

## Re: Compressing file system, NFS

**Source:** <http://unix.derkeiler.com/Newsgroups/comp.unix.bsd.freebsd.misc/2003-10/2051.html>

---

**From:** Joe Douppnik ([jrd\\_at\\_cc.usu.edu](mailto:jrd_at_cc.usu.edu))

**Date:** 10/30/03

Date: 30 Oct 03 15:05:47 MDT

In article <bnrr8i\$1dq\$1@inn.jlab.org>, notformail <notformail@jlab.org> writes:

> *Nick Hilliard wrote:*

>>> *Is there any compressing file system available for FreeBSD 5.1-RELEASE*

>>> *(or*

>>> *later; i386)?*

>>

>>

>> *At the moment, no.*

>>

>>> *The systems are running as file servers (NFS). Therefore the file system  
>>> needs to be NFS exportable. A userland NFS server or a modified kernel  
>>> based NFS server would do the trick, too. All I need is on-the-fly  
>>> decompression. Right now the files are already compressed with gzip. The  
>>> idea is to decompress the files on the servers so that the clients see  
>>> only*

>>> *the original files. The clients are busy with analyzing the data and must  
>>> not waste CPU time with decompression. The NICs are Gigabit cards. The  
>>> network transfer rate is not an issue here.*

>>

>>

>> *Really, you're talking about moving the compression / decompression  
>> processing to the NFS server. If there were a compressed filesystem  
>> available, this would almost certainly cause really serious performance  
>> problems, because while decompression uses a fair amount of cpu,  
>> compressing stuff really chews it up. So if there were any sort of i/o  
>> load on the system, cpu usage would rocket, and performance would fall  
>> through the floor.*

>

> *Decompressing with GZIP gives me a transfer rate of 25 Mbyte/s. 6 (servers)  
> times 25 Mbyte/s is 150 Mbyte/s, more than I need. Compressing the data is  
> in my case no big deal since the data already came from a tape silo, were  
> once compressed on the servers and stay there compressed.*

>

> *Beside, the servers have sufficient CPU power to handle any additional load  
> for decompression and could even do compression (which I don't need at the  
> moment) without slowing down the clients. They are fast enough to serve the  
> disks for the clients which have to read and analyse the data and then  
> write back the result.*

>  
>>  
>>> *Before anybody tells me that disk space is cheap: the current set*  
>>> *contains*  
>>> *6\*8\*200 Gbytes. GZIP gives me a factor 2.5, which I really need.*  
>>> *Actually I*  
>>> *need more. ;-) Adding the difference in disks costs some 10,000 bucks.*  
>>  
>>  
>> *For a 10 terabyte system, that's not a large amount of money.*  
>  
> *But for me it is a lot.*  
>  
>>  
>> *Nick*  
>  
> *I found cfs, an encrypting userland NFS server. Unfortunately this is only*  
> *a single-threaded server. Each of my clients (all double CPU systems) may*  
> *read as many as two or more files.*  
>  
> *Does anybody know about a modified nullfs or a modified NFS server that*  
> *could do transparent decompression, and for now no compression, on the*  
> *servers?*  
>  
> *Perhaps someone who knows the details of the NFS server could point me to*  
> *the best place where to insert a decompression routine? The compressed*  
> *files already have the file extension .gz which identifies them nicely.*  
> *The server would also have to strip the extension. Any idea?*

-----  
I think you are still looking at the wrong part of this. NFS is not given a request of dealing with an entire file, or even heirarchy of files. The client asks for a byte range, a clump of data, typically less than 64KB, and that is moved over the wire. File compression operates on whole files, unlike what NFS requests do.

You are going to find the same thing with existing file systems on FBSD, say redoing nullfs to slip in compression. As a simple example, ask what you would code when an application asks for bytes N..N+4K while in the middle of a file. Is that "original receipe or extra crispy compressed bytes" being counted? And if a piece is requested, one byte added, and the piece is re-stored, then what happens to the following pieces?

I suspect that cfs item above is compressing on the wire, not on permanent storage media. But this is just my guess.

There is an operating system which does provide real file compression as an integral part of the file system, applications are unaware of it and these byte count problems do not appear. It is Novell NetWare. NetWare supports NFS, and it supports very large files and disk farms.

Joe D.