the computer or software to switch over to the other drive, and probably start from scratch as well.

As you may have noticed, RAID 1 does not use parity.

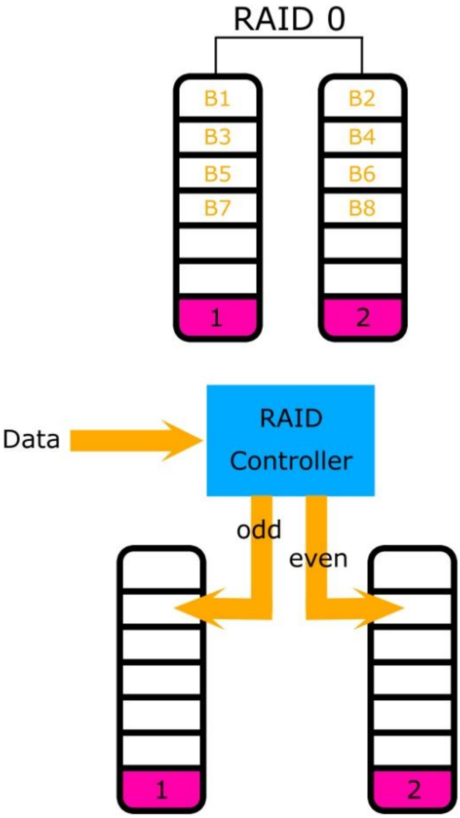
The advantages of RAID 1 are that it is simple to implement, and is compatible with many different kinds of systems. It gets along with almost everybody. Also, it can theoretically double the read rate of each drive. I say, theoretically, because it requires a special bit of juggling by the Controller to make this happen, and not all do.

The disadvantages of RAID 1 are that it can only write as fast as each individual drive's write rate. Also, we'll need at least double the capacity for each block of data. Adding more drives just makes it worse. Each drive will have the same information as the first drive. Five drives is five copies, with a space efficiency of 1/5 or 20%.

Oops, that's why we need other levels of RAID.

RAID 1 is great when capacity isn't a concern, two drives are good enough and the drive isn't used much for writing, only reading. Within these limitations, it is king. If you need only two drives, and are only reading source footage for your NLE or workstation, then nothing beats a RAID 1 for redundancy (with the right controller).

In the last chapter we also mentioned a RAID that is not really a RAID. That's RAID 0. RAID 0 is a wily character, and wasn't easy to find. When the detectives entered the domain of RAID 0, they encountered the outcast.



Copyright © Sareesh Sudhakaran 2012

From afar, RAID 0 looks like RAID 1. Two drives, same nunchucks. Nope. They are as different as male and female.

In RAID 0, data is written to both drives simultaneously, filling both of them up. First block here, second block there. Third block here, fourth block there, and so on. They don’t have the same information.

RAID 0 has twice the capacity of RAID 1.

“You’re no good,” the detectives thought about RAID 0. “What good is a RAID that doesn’t have redundancy? You’re a blotch to your name.”

“Don’t judge me so quickly, detective,” RAID 0 answered. “I have a skill nobody else has. I am twice as fast.”

“That’s BS. Data is written to one drive, then another. How can you be twice as fast?”

The data written to RAID 0 is split into half. The first block is written to drive one, and the second to drive two. The third is written to drive one, and the fourth to drive two, and so on, until all the data is written.

This way, no drive has to wait around, and as long as the computer can push data through the system at twice the speed the RAID 0 array can read and write at that speed.

“What about backup?” asked a sharp detective.

“I don’t have backup. I am built for speed.”

The great advantage of the RAID 0 array is that it can theoretically increase the total speed of the array depending on how many drives it has. If the array has two drives, the speed is doubled. If the array has ten drives, the speed is ten times the speed of a single drive!

RAID 0 leverages the speed limit imposed on data transfers due to hard drives and turns it on its head.

One tortoise can only nibble at a whale's ears. Many tortoises strung together can eat the whale whole. With RAID 0, you can theoretically build a drive array that can out pace the entire computer or network system, so that your hard drives are no longer a liability.

The cutting up of data into pairs for writing or reading, as in the case of RAID 0, is called Striping.

All RAIDs do at least one of three things – Mirroring, Striping and Parity. This is what sets them apart from JBODs.

RAID 1 has no striping or parity. RAID 0 has no mirroring or parity.

The biggest disadvantage of RAID 0 is, if one drive fails, all is lost. Since there's no redundancy at all, you lose all your data in that array, and will need to start from scratch.

Before you complain, remember that this is very similar to a single hard drive failure. However, one would be foolish to expect redundancy from such a system. RAID 0 is used for the best read and write speed, nothing else.

Which makes it the best candidate for real-time uncompressed media streaming – as long as everything else can keep pace.

It didn't take long for people to realize if RAID 0 and RAID 1 were the gods of the RAID world, surely they can get along together.

What if we could have both mirroring and striping – speed and redundancy? How do we do it – RAID 0+1 or 1+0? Let's look at that next.

Takeaways:

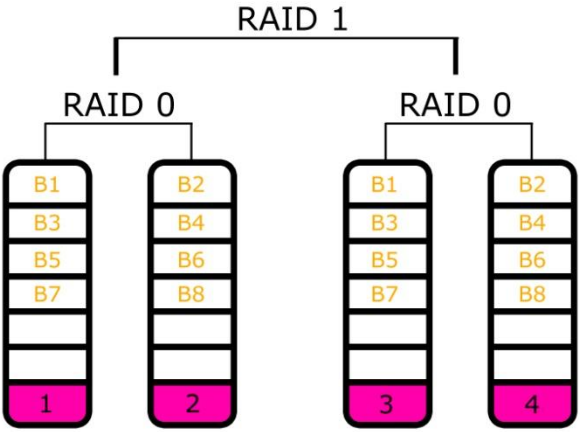
- **Mirroring** – the duplication of data to two drives.
- **Striping** – Dividing the data into different drives.
- **RAID 1** – Two drives, mirrored. Acts as one drive. If a single drive fails, you can carry on working because the other drive will step up to the plate.
- **RAID 0** –Two drives, striped. Acts as one drive. It can deliver data at twice the speed of a single drive. The more the drives in RAID 0, the faster it gets. Theoretically, at least.

Chapter 9:

RAID 0+1 and RAID 10

RAID 0+1 (or sometimes called RAID 01) is an array of many hyperactive RAID 0 kids in a play pen governed by a level-headed but slow RAID 1 nanny.

This is what it looks like:



Copyright © Sareesh Sudhakaran 2012

It potentially has the same drawbacks of RAID 1 – one needs double the capacity. It also puts a cap on the speed advantage of RAID 0 on the total throughput of the array.

Data flows to the first RAID array and is striped, just like I showed in the last chapter. Once the data is written, the system then acts like a RAID 1 array, and the data is mirrored to the next RAID 0 array.

If you're having trouble imagining it just remember that each RAID 0 array acts like a drive, so the whole system works like a RAID 1 array.

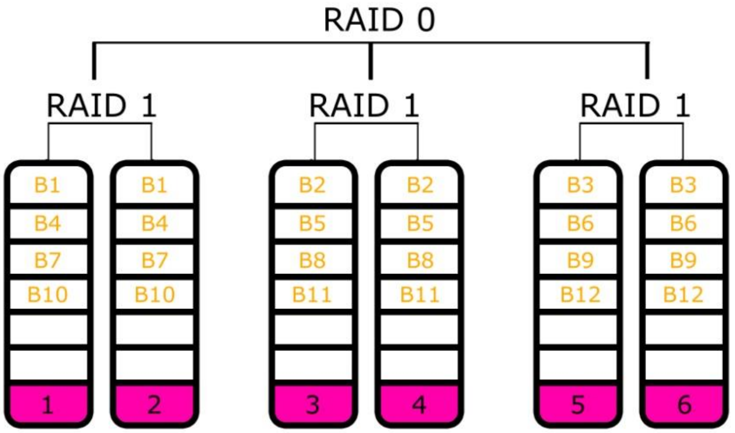
Think fast cars being transported on a slow ship. Instead of marrying the advantages of RAID 0 and RAID 1, we seem to have married the disadvantages!

So what the hell use is it? Whenever you need a RAID 1 array but each drive isn't fast enough to get the transfer speeds you want you can boost performance by using RAID 0+1 instead. The failure of a single drive doesn't theoretically stop the operation of the array.

Let's say a SATA II drive has a theoretical speed of 3 Gbps. Striping two of them in RAID 0 doubles the transfer speed to 6 Gbps. If two of these arrays are mirrored via a RAID 1 array, thus making a RAID 0+1 array, each RAID 0 array behaves like a 6 Gbps drive. Cool, right?

The other way to marry RAID 0 and RAID 1 is by RAID 10 (or RAID 1+0).

This is what it looks like:



Copyright © Sareesh Sudhakaran 2012

If RAID 0+1 was a mirror of stripes, then RAID 10 is a stripe of mirrors. Ha! The detectives were caught in a twist. They couldn't digest the jargon.

At this point they sat down and ordered a large café latte each with an extra shot of espresso. The sodas just didn’t cut it anymore.

It was hard enough keeping track of RAID 0 and RAID 1, but RAID 0+1 and 10 took the cake. After the small break, they decided to plough ahead.

An array that spreads its data around is said to be a stripe. An array that mirrors is said to be a mirror.

RAID 10 is a stripe of mirrors. Many quiet elderly folk (RAID 1s) in an old-age home run by a hyperactive nanny (RAID 0). Or think slow cars being transported on a supersonic jet.

Each RAID 1 array acts like a drive, with the advantages and disadvantages of a RAID 1 array. But by being striped together, their speed disadvantages vanish, while their mirroring ability remains.

Each block of data is mirrored or duplicated, yet the entire system also has the speed advantages of a RAID 0. The almost perfect match.

If any drive in a RAID 10 array fails, the system still works. In fact, one drive in every RAID 1 array can fail, as long as both don’t. The great advantage this has over a simple RAID 1 array is that the capacity hit is fixed.

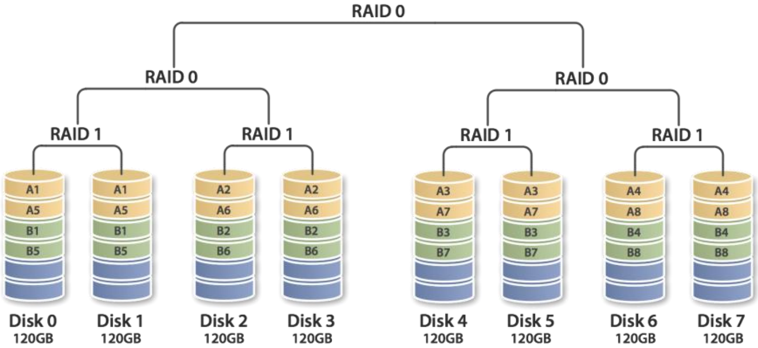
Let's say two drives are used per RAID 1 array in a RAID 10 array, then the capacity will stay at 50% no matter now many more arrays are added.

As we have seen earlier, this is not the case in RAID 1, where the more drives you add, the worse the capacity gets.

This is how RAID 10 combines the speed of RAID 0 with the mirroring ability of RAID 1, while keeping the capacity hit to less than 50%.

RAID 10 is used for both read and write-intensive applications such as database, email, and web servers, as well as for any other use requiring high disk performance – like video streaming and transfer in a post production studio.

One can also have a RAID 100, which looks like this:



Author: Kauberry

A RAID 100 is a stripe of RAID 10s, which is a stripe of RAID 1 mirrors. Getting the hang of it? Think about my box, truck and ship analogy.

What’s the advantage in being striped ‘two-ways’ - a double RAID 0?

It gives you the ability to use multiple RAID controllers to spread the load. The more drives you add to an array the more controllers you'll need. It gets expensive very quickly.

You see, each Controller has a fixed number of ports. After a point it makes more sense to add more controllers than ports. Why?

Increasing the number of controllers also increases the efficiency of random read performance. If your data is not being read in large chunks, but in small packets spread all over the array, then this is your ticket.

It might not make sense theoretically, but practical necessity and economics forces this kind of 'compromises' on us. Very few manufacturers make good controllers, so the price points they set are what we have to contend with, for better or for worse.

For this reason, RAID 100 is often used for large databases, where hardware RAID Controllers limit the number of drives that can be connected per array. The speed advantages are obvious enough, and the complexity and costs increases exponentially.

"I didn't sign up for this!"

The detectives had had enough. They hadn't bargained for such complexity.

The head detective wondered how many would call in sick the next day.

“Let’s forget about RAIDs 0, 1, 01, 10 and 100 boys. I think they have enough problems of their own. We’ll move on, tomorrow.”

He couldn’t say for sure whether the grunts that followed his statement were of assent or dissent.

Stripes of mirrors of stripes indeed! What’ll happen if we added parity to the mix?

Takeaways:

- **RAID 0+1 or RAID 01** – A bunch of RAID 0 arrays mirrored (in RAID 1).
- **RAID 1+0 or RAID 10** – A bunch of RAID 1 arrays striped (in RAID 0).
- The most common notation for RAID 0+1 is ‘RAID 0+1’, and for RAID 1+0 is ‘RAID 10’.

Chapter 10:

RAID 2, 3, and 4

Nobody except the head detective showed up the next day.

I'll take it easy today, he thought. "How many more RAID levels are there?" he asked the OS.

"More than you can count, detective."

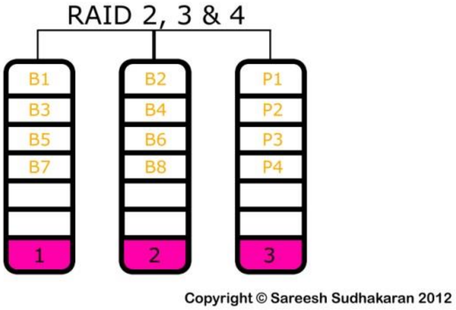
"I'm going to ask you again. If I don't get the right answer this time I'm going to shut you down."

"Okay, please, don't shut me down or restart me. There are many RAID levels, really."

"I want to take it easy today."

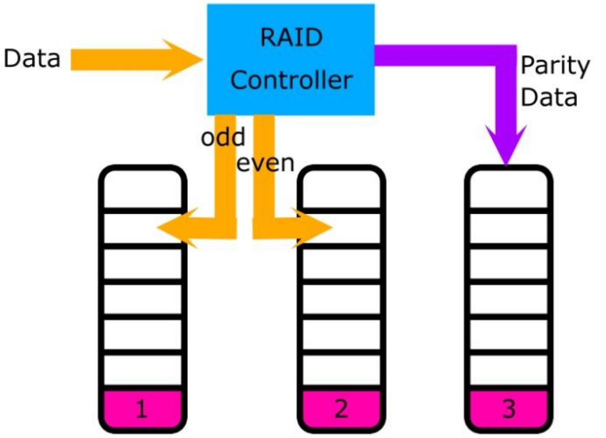
"Say no more. I think you should check out RAID 2, 3 and 4. You could finish by lunch, no sweat."

The detective wasn't convinced by the OS' answer but what choice did he have? He trudged all the way to RAID 2. He wasn't prepared for what he saw next. RAIDs 2, 3 and 4 all looked the same from the outside!



Copyright © Sareesh Sudhakaran 2012

This is how data flows:



Copyright © Sareesh Sudhakaran 2012

We've seen earlier that data is written in blocks. However, there are situations where you'd want data to be written in bits (the smallest possible) and bytes (8 bits, the standard word size).

RAID 2 lets you write in bits. In fact, it lets you stripe in bits – each subsequent bit is written on a separate drive. Then, to keep its redundancy tag intact, it adds parity.

Where are the parity bits written?

On a separate drive. This is why RAID 2 is called bit-level striping with dedicated parity. Parity data is on a dedicated drive.

If RAID 2 has bit-level striping, then RAID 3 has *byte*-level striping. Like RAID 2, RAID 3 also has a dedicated parity drive, and parity isn’t mixed with the main information.

You know what's coming next.

RAID 4 is *block*-level striping with dedicated parity. All parity data is written to a single drive.

Because the parity drive is written to a single drive the overall write performance will depend directly on how good the write performance of this drive is. This could cause bottlenecks.

This is the main complication with parity anyway. Parity data needs to be calculated while new data is being added. If a drive fails, the information is rebuilt again. These calculations are taken care of by the controller, but it adds to the overhead.

RAID 2 and RAID 3 are almost extinct. RAID 4 is very rare. Why?

Because there is a way to spread the parity data to multiple disks, eliminating many of the bottleneck problems associated with RAIDs 2, 3 and 4. Good to know neither of these were involved in the 'murder'.

Lunch time! The detective though it was best to put it off until tomorrow. At least he was ahead of the curve, and in the home stretch. Or so he thought.

Takeaways:

- **RAID 2** – Bit-level striping with parity.
- **RAID 3** – Byte-level striping with parity.
- **RAID 4** – Block-level striping with parity.

Chapter 11:

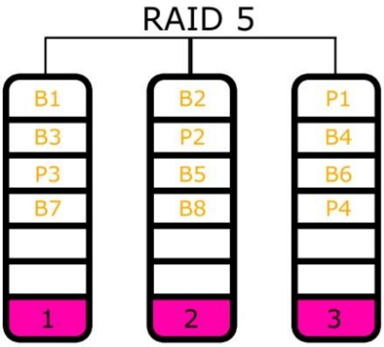
RAID 5, 6, and 50

All the detectives showed up on time the next day, each hoping their boss had finished ‘the dirty work’ the day before.

The head detective had other plans.

“Boys, today we will look at the most complicated RAID setups ever – RAIDs 5 and 6. I’m sensing we are close to finding the culprit, and we have no time to lose. We’re not stopping for anything today – food or drink – until we find the guy who did it!”

This time he didn’t care to evaluate the loud grunts that followed. RAID 5 looks like this:



Copyright © Sareesh Sudhakaran 2012

It is very similar to RAID 4, except that parity data is distributed over multiple disks. For this reason it is also called Block-level striping with distributed parity. RAID 5 can take one drive failure only. When that one acceptable 'death' happens, performance is affected because data must be calculated from parity.

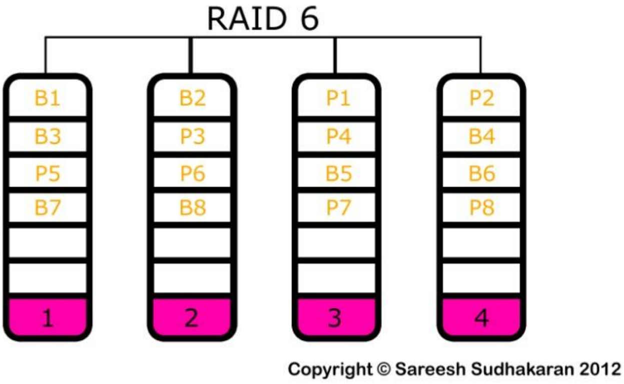
Why would anyone want to use a RAID setup so complicated? RAID 5 has a unique advantage over RAID 10 and RAID 1 - it offers better capacity - called Space Efficiency. If four drives of 1 TB each are added in a RAID 5 array, the capacity achieved is 3 TB, with a space efficiency of 75%. The same number of drives in RAID 1 will give 1 TB, and in RAID 10 will give 2 TB.

Some people consider this an acceptable compromise. RAID 5 is also widely supported, like RAID 0 and RAID 1.

RAID 5 offers excellent read speeds, similar to RAID 0, which is why it is very popular for servers that stream media. Write performance is a big question mark, with nowhere near the performance of other RAID systems. This is mainly due to the parity calculations that need to be performed and written every time data is added, changed or deleted.

RAID 5 is what people opt for when RAID 1 isn't enough and a better space efficiency is needed. The flip side is that, by adding more drives, one increases the chances of multiple drive failures, which is the death of a RAID 5 array.

What about RAID 6? This is what it looks like:



Copyright © Sareesh Sudhakaran 2012

If RAID 5 needs a minimum of 3 drives, RAID 6 needs a minimum of 4 and it's called Block-level striping with *double* distributed parity. What's that?

RAID 6 gives us an additional drive failure when compared to RAID 5.

The maximum number of drives that can fail without data loss in an array is called its Fault Tolerance. RAID 5 has a fault tolerance of 1. RAID 6 has a fault tolerance of 2.

Double-parity gives the Controller additional time to rebuild the array without the data being lost. It does this by doubling the parity information to different drives, thereby almost halving the risk. But because of this, it needs an extra drive compared to RAID 5.

In our 4 TB example, a RAID 6 would give us only 2 TB. However, as the number of drives increase, the difference between RAID 5 and RAID 6 as far as space efficiency is concerned decreases, and almost becomes negligible. The read speeds are similar to RAID 5, while the write speeds are slightly worse, the difference which again reduces as the number of drives increases.

RAID 6 is used for reducing the risk factor, with other requirements similar to a RAID 5 array. The rule of thumb (it's a flimsy rule, really) is to use RAID 5 for smaller arrays and RAID 6 for larger arrays - but there are exceptions.

"What do you think, boys?" asked the head detective. "I'm of the opinion that none of these guys are guilty."

"We feel the same, sir."

"I think it's time to talk to the OS again."

The OS wasn't happy at all to see the detectives back at his office.

"What can I do for you gentlemen?"

"Mr. Operating System, we've been around to all the small RAID arrays you have here at Hotel Network, but we feel there's something you're not telling us."

"I haven't kept anything from you."

"Yes, you have! What is the RAID level used for your…your personal storage?"

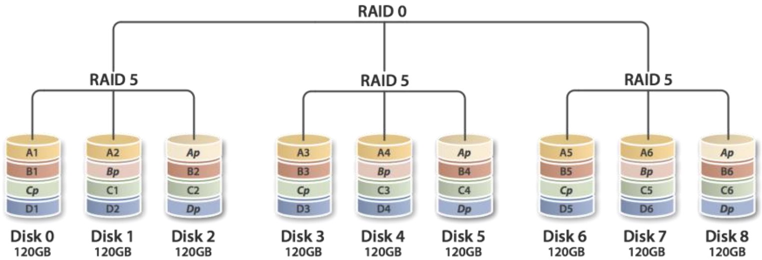
“I don’t know what you’re talking about.”

“Pull the plug!”

“No, wait! You mean, my NAS?”

The detective was too tired to ask what a NAS was. “Yes, your NAS”.

"Our NAS is on RAID 50."



Author: Kauberry

A RAID 50 combines the striping of RAID 0 with the distributed parity of RAID 5. It needs at least 6 drives.

Each RAID 5 set can only tolerate one drive failure, it is common to have RAID 50 systems with 'hot spares' ready to take over when a drive fails. This is how Hotel Network continued running when the drive died.

RAID 50 is RAID 5 with better write performance and fault tolerance. It is the recommended solution for arrays that need a high fault tolerance without sacrificing space efficiency. This is at the expense of rebuilding time, which takes quite some time.

The head detective heaved a sigh of relief. The case was finally cracked. Unfortunately, there was no murderer.

"So how did our hard drive die?"

The OS replied without batting an eye lid: "Natural causes."

“We should take his NAS back to the station!”

The head detective thundered: "Natural causes! Wasn't he overworked? Didn't the stress of performance do him in?"

The OS hung his head in shame.

"Shut down this joint."

"But-"

"Only for today. Let them learn their lesson. Think of it as respectful silence for the duly departed. You may reboot tomorrow."

Before shutting down Hotel Network for the day, though, all the detectives had another round of sodas. They hoped they didn't have to come back to his hotel ever again, unless it was for free food and booze.

Takeaways:

- **Space Efficiency** – the amount of usable space you actually get, as a percentage of the total capacity of the drives used.
- **Fault Tolerance** – the maximum number of drives that can fail without interrupting work.
- **RAID 5** – Block-level striping, but with distributed parity. It has better space efficiency than RAID 1, 10 or RAID 6.
- **RAID 6** – Block-level striping with double distributed parity. It has better fault tolerance than RAID 5.
- **RAID 50** – A group of RAID 5 arrays striped.

Chapter 12:

Do you need RAID?

By now, this primer should have helped you understand what RAID is, *theoretically*. However, you're probably not at the point where you can confidently choose the right RAID for your project.

Even though this guide doesn't presume to go into detail on the intricacies of RAID, it wouldn't be of much use unless there was some practical advice to at least get you started. From here on, we are going to deal with the practicalities of setting up storages for your post production needs.

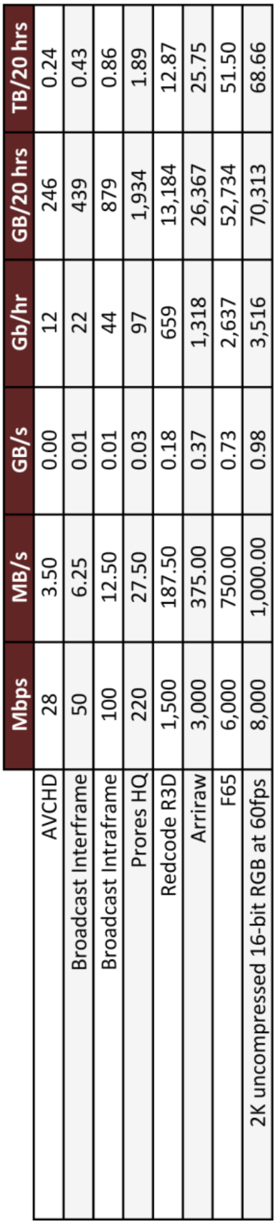
There are two big questions that need to be answered:

1. Do I need RAID?
2. Which RAID is right for my project?

It's one thing to throw around theoretical numbers on paper, and another to achieve these numbers in the real world.

In the real world, working in RAID is like having an alligator for a pet. Your undivided attention, focus, patience and tact are mandatory.

If you take it for granted, you're shooting yourself in the foot.



	Mbps	MB/s	GB/s	Gb/hr	GB/20 hrs	TB/20 hrs
AVCHD	28	3.50	0.00	12	246	0.24
Broadcast Interframe	50	6.25	0.01	22	439	0.43
Broadcast Intraframe	100	12.50	0.01	44	879	0.86
Prores HQ	220	27.50	0.03	97	1,934	1.89
Redcode R3D	1,500	187.50	0.18	659	13,184	12.87
Arriraw	3,000	375.00	0.37	1,318	26,367	25.75
F65	6,000	750.00	0.73	2,637	52,734	51.50
2K uncompressed 16-bit RGB at 60fps	8,000	1,000.00	0.98	3,516	70,313	68.66

Sometimes you feel you might need RAID just because everyone else is shouting its name from his or her roof tops. Not so. You will be surprised to see how far simplicity goes.

As we have seen earlier, the first step is to estimate how much footage you are going to work with. How much space will all your source footage take?

Take a look at the chart on the left to get an idea. E.g., working with Prores HQ, a full project with about 20 hours of material will need only 2 TB. The data rate for one stream is 220 Mbps, or 27.5 MB/s. A typical consumer-grade 7,200 rpm hard drive averages over 50 MB/s, and tops out at about 100 MB/s on average.

So, if you have two drives – 2 TB each, you can use them in RAID 1 to get a theoretical maximum of 200 MB/s read speed – or 7 streams of Prores HQ, and 100 MB/s write speed, or 3 streams.

That's *if* you needed that kind of data transfer rates. Most people don't. Most professionals who shoot sub $20,000 dollar cameras shoot in the internal codec (which corresponds to the broadcast codecs above) or via an external recorder to Prores or DNxHD 220 Mbps. This footage is edited natively and finished for Television, Blu-ray, DVD or the Internet.

Simplicity of workflow is paramount. Budgets are tight and it is important to get to the end product as fast as possible.

What if we have 2 drives of 2 TB each, one with footage and the other as backup, not in RAID? For renders or writes another drive is used, maybe 1 TB or so. For cache/page/temp files, one uses a small 120 GB SSD drive, or a 128 GB CF Card in a laptop slot! Finally, for the OS and Software, one uses another 120 GB SSD drive.

The simplicity of dividing read and write drives in this way is that you are not clogging a particular transport pipe. You are eliminating bottlenecks. Remember, a drive can only do one thing at a time – so why not let it do one thing well?

Five relatively cheap drives (or more if you need additional backups) on a simple SATA II interface can be reused for every subsequent project. So, where's the need for a RAID at all?

But – what if you were *compelled* to use RAID, simply because you have many streams of data to transfer? You might have a network set up in a small facility with a few editors, effects artists and graders, etc.

Calculate how many people (or streams) will be reading from the source footage at any given time. Multiply that by the data rate. Is this value higher than the average transfer rate of your hard drive or network? Then you might need RAID.

You see, even if tomorrow someone invents a super fast hard disk array for $10, one can still use RAID to *further increase* speed and provide redundancy. Enough is never enough. This technology isn't going anywhere soon!

Okay, maybe you need speed and maybe you don't. The more important reason why one might need RAID is redundancy.

We have already seen that RAID is used so that one can continue working even if a drive fails. Now's the time to calculate whether you can tolerate a drive failure, and if yes, how fast can you get the data from your backup drive to continue working?

Google reports that 3% of drives fail each year in the first 3 years of life. Then, this percentage increases each subsequent year. The older the drive the higher its chances of failure.

In most scenarios, it would be wise to peg this figure at 5%. Less is good luck. More is definitely unusual and cause to re-evaluate the drive or the setup or both.

Video producers don't have to worry about long term drive failure. A feature film has a maximum data life span of 2 years, and most productions are under the 1 year mark. We are talking about building redundancy for work, not archival. For true archival, other solutions exist.

Today's consumer hard drives are good enough for most video production work. When it isn't, that's a good indication it's time to use RAID.

When working in a multi-user environment, it is important to keep everyone busy. There's no time to hunt for that backup drive and rebuild or relink your data – not when clients are waiting impatiently, hopefully with a check in their hands.

In this scenario, having redundancy makes perfect sense.

However, you must understand that redundancy is not backup. Having footage on one system ensures that everyone is 'on the same page' and working with the same material. On the flip side, if you don't manage the networking and permissions well, someone might accidentally write over your valuable data. The biggest cause of data loss is human error. But everyone likes to blame hardware.

This happens even if you're the only user. We all hit delete by mistake, or forget to save work, or whatever. Redundancy will not help you when this happens. Redundant data in RAID is always the *most latest* data. Therefore, when considering RAID, do not ignore the fact that you still need to account for backup.

That's what I meant when I said having a RAID is like having an alligator for a pet. It's an additional burden, a new task to learn and manage over and above everything else.

What you're doing is estimating your 'chances'. How much can you get away with? Sometimes the answer is very clear. Sometimes it isn't. It's balancing trade-offs between fault tolerance, cost and performance.

Ask yourself these questions:

- Are you sure your bottlenecks are caused by your hard drive system and not any other component?
- What is the size and speed of your workflow?
- Can single drives performing one and only one function fulfill your speed requirements?
- Are you a one or two-person facility that can quickly pick up work if a drive fails?
- Can you handle the additional burden of running and troubleshooting a RAID?

I hope I have given you enough ammunition to at least find out whether or not you need RAID. In the next chapter we'll take a look at the hardware needed to get a RAID up and running.

Takeaways:

- RAID is NOT backup.
- For shooting H.264, Prores HQ, AVCHD, etc., you don't need the speed advantages of RAID. A regular hard drive will do.
- If keeping spare copies can circumvent redundancy, and the time delay is acceptable, RAID is not worth it.
- If redundancy is critical, then RAID is a necessity.
- If you need blistering speeds for large quantities of data, something SSDs cannot do without getting overly expensive, then RAID is your best bet.
- If you are not sure how to choose, then read Chapter 14.

Chapter 13:

Things you Need for a RAID System

Okay, you have decided that you want RAID.

The first step is to decide who will control your RAID array. Why, the RAID Controller, of course!

There are four broad kinds:

1. Software RAID
2. Motherboard RAID
3. Fake RAID (or Host RAID)
4. Hardware RAID

Software RAID

Software RAID means you don't need a hardware controller. Your operating system controls your RAID array.

Operating Systems are capable of simple RAID configurations, and are surprisingly fast and efficient. Here's a list of recommendations of various operating systems with the RAID levels they support*:

- Windows 7 Professional and Ultimate: JBOD, RAID 0 and RAID 1
- Windows Server: JBOD, RAID 0, RAID 1 and RAID 5
- Linux with the mdadm controller: RAID 0, RAID 1, RAID 4, RAID 5, RAID 6, RAID 10 (1+0)
- Mac OS X and Server: RAID 0, RAID 1 and RAID 10

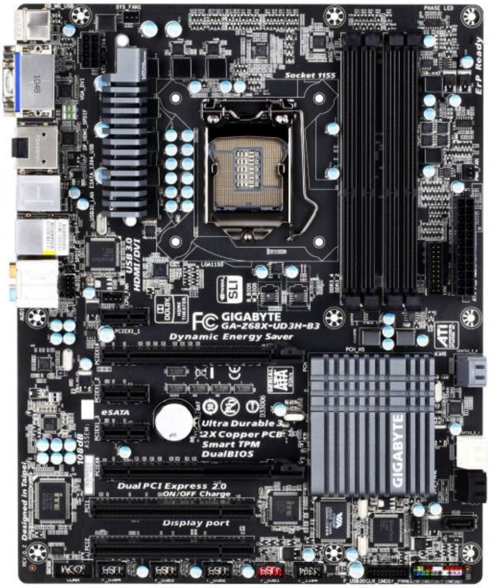
**This is as far as I know. Please check the manufacturer's website for exact compatibility.*

Software RAID works well for non-parity RAIDs like RAID 0, RAID 1 and RAID 10, etc.

Using RAID 5, e.g., when supported, will tax the CPU to make all those heavy parity calculations, and can impact the performance of other software running on the same machine. Modern CPUs have really reduced this 'problem', but whether or not this impact is acceptable is for you to decide. As a rule of thumb I would delegate all parity RAID systems to a dedicated controller.

If I had to choose the most robust operating system for RAID, with the best support, I'd choose Linux. And it's free! The sad part is, most video professionals stick to Windows or OS X.

Motherboard RAID



Modern motherboards come with many bells and whistles. For better or for worse, they try to incorporate bits of the CPU, OS, Hard Drive, Memory, etc., to increase performance. So why not RAID as well?

Higher-end consumer grade motherboards like the ones made by Gigabyte and Asus sometimes have on-board RAID controllers. Most of the time, the levels are restricted to RAID 0, RAID 1, RAID 5 and/or RAID 10. Be very careful while selecting a motherboard for RAID. Not all RAID levels 'supported' will work the way you intended to, with the Operating system you use.

The big risk with motherboard RAIDs are that if your motherboard dies, you might have to find the exact same model, or at least a compatible motherboard that also somehow takes all your other hardware. As anyone who has replaced quite a few motherboards will know, this is a scary thought. Why?

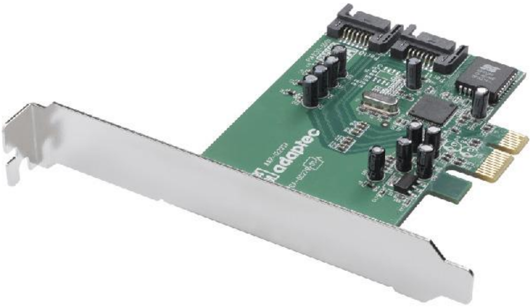
Of all the components in a computer, the motherboard tends to be the one with the smallest life cycle. If an OS controlled RAID crashes, you can re-install the same Operating System and continue working. It's not that simple if a motherboard dies at the end of its life cycle. Motherboards, especially for laptops, are hard to find once the models become obsolete.

I wouldn't recommend motherboard RAIDs with today's technology for consumer-grade computers. However, if you're running servers, and you only need a simple RAID setup, a motherboard by a good manufacturer like Supermicro might be worth it. Server motherboards tend to have a greater shelf life.

As a rule of thumb though, my advice is to stay away from motherboard controlled RAID arrays.

Fake RAID

What is a 'fake' RAID? A fake RAID or Host RAID is a hardware 'card' that you stick to your motherboard. Having done that, you might think it is a hardware RAID card, but it isn't.



These are cheap cards that do not contain a dedicated RAID controller chip. Instead they just have a hard drive controller with special firmware and drivers (software).

The burden of RAID processing lies on the CPU, just as in the case of software RAID. This is why it is called ‘Fake’ RAID. Adaptec calls it Host RAID. Usually, these cards only support JBOD, RAID 0 and RAID 1.

The word ‘fake’ isn’t really a fair description. A motherboard RAID is also a kind of ‘fake’ RAID.

You might be thinking, if this card is just software, and the CPU is doing all the work, then what the hell use is it?

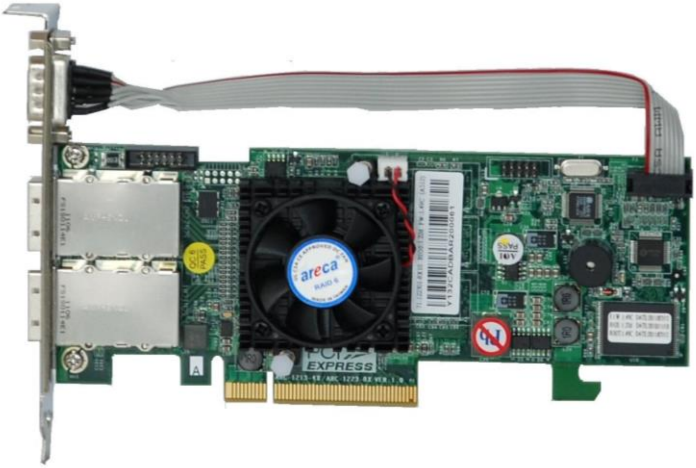
It has one exceptional use: You can change your motherboard and your OS, and still keep your drives in RAID. These cards have an even longer shelf life than software or motherboards. It’s another step up, for those who can’t afford (or don’t need the features of) a hardware RAID controller card.

I recommend this solution over motherboard RAID any day, thought its efficacy over software RAID is debatable. The company that I recommend for this is Adaptec, and their solution is called Host RAID. The software inside these little beauties is the most important part; so don’t just think of them as adapters. When looking for a fake RAID solution, check for compatibility with your motherboard, OS and drives.

Hardware RAID

A hardware RAID is a true RAID Controller that does all the heavy lifting, or at least should. They are mini-computers capable of making their own decisions. Therefore, you shouldn’t be surprised that sometimes they cost as much as a computer!

Hardware Controllers are almost always used in mission critical systems. You will see a familiar trend here – that of using specialized components for each separate task. This allows you to change other components or software while continuing to work – the whole point of RAID.



The cost of hardware controllers are governed by how many hard drives it can handle, the protocol (SAS over SATA, etc.) and what kind of RAID capability it has. The 'best' cards can take many drives and might support many practical RAID levels. Good controllers also come with good software that make installation, maintenance and configurations easy.

The two manufacturers I recommend for this are Areca and Adaptec.

Not everything in the hardware RAID world is hunky dory. These are proprietary pieces of gear, and if you suddenly decide one manufacturer isn't your cup of tea, you can't move to another without a complete restart from scratch.

On the positive side, these controllers are built to take into account many challenging scenarios in which lesser RAID mortals fail miserably. They usually have caches that store pending writes in case of a power failure. They also guarantee "as-advertised" speeds in real world performance, and offer years of trouble-free use.

How it works out for you, though, is dependent on how you plan to use it. Your mileage may vary.

Once you’ve selected the right type of controller, you are ready to put together your RAID system. Let’s look at the rest of the parts.

Platter Hard Drives

The ‘I’ in RAID started out as ‘Inexpensive’. Manufacturers quickly got around to changing that to ‘independent’, but we all wish otherwise. The cheapest drives for RAID, especially for large volumes of data, is the simple 7,200 rpm (or 5,400 rpm) Hard disk drive, commonly also called a platter or spinning drive.

I recommend the following manufacturers: Hitachi, Western Digital and Seagate.

You might hear many people harping the merits, resilience or performance of one brand over another. After having used all of these manufacturers over the last 17 years, I can tell you that by the time you buy and setup your RAID array, another manufacturer will have forged ahead in the rat race.

These three manufacturers have been consistently reliable, and at the time of this writing, even though my personal favorite is Western Digital, the most recommended drive for RAID considering longevity is from Hitachi, and the fastest, Seagate. Go figure.

SSD Drives

SSD drives are the future (maybe). They are faster and are supposed to last longer than platter drives because they don’t have moving parts. They are also cooler and need less power. Price aside, they are perfect.

SSDs come in two flavors:

1. PCI based, or
2. SATA/SAS based.

For SATA/SAS, I recommend Intel or OCZ drives.

PCI based drives are much faster and proportionately more expensive. The manufacturer I recommend for PCI based SSD drives is OCZ.

PCI based SSDs make excellent temp/cache/page drives for really fast reads and writes. Also, be careful when trying to build a RAID array with SSD drives – these drives have their own quirks.

Enclosures

One or two drive RAID arrays can be placed inside a PC case. However, external enclosures are the most used method of storing large RAID arrays. There are two ways you can go: bare boxes and boxes with backplanes.

For simple enclosures, I recommend Icy Dock and Lian Li.

For server style rack enclosures that need backplanes, I recommend Norco and Supermicro.

Remember, these systems need power supplies, and possibly additional drive ports depending on the number of drives you have in the enclosure.

We are not finished yet! A good RAID array needs one important device that most people ignore, and pay the price.

UPS

I mentioned earlier that human error was the biggest cause of data loss. The second biggest is power failure. A UPS (Uninterrupted Power Supply) gives you that extra time that can turn out to be a lifesaver.

There’s only one brand I will recommend here, and that is APC.

If you don’t want the hassle of building your own RAID array, you can always buy a ready-made system, which we’ll cover in subsequent chapters.

Takeaways:

- The basic parts of a RAID array are:
    - The Controller (Software, Motherboard, Fake or Hardware)
    - Drives (Platter or SSD)
    - Enclosure (PC Tower or special enclosure)
    - Connection (SATA, SAS, Thunderbolt, USB, Ethernet, etc.)
    - UPS
- Whenever possible, opt for a hardware-based RAID controller (the most expensive kind).
- If you can’t afford a RAID controller, choose software RAID.

Chapter 14:

Which is the Best RAID Level for Video Editing and Post Production?

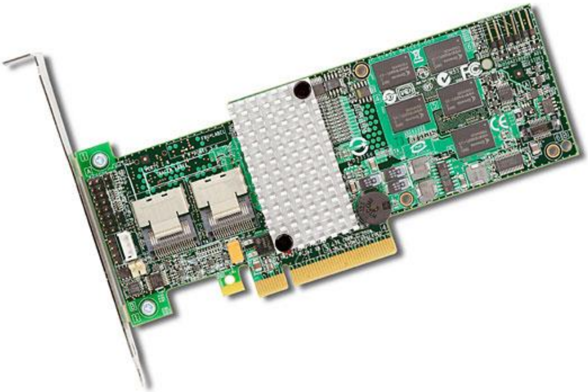
There are three ways to make a decision on the best RAID level for your video editing and post production needs:

1. Gut feeling or instinct – also called an impulse buy.
2. Years of first-hand experience running various RAID levels and systems.
3. Analyzing available data to find the best odds for each RAID level.

The first-time RAID buyer should attempt to understand the complexity that goes into selecting the 'right' option. Something that looks good today will prove totally inadequate tomorrow. Those who cannot fathom or don't want to deal with the complexity will rely on opinions or their gut feeling (If you read too many third-party opinions you'll be more confused than ever!). You could get lucky or unlucky. The sad part is you don't have a say in the matter either way.

If you have years of first-hand experience you wouldn't be looking for an answer anyway. You're already convinced about what's best for you. This chapter explores the last option.

The analysis will focus on the small business post facility or single video editor.



The RAID levels that I'll be looking at are 0, 1, 5, 6 and 10.

These are not the only available RAID levels. As you know, there are nested RAID levels (50, 100, etc.) and proprietary/non-standard RAID levels (Z, Diff-RAID, etc.) that I won't look at. The choices are complex as it is. The smart thing to do is begin with the easy ones.

Do you need Redundancy?

The biggest benefit that RAID brings to the table is 'Redundancy' – in short, it is *the ability of the system to let you continue working without delays even if there's one or more drive failures.*

Do you need redundancy? Imagine this situation:

You are editing on one 4 TB 7,200 rpm drive that sits in your computer or laptop. You have a backup on an external drive somewhere. Suddenly, your 4 TB drive fails or is erased or whatever. Here are the sequence of events that must take place to get you back to 'as you were':

- Format your drive (4 TB drives can take many hours, even a whole day). If it's dead, you will replace it (Order and delivery means it'll take days).
- Connect your external backup to your computer via USB 2.0, 3.0, eSATA or Thunderbolt (Fifteen minutes, if the drive is nearby. Maybe a day or more if the drive is in another location).
- Copy your footage to the formatted or new drive. To copy it via USB 2.0, it'll take about 6 hours per terabyte. Via USB 3.0, Thunderbolt 1 or eSATA, it'll take about 3 hours per terabyte.
- Restart your NLE and hope everything links automatically (Ten minutes if everything goes well).

If your total data size is low, you might experience the loss of half a day. If your data size is significant, you will lose at least two days. If your drive dies, you might lose a week.

Of course, the smarter way to work is by having two 4TB internal drives. This way, if one crashes or dies, you can link to the other one and continue working (assuming you were alert enough to duplicate your data beforehand). The downtime in this case is about an hour, not more. However, not everybody has space for two internal drives.

Can you afford to take this loss in productivity? Run the numbers based on your own data needs. If you are okay with the down time, then you don’t need RAID for redundancy.

The ideal drive size for a small business post facility or editor

There are many variables that go into estimating the right RAID requirements. But, there are three factors you’ll need to know right at the beginning:

- What is the total size of your data?
- What is the data rate you’ll be needing on a regular basis?
- How many layers do you typically have on your timeline overlapping each other?

The first determines the total capacity of your RAID array. The market forces you to choose the size of one drive. From the two numbers, you will arrive at the total number of drives needed to make up your required capacity. This, and the second and third factors will tell you how fast it has to be.

Working backwards to find your budget

Among all the available RAID levels that give you the capacity and speed you need, find both the cheapest and the most expensive. Does your budget lie somewhere in between?

In either case, you are limited to drive sizes that cap at 4 TB. If speed is critical, you will want to invest in 7,200 rpm drives (though 5,400 rpm or similar drives should work okay too). The cost of a 4 TB 7,200 rpm is about $300 or more (or whatever the price is). More drives means a linear increase in the total price. There’s no way of getting around it (unless you are buying hundreds of drives).

You are also limited by the connectivity options, so you cannot go about multiplying your speed. E.g., what’s the point in designing an 8-bay RAID 0 array filled with SSDs that can read at 500 MB/s. The read speed you'll get from such a beast is about 30 Gbps. There isn't any cheap external connectivity option that delivers data at this rate. The closest I can think of is 32G Fiber Channel.

See? You can't aim high, and you can't live with the low. It's like walking into a large cave filled with treasure. There's only so much you can carry in one trip. Finding the right RAID solution is a balancing act.

Every RAID solution offers speed. The more drives you add, the faster your array becomes. Many people automatically assume they need RAID 0 if speed is their goal. That's rubbish. As I've mentioned above, you cannot multiply drives for speed without limits (just like speed limits on our roads). That means, there are options where you can have your redundancy as well as your speed – all under budget.

Which speed is the most important – write or read?

Footage recorded on set is sacrosanct. You would almost never write over it. You will only read from it.

While editing, you might render out certain scenes with effects. Or, you might create motion graphics or 3D elements that are baked in as your edit proceeds. This means, there is some writing that happens. One cannot categorically say that you'll never write to your array. You might.

Let's see how this affects speed. There are three major data rate 'ranges' in video today:

- 50 Mbps (DSLRs, Canon C300, etc.)
- 220 Mbps (Prores HQ, DNxHD 220)
- 150 MB/s (R3D, Cinema DNG from the BMCC, etc.)

Here's the data rate (in MB/s) for the number of streams you need to read at one go:

Streams	3	5	10
50 Mbps	18.75	31.25	62.5
220 Mbps	82.5	137.5	275
150 MB/s	450	750	1500

As you can see, at 50 Mbps, you can easily read 10 streams of 1080p content even with one 7,200 rpm drive (they deliver 100 MB/s or better). However, reading even one stream of R3D footage in full quality is tough if not impossible.

Writing offers the same problems. Leaving aside the time it'll take your CPU or GPU to compute the new pixels, writing to the same format will essentially mean you need the same speeds, maybe more. Why more? If you're writing to the same array being used to read the source material, your speeds will be effectively cut down by a minimum of half. E.g., if your editing timeline has four streams of footage, and you're reading from them and rendering the composited version at the same time, your array will need enough bandwidth to manage 5 (4 read + 1 write) streams of data. That means, whatever your desired write speed is, multiply it by 5! If it's one stream per read and one stream per write, multiply it by 2.

You see, because you are always reading from your RAID array, your RAID array will have to work extra hard to also write at the same time. For this simple reason, I strongly advise against writing to a RAID array (or even a single drive) while you're working. The best way to manage this is to perform your renders at the end of the day or a break and come back when it's done. If this isn't workable, get two arrays.

Reading is the primary activity of a RAID array containing source material for editing. Reading is more important, and therefore must take absolute priority. If anything interferes with this (even if that's writing) then that's a bad thing.

Let's look at it from another angle. You might read multiple streams at the same time (that's the nature of editing), but you'll hardly ever write more than one stream at the same time. Most batch rendering applications perform one render at a time. You can safely say that no matter what you do, it'll be safe to design a RAID array in this way:

- Desired read speed = Number of read streams x data rate. It would be very rare for the number of streams to exceed 5.
- Desired write speed = data rate of rendered format.

What does that tell you? The read speed is *at least* twice as important as the write speed. On average, it is three times more important.

Read and Write speeds of various RAID levels

Let's look at the theoretical read and write speeds offered by various RAID levels.

Notes: I'm assuming the following:

- Drive speed (read or write) is 100 MB/s, though you have faster platter drives.
- Drive size is 4 TB.
- Price of a 4 TB drive is $300.
- Failure rate of a 4 TB drive is 5% (You buy a 100 and five will fail within the first year).

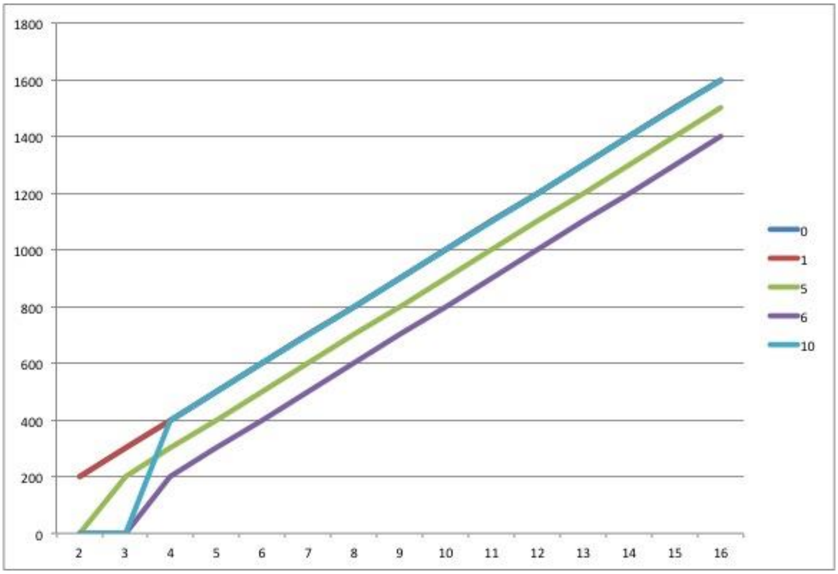
On the left is the number of drives in the array. Speeds are in MB/s. An 'x' means – RAID 5 needs a minimum of 3 drives. RAID 6 and 10 needs a minimum of 4 drives. RAID 10 needs an even number of drives.

Read speeds

			Read Speed (MB/s)		
Drives	**RAID 0**	**RAID 1**	**RAID 5**	**RAID 6**	**RAID 10**
2	200	200	x	x	x
3	300	300	200	x	x
4	400	400	300	200	400
5	500	500	400	300	x
6	600	600	500	400	600
7	700	700	600	500	x
8	800	800	700	600	800
9	900	900	800	700	x
10	1000	1000	900	800	1000

11	1100	1100	1000	900	x
12	1200	1200	1100	1000	1200
13	1300	1300	1200	1100	x
14	1400	1400	1300	1200	1400
15	1500	1500	1400	1300	x
16	1600	1600	1500	1400	1600
32	3200	3200	3100	3000	3200
64	6400	6400	6300	6200	6400

If looking at the table is giving you headaches, here’s a simple graph of the same table:



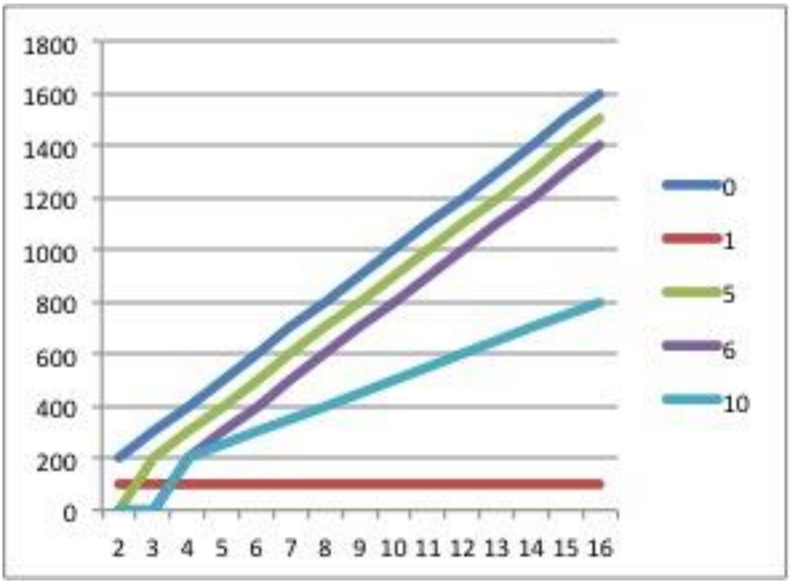
It doesn’t take a rocket scientist to realize that all RAID levels offer a linear increase in read speed as you increase the number of drives. RAID 10 and RAID 1 are as fast as RAID 0. RAID 5 and 6 are no slouches either.

See? Speed does not categorically mean RAID 0. Get that out of your head.

Write speeds

Write speeds are a bit weirder. From this point on we enter the twilight zone. Brace yourselves.
Most drives offer a slightly lower write speed when compared to read speeds. It's not because of any physical thing but simply because the drive needs to find the right spot to write data in before it can write. This additional calculation makes it slower than reading. For simplicity's sake, I'm going to assume that the write speed is equal to the read speed. When you make calculations in the real world, you'll never be at the limit. You must always account for some wastage or loss. I'm assuming your total data rate is well below both the read and write limits of your drive.

So, if we assume the write speed is the same as the read speed, then, in an ideal world, the write graph should look like this:



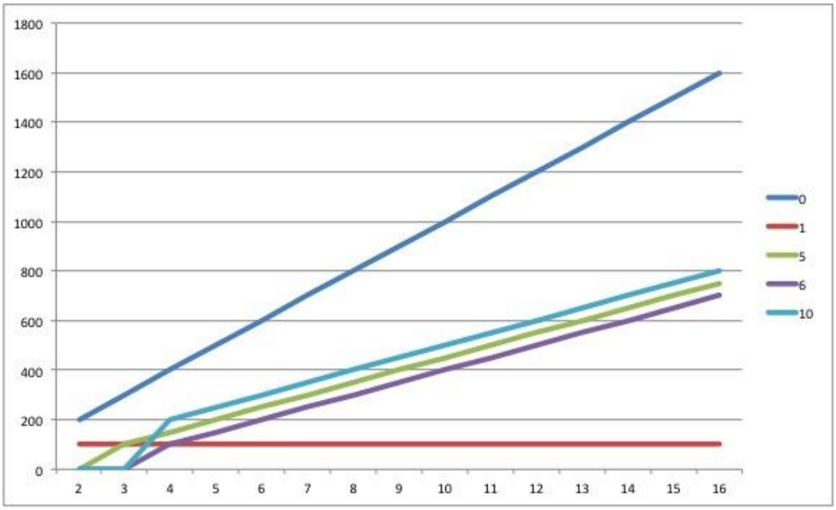
However, this doesn't happen in practice. For RAID 5 and 6, data isn't duplicated. Instead, parity data is calculated and written alongside the actual write data. E.g., if your write data rate is 100 MB/s, in reality a RAID 5 or RAID 6 array will write 100+x MB/s (x is parity). It gets even more complicated. If you are overwriting data (rendering a new version of a composition or chroma key, for example) then the old parity data needs to be read, and only then will the new one be calculated. Then, the

new one will be compared to the old one to see if it needs to be changed, and then the writing takes place. Phew!
RAID 5 and 6 perform worse for sequential data (video is sequential data) because *that many* more calculations have to happen per second.

There are various factors that affect the overall speed of writing parity data. The most important is the RAID Controller that must calculate the parity data before writing it. The really good (expensive) controllers have correct caching to eliminate a lot of this time wastage. The cheaper hardware RAID controllers don't do this so well. For this reason, there is no universal formula for calculating the average write speeds for RAID 5 and RAID 6 arrays, but they'll look something like this in the best case:

Drives	**Write Speed (MB/s)**				
	RAID 0	**RAID 1**	**RAID 5**	**RAID 6**	**RAID 10**
2	200	100	x	x	x
3	300	100	100	x	x
4	400	100	150	100	200
5	500	100	200	150	x
6	600	100	250	200	300
7	700	100	300	250	x
8	800	100	350	300	400
9	900	100	400	350	x
10	1000	100	450	400	500
11	1100	100	500	450	x
12	1200	100	550	500	600
13	1300	100	600	550	x
14	1400	100	650	600	700
15	1500	100	700	650	x
16	1600	100	750	700	800
32	3200	100	1550	1500	1600
64	6400	100	3150	3100	3200

And here's the headache-free graph:



RAID 1 is good with two drives. Adding drives to a RAID 1 does not increase write speed in any way. RAID 0 is best, and every other RAID is similar (Don't forget, RAID 5 and 6 only perform this way with really good hardware RAID Controllers, otherwise the performance is going to be very poor). As a rule of thumb, you can estimate the write speeds for a RAID 5 or 6 array to be 25% or lower than the read speed. I've estimated it at 50% because that's the best that you can get.

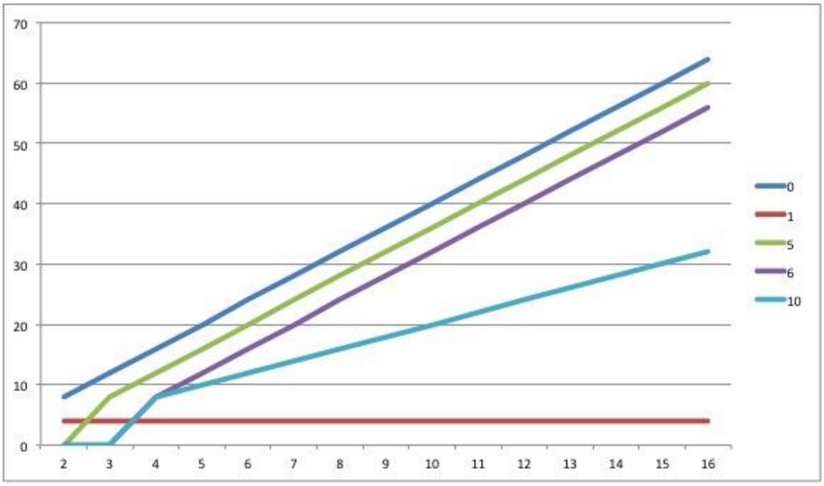
Which RAID level gives the best capacity?

Here are the various RAID levels compared on actual capacity according to the number of drives (size of one drive is 4 TB):

Drives	**Total Capacity (TB)**				
	RAID 0	**RAID 1**	**RAID 5**	**RAID 6**	**RAID 10**
2	8	4	x	x	x
3	12	4	8	x	x
4	16	4	12	8	8
5	20	4	16	12	x
6	24	4	20	16	12
7	28	4	24	20	x
8	32	4	28	24	16
9	36	4	32	28	x

10	40	4	36	32	20
11	44	4	40	36	x
12	48	4	44	40	24
13	52	4	48	44	x
14	56	4	52	48	28
15	60	4	56	52	x
16	64	4	60	56	32
32	128	4	124	120	64
64	256	4	252	248	128

And here's the graph:



RAID 1 is the biggest loser here. Once you pass two drives, you are just duplicating the same data to every additional drive. The total capacity of a RAID 1 drive is always equal to the capacity of one drive.

RAID 10 has a 50% drive penalty, because in affect it behaves like a RAID 1, had it been capable of scaling up. RAID 5 and RAID 6 do much better, because they don't have to duplicate data, but just write a few parity bits, which don't require a lot of overhead.

RAID 0 is the best, because it gives you the full capacity of each drive. Don't forget to note that if a drive is rated at X TB, the actual capacity is always a bit smaller.

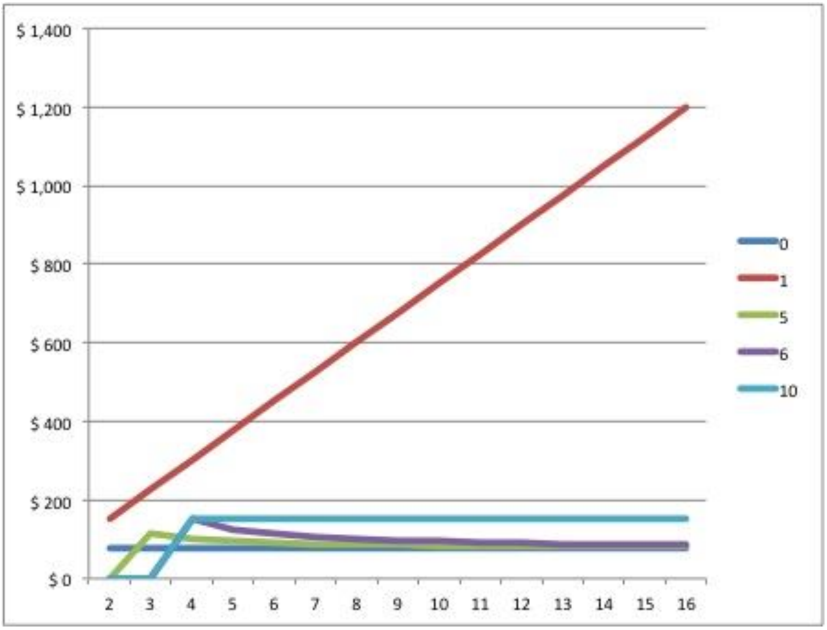
If you're short on cash and absolutely need to make the most out of every drive, the above chart tells you which options to go for. This is why RAID 5 is very popular. It gives you a lot of space, and offers you some protection (redundancy). But it comes at a price, which we'll see soon.

Which RAID level is the cheapest on a Cost per terabyte basis?

Obviously, if an array cannot use up all its capacity, then the cost per terabyte goes up. Let's see by how much:

Drives	**Cost/TB**				
	RAID 0	**RAID 1**	**RAID 5**	**RAID 6**	**RAID 10**
2	$ 75	$ 150	x	x	x
3	$ 75	$ 225	$ 113	x	x
4	$ 75	$ 300	$ 100	$ 150	$ 150
5	$ 75	$ 375	$ 94	$ 125	x
6	$ 75	$ 450	$ 90	$ 113	$ 150
7	$ 75	$ 525	$ 88	$ 105	x
8	$ 75	$ 600	$ 86	$ 100	$ 150
9	$ 75	$ 675	$ 84	$ 96	x
10	$ 75	$ 750	$ 83	$ 94	$ 150
11	$ 75	$ 825	$ 83	$ 92	x
12	$ 75	$ 900	$ 82	$ 90	$ 150
13	$ 75	$ 975	$ 81	$ 89	x
14	$ 75	$ 1,050	$ 81	$ 88	$ 150
15	$ 75	$ 1,125	$ 80	$ 87	x
16	$ 75	$ 1,200	$ 80	$ 86	$ 150
32	$ 75	$ 2,400	$ 77	$ 80	$ 150
64	$ 75	$ 4,800	$ 76	$ 77	$ 150

And here's the graph:



It is instructive to know that RAID 0 and RAID 10 have no impact on the cost per TB, no matter how many drives you buy. This means scaling up has no benefits unless the drive vendor gives you any discounts.

On the other hand, RAID 5 and RAID 6 gets cheaper as you increase the number of drives, regardless of any discounts you get for the actual drives. They do stagnate after a point, though – which is about the 32-drive mark. Interestingly, once you increase the number of drives to 64 or more, RAID 6 actually catches up with RAID 5 in terms of cost per TB; and they both catch up with RAID 0. But 64 is a large number of drives for one array, and you'll be lucky to find a controller (or stacking controllers even) with that many connectors!

RAID 1, though, is a total waste of money beyond two drives, but we knew that already.

Drive and Array failure rates for various RAID levels

There are two things to consider when looking at the failure rate of a RAID array:

1. How many drives can fail?
2. What are the chances of failure?

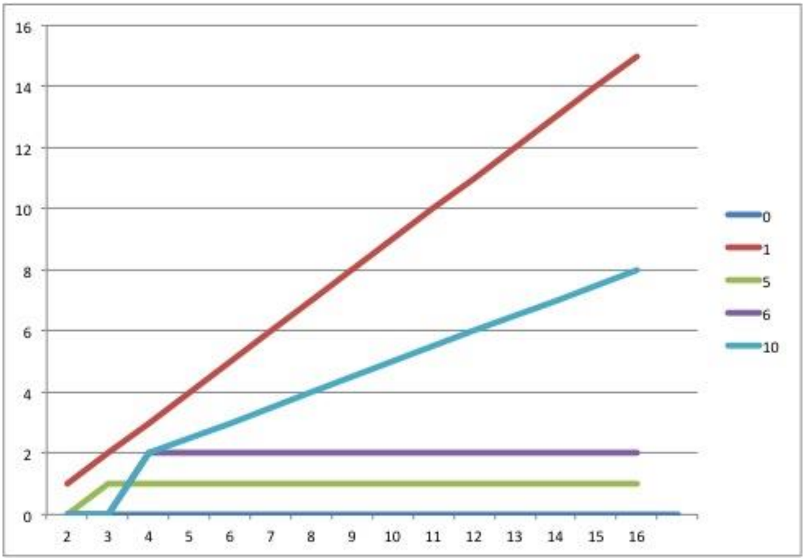
1. The number of drives that can fail for different RAID levels

Here’s how they stack up:

Drives	**How many drives can fail**				
	RAID 0	**RAID 1**	**RAID 5**	**RAID 6**	**RAID 10***
2	0	1	x	x	x
3	0	2	1	x	x
4	0	3	1	2	2
5	0	4	1	2	x
6	0	5	1	2	3
7	0	6	1	2	x
8	0	7	1	2	4
9	0	8	1	2	x
10	0	9	1	2	5
11	0	10	1	2	x
12	0	11	1	2	6
13	0	12	1	2	x
14	0	13	1	2	7
15	0	14	1	2	x
16	0	15	1	2	8
32	0	31	1	2	16
64	0	63	1	2	32

**Half the number of drives in a RAID 10 array can fail, but only one from each span. E.g., if you divide your 16-bay array into 8 groups, then only one drive per group can fail (or 8 drives total). The actual calculation of the probability of failure for RAID 10 is far more complicated.*

And here’s the graph:



Obviously, RAID 0 performs poorly. No matter how many drives you add, a RAID 0 array will not tolerate a single drive failing. RAID 1 performs best, because it just duplicates data into as many drives as you can add. RAID 6 offers better protection than RAID 5, and I guesstimate that RAID 10 offers better odds than RAID 6.

2. The failure rate of a RAID array

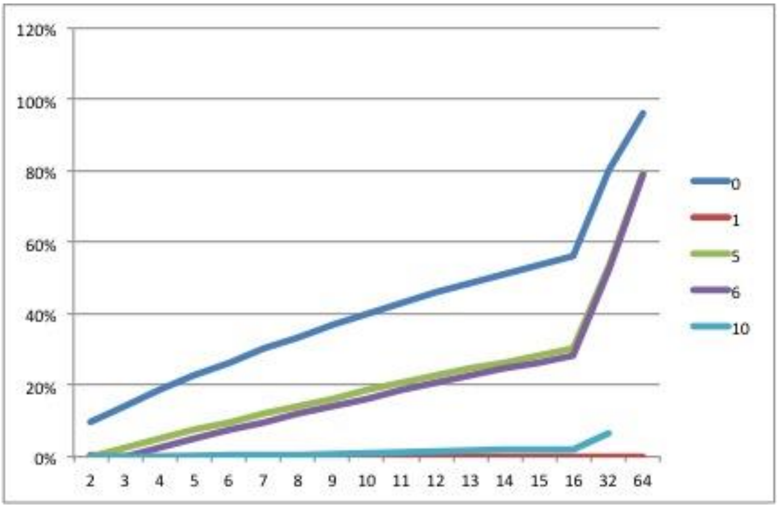
How does the number of drives affect the failure rate of an entire array? Here are the stats:

Drives	**Array failure rate**				
	RAID 0	**RAID 1**	**RAID 5**	**RAID 6**	**RAID 10****
2	10%	0.25%	x	x	x
3	14%	0.01%	3%	x	x
4	19%	0.00%	5%	3%	0.186%
5	23%	0.00%	7%	5%	x
6	26%	0.00%	10%	7%	0.397%
7	30%	0.00%	12%	10%	x
8	34%	0.00%	14%	12%	0.673%
9	37%	0.00%	16%	14%	x

10	40%	0.00%	19%	16%	1.003%
11	43%	0.00%	21%	19%	x
12	46%	0.00%	23%	21%	1.379%
13	49%	0.00%	25%	23%	x
14	51%	0.00%	26%	25%	1.793%
15	54%	0.00%	28%	26%	x
16	56%	0.00%	30%	28%	2.239%
32	81%	0.00%	54%	52%	6.450%
64	96%	0.00%	80%	79%	15.4%

***Calculating the failure rate of a RAID 10 array isn't easy. I've used a simple formula that might be totally incorrect. Each RAID 10 array is split into n/2 RAID 1 spans where n is the total number of drives. The failure rate of each span is equal to 0.25%, the failure rate of a 2-drive RAID 1 array. All these spans are striped (RAID 0), and as the number of spans increases, the chances for failure rises (just like RAID 0). The formula I've used for RAID 10 = rate of RAID 1 (0.25%) x number of spans (n/2) x rate of RAID 0 for n/2, which is the number of spans.*

Here's the graph:



Obviously, RAID 0 is dangerous even with two drives. But as you increase drives it becomes an unacceptable risk. E.g., If you have about 12 drives in RAID 0, the chance of failure is 50:50 –

that means ‘anytime’. If you have 64 drives, it’s almost a 100% – which means failure is imminent. What’s the difference? With 12 drives, you get to flip a coin. With 64 drives, you get to flip a coin, but it’s given to you by Two-Face.

RAID 1 is the simplest way to go if all you needed were two drives. Add one more to a RAID 1 array and you’ll be as safe as possible. The same applies to RAID 10 – the sheer number of drives makes failure in the real world a minimal possibility, even for smaller arrays. But it is an uncomfortable peace.

But look at RAID 5 and RAID 6. They’re both unacceptably dangerous as you increase the size of your array. Remember what I said earlier about not being able to carry out all the treasure from the cave in one go? So far RAID 5 and RAID 6 have seemed like great solutions, but not for large drive arrays. There’s always a trade-off with RAID somewhere.

It’s time for an important disclaimer:

There are many things I have not included. Small things make a big difference, so please don’t take any decision based on these calculations. The aim of this article is to help you boil down your choices to a manageable number, not make the choice for you. You are responsible for your actions.

Bringing everything down to one number
What are the things we’re worried about most when building a RAID array? Here are some important points:

- Total capacity
- Value for money
- Redundancy
- Speed

There are others, too:

- Ease of setup
- Cost and availability of hardware RAID controllers
- Number of drives and size of chassis and heating solutions

- Compatibility of hardware and software
- CPU usage
- Battery Backup for the controller
- Caching
- Buying drives from various sources to better the odds

Let’s focus on the first four for now. That offers us a theoretical starting point. After all, if a RAID level doesn’t work for us theoretically, there’s no point worrying about its technicalities. The objective is to bring each important point to a common number.

Luckily, RAID offers us simple formulas that can be scaled equally among all levels. E.g., if a drive is slow or if its failure rate is high, every RAID level suffers. If the price changes, everyone’s affected equally.

Mathematically, there are an infinite number of ways in which such numbers can be produced. I’m going for the simplest method, possibly the most error-prone, but who cares as long as it makes some sense? The two types of numbers I’m going to be reducing our data to are:

1. Factors
2. Value

1. Factors

A factor mustn’t have a unit. It’s just a number that should ideally be between 0 and +1 (1 being the best and 0 being the worst).

2. Value

You will spend X amount of money on your RAID array. But how much value do you get out of it? Obviously this is a subjective matter because each individual perceives value differently. Just for fun, I multiply the total cost of the array by the factor I’ve obtained to get the value.

Let’s see how this works in practice.

Speed factor

We have seen that on average the ability of a RAID array to read quickly is three times more important than its ability to write fast. The ‘Speed Factor’ (S) should include both read and write speeds (you can’t buy them separately you know). I used this formula:

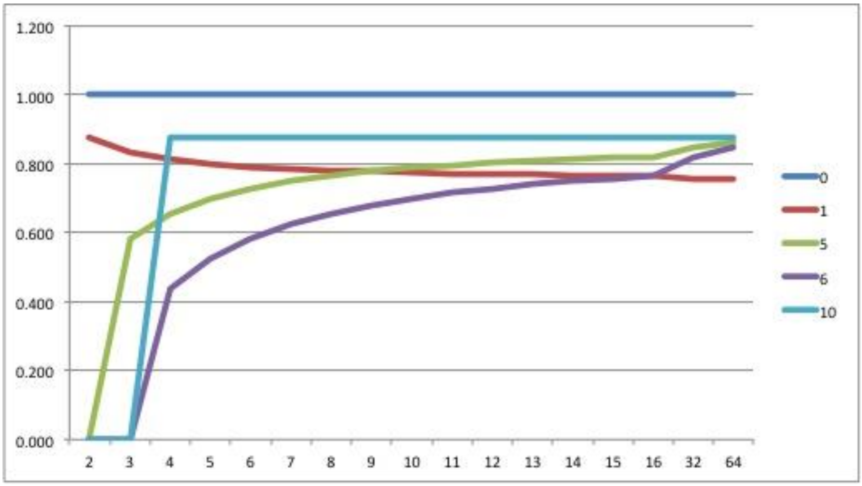
Speed Factor (S) =

(3 x Read speed + 1 x Write Speed)
(4 x Maximum speed of the array)

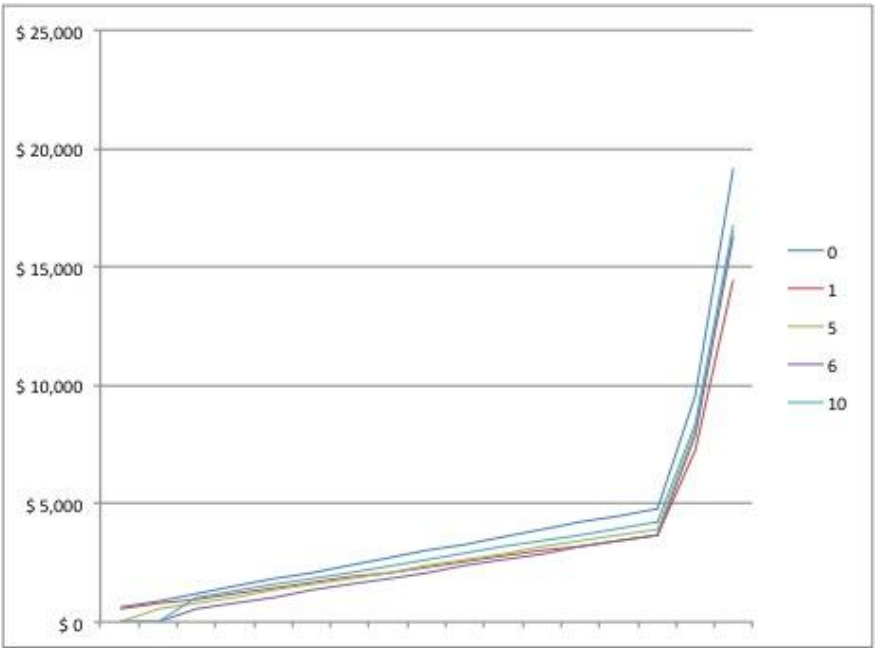
Here’s what we get:

Drives	**Speed factor (S)**				
	RAID 0	**RAID 1**	**RAID 5**	**RAID 6**	**RAID 10**
2	1.000	0.875	x	x	x
3	1.000	0.833	0.583	x	x
4	1.000	0.813	0.656	0.438	0.875
5	1.000	0.800	0.700	0.525	x
6	1.000	0.792	0.729	0.583	0.875
7	1.000	0.786	0.750	0.625	x
8	1.000	0.781	0.766	0.656	0.875
9	1.000	0.778	0.778	0.681	x
10	1.000	0.775	0.788	0.700	0.875
11	1.000	0.773	0.795	0.716	x
12	1.000	0.771	0.802	0.729	0.875
13	1.000	0.769	0.808	0.740	x
14	1.000	0.768	0.813	0.750	0.875
15	1.000	0.767	0.817	0.758	x
16	1.000	0.766	0.820	0.766	0.875
32	1.000	0.758	0.848	0.820	0.875
64	1.000	0.754	0.861	0.848	0.875

And here’s the graph:



For simplicity’s sake, I’m not showing you the value chart but you can multiply the cost of the array with the speed factor to get the Value for Speed (ValueS) for each array. Here’s what it looks like on a graph:



If all you cared for was speed, it is pretty obvious that RAID 0 is the way to go, followed quickly by RAID 10. As far as value for money for speed is concerned, RAID 0 is ideal, followed by RAID 10.

Capacity factor

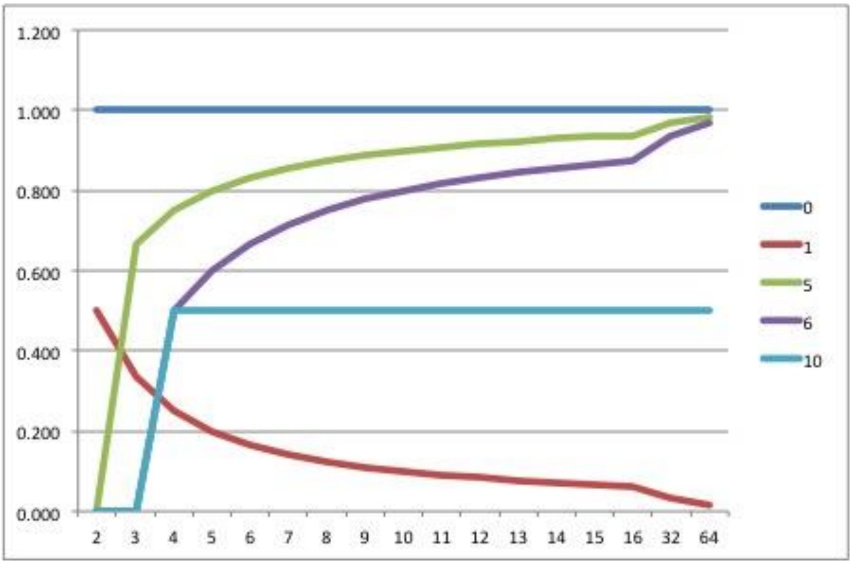
The capacity factor is easier to calculate. It is:

$$\text{Capacity Factor (C)} = \frac{\text{Actual capacity of the array}}{\text{Maximum possible capacity}}$$

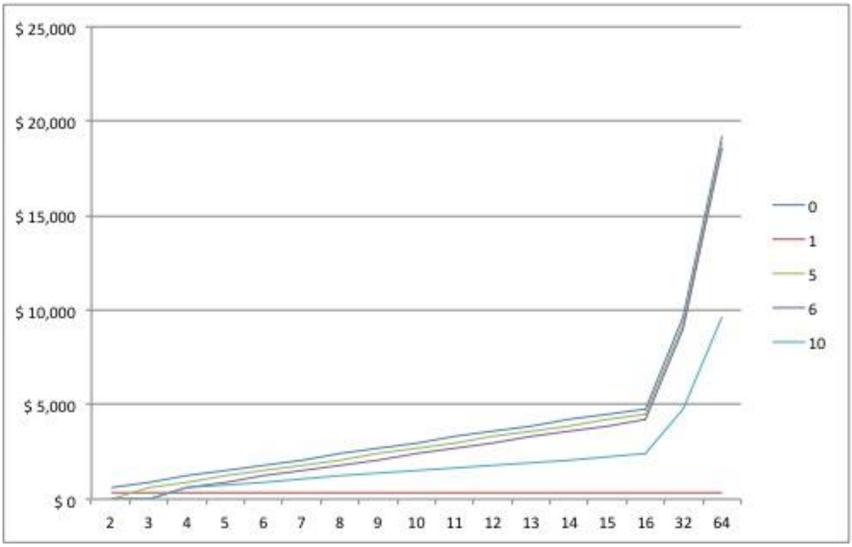
Here is the chart:

Drives	**Capacity Factor (C)**				
	RAID 0	**RAID 1**	**RAID 5**	**RAID 6**	**RAID 10**
2	1.000	0.500	x	x	x
3	1.000	0.333	0.667	x	x
4	1.000	0.250	0.750	0.500	0.500
5	1.000	0.200	0.800	0.600	x
6	1.000	0.167	0.833	0.667	0.500
7	1.000	0.143	0.857	0.714	x
8	1.000	0.125	0.875	0.750	0.500
9	1.000	0.111	0.889	0.778	x
10	1.000	0.100	0.900	0.800	0.500
11	1.000	0.091	0.909	0.818	x
12	1.000	0.083	0.917	0.833	0.500
13	1.000	0.077	0.923	0.846	x
14	1.000	0.071	0.929	0.857	0.500
15	1.000	0.067	0.933	0.867	x
16	1.000	0.063	0.938	0.875	0.500
32	1.000	0.031	0.969	0.938	0.500
64	1.000	0.016	0.984	0.969	0.500

And here is the graph:



The value for money for capacity (ValueC) is as follows:



RAID 0 is the best if capacity is your only concern. However, RAID 5 and RAID 6 both offer excellent value as the number of drives go up. RAID 10 offers half the value.

Failure factor

The failure factor is complicated, and is possibly the most error-prone part of my calculations. Array failure rates depend on the failure rate of a drive (constant) and the number of drives. However, each RAID type also differs in the number of drives that can fail. I feel both these variables must be accounted for in the Failure Factor (F).

The first step is to find the drive failure factor, which is the number of drives that are allowed to fail divided by the total number of drives. Let's call this 'd'. To keep things simple I'm not showing you that chart. You should know that I changed the RAID 0 value (0) to 0.001 just so we don't encounter a 'divided by zero' scenario. We have already calculated the array failure rate earlier.

Resistance to Failure Factor (F) =

$$\frac{(1 \text{ Array Failure Rate} + d)}{2}$$

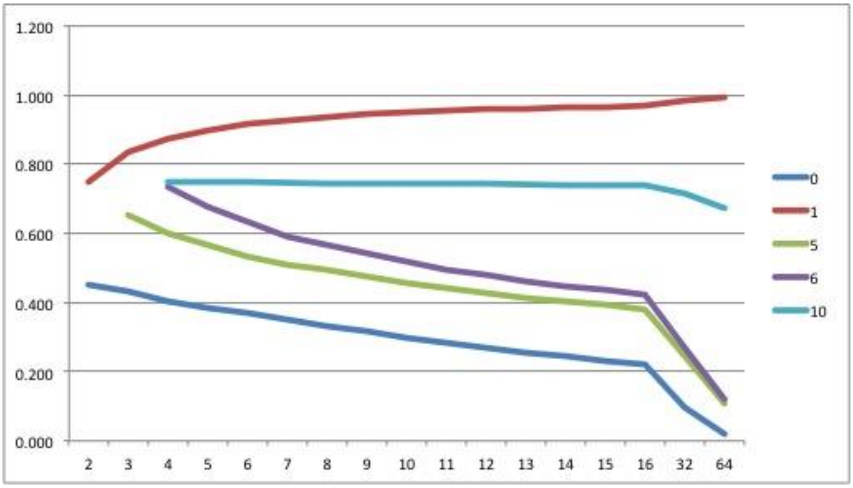
I wanted to create the factor such that the higher number offered greater 'protection'. This makes all factors behave commonly – the higher the better. All I've done is average the two, by giving both factors equal importance.

Here's the chart:

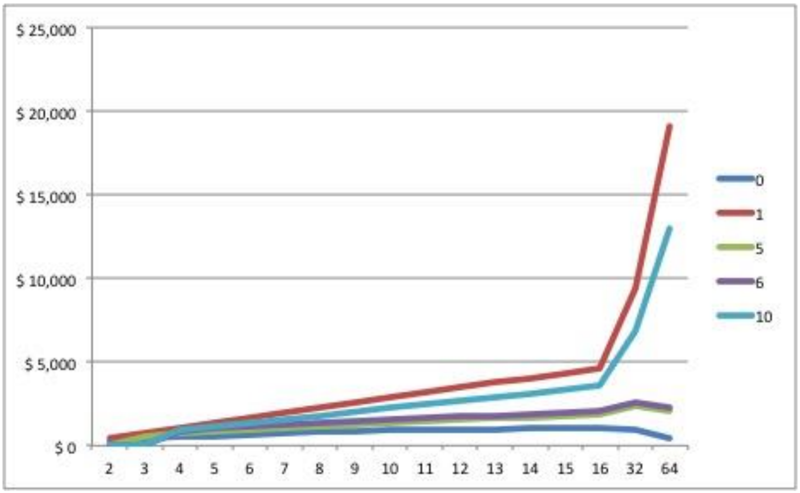
Drives	Resistance to Failure Factor (F)				
	RAID 0	RAID 1	RAID 5	RAID 6	RAID 10
2	0.450	0.749	x	x	x
3	0.430	0.833	0.652	x	x
4	0.405	0.875	0.600	0.735	0.749
5	0.385	0.900	0.565	0.675	x
6	0.370	0.917	0.533	0.632	0.748
7	0.350	0.929	0.511	0.593	x
8	0.330	0.937	0.493	0.565	0.747
9	0.315	0.944	0.476	0.541	x
10	0.300	0.950	0.455	0.520	0.745

11	0.285	0.955	0.440	0.496	x
12	0.270	0.958	0.427	0.478	0.743
13	0.255	0.962	0.413	0.462	x
14	0.245	0.964	0.406	0.446	0.741
15	0.230	0.967	0.393	0.437	x
16	0.220	0.969	0.381	0.423	0.739
32	0.095	0.984	0.246	0.271	0.718
64	0.020	0.992	0.108	0.121	0.673

Here’s the graph:



And here’s the value for money for resistance (ValueF) graph:



RAID 1, in pure protection terms, is always the best, regardless of the number of drives. RAID 10 is the next best thing. RAID 0 is the worst.

The R Factor

The idea behind creating three factors is to multiply them to form one final factor – R.

R = S x C x F

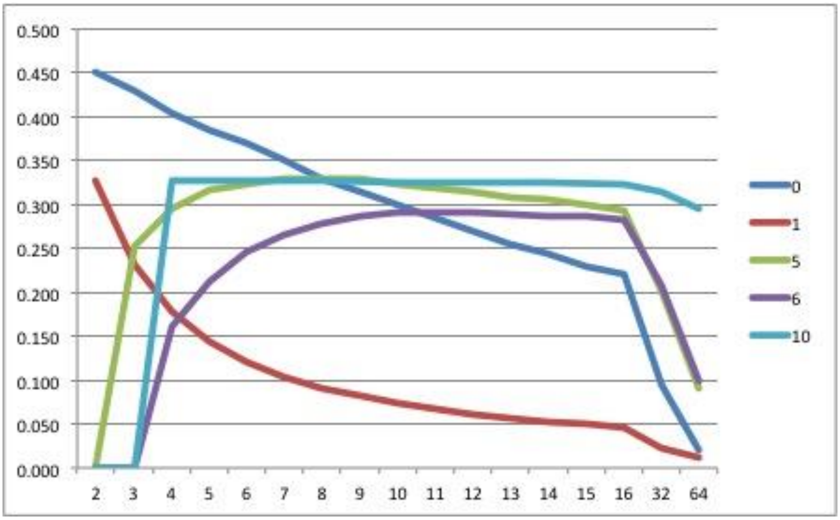
This offers some flexibility, though. What if you didn't care about capacity, for example? In that case, ignore it, and R = S x F. You can ignore two factors and just focus on one thing. Choose any combination you want.

I'm going to leave the permutations and combinations to you, and will only provide the results of combining all three factors.

Here's the chart:

Drives	R Factor				
	RAID 0	RAID 1	RAID 5	RAID 6	RAID 10
2	0.450	0.328	x	x	x
3	0.430	0.231	0.253	x	x
4	0.405	0.178	0.295	0.161	0.328
5	0.385	0.144	0.316	0.213	x
6	0.370	0.121	0.324	0.246	0.327
7	0.350	0.104	0.329	0.265	x
8	0.330	0.092	0.330	0.278	0.327
9	0.315	0.082	0.329	0.286	x
10	0.300	0.074	0.322	0.291	0.326
11	0.285	0.067	0.319	0.290	x
12	0.270	0.062	0.314	0.291	0.325
13	0.255	0.057	0.308	0.289	x
14	0.245	0.053	0.306	0.287	0.324
15	0.230	0.049	0.300	0.287	x
16	0.220	0.046	0.293	0.283	0.323
32	0.095	0.023	0.202	0.209	0.314
64	0.020	0.012	0.091	0.099	0.295

And here's the graph:



What does this tell us? In no uncertain terms:

- If your RAID array is between 2-8 drives large, RAID 0 is best.
- If your RAID array is between 8-10 drives large, RAID 5 is best.
- If your RAID array is greater than 10 drives, RAID 10 is best.

What?? How can RAID 0 be best, even though we all know a single drive failure will ruin our work? We have already calculated the array failure rate for each RAID array. To find how many years the array will survive based on those rates, look at this table:

Drives	**Life Expectancy (years)**				
	RAID 0	**RAID 1**	**RAID 5**	**RAID 6**	**RAID 10**
2	10	400	x	x	x
3	7	-	33	x	x
4	5	-	20	33	526
5	4	-	14	20	x
6	4	-	10	14	256
7	3	-	8	10	x
8	3	-	7	8	147
9	3	-	6	7	x
10	3	-	5	6	100
11	2	-	5	5	x
12	2	-	4	5	72
13	2	-	4	4	x
14	2	-	4	4	56
15	2	-	4	4	x
16	2	-	3	4	45
32	1	-	2	2	15
64	1	-	1	1	7

What is the typical warranty period of a hard drive? Three years? So, one can safely say that a RAID array must ‘live’ for three years before it fails (which it will – all arrays will fail at some point). Going by that, a RAID 0 array with 10 drives will live for three years.

A RAID 5 or 6 array with 16 drives will live up to three years; and a RAID 10 drive with 64+ drives will live longer than 3 years. RAID 1 of course, lives up to 400 years even with two drives. Remember, RAID is NOT backup. Regardless of what RAID you choose, you must always keep backups of your data. When you know you have sufficient backups, an array failure is no longer such a fearful thing, is it?

And don’t forget, your RAID array can be brought to its knees by the controller, the motherboard, the CPU, RAM, the OS, or just human error.

On the one hand we are bombarded with information on how dangerous RAID 0 is, but the numbers don’t support that FUD.

But what about bad luck, you might be wondering? Bad luck can happen to anybody at any time. There is no mathematical basis for luck, only chance. You either believe in luck (didn’t I tell you earlier you could select a RAID array with just your gut feeling?) or believe in chance. It’s not for me to decide for you.

Go make your own luck.

Value for money

What about value for money? Here’s the formula:

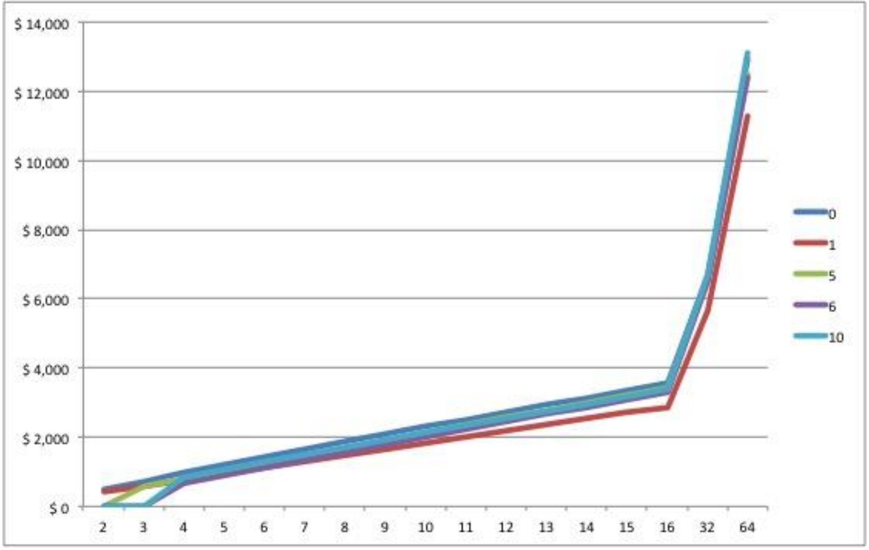
$$\text{Value for Money (V)} = \frac{(\text{ValueS} + \text{ValueC} + \text{ValueF})}{3}$$

Here’s the chart:

Drives	**Value for money (V)**				
	RAID 0	**RAID 1**	**RAID 5**	**RAID 6**	**RAID 10**
2	$ 490	$ 425	x	x	x
3	$ 729	$ 600	$ 571	x	x
4	$ 962	$ 775	$ 803	$ 669	$ 850
5	$ 1,193	$ 950	$ 1,033	$ 900	x
6	$ 1,422	$ 1,125	$ 1,258	$ 1,129	$ 1,274

7	$ 1,645	$ 1,300	$ 1,483	$ 1,353	x
8	$ 1,864	$ 1,475	$ 1,707	$ 1,577	$ 1,697
9	$ 2,084	$ 1,650	$ 1,928	$ 1,800	x
10	$ 2,300	$ 1,825	$ 2,143	$ 2,020	$ 2,120
11	$ 2,514	$ 2,000	$ 2,360	$ 2,233	x
12	$ 2,724	$ 2,175	$ 2,575	$ 2,449	$ 2,542
13	$ 2,932	$ 2,350	$ 2,788	$ 2,663	x
14	$ 3,143	$ 2,525	$ 3,006	$ 2,875	$ 2,963
15	$ 3,345	$ 2,700	$ 3,215	$ 3,093	x
16	$ 3,552	$ 2,875	$ 3,423	$ 3,301	$ 3,382
32	$ 6,704	$ 5,675	$ 6,599	$ 6,493	$ 6,696
64	$ 12,928	$ 11,275	$ 12,503	$ 12,397	$ 13,108

Here’s the graph:



Only RAID 1 falls behind as the number of drives increases, but generally, all RAID levels are good value for money! RAID 0 being the best until a drive count of 64, where RAID 10 takes over.

Takeaways

The numbers, as far as I can tell, don’t lie:

- For 2-8 drives, you'll be safe with RAID 0.
- For 8-10 drives, you'll be happy with RAID 5.
- For more than 10 drives, you'll be thrilled to have RAID 10.

What should you pick? It's not always this difficult. The total size of your array, your data rate and budget will limit your possibilities. Within that, you can use my methodology to come to a final decision. You could either give all factors equal importance, or leave out the ones you don't care about.

You could assume that you will not accept anything less than five years life-expectancy for your drive array. In that case RAID 0 works great for up to 4 drives, and RAID 10 beyond that point. Heck, even if you chose each factor one at a time, it is difficult to argue against RAID 0 or RAID 10.

Going by our charts, RAID 5 and RAID 6 doesn't look good for the kind of work we are into at all! That's actually a relief, because you really don't need to spend extra on RAID controllers when you can use software RAID. Look at the operating systems that support RAID 0, 1 and 10:

- Windows 8**
- Linux+mdadm
- Mac OS X

***Windows 8 has a feature called Storage Spaces that let you create something similar to RAIDs 0, 1 and 5, but not RAID 10. However, Windows also has a software RAID option for striping (RAID 0). You can combine the two to create a RAID-10-like array.*

Let's run by some of the other factors we listed earlier:

- Ease of setup – RAIDs 0 and 1 are the easiest to set up, followed by RAID 10.
- Cost and availability of hardware RAID controllers – the problem with RAID controllers is that if it fails, you will have to find a 'matching' model that supports your existing array.

- Number of drives and size of chassis and heating solutions – RAID 0 will keep your drives to a minimum. RAID 10 takes the most drives.
- Compatibility of hardware and software – See above.
- CPU usage – If no parity calculations are required, you don’t need to tax your CPU.
- Battery Backup for the controller – No controller means no Battery Backup Module (BBM) required. Here, too, compatibility is a problem.
- Caching – Hardware caching is always a good thing.
- Buying drives from various sources to better the odds – is common to all RAID arrays, accept maybe RAID 1.

You do the math.

Takeaway:

But spell it out dammit! Which is the best RAID level for video?

I’m not mincing words: RAID 0 is the best RAID level for video up to 8 drives. After that point RAID 10 is champion.

Chapter 15:

What is a DAS and how to Choose One?

The way things are going, you could safely bet that you'll always need more storage, never less. Nobody wants to transfer data every few months once their drives run out, and it is not always a good idea to put everything in internal drives because then it is totally dependent on your computer.

What do we do, then? You could opt to build an external RAID array by yourself. Or, you could invest in a DAS.

DAS stands for Direct-Attached Storage solution. This is nothing but a fancy word for a desktop external storage solution that directly connects to your computer.

You could divide external solutions into the following groups:

- Portable – they don’t need an external power supply.
- Desktop – they need an external power supply. These are DAS solutions. DAS is designed to be used by only one computer.
- NAS/SAN – these are DAS-types with one fundamental difference. They have Gigabit Ethernet connectivity because they are designed to be part of a network of computers. We’ll look at these in the next chapter.

Formerly, USB 2.0, Firewire and eSATA used to be the choice connection options for DAS solutions, but not anymore. Today, the two most important standards are:

1. USB 3.0
2. Thunderbolt

What is the difference between USB 3.0 and Thunderbolt?

It’s too early to call a winner. It’s not even clear if there *should* be a winner. So far, tests have favored Thunderbolt as far as speed is concerned. But it isn’t supported on all systems, and that is a huge negative. I don’t understand why Thunderbolt drives don’t come with USB 3.0 or at least eSATA options.

As far as speed is concerned, USB 3.0 has a theoretical maximum of 5 Gbps (which might expand to 10 Mbps in the future). This works out to be 625 MB/s. I think it’s fair to say you can’t count on more than 500 MB/s over USB 3.0 at present.

Thunderbolt 2 is theoretically capable of 20 Gbps (10 Gbps for Thunderbolt 1). This works out to be 2,500 MB/s or 2.4 GB/s. I think it’s fair to say Thunderbolt is capable of sustaining 1 GB/s, and this is huge for large file transfers and real-time playback.

The choice is not that simple. The actual speeds depend on many factors, and even conducting a fair side-by-side test is tough. It’s a lose-lose scenario. By the time you’ve got to the ‘bottom of things’, something newer will have hit the market.

Bottom line: If you’re on a Mac, and a newer one at that, I believe Thunderbolt 2 to be a good investment. Remember, there’s no shame in opting for USB 3.0 today and Thunderbolt tomorrow. The whole point of a DAS is that it is temporary, and must return its value and more in a short period of time.

Buy what you need today and tomorrow, not the day after tomorrow.

In any case, I consider Thunderbolt 2 to be a far superior technology, for the following reasons:

- Speed.
- Long data cables, up to 30m (100 feet) at the time of this writing.
- Can display 4K video as well when required.
- Can be daisy chained.

Comparison of Thunderbolt RAID storage solutions

How do you compare DAS solutions? To help you out I have selected 10 Thunderbolt 1 RAID storage solutions (actually, it's only 9, the 10th one is USB 3.0). I've chosen the following models:

1. Promise Pegasus
2. LaCie 2big
3. Areca ARC-8050
4. G-Technology G-RAID
5. BUFFALO DriveStation Duo
6. Drobo 5D
7. WD My Book Thunderbolt Duo
8. Seagate Backup Plus
9. Sonnet Fusion DX800RAID
10. CalDigit T2

Here's a detailed comparison chart (*if you're having trouble reading it, go here: http://wolfcrow.com/blog/wp-content/uploads/2013/06/DASComparison.jpg*):

Manufacturer	Promise Technology, Inc.	LaCie	Areca	G-technology	Buffalo	Drobo	Western Digital	Seagate	Sonnet	CalDigit
Series/Model	Pegasus Thunderbolt	2big Thunderbolt	ARC-8050	G-RAID with Thunderbolt	Drivestation Duo	Drobo 5D	My Book Thunderbolt Duo	Backup Plus with TB adaptor	Fusion DX800RAID Thunderbolt	T2**
Maximum total size	8 TB	8 TB	32 TB	8 TB	4 TB	16 TB	4 TB	4 TB	32 TB	4 TB
Available bays (4 TB)	4	2	8	2	2	5	2	1	8	2
RAID levels	0, 1, 5, 50, 6, 60 and 10	0 / 1 / JBOD	0, 1, 1E, 3, 5, 6, 10, 30, 50, 60, Single Disk or JBOD	0 / 1 / JBOD	0 / 1 / JBOD	BeyondRAID (1 / 5 / 6 possibly)	0 / 1 / JBOD	No	RAID 0, 1, 4, 5, 6, and 10 / JBOD	RAID 0, 1 and JBOD
Thunderbolt ports	2	2	2	2	No	2	2	2	2	2
USB 3.0 ports	No	No	No	No	1	1	No	No	No	No
eSATA ports	No	No	No	No	No	No	No	No	No	No
GigE Port	No	No	Yes	No	No	No	No	No	No	No
Drive speed compatibility	7200 rpm	7200 rpm	All	7200 rpm	5400 rpm	7200 rpm	5400 rpm (Green)	5400 rpm	7200 rpm	7200 rpm
Maximum RAID 0 Speed	500 MB/s	250 MB/s	> 1000 MB/s	250 MB/s	250 MB/s	600 MB/s	250 MB/s	125 MB/s	725 MB/s	630 MB/s
Power at Full Load	78 W	?	?	?	?	150 W	?	?	300 W	300 W
HDD Standards	SATA	SATA	SAS/SATA	SATA	SATA	SATA	SATA	SATA	SATA/SAS	SATA
Warranty	2 years	?	?	3 years	1 year	2 years	?	?	5 years	?
Thunderbolt Cable included?	No	Yes	No	No	n/a	Yes	No	Yes	Yes	Yes
Price for 4 TB including drives*	$1,025	$491	$1,679	$590	$296	$956	$550	$450	$4,765	?

Copyright © Sareesh Sudhakaran 2013

Notes:

**The price is ‘normalized’ to 4 TB 7,200 rpm drives, but only if they don’t come with the package. For 8-bay enclosures, I’ve only added the price of 4 drives (Areca) or subtracted the price of 4 drives (Sonnet). Prices are not meant to be accurate. Please refer to the manufacturer’s specifications for actual values and prices.*

***This model isn’t available yet. I’ve included a NAS+Thunderbolt option, the Areca ARC-8050, simply to point out the differences. I’ve included a USB 3.0-only option, the Buffalo Drivestation Duo, for similar reasons.*

Promise Pegasus2 has been released, with Thunderbolt 2. Since other manufacturers haven’t followed suit yet, we can’t have a fair comparison.

Please note that the chart might not be totally accurate. Please refer to the manufacturer’s specifications for actual values and prices.

General Observations:

- All 2-bay storage solutions tend to be in the $500 range for 4 TB.
- RAID 0 for a 7,200rpm 2-bay system should sustain 200 MB/s, which is fine for Redcode and CinemaDNG playback. It is not enough for multitrack real-time editing, or playing back image

sequences like TIFF, DPX, etc. RAID 1 speeds tend to be *slower* than one drive, which is a bummer.
- Brands that have traditionally only supplied consumer-oriented external solutions tend to use 5,400 rpm drives. E.g., WD My Book, Seagate and Buffalo, etc. Western Digital has a Velociraptor Thunderbolt solution, but it stops at 2 TB. These drives are perfectly fine for basic editing work with most consumer and intermediary codecs. These drives tend to power down when not required, and are sometimes variable speed drives (like the WD Green series).
- 4 or 5-bay solutions hover in the $1,000 range. The more the number of drives, the greater the read throughput in RAID 0, 5, 6 and 10.

Which Thunderbolt storage solution is the best?

There is no cut and dried answer, since each solution has its own market. The needs of filmmakers vary a great deal. There is also the question of availability, since not all of these models are available in all countries. When available, they might not be at the same price points your see here. Lastly, just because the service is good in one country or region doesn't mean you'll get the same level of service in another country or region.

Here are some lines of thought to get you started:

If you want a storage solution for real-time playback, your choices are fairly limited. It is a simple process to estimate your drive speed requirements for playback and then work your way backwards. Use Deconstructing RAW and the Costs of working with 2K and 4K uncompressed footage to get going.

A 2-bay solution is small and consumes less power. If you're working with drives on a project-by-project basis, it might be a good idea to take things one enclosure at a time. Eventually, if you're a freelancer or studio, you will accumulate many such drives, which may or may not play well together five years from now (five years is a large time-gap in computer technology terms).

A simpler but more time-consuming alternative is to estimate your yearly footage requirements, and then design your storage according to that. For large studios or centers, the choice boils down to a NAS or SAN. If your yearly requirements are in the 4 to 6 TB range, then you might be able to use just one Thunderbolt or USB 3.0 storage solution and have all your data in RAID (5, 6 or 10) for an uninterrupted workflow. You will still need additional backups, there's no escaping that no matter what your storage solution.

I'm going to run with this last option.

At this point in time, if you're a small-scale filmmaker, going by yearly trends, like I am, then the following 'calculations' (or miscalculations!) might help you:

- I usually shoot on a C300 at 50 Mbps. My shooting ratio varies greatly, but I'd say a 5 minute end product might require 2 to 3 hours of footage, including B-roll. I tend to estimate 5 hours of source material (I don't want to max out my drives, ever) per project. This works out to be about 100 GB.
- I tend to keep three (actually four) copies of my source material – Two in RAID (RAID 0 or 1, both work for 50 Mbps) for the editor, one at home, and one at the office. That's about 400 GB per project. Since I can't buy 100 GB drives I'll have to at least make sure I have 4 x 1 TB drives for a project.
- In the coming year I expect to shoot at data rates higher than 50 Mbps. E.g. CinemaDNG or Redcode is easily 150 MB/s, which works out to 2.5 TB of source material per project, or a total of 4 x 3 TB = 12 TB of storage for one project.
- By next year I might be able to reuse some of the extra office/home drives, or I may not. It depends on what it contains, who it is for (if it's a big client it'll stay reserved) or what new technology is around. Sadly, I can't make that call today.
- I'm estimating I might be working with close to 50 TB, at least, for everything, for the coming year. I might end up using less than this or more. There are too many variables.
- 48 TB divided by four is 12 TB. I could look into a 24TB Thunderbolt DAS for my editing RAID system. This will

most likely be in RAID 10 (I don't like RAID 5 or 6, but others find them okay).

- There is a serious decision to be made: Should I go for one big solution, or two smaller ones? The smaller ones will let me continue working if one fails. The bigger one is a bigger risk as well!
- So, how's a 12 TB Thunderbolt solution in RAID 10? It'll give me 6 TB of space. For backups, I can customize the size on a project-to-project basis.
- Is there a solution in my chart that meets my needs? I'll have to eliminate all of the dual bay solutions because if I'm working with RAW I'll need the speed. This leaves me four choices: A Pegasus, a Sonnet Fusion, an Areca or a Drobo.
- I'm not very keen on Drobo's proprietary RAID so I pass on that. The Sonnet is out of my price range. That leaves two.
- The funny thing is, in my country, none of these models are available locally. If I import them, I won't be covered by warranty, and I probably won't get good service.
- Deep breath.

For my needs, the Areca ARC-8050 seems like a solid deal. However, it doesn't fit in with my strategy. It's overkill. I really have to 'commit' to it for the long term.

The Areca ARC-8050 Thunderbolt RAID 8-Bay System

If I had to buy, I'd go for the Areca ARC-8050. Nothing comes close price-wise, and the reviews so far are good. If you're on a Mac station and feel comfortable with Thunderbolt, then one can't fault the Promise Pegasus2 either. In fact, if you're on the newer Apple Mac Pro with Thunderbolt 2, then at the time of this writing there's no substitute to the Pegasus2.

Takeaways:

- **DAS – Direct Attached Storage** – is any storage (single drive or multiple, RAID or otherwise) that connects to your computer externally, usually via USB 3.0 or Thunderbolt.
- The best Thunderbolt 1 option for the money is the Areca ARC-8050 Thunderbolt RAID 8-Bay System.
- The best Thunderbolt 2 option is the Promise Pegasus2.
- You might want to wait for a few months to see what new Thunderbolt 2 options are available. One hopes PC makers will incorporate Thunderbolt 2 into their motherboards, and that will change video editing forever.

Chapter 16:

What is a NAS or SAN, and how to tell which is which?

***First, a disclaimer:** I'm not an expert in either a SAN or a NAS. My understanding is, to put it simply, my understanding. This could be wrong or inaccurate. Like you, my first priority is video, and I don't have the time or brainpower to figure out the complications of a SAN or NAS before I invest in them. I want the bottom line, and this is my attempt to understand which is which, and which one I should be interested in. You are responsible for your own choices and actions.*

If you're in the video world you might have heard about a NAS (Network Attached Storage) or a SAN (Storage Area Network). Let's try understanding them without killing ourselves.

The problem of understanding a SAN or NAS is like a tree. You can't get the whole tree without doing a bit of digging. So, we dig.

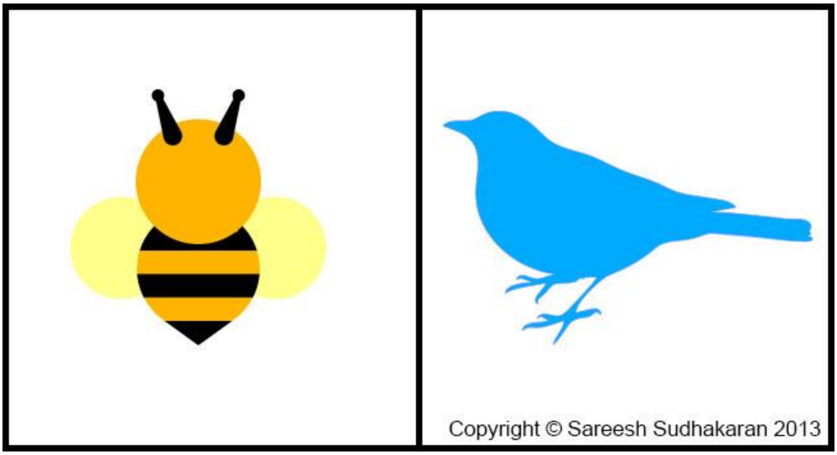
The birds and the bees

Contrary to popular belief, the choice between a NAS or SAN is not similar to a Pepsi vs Coke battle, but more like a Pepsi vs Mirinda battle, where Pepsi is orange in color.

I'm sure this analogy doesn't make much sense. That's the point! Don't get mad at me; let me redeem myself by explaining.

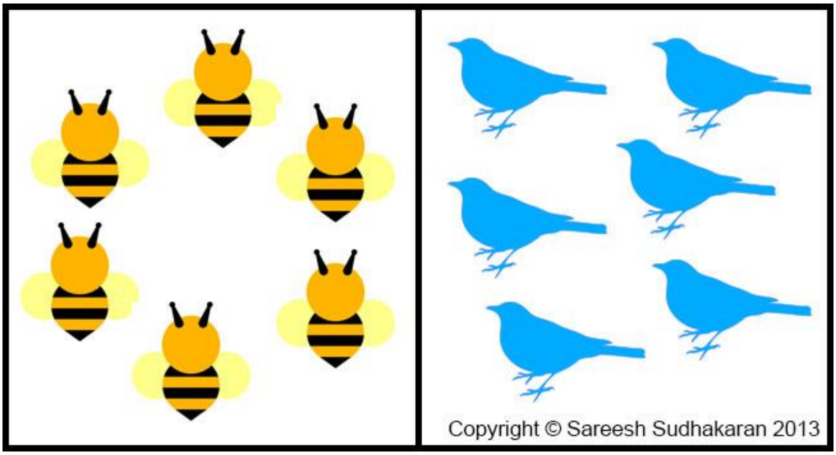
Let me use a better analogy to help you understand. Since we're in the subject of nature, let's choose the birds and the bees as our Muses.

In the beginning, there was a bird and a bee:



Copyright © Sareesh Sudhakaran 2013

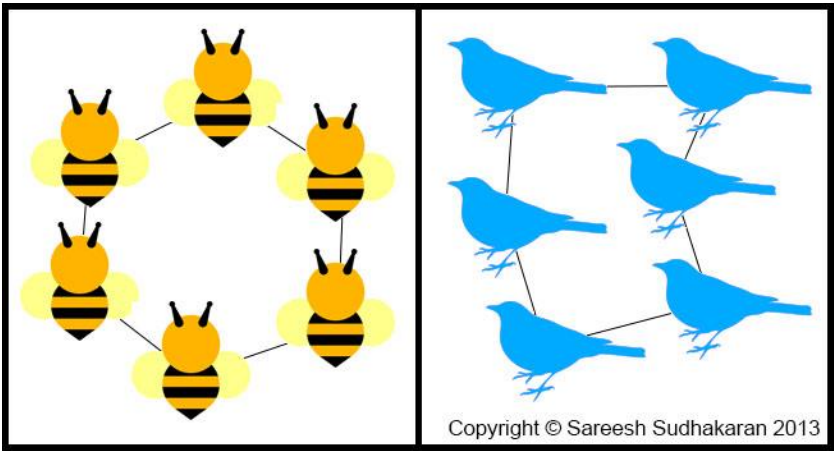
Birds of a feather flock together, so they look like this:



Copyright © Sareesh Sudhakaran 2013

Nothing fancy, just regular chums hanging out.

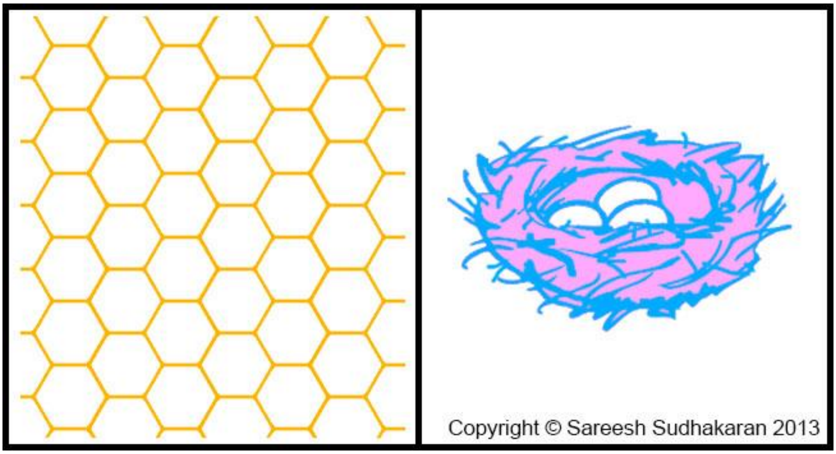
Birds and bees don’t communicate globally by email or Facebook. They tend to remain as local groups. Let’s call this group a Local Area Network, or LAN. Here’s one version:



Copyright © Sareesh Sudhakaran 2013

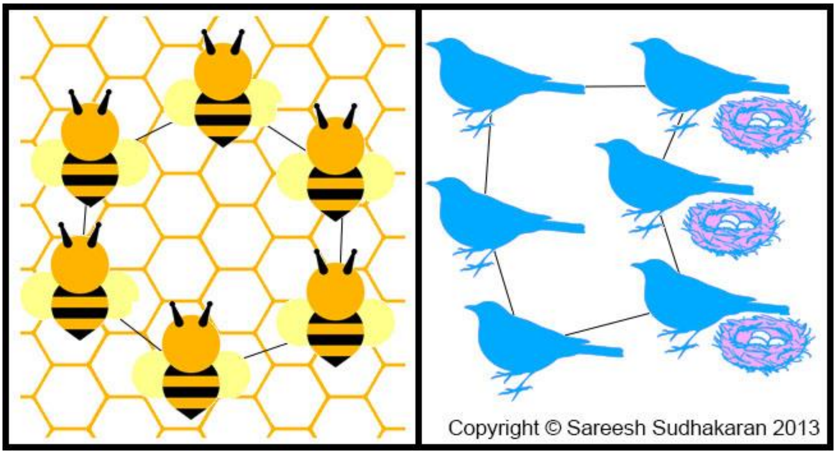
Straightforward enough, right? Birds have pals, bees have pals. They stay connected in various permutations and combinations. No big deal.

They have to eat, so they gather food. They also need to store food somewhere. This is how they do it:



Copyright © Sareesh Sudhakaran 2013

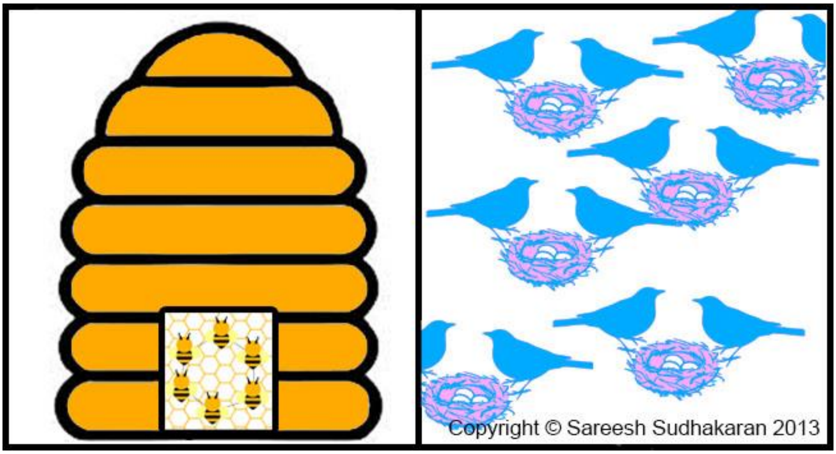
How do bees and birds use their storage areas? Here’s how:


Copyright © Sareesh Sudhakaran 2013

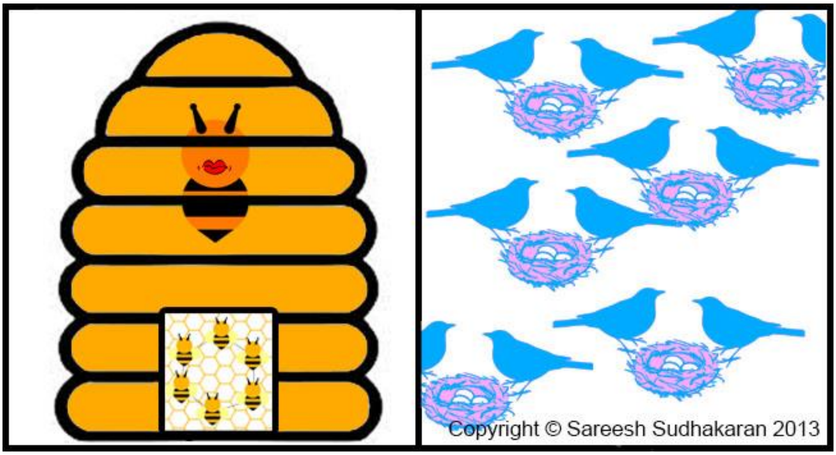
Wait a second, that's a big difference, isn't it? Bees have one home, where they make excess honey anticipating the future winter. Birds prepare nests to feed their young, but otherwise just hang around on trees in groups. Now, here's the killer:

Unlike a beehive, where each bee works for itself *and* the other, birds tend to be selfish creatures. They don't interfere in each other's storage areas. I mean, they *could*, but it's frowned upon. A bird that acts un-bird-like is an outcast.

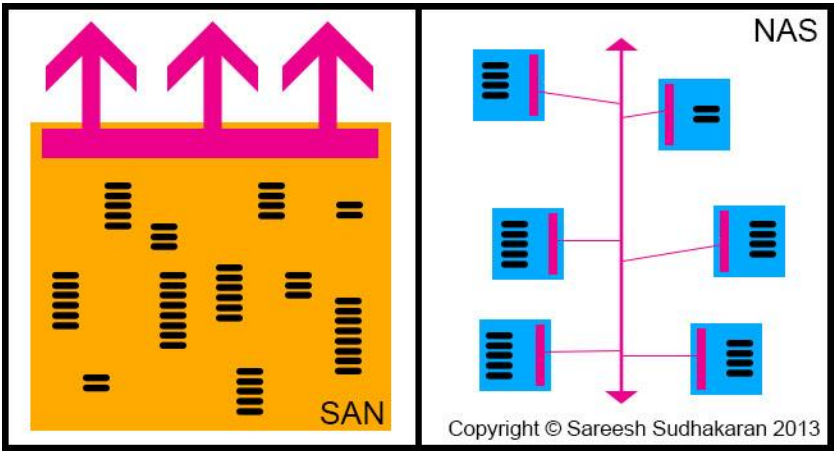
Now, let's assume both groups (LANs) want to grow their storage areas. This is what it'll look like:


Copyright © Sareesh Sudhakaran 2013

Interesting. Who is the captain of both these storage areas?



In the case of the birds, each couple is in charge of their own nest. In the case of the bees, the Queen bee is in charge. Let's leave our birds and bees, and replace them with computers:



Don't be shocked. Look at the colors:

- Bees and Birds act like computers connected to a LAN.
- Blue stands for one NAS. Six blue boxes = Six NAS systems.

- Orange is the whole group of bees, or the hive. One big orange box = One SAN system.
- Pink is the captain. In the case of a SAN (bees), there is only one captain. Each NAS system has its own captain.
- The pink arrow lines are ‘demands from computers’ in the network. In a SAN, they have to go through the captain. It’s the same for a NAS!

In a SAN your storage systems can be of different types: drives, tapes, whatever. They are all pooled together and managed by the captain. A SAN acts like one big hard drive.

A NAS can also have different types of storage: drives, tapes, whatever. But each NAS is designed to only hold one type of storage, and they behave selfishly.

SAN or NAS? What’s the difference?

Ready for the fundamental definition? The answers are in the names themselves! Look:

- SAN – is a Storage Area **Network** – is a Blah Blah Network – is a **Network**
- NAS – is a Network Attached **Storage** – is a Blah Blah Storage – is a **Storage**

A SAN is a network of storages (network of drives or drive systems or tape systems or whatever), while a NAS is storage (either drives, tapes, whatever, but only one kind) that is part of a network.

It is from this fundamental difference that their major properties arise:

- Each computer connected to a LAN feels like the SAN is its own (bees feel the whole beehive is their own).
- Each NAS connected to a LAN is separate, and don’t care about each other. A computer that connects to the LAN will have to ‘reach out’ separately to each NAS. This makes the computer feel that the NAS is a shared thing, and not its own.
- If the captain of the SAN goes down, the entire SAN fails.

- If one NAS fails, the other NAS-holes don't care.
- You can detach a NAS from the LAN and nothing much happens.
- You remove a whole SAN and the computers no longer have access to your centralized storage.
- When a SAN grows, it physically consumes more space.
- Both SAN and NAS systems are designed for LANs only (studios, post facilities, etc.), and not for Wide Area Networks (College campus, the Internet, etc.).

The way a NAS or SAN 'communicates' is different. Bees and birds speak different languages. They can't talk to each other, or they'd be complaining about us.

Here are some properties:

- Since a NAS is expected to be plugged-in to a LAN wherever the need arises (different rooms, or even across the globe to another LAN), it follows standard protocols (like HTTP, TCP/IP, NFS, etc.) that work over regular Ethernet.
- Since a SAN is expected to stay put in a facility, and many computers access them at once, speed is of the essence. For this reason, you'll find many SANs using Fiber Channel over protocols like SCSI, etc.

You see now? They are designed for different usage scenarios, and the underlying technologies try to match those requirements.

When seen from afar, they all look the same, which is what I meant by the Orange Pepsi vs Mirinda analogy. If you can't be bothered to dig deeper for the whole tree, then you don't get the tree.

Just because they share the same three letters don't assume they are related. They're not.

How a SAN or NAS handles data transfer

If both of them look alike from afar, how do we tell them apart? One simple way is to look at how they handle files.

Let’s imagine you want to go backstage to get an autograph from your hero. There are two ways it can be done:

1. You are given a backstage pass, you meet your favorite star and get the autograph from those divine hands, or
2. You are asked to wait outside, while your cap or handkerchief or whatever is taken backstage, and then brought back with a signature.

In the real world, if the autograph represents data, there are two ways it can be had:

1. Through a server (like getting a file off the internet), or
2. Getting it directly from your hard drive.

In the first instance you need a browser (HTTP) or FTP to reach and download the file. In the second instance the file is directly available to you.

So –

- SAN – acts like your personal hard drive, or a DAS.
- NAS – acts like a server, which it is.

As we saw earlier, a SAN is built for speed and reliability. To make the SAN behave like your hard drive, you'll have to eliminate the middleman – the Operating System/File System. A SAN actually lets you reach and take what you want from a hard drive directly. The technical word for this is: *having block-level access* – the ability to read files directly in blocks (many at a time) to get the fastest data transfer possible.

The file system isn't eliminated, of course. You always need a file system.

A NAS, on the other hand, is a server, which must have an Operating System and a file system. To get something from a NAS, you must go through its rules, which is called a protocol. What you must ask about a SAN or NAS then, are the following critical questions:

- What connection is used: Fiber Channel, Ethernet, etc.
- How would you get that on your computer?
- How much data do you handle?
- How many people work with the same data?
- How many computers will be used simultaneously?
- Do all the computers have the same connections as your SAN or NAS?
- Do you use Macs, PCs or both?
- How much money and time do you have?
- How much space do you have?
- If it fails, how much downtime can you suffer?
- What's your backup strategy?

The answers to these questions, within the context of what we're going to look at next, will point you in the right direction.

When to use a NAS or SAN for video work?

For the sake of simplicity, we can divide workflows for editing and finishing into the following groups:

1. On-set data station
2. Single computer or laptop
3. Small production company (2-5 computers)
4. Post house or studio (5+ computers)
5. Large VFX facility (50+ computers)

Let's look at them one by one:

1. On-set Data Station

This is a device or computer that needs to ingest, transcode and backup data. The traffic is usually from the computer to the drives. The fastest way to do this by using a DAS. Connect all your disk arrays to your computer, and that should be enough. When data does need to travel to a video village or monitor or switcher, the established system is HD-SDI.

There is no network involved, so both a SAN or NAS is unnecessary.

2. Single Computer or Laptop

The same logic holds as above. There is no network involved, so it is pointless to use a NAS or SAN where a DAS will serve you perfectly.

If you compare this to a NAS, the Gigabit Ethernet connection will prove to be far slower than a USB 3.0 or Thunderbolt port. Here is how they compare:

Connection	**Gbps***	**MB/s***
1 GbE	1	125
10 GbE	10	1,250
USB 3.0	5 (10)	625
Thunderbolt 2	20	2,500
HD-SDI	1.485	186
3G-SDI	2.97	371
Speed of one 7,200 rpm drive		*120*

**Except for the hard drive, all speeds are maximum speeds. Read-world performance is always lower.*

A 1 GbE connection (what you get on most computers) is only as fast as one hard drive (actually much lower!). At some point you must ask yourself: Why would I connect four drives in RAID and then bottleneck myself with a 1 Gbps connection (a NAS), when I have USB 3.0 and Thunderbolt at my disposal?

3. Production Company (2-5 computers)

Let’s say you have three computers for editing, one for finishing and the third one as your ‘office’ computer or whatever. For some reason you want all of these to have access to a shared storage.

You’ll need a SAN if you:

- Can afford it.
- Need all editors and computers to access the *same data*. It is very rare for a small production company to have their editors working on the same project!
- Anticipate scaling your storage in the future.
- Work with footage that has a data rate greater than 150 MB/s, *all the time*.

Take a look at our chart above. Is a super-expensive SAN really worth the price; say, over a few Thunderbolt or USB 3.0 DAS systems? What happens if you spend your last dollar on a SAN and it goes down?

You could play Counterstrike while someone fixes your SAN…oh but wait! Your saved games were in the SAN!

What about a NAS? You’ll need a NAS if you:

- Need only some of your computers to share the same data.
- Need to move that box around for some reason.
- Can live with 1 GbE speeds and weird file systems.
- Can afford a 10 GbE NAS if you need Thunderbolt-like speed.

At what point does a NAS make more sense than, say, a DAS or an Apple 3 TB Time Capsule?

These are questions only you can answer, based upon your clientele, workflow, business model and ambitions. My advice? Small production companies with 5 computers don’t need a NAS or SAN for daily work. If you want a centralized backup system, that’s *another* thing. You can always build a cheap backup and archival system.

4. Post house or Studio (5+ computers)

If you are post house with more than five computers, the chances are you have multiple computers dedicated to one project. At any given point you might be moving between editing, color grading, effects, audio, a few render nodes, and so on. It is for this scenario that SANs make sense (if you can survive the sticker shock). If you’re a successful post house that works with so much data that moving hard drives around is no longer fun or efficient, it’s time to get a SAN.

A SAN will:

- Help you scale.
- Get you critical support and updates as and when required.
- Give you speed per node, with a reliable infrastructure to back that 24/7.
- Keep things compact and neat.
- Help you work without needing servers. You can use your computers to directly interface with the SAN.
- Churn out work as fast as you can, using a DAS on one computer.
- Take away the pain of troubleshooting.

What a SAN really does is it allows you to continue working as if nothing has changed, even when you add multiple computers and people. A reliable solution takes the complexity off your shoulders. It’s only in this scenario that the price of a SAN pays off.

The decision is easy. If you can’t afford a SAN, then you can’t afford it. Unless you have a brilliant IT wizard who can put together a SAN, ensure it interfaces with your computers, operating systems and workflow, and is able to discover problems and correct them in less than 24 hours, don’t try to build a cheap SAN.

What about a NAS? Would that do any good? A NAS will:

- Help you divide your studio or post house into pockets.
- Get you some kind of support.
- Keep things somewhat compact and neat.
- Slow you down unless you have methodically planned its use.
- Not help you scale.
- Not keep things compact and straightforward.
- Not feel like a DAS.

Breaking down your workflow into NAS units is, in my opinion, not very different from having DAS units. I don’t recommend it for this scenario. A NAS is beneficial for backups and archival, but I don’t see the benefit of a NAS for a small post house or studio.

5. Large VFX facilities (50+ computers)

Large VFX facilities have custom-made network infrastructures that simply must have a SAN. The best way to visualize data flow in such a facility is by looking at the electrical wiring. Electricity flows through every cable to every station, light and air-conditioner – that’s exactly how data must flow for any work to get done. It must appear and *be seamless*.

So, what’s the bottom line? As far as I understand, I prefer a DAS for all small work groups – it is manageable, even when working with 150 MB/s footage. An 8-bay Areca ARC-8050 Thunderbolt DAS hybrid costs only about $1,700, and you can stuff 32TB of data in there. If you’re on RAID 6, you get about 24TB, and that’s 44 hours worth of 150 MB/s footage.

With the price of DAS systems falling rapidly, it is almost criminal to complicate your life unnecessarily by investing time, money and effort into either a NAS or SAN if you absolutely don’t need it 100%.

When you grow into a larger post facility, with many people accessing the same data, a SAN is definitely the right way to go. You could also justify a SAN as a backup system rather than as a ‘work’ system, but that’s a call only you can take (It’s expensive!).

As for NAS units, I’m beginning to believe they are unnecessarily complicated for video work. They’re not cheaper than a DAS, nor much cheaper than a SAN all things considered. I’d stay away from the 1 GbE Ethernet NAS units that are primarily intended for business storage and archival (even then, transferring large amounts of video data at only 1 Gbps when you can do it at 20 Gbps is stupid). Video demands better speed and throughput. A DAS gives you that easily. When it’s time to move to a SAN, you’ll know.

Takeaways:

- NAS – Network Attached Storage – is a STORAGE that is connected to a network, usually via Ethernet.
- SAN – Storage Area Network – is a NETWORK of storages (can be different types) that act collectively like a DAS.
- Video professionals – stay away from a NAS, period.
- If you are a large VFX or post facility, a SAN is the best way to go.
- For everyone else, stick to a DAS.

Chapter 17:

How to get SAN Storage for Video Production

SAN Storage is a mission critical part of your production or post production workflow. If it isn't, then you don't need one. This article is for the first-time SAN storage hunter who is looking to get up to speed quickly.

A disclaimer: If you really want detailed and correct advice, the person or organization you should be consulting is called a Systems Integrator.

If you have twenty workstations but they don't need to access the same data, then a SAN is overkill. Any DAS will do.
On the other hand, if you meet all criteria for a SAN but can't afford a SAN, then there's something wrong with your business model!

How to select the right Systems Integrator

If you're an IT or networking expert, then you could build a SAN yourself. It isn't *that* hard (not like building a camera or programming an NLE from scratch). Actually, in many cases if you're halfway there (have some experience in networking and have the time and inclination to make a determined effort) it might be worth the risk.

On the other hand, 99% of video professionals are not engineers, and don't want to worry about the million problems that come with setting up a network storage architecture. The person or organization that gets a SAN working is called a systems integrator, or simply, integrator. Their job is to study your workflow needs for the foreseeable future and suggest the right vendors. If you pay them enough, they will oversee the installation and also take responsibility for maintenance.
Here are some things you should look for in a good systems integrator (not very different from finding a good plumber):

- Must be in the same area as you are, so they can be physically present whenever needed.

- Must have already installed SAN storage for several facilities similar to yours, hopefully ones that you can call and learn from.

- Must not be affiliated to any vendor. I.e., no hidden agenda. This is deadly.

- Must know networking inside out. If they can’t or won’t answer your questions, or if they speak down to you, avoid them.

- Must offer proactive suggestions. If you’re ‘just another client’ to them, then don’t become their client. Don’t fall for false promises.

- They must be willing to tell you how to build a DIY SAN. If they have ‘trade secrets’, then let them go. There are no secrets in networking that isn’t available for free online.

Only you can decide if you need a systems integrator or not. Here’s one way to know: If you are having trouble reading this chapter, then you’re going to find it hard without the help of a good systems integrator.

Should you build a SAN yourself?

You can't understand how cold the water is without getting some part of your body wet. I strongly urge you to consider building a SAN yourself before you consult anybody. It'll take a week or so of research, but it'll be invaluable. Most of us will fail in this endeavor, but we will get the bigger picture. Two reasons for doing this:

- You know your workflow better than anyone else. You will be able to speak in a language you now understand.
- Nobody can screw around with you. The more you know, the more you can 'challenge' your integrator.
- When your business grows further and the time comes to expand or replace your SAN, you'll have learnt a whole lot more.

What do you need to build a SAN?

You can build a SAN with commonly available hardware. That's the easy part. The hard part is software. Let's break it down to four major divisions:

1. Storage
2. Network
3. Workstations
4. Software

Let's look at each division one by one. Remember, the reason why building a SAN is tough is because there are too many choices at every turn. It's like chess, where one move might have multiple ramifications down the line that you can't even see right now.

Important: *I might give you examples of parts with model numbers, but for heaven's sake don't assume they will work well together as a SAN! I only mention specific models so you can see what they look like. Don't use it as a shopping list.*

Storage

Many people incorrectly assume that a SAN is just storage. In fact, it's everything in the network (Storage Area **Network**, anyone?). Storage is just one part of the SAN.

How to estimate storage requirements

You need drives to store data. Since data will be accessed simultaneously, you will need fast storage. The only cheap way to do this with redundancy is RAID.

Your SAN is a working beast, and is not always required to archive your data. E.g., if you work predominantly with Prores HQ footage, your data rate for one stream is 27.5 MB/s. If you have multiple editors (say three) working on a documentary with 200 hours of footage, you'll need:

- A read speed of 82.5 MB/s for just one stream.
- 20 TB storage.

Let's say you are using 4 TB 7,200 rpm drives in your SAN, in RAID 6. You'll need 8 drives (total 32 TB) of which you'll get 24 TB of storage.

If each drive can sustain 100 MB/s, then this array will deliver a maximum read throughput of 600 MB/s (7 streams per editor). That'll do fine for Prores HQ workflows.

The drives

Welcome to the first big choice. We assumed 7,200 rpm drives, but it's not that simple. You have the following choices:

- SATA III 6 Gbps
- SAS 6 Gbps

SAS (Serial Attached SCSI) is what is used in servers, because by design they are supposed to be the most reliable. They operate at higher voltages, due to which you can run SAS cables up to 33 feet (10 m), while SATA can only go up to 3.3 feet (1 m).

On the other hand, SATA drives are cheaper, and tend to have faster versions (10,000 rpm, 15,000 rpm, etc.).

Which should you choose? If you're building your own SAN, then you should be okay with SATA (If Backblaze can live with it, then so can you).

Then there's the choice of platter drives vs SSDs. Of course, SSDs are damn expensive ($650 for 1 TB, as opposed to $100 for 2 TB), so you won't be seeing them on SANs any time soon. A good drive to use is the Hitachi (HGST) Ultrastar 4 TB 7,200rpm drives with 64 MB cache.

The RAID controller

To get the 8 drives working like clockwork, you'll need a solid hardware RAID controller. The features you'll need to look out for are:

- The RAID controller should take 8 drives.
- The RAID controller must support RAID 6.
- It's SAS or SATA or both as the case may be.
- What PCI interface technology it supports.
- What operating system it supports (A server will most likely run on Linux).

Some brands that make RAID controllers are:

- LSI
- Areca
- Highpoint
- Adaptec
- ATTO
- And many more. One example that works in our case is the Areca ARC-1223-8* PCIe 2.0 x8 SATA/SAS RAID card. The SATA/SAS means it has a SAS connector and they sell SAS to SATA adapter cables so you can use SATA drives. Don't forget to add the cables.

The motherboard (backplane)

Now you have to buy a motherboard that supports PCIe 2.0 x 8 (you get the idea). Supermicro is a great value-for-money brand when it comes to motherboards and servers. Once you have the right motherboard, you can put a processor in it and a case around it. The size of the case (called chassis) depends on the size of the motherboard and the number of drives that have to sit in it. In addition, you'll need a slot for a boot drive (ideally SSD but it can be platter). 4U is a common chassis size for SANs.

Redundancy

You are already using RAID for data redundancy. But there are other kinds:

- Power supply – You will need a redundant power supply so that if one fails the SAN will still continue working. It will obviously be used in tandem with a UPS and surge protector.
- Battery Backup Module (BBM) for the RAID controller – prevents loss of data in cache if there's a power failure.
- Extra hard drives.
- ECC RAM.
- Another server (called a redundant server).

Let's revert back to the first line of this article. If the SAN is not mission critical, it is pointless. Redundancy is how you ensure a system continues to run in case of failure.

The server

All the parts together form the server. This server is designed for storage, and is called a storage array. The case is usually rack-mounted, so it can sit neatly in a well-cooled machine room (server room). It will have large fans to keep it cool from within. It will have a Xeon processor that is designed to work 24/7. Every part must be chosen to fit into the chassis and run smoothly. The cabling is done neatly so as to allow for ease of maintenance when (not if) things fail.

In fact, all the components of a server are designed to run 24/7, and if it fails, the redundant server takes over. This is how big websites stay online all the time. A busy post production facility that depends on its SAN needs no less. All it takes to bring a storage array to its knees is a faulty cable or dead processor – just *one thing* among the thousands that make it.

So far we've built our storage array, and it is time to go to more complex things. There are still a few parts that belong to the storage array that we have omitted, because they are easier to understand as part of their own divisions.

Workstation

Of course, the first requirement of a SAN is that you have multiple workstations, each like a young chick wanting a bite of the food that mother bird (SAN) is dangling before it. A workstation can be:

- An NLE (Adobe Premiere Pro, Avid Media Composer, FCP-X, etc.)
- A VFX system (Maya, Nuke, etc.)
- A full finishing system (Autodesk Smoke, Adobe After Effects, etc.)
- A color grading system (DaVinci Resolve, Adobe Speedgrade, etc.)
- A testing or viewing system
- A render machine or farm
- A sound editing or mixing system

And many more. I'm not going to include the costs of the workstation in our analysis. Some like Mac Pros, others like off-the-shelf products, and some just like to build their own. The workstation will need to connect to your network, but we'll get into that in the next section.

The Network

As I mentioned earlier a SAN is not just the storage array but also the network. It needs its own network, and that's how it is designed to work. The first thing we have to figure out is how fast the network should be.

How to estimate network bandwidth requirement

Continuing with our test case, you'll need a network capable of sustaining a rate of 600 MB/s (roughly 5 Gbps). There are five ways to go here:

1. 1 GbE Ethernet
2. 1 GbE multiplied (E.g. 4 x 1 GbE = 4 Gbps)
3. Fiber Channel (From 2 Gbps to 16 Gbps)
4. Fiber Channel over Ethernet (FCoE)
5. iSCSI over 10 GbE Ethernet, or other forms of 10 GbE

1 GbE is surprisingly usable in small post houses. Walter Biscardi of Biscardi Creative Media has written about his experiences with a 1 GbE network here. Also read his follow-up article on the limits of 1 GbE. 1 Gbps means about 125 MB/s. In real-world use, you can get up to 100 MB/s on a well-designed system. That's about 5 streams of Prores HQ at 220 Mbps. This is possibly the cheapest system to set up, but also the most restrictive. The moment you want to edit something like Red RAW or Prores 444 or uncompressed HD, etc., your system will come to a stand-still.

In the second case, what you're doing is trying to combine four 1GbE Ethernet to get 4 Gbps. This process is called link aggregation or bonding. Bob Zelin has written about it here. In our example case, this will not work. In fact, I don't recommend this architecture because it's 'temporary' and high-maintenance, having the same limitations as 1 GbE. You'll need to add more cables and switches as you scale.

Fiber Channel (FC) is the current king of the SAN world. It is also expensive, which is one reason why SANs are expensive. For what it's worth, FC works, because it has been around long enough and most vendors and integrators know how to make it work. Don't discount it from your considerations. FCoE, the fourth option, is not in widespread use.

iSCSI is newer technology, and a competitor to FC. It can be used with regular Ethernet technology. However, not everyone agrees that iSCSI is better, and FC still maintains top billing at least for now. Nobody really knows which way technology will swing. If you're currently looking to develop your own SAN, you need to live on the bleeding edge of technology – you need to pick a side that might be champions today but out of the league a few years from now, or the other way around. Today, that war is between FC vs 10 GbE Ethernet. For further research, you can start with the following:

- FC 8 Gbps vs 10 GbE Ethernet (PDF)
- Comparing Performance Between iSCSI, FCoE, and FC
- FC and Ethernet at the same time

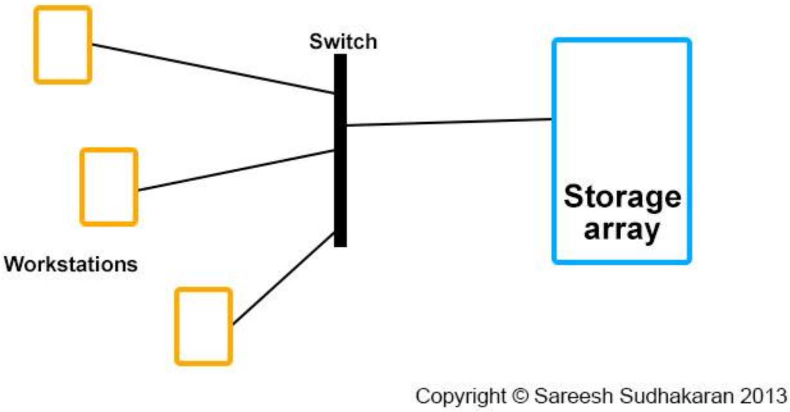
In our test case, we'll need an 8 GFC FC (8 Gbps, 800 MB/s throughput, 1.6 GB/s in full duplex) network. If we're going with 10 GbE, then that's what we have. In either case, we're aiming for a network with a bandwidth potential of 1 GB/s. I believe that any SAN you build today must be ready for at least this much bandwidth.

1 GB/s is 37 streams of Prores HQ or about 7 streams of 5K 3:1 Red Epic/2.5K uncompressed CinemaDNG footage. If you have compositing stations that demand more than this (editors rarely do), you could use DAS solutions for better speed. It does not make sense to spend money on the whole SAN just to feed one compositor. If you're a VFX facility, then it's another matter.

The network architecture

You have invited your guests (workstations and storage array) for dinner, now it's time to place them on the table. Your choice of network is dependent on the underlying architecture (FC, iSCSI, 10 GbE Ethernet, etc.). I won't go into detail, so we'll only consider one scenario.

Under Fiber Channel, you'll be arranging everything in what is called a Switched Fabric (FC-SW) architecture. Instead of stringing them along in one chain (if one fails the entire thing goes down), you connect each computer (workstation, node) to a Switch. Hence, 'switched' fabric. This is what it looks like:



This means you need a switch.

The switch

The switch is the mediator. Every node goes through the switch. All devices can talk to each other (if you allow it). The big four brands that make switches are:

- QLogic
- Cisco Systems
- Brocade
- Emulex

A completely redundant SAN will deploy two switches, so that if one fails the other continues the job as if nothing happened. One example of an 8-port 8 Gbps FC switch that isn't very expensive is the QLogic 3810 FC switch.

Important: Each node that connects to a switch gets the full bandwidth. E.g., an 8 Gbps switch will potentially give 8 Gbps to each workstation, though it is limited by the speed of the storage array (600 MB/s or 5 Gbps in our example). You can have a faster array (more disks) and then use more than one connector (if your array throughput is greater than 8 Gbps), i.e., two or more per switch.

The switch has multiple ports to connect cables to. So, we need cables.

Connecting the storage to our network

Do you know the problem with USB 2.0 or HDMI connectors, where there is more than one kind of connector for the same thing? Fiber Channel is similar, but far worse. There are many kinds of connectors to choose from, of which two are common for SANs:

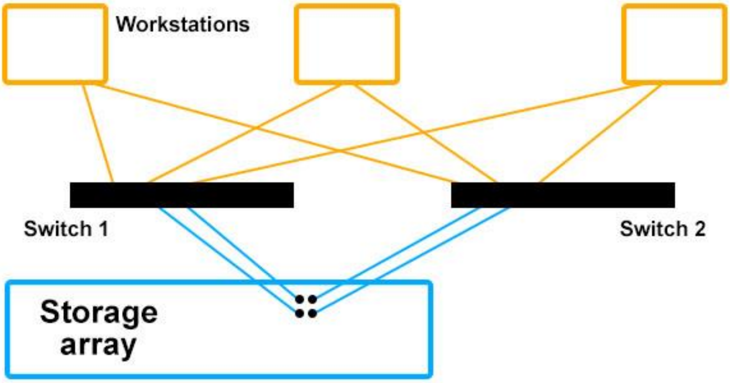
- SC Connector
- LC Connector

The LC connector is the one that you would probably be using. Many manufacturers prefer you use their entire SANs rather than just bits and pieces. This way, they can add some 'secret sauce' for which they charge more. It works, though. The QLogic switch I've referred to above is supposed to work with third-party SAN systems (They call it transparent routing).

In addition to this you also have the choice of wiring with copper or optical cable. Copper is used for short distances, whereas fiber optic cables are preferred for larger facilities where the server room might be in the basement and a workstation might on the third floor or whatever.

Then there's the choice of Singlemode fiber vs Multimode fiber. You'll be using the latter most likely because they are larger in diameter (50/125 and 62.5/125 in microns), faster and are only good for short distances. They are usually orange in color:

The choice of connector and cable type is entirely up to you, and the parts of your system. Even here, you have multiple choices that might make or break your system. You need to find a switch that you can control as a DIY system. If you have two switches, your network will look like this:



Copyright © Sareesh Sudhakaran 2013

Ah, but we're not done yet. We left out two things in the Storage and Workstation parts. Now it's time to add them.

The Host Bus Adapter or Network Interface Card

None of the fiber optic connectors are available on workstations. Where do you connect that cable then? You'll need to create a new port, which means you'll need a PCIe (or similar) card that plugs into the motherboard. This is called the Host Bus Adapter (HBA).

One HBA that supports full duplex 8 Gbps is the ATTO Celerity FC-82EN:

You'll need one for each workstation or storage array. The costs can add up quickly!

If you're on 10 GbE instead of FC, you don't need an HBA. But you need a Network Interface Card, or NIC. These tend to be cheaper than HBAs, sometimes by half or more. All the manufacturers that make switches also make HBAs. If you're using iSCSI, then the NIC won't be so simple. It will need a special hardware component called an 'initiator' that separates it (read: more expensive) from cheap NICs. There's no free lunch.

As you have seen, each and every hardware choice you make must be part of an overall plan. It is critical that you have the entire thing mapped out before you even begin to buy anything.

Software

So far, we've looked at parts that can be bought online and assembled together. That was the easy part. The hard part is finding the right software and using it to take advantage of the network as you've designed it.

This means, you *shouldn't* put together a SAN until you know which software to run and how to run it. So, what else is new?

The installation

Don't you wish these things were plug-and-play? They're not. When you install an HBA, you'll also need to install its software. Once each HBA is installed, you'll need to check if the firmware is up to date (just like you'd do on a GPU). Here's an article on how to set up HBAs.

Storage management

What about the storage? Your storage array might be one large logical drive, or might be partitioned. In either case, it needs the LUN treatment. The logical segments (LUNs) need to be configured and zoned so that a central server can control who can read and write to what. You don't want somebody writing over your source footage!

Anyway, when you buy a storage array, it might be limited by the number of LUNs you can create. This in turn affects the number of workstations that can connect to it.

The software that manages your storage array is critical. You may have heard of FreeNAS, which is a very popular open source solution for DIY NAS builds. Similarly, you need to find the SAN equivalent. The ideal software should work with HFS+ (Mac) and NTFS (Windows) file systems at the same time. One such operating system is Openfiler, which is both a NAS and SAN.

If you want to run your own operating systems (which is what you'll want, really), then you need a 'layer' that runs over the OS that makes each node see all files as one. Such a software is called a shared storage file system or SAN file system.

They run over the operating system and create a file-system that can be read by everybody. What this means, though, is that this software must be installed on *every* workstation in the network. These are usually licensed per node – the more the computers, the more the licensing. It adds up!

Some names are:

- metaSAN
- XSan
- StorNext
- Red Hat GFS
- VMWare VMFS
- QFS, from Sun

What you really need is a file system that has block-level access. Otherwise your whole SAN is a waste of time. XSan, which is compatible with StorNext, is free with OS X.

Metadata controller

A metadata controller is software that takes care of file locking among other things. E.g., if more than one person wants to read the same file at the same time, that's cool. But what if they want to write to the same file at the same time? That's not cool. Software like XSan and metaSAN includes a metadata controller.

It is not uncommon for metadata (this is not video metadata, mind you) to be transferred by a different network, or the Ethernet network that connects all the workstations at the facility. This keeps it 'safe'.

These are just the basic things. Configuring each software, making sure all the devices in the network can see and communicate with each other, and then ensuring the total bandwidth is utilized in the most efficient manner – these tasks are not easy, even for experienced pros.

For anyone trying to build a SAN, configuring software is the most nerve-racking and frustrating part. Be prepared.

Costs of building a SAN

If you were to put together the DIY SAN yourself, how much would it cost? Here's an incomplete spec sheet, but good enough to give you an indication of the ballpark:

E3 Quad-core Xeon Processor	$ 250
16 GB ECC RAM	$ 180
Motherboard	$ 180
Chassis with Redundant Power Supply	$ 700
RAID Controller	$ 470
Dual SSDs for OS	$ 200
8×4 TB SATA III drives	$ 2,800
Battery Backup Module	$ 120
HBA for FC	$ 1,200
Cables	$ 200
Sub-total	**$ 6,300**

I've used Supermicro parts. You could find things for cheaper of course, though I don't see how it would go below $5,000 for a 32 TB (24 TB available in RAID 6) storage array. I've also not included the price of the rack unit (this might weigh more than 60 lbs or 30 kg!), and a cool (air-conditioned) and quiet (the fans will be screaming loud) area. These are costs common to any storage array.

The costing is also excluding any switches or HBAs for the workstations, or proprietary software for running the SAN file system. The server will most likely run Linux. You could make the same thing with an Apple Mac Pro and Xsan, but the new Mac Pros are not rack units and don't have space for storage! Instead of building your own server, you might also invest in a ready-made solution like the HP Proliant DL380p Gen8 system. It will be more expensive than a DIY server but it comes with invaluable 3 years onsite warranty and service. As you can see clearly, the bulk of the expense involves the drives, the RAID controller with BBM and the HBA.

Three things that I haven't mentioned are:

1. You need a system that can be expanded without changing its internals. E.g., if you want to add a 24-bay

storage array to your SAN in a year’s time, you should be able to do that.

2. You need media readers for various media that will be ingested and copied directly into your array. E.g., LTO, SD, CF, SSD, SxS, and so on.
3. You might also need I/O devices that can directly record from different sources, like the ones made by Aja, Blackmagic Design, Matrox, etc.

On top of the server and storage array you have one or two switches and three (as per our test case) HBAs:

Switch x 2	$ 3,400
HBA x 3	$ 3,600
Cables	$ 200
Sub-total	**$ 7,000**

Running FC isn’t cheap. There’s also software, and the cost per seat. E.g., metaSAN costs about $995 per computer. That’s about $2,000 for our three workstations and server.

The total? Around **$15,000**, excluding the time it’ll take you to buy, install, troubleshoot, test and configure everything till it works like clockwork. And don’t forget to add the amount of time you’ll spend troubleshooting problems as it happens from time to time. If you’re the owner of a small and modest post facility, I’d say you might be worth between $200 to $500 an hour. If you spend 10 hours per year on your SAN, you’ve already spent between $2,000 to $5,000 (or whatever your time is worth). 10 hours isn’t much at all. If you’re totally new to this, you’ll be spending *weeks* on this!

This is why most people who try to build their own SANs are warned not to! This is what I figure as well, it is far better to invest in a turnkey solution with the help of an experienced systems integrator than to build your own SAN (assuming you can get it running, which is itself a tall order). Hope I’ve made my point.

To know how to research and buy shared storage for video, read this excellent whitepaper (email required) from Promax.

What if you're a networking jedi? Could you put together a SAN in a day? Sure. I figure the cost of a dependable DIY SAN should be well below $10,000 for 32 TB Fiber Channel. The skill level would be similar to programming an NLE application from scratch, or designing a Ford Model T from scratch.

The players

When you hunt for SAN storage solutions, you'd be wise to hunt for video production-centric vendors. If they have spent time working with various software that are typically found on our workstations, and know the specific requirements of video delivery over a network, their experience and expertise is worth their weight in gold.

Here are some of the major specialized SAN solutions in our industry:

- Promise VTrak A-Class
- Facilis Terrablock
- Small Tree GraniteSTOR
- Promax Platform
- Avid ISIS
- Studio Network Solutions EVO
- Maxx Digital ActiveRAID
- Grass Valley K2
- EditShare XStream (or Energy)
- Apace vStor
- dotHILL AssuredSAN

I'm pretty sure there are others, but 11 vendors is a good starting point. Let's see how they compare.

Comparison of ready-made SAN solutions

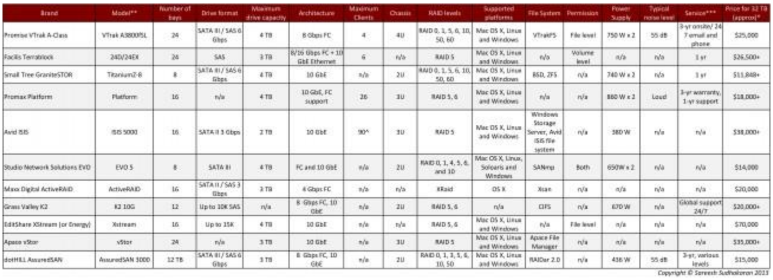
A detailed comparison is tough, if not impossible. Every storage solution has variations that change things – there are too many 'moving parts' so to speak. In addition, you really can't tell by the numbers who's good or bad. You will need the experience and advice of thoroughbred professionals to tell you the real-world pros and cons of each brand or model.

Furthermore, not all vendors ship to every country or region, and it might be difficult to find an integrator experienced with a model or brand not currently ‘in vogue’.

Therefore, all I’m offering is a brief comparison of these brands on the basis of:

- Price for 32 TB, our test case
- Number of clients supported
- Number of bays, drive type and maximum drive size
- FC or 10 GbE support
- Physical attributes
- Supported RAID levels
- Supported operating systems and file systems
- Permission levels

Here’s the comparison chart (*To view properly, go here: http://wolfcrow.com/blog/wp-content/uploads/2013/08/SANComparisonChart.jpg*):



Brand	Model**	Number of bays	Drive format	Maximum drive capacity	Architecture	Maximum Clients	Chassis	RAID levels	Supported platforms	File System	Permission	Power Supply	Typical noise level	Service***	Price for 32 TB (approx)*
Promise VTrak A-Class	VTrak A3800fSL	24	SATA III / SAS 6 Gbps	4 TB	8 Gbps FC	4	4U	RAID 0, 1, 5, 6, 10, 50, 60	Mac OS X, Linux and Windows	VTrakFS	File level	750 W x 2	55 dB	3-yr onsite/ 24 7 email and phone	$25,000
Facilis Terrablock	24D/24EX	24	SAS	3 TB	8/16 Gbps FC + 10 GbE Ethernet	6	n/a	RAID 5	Mac OS X, Linux and Windows	n/a	Volume level	n/a	n/a	1 yr	$26,500+
Small Tree GraniteSTOR	Titanium Z-8	8	SATA III / SAS 6 Gbps	4 TB	10 GbE	n/a	2U	RAID 0, 1, 5, 6, 10, 50, 60	Mac OS X, Linux and Windows	BSD, ZFS	n/a	740 W x 2	n/a	1 yr	$11,848+
Promax Platform	Platform	16	n/a	4 TB	10 GbE, FC support	26	3U	RAID 5, 6	Mac OS X, Linux and Windows	n/a	n/a	860 W x 2	Loud	3-yr warranty, 1-yr support	$18,000+
Avid ISIS	ISIS 5000	16	SATA II 3 Gbps	2 TB	10 GbE	90^	3U	RAID 5	Mac OS X, Linux and Windows	Windows Storage Server, Avid ISIS file system	n/a	380 W	n/a	n/a	$38,000+
Studio Network Solutions EVO	EVO 5	8	SATA III	4 TB	FC and 10 GbE	n/a	2U	RAID 0, 1, 4, 5, 6, and 10	Mac OS X, Linux, Solaris and Windows	SANmp	Both	650W x 2	n/a	n/a	$14,000
Maxx Digital ActiveRAID	ActiveRAID	16	SATA II / SAS 3 Gbps	3 TB	4 Gbps FC	n/a	n/a	XRaid	OS X	Xsan	n/a	n/a	n/a	n/a	$20,000
Grass Valley K2	K2 10G	12	Up to 10K SAS	n/a	8 Gbps FC, 10 GbE	n/a	2U	RAID 5, 6	n/a	CIFS	n/a	670 W	n/a	Global support 24/7	$20,000+
EditShare XStream (or Energy)	Xstream	16	Up to 15K	4 TB	10 GbE	n/a	n/a	RAID 5, 6	Mac OS X, Linux and Windows	n/a	File level	n/a	n/a	n/a	$70,000
Apace vStor	vStor	24	n/a	3 TB	10 GbE	n/a	3U	RAID 5	Mac OS X, Linux and Windows	Apace File Manager	n/a	n/a	n/a	n/a	$35,000+
dotHILL AssuredSAN	AssuredSAN 3000	12 TB	SATA III / SAS 6 Gbps	3 TB	8 Gbps FC, 10 GbE	n/a	2U	RAID 0, 1, 3, 5, 6, 10, 50	Mac OS X, Linux and Windows	RAIDar 2.0	n/a	436 W	55 dB	3-yr, various levels	$15,000

Copyright © Sareesh Sudhakaran 2013

Notes:

**Where price is not available for 32 TB, I’ve added or subtracted the drive price to get to 32 TB. These prices are from ‘here and there’ all over the Internet, and might be totally inaccurate or off-base. Please contact the manufacturer directly for correct prices.*

***There might be models within the brand that I’ve missed, and maybe there are better models available for our 32 TB SAN.*

****Service levels are different for different countries or regions. Extended or other SLAs might be applicable. Please speak to the vendor directly to know what applies in your case.*

^90 licenses included in price. If you want more, you pay more.

Important: *The information provided in the above chart might be inaccurate or plain wrong. I had to dig deep into the manufacturers' websites to get most of these. Those marked 'n/a' were not available. Don't take any buying or installation decisions based on this chart!*

What do you see? I can see the following general trends:

- The market is split between FC and 10 GbE.
- Some vendors provide switches with their arrays, among other features, and this skews the price. Don't compare the prices directly.
- Rack-mounted is the way to go. You will need a server room.
- By far the most popular RAID levels are 5 and 6.
- Many vendors have their own file systems. This is the key really. Anybody can buy and put together hardware. Making it work is an entirely different ball game.
- Most of them offer redundant power supplies.
- You're staring at a price much higher than $20,000 all said and done. A mid-level post facility could invest in a SAN from anywhere between $20K to $100K.

Takeaways:

- The person or organization that will build a SAN for you is called a Systems Integrator.
- The starting price for a powerful and dependable SAN is somewhere in the region of $20,000 or more.
- You need to commit to either Fiber Channel or 10 GbE, and it's a big commitment!
- The most popular RAID levels for SANs are RAID 5 and RAID 6

We’re done.

Look at how far we’ve come! We started with learning about files, computers and servers.

Before we knew it, we got embroiled in a murder mystery, thanks to which we learnt what RAID is. We learnt what various RAID levels meant, and what parts we needed to put together a RAID array.

We left our detectives behind, put on our rocket boosters, and discovered the best RAID level for editing. While were were flying high we also learnt how to choose a DAS for our video work.

Finally, we moved into the stratosphere with an understanding of the NAS and the SAN, and we looked at how one can put together a SAN if they wanted to.

Seriously, as far as video storage for post production is concerned, there’s very little we haven’t covered.

I hope this detailed primer has answered all your questions (even the ones you didn’t know you were supposed to ask!) about RAID and its use for video editing and post production. The additional knowledge will serve you well when it’s time to grow and move up your career ladder.

Never be AFRAID again.

THE END