

Re: [PATCH] xfs: revert to double-buffering readdir

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2007-12/msg00132.html>

- *From:* Stephen Lord <lord@xxxxxxx>
 - *Date:* Sat, 1 Dec 2007 07:04:27 -0600
-

On Nov 30, 2007, at 5:04 PM, Chris Wedgwood wrote:

On Fri, Nov 30, 2007 at 04:36:25PM -0600, Stephen Lord wrote:

Looks like the readdir is in the bowels of the btree code when filldir gets called here, there are probably locks on several buffers in the btree at this point. This will only show up for large directories I bet.

I see it for fairly small directories. Larger than what you can stuff into an inode but less than a block (I'm not checking but fairly sure that's the case).

I told you I did not read any code..... once a directory is out of the inode and into disk blocks, there will be a lock on the buffer while the contents are copied out.

Just rambling, not a single line of code was consulted in writing this message.

Can you explain why the offset is capped and treated in an 'odd way' at all?

```
+ curr_offset = filp->f_pos;
+ if (curr_offset == 0x7fffffff)
+ offset = 0xffffffff;
+ else
+ offset = filp->f_pos;
```

Re: [PATCH] xfs: revert to double-buffering readdir

and later the offset to filldir is masked. Is that some restriction in filldir?

Too long ago to remember exact reasons. The only thing I do recall is issues with glibc readdir code which wanted to remember positions in a dir and seek backwards. It was translating structures and could end up with more data from the kernel than would fit in the user buffer. This may have something to do with that and special values used as eof markers in the getdents output and signed 32 bit arguments to lseek. In the original xfs directory code, the offset of an entry was a 64 bit hash+offset value, that really confused things when glibc attempted to do math on it.

I also recall that the offsets in the directory fields had different meanings on different OS's. Sometimes it was the offset of the entry itself, sometimes it was the offset of the next entry, that was one of the reasons for the translation layer I think.

Steve

—

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

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

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