

Présentation.

Dans le cadre du partenariat d'infogérance avec Dédibox j'ai eu l'opportunité de participer au programme de test de la nouvelle gamme de serveur lancée par Dédibox.

Dedibox V2.

- Processeur **Celeron 220 1.20 Ghz**
- Mémoire vive de **1 Go.**
- Disque dur de **160 Go.**
- Prix : **29,99 Euros HT / mois, soit 35,90 Euros TTC.**

Dedibox XL.

- Processeur **Core2Duo T7200 2.00GHz**
- Mémoire vive de **3 Go.**
- Disque dur de **500 Go à enregistrement perpendiculaire.**
- Prix : **49,99 Euros HT / mois, soit 59,80 Euros TTC.**

Afin de comparer objectivement ces nouveaux serveurs aux offres concurrentes nous avons aussi passé les mêmes séries de tests sur un Gandi de 8 parts, et un serveur OVH Superplan 2008.

Gandi 8 parts.

- Processeur **Dual-Core AMD Opteron 8218.**
- Mémoire vive de **2 Go.**
- Disque dur de **40 Go.**
- Prix : **48 Euros HT / mois, soit 57,40 euros TTC.**

Superplan 2008.

- Processeur **Intel Core2Duo E6550.**
- Mémoire vive de **1 Go.**
- Disque dur : **2 fois 250 Go.**
- Prix : **69 euros HT / mois, soit 82,53 euros TTC.**

Table des matières

Présentation.....	1
Dedibox V2.....	1
Dedibox XL.....	1
Gandi 8 parts.....	1
Superplan 2008.....	1
Encodage de vidéos en flv par ffmpeg.....	3
Ligne de commande utilisée :.....	3
Chiffrement de fichier par openssl.....	5
Attention.....	5
Ligne de commande utilisée.....	5
Traitements de statistiques à partir des logs Apache.....	7
Description.....	7
Exemple.....	7
Plusieurs traitements de statistiques en parallèle.....	12
Description.....	12
Résultats.....	12
Conclusion.....	14
Plusieurs encodages ffmpeg en parallèle.....	15
Description.....	15
Résultats.....	15
Conclusion.....	17
Plusieurs traitements statistiques et plusieurs encodages.....	18
Description.....	18
Résultats.....	18
Conclusion.....	19

Encodage de vidéos en flv par ffmpeg.

```
/usr/local/ffmpeg/bin/ffmpeg -version
```

FFmpeg version SVN-r9211, Copyright (c) 2000-2007 Fabrice Bellard, et al.

```
configuration: --enable-pthreads --enable-liba52 --enable-liba52bin --enable-  
libfaad --enable-libfaac --enable-libmp3lame --enable-libogg --enable-libtheora  
--enable-libvorbis --enable-x264 --enable-xvid --enable-gpl --disable-ipv6 --  
disable-ffserver --disable-ffplay --enable-memalign-hack --  
prefix=/usr/local/ffmpeg
```

```
libavutil version: 49.4.0
```

```
libavcodec version: 51.40.4
```

```
libavformat version: 51.12.1
```

```
built on Feb 24 2008 13:03:31, gcc: 4.1.2 20061115 (prerelease) (Debian  
4.1.1-21)
```

```
ffmpeg SVN-r9211
```

```
libavutil 3212288
```

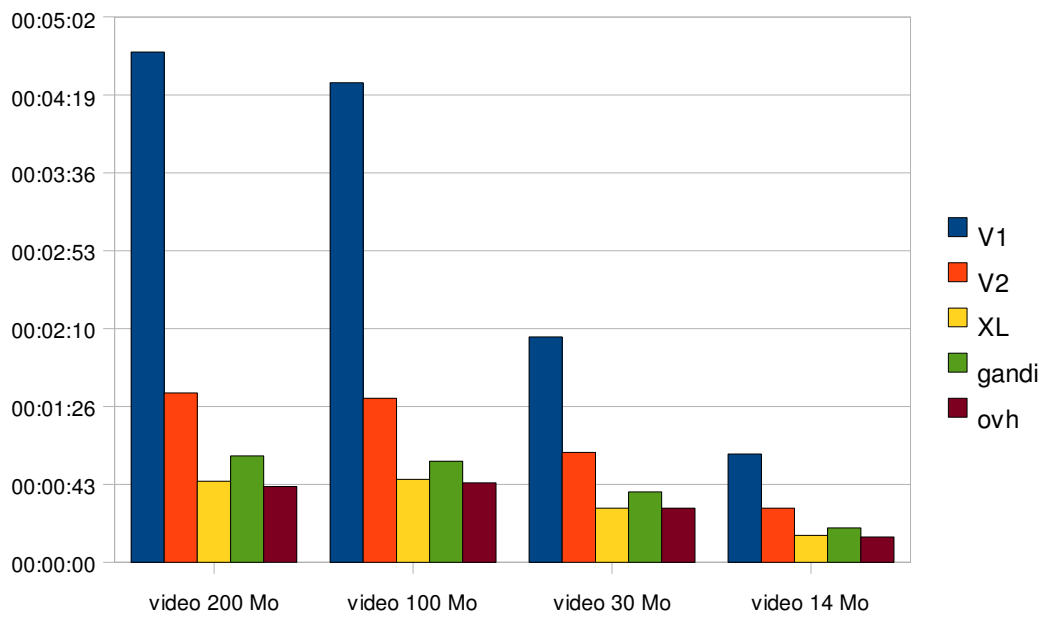
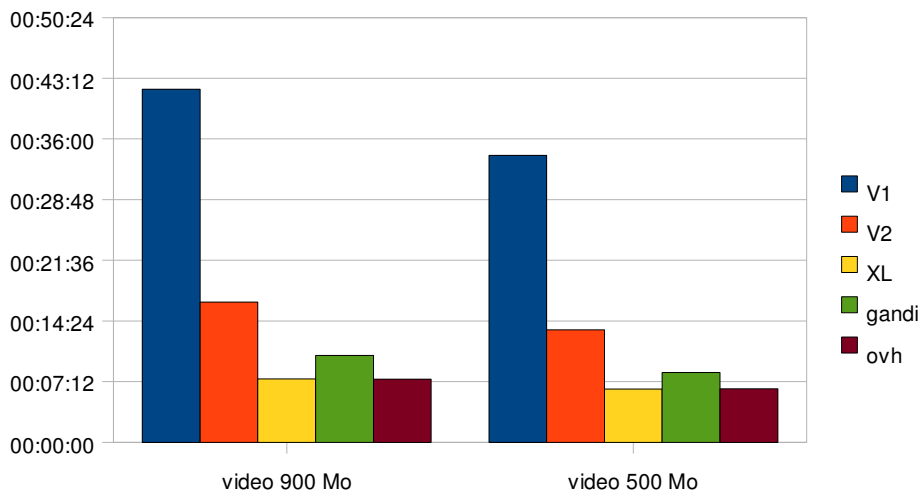
```
libavcodec 3352580
```

```
libavformat 3345409
```

Ligne de commande utilisée :

```
/usr/local/ffmpeg/bin/ffmpeg -y -i huntitled.divx -bufsize 1000k -f flv -vcodec  
flv -r 25.0 -maxrate 1200k -qmin 4 -qmax 5 -s 320x240 -acodec mp3 -ab 128k  
-ac 2 -ar 44100 -vol 256 1.flv
```

Taille vidéo (mo)	V1	V2	XL	gandi-8	OVH-S
14	00:01:00	00:00:30	00:00:15	00:00:19	00:00:14
30	00:02:05	00:01:01	00:00:30	00:00:39	00:00:30
100	00:04:26	00:01:31	00:00:46	00:00:56	00:00:44
200	00:04:46	00:01:34	00:00:45	00:00:59	00:00:42
500	00:34:03	00:13:21	00:06:19	00:08:18	00:06:21
900	00:41:54	00:16:39	00:07:32	00:10:19	00:07:30



Chiffrement de fichier par openssl.

Attention

Le but ici est de mesurer le temps de chiffrement de gros fichiers en utilisant openssl, il faut donc tenir compte du fait qu'il y a l'overhead disque ce n'est pas du chiffrement pur.

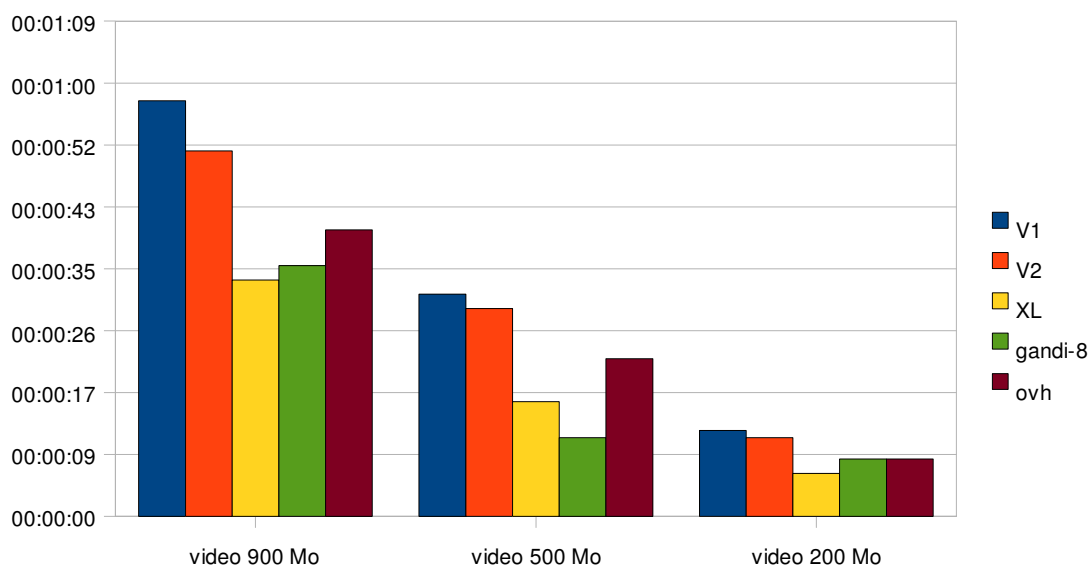
Les fichiers de tests sont les vidéos utilisées pour le test d'encodage ffmpeg.

Ligne de commande utilisée.

```
openssl enc -aes-256-cbc -salt -in huntitled.divx -out file1.in.enc -pass  
file:/home/ptrognon/encodage\_direct/passphrase
```

Sur C7 l'option -engine padlock est utilisée.

Taille vidéo (mo)	V1 (padlock)	V2	XL	gandi-8	OVH-S
14	00:00:00	00:00:00	00:00:00	00:00:00	00:00:00
30	00:00:01	00:00:00	00:00:00	00:00:01	00:00:00
100	00:00:06	00:00:05	00:00:01	00:00:01	00:00:04
200	00:00:12	00:00:11	00:00:06	00:00:08	00:00:08
500	00:00:31	00:00:29	00:00:16	00:00:11	00:00:22
900	00:00:58	00:00:51	00:00:33	00:00:35	00:00:40



Traitements de statistiques à partir des logs Apache.

Description.

Parsing et agrégation des logs d'apache dans une base mysql de type MyISAM.
Les informations stockées étant du type :

- Nombre d'accès à un fichier par mois.
- Nombre d'accès à un fichier par mois et par serveur.
- Nombre d'accès à un fichier par jour
- Nombre d'accès à un fichier par jour et par serveur.

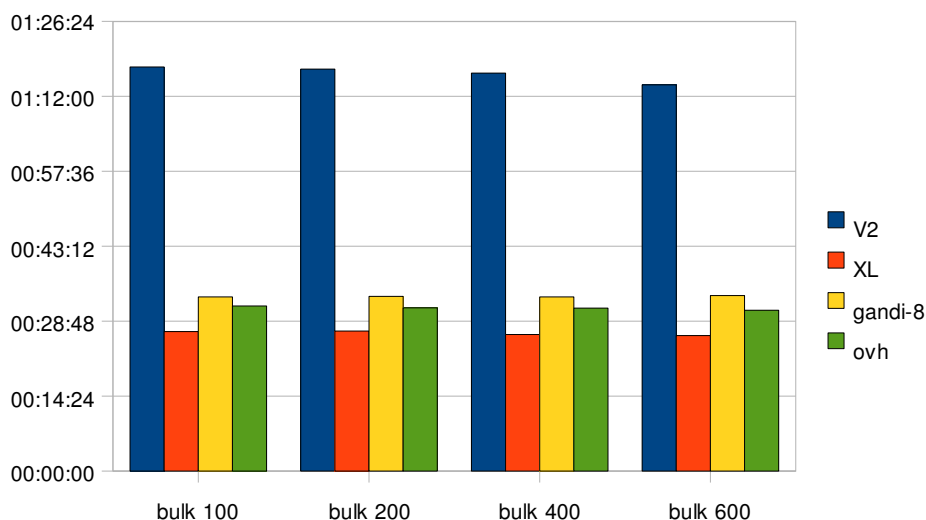
Les données sont parsées par un daemon perl qui traite les logs apache au fur et à mesure de l'append fait par apache dans son fichier de log, les queries de mise à jour des compteurs dans les tables sont du type bulk insert + on duplicate primary key.

Exemple

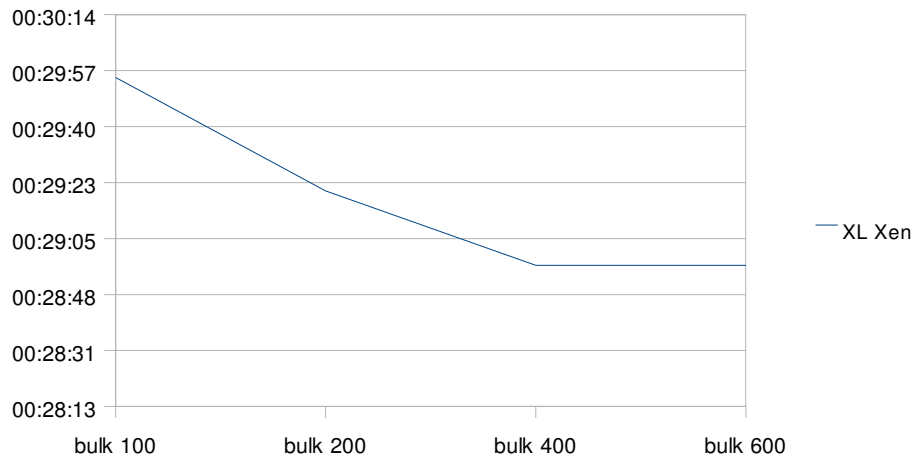
```
INSERT INTO DFA(f_id,counter) values(1,3),(2,1),(3,9) ON DUPLICATE KEY UPDATE counter = counter + values(counter) ;
```

4 tailles de bulk ont été utilisées, plus la taille du bulk est importante plus MyISAM est sollicité.

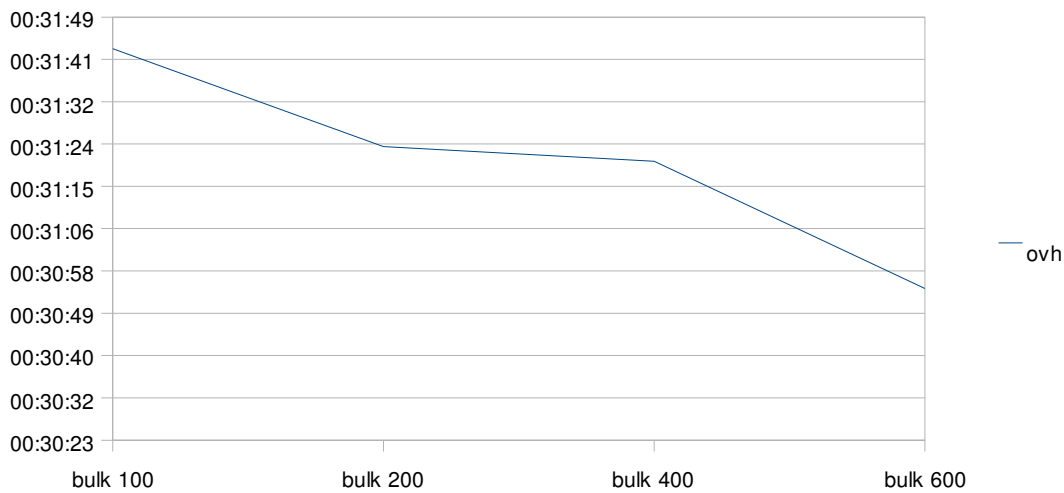
bulk size	V2	XL	gandi-8	OVH-S
100	01:17:40	00:26:47	00:33:28	00:31:43
200	01:17:14	00:26:55	00:33:35	00:31:23
400	01:16:30	00:26:13	00:33:27	00:31:20
600	01:14:13	00:26:04	00:33:44	00:30:54



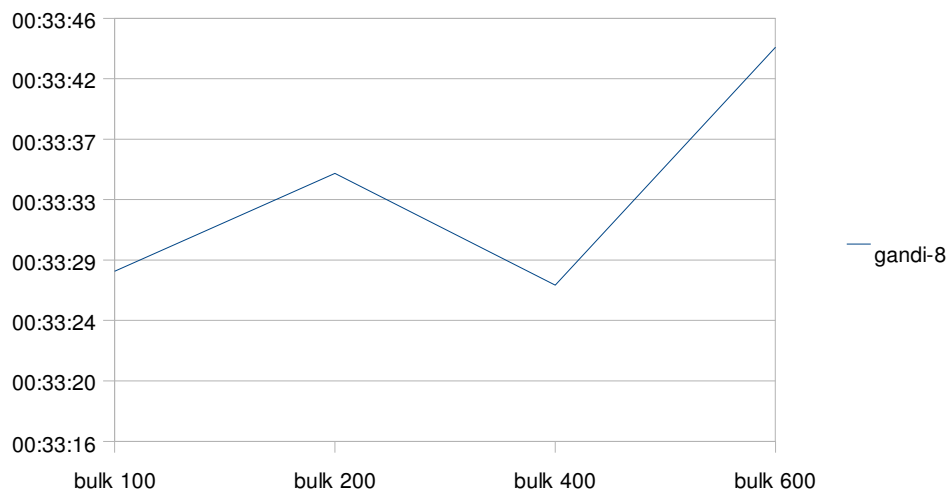
Sur la dédibox XL. Ce graphique illustre le temps de traitement avec augmentation de la taille du buffer des traitements bulk



Sur la Superplan OVH. ce graphique illustre le temps de traitement avec augmentation de la taille du buffer des traitements bulk



Sur la gandi 8 parts. Ce graphique illustre le temps de traitement avec augmentation de la taille du buffer des traitements bulk



Alors que normalement plus la taille du bulk augmente plus le temps de traitement diminue (jusqu'à une certaine limite bien sur) dans le cas du serveur Gandi ce n'est pas ce que l'on constate, la VM Gandi serait elle perturbée par les autres hébergées sur le même hôte, ou est-ce un effet de Xen ?

Plusieurs traitements de statistiques en parallèle.

Description.

Le but est de lancer 2 fois le même traitement de statistiques sur 2 fichiers de logs différents, dans les deux cas l'échantillon contient 150 000 lignes de logs.

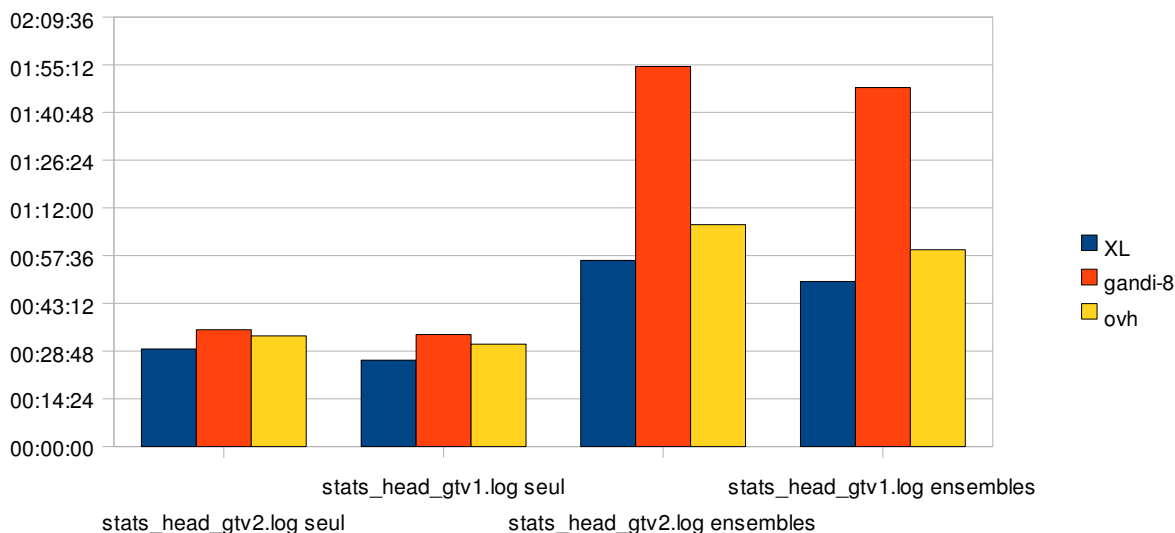
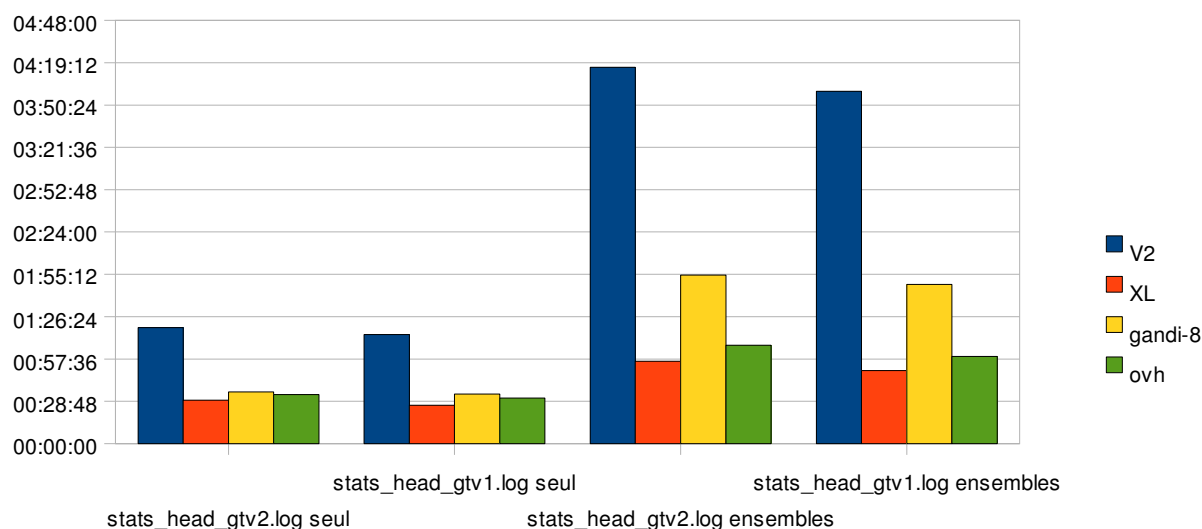
Le fichier contenant les logs d'un premier serveur apache se nomme stats_1.log.
Le fichier contenant les logs d'un second serveur apache se nomme stats_2.log.

La taille du bulk a été fixée à 600, ce qui correspond au paramètre utilisé en production.

Résultats.

	V2	XL	gandi-8	OVH-S
(1) stats_1 seul	01:14:13	00:26:28	00:33:44	00:30:54
(2) stats_2 seul	01:18:57	00:29:28	00:35:12	00:33:24
(3) stats_1 et stats_2	03:59:48	00:49:46	01:48:16	00:59:22
(4) stats_2 et stats_1	04:15:59	00:56:07	01:54:39	01:06:58

- (1) : temps de traitement du fichier stats_1.log seul, **rien d'autre ne tourne sur le serveur mis à part ce traitement.**
- (2) : temps de traitement du fichier stats_2.log seul, **rien d'autre ne tourne sur le serveur mis à part ce traitement.**
- (3) : temps de traitement du fichier stats_1.log **alors que le fichier stats_2.log est aussi en cours de traitement.**
- (4) temps de traitement du fichier stats_2.log **alors que le fichier stats_1.log est aussi en cours de traitement.**



Conclusion.

On voit très nettement ici que le point de contention est sur la base mysql puisque même sur la XL disposant de 2 cS sur on double quasiment le temps de traitement quand les deux sont lancés en parallèle, à remarquer que ce temps est plus que double sur une V2 ayant un seul cS ur.

On remarque aussi le temps nécessaire au serveur Gandi quand on veut passer les deux traitements en même temps.

On peut aussi remarquer une légère différence de puissance processeur entre l'offre OVH et la XL, mais la dedibox XL est largement avantagée par son disque dur dernière génération, dont la différence avec le Superplan 2008 est importante.

Plusieurs encodages ffmpeg en parallèle.

Description.

Pour ce test 3 vidéos avi ont été utilisées.

video1.avi 900 Mo
video2.avi 800 Mo
video3.avi 700 Mo

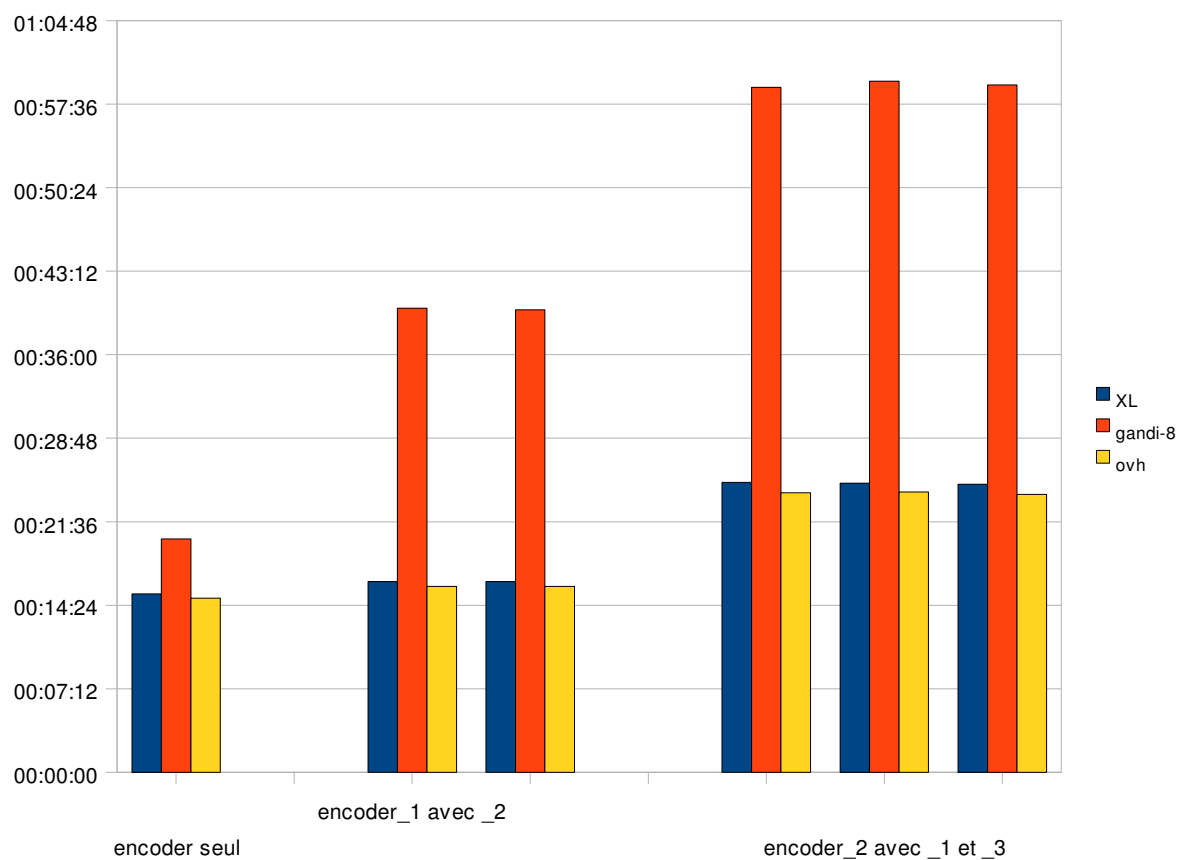
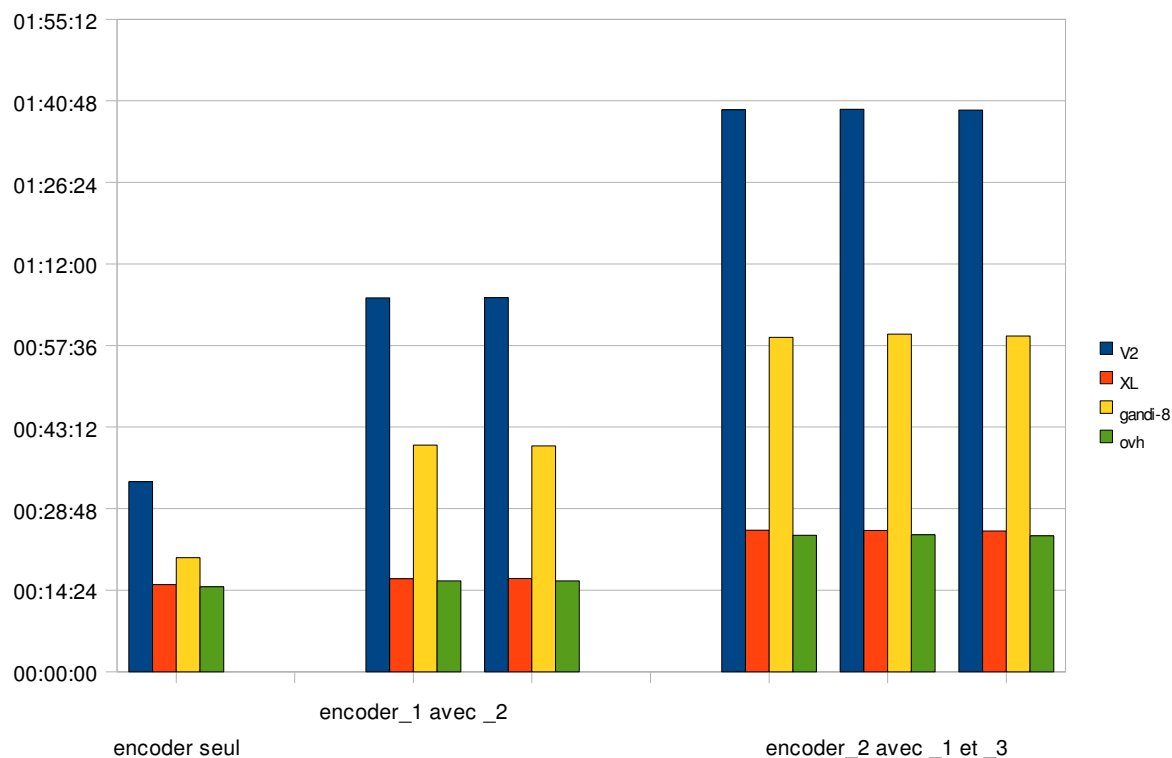
script1.sh enchaîne l'encodage de video1, video2, video3.
script2.sh enchaîne l'encodage de video3, video1, video2.
script3.sh enchaîne l'encodage de video2, video3, video1.

Le temps de traitement de chaque script est mesuré.

Résultats.

	V2	XL	gandi-8	OVH-S
(1) script1	00:33:34	00:15:33	00:20:07	00:15:01
(2) script2	00:33:34	00:15:33	00:20:07	00:15:01
(3) script3	00:33:34	00:15:33	00:20:07	00:15:01
(4) script1 script2	01:06:00	00:16:16	00:40:01	00:16:02
(5) script2 script1	01:06:04	00:16:27	00:39:53	00:16:01
(6) script1 script2 script3	01:39:12	00:25:00	00:59:03	00:24:06
(7) script2 script1 script3	01:39:18	00:24:55	00:59:35	00:24:10
(8) script3 script1 script2	01:39:10	00:24:50	00:59:16	00:23:58

- (1) : seul script1 est lancé.
- (2) : seul script2 est lancé.
- (3) : seul script3 est lancé.
- (4) : script1 est lancé avec script2, temps de traitement de script1.
- (5) : script1 est lancé avec script2, temps de traitement de script2.
- (6) : les trois scripts sont lancés, temps de traitement de script1.
- (7) : les trois scripts sont lancés, temps de traitement de script2.
- (8) : les trois scripts sont lancés, temps de traitement de script3.



Conclusion.

On voit ici l'intérêt du multi- cS ur qui permet de garder quasiment le même temps de traitement quand on en a 2 lourds, et qui ne demande pas beaucoup plus si on passe à 3 traitements lourds.

Une fois de plus le Gandi n'ayant qu'un seul cS ur est très pénalisé par les tests de charge.

Plusieurs traitements statistiques et plusieurs encodages.

Description.

Pour ce test les deux traitements des statistiques sont lancés avec les 3 encodeurs vidéos.

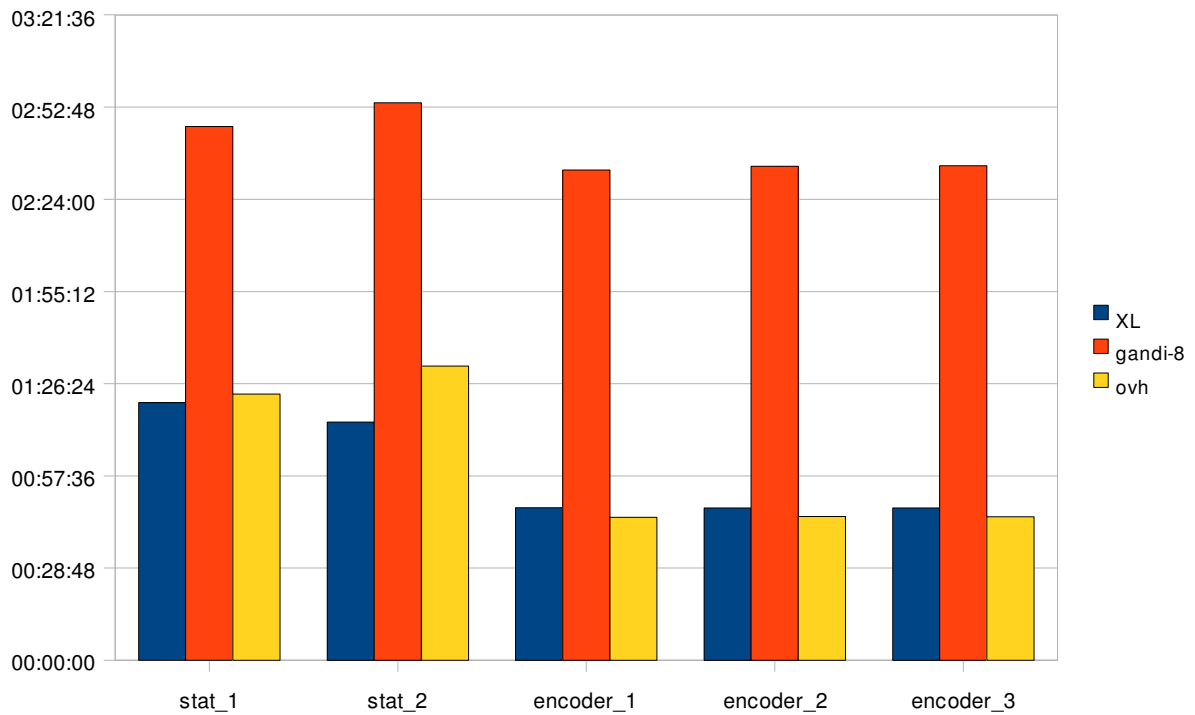
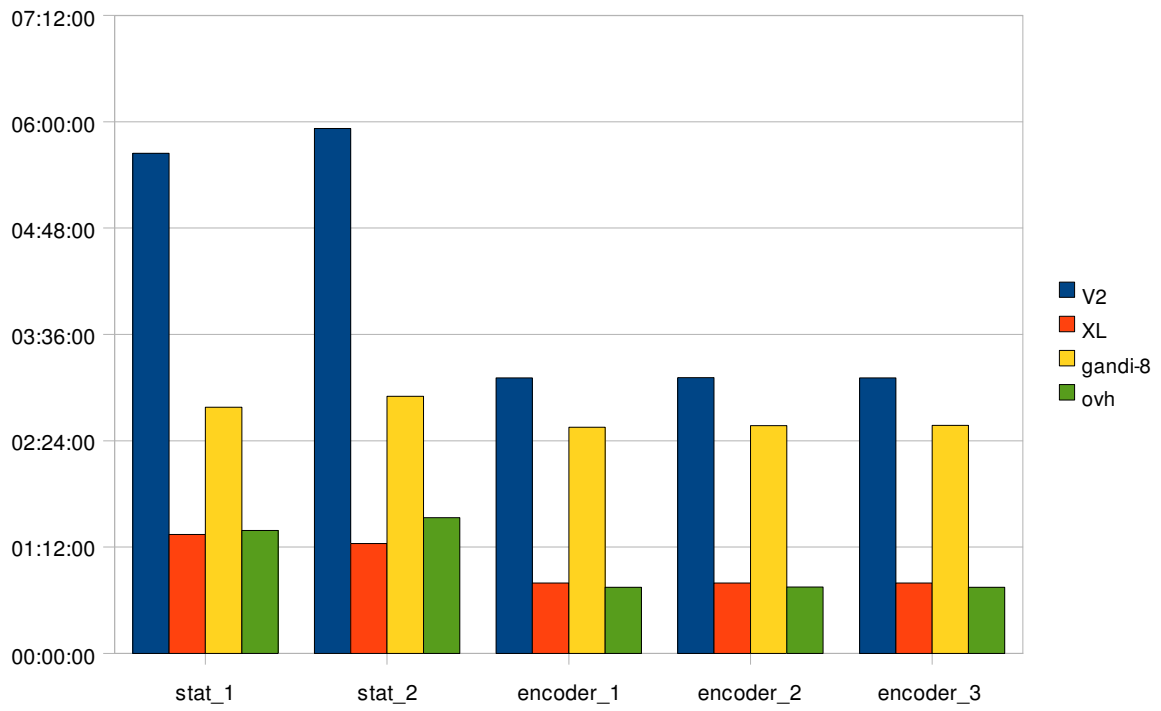
Les traitements de statistiques sont les mêmes que ceux du chapitre précédent. Les trois encodeurs vidéos sont les mêmes que ceux utilisés dans le test précédent.

Dans ce test les 5 traitements sont lancés en même temps.

Résultats.

	V2	XL	gandi-8	OVH-S
(1) stats 1	05:38:44	01:20:29	02:46:43	01:23:11
(2) stats 2	05:55:27	01:14:23	02:54:10	01:31:54
(3) encoder 1	03:06:36	00:47:41	02:33:10	00:44:42
(4) encoder 2	03:06:48	00:47:35	02:34:19	00:44:57
(5) encoder 3	03:06:33	00:47:36	02:34:27	00:44:49

1. : temps de traitement de premier fichier de logs.
2. : temps de traitement du second fichier de logs.
3. : temps du traitement du premier encodeur ffmpeg.
4. : temps de traitement du second encodeur ffmpeg.
5. : temps de traitement du troisième encodeur ffmpeg.



Conclusion.

Ce dernier graphique est très intéressant car il met encore plus en évidence l'importance des disques à enregistrement perpendiculaire.

On voit en effet sur les traitements sollicitant principalement le CPU (encodage ffmpeg) le serveur OVH légèrement avantagé, par contre sur les traitements sollicitant l'ensemble de la puissance du serveur (statistiques) la XL sort largement en tête.

Tests réalisés en mars 2008 par Patrice Trognon,
infogérant de serveur dédiés et partenaire officiel
dedibox.