Migration - Failure Handling

Arun Ramanathan

Agenda

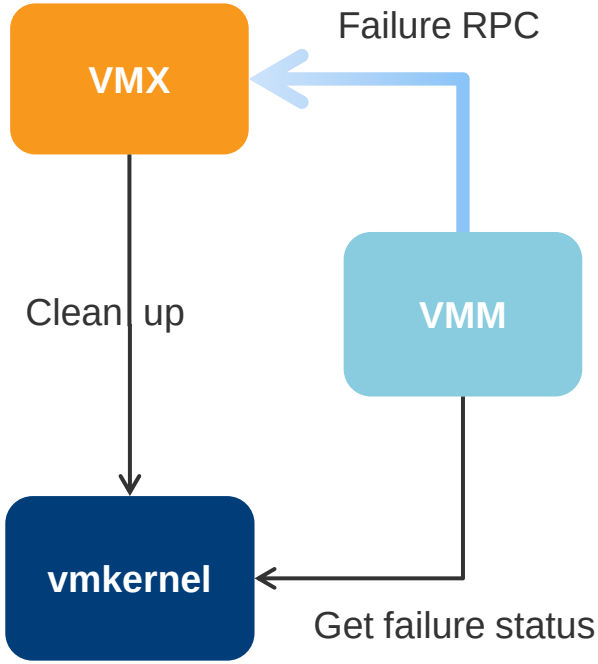
- **Failure handling overview**
- **Failure routines**
  - VMX, VMM and vmkernel routines
- **Failure detection**
  - Who calls the shot?

Failure handling

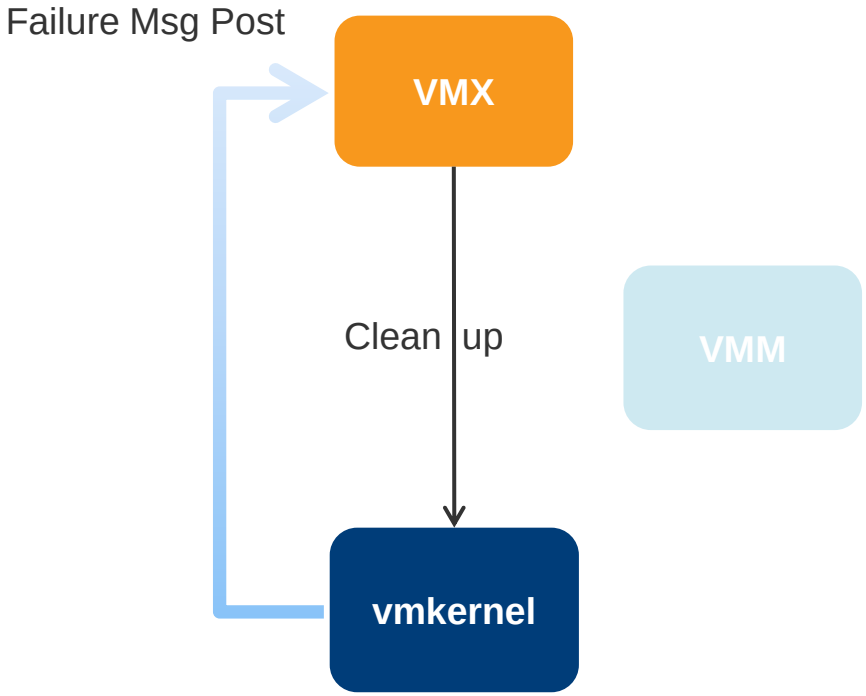
- **For a migration, VMX, VMM and vmkernel is involved.**
  - 3 address spaces
  - 3 state machines
  - 3 different cleanups!

- **Migration failure may involve cleanup at all 3 places**
  - Always involves VMX!
    - Failure handling initiated from VMX
    - After Migrate_To i.e migration request
  - VMM and vmkernel may be involved
    - Depends on migration source/destination, phase of the migration etc
    - For example, on source
      - if migration is in precopy phase then VMM is involved.
      - If migrate start request is received then vmkernel is involved.

vmkernel vs VMM initiated failures



Precopy Failure



Resume handshake failure

VMX Migration Failure

- Migrate_VMXMigrationFailure
  - VMX informs vmkernel of a failure
  - Message
    - Migration considered a failure by the VMX. It is most likely a timeout, but check the VMX log for the true error

- Most likely timeout - Why?
  - Unfortunately, VMX uses different error codes from vmkernel
  - There isn’t any mapping between the two.
  - For simplicity assume timeout errors.
  - Print a message in vmkernel log to check the vmx log for real error.

Failure Routines

What is done on a failure?

Who does what?
(VMX, VMM, vmkernel)

Failure Routines

- **VMX**
  - Migrate_SetFailureMsgList
  - Migrate_Cleanup
  - SVMotion_Cleanup
- **vmkernel**
  - Migrate_VMXMigrationFailure
  - Migrate_VMXMigrationCleanup
    - Type bridge cleanup
      - VMotion_CleanupMigration
      - FSR_CleanupMigration
- **VMM**
  - MigratePreCopyFail

VMX - Migrate_SetFailureMsgList

- **Write to hostlog**
  - Write to hostlog on SAN.
  - Helps to maintain the VPX invariant that VMs are on one host at any give time
- **Set migration state to finished**
- **Remove timer callbacks**

• MigrateToExpiredCB	WAIT_FOR_START_TIMEOUT	90 secs
• Migrate ResumeVM	VM_RESUME_RETRY_PERIOD	120 secs
• MigrateGetProgress	SMP-FT periodic progress updates	1 sec

VMX - Migrate_SetFailureMsgList (2)

- **MIGRATE_TO**
  - Undo operations done at the start of migration
    - VMX config file – revert read only mode
    - NVRAM – fall back to old mode (from NONPERSISTENT)
  - Switch to new log file
  - Write config file to disk
    - Any changes during migration are lost! – older ESX versions?

- **Set failure code**
  - MigrateEncodeErrCode – migret, err

- **Notify VMDB with failure info**
  - MigrateNotifyVmdbLast – type, uuid, failureCode, msgList, timeStamp

VMX - Migrate_SetFailureMsgList (3)

- **Inform Vigor on failure**

- **Inform the platform**
  - Allow vmkernel migrate to deal with failure
  - Migrate_VMXMigrationFailure - set kernel state to failure, collect VOBs

- **Migrate fire event MIGRATE_EVENT_SET_FAILURE**
  - Inform FT and vFlash?

- **Migrate_Cleanup**
  - On Dest – Only if dest hasn't open files
  - On Source – if not checkpointing else SVMotion_Cleanup

VMX - Migrate_Cleanup

- **If checkpointing**
  - Restore original dumper
- **Platform Cleanup**
  - Migrate_VMXMigrationCleanup
- **Migrate RPC – Cleanup state**
- **SVMotion Cleanup**
- **Update remote UIs**

VMX - Migrate_Cleanup (2)

- **Vigor complete**
- **Migrate set state to None**
- **Free migration spec**
- **Setup migration failure code**
- **Unset migration status in VMX/VMDB**
  - Ready for a new migration

vmkernel migrate

Failure routines

vmkernel – Migrate_VMXMigrationFailure

- **Set migration state to failed**
- **Call migration specific failure function**
  - Type bridge call
    - VMotion_MigrationFailed
    - FSR_MigrationFailed
- **Collect VOBs and merge them into current context**

VMotion_MigrationFailed

- **Inform DVS on migration failure**
  - On Source
    - Activate DVS ports
    - OOB Runtime state completed
  - On Dest
    - Cleanup active or shadow ports

- **Remove resume VM timeout - 1 sec**
- **Remove RDPI transition timer - 1 ms**

Source - VMotion_MigrationFailed (2)

- **Source**
  - Calculate network bandwidth estimate and log
  - Failure during precopy
    - Vmm precopy - Post action to VMM to fail migration
    - Vmk precopy - Post a message to VMX to fail migration
    - Cleanup the source migration swap file
  - Failure during Stun
    - RDPI + Resume handshake sent
      - Post msg to VMX to fail migration and poweroff src
    - If not RDPI
      - Post msg to VMX to fail migration and resume src
      - Request swap file prefault before source resume.
    - Failure to post a message to VMX - Panic the VM

Dest side - VMotion_MigrationFailed (3)

- **Destination**
  - Calculate network bandwidth estimate and log
  - Send resume handshake failure to source
    - Power off set to false
  - RDPI + Resume handshake sent
    - Post msg to VMX to set failure and power off dest VM
    - Failure to post the message panic the VM
  - Wake up a VMM world that may be waiting in VMotion_ResumeDone
    - Waiting for changed pages to be sent

vmkernel – Migrate_VMXMigrationCleanup

- **Set migration state to failed if not already set**
- **Relay migration end event to migrate plugins**
- **Type bridge cleanup**
  - VMotion_Migration Cleanup
  - FSR_MigrationCleanup
- **Wait for Migrate Info cleanup free reference count to drop to 0**
- **Remove migrate info from the migration list**
  - The cleanup of migrate info happens when the last reference is dropped.
    - MigrateInfo_Release
    - This can be called from any of vMotion helper worlds or VMX contexts.

VMotion_CleanupMigration

- **Tell helper worlds to exit**
  - Set exit requested
  - Wakeup the helper worlds
    - Send, recv, disk (deprecated?)
    - Stream helper worlds
    - Stream completion helper

- **In case of XVMotion, on destination**
  - Clear all outstanding IOs
    - Vmotion_CloseDisks
      - Wait for outstanding IO callbacks
      - Flush the XVMotion stream
      - End the XVMotion stream

MigrateInfo_Release

- **MigrateInfoFree**
  - Type bridge free
    - VMotion_FreeMigration
  - Release reference to network stack instance
  - Release migrate log data entry
  - Free remote user messages
  - Migrate lock cleanup
  - VOB
    - Destroy saved VOB contexts
    - Destroy migrant VOB contexts
  - Free vNic backing change
  - Free migration info structure
  - Release the migrate heap

VMotion_FreeMigration

- **If its an XVMotion**
  - XVMotion_CleanupMigration
    - Remove XVMotion timer
    - Free the XVMotion slice
    - Free XVMotion structure

- **VMotion Info cleanup**
  - Cleanup the send queue
  - Close all recv sockets
  - Close the send socket
  - Free VMotion Info structure
    - VMotionFreeData
      - Precopy data, checkpoint cache, net callbacks, DVS, swap, lock cleanup etc

Virtual Machine Monitor

Failure Routine

VMM - MigratePreCopyFail

- **Inform VMX of failure**
  - Get error status from vmkernel
  - Most importantly, inform VMX to initiate failure handling
    - MigrateUpdateUserlevel

- **VMM level cleanup**
  - Kernel synchronization point
    - Revoke reference to vmkernel state – bitmap MPN
  - Clear precopy statistics
    - Traces installed, fired, pages copied etc
  - Deactivate pass through manager
  - Reenable large page allocations (disabled at start of migration)

Failure Detection

Who calls the shot?

Who calls the shot?

- **Failure can be detected by VMX, VMM or vmkernel**

- **But the failure handling is always initiated from VMX**
  - VMM or vmkernel may be first to detect failure
  - In that case, they inform VMX of the failure
    - vmkernel - post a failure message to VMX
    - VMM - RPC to VMX
  - Then VMX initiates migration cleanup by calling into platform

- **Why always VMX?**
  - Irrespective of migration phase or src/dst, VMX is available
  - One point of initiation, makes it simpler and easier to reason!

Failure detection examples

VMX

- At start on src

  Migrate_To

  MigrateToExpiredCB

- Checkpoint Failures

  on src for FT

VMM

- Precopy on src

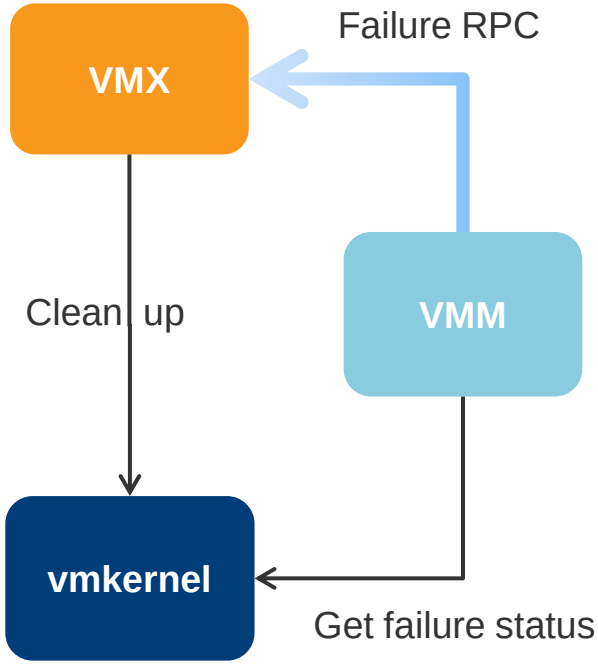
  VMM is reponsible for driving failure even if the migration failed in vmkernel

- Restore Done on Dest

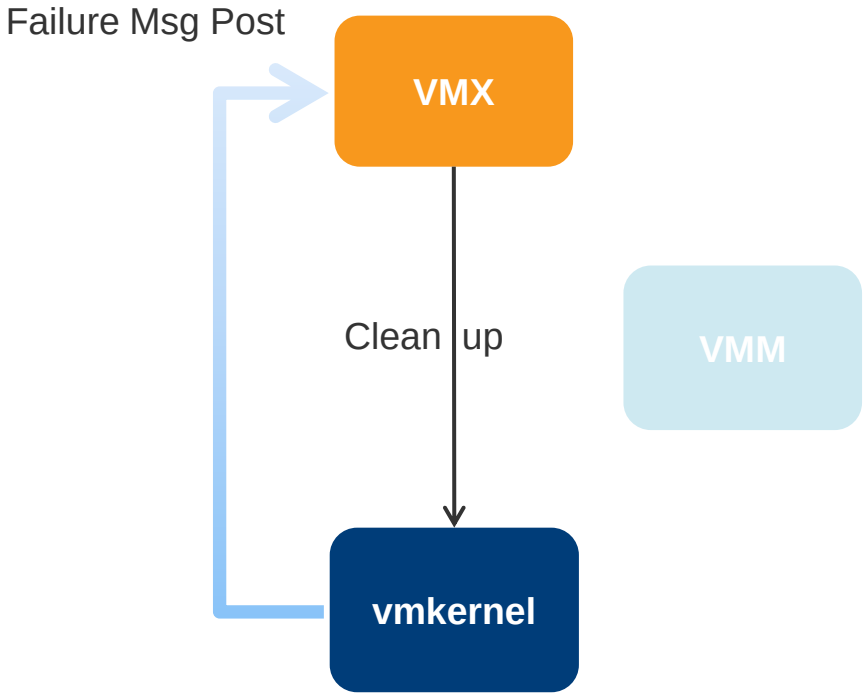
vmkernel

- Resume handshake failure on src

vmkernel vs VMM initiated failures



Precopy Failure



Resume handshake failure

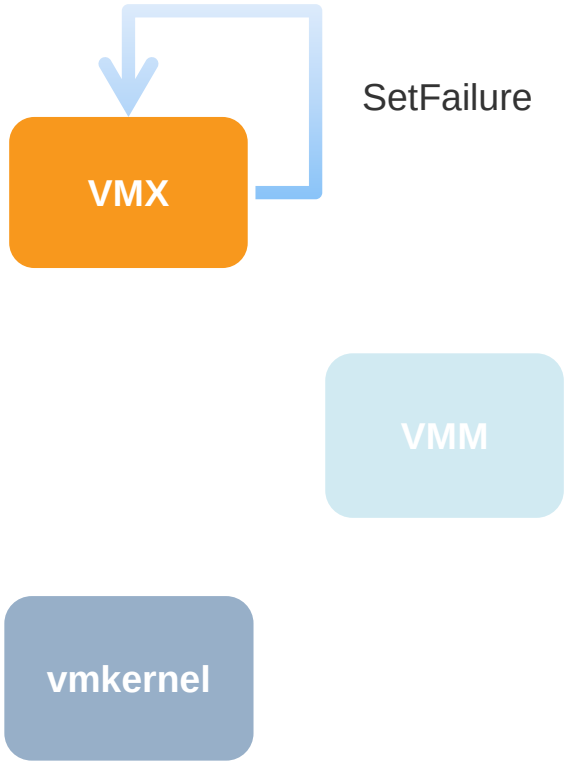
VMX – Prepare source timeout

- **Prepare source timeout - 90 sec**

  Migrate_To

  MigrateToExpiredCB

  Migrate_SetFailure



- **VMM and vmkernel are not involved**
  - Migration has not started

VMX – Checkpointing Failures

- **In case of FT migrations, checkpoint restore failures are handled in VMX level on destination**

  - CheckpointRestoreFailure

    Migrate_RestoreFailure

    Migrate_SetFailureMsgList

vmkernel – Resume Handshake failure

- **Drives failure on receiving resume handshake failure from dest**
  - MIGRATE_VMKMSG_SET_FAILURE_AND_RESUME_SRC

VMotionRecv_ResumeHandshake

MigrateState_SetFailure

MigrateTypeBridge_MigrationFailed

VMotion_MigrationFailed

VMotionSourceFailure

VMotion_PostVmxMsg

- **vmkernel detects failure and informs VMX**

VMM – Precopy failure handling

- **During precopy VMM is responsible for detecting failures**

- **Source**
  - MigrateFetchInitialBitmapMPN
    - VMKCall_MigrateMemPreCopy
    - MigratePreCopyFail
      - MigrateUpdateUserLevel => SET_PRECOPY_FAILURE_SRC
        - UserRPC

- **Destination**
  - Migrate_Sync Called on checkpoint sync
    - CPT_RESTORE_SYNC
      - VMKCall_MigrateRestoreDone
      - MigrateUpdateUserLevel => SET_FAILURE_POWEROFF_DST

Failure handling summary

- **VMX, VMM and vmkernel have separate failure routines**
- **VMX always initiates migration failure cleanup**
  - vmkernel and vmm may or may not be involved in failure handling.
  - Depends on the migration phase and location i.e src/dest
- **Following various failure routine call stacks will help determine all possible failures during migration lifecycle**