

## Ceph - Tasks #889

### librbd.cc : more int ->uint64\_t

03/15/2011 02:08 AM - changping Wu

<b>Status:</b>	Resolved	<b>% Done:</b>	0%
<b>Priority:</b>	Normal	<b>Spent time:</b>	0.00 hour
<b>Assignee:</b>			
<b>Category:</b>	librbd		
<b>Target version:</b>	v0.25.2		
<b>Tags:</b>		<b>Affected Versions:</b>	
<b>Reviewed:</b>		<b>Pull request ID:</b>	

#### Description

Hi ,  
1.  
commit : 4ee75a881ec637e2b0c5b74b16b1e44ac710707c  
still exist some of params that need modify int ->uint64\_t .  
example : total\_read ,total\_write.

at read\_iterate function, should modify

```
"int total_read = 0;"
```

to

```
"uint64_t total_read = 0;"
```

,or rbd export file size >2GB ,still fail.

i listed some of the functions at the below.

```
int read_iterate(ImageCtx *ictx, uint64_t off, size_t len,  
int (*cb)(uint64_t, size_t, const char *, void *),  
void *arg) {  
int r = ictx_check(ictx);  
if (r < 0)  
return r; {  
if (!len)  
return 0;
```

```
int r = ictx_check(ictx);  
if (r < 0)  
return r;
```

```
int total_write = 0;
```

```
.....
```

```
int aio_read(ImageCtx *ictx, uint64_t off, size_t len,  
char *buf,  
AioCompletion *c) {  
.....
```

```
int total_read = 0;
```

```
.....
```

```
}
```

```
int write(ImageCtx *ictx, uint64_t off, size_t len, const char *buf) {  
if (!len)  
return 0;
```

```
int r = ictx_check(ictx);  
if (r < 0)
```

```
    return r;

int total_write = 0;
.....
}
```

2. to suggest, modify some of functions that "int" ->"int64\_t"

example :

modify:

```
int read_iterate(ImageCtx *ictx, uint64_t off, size_t len,
int (*cb)(uint64_t, size_t, const char *, void *),
void *arg)
```

to

```
int64_t read_iterate(ImageCtx *ictx, uint64_t off, size_t len,
int (*cb)(uint64_t, size_t, const char *, void *),
void *arg)
```

read,write,aio\_read,aio\_write ,etc.

## History

---

### #1 - 03/15/2011 09:39 AM - Sage Weil

In both of these cases, we can read no more than the `size_t len`. In practical terms, we will never read more than what will fit in addressable memory. Is there any case where those ints could *actually* overflow? I guess we could switch them to unsigned to allow a 2GB read operation on 32bit...

### #2 - 03/15/2011 09:44 AM - Sage Weil

Oh, nevermind, I see why it matters (`read_iterate` can cover more data because it goes in pieces).

should be fixed by [174aa56c4bdf56454d54b8898f21204511ed6aaa](#)

### #3 - 03/15/2011 01:18 PM - Sage Weil

- Status changed from New to Resolved

### #4 - 03/15/2011 01:18 PM - Sage Weil

- Target version set to v0.25.2