```
**********************************************************
  149009 Thu Nov 20 20:48:54 2014
new/usr/src/uts/common/fs/zfs/arc.c
5179 remove unused ZFS ARC functions
**********************************************************
______unchanged_portion_omitted_

1313 void *
1314 arc_data_buf_alloc(uint64_t size)
1315 {
1316         if (arc_evict_needed(ARC_BUFC_DATA))
1317                 cv_signal(&arc_reclaim_thr_cv);
1318         atomic_add_64(&arc_size, size);
1319         return (zio_data_buf_alloc(size));
1320 }

1322 void
1323 arc_data_buf_free(void *buf, uint64_t size)
1324 {
1325         zio_data_buf_free(buf, size);
1326         ASSERT(arc_size >= size);
1327         atomic_add_64(&arc_size, -size);
1328 }

1313 arc_buf_t *
1314 arc_buf_alloc(spa_t *spa, int size, void *tag, arc_buf_contents_t type)
1315 {
1316         arc_buf_hdr_t *hdr;
1317         arc_buf_t *buf;

1319         ASSERT3U(size, >, 0);
1320         hdr = kmem_cache_alloc(hdr_cache, KM_PUSHPAGE);
1321         ASSERT(BUF_EMPTY(hdr));
1322         hdr->b_size = size;
1323         hdr->b_type = type;
1324         hdr->b_spa = spa_load_guid(spa);
1325         hdr->b_state = arc_anon;
1326         hdr->b_arc_access = 0;
1327         buf = kmem_cache_alloc(buf_cache, KM_PUSHPAGE);
1328         buf->b_hdr = hdr;
1329         buf->b_data = NULL;
1330         buf->b_efunc = NULL;
1331         buf->b_private = NULL;
1332         buf->b_next = NULL;
1333         hdr->b_buf = buf;
1334         arc_get_data_buf(buf);
1335         hdr->b_datacnt = 1;
1336         hdr->b_flags = 0;
1337         ASSERT(refcount_is_zero(&hdr->b_refcnt));
1338         (void) refcount_add(&hdr->b_refcnt, tag);

1340         return (buf);
1341 }
______unchanged_portion_omitted_
```

****
4446 Thu Nov 20 20:48:55 2014
new/usr/src/uts/common/fs/zfs/sys/arc.h
5179 remove unused ZFS ARC functions
****
_____unchanged_portion_omitted_

```
  84 void arc_space_consume(uint64_t space, arc_space_type_t type);
  85 void arc_space_return(uint64_t space, arc_space_type_t type);
  86 void *arc_data_buf_alloc(uint64_t space);
  87 void arc_data_buf_free(void *buf, uint64_t space);
  86 arc_buf_t *arc_buf_alloc(spa_t *spa, int size, void *tag,
  87     arc_buf_contents_t type);
  88 arc_buf_t *arc_loan_buf(spa_t *spa, int size);
  89 void arc_return_buf(arc_buf_t *buf, void *tag);
  90 void arc_loan_inuse_buf(arc_buf_t *buf, void *tag);
  91 void arc_buf_add_ref(arc_buf_t *buf, void *tag);
  92 boolean_t arc_buf_remove_ref(arc_buf_t *buf, void *tag);
  93 int arc_buf_size(arc_buf_t *buf);
  94 void arc_release(arc_buf_t *buf, void *tag);
  95 int arc_released(arc_buf_t *buf);
  96 void arc_buf_freeze(arc_buf_t *buf);
  97 void arc_buf_thaw(arc_buf_t *buf);
  98 boolean_t arc_buf_eviction_needed(arc_buf_t *buf);
  99 #ifdef ZFS_DEBUG
 100 int arc_referenced(arc_buf_t *buf);
 101 #endif

 103 int arc_read(zio_t *pio, spa_t *spa, const blkptr_t *bp,
 104     arc_done_func_t *done, void *private, zio_priority_t priority, int flags,
 105     uint32_t *arc_flags, const zbookmark_phys_t *zb);
 106 zio_t *arc_write(zio_t *pio, spa_t *spa, uint64_t txg,
 107     blkptr_t *bp, arc_buf_t *buf, boolean_t l2arc, boolean_t l2arc_compress,
 108     const zio_prop_t *zp, arc_done_func_t *ready, arc_done_func_t *physdone,
 109     arc_done_func_t *done, void *private, zio_priority_t priority,
 110     int zio_flags, const zbookmark_phys_t *zb);
 111 void arc_freed(spa_t *spa, const blkptr_t *bp);

 113 void arc_set_callback(arc_buf_t *buf, arc_evict_func_t *func, void *private);
 114 boolean_t arc_clear_callback(arc_buf_t *buf);

 116 void arc_flush(spa_t *spa);
 117 void arc_tempreserve_clear(uint64_t reserve);
 118 int arc_tempreserve_space(uint64_t reserve, uint64_t txg);

 120 void arc_init(void);
 121 void arc_fini(void);

 123 /*
 124  * Level 2 ARC
 125  */

 127 void l2arc_add_vdev(spa_t *spa, vdev_t *vd);
 128 void l2arc_remove_vdev(vdev_t *vd);
 129 boolean_t l2arc_vdev_present(vdev_t *vd);
 130 void l2arc_init(void);
 131 void l2arc_fini(void);
 132 void l2arc_start(void);
 133 void l2arc_stop(void);

 135 #ifndef _KERNEL
 136 extern boolean_t arc_watch;
 137 extern int arc_procfd;
 138 #endif

 140 #ifdef  __cplusplus
```



```
 141 }
```
_____unchanged_portion_omitted_