

Re: Reiser4 status: benchmarked vs. V3 (and ext3)

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-07/1767.html>

From: Yury Umanets (*umka_at_namesys.com*)

Date: 07/27/03

To: Daniel Egger <degger@fhm.edu>

Date: Sun, 27 Jul 2003 14:30:11 +0400

On Sat, 2003-07-26 at 19:21, Daniel Egger wrote:

> *Am Sam, 2003-07-26 um 16.54 schrieb Yury Umanets:*

>

> *Now we're talking. :)*

>

> > *Reiserfs cannot be used efficiently with flash, as it uses block size 4K*

> > *(by default) and usual flash block size is in range 64K - 256K.*

>

> *Don't confuse block size with erase size. The former is the layout of*

> *the fs' data on the medium while the latter is the granularity of the*

> *erase command which is important insofar that flash has to be erased (in*

> *most cases) before one can write new data on it.*

So what? I mean, that if an IO request size does not equal to flash erase size, then corresponding block device driver can't just submit data to flash, but need maintain some cache, and cache size the same as erase size for particular flash device. And in the case when WRITE request is encountered, and write sector does not equal to start sector of cached data or cache is empty, block device driver should read data from flash first to fill cache up. This is redundant IO operation.

>

> *However since you said that one can plug in a different block allocation*

> *scheme, I think it might be possible to work around that limitation by*

> *writing a block allocator which works around the limitations of the*

> *erase size.*

This is some misunderstanding :) First we've spoken about reiser4, then you asked how does reiserfs behave on flash devices and is it convenient for flash at all.

Just make sure, that we're speaking about the same thing:

Plugin-based architecture is used in reiser4, not in reiserfs (reiser3).

Reiser4 is fully different, written from the scratch filesystem.

>
> > *Also reiserfs does not use compression, that would be very nice of it*
> > *;) , because flash has limited number of erase cycles per block (in range*
> > *100.000)*
>

> *I don't see what the compression has to do with the limited number of*
> *erase/write cycles.*

Compressed data which should be written is smaller than uncompressed one, thus, its writing affects smaller number of blocks. Each block will be erased rarely, that will prolong flash live.

>
> > *and it is about three times as expensive as SDRAM.*
>
> *That's true but not important to us. The system right now fits nicely on*
> *a 128MB CF card when using ext2 or on 64MB when using JFFS2. The latter*
> *is far more stable and reliable but dogslow. Since the price difference*
> *between 128MB and 64CF is rather small and the cost of the overall*
> *system relatively high this is no argument for us.*

So, you prefer speed? What do you use for this x86 box with flash?

>
> > *So, it is better to use something more convenient. For instance jffs2.*
>
> *Convenient only insofar that it's more reliable.*

I'd not say, that ext2 is too reliable though.

> *It's a pain in the neck*
> *to setup for non hardwired flash chips and to boot, it also takes*
> *forever to mount and to write on it.*
>
> > *(1) Make the journal substantial smaller of size.*
> > *(2) Don't turn tails off. This is useful to prolong flash live.*

>
> *Thanks. But first I'll have a look at your plugin architecture to see*
> *how feasible a different implementation of block allocation especially*
> *for flash devices would be.*

You should take a look to reiser4, not to reiserfs. Don't forget :)

But I don't understand, why do you want to make changes in current block allocator plugin? In other words, what is wrong with current implementation, which is willing to allocate blocks closer one to another one?

Linux-Kernel: Re: Reiser4 status: benchmarked vs. V3 (and ext3)

I thought, if blocks lie side by side, as current block allocator does, this increases probability of flash block device cache hitting (take a look to drivers/mtd/mtdblock.c), what is definitely good. Isn't it?

Regards.

--

We're flying high, we're watching the world passes by...

-

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>