Admins - Demande #3977

Investiguer les problèmes d'IO sur l'infra

31/10/2019 14:33 - Quentin Gibeaux

Statut:	En cours de traitement	**Début:**	31/10/2019
Priorité:	Normale	**Echéance:**	
Assigné à:	Romain H.	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0.00 heure
Version cible:	Avril 2024	**Temps passé:**	0.00 heure
Difficulté:	2 Facile		

Description

Sur certaines VM qui y a des gros problèmes d'IOWait (par exemple nextcloud).

Hypothèse : sparse file dans QCOW + goulot l'étranglement DRBD

Historique

#1 - 03/11/2019 10:54 - François Poulain

Sur Calamus

test de débit

```
(April) root@calamus:~# dd if=/dev/zero of=/root/testfile bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 9.89863 s, 108 MB/s
(April) root@calamus:~# dd if=/dev/zero of=/root/testfile bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 10.4263 s, 103 MB/s
(April) root@calamus:~# dd if=/dev/zero of=/root/testfile bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 9.49281 s, 113 MB/s
```

C'est plutôt bon.

test de latence

```
(April) root@calamus:~# dd if=/dev/zero of=/root/testfile bs=512 count=1000 oflag=dsync
1000+0 records in
1000+0 records out
512000 bytes (512 kB, 500 KiB) copied, 143.821 s, 3.6 kB/s
(April) root@calamus:~# dd if=/dev/zero of=/root/testfile bs=512 count=1000 oflag=dsync
1000+0 records in
1000+0 records out
512000 bytes (512 kB, 500 KiB) copied, 145.381 s, 3.5 kB/s
```

C'est franchement moins bon que ce que j'ai sur des serveurs perso (25 - 40 s sur des machine de gamme comparable).

#2 - 03/11/2019 11:14 - François Poulain

Sur DRBD / Calamus

test de débit

```
(April) root@calamus:~# dd if=/dev/zero of=/var/lib/libvirt/calamus/testfile bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 12.7076 s, 84.5 MB/s
(April) root@calamus:~# dd if=/dev/zero of=/var/lib/libvirt/calamus/testfile bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 11.8623 s, 90.5 MB/s
```

C'est plutôt bon.

test de latence

```
(April) root@calamus:~# dd if=/dev/zero of=/var/lib/libvirt/calamus/testfile bs=512 count=1000 oflag=dsync
1000+0 records in
1000+0 records out
512000 bytes (512 kB, 500 KiB) copied, 138.375 s, 3.7 kB/s
(April) root@calamus:~# dd if=/dev/zero of=/var/lib/libvirt/calamus/testfile bs=512 count=1000 oflag=dsync
1000+0 records in
1000+0 records out
512000 bytes (512 kB, 500 KiB) copied, 137.983 s, 3.7 kB/s
```

C'est convenable eu égard au test précédent.

#3 - 03/11/2019 11:33 - François Poulain

Sur Lamp

test de débit

```
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=1G count=1 oflag=dsync
Killed
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=1G count=1 oflag=dsync
Killed
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=100M count=1 oflag=dsync
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 9,85485 s, 10,6 MB/s
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=100M count=1 oflag=dsync
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,93169 s, 35,8 MB/s
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=100M count=1 oflag=dsync
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 3,64552 s, 28,8 MB/s
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=100M count=1 oflag=dsync
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 1,88194 s, 55,7 MB/s
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=100M count=1 oflag=dsync
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,11818 s, 49,5 MB/s
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=100M count=1 oflag=dsync
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 1,5165 s, 69,1 MB/s
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=100M count=1 oflag=dsync
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,13045 s, 49,2 MB/s
```

C'est questionnant.

test de latence

```
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=512 count=1000 oflag=dsync
1000+0 records in
1000+0 records out
512000 bytes (512 kB, 500 KiB) copied, 139,113 s, 3,7 kB/s
(April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=512 count=1000 oflag=dsync
1000+0 records in
1000+0 records out
512000 bytes (512 kB, 500 KiB) copied, 130,378 s, 3,9 kB/s
```

C'est convenable eu égard au test précédent.

#4 - 03/11/2019 11:42 - François Poulain

Sur Lamp-data

test de débit

```
(April) root@lamp:~# for i in $(seq 10); do dd if=/dev/zero of=/var/www/testfile bs=100M count=1 oflag=dsync;
done
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,56997 s, 40,8 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 3,04714 s, 34,4 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,01295 s, 52,1 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,85281 s, 36,8 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 3,27431 s, 32,0 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 4,93176 s, 21,3 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,14846 s, 48,8 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,75109 s, 38,1 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 1,54045 s, 68,1 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,13205 s, 49,2 MB/s
(April) root@lamp:~# dd if=/dev/zero of=/var/www/testfile bs=1G count=1 oflag=dsync
1+0 records in
1+0 records out
1073741824 bytes (1,1 GB, 1,0 GiB) copied, 139,388 s, 7,7 MB/s
```

C'est questionnant.

test de latence

```
(April) root@lamp:~# dd if=/dev/zero of=/var/www/testfile bs=512 count=1000 oflag=dsync
1000+0 records in
1000+0 records out
512000 bytes (512 kB, 500 KiB) copied, 153,41 s, 3,3 kB/s
```

C'est convenable eu égard au test précédent.

#5 - 03/11/2019 11:47 - François Poulain

Le kill oom:

```
[69905.342766] mysqld invoked oom-killer: gfp_mask=0x6200ca(GFP_HIGHUSER_MOVABLE), nodemask=(null), order=0, o
om_score_adj=0
[69905.342767] mysqld cpuset=/ mems_allowed=0
[69905.342770] CPU: 1 PID: 28237 Comm: mysqld Not tainted 4.19.0-6-amd64 #1 Debian 4.19.67-2+deb10u1
[69905.342771] Hardware name: QEMU Standard PC (i440FX + PIIX, 1996), BIOS 1.10.2-1 04/01/2014
[69905.342771] Call Trace:
[69905.360288]  dump_stack+0x5c/0x80
[69905.370449]  dump_header+0x6b/0x283
[69905.371082]  ? do_try_to_free_pages+0x2ec/0x370
[69905.371085]  oom_kill_process.cold.30+0xb/0x1cf
[69905.371086]  ? oom_badness+0x23/0x140
[69905.371088]  out_of_memory+0x1a5/0x430
[69905.371090]  __alloc_pages_slowpath+0xbd8/0xcb0
[69905.371194]  __alloc_pages_nodemask+0x28b/0x2b0
[69905.371196]  filemap_fault+0x3bd/0x780
[69905.371300]  ? alloc_set_pte+0x49e/0x560
[69905.371301]  ? filemap_map_pages+0x139/0x3a0
[69905.371318]  ext4_filemap_fault+0x2c/0x40 [ext4]
[69905.371320]  __do_fault+0x36/0x130
[69905.371408]  __handle_mm_fault+0xe6c/0x1270
```

```
[69905.371528]  handle_mm_fault+0xd6/0x200
[69905.371530]  __do_page_fault+0x249/0x4f0
[69905.371536]  ? async_page_fault+0x8/0x30
[69905.371537]  async_page_fault+0x1e/0x30
[69905.371797] RIP: 0033:0x563a3916f030
[69905.371802] Code: Bad RIP value.
[69905.371803] RSP: 002b:00007f212c0f54d8 EFLAGS: 00010246
[69905.371804] RAX: 00007f20bc0071e8 RBX: 00007f20bc000bb8 RCX: 0000000000006018
[69905.371805] RDX: 0000000000006000 RSI: 0000000000000001 RDI: 00007f20bc004800
[69905.371806] RBP: 00007f212c0f5d70 R08: 0000000000000000 R09: 00006bdb5ffddf01
[69905.371807] R10: 0019c191079cec5d R11: 0000000000000001 R12: 0000000000004000
[69905.371807] R13: 0000000000000000 R14: 00007f20bc002638 R15: 0000563a39eddf80
[69905.371843] Mem-Info:
[69905.371847] active_anon:101094 inactive_anon:102278 isolated_anon:32
                active_file:180 inactive_file:203 isolated_file:32
                unevictable:0 dirty:0 writeback:0 unstable:0
                slab_reclaimable:10474 slab_unreclaimable:15634
                mapped:2409 shmem:2903 pagetables:2737 bounce:0
                free:12108 free_pcp:349 free_cma:0
[69905.371851] Node 0 active_anon:404376kB inactive_anon:409112kB active_file:720kB inactive_file:812kB unevic
table:0kB isolated(anon):128kB isolated(file):128kB mapped:9636kB dirty:0kB writeback:0kB shmem:11612kB shmem_
thp: 0kB shmem_pmdmapped: 0kB anon_thp: 2048kB writeback_tmp:0kB unstable:0kB all_unreclaimable? no
[69905.371851] Node 0 DMA free:4492kB min:732kB low:912kB high:1092kB active_anon:5808kB inactive_anon:4140kB
active_file:0kB inactive_file:0kB unevictable:0kB writepending:0kB present:15992kB managed:15908kB mlocked:0kB
 kernel_stack:12kB pagetables:116kB bounce:0kB free_pcp:0kB local_pcp:0kB free_cma:0kB
[69905.371855] lowmem_reserve[]: 0 940 940 940 940
[69905.371858] Node 0 DMA32 free:43940kB min:44320kB low:55400kB high:66480kB active_anon:398540kB inactive_an
on:404912kB active_file:1080kB inactive_file:596kB unevictable:0kB writepending:0kB present:1032036kB managed:
994412kB mlocked:0kB kernel_stack:3044kB pagetables:10832kB bounce:0kB free_pcp:1396kB local_pcp:1320kB free_c
ma:0kB
[69905.371864] lowmem_reserve[]: 0 0 0 0 0
[69905.371867] Node 0 DMA: 41*4kB (UME) 9*8kB (UME) 20*16kB (UME) 21*32kB (UME) 13*64kB (UME) 7*128kB (UME) 4*
256kB (UM) 1*512kB (M) 0*1024kB 0*2048kB 0*4096kB = 4492kB
[69905.371877] Node 0 DMA32: 377*4kB (UMEH) 522*8kB (UMEH) 406*16kB (UMEH) 335*32kB (UMEH) 171*64kB (UMEH) 47*
128kB (UE) 12*256kB (UMEH) 1*512kB (M) 1*1024kB (H) 0*2048kB 0*4096kB = 44468kB
[69905.371888] Node 0 hugepages_total=0 hugepages_free=0 hugepages_surp=0 hugepages_size=2048kB
[69905.371889] 5079 total pagecache pages
[69905.371890] 1701 pages in swap cache
[69905.371891] Swap cache stats: add 1281557, delete 1279827, find 1496152/1817655
[69905.371891] Free swap  = 0kB
[69905.371892] Total swap = 974844kB
[69905.371892] 262007 pages RAM
[69905.371893] 0 pages HighMem/MovableOnly
[69905.371893] 9427 pages reserved
[69905.371894] 0 pages hwpoisoned
[69905.371894] Tasks state (memory values in pages):
[69905.371895] [  pid  ]   uid  tgid total_vm      rss pgtables_bytes swapents oom_score_adj name
[69905.371901] [    256]     0   256    10125        0    90112      342             0 systemd-journal
[69905.371903] [    274]     0   274     5580        2    65536      340         -1000 systemd-udevd
[69905.371905] [    508]   110   508     1608        1    53248      151          -500 nrpe
[69905.371907] [    509]     0   509     1378        1    53248       62             0 cron
[69905.371909] [    515]     0   515      581        0    40960       34             0 acpid
[69905.371910] [    519]     0   519    56495        0    90112      375             0 rsyslogd
[69905.371912] [    521]     0   521     4878        0    73728      258             0 systemd-logind
[69905.371914] [    523]   105   523     2175        7    61440      133          -900 dbus-daemon
[69905.371916] [    602]   108   602    19117       14    69632      132             0 ntpd
[69905.371917] [    609]     0   609     1481        0    45056       33             0 agetty
[69905.371919] [    706]   111   706   663429      971  1011712    87555             0 mysqld
[69905.371921] [    786]     0   786    10868        0    77824      210             0 master
[69905.371923] [    788]   109   788    10906        0    77824      234             0 qmgr
[69905.371925] [   1168]     0  1168     3963        2    73728      216         -1000 sshd
[69905.371927] [  14432]     0 14432    57959       17   200704     1876             0 php-fpm7.3
[69905.371935] [  10554]    33 10554    60535        0   258048     3215             0 php-fpm7.3
[69905.371937] [  10780]    33 10780    60497        1   258048     3135             0 php-fpm7.3
[69905.371938] [  10781]    33 10781    60464        0   258048     3150             0 php-fpm7.3
[69905.371940] [  21652]     0 21652     5763        1    94208     1911             0 sshd
[69905.371942] [  21655]     0 21655     5327        0    86016      387             0 systemd
[69905.371944] [  21657]     0 21657    42937        0   106496      720             0 (sd-pam)
[69905.371946] [  21671]     0 21671    11351        1   139264     9588             0 rsync
[69905.371948] [  15671]     0 15671     4154        1    73728      286             0 sshd
[69905.371949] [  15677]     0 15677     2397        3    61440      695             0 bash
[69905.371951] [  17359]     0 17359    71639       57   307200     2360             0 apache2
[69905.371953] [  19764]    33 19764    76896      496   507904     5504             0 apache2
[69905.371955] [  19766]    33 19766    76574     1373   495616     4664             0 apache2
[69905.371956] [  19773]    33 19773    76385      350   495616     5150             0 apache2
```

```
[69905.371958] [  19943]    33 19943    77011     2023   495616     4679             0 apache2
[69905.371960] [  20262]    33 20262    95015      589   499712     5105             0 apache2
[69905.371965] [  20470]    33 20470    76279      430   487424     4847             0 apache2
[69905.371967] [  20725]    33 20725    76457      379   475136     5030             0 apache2
[69905.371968] [  21754]   109 21754    10875        0    77824      223             0 pickup
[69905.371970] [  23725]    33 23725    76745     2115   483328     4536             0 apache2
[69905.371972] [  25218]     0 25218     4156        0    69632      299             0 sshd
[69905.371974] [  25224]     0 25224     2397        1    49152      696             0 bash
[69905.371976] [  25432]     0 25432     4156        0    73728      288             0 sshd
[69905.371978] [  25438]     0 25438     2397        1    65536      697             0 bash
[69905.371979] [  25817]    33 25817    75263     2240   462848     2974             0 apache2
[69905.371981] [  25960]     0 25960     1401        0    53248       36             0 tail
[69905.371983] [  27350]    33 27350    75382     2316   450560     3359             0 apache2
[69905.371985] [  28149]     0 28149     4156        0    73728      278             0 sshd
[69905.371987] [  28155]     0 28155     2397        0    61440      700             0 bash
[69905.371989] [  28169]   109 28169    10875        0    81920      203             0 showq
[69905.371991] [  28214]     0 28214     4154        0    69632      276             0 sshd
[69905.371992] [  28220]     0 28220     2398       49    53248      650             0 bash
[69905.371994] [  28231]     0 28231   263550   189891  2154496    72277             0 dd
[69905.371996] [  28234]    33 28234    73014     1457   327680     2050             0 apache2
[69905.371998] [  28235]    33 28235    71706     1452   311296     1828             0 apache2
[69905.371999] [  28236]    33 28236    71647       54   286720     2357             0 apache2
[69905.372000] Out of memory: Kill process 28231 (dd) score 529 or sacrifice child
[69905.372054] Killed process 28231 (dd) total-vm:1054200kB, anon-rss:759560kB, file-rss:4kB, shmem-rss:0kB
```

#6 - 03/11/2019 12:01 - Christian P. Momon

Hypothèse des trous dans le QCow.

https://fr.wikipedia.org/wiki/Qcow2 :

```
Qcow signifie QEMU Copy On Write et utilise une stratégie d'optimisation
de stockage sur disque qui retarde l'allocation de stockage jusqu'à ce que
cela soit réellement nécessaire.
```

Nos fichiers Qcow2 sont troués (sparse file), voir ci-après la colonne de gauche et celle de droite.

Question : à chaque fois que le fichier qcow2 a besoin de grandir, une allocation de stockage est faite et pénalise les performances.

- Est-ce suffisant pour expliquer nos ralentissement ?
- Cela fragmente-t-il les partitions et augmente-t-il les déplacements de la tête de lecture de nos disques à plateaux ?
- Faudra-t-il privilégier les disques SSD pour les prochains serveurs physiques ?

```
(April) root@calamus:/var/lib/libvirt/calamus# ll -hisa
total 503G
 2 4.0K drwxr-xr-x  3 root         root         4.0K Nov  3 10:19 ./
 2 4.0K drwxr-xr-x 10 root         root         4.0K Oct  2 11:06 ../
13 1.2G -rw-------  1 libvirt-qemu libvirt-qemu 1.2G Nov  3 10:35 adl-mysql.qcow2
12  16G -rw-r--r--  1 libvirt-qemu libvirt-qemu  26G Nov  3 10:38 adl.qcow2
17 1.7G -rw-r--r--  1 libvirt-qemu libvirt-qemu 8.6G Nov  3 10:34 agir-data.qcow2
16 9.2G -rw-r--r--  1 libvirt-qemu libvirt-qemu  33G Nov  3 10:38 agir.qcow2
18  11G -rw-r--r--  1 libvirt-qemu libvirt-qemu  23G Nov  3 10:38 bots.qcow2
25 9.3G -rw-r--r--  1 libvirt-qemu libvirt-qemu  23G Nov  3 10:37 candidatsbe.qcow2
22  11G -rw-r--r--  1 libvirt-qemu libvirt-qemu  12G Nov  3 10:38 candidatsfr-data.qcow2
21  22G -rw-r--r--  1 libvirt-qemu libvirt-qemu  33G Nov  3 10:38 candidatsfr.qcow2
15 240G -rw-r--r--  1 libvirt-qemu libvirt-qemu 512G Nov  3 10:38 lamp-data.qcow2
14  31G -rw-r--r--  1 libvirt-qemu libvirt-qemu  42G Nov  3 10:38 lamp.qcow2
11  16K drwx------  2 root         root          16K Mar  7  2016 lost+found/
24 2.8G -rw-r--r--  1 libvirt-qemu libvirt-qemu 9.3G Nov  3 10:37 pad-data.qcow2
23  17G -rw-r--r--  1 libvirt-qemu libvirt-qemu  31G Nov  3 10:37 pad.qcow2
32  22G -rw-------  1 libvirt-qemu libvirt-qemu  31G Nov  3 10:36 photos-data.qcow2
29  18G -rw-------  1 libvirt-qemu libvirt-qemu  28G Nov  3 10:38 photos.qcow2
31  14G -rw-r--r--  1 libvirt-qemu libvirt-qemu  34G Nov  3 09:14 pouet-data.qcow2
30  28G -rw-r--r--  1 libvirt-qemu libvirt-qemu  43G Nov  3 10:38 pouet.qcow2
20  13G -rw-r--r--  1 libvirt-qemu libvirt-qemu  21G Nov  3 03:01 republique-numerique-data.qcow2
19 6.8G -rw-r--r--  1 libvirt-qemu libvirt-qemu  23G Nov  3 10:38 republique-numerique.qcow2
27  19G -rw-r--r--  1 libvirt-qemu libvirt-qemu  27G Nov  3 00:55 scm-data.qcow2
26 6.5G -rw-r--r--  1 libvirt-qemu libvirt-qemu  25G Nov  3 10:37 scm.qcow2
28 9.8G -rw-------  1 libvirt-qemu libvirt-qemu  23G Nov  3 10:38 webchat.qcow2
```

#7 - 16/02/2020 08:21 - Christian P. Momon

- Statut changé de Nouveau à En cours de traitement

- Assigné à mis à Christian P. Momon

- Version cible changé de Backlog à Février 2020

François Poulain a écrit :

> (April) root@lamp:~# dd if=/dev/zero of=/root/testfile bs=1G count=1 oflag=dsync
> Killed
> […]
> [69905.371891] Free swap = 0kB
> [69905.371892] Total swap = 974844kB
> [69905.371895] [pid]   uid tgid total_vm   rss pgtables_bytes swapents oom_score_adj name
> [69905.371994] [28231]   0 28231   263550   189891 2154496   72277   0 dd
> [69905.372000] Out of memory: Kill process 28231 (dd) score 529 or sacrifice child
> [69905.372054] Killed process 28231 (dd) total-vm:1054200kB, anon-rss:759560kB, file-rss:4kB, shmem-rss:0kB

Pourquoi dd mange-t-il le swap ? Parce qu'on ne lui laisse pas le choix.
En effet, sur la vm lamp, il y a 1 Go de RAM et 1 Go de swap.
Or, la mémoire est dimensionnée pour correspondre à l'espace nécessaire des processus utilisés donc la mémoire vive disponible est faible.
À cela, via le bs=1G, on demande à dd de consommer 1 Go de plus, soit la taille du swap.
Conclusion : dd est obligé d'aller puiser dans le swap et comme le swap est inférieur à son besoin alors oom kill.

De fait, l'utilisation du swap fausse les tests. Donc il y a nécessité à respecter la mémoire disponible afin de ne pas du tout utiliser le swap.
Ou alors, augmenter temporairement la mémoire vive de la vm pour la période des tests.

#8 - 17/02/2020 07:58 - Christian P. Momon

J'ai posé la question :

> Cela fragmente-t-il les partitions et augmente-t-il les déplacements de la tête de lecture de nos disques à plateaux ?

Pour y répondre, j'ai cherché à mesurer l'état de fragmentation de nos partitions.

Fragmentation de la partition Calamus :

```
(April) root@calamus:/var/lib/libvirt/calamus# e4defrag -c .
e4defrag 1.44.5 (15-Dec-2018)
<Fragmented files>                            now/best       size/ext
1. /var/lib/libvirt/calamus/agir.qcow2        7276/5          1350 KB
2. /var/lib/libvirt/calamus/scm.qcow2         3103/4          2244 KB
3. /var/lib/libvirt/calamus/agir-data.qcow2
                                               510/1          3521 KB
4. /var/lib/libvirt/calamus/webchat.qcow2
                                              3242/6          3734 KB
5. /var/lib/libvirt/calamus/republique-numerique.qcow2
                                              1262/4          5875 KB

 Total/best extents                           42513/331
 Average size per extent                      15750 KB
 Fragmentation score                          0
 [0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
 This directory (.) does not need defragmentation.
 Done.

(April) root@calamus:/var/lib/libvirt/calamus# e4defrag -c agir.qcow2
e4defrag 1.44.5 (15-Dec-2018)
<File>                                        now/best       size/ext
agir.qcow2                                    7276/5          1350 KB

 Total/best extents                           7276/5
 Average size per extent                      1350 KB
 Fragmentation score                          2
 [0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
 This file (agir.qcow2) does not need defragmentation.
 Done.
(April) root@calamus:/var/lib/libvirt/calamus# e4defrag -c agir-data.qcow2
e4defrag 1.44.5 (15-Dec-2018)
<File>                                        now/best       size/ext
agir-data.qcow2                               510/1           3521 KB

 Total/best extents                           510/1
 Average size per extent                      3521 KB
 Fragmentation score                          1
```

```
 [0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
 This file (agir-data.qcow2) does not need defragmentation.
 Done.
```

Fragmentation de la partition Virola :

```
(April) root@virola:/var/lib/libvirt/virola# e4defrag -c .
e4defrag 1.44.5 (15-Dec-2018)
<Fragmented files>                              now/best       size/ext
1. /var/lib/libvirt/virola/lsd.qcow2             44/1             77 KB
2. /var/lib/libvirt/virola/dns.qcow2           3191/4           2233 KB
3. /var/lib/libvirt/virola/sympa.qcow2         4303/8           3448 KB
4. /var/lib/libvirt/virola/cms-dev.qcow2       3130/8           4993 KB
5. /var/lib/libvirt/virola/sympa-data.qcow2
                                               5759/18          6341 KB

 Total/best extents                              28753/179
 Average size per extent                         12376 KB
 Fragmentation score                             0
 [0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
 This directory (.) does not need defragmentation.
 Done.
```

Fragmentation sur la vm agir :

```
(April) root@agir:~# lsblk
NAME                 MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sr0                   11:0    1 1024M  0 rom
vda                  254:0    0   20G  0 disk
└─vda1               254:1    0   20G  0 part
  ├─vg_agir-root     253:0    0  3.7G  0 lvm  /
  ├─vg_agir-swap     253:1    0  952M  0 lvm  [SWAP]
  ├─vg_agir-tmp      253:4    0  952M  0 lvm  /tmp
  └─vg_agir-var      253:5    0  3.7G  0 lvm  /var
vdb                  254:16   0    8G  0 disk
├─vg_agir_data-mysql 253:2    0  800M  0 lvm  /var/lib/mysql
└─vg_agir_data-git   253:3    0    2G  0 lvm  /srv/git

(April) root@agir:~# e4defrag -c / /var/ /var/lib/mysql/ /srv/git/
e4defrag 1.44.5 (15-Dec-2018)
<Fragmented files>                              now/best       size/ext
1. /etc/.git/logs/HEAD                            7/1              4 KB
2. /etc/.git/logs/refs/heads/master               7/1              4 KB
3. /srv/common/.git/logs/HEAD                     3/1              4 KB
4. /etc/aliases.db                                3/1              4 KB
5. /root/.bash_history                           17/1              5 KB

 Total/best extents                              47646/46925
 Average size per extent                         54 KB
 Fragmentation score                             1
 [0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
 This directory (/) does not need defragmentation.
 Done.

<Fragmented files>                              now/best       size/ext
1. /var/log/wtmp.1                               13/1              4 KB
2. /var/log/wtmp                                 59/1              4 KB
3. /var/log/user.log.1                            8/1              4 KB
4. /var/log/apache2/error.log.1                   7/1              4 KB
5. /var/log/mysql/error.log                       6/1              4 KB

 Total/best extents                              41394/40806
 Average size per extent                         38 KB
 Fragmentation score                             1
 [0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
 This directory (/var/) does not need defragmentation.
 Done.

<Fragmented files>                              now/best       size/ext
1. /var/lib/mysql/mysql/help_keyword.MYI          2/1              8 KB
2. /var/lib/mysql/mysql/help_relation.MYI
                                                  2/1             10 KB
3. /var/lib/mysql/mysql/help_topic.MYI            2/1             10 KB
```

```
4. /var/lib/mysql/redmine/attachments.ibd
                                                27/1              26 KB
5. /var/lib/mysql/mysql/transaction_registry.ibd
                                                 5/1              28 KB

 Total/best extents                             326/195
 Average size per extent                        732 KB
 Fragmentation score                            2
 [0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
 This directory (/var/lib/mysql/) does not need defragmentation.
 Done.

<Fragmented files>                             now/best       size/ext
1. /srv/git/catalibre/objects/90/e427ace6eeaec56a2267cbbd4f5464b65d2f1c
                                                 3/1            2090 KB
2. /srv/git/catalibre/objects/fa/728ea15506a0aa1f50b8f528d39466c87e961d
                                                 2/1            2224 KB
3. /srv/git/agirstatool.git/objects/pack/pack-d65782fbe6e91f69ebf72fe6b7ad11601936cfd9.pack
                                                 2/1            2992 KB
4. /srv/git/catalibre/objects/97/55e6821e6cd4432e315488f9d86c7418e3f1d1
                                                 2/1            3070 KB
5. /srv/git/catalibrassociation/objects/pack/pack-85f14c8974e10c71e351c460471a964aaf030728.pack
                                                 2/1            5440 KB

 Total/best extents                             11579/11561
 Average size per extent                        113 KB
 Fragmentation score                            0
 [0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
 This directory (/srv/git/) does not need defragmentation.
 Done.
```

Si l'on s'en tient à ces données alors on peut conclure que la fragmentation est négligeable. Voilà de quoi être déconcerté.

Des gens pensent que ext4 est vraiment bien conçu : https://unix.stackexchange.com/questions/23009/fragmentation-and-ext4

```
The kernel caches the writes and lazily flushes them to disk in the background,
allocating disk space as it does so in such a way that it minimizes fragmentation.
In other words, you're over thinking things -- don't worry about it.

More specifically when it does to to flush some dirty cache buffers,
ext4 goes to allocate enough disk space to hold all of the dirty buffers in the cache,
as well as reserving additional space for further growth.
```

Les faits tendent à démontrer que c'est le cas. Si c'est vrai alors je suis vraiment très impressionné.

Conclusion : a priori, l'utilisation de partition qcow2 trouées génère peu de fragmentation et n'expliquerait pas les lenteurs I/O du SI.

#9 - 26/02/2020 22:24 - Quentin Gibeaux

- Version cible changé de Février 2020 à Mars 2020

#10 - 25/03/2020 21:37 - Quentin Gibeaux

- Assigné à changé de Christian P. Momon à Edouard Dausque

#11 - 25/03/2020 21:37 - Quentin Gibeaux

- Version cible changé de Mars 2020 à Avril 2020

#12 - 29/04/2020 21:23 - Quentin Gibeaux

- Version cible changé de Avril 2020 à Mai 2020

#13 - 27/05/2020 22:06 - Quentin Gibeaux

- Version cible changé de Mai 2020 à Juin 2020

#14 - 01/07/2020 21:50 - Quentin Gibeaux

- Version cible changé de Juin 2020 à Été 2020

#15 - 02/09/2020 21:31 - Quentin Gibeaux

- Assigné à changé de Edouard Dausque à Romain H.

#16 - 02/09/2020 21:53 - Quentin Gibeaux

- Version cible changé de Été 2020 à Septembre 2020

#17 - 30/09/2020 21:45 - Quentin Gibeaux

- Version cible changé de Septembre 2020 à Octobre 2020

#18 - 20/10/2020 16:49 - François Poulain

Visiblement ça semble être surtout au niveau du qcow la perte de perfs.

D'après la doc de proxmox la config qu'ils mettent par défaut est cache=none.
https://pve.proxmox.com/wiki/Performance_Tweaks

Chez nous le paramètre n'est pas défini donc visiblement c'est writethrough ou writeback selon la version de Qemu. D'après https://www.qemu.org/docs/master/system/invocation.html ça serait writeback (chercher cache=cache dans la page).

Idée : faire un bench sur une VM secondaire après reboot ; changer le cache pour none ; rebot ; refaire le bench.

#19 - 20/10/2020 17:16 - Christian P. Momon

Extrait de **man qemu-system** :

```
           cache=cache
               cache is "none", "writeback", "unsafe", "directsync" or "writethrough" and controls how the hos
t cache is used to access block data. This is a shortcut that sets the cache.direct and cache.no-flush options
 (as in -blockdev), and additionally
               cache.writeback, which provides a default for the write-cache option of block guest devices (as
 in -device). The modes correspond to the following settings:

                                    | cache.writeback   cache.direct   cache.no-flush
                                ----+-----------------------------------------------
                                writeback     | on                off            off
                                none          | on                on             off
                                writethrough  | off               off            off
                                directsync    | off               on             off
                                unsafe        | on                off            on

               The default mode is cache=writeback.
```

Du coup, la valeur par défaut est **writeback**.

#20 - 20/10/2020 18:05 - François Poulain

Le wiki Debian conseille également cache=none
https://wiki.debian.org/KVM

#21 - 20/10/2020 18:09 - François Poulain

Encore une note de confirmation :
https://bugs.launchpad.net/ubuntu/+source/virt-manager/+bug/568445
http://www.linux-kvm.org/page/Tuning_KVM

#22 - 20/10/2020 18:38 - François Poulain

Par ailleurs il est dit ici (https://wiki.qemu.org/Features/Snapshots#Live_Snapshot_Merge) que « Without the ability to merge and flatten snapshot images, the snapshot chain will continue to grow as new snapshots are made, which may become difficult to manage, in addition to introducing performance concerns. »

Or :

```
(April) root@virola:~# for vm in $(virsh list --name); do virsh snapshot-list $vm; done
 Name                 Creation Time               State
------------------------------------------------------------
 avant_migration_buster   2019-11-02 12:02:23 +0000   running

 Name                       Creation Time               State
------------------------------------------------------------------
 2019-11-02_before-buster   2019-11-02 10:33:44 +0000   running
 bastion-pre-migration      2017-11-25 18:10:00 +0000   running

 Name   Creation Time   State
------------------------------
```

```
 Name   Creation Time   State
------------------------------

 Name        Creation Time               State
-----------------------------------------------
 snapshot1   2017-04-04 07:55:28 +0000   running

 Name                     Creation Time               State
---------------------------------------------------------------
 avant_migration_buster   2019-11-02 13:57:11 +0000   shutoff
 pre-upgrade-stretch      2018-03-20 15:32:37 +0000   running

 Name                             Creation Time               State
-------------------------------------------------------------------------
 migration_avant_buster           2019-11-02 07:58:24 +0000   running
 Snapshort avant upgrade strech   2017-10-10 18:11:20 +0000   shutoff

 Name                     Creation Time               State
---------------------------------------------------------------
 migration_avant_buster   2019-11-01 22:10:23 +0000   shutoff

 Name   Creation Time   State
------------------------------
```

#23 - 20/10/2020 18:39 - François Poulain

Et :

```
(April) root@calamus:~# for vm in $(virsh list --name); do virsh snapshot-list $vm; done
 Name                     Creation Time               State
---------------------------------------------------------------
 avant_migration_buster   2019-11-02 07:55:04 +0000   running

 Name   Creation Time   State
------------------------------

 Name                     Creation Time               State
---------------------------------------------------------------
 avant_migration_buster   2019-11-02 07:54:45 +0000   shutoff

 Name                     Creation Time               State
---------------------------------------------------------------
 avant_migration_buster   2019-11-02 10:00:57 +0000   shutoff
 snapshot-prestretch      2017-11-25 11:10:48 +0000   shutoff

 Name                      Creation Time               State
-----------------------------------------------------------------
 20191008-pre-maj-buster   2019-10-08 17:16:12 +0000   shutoff
 avant_migration_buster    2019-11-02 12:17:12 +0000   running
 lamp-pre-maj-stretch      2017-11-25 10:55:38 +0000   shutoff

 Name                      Creation Time               State
-----------------------------------------------------------------
 avant_migration_buster    2019-11-01 21:30:33 +0000   running
 avant_migration_stretch   2017-11-25 11:35:01 +0000   shutoff

 Name                                 Creation Time               State
-------------------------------------------------------------------------------
 avant_migration_buster               2019-11-01 21:31:41 +0000   running
 Snapshot avant installation testing  2017-04-19 08:33:04 +0000   running
 snapshot_avant_migration_stretch     2017-11-26 08:45:17 +0000   running

 Name   Creation Time   State
------------------------------

 Name   Creation Time   State
------------------------------

 Name   Creation Time   State
------------------------------

 Name                      Creation Time               State
-----------------------------------------------------------------
```

```
2019-11-02_before-buster   2019-11-02 08:54:22 +0000   shutoff
```

#24 - 20/10/2020 18:52 - Christian P. Momon

Bien vu pour les snapshots. Oui, les vieux snapshots ne présentent aucun intérêt. À supprimer.

Est-ce qu'en avoir un neuf par vm pourrait être utile ?

#25 - 20/10/2020 18:56 - Christian P. Momon

La différence entre **writeback** et **none**, c'est **cache.direct** qui passe de *off* à *on* :

```
                              | cache.writeback   cache.direct   cache.no-flush
                    ----------+-----------------------------------------------
                    writeback | on                off            off
                    none      | on                on             off
```

Toujours le man de **qemu-system** :

```
               "cache.direct"
                   The host page cache can be avoided with cache.direct=on. This will attempt to do disk IO di
rectly to the guest's memory. QEMU may still perform an internal copy of the data.
```

De ce que je comprends, ça fait une étape en moins entre le cache de la vm et le cache disque de la pm. Du coup, super tentable, nan ?

#26 - 20/10/2020 19:01 - François Poulain

> Bien vu pour les snapshots. Oui, les vieux snapshots ne présentent aucun intérêt. À supprimer.

Oui je fais le ménage de suite.

> Est-ce qu'en avoir un neuf par vm pourrait être utile ?

Il n'y en avait pas sur le cluster quand il était neuf. :)

#27 - 20/10/2020 19:10 - François Poulain

> Du coup, super tentable, nan ?

Oui ; perso je serais pour un sed dans les xml.

#28 - 20/10/2020 19:44 - François Poulain

Je pense qu'on voit les premier résultats concrets : les graphes se chargent du premier coup dans icinga2.

#29 - 28/10/2020 21:45 - François Poulain

nota bene: hurdman : https://github.com/fghaas/drbd-documentation/blob/master/users-guide/benchmark.txt

#30 - 28/10/2020 21:46 - Quentin Gibeaux

- Version cible changé de Octobre 2020 à Novembre 2020

#31 - 28/10/2020 22:03 - Romain H.

- Version cible changé de Novembre 2020 à Octobre 2020

À faire :

- Tester impact sur les performances de cache=none et mettre en place
- Voir si le lien RPN est saturé
- Tester les perfs de DRBR, voir #29

#32 - 28/10/2020 22:03 - Romain H.

- Version cible changé de Octobre 2020 à Novembre 2020

#33 - 25/11/2020 21:28 - Quentin Gibeaux

- Version cible changé de Novembre 2020 à Décembre 2020

#34 - 28/11/2020 12:42 - Romain H.

La configuration de cache=none a l'air d'avoir un impact négatif sur les performances.
Le cache rajoutait une étape supplémentaire mais celle-ci semble avoir un impact positif dans l'état actuel.
Je pense qu'il faut plus chercher du côté du RPN et du DRDB.

Avant cache=none :

```
# dd if=/dev/zero of=/tmp/test1.img bs=256M count=4 oflag=dsync
4+0 records in
4+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 22.224 s, 48.3 MB/s

# dd if=/dev/zero of=/tmp/test1.img bs=256M count=4 oflag=dsync
4+0 records in
4+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 22.6473 s, 47.4 MB/s

# time /etc/rrsync.d/dump-mysql

real    4m11.024s
user    3m34.601s
sys    0m4.207s

# time /etc/rrsync.d/dump-mysql

real    4m4.426s
user    3m33.345s
sys    0m4.042s
```

Après cache=none

```
# dd if=/dev/zero of=/tmp/test1.img bs=256M count=4 oflag=dsync
4+0 records in
4+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 142.853 s, 7.5 MB/s

# dd if=/dev/zero of=/tmp/test1.img bs=256M count=4 oflag=dsync
4+0 records in
4+0 records out
1073741824 bytes (1.1 GB, 1.0 GiB) copied, 162.102 s, 6.6 MB/s

# time /etc/rrsync.d/dump-mysql

real    5m53.383s
user    3m32.077s
sys    0m3.187s

# time /etc/rrsync.d/dump-mysql

real    5m35.521s
user    3m31.667s
sys    0m4.035s
```

#35 - 28/11/2020 15:22 - Romain H.

- Fichier io_wait_vs_brandwith_annote.png ajouté

J'ai essayé de mettre en relation les moments de saturation du lien RPN et les pics d'IO WAIT.
Il y a l'air d'y avoir une corrélation partielle, aux plus gros pics de saturation réseau correspondent ceux des IO WAIT.
Il y a cependant quelques pics d'IO WAIT auxquels ne correspond aucune saturation réseau (PW0 et PW1), je ne pense donc pas que ça soit suffisant pour prouver un lien de causalité.

io_wait_vs_brandwith_annote.png

#36 - 30/12/2020 22:18 - Quentin Gibeaux

- Version cible changé de Décembre 2020 à Janvier 2021

#37 - 21/01/2021 20:27 - François Poulain

J'ai installé atop sur les deux hyperviseurs pour obtenir de la data concernant la gourmandise relative des VM entre elles.

#38 - 27/01/2021 21:59 - Quentin Gibeaux

- Version cible changé de Janvier 2021 à Février 2021

#39 - 28/01/2021 21:12 - François Poulain

J'ai ajouté sur les machines baremetal une sonde qui parse la sortie de atopsar -D pour identifier les 3 processes qui bouffent le plus d'IO. Lorsque un des process en question est qemu-img, je parse la ligne de commande pour mettre en avant le nom de la VM. Les sorties sur les perfs data permettront de suivre un historique.

#40 - 24/02/2021 21:49 - Quentin Gibeaux

- Version cible changé de Février 2021 à Mars 2021

#41 - 31/03/2021 22:09 - Quentin Gibeaux

- Version cible changé de Mars 2021 à Avril 2021

#42 - 31/03/2021 22:19 - François Poulain

- Fichier Capture d'écran de 2021-03-31 22-10-23.png ajouté

- Fichier Capture d'écran de 2021-03-31 22-16-03.png ajouté

Deux captures des graphes qui mettent en évidence les pompeurs d'IO sur virola et calamus.

En vert et bleu, adl et lamp sur calamus.

En vert et orange, admins et bastion sur virola.

#43 - 28/04/2021 21:43 - Quentin Gibeaux

- Version cible changé de Avril 2021 à Mai 2021

#44 - 26/05/2021 21:57 - Quentin Gibeaux

- Version cible changé de Mai 2021 à Juin 2021

#45 - 30/06/2021 21:46 - Quentin Gibeaux

- Version cible changé de Juin 2021 à Été 2021

#46 - 01/09/2021 22:20 - Quentin Gibeaux

- Version cible changé de Été 2021 à Septembre 2021

#47 - 29/09/2021 22:06 - Quentin Gibeaux

- Version cible changé de Septembre 2021 à Octobre 2021

#48 - 27/10/2021 21:55 - Quentin Gibeaux

- Version cible changé de Octobre 2021 à Novembre 2021

#49 - 22/11/2021 09:59 - François Poulain

Suite à la migration hetzner je refais les tests dd sur lamp data : pas d'amélioration. Voire, la latence est moins bonne.

```
(April) root@lamp:~# for i in $(seq 10); do dd if=/dev/zero of=/var/www/testfile bs=100M count=1 oflag=dsync;
done
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,45701 s, 42,7 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,50311 s, 41,9 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 1,94212 s, 54,0 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 1,76595 s, 59,4 MB/s
1+0 records in
```

```
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,01781 s, 52,0 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,25676 s, 46,5 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,97005 s, 35,3 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 1,63784 s, 64,0 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,01444 s, 52,1 MB/s
1+0 records in
1+0 records out
104857600 bytes (105 MB, 100 MiB) copied, 2,3155 s, 45,3 MB/s

(April) root@lamp:~# dd if=/dev/zero of=/var/www/testfile bs=512 count=1000 oflag=dsync
1000+0 records in
1000+0 records out
512000 bytes (512 kB, 500 KiB) copied, 342,789 s, 1,5 kB/s
```

#50 - 24/11/2021 21:29 - Quentin Gibeaux

- Version cible changé de Novembre 2021 à Décembre 2021

#51 - 29/12/2021 21:06 - Quentin Gibeaux

- Version cible changé de Décembre 2021 à Janvier 2022

#52 - 26/01/2022 21:16 - Quentin Gibeaux

- Version cible changé de Janvier 2022 à Février 2022

#53 - 23/02/2022 21:17 - Quentin Gibeaux

- Version cible changé de Février 2022 à Mars 2022

#54 - 30/03/2022 21:28 - Quentin Gibeaux

- Version cible changé de Mars 2022 à Avril 2022

#55 - 27/04/2022 21:31 - Quentin Gibeaux

- Version cible changé de Avril 2022 à Mai 2022

#56 - 01/06/2022 21:20 - Quentin Gibeaux

- Version cible changé de Mai 2022 à Juin 2022

#57 - 29/06/2022 21:25 - Quentin Gibeaux

- Version cible changé de Juin 2022 à Été 2022

#58 - 31/08/2022 21:44 - Quentin Gibeaux

- Version cible changé de Été 2022 à Septembre 2022

#59 - 28/09/2022 21:17 - Quentin Gibeaux

- Version cible changé de Septembre 2022 à Octobre 2022

#60 - 26/10/2022 21:23 - Quentin Gibeaux

- Version cible changé de Octobre 2022 à Novembre 2022

#61 - 30/11/2022 21:17 - Quentin Gibeaux

- Version cible changé de Novembre 2022 à Décembre 2022

#62 - 04/01/2023 21:13 - Quentin Gibeaux

- Version cible changé de Décembre 2022 à Janvier 2023

#63 - 25/01/2023 21:19 - Quentin Gibeaux

- Version cible changé de Janvier 2023 à Février 2023

#64 - 01/03/2023 21:40 - Frédéric Couchet

- Version cible changé de Février 2023 à Mars 2023

#65 - 29/03/2023 21:12 - Quentin Gibeaux

- Version cible changé de Mars 2023 à Avril 2023

#66 - 26/04/2023 21:18 - Quentin Gibeaux

- Version cible changé de Avril 2023 à Mai 2023

#67 - 31/05/2023 21:15 - Quentin Gibeaux

- Version cible changé de Mai 2023 à Juin 2023

#68 - 30/06/2023 16:05 - Quentin Gibeaux

- Version cible changé de Juin 2023 à Été 2023

#69 - 30/08/2023 21:19 - Quentin Gibeaux

- Version cible changé de Été 2023 à Septembre 2023

#70 - 27/09/2023 21:22 - Quentin Gibeaux

- Version cible changé de Septembre 2023 à Octobre 2023

#71 - 25/10/2023 21:18 - Quentin Gibeaux

- Version cible changé de Octobre 2023 à Novembre 2023

#72 - 29/11/2023 21:12 - Quentin Gibeaux

- Version cible changé de Novembre 2023 à Décembre 2023

#73 - 27/12/2023 21:19 - Quentin Gibeaux

- Version cible changé de Décembre 2023 à Janvier 2024

#74 - 31/01/2024 21:15 - Quentin Gibeaux

- Version cible changé de Janvier 2024 à Février 2024

#75 - 28/02/2024 21:21 - Quentin Gibeaux

- Version cible changé de Février 2024 à Mars 2024

#76 - 27/03/2024 21:18 - Quentin Gibeaux

- Version cible changé de Mars 2024 à Avril 2024

Fichiers

io_wait_vs_brandwith_annote.png	2,19 Mo	28/11/2020	Romain H.
Capture d’écran de 2021-03-31 22-10-23.png	356 ko	31/03/2021	François Poulain
Capture d’écran de 2021-03-31 22-16-03.png	307 ko	31/03/2021	François Poulain