

```

*****
84039 Thu Feb 6 16:49:02 2014
new/usr/src/uts/common/fs/zfs/dsl_dataset.c
4574 get_clones_stat does not call zap_count in non-debug kernel
Reviewed by: Matthew Ahrens <mahrens@delphix.com>
Reviewed by: Marcel Telka <marcel@telka.sk>
Approved by:
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22 * Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
23 * Copyright (c) 2013 by Delphix. All rights reserved.
24 * Copyright (c) 2012, Joyent, Inc. All rights reserved.
25 * Copyright (c) 2014 RackTop Systems.
26 */

28 #include <sys/dmu_objset.h>
29 #include <sys/dsl_dataset.h>
30 #include <sys/dsl_dir.h>
31 #include <sys/dsl_prop.h>
32 #include <sys/dsl_synctask.h>
33 #include <sys/dmu_traverse.h>
34 #include <sys/dmu_impl.h>
35 #include <sys/dmu_tx.h>
36 #include <sys/arc.h>
37 #include <sys/zio.h>
38 #include <sys/zap.h>
39 #include <sys/zfeature.h>
40 #include <sys/unique.h>
41 #include <sys/zfs_context.h>
42 #include <sys/zfs_ioctl.h>
43 #include <sys/spa.h>
44 #include <sys/zfs_znode.h>
45 #include <sys/zfs_onexit.h>
46 #include <sys/zvol.h>
47 #include <sys/dsl_scan.h>
48 #include <sys/dsl_deadlist.h>
49 #include <sys/dsl_destroy.h>
50 #include <sys/dsl_userhold.h>
51 #include <sys/dsl_bookmark.h>

53 #define SWITCH64(x, y) \
54     { \
55         uint64_t __tmp = (x); \
56         (x) = (y); \
57         (y) = __tmp; \
58     }

```

```

60 #define DS_REF_MAX        (1ULL << 62)

62 #define DSL_DEADLIST_BLOCKSIZE SPA_MAXBLOCKSIZE

64 /*
65  * Figure out how much of this delta should be propagated to the dsl_dir
66  * layer.  If there's a reservation, that space has already been
67  * partially accounted for in our ancestors.
68  */
69 static int64_t
70 parent_delta(dsl_dataset_t *ds, int64_t delta)
71 {
72     uint64_t old_bytes, new_bytes;

74     if (ds->ds_reserved == 0)
75         return (delta);

77     old_bytes = MAX(ds->ds_phys->ds_unique_bytes, ds->ds_reserved);
78     new_bytes = MAX(ds->ds_phys->ds_unique_bytes + delta, ds->ds_reserved);

80     ASSERT3U(ABS((int64_t)(new_bytes - old_bytes)), <=, ABS(delta));
81     return (new_bytes - old_bytes);
82 }
    unchanged_portion_omitted

1350 static void
1351 get_clones_stat(dsl_dataset_t *ds, nvlist_t *nv)
1352 {
1353     uint64_t count = 0;
1354     objset_t *mos = ds->ds_dir->dd_pool->dp_meta_objset;
1355     zap_cursor_t zc;
1356     zap_attribute_t za;
1357     nvlist_t *propval = fnvlist_alloc();
1358     nvlist_t *val = fnvlist_alloc();

1360     ASSERT(dsl_pool_config_held(ds->ds_dir->dd_pool));

1362     /*
1363      * There may be missing entries in ds_next_clones_obj
1364      * due to a bug in a previous version of the code.
1365      * Only trust it if it has the right number of entries.
1366      */
1367     if (ds->ds_phys->ds_next_clones_obj != 0) {
1368         VERIFY0(zap_count(mos, ds->ds_phys->ds_next_clones_obj,
1369             &count));
1370     }
1371     if (count != ds->ds_phys->ds_num_children - 1)
1372         goto fail;
1373     for (zap_cursor_init(&zc, mos, ds->ds_phys->ds_next_clones_obj);
1374         zap_cursor_retrieve(&zc, &za) == 0;
1375         zap_cursor_advance(&zc)) {
1376         dsl_dataset_t *clone;
1377         char buf[ZFS_MAXNAMELEN];
1378         VERIFY0(dsl_dataset_hold_obj(ds->ds_dir->dd_pool,
1379             za.za_first_integer, FTAG, &clone));
1380         dsl_dir_name(clone->ds_dir, buf);
1381         fnvlist_add_boolean(val, buf);
1382         dsl_dataset_rele(clone, FTAG);
1383     }
1384     zap_cursor_fini(&zc);
1385     fnvlist_add_nvlist(propval, ZPROP_VALUE, val);
1386     fnvlist_add_nvlist(nv, zfs_prop_to_name(ZFS_PROP_CLONES), propval);
1387 fail:
1388     nvlist_free(val);

```

new/usr/src/uts/common/fs/zfs/dsl_dataset.c

3

```
1389         nvlist_free(propval);  
1390     }  
_____unchanged_portion_omitted_____
```