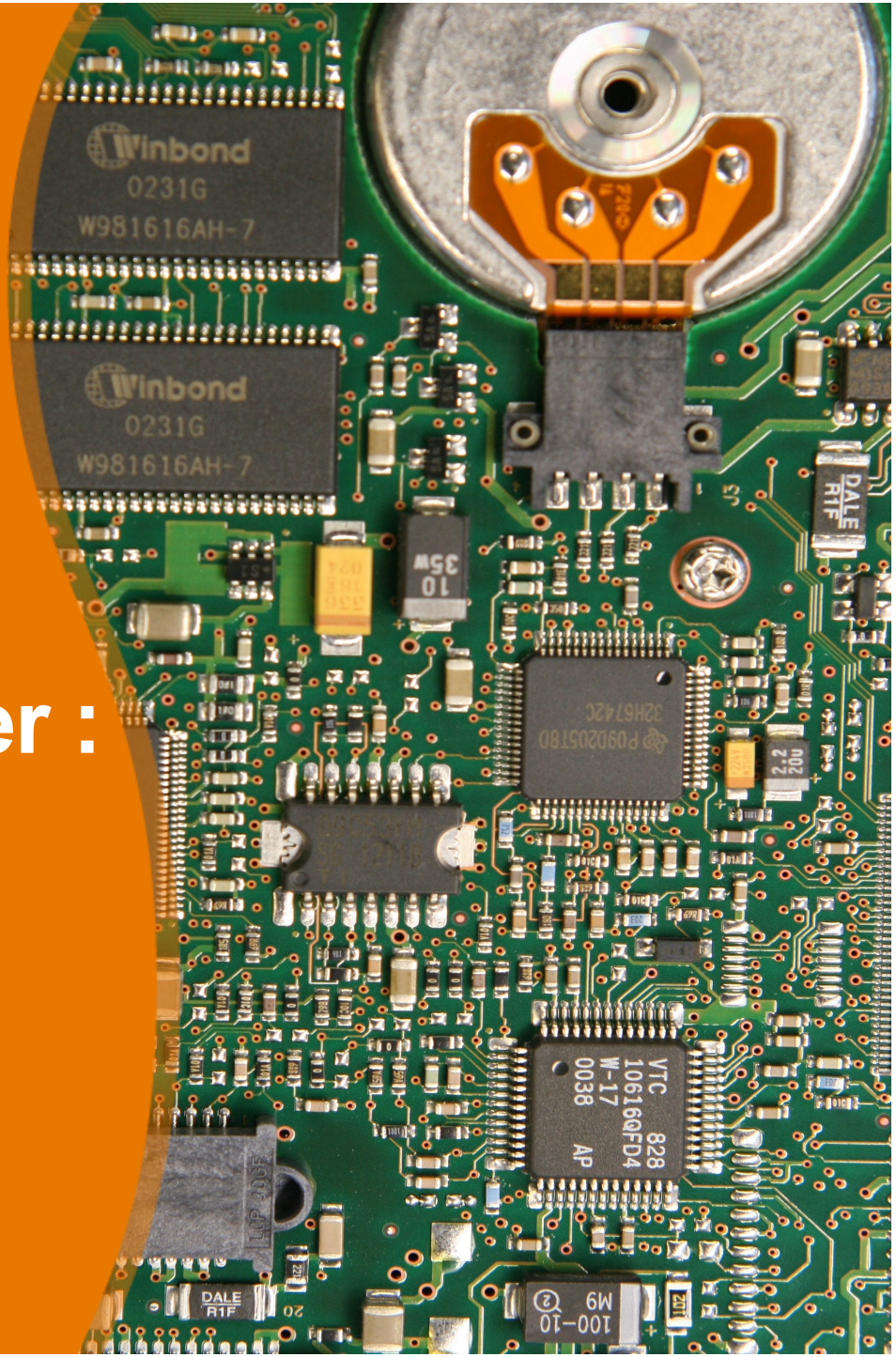
Solaris Volume Manager : FailFast

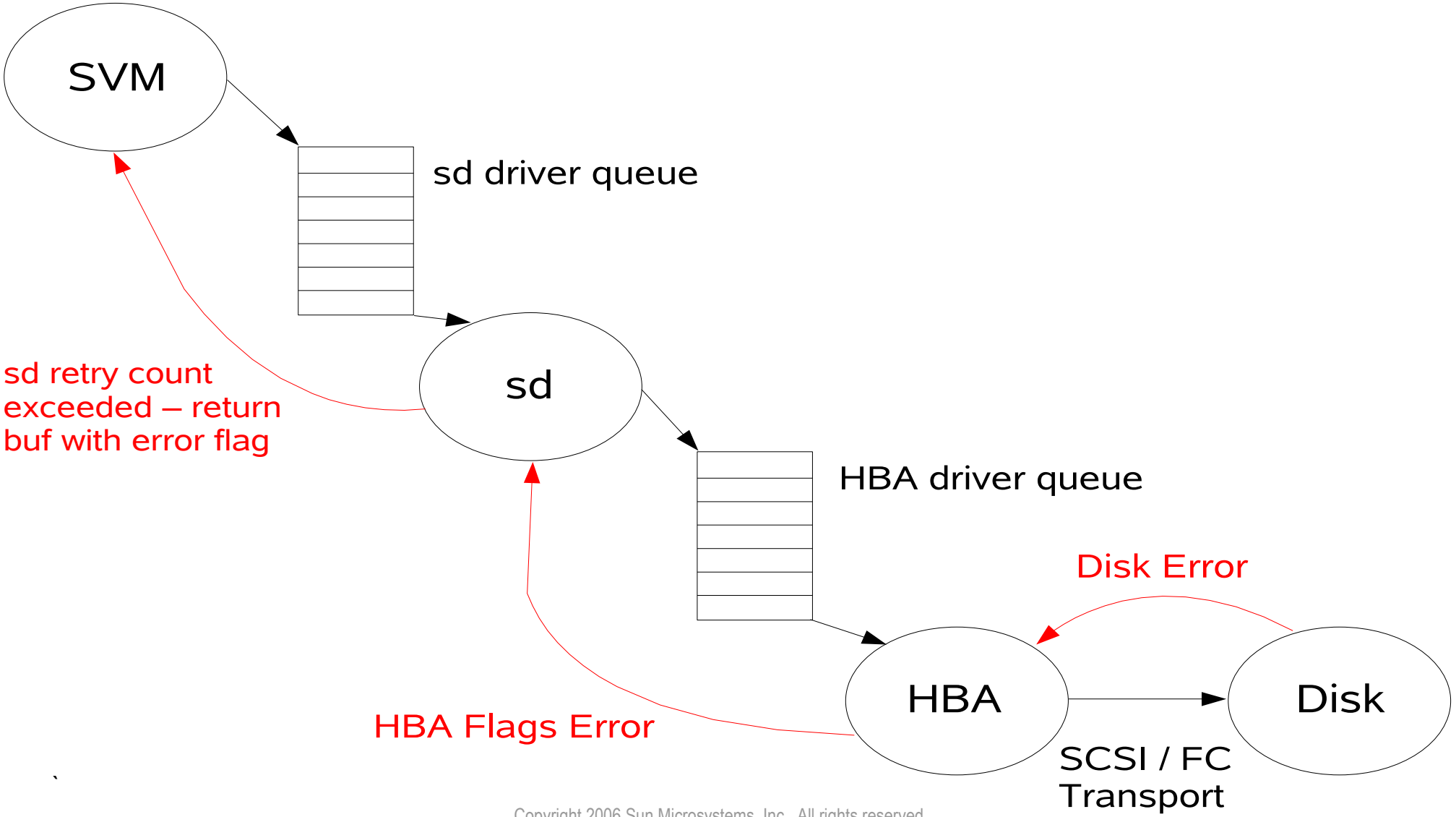


FailFast

- The Problem
- Two-Stage Solution
  - > sd driver changes
  - > SVM code changes

Copyright 2006 Sun Microsystems, Inc. All rights reserved.

Driver Retries – Retries Failed



Copyright 2006 Sun Microsystems, Inc. All rights reserved.

Driver Retries

- Can be very slow
  - > Each retry for a selection timeout can be 60 seconds
  - > Each retry goes to be back of the queue
  - > Other I/O's ahead also each take 60 seconds to fail
- No failure back to SVM until retries are exhausted
- Application may timeout before error is noticed
- Clusters may try to failover assuming a hang

`

Copyright 2006 Sun Microsystems, Inc. All rights reserved.

Bug 4500536

- Introduced the B_FAILFAST interface
- Allows an I/O to be tagged as for a “reliable device”
- These I/O's can then be errored faster by not retrying
- All tagged I/O's will be errored when any one fails
- Devices support for B_FAILFAST flagged as ddi property
  - > ddi-failfast-supported property

`

Copyright 2006 Sun Microsystems, Inc. All rights reserved.

DDI Changes – sd driver

```
        /*
         * Add a boolean property to tell the world we support
         * the B_FAILFAST flag (for layered drivers)
         */
        (void) ddi_prop_create(DDI_DEV_T_NONE, devi,
DDI_PROP_CANSLEEP,
            "ddi-failfast-supported", NULL, 0);
```

`

Copyright 2006 Sun Microsystems, Inc. All rights reserved.

SVM Implementation

- Sets B_FAILFAST for sub-mirrors
  - > Checks all sub-mirrors support B_FAILFAST
  - > mirror_check_failfast() does the work
  - > Sets MD_SM_FAILFAST flag on mirror
- Allows for a single retry when an I/O fails
- Only used when two or more sub-mirrors are OKAY
  - > B_FAILFAST turned off for last good side

Copyright 2006 Sun Microsystems, Inc. All rights reserved.

SVM Implementation

```
                if (un->un_sm[i].sm_flags & MD_SM_FAILFAST
&&
                    cs != NULL) {
                        cs->cs_buf.b_flags |= B_FAILFAST;
                }

        cb = md_bioclone(pb, offset, bcount, dev, blkno,
    mirror_done,
            cb, KM_NOSLEEP);
        if (war)
                cb->b_flags = (cb->b_flags & ~B_READ) | B_WRITE;

        if (un->un_sm[i].sm_flags & MD_SM_FAILFAST) {
                cb->b_flags |= B_FAILFAST;
        }
```

Copyright 2006 Sun Microsystems, Inc. All rights reserved.

SVM Implementation

```
if (cb->b_edev ==
md_dev64_to_dev(un->un_sm[i].sm_dev)) {
    /*
     * This is the submirror that had the error.
     * Check if it supports failfast.
     */
    if (un->un_sm[i].sm_flags & MD_SM_FAILFAST) {
            daemon_queue_t *dqp;

            mutex_exit(&ps->ps_mx);
            dqp = (daemon_queue_t *)cs;
            dqp->dq_prev = dqp->dq_next = NULL;
            daemon_request(&md_done_daemon,
                mirror_retry, dqp, REQ_OLD);
            return (1);
    }
```

Copyright 2006 Sun Microsystems, Inc. All rights reserved.

Metadb Handling

- I/O's to metadb's do not use B_FAILFAST
- These I/O's are fixed to a single retry
  - > Use the normal sd retry mechanism as well
- Can still slow down I/O handling on a device
  - > Mirror resync regions are in the metadbs

Copyright 2006 Sun Microsystems, Inc. All rights reserved.

SVM FailFast