

```

*****
50305 Fri Aug 2 15:41:16 2013
new/usr/src/uts/common/fs/zfs/dsl_scan.c
3970 add tunable for maximum number of blocks freed in one txg
Reviewed by: Matthew Ahrens <mahrens@delphix.com>
Reviewed by: Adam Leventhal <ahl@delphix.com>
*****
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) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
23 * Copyright (c) 2013 by Delphix. All rights reserved.
24 */

26 #include <sys/dsl_scan.h>
27 #include <sys/dsl_pool.h>
28 #include <sys/dsl_dataset.h>
29 #include <sys/dsl_prop.h>
30 #include <sys/dsl_dir.h>
31 #include <sys/dsl_synctask.h>
32 #include <sys/dnode.h>
33 #include <sys/dmu_tx.h>
34 #include <sys/dmu_objset.h>
35 #include <sys/arc.h>
36 #include <sys/zap.h>
37 #include <sys/zio.h>
38 #include <sys/zfs_context.h>
39 #include <sys/fs/zfs.h>
40 #include <sys/zfs_znode.h>
41 #include <sys/spa_impl.h>
42 #include <sys/vdev_impl.h>
43 #include <sys/zil_impl.h>
44 #include <sys/zio_checksum.h>
45 #include <sys/ddt.h>
46 #include <sys/sa.h>
47 #include <sys/sa_impl.h>
48 #include <sys/zfeature.h>
49 #ifdef _KERNEL
50 #include <sys/zfs_vfsops.h>
51 #endif

53 typedef int (scan_cb_t)(dsl_pool_t *, const blkptr_t *, const zbookmark_t *);

55 static scan_cb_t dsl_scan_defrag_cb;
56 static scan_cb_t dsl_scan_scrub_cb;
57 static scan_cb_t dsl_scan_remove_cb;
58 static void dsl_scan_cancel_sync(void *, dmu_tx_t *);
59 static void dsl_scan_sync_state(dsl_scan_t *, dmu_tx_t *tx);

```

```

61 int zfs_top_maxinflight = 32; /* maximum I/Os per top-level */
62 int zfs_resilver_delay = 2; /* number of ticks to delay resilver */
63 int zfs_scrub_delay = 4; /* number of ticks to delay scrub */
64 int zfs_scan_idle = 50; /* idle window in clock ticks */

66 int zfs_scan_min_time_ms = 1000; /* min millisecs to scrub per txg */
67 int zfs_free_min_time_ms = 1000; /* min millisecs to free per txg */
68 int zfs_resilver_min_time_ms = 3000; /* min millisecs to resilver per txg */
69 boolean_t zfs_no_scrub_io = B_FALSE; /* set to disable scrub i/o */
70 boolean_t zfs_no_scrub_prefetch = B_FALSE; /* set to disable scrub prefetching */
71 enum ddt_class zfs_scrub_ddt_class_max = DDT_CLASS_DUPLICATE;
72 int dsl_scan_delay_completion = B_FALSE; /* set to delay scan completion */
73 /* max number of blocks to free in a single TXG */
74 uint64_t zfs_free_max_blocks = UINT64_MAX;

76 #define DSL_SCAN_IS_SCRUB_RESILVER(scn) \
77 ((scn->scn_phys.scn_func == POOL_SCAN_SCRUB || \
78 (scn->scn_phys.scn_func == POOL_SCAN_RESILVER))

80 extern int zfs_txg_timeout;

82 /* the order has to match pool_scan_type */
83 static scan_cb_t *scan_funcs[POOL_SCAN_FUNCS] = {
84     NULL,
85     dsl_scan_scrub_cb, /* POOL_SCAN_SCRUB */
86     dsl_scan_scrub_cb, /* POOL_SCAN_RESILVER */
87 };
_____ unchanged_portion_omitted _____

1315 static boolean_t
1316 dsl_scan_free_should_pause(dsl_scan_t *scn)
1317 {
1318     uint64_t elapsed_nanosecs;

1320     if (scn->scn_visited_this_txg >= zfs_free_max_blocks)
1321         return (B_TRUE);

1323     elapsed_nanosecs = gethrtime() - scn->scn_sync_start_time;
1324     return (elapsed_nanosecs / NANOSEC > zfs_txg_timeout ||
1325         (NSEC2MSEC(elapsed_nanosecs) > zfs_free_min_time_ms &&
1326         txg_sync_waiting(scn->scn_dp)) ||
1327         spa_shutting_down(scn->scn_dp->dp_spa));
1328 }
_____ unchanged_portion_omitted _____

```