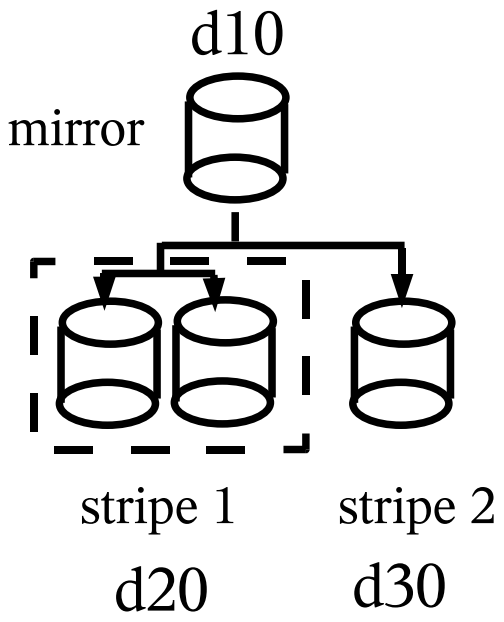
Solaris Volume Manager : Mirror Init Example

SVM Mirror Init Example

- Worked example of code flow
- Creating a mirror from scratch



SVM Mirror Init Example

- Commands :
  - > metainit d30 1 1 /dev/dsk/c0t1d0s0
  - > metainit d10 -m d30
- Follow the stripe initialisation, then the mirror

Metainit – Start of Day Code

```
main
    sdssc_bind_library
      dlopen
    md_init
      md_init_nosig
        open_admin
        metaflushnames
         metaflushhspnames
         metaflushdrivenames
         metaflushsetnames
         metaflushctlrcache
         metaflushfastnames
         metaflushstatcache
      sigfillset
      md_pushsig
    meta_check_root
      geteuid
    getopt
    meta_setup_db_locations
    init_name
```

Metainit – Start of Day

- Bind SunCluster library
  - > Proxy commands to primary node if applicable
- Load drivers
  - > Could be initial config at boot
- Open admin device
  - > Kernel level called via ioctls
- Install signal handler
- Check user privilges
  - > Must run as root

`

Metainit – Initialise Device Code

```
getopt
meta_setup_db_locations
init_name
  meta_tab_parse
  is_hspname
  is_metaname
    parse_metadevice
  meta_init_name
    is_hspname
    is_metaname
    meta_name_getname
    meta_init_make_device
     metaioctl(MD_IOCMAKE_DEV)
                    ddi_copyin_data
                    mkdev_ioctl
                      md_create_minor_node
                        ddi_create_minor_node
                    ddi_copyout
     di_devlink_init
    meta_init_stripe
     metaname
       metaname_common
```

Metainit – Initialise Device

- Parse md.tab
  - > Can specify device either in md.tab or on cmd line
  - > Cannot mix both sources
- Create device nodes
  - > metaioctl call to md driver
  - > Kernel call to ddi_create_minor_node
- Determine metadevice type
- Call specific device init routine

Metainit – Stripe Code

```
di_devlink_init
meta_init_stripe
 metaname
   metaname_common
     meta_name_getname
     getrawnames
      getrname
     metainitdrivename
     getparts
     metainitname
 metachkmeta
   metaismeta
metagetmiscname
   meta_getminor
   metaioctl(MD_IOCGET_DRVNM)
                  ddi_copyin
                  getdrvnm_ioctl
                    md_snarf_db_set
                    md_get_setstatus
                    MD_SETDRIVERNAME
                  ddi_copyout
[Parse the cmd line options]
[Build entries for each component in memory]
meta_create_stripe
```

Metainit – Stripe Code

- Create new device entries in kernel memory
- Get driver name for each component
- Parse stripe cmd line arguments
  - > Interlace, hotspare pool etc
- Build in-core entries for each component
  - > Store dev_t for each component

`

Metainit – Create Stripe Code

```
[Build entries for each component in memory]
meta_create_stripe
  [Calculate stripe unit size]
  roundup
  [Walk each component : ]
    metagetsize
     metagetvtoc
       ioctl(DKIOCGGEOM)
    metagetstart
     metagetend
       [Check for statedb, and add offset if required]
     metagetlabel
       [Check for disk label, and add offset if req.]
     metagetgeom
       [Round up to nearest cylinder]
   [Round down to interlace size]
   [Round down to size of smallest component]
   add_key_name
    build_sidenamelist
    add_name
      metaioctl(MD_IOCSET_NM)
                ddi_copyin
```

Metainit – Create Stripe

- Calculate stripe unit size
  - > Interlace x number of components in a row
- Walk each component
  - > Calculate the safe starting point
  - > Round to fit the interlace size
  - > Round down to size of smallest component
  - > Add metadb replica record

Metainit – Add Replica Record Code

```
add_key_name
 build_sidenamelist
 add_name
   metaioctl(MD_IOCSET_NM)
              ddi_copyin
              setnm_ioctl
                md_snarf_db_set
                md_get_setstatus
                ddi_copyin
                md_setdevname
                  makedevice
                  get_first_record
                   create_record
                     mddb_createrec
                       mddb_setenter
                       checkstate
                        md_get_setstatus
                        selectreplicas
                       [Create & add new record entry]
                       [Write entry to replicas]
                     mddb_commitrecs_wrapper
                       mddb_commitrecs
```

Metainit – Add Replica Record

- metaioctl call to md driver
- Snarf the metaset if not already done
- Get metaset status
- Get first record, allocate new record if none found
- Add data to record
- Write to mddb's on disk

Metainit – Add Replica Record Code

```
                    [Write entry to replicas]
                  mddb_commitrecs_wrapper
                    mddb_commitrecs
                     mddb_setenter
                     checkstate
                     [Check each replica for latest update]
                     [Update commit count and timestamp]
                [Create new namespace record entry]
                mddb_commitrecs_wrapper
                [Add devids for new devices]
            ddi_copyout
    empty_devicelist
  [Setup component entry]
[Setup row entry, check device size limit]
[Setup stripe size]
stripe_geom
```

Metainit – Add Replica Record

- Commit new record
  - > Check each on-disk replica for the latest entry
  - > Update commit count to acknowledge the entry
  - > Update replica timestamp
- Create in-core record
- Add deviceid data to records

Metainit – Build Geometry Code

```
   [Setup component entry]
[Setup row entry, check device size limit]
[Setup stripe size]
stripe_geom
   [Walk each row]
    metagetgeom
       metagetvtoc
          ioctl(DKIOCGGEOM)
    [Set each row to size of smallest row]
   metagetgeom
   metagetmiscname
    metaioctl(MD_IOCGET_DRVNM)
                  ddi_copyin
                  getdrvnm_ioctl
                    md_snarf_db_set
                    md_getsetstatus
                    MD_SETDRIVERNAME
                  ddi_copyout
   meta_setup_geom
   [Check any truncation leaves a valid device behind]
meta_check_devicesize
metaioctl(MD_IOCSET)
```

Metainit – Build Geometry

- Calculate stripe size after rounding of components
- Get geometry of each component
  - > DKIOCGGEOM ioctl
- Set each row to size of smallest row
- Get geometry of first component
- Check first component for SoftPartition
  - > Allow space for watermarks if it is
- Fake geometry from first device
- Check any truncation left a valid device behind

Metainit – Build In-Core Device Code

```
meta_check_devicesize
metaioctl(MD_IOCSET)
        ddi_copyin
        stripe_set
          md_get_setstatus
          md_getshared_key
            md_load_namespace
            getshared_key
             get_first_record
             lookup_shared_entry
               [Walk records, match on name or devid]
          mddb_createrec
          ddi_copyin
          stripe_build_incore
            [Walk each component in each row]
             md_getdevnum
               md_loadnamespace
               get_first_record
               lookup_entry
               getshared_name
                 get_first_record
                 lookup_shared_entry
               ddi_lyr_devid_to_devlist
               md_expldev
```

Metainit – Build In-Core Device Code

md_expldev
ddi_major_to_name
ddi_lyr_free_devlist
build_device_number
  getshared_name
  ddi_name_to_major
  md_makedevice
md_getmajor
  md_xlate_targ_2_mini
   md_cmpldev
   md_expldev
  ddi_lyr_get_devid
  ddi_lyr_get_minor_name
  [Add deviceid for each component to the record]
  mddb_commirrecs_wrapper
  ddi_devid_free
  md_makedevice
[Add parent data to each component's record]
mddb_commitrecs_wrapper
md_create_unit_incore
  md_get_setstatus
  [Setup mdi_unit_t structure]

Metainit – Build In-Core Device

- Get deviceid based record for the stripe
  - > Shared record type can be imported on other hosts
- Create in-core record space
- Copy in record data
  - > Major / minor numbers
  - > Resulting dev_t entry
  - > Component deviceid data
- Change parent entry in component records
  - > Set parent to be the new stripe

`

Metainit – mdi_unit_t_Structure Code

```
                    [Add parent data to each component's record]
                    mddb_commitrecs_wrapper
                    md_create_unit_incore
                      md_get_setstatus
                      [Setup mdi_unit_t structure]
                  ddi_copyout
metafreenamelist
  Free
   free
```

Metainit – mdi_unit_t Structure

- Allocate mdi_unit_t with md_create_unit_incore

```
typedef struct mdi_unit {
        md_link_t       ui_link;
        ulong_t         ui_readercnt;   /* number of unit readers */
        ulong_t         ui_wanabecnt;   /* # pending on becoming unit writer */
        ulong_t         ui_lock;
        kmutex_t        ui_mx;
        kcondvar_t      ui_cv;
        int             ui_opsindex;
        uint_t          ui_ocnt[OTYPCNT]; /* open counts */
        md_io_lock_t    *ui_io_lock;    /* pointer to io lock */
        kstat_t         *ui_kstat;      /* kernel statistics */
        kthread_id_t    ui_owner;       /* writer thread */
        uint_t          ui_tstate;      /* transient state bits */
        uint_t          ui_capab;       /* Capability bits supported */
} mdi_unit_t;
```

Metainit – Cleanup & Exit Code

[Setup mdi_unit_t structure]
ddi_copyout
metafreenamelist
Free
free
printf
meta_free_stripe
[Walk each row]
Free
Free
meta_update_md_cf
meta_print_all
meta_print_trans
meta_logs_print
meta_mirror_print
meta_raid_print
meta_stripe_print
meta_sp_print
meta_hsp_print
md_exit

Metainit – Cleanup & Exit

- Free() structures
  - > Wrapper around free()
- Display message
  - > “d30: Concat/Stripe is setup”
- Free each component's working space
- Update md.cf file : meta_update_md_cf
  - > Overwrites existing file

`

Metainit – Start of Day Code

```
main
    sdssc_bind_library
      dlopen
    md_init
      md_init_nosig
        open_admin
        metaflushnames
         metaflushhspnames
         metaflushdrivenames
         metaflushsetnames
         metaflushctlrcache
         metaflushfastnames
         metaflushstatcache
      sigfillset
      md_pushsig
    meta_check_root
      geteuid
    getopt
    meta_setup_db_locations
    init_name
      meta_tab_parse
      is_hspname
      is_metaname
        parse_metadevice
```

Metainit – Start of Day Code

```
meta_tab_parse
is_hspname
is_metaname
  parse_metadevice
meta_init_name
  is_hspname
  is_metaname
  meta_name_getname
  meta_init_make_device
   metaioctl(MD_IOCMAKE_DEV)
                    ddi_copyin_data
                    mkdev_ioctl
                      md_create_minor_node
                        ddi_create_minor_node
                    ddi_copyout
   di_devlink_init
  meta_init_mirror
   metaname
```

Metainit – Start of Day

- As before :
  - > Bind SC library
  - > Load drivers
  - > Open admin device
  - > Install signal handler
  - > Check user privileges
- Parse md.tab
- Create device nodes
- Determine metadevice type

`

Metainit – Mirror Code

di_devlink_init
meta_init_mirror
metaname
  metaname_common
    meta_name_getname
    getrawnames
     getrname
    metainitdrivename
    getparts
    metainitname
metachkmeta
  metaismeta
metagetmiscname
  meta_getminor
[Parse options]
[Allocate md_mirror_t structure]
[Walk sub-mirrors]
  metaname

Metainit – Mirror Code

- Create new name pointer entries in kernel memory
- Parse mirror cmd line arguments
- Allocate md_mirror_t structure

```
struct md_mirror_t {
        md_common_t common;
        mm_rd_opt_t read_option;
        mm_wr_opt_t write_option;
        mm_pass_num_t pass_num;
        int percent_done;
        int percent_dirty;
        md_submirror_t submirrors[NMIRROR];
};
typedef struct md_mirror_t md_mirror_t;
```

Metainit – Check Sub-MIrrors Code

meta_getminor
[Parse options]
[Allocate md_mirror_t structure]
[Walk sub-mirrors]
  metaname
  [Add each sub-mirror to md_mirror_t structure]
meta_create_mirror
  meta_check_mirror
    [Walk sub-mirrors]
     [Count number of sub-mirrors present]
    [Walk sub-mirrors]
     meta_check_submirror
       metachkmeta
        metaismeta
       meta_check_primary_mirror
        meta_get_current_root
       meta_check_inuse
        meta_check_mounted
        meta_check_swapped
        meta_check_dump

Metainit – Check Sub-Mirrors Code

```
 meta_check_dump
meta_check_inset
 metaislocalset
 metagetset
 metaissameset
meta_get_unit
 metachkmeta
 meta_get_stripe
   meta_get_stripe_common
     metagetmiscname
     meta_get_mdunit
      metachkmeta
      metagetmiscname
      MD_SETDRIVERNAME
      meta_getminor
      metaioctl(MD_IOCGET)
          ddi_copyin
          stripe_get
             ddi_copyout
     [Walk stripe rows]
      [Populate component data]
     Free
     meta_free_stripe
  [Check device can accept a parent device]
metagetsize
```

Metainit – Check Sub-Mirrors

- Must have at least one
- Ignore any greater than NMIRROR
- Must be a metadevice
- Are we mirroring root ?
  - > Block metattach until after a remount
- Check for mounted fs, swap or dump devices
  - > Require force flag
- Check components are in the same set
- Check sub-mirrors can accept a parent device

Metainit – Initialise Sub-Mirrors Code

```
            meta_free_stripe
          [Check device can accept a parent device]
        metagetsize
          metagetvtoc
            ioctl(DKIOCGGEOM)
        check_twice
          meta_check_overlap
            meta_check_samedrive
            metagetvtoc
      [Allocate mm_unit_t structure]
      meta_gettimeofday
        gettimeofday
```

Metainit – Initialise Sub-Mirrors

- Get size of each sub-mirror
- Check no component is used twice
- Allocate mm_unit_t structure

`

Metainit – mm_unit_t Structure

```
typedef struct mm_unit {
        mdc_unit_t      c;                  /* common stuff */

        int             un_last_read;       /* last submirror index read */
        uint_t          un_changecnt;
        ushort_t        un_nsm;             /* number of submirrors */
        mm_submirror_t  un_sm[NMIRROR];
        int             un_ovrlap_chn_flg;
        mm_rd_opt_t     un_read_option;     /* mirror read option */
        mm_wr_opt_t     un_write_option;    /* mirror write option */
        mm_pass_num_t   un_pass_num;        /* resync pass number */
```

Metainit – mm_unit_t Structure

```
/*
* following used to keep dirty bitmaps
*/
uint_t          un_resync_flg;
uint_t          un_waiting_to_mark;
uint_t          un_waiting_to_commit;
uint_t          un_rrd_blksize;   /* The blocksize of the dirty bits */
uint_t          un_rrd_num;       /* The number of resync regions */
mddb_recid_t    un_rr_dirty_recid; /* resync region bm db record id */
```

`

Metainit – mm_unit_t Structure

```
/*
 * following stuff is private to resync process
 */
int             un_rs_copysize;
int             un_rs_dests;            /* destinations */
diskaddr_t      un_rs_resync_done;      /* used for percent done */
diskaddr_t      un_rs_resync_2_do;      /* user for percent done */
int             un_rs_dropped_lock;
uint_t          un_rs_type;             /* type of resync */
/*
 * Incore only elements
 */
mm_submirror_ic_t un_smic[NMIRROR];     /* NMIRROR elements array */
mm_mirror_ic_t  un_mmic;
kmutex_t        un_rrp_inflight_mx;
```

Metainit – mm_unit_t Structure

```
        /*
         * resync thread control
         */
        kthread_t       *un_rs_thread;          /* Resync thread ID */
        kmutex_t        un_rs_thread_mx;        /* Thread cv mutex */
        kcondvar_t      un_rs_thread_cv;        /* Cond. Var. for thread */
        uint_t          un_rs_thread_flags;     /* Thread control flags */
        md_mps_t        *un_rs_prev_ovrlap;     /* existing overlap request */
        timeout_id_t    un_rs_resync_to_id;     /* resync progress timeout */
        kmutex_t        un_rs_progress_mx;      /* Resync progress mutex */
        kcondvar_t      un_rs_progress_cv;      /* Cond. Var. for progress */
        uint_t          un_rs_progress_flags;   /* Thread control flags */
        void            *un_rs_msg;             /* Intra-node resync message */
} mm_unit_t;
```

Metainit – Initialise Sub-Mirrors Code

```
        metagetvtoc
[Allocate mm_unit_t structure]
meta_gettimeofday
  gettimeofday
[Walk sub-mirrors]
  metagetsize
  [Set mirror size to smallest sub-mirror size]
  add_key_name
  [Setup sub-mirror entry in mirror structure]
[Setup top-level mirror data in mirror structure]
meta_check_devicesize
metaioctl(MD_IOCSET)
```

Metainit – Initialise Sub-Mirrors

- Set update time for the mirror
- Walk through the sub-mirrors :
  - > Set mirror size to smallest sub-mirror's size
  - > Add sub-mirror entries to mm_unit_t struct
- Add main mirror data to mm_unit_t struct
- Check mirror size
  - > Devices > 1Tb restricted to 64-bit kernels

Metainit – Setup Mirror Code

```
meta_check_devicesize
metaioctl(MD_IOCSET)
        ddi_copyin
        mirror_set
          mirror_getun
            [Check set / device flags]
          md_getshared_key
          mddb_createrec
          mddb_getrecaddr_resize
            mddb_setenter
            [Copy record to new record, allowing for extra in-core fields]
          ddi_copyin
          [Set mirror record attributes]
          [Walk sub-mirrors]
            md_getmajor
            md_getparent
          mirror_build_incore
```

Metainit – Setup Mirror Code

```
[Walk sub-mirrors]
  md_getmajor
  md_getparent
mirror_build_incore
  mirror_are_submirrors_available
   md_getmajor
   md_getminor
  [Walk sub-mirrors]
   build_submirror
     md_getmajor
     md_getminor
     [Setup sub-mirror parameters]
       md_get_named_service
     md_set_parent
       md_getmajor
       md_getminor
  unit_setup_resync
   [Setup mirror resync parameters]
```

Metainit – Setup Mirror

- metaioctl(MD_IOCSET)
  - > Calls into md_mirror driver
- Create metadb replica entry
- Add extra fields to in-core replica
- Walk through the sub-mirrors
  - > Check they are online
  - > Set the parent device to the mirror

`

Metainit – Resync Parameters Code

```
    md_set_parent
      md_getmajor
      md_getminor
  unit_setup_resync
   [Setup mirror resync parameters]
   create_unit_resync
     md_getshared_key
     mddb_createrec
     [Setup DRL pointers in mirror structure]
     mddb_commitrec_wrapper
     mirror_commit
       md_get_setstatus
       [Walk sub-mirrors]
        md_getmajor
        md_getminor
        [Copy sub-mirror recordid into mirror's records]
       mddb_commitrecs_wrapper
  [Setup incore bitmaps for DRL etc]
  md_get_setstatus
  [Walk sub-mirrors]
    [Mark sub-mirror as needing a resync]
  [Init mutexes for mirror operations]
mirror_commit
```

Metainit – Resync Parameters

- Setup in-core resync parameters
- Setup DRL's in mddb replicas
- Setup in-core DRL pointers
- Walk through the sub-mirrors
  - > Mark each one for needing a resync

`

Metainit – Initialise Mirror Code

```
md_get_setstatus
[Walk sub-mirrors]
   [Mark sub-mirror as needing a resync]
[Init mutexes for mirror operations]
mirror_commit
md_create_unit_incore
mirror_check_failfast
[Walk sub-mirrors]
   md_get_named_service
   sm_get_component_count
 [Walk sub-mirror components]
    getmajor
    e_ddi_hold_devi_by_dev
    [Search ddi record for "ddi-failfast-supported]
[Check all sub-mirrors support failfast on all components]
[Set MD_SM_FAILFAST flag]
```

Metainit – Initialise Mirror Code

```
                    [Search ddi record for "ddi-failfast-supported]
                  [Check all sub-mirrors support failfast on all components]
                  [Set MD_SM_FAILFAST flag]
                resync_start_timeout
                  md_get_setstatus
                  timeout
              ddi_copyout
        Free
        metafreenamelist
     printf
     meta_free_mirror
meta_update_md_cf
   meta_print_all
     meta_print_trans
     meta_logs_print
     meta_mirror_print
     meta_raid_print
     meta_stripe_print
     meta_sp_print
     meta_hsp_print
md_exit
```

Metainit – Initialise Mirror

- Allocate and populate mdi_unit_t structure
- Check each component for B_FAILFAST support
  - > Search the ddi data for “ddi_failfast_supported”
  - > If all components support it, set MD_SM_FAILFAST
- Set the resync timeout value
- Cleanup and exit
  - > Free memory
  - > Display “d10: Mirror is setup” message
  - > Update md.cf file

`

SVM Mirror Init Example