

Data Storage - When reality clashes with theory

Lars Nielsen, *Postdoctoral Researcher*

Aarhus University - Department of Electrical and Computer Engineering

Researchers are often guilty of only celebrating
the final success

I want to present the carnage that can happen
behind the scene

To create a file system based on a new emerging technology called generalised deduplication

We constantly ran out of disk space

But all analytics tools told us that we had ample
space left

So what was wrong?

- Our code
- Something outside our code

- Our code
 - Something outside our code
- Our code
 - Full code analysis revealed exactly nothing
 - Follow systems calls, revealed. . . well, nothing

- Our code
- Something outside our code
- Something outside our code
 - We used EXT4 as an external storage
 - It has something called a directory index

- An index of all files in a folder
- It is (most often) loaded in to memory
- Used to speed up look up operations
 - `ls`, `stat`, etc.

- An index of all files in a folder
- It is (most often) loaded in to memory
- Used to speed up look up operations
 - `ls`, `stat`, etc.

So what is the problem?

EXT4 can in theory stored an "unlimited"
amount of files

Directory Index has a different limit

EXT4 can in theory stored an "unlimited"
amount of files

Directory Index has a different limit

Relax it gets worse the limit varies between Linux
distributions and even version.

- Directory Index limits a system that should be limited
- The limit is not well defined

OS	Observed limit
Ubuntu 18.04	12mio
Ubuntu 18.04 server	16mio
Fedora 31	32mio
Fedora 31 Server	32mio
Ubuntu 20.04	20mio
Fedora 33	64mio

- Disable the directory index
 - That is an option
 - It solves the issue
 - But it heavily damages performance

- Disable the directory index
 - That is an option
 - It solves the issue
 - But it heavily damages performance
- So let us hack away around
- We will turn the disadvantage of the directory index into an advantage
- It is only a "solution".

- Our files all have an SHA-1 identifier
 - 20 bytes or converted to hexadecimal string 40 bytes

Let us use that to create a grouping system.

- Our files all have an SHA-1 identifier
 - 20 bytes or converted to hexadecimal string 40 bytes

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Let us use that to create a grouping system.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

If hash for files shares the 2 first characters, they belong to the same *major group*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

and say files that share the first 4 characters belongs to the same *minor group*

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

- Then, we create a folder for each major group
 - In the major group, we create folders for all minor groups
 - Then, we place all files belong to that specific minor group in a folder
- Make structure picture

What this does is:

- Reduce the probability of hitting the directory index limit
 - all though still present
- But we retain the power of the directory index
- With minimum damage to storage usage -4kB (minimum per folder)
- Zero impact on RAM usage
- Work also for EXT2, EXT3, and ZFS

- Keep a registry in memory of all files stored
- But it increases the RAM consumption of the file system
 - and do you really want your file system to play Google Chrome?

Thank you for your attention
