

# Solaris Volume Manager : Metainport



# What is metainport?

- Allows disksets to move between hosts without the receiving host knowing about the set prior to the import
- May be used in a planned movement of disksets
- May be used as a means of disaster recovery

# Overview – regular import

- All disks in a set now have a master block, either a “normal” mb or a “dummy” mb
- If there is a “normal” mb on disk, get list of disks in set by reading the on disk mddb
- Create linked list of importable sets on the system (misp)
- Under each set, create linked list of disks in the set (midp)
- For each disk, create linked list of replicas (mirp)

# Overview – regular import (cont)

- Use information in misp/midp/mirp to create set records and drive records
- Set is taken at end of import

# Overview – replicated import

- Replicated diskset assumed if master block devid is not the same as the disk's devid
- Mostly the same code as for the regular diskset import
- Must update the devids in the namespace, locator block, and master block to the disk's real devid

# Overview – partial import

- Available for both regular and replicated disksets
- Information for the unavailable disk(s) is relative to the old configuration i.e. ctds's are “old”
- If the diskset is stale, metadbs on the unavailable disks are deleted by the import

# Overview – verbose output

- metadb output
- metastat output
- Information is read from the disk

# Master Block Structure

```
typedef struct mddb_mb {
    int          mb_magic;          /* used for verification */
    uint_t      mb_revision;       /* used for verification */
    uint_t      mb_checksum; /* used for verification */
#ifdef __LP64
    uint32_t    mb_next;           /* incore to next mb */
#else
    struct mddb_mb mb_next;      /* incore to next mb */
#endif /* __LP64 */
    daddr32_t   mb_nextblk;        /* block # for next mb */
    md_timeval32_t mb_timestamp;    /* timestamp */
    daddr32_t   mb_blkcnt;         /* size of blkmap */
    daddr32_t   mb_blkno;         /* physical loc. For this MB */
    set_t      mb_setno;          /* used for verification */
    struct timeval32 mb_setcreatetime; /* set creation timestamp */ <----- new
    int        spares[7];
    mddb_map_t mb_blkmap;         /* logical->physical blk map */
    int        mb_devid_magic;    /* verify devid in mb */ <----- new
    short      mb_devid_len;     /* length of following devid */ <----- new
    cahr       mb_devid[1];      /* devid byte array */ <----- new
} mddb_mb_t;
```



# Import Structures

```
typedef struct md_im_set_desc {
    struct md_im_set_desc    *mis_next;
    int                      mis_flags;
    int                      mis_oldsetno;
    md_im_drive_info_t      *mis_drives;
    int                      mis_active_replicas;
    int                      mis_partial;
} md_im_set_desc_t;

typedef struct md_im_drive_info {
    struct md_im_drive_info  mid_next;          /* next drive in this set */
    mddrivename_t           *mid_dnp;
    void                    *mid_devid;
    void                    *mid_o_devid;
    int                     mid_devid_sz;
    int                     mid_o_devid_sz;
    char                    mid_minor_name[MDDDB_MINOR_NAME_MAX];
    minor_t                 mid_mnum;
    int                     mid_available;
    md_timeval32_t          mid_setcreatetimestamp;
    char                    *mid_driver_name;
    char                    *mid_devname;
    md_im_replica_info_t    *mid_replicas;
    int                     overlapped_disk;
    struct md_im_drive_info  overlap;          /* chain of overlap disks */
} md_im_drive_info_t;
```

# Import Structures (cont)

```
typedef struct md_im_replica_info {
    struct md_im_replica_info *mir_next;
    int mir_status;
    int mir_flags;
    daddr32_t mir_offset;
    daddr32_t mir_length;
    md_timeval32_t mir_timestamp;
} md_im_replica_info_t;
```

# Overlapping disks

- The same disk may be found in more than one set
- Marked CONFLICT
- Disk is in conflict if `mb_set_createtimestamp` is different from others in the set or it's `ctds` is the same as that of a disk in another set

# “Good” disks

- Must be available
- Replica on it must be active
- Must have a `mb_set_createtimestamp` that is the same as the replica set creation timestamp

# Example – regular diskset

```
# metainport -r
```

Disksets eligible for import:

1) Found regular diskset containing disks:

c1t5d0

c1t8d0

c1t9d0

Creation time: Tue Dec 20 14:05:00 2005

For more information about this diskset:

```
metainport -r -v c1t5d0
```

To import this diskset:

```
metainport -s <newsetname> c1t5d0
```

```
# metainport -s blue c1t5d0
```

Importing regular diskset containing disks:

c1t5d0

c1t8d0

c1t9d0

Creation time: Tue Dec 20 14:05:00 2005

# Example – partial diskset

```
# metainport -r
```

Disksets eligible for import:

1) Found partial regular diskset containing disks:

c1t5d0

c1t8d0 (UNAVAIL)

c1t9d0

(UNAVAIL) WARNING: This disk is unavailable on this system.

Import may corrupt data in the diskset.

Creation time: Tue Dec 20 14:48:43 2005

For more information about this diskset:

```
metainport -r -v c1t5d0
```

To import this diskset:

```
metainport -f -s <newsetname> c1t5d0
```

# Example – verbose output

```
# metainport -rv
```

Disksets eligible for import:

1) Found regular diskset containing disks:

c1t5d0

c1t8d0

c1t9d0

Metadatabase information:

flags		first blk	block count	
a	u	16	8192	/dev/dsk/c1t5d0s7
a	u	16	8192	/dev/dsk/c1t8d0s7
a	u	16	8192	/dev/dsk/c1t9d0s7

Metadevice information:

d10	m	2.0GB	d1	d2	d3
d1	s	2.0GB	c1t5d0s0		
d2	s	2.0GB	c1t8d0s0		
d3	s	2.0GB	c1t9d0s0		

Creation time: Tue Dec 20 14:05:00 2005

Last modified time: Tue Dec 20 14:05:17 2005

To import this diskset:

```
metainport -s <newsetname> c1t5d0
```

# Example – disks in conflict

Disksets eligible for import:

1) Found regular diskset containing disks:

c1t5d0

c1t8d0 (CONFLICT)

c1t9d0

(CONFLICT) WARNING: This disk has been reused in another diskset.

Import may corrupt data in the diskset.

Creation time: Tue Dec 20 14:05:00 2005

For more information about this diskset:

```
metaimport -r -v c1t5d0
```

To import this diskset:

```
metaimport -f -s <newsetname> c1t5d0
```

2) Found regular diskset containing disks:

c1t8d0 (CONFLICT)

c1t10d0

c1t11d0

(CONFLICT) WARNING: This disk has been reused in another diskset.

Import may corrupt data in the diskset.

Creation time: Tue Dec 20 14:18:41 2005

For more information about this diskset:

```
metaimport -r -v c1t8d0
```

To import this diskset:

```
metaimport -f -s <newsetname> c1t8d0
```

Number of disksets eligible for import: 2

Warning: The following disks have been detected in more than one set.

Import recommendation based upon set creation time.

Proceed with the import with caution.

c1t8d0 - must import with set created at : Tue Dec 20 14:18:41 2005





# Solaris Volume Manager : Metainport