


Choix d'un système de fichiers pour Firebird sous linux

par Philippe Makowski ([IBPhoenix](#)) ([Blog](#)) Traduction par Evaris NGOUZO ([Page perso](#))

Date de publication : 06 mars 2010

Dernière mise à jour :


Cet article est la traduction de  [Choosing a File System on Linux for Firebird](#)
Commentez cet article :

I - Introduction.....	3
II - Méthode.....	3
III - Résultats.....	3
III-1 - Avec les paramètres par défaut.....	3
III-2 - Avec l'option Sync activée.....	3
III-3 - Deadline de l'ordonnanceur E/S.....	4
IV - Conclusion.....	4
V - Remerciements.....	5

I - Introduction

Si je pouvais utiliser un seul mot pour parler de Linux, ce serait "choix". Après avoir choisi une distribution, vous devez choisir le système de fichiers pour stocker vos données. Aujourd'hui, nous avons principalement le choix entre Ext2, Ext3, Ext4 et XFS. Voyons l'impact de ces systèmes de fichiers sur les performances lors de l'utilisation de bases de données Firebird.

II - Méthode

Pour tester ces systèmes de fichiers, j'ai utilisé les benchmarks TPC-B ( <http://sourceforge.net/projects/dbbench/>) et TPC-C (1) avec Firebird 2.5 RC1 SuperClassic. En résumé, le benchmark TPC-C simule un très grand système de gestion des stocks tandis que TPC-B est conçu pour être un test de stress sur le cœur du système de base de données. Les deux s'exécutent pour un temps limité et comptent le nombre de transactions par minute. Les deux effectuent des opérations de type "CRUD".

Tous les tests ont été exécutés sous une machine virtuelle VmWare dotée de 30Go d'espace disque SCSI virtuel pré-alloué. La machine virtuelle a été configurée sous Mandriva 2010.0 x86_64, et Firebird a été compilé sur ce système. Je n'ai modifié aucun paramètre de configuration de Firebird, gardant le cache de base de données par défaut.

Le but n'est pas de faire un benchmark sur Firebird lui-même, mais de voir l'influence des systèmes de fichiers et des différents réglages sur ce dernier. Pour avoir un référentiel, les benchmarks ont d'abord été exécutés sur une base de données située sur un RAW device, c'est à dire sans système de fichiers. Tous les résultats de cet article sont exprimés en base 100 par rapport à ce premier résultat. Nous mesurons donc bien les performances relatives des différents systèmes, pas leur performances absolue.

III - Résultats

III-1 - Avec les paramètres par défaut

Par défaut, la distribution Linux monte un répertoire sans l'option *sync*, et avec l'option *atime* activée. Ce qui signifie que toutes les opérations d'entrée/sortie sont exécutées de manière asynchrone et que le temps d'accès à un inode est actualisé pour chaque accès. Ce n'est certainement pas la meilleure configuration pour une base de données Firebird si vous voulez être sûrs que toutes les données sont écrites rapidement sur le disque, mais cela nous donne une première figure de comparaison

raw	Ext2	Ext3	Ext4	XFS	JFS
100	102.81	102.91	102.22	104.17	104.63

Comme vous pouvez le constater, avec ces options de montage, le système de fichiers *JFS* est 4.63% plus rapide que le système RAW, et tous les systèmes de fichiers *Ext* ont presque la même performance.

Si nous désirons vraiment des écritures sûres et être certains que toutes nos opérations E/S sont effectuées de manière synchrone, nous devons utiliser des options de montage différentes.

III-2 - Avec l'option Sync activée

Maintenant nous effectuons les mêmes tests, mais avec les options de montage *sync* et *noatime* activées. Ce qui signifie que toutes les opérations E/S sont effectuées de manière synchrone et que le temps d'accès aux inodes n'est pas mis à jour (pour un accès plus rapide).

Avec ces nouveaux paramètres, nous voyons de gros changements : XFS et Ext3 ont une piètre performance, et JSF est le vainqueur, plus rapide que dans notre premier test, avec des valeurs par défaut. Les différences ne sont toutefois pas vraiment importantes, puisqu'elles sont juste d'environ 7 points au maximum :

raw	Ext2	Ext3	Ext4	XFS	JFS
100	105.82	99.11	101.77	93.84	107.54

III-3 - Deadline de l'ordonnanceur E/S

Le noyau Linux, le cœur du système d'exploitation, est responsable du contrôle d'accès au disque en utilisant l'ordonnanceur E/S. Dans une parution du Red Hat Magazine (2) nous avons dit :

L'ordonnanceur Completely Fair Queuing (CFQ) est l'algorithme par défaut dans Red Hat Enterprise Linux 4. Comme son nom l'indique, CFQ maintient une queue E/S par processus dynamique et essaie de distribuer la bande passante E/S disponible de manière équitable entre toutes les requêtes E/S. CFQ est adapté aux moyens et larges systèmes multiprocesseurs ainsi qu'aux systèmes nécessitant une performance E/S répartie entre de multiples LUNs et contrôleurs E/S. La Deadline elevator utilise un algorithme deadline afin de minimiser la latence E/S pour une requête E/S donnée. L'ordonnanceur fournit un comportement proche du temps réel et utilise une politique round robin afin d'essayer d'être partiel entre plusieurs requêtes E/S et éviter "process starvation". En utilisant cinq queues E/S, cet ordonnanceur va de manière agressive réordonner les requêtes afin d'améliorer la performance des E/S.

Testons à présent le deadline elevator.

	raw (CFQ)	Ext2	Ext3	Ext4	XFS	JFS
TPC-B	100	106.44	107.52	108.84	106.69	109.39
TPC-C	100	102.37	101.39	105.52	89.44	86.13
Moyenne	100	104.41	104.46	107.18	98.07	102.65

Il n'y a pas de gagnant. Les deux tests TPC ne donnent pas le même podium mais, clairement, l'ordonnanceur a eu un impact sur les performances.

Les tests ont été effectués avec un noyau 2.6.31 x86_64, deux CPUs et 4Go de mémoire vive. Pour voir si l'ordonnanceur a toujours un impact sur les performances, j'ai utilisé une nouvelle machine virtuelle avec Centos 4, un noyau 2.6.9 i386, un CPU et 2Go ram. Une machine virtuelle bien différente comme vous les voyez. Effectuons de nouveau le test TPC-B sur cette nouvelle machine virtuelle et voyons les résultats. Pas de JFS ni de Ext4 cette fois, parce que Ext4 est stable uniquement depuis le noyau 2.6.28 et que JFS n'est pas supporté en standard sur Red Hat Enterprise Linux 4 ou Centos 4.

ordonnanceur	raw (CFQ)	Ext2	Ext3	XFS
cfq	100	105.16	49.02	111.51
deadline	100	109.44	100.96	103.7

Le choix de l'ordonnanceur peut avoir un gros impact sur les performances, comme nous le constatons, mais nous ne pouvons affirmer qu'un bon ordonnanceur soit toujours meilleur qu'un autre. Nous pouvons juste dire que l'ordonnanceur est un point qui doit être pris en considération lorsque vous désirez obtenir le maximum de performance de votre système.

IV - Conclusion



"Ne vous basez pas sur les benchmarks effectués par quelqu'un d'autre que vous même !"

Nous avons observé qu'hormis l'effet de l'ordonnanceur sur un vieux noyau 2.6 et le système Ext3, il n'y a pas de grosses différences de performance. La conclusion est la suivante : pour avoir les meilleures performances pour vos bases de données Firebird, vous vous devez d'effectuer vos propres tests en premier afin de découvrir ce qui est

le mieux pour votre matériel. Commencez par choisir le système de fichier que vous estimez stable/sûr et, ensuite, essayez de changer l'ordonnanceur afin de trouver le plus adéquat/performant. Gardez à l'esprit qu'il n'y a pas de règles générales ! Une combinaison de matériel et de noyau différents va faire apparaître un comportement différent en fonction de l'ordonnanceur.

V - Remerciements

Je tiens à remercier **eusebe19** pour sa relecture orthographique.

- 1 : TPC, Transaction Processing Performance Council, disponible à  <http://www.tpc.org>
- 2 : D. John Shakshober, Choosing an I/O Scheduler for Red Hat® Enterprise Linux® 4 and the 2.6 Kernel, (Juin 2005), disponible à  <http://www.redhat.com/magazine/008jun05/features/schedulers/>