



MULTIPATH INTERVALS

CHILD 1

- ↳ INIT CHECKERS, PRIOR
- ↳ PID, SIGNAL, SCHED
- ↳ STARTS EVENT LOOP

CONFIGURE

↳ ALLOC PATHVEC, MPVEC

PATH DISCOVERY

ON CURRENT PATHVEC
KEEP OLD PATHS

↳ AND 1 EMPTY MPVEC FOR
VALUES MAPS
AND ALL FUNCTIONS USING
MPVEC

- UDEV ENUM "block"

- PATH_DISCOVER() UDEVICE

↳ STORE_PATH_INFO() → PATHVEC
PATH_INFO

TYPE=DISK

DISCOVER PATHS

DISCOVER EXISTING MULTIPATHS

ORGANIZE PATH GROUPS

↳ REMOVE BLACKL PATHS

MAP-DISCOVERY

POPULATES
NEW MPVEC

DM-GET-MAPS

FOREACH MPP

DM_DEVICE_LIST

IF: DM_TYPE = TGT_PATH

PLACE MPP IN MPVEC NEW

SETUP_MULTIPATH ★

↳ CHECK INFO

↳ CHECK DM_MAP → DM "TABLE"

UPDATE_MULTIPATH_STRINGS

- RR_WEIGHT, PG FAIL BACK
- NO PATH RETRY, ...

- FREE MP-OPG

① UPDT MP TABLE

② UPDT MP STATUS

CHECK EXISTING
MPATH TABLE
TO CONSTRUCT
PGS

① DM-GET-MAP

DISASSEMBLE_MAP

- HW HANDLER

- PATH GROUPS

- 1ST PG TO TRY

- PATHS

- SELECTOR

CREATE PGS
IN MPP

② DM-GET STATUS

DISASSEMBLE-STATUS

FEATURES, PGS

PG STATUS, FAIL CNT

- SELECTOR

COALESCE - ~~PATHS~~

GLUE PATHS INTO MULTIPATHS

ONLY THE QUES THAT ARE NOT COALESCED BY

MAP_DISCOVERY

CREATE MAPS AND PUSH CHANGED QUES INTO DM

FOR EACH PATH
DISCARD
CONCESSED
DISAPPEARED
BLACKLIST...

PATHVEC

SHOULD-MULTIPATH?

YES

NO

FOR EACH PATH
CHECK FOR SAME WWID → YES
CHECK CONF FOR WWID
→ SHOULD MP IF THERE

ORPHAN PATH

UPDATE_MPP_PATHS:

FOR EACH PG IN MPP:

FOR EACH PATH IN PG:

- IF NOT IN MPP → PATHS
- IF EXISTS IN PATHVEC

STORE IN MPP → PATHS

MPP

CREATED

FOR EACH PATH

CHECK WWID
CHECK SIZE

ACTION REJECT IF DIFF SIZE

FOR EACH PATH IN PATHVEC:

- IF PATH WWID = MPP.WWID → DOWNER

- ADD PATH IN MPP → PATHS IF NOT THERE

- PATH INFO | - CREATE FD

- GET PATH STATE (SYNFS)

VERIFY_PATHS()

CHECK IN SYNFS

FOR EACH PATH:

- SYNFS-GET-DEV

STATE? PATH-DOWN?

SETUP_MAP()

SETS MPP OPTIONS BASED ON CONF

- MPP FEATURES, SELECTOR, HWHANDLER CREATED

- PG FAILBACK, PG POLICY, SELECTOR, FEATURES, HWHANDLER, RR-WEIGHT, MIN-IO, NO PATH REPLY

- SYNFS SET TMO

EXAMPLE, MPP GIVEN

GROUP-BY-PRIO:

- MP-DPG ALLOC VECTOR

⊗ FOR PATH INSIDE MPP-DPG

- REMOVE PATH FROM MPP-DPG
- INSERT PATH INTO PG
 - ↳ CLASSIFYING IN PRIOS BY ORDER OF INDEX
- ALLOC PG
- FREE MP-DPG

- CLEAR MPP-DPG
- CALLS POLICY TO RE-CREATE MPP-DPG
- COUNTS PATH UP (USING MPP-DPG)
- SELECTS BEST PG
- ASSEMBLE_MAP

RECREATES PGS AND DM PARAM

↳ TRANSFORMS (MP-DPG)

VECTORS OF PATH
VECTORS INTO

MP-DPG PARAMS

↳ TO FEED DM
FOR MULTIPATH DEV

↳ ACTIONS IN MPP: PREPARE ACTIONS FOR DOMAP
↳ CHECK IF RELOAD IS NEEDED, WWID CHANGED, ETC

↳ DOMAP W/ PARAMS

↳ ACTIONS:

↳ QUEUE-IF-NO-PATH

↳ DM-SET-PG-TIME/OT

↳ ADD MPP INTO NEW

END

⊗ 1 MPP

ACT REJECT

ACT NOTHING

ACT SWITCH PG

ACT CREATE

ACT RELOAD

ACT RESIZE

ACT RE-MAP

DEVICE MAPPER CMDS

STOP_WAITER_THREAD
ORPHAN_PATHS
DEL FROM MPP

FOREACH MPP IN MPVECS:

REMOVE "DEAD MAP"

↳ MPP → PG → PATHS → DEV

↳ REMOVE_MAP()

↳ DM_FLUSH_MAP() STRLEN <= 0

END OF COALESCE MAPS PATHS

NEW MPVEC

CLEANUP AT THE END

MAPS WITH DEAD PATHS

COALESCE_MAPS

RECREATES
MPVEC
FROM "VECS"
GIVEN TO

CONFIGURE

REMOVES MPP
FROM VECs

REMOVE_MAPS(VECS)

↳ ~~REMOVE_MAPS~~

VECS → MPVEC CLEARED

VECS → MPVEC = NULL

OLD

FOREACH ~~OLDMPATH~~ IN OLD MPVEC:

- TRY FIND OLDMPATH IN NEW MPVEC

↳ ~~NO~~ NO??

- REMOVE MAPS NOT ALLOWED BY CURRENT CONFIG

DM_FLUSH_MAP(OLDMPATH)

↳ ERROR? DEVICE IS OPENED

↳ PUT OLDMPATH INTO NEW MPVEC

↳ RUN SETUP_MULTIPATH ★

↳ DEL OLDMPATH FROM OLD MPVEC

↳ OK? DEVMAP REMOVED

MAY NEED TO REMOVE
E.G. CHANGES
BLACKLIST IN

VECS → MPVEC = MPVEC NEWLY CREATED

→ FOR EACH MPP IN MVECS → DMPVEC :

END OF CONFIGURE

- SETUP_MULTIPATH() ★ → RECREATES PG IN MPP

- START_WAITER_THREAD()

↳ FOR EVERY MPP IN THE NEW MVEC.

- ALLOC WAITER (EVENT THREAD)

- THREAD = WAIT_EVENT()

- MPP → WAITER = THREAD

DM-TASK
THREAD
EVENT-NR
MAP NAME
VECS

→ MPP → ALIAS

→ VECs = VEC

WAITER

EVERY MULTIPATH HAS ONE THREAD

LOCKS MVEC

→ WAIT_EVENT

↳ WAIT_EVENT LOOP

↳ DM_TASK_CREATE (DM_DEVICE_WAIT_EVENT) = DMT

DM-TASK-SET-NAME

DM-TASK-SET_EVENT-NR

DM-TASK-RUN() BLOCKS

PER MPPATH

UPDATE_MULTIPATH :

- FREE PGVEC (MPP → PG)

- SETUP_MULTIPATH ★ PG

POSSIBLE EVENTS :

- TABLE RELOAD [MPP IS OBSOLETE CALL UPDATE_MULTIPATH]

- PATH FAILED BY DM [MARK IT USING UPDATE_MULTIPATH]

- MAP GONE AWAY [STOP THREAD]

- PATH REINSTATE OR SWITCH GROUP DO [NOTHING]

VECTOR

ALLOCATED
** PTR (slot)

HWTABLE VECTORS

HWE

HWE

HWE

...

EXAMPLE ↑

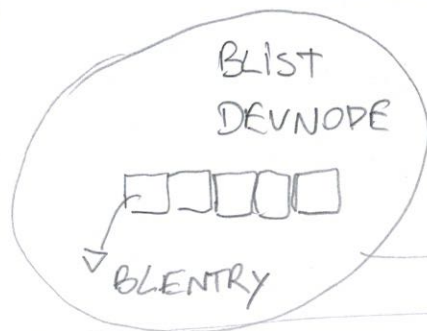
DEVICES

HW ENTRY

- VENDOR
- PRODUCT
- GETUID
- FEATURES
- HW HANDLER

DEFAULTS

BLACKLIST: VECTORS (+ BL EXCEPTIONS)



BLIST
WWID



BLIST
DEVICE



DEFAULT BLACKLIST

- RAM, LOOP, FD, *
- HDA-2
- CSSBLK ...
- CCSS ...
- NUME

MULTI PATHS: VECTOR

CONF → MPTABLE



MPENTRY:

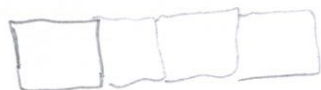
- WWID
- ALIAS
- GETUID
- SELECTOR
- PG POLICY
- PG FAIL BACK
- RR WEIGHT
- ...

VECTORS

- MPVEC

- PATH VEC

- LOCK



STRUCT PATH

COPY #1

- DEV
- DEV-T
- STRUCT UDEV_DEV
- STRUCT SCSI_IDLUN
- STRUCT SG-ID

- WWID
- VENDOR-ID
- ...

- STRUCT PRIO
- STRUCT CHECKER
- STRUCT MULTIPATH
- STRUCT HWENTRY

if kernel = "dm"

DEVICE MAPS

CHANGE, REMOVE, UDMOUNT

PATHS

ADD, REMOVE, CHANGE

- WWID
- ALIAS_OLD
- PG POLICY, PG POLICYFN()
- NEXT PG
- BEST PG
- QUEUEDIO
- ACTION
- PG FAILBACK
- ...
- RR-WEIGHT
- NR-ACTIVE
- NO-PATH-RETRY
- PG-TIMEOUT

- { UDEV
- ADD_MAP()
- REMOVE_MAP()
- UDMOUNT_MAP()

UEVENT TRIGGER

STRUCT PATH

STRUCT PATHGRP

- VECTOR PATHS

- VECTOR PG

- ALIAS
- SELECTOR
- HWHANDLER

STRUCT HWENTRY

STRUCT MPENTRY

CONF-DMPTABLE

CONF FILE LUN SETTINGS

UEVENT

- NODE ->
- UDEV-DEVICE * UDEV

- BUFFER

- DEV PATH

- ACTION

- KERNEL

- ENVP

STRINGS

STRUCT MULTIPATH

3

MPVEC



STRUCT MULTIPATH
STRUCT MULTIPATH
STRUCT MULTIPATH

- wwid
- ...
- VECTOR **PG** (PATH GROUP)
- SELECTOR
- HWE

↳ FROM HWTABLE VECTOR

- VECTOR **PATHS**

COPY #2



UPDATE_MPP_PATHS()

STRUCT **PATHGROUP**

- id
- STATUS
- PRIORITY
- ENABLED PATHS
- SELECTOR
- VECTOR **PATHS**

DMSETUP TABLE
"ROUND ROBIN" → ARGS
SELECTOR



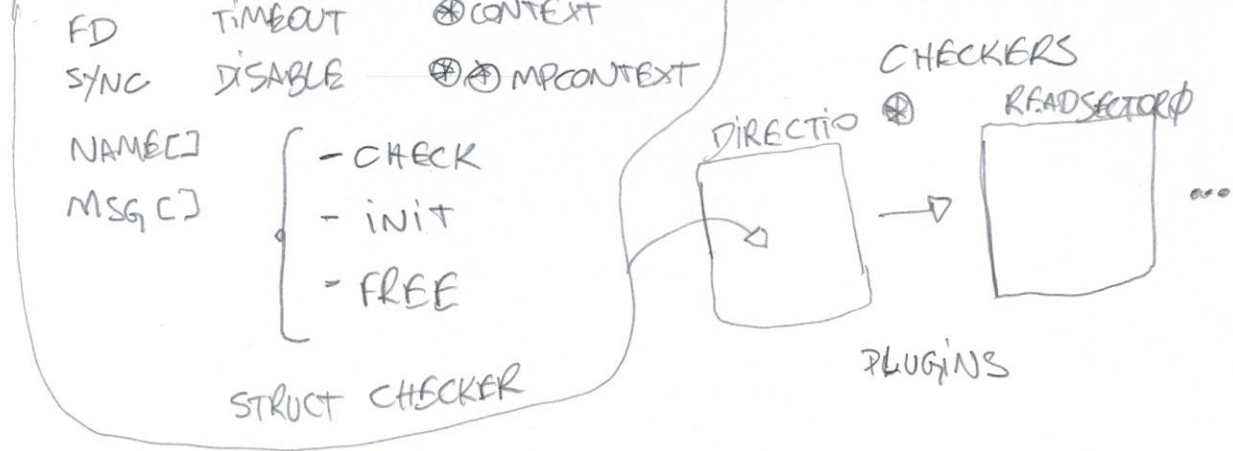
PATHS 1ST STORED
IN PATHVEC AND
THEN ADDED HERE
ALSO

STRUCT **PATH**

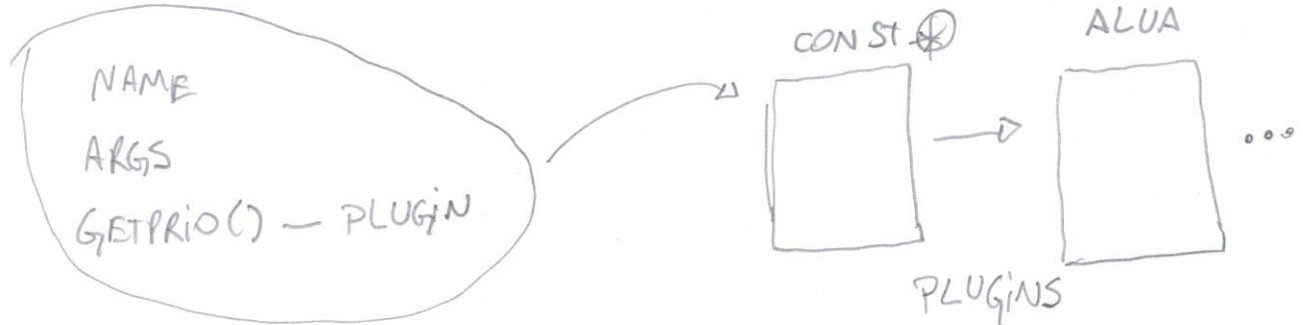
- DEV
- DEV.T
- UDEV_DEVICE
- SCSI_ID
- WWID
- ...
- STRUCT PRIO
- STRUCT CHECKER
- STRUCT **MULTIPATH** MPP
- INT FD
- STRUCT HWENTRY

COPY #3

- ADD-CHECKER



- ADD-PRIO



UEVQ LOOP

↳ UEVENT DISPATCH

FOR EACH UEVENT:

- UEV_TRIGGER (UEVS)

↓
SERVICE THE
UEVENT QUEUE

DM (ADD, REMOVE, CHANGE)

↓
PATH ADD, REMOVE, CHANGE

↳ UEV_ADD_PATH()
↳ UEV_REMOVE_PATH()
↳ UEV_UPDATE_PATH()

UEV_ADD_MAP() → EV_ADD_MAP()

↳ UEV_REMOVE_MAP()
↳ UEV_UMOUNT_MAP()

ADD_MAP_W/O_PATH()

↓
START_WAITER_THREAD()

↳ WAITS FOR A DM
EVENT ON THE
CREATED MPP

{
- TABLE RELOAD
- PATH FAILED
- MAP GONE AWAY
...
}

EVENTS

↳ WAITS PTHREAD COND
UEV-COND P

↳ THREAD: UEVENT LOOP (5)

↳ CREATES MPP
STRUCT

CHECKER LOOP ()

FOR EACH PATH IN PATHVEC:
 PATH → CHECKINT = CONF → CHECKINT

WHILE (1):

 LOCK VECs

 "tick"

 FOR EACH PATH IN PATHVEC:

 CHECK-PATH()

 DEFERRED-FAILBACK-TICK()

 FOR EACH MULTIPATH IN MPVEC:

 MPP → FAILBACK-TICK--

 NO ^{FB}TICK AND NEED-SWITCH-PG()?

 SWITCHPATHGROUP()

 RETRY-COUNT-TICK()

 DISABLES QUEUE IF NO PATH BASED ON "RETRY-TICK"
 PER MPP

GARBAGE COLLECTOR ON MPVEC()

UNLOCK VECs

CHECK-PATH()

NEW STATE = PATH-OFFLINE()
CHECKS SYSFS

IF PATH-UP:

NEW STATE = GET-STATE()
PP → CHECKER?

 SELECT CHECKER()

 CHECKER SET FD()

 CHECKER WITH()

 STATE = CHECKER-CHECK()

 RETURN STATE

STATE CHANGED

IF NEW STATE ≠ PP-STATE

 OLD STATE = PP-STATE ?

 IF NEW STATE = PATH-DOWN

 UPDATE-MULTIPATH-STRINGS
 USING PP-MPP AND PATHVE

 IF OLD STATE = PATH-UP

 FAIL-PATH(PP) ?

 IF OLD STATE NOT PATH-UP!

 REINSTATE-PATH(MPP)

 CHECK FAILBACK AND
 SWITCH PATH GROUP IF NEEDED

 ENABLE(PG) IF NEW STATE = UP

GARBAGE COLLECTOR:

E.G. MPP DEVICE DISAPPEARED

FOR EACH MPP IN MPVEC:

└ if IM_MAP_PRESENT FAILS:

└ REMOVE MAP AND STOP WAITER()

└ THREAD_CANCEL(MPP_THREAD)

└ ORPHAN_PATHS(MPP)

└ FOR ALL PATHS, PP → MPP = NULL

└ DEL MPP FROM MPVEC

└ FREE MULTIPATH

ELSE IF: NEWSTATE = UP

└ PP → TICK = PP → CHECKINT

ELSE IF: NEWSTATE = DOWN

└ LOGC

- PATH → STATE = NEWSTATE

- PATHINFO(PP)

- LOGIC TO SWITCH PG BASED ON PG FAILBACK, + FAILBACK - TICK OF MPP