

Every migration in vmx & vmkernel is identified by a migration id.

In logs like these:

```
2014-01-31T07:18:16.037Z| vmx| I120: Received migrate 'from' request  
for mid id 1391181165680873, src ip <127.0.0.1>.
```

it is 1391181165680873.

This migration id is also present in hostd logs, from these logs, try to get the op-ID, that is used by hostd and VC for a migration.

/var/run/log/hostd.log

PrepareSource request sent:

```
2014-01-31T07:16:33.571Z [6BFC2B70 info 'Vcsvc.VMotion' opID=2dfb99e0-b-76-49b0  
user=vpxuser:root] PrepareSourceEx [1391181165680873], VM = '2'
```

```
2014-01-31T07:16:33.571Z [6BFC2B70 info 'Vcsvc.VMotionSrc (1391181165680873)'  
opID=2dfb99e0-b-76-49b0 user=vpxuser:root] VMotionEntry: migrateType = 2
```

Sending migration start request to source vmx: Notice 'to' in the log.

```
2014-01-31T07:16:33.595Z [6BFC2B70 info 'Vmsvc.vm:/vmfs/volumes/36ce7643-83741462-  
0000-000000000000/C1/C1.vmx' opID=2dfb99e0-b-76-49b0 user=vpxuser:root]  
VMotionPrepare (1391181165680873): Sending 'to' srcIp=127.0.0.1 dstIp=127.0.0.1, type=2,  
encrypted=false,  
remoteThumbprint=35:10:6B:4C:D1:39:F0:F0:7A:F0:49:08:DD:85:32:1D:BD:BC:69:A2  
SIOC: SIOC is notified not to start injector
```

PrepareSourceEx complete: Check timestamp between start and complete request to identify timeout possibilities.

```
2014-01-31T07:16:41.219Z [6C5C1B70 verbose 'Vmsvc.vm:/vmfs/volumes/36ce7643-  
83741462-0000-000000000000/C1/C1.vmx'] VMotionStatusCb [1391181165680873]:  
Succeeded
```

```
2014-01-31T07:16:41.219Z [6C5C1B70 info 'Vmsvc.vm:/vmfs/volumes/36ce7643-83741462-  
0000-000000000000/C1/C1.vmx'] VMotionStatusCb [1391181165680873] : Prepare task  
completed successfully
```

PrepareDestination request sent:

```
2014-01-31T07:16:42.706Z [FF9807D0 info 'Vcsvc.VMotion' opID=2dfb99e0-b-db-49c3  
user=vpxuser:root] PrepareDestinationEx [1391181165680873]
```

```
2014-01-31T07:16:42.706Z [FF9807D0 info 'Vcsvc.VMotionDst (1391181165680873)'  
opID=2dfb99e0-b-db-49c3 user=vpxuser:root] VMotionEntry: migrateType = 2
```

PrepareDestination completed:

```
2014-01-31T07:16:42.708Z [FF9807D0 info 'Vcsvc.VMotion' opID=2dfb99e0-b-db-49c3
```

user=vpuser:root] Completed PrepareDestinationEx [1391181165680873]

InitiateDestination request sent:

2014-01-31T07:16:43.777Z [6C7C1B70 info 'Vcsrc.VMotion' opID=2dfb99e0-b-5e-49c8 user=vpuser:root] InitiateDestination [1391181165680873], VM = '/vmfs/volumes/c3c708e0-281d081e-0000-000000000000/C1/C1.vmx'

Sending migration start request to destination vmx: Notice 'from' in the log.

2014-01-31T07:17:42.495Z [6C7C1B70 info 'Vmsvc.vm:/vmfs/volumes/c3c708e0-281d081e-0000-000000000000/C1/C1.vmx' opID=2dfb99e0-b-5e-49c8 user=vpuser:root] VMotionPrepare (1391181165680873): Sending 'from' srcIp=127.0.0.1 dstIp=127.0.0.1, type=2, encrypted=false, remoteThumbprint=35:10:6B:4C:D1:39:F0:F0:7A:F0:49:08:DD:85:32:1D:BD:BC:69:A2

SIOC: SIOC is notified not to start injector

Destination waits for dest VM WID

2014-01-31T07:17:48.345Z [6C7C1B70 info 'Vcsrc.VMotionDst (1391181165680873)' opID=2dfb99e0-b-5e-49c8 user=vpuser:root] Initiate: Waiting for WID

Failure log: message -> timedout for migration start request

2014-01-31T07:18:11.066Z [6C5C1B70 verbose 'Vmsvc.vm:/vmfs/volumes/36ce7643-83741462-0000-000000000000/C1/C1.vmx'] VMotionStatusCb [1391181165680873]: Failed with error [N3Vim5Fault20GenericVmConfigFaultE:0x6b2ccbd8]

2014-01-31T07:18:11.066Z [6C5C1B70 verbose 'Vmsvc.vm:/vmfs/volumes/36ce7643-83741462-0000-000000000000/C1/C1.vmx'] VMotionStatusCb: Firing ResolveCb

2014-01-31T07:18:11.066Z [6C5C1B70 info 'Vcsrc.VMotionSrc (1391181165680873)'] ResolveCb: VMX reports needsUnregister = false for migrateType MIGRATE\_TYPE\_FSR

2014-01-31T07:18:11.066Z [6C580B70 verbose 'Hbrsrc'] Replicator: VmReconfig ignoring VM 2 not configured for replication

2014-01-31T07:18:11.066Z [6C5C1B70 info 'Vcsrc.VMotionSrc (1391181165680873)'] ResolveCb: Failed with fault: (vim.fault.GenericVmConfigFault) {

```
-->  faultCause = (vmodl.MethodFault) null,
-->  faultMessage = (vmodl.LocalizableMessage) [
-->    (vmodl.LocalizableMessage) {
-->      key = "msg.migrate.expired",
-->      message = "Timed out waiting for migration start request.
--> ",
-->    }
-->  ],
-->  reason = "Timed out waiting for migration start request.
--> ",
```

```

--> messageInfo = (vim.vm.Message) [
-->   (vim.vm.Message) {
-->     id = "msg.migrate.expired",
-->     text = "Timed out waiting for migration start request.
--> ",
-->   }
--> ]
--> msg = "Timed out waiting for migration start request.
--> "
--> }

```

Finally got `initateDestination` returned and we got WID: notice timestamp

2014-01-31T07:19:03.866Z [6C7C1B70 verbose 'Vcsvc.VMotionDst (1391181165680873)' opID=2dfb99e0-b-5e-49c8 user=vpxuser:root] JrnlPutWid: Wrote wid=1000132085

2014-01-31T07:19:03.866Z [6C7C1B70 info 'Vcsvc.VMotionDst (1391181165680873)' opID=2dfb99e0-b-5e-49c8 user=vpxuser:root] Initiate: Got WID 1000132085

2014-01-31T07:19:03.866Z [6C7C1B70 verbose 'Vcsvc.VMotionDst (1391181165680873)' opID=2dfb99e0-b-5e-49c8 user=vpxuser:root] Migration changed state from BEGIN to MIGRATING

Workflow does not remember this migration has failed, continues from where dest returns: `initateSource`

2014-01-31T07:19:04.213Z [6C380B70 info 'Vcsvc.VMotion' opID=2dfb99e0-b-ba-4a3c user=vpxuser:root] InitiateSource [1391181165680873], WID = 1000132085

Source has already declared timeout, done cleanup and failed:

2014-01-31T07:19:04.213Z [6C380B70 warning 'Vcsvc.VMotionSrc (1391181165680873)' opID=2dfb99e0-b-ba-4a3c user=vpxuser:root] Initiate: VM state is 4. Cannot initiate

2014-01-31T07:19:04.214Z [6C380B70 info 'Default' opID=2dfb99e0-b-ba-4a3c user=vpxuser:root] AdapterServer caught exception: vim.fault.InvalidState

Destination timesout waiting for source to send data. Hence the error message.

2014-01-31T07:21:06.342Z [6C380B70 info 'Vcsvc.VMotionDst (1391181165680873)'] ResolveCb: VMX reports needsUnregister = true for migrateType MIGRATE\_TYPE\_FSR

2014-01-31T07:21:06.342Z [6C380B70 info 'Vcsvc.VMotionDst (1391181165680873)'] ResolveCb: Failed with fault: (vim.fault.GenericVmConfigFault) {

```

--> faultCause = (vmodl.MethodFault) null,
--> faultMessage = (vmodl.LocalizableMessage) [
-->   (vmodl.LocalizableMessage) {
-->     key = "msg.checkpoint.migration.noprogress",

```

```

-->     message = "Timed out waiting for migration data.
--> ",
-->   }
--> ],
--> reason = "Timed out waiting for migration data.
--> ",
--> messageInfo = (vim.vm.Message) [
-->   (vim.vm.Message) {
-->     id = "msg.checkpoint.migration.noprogress",
-->     text = "Timed out waiting for migration data.
--> ",
-->   }
--> ]
--> msg = "Timed out waiting for migration data.
--> "
--> }

```

## VMX LOGS:

### Source vmware.log

```

2014-01-31T07:16:38.673Z| vmx| I120: MigrateSetInfo: state=1
srcIp=<127.0.0.1> dstIp=<127.0.0.1> mid=1391181165680873
uuid=4c4c4544-004e-4c10-8042-c6c04f464e31 priority=low

```

```

2014-01-31T07:18:09.921Z| vmx| I120: [msg.migrate.expired] Timed out
waiting for migration start request.

```

```

2014-01-31T07:18:10.078Z| vmx| I120: Migrate: cleaning up migration
state.

```

### Destination vmware.log

```

2014-01-31T07:18:16.037Z| vmx| I120: Received migrate 'from' request
for mid id 1391181165680873, src ip <127.0.0.1>.

```

```

2014-01-31T07:21:06.320Z| vmx| I120: MigrateSetStateFinished: type=2
new state=12

```

```

2014-01-31T07:21:06.320Z| vmx| I120: MigrateSetState: Transitioning
from state 9 to 12.
2014-01-31T07:21:06.320Z| vmx| I120:
[msg.checkpoint.migration.noprogress] Timed out waiting for migration
data.
2014-01-31T07:21:06.320Z| vmx| I120: Migrate: cleaning up migration
state.
2014-01-31T07:21:06.329Z| vmx| W110: Vigor: Bootstrap command 14 went
122467821us without progress update

```

It is very important to know what phase of SvMotion did migration fail in. Grep for this message in source vmx log file:

```
1059 Log("SVMotion: Enter Phase %d\n", newPhase);
```

Something like this:

```

1286:2013-12-04T10:48:07.049Z| vmx| I120: SVMotion: Enter Phase 1
1317:2013-12-04T10:48:12.093Z| Worker#0| I120: SVMotion: Enter Phase 2
1329:2013-12-04T10:48:13.843Z| Worker#0| I120: SVMotion: Enter Phase 3
1471:2013-12-04T10:48:14.242Z| Worker#0| I120: SVMotion: Enter Phase 4
1478:2013-12-04T10:48:14.267Z| Worker#0| I120: SVMotion: Enter Phase 5
1493:2013-12-04T10:48:14.297Z| Worker#0| I120: SVMotion: Enter Phase 6
5961:2013-12-04T11:06:51.648Z| Worker#0| I120: SVMotion: Enter Phase 8
5984:2013-12-04T11:06:52.288Z| Worker#0| I120: SVMotion: Enter Phase 9
5986:2013-12-04T11:06:52.290Z| Worker#0| I120: SVMotion: Enter Phase 10
6014:2013-12-04T11:06:53.037Z| vcpu-0| I120: SVMotion: Enter Phase 11
6031:2013-12-04T11:06:53.123Z| Worker#1| I120: SVMotion: Enter Phase 0

```

The phase number is picked from the table below:

```

292 /*
293  * We declare a SVMotionPhase to help us with the cleanup
handler. Knowing
294  * the phase tells us which disks to close, delete, etc.
295  */
296 typedef enum SVMotionPhase {
297     /* Null: SVMotion not running or completed. */
298     SVMPhase_Null,
299     /* DiskSetup: Disks are allocated and added to the disks
list */
300     SVMPhase_DiskSetup,
301     /* DiskCreate: Disk create phase *Mirrored Mode only* */
302     SVMPhase_DiskCreate,
303     /* DiskLoad: Dest disks are opened if needed and
registered with kernel */
304     SVMPhase_DiskLoad,

```

```

305      /* Stun: VM is stunned to allow activation of the mirror
filter */
306      SVMPhase_Stun,
307      /* Unstun: VM is unstunned. */
308      SVMPhase_Unstun,
309      /* DiskCopy: Disk copy phase *Mirrored Mode only* */
310      SVMPhase_DiskCopy,
311      /* CopyDirtyInfo: We are stunning and we need the dirty
block info */
312      SVMPhase_CopyDirtyInfo,
313      /* CopyDirtyData: We are in the stunned state and copying
any dirty data */
314      SVMPhase_CopyDirtyData,
315      /* DiskCopyDone: Finished copying dirty blocks and
waiting for FSR. */
316      SVMPhase_DiskCopyDone,
317      /* Complete: Migration Complete */
318      SVMPhase_Complete,
319      /* Cleanup phase. *Mirrored Mode only* */
320      SVMPhase_Cleanup,
321 } SVMotionPhase;

```

Based on what phase the migration is, you will know what timeout could occur.

During diskCopy, if the disk transfer rate is low, then we can see these logs in the source vmx.

```

2013-12-10T18:41:46.281Z| vmx| W110: SVMotion: scsi0:0: Disk transfer
rate slow: 0 kB/s over the last 10.00 seconds, copied total 0 MB at 0
kB/s.

```

```

2013-12-10T18:41:56.291Z| vmx| W110: SVMotion: scsi0:0: Disk transfer
rate slow: 0 kB/s over the last 10.01 seconds, copied total 0 MB at 0
kB/s.

```

....

```

2013-12-10T18:46:06.446Z| vmx| W110: SVMotion: scsi0:0: Disk transfer
rate slow: 6547 kB/s over the last 10.01 seconds, copied total 320 MB
at 1212 kB/s.

```

This log is printed every 10 seconds, if you see it consecutively for 120 seconds (11 messages with timestamp 10 seconds apart), it will be followed by 120 seconds migration timeout failure.

Something like this:

```

2013-12-10T18:48:06.470Z| Worker#1| I120: Migrate: Remote Log:
Destination waited for 496.04 seconds.

```

This message captures the time since destination has started accepting data, it does not mean how long it waited for this

particular data. For instance, there could be a 120 seconds timeout and this log could said destination waited for 496 seconds. That does not mean destination waited for 496 seconds before timing out. This log is always seen, even in case of success.

The slow transfer rate is usually paired with these kernel logs that complain if each IO takes longer than 10 seconds. The logs are something like these:

```
2013-12-10T10:39:30.472Z cpu14:1000046508)WARNING: XVMotion:
AsyncIOWriteDone:4793: 1386700947274651 D: 65536 byte IO (1 element)
to disk 0 took 19970ms to complete (0ms blocked, 0 retries).
2013-12-10T10:39:30.561Z cpu14:1000014579)WARNING: XVMotion:
AsyncIOWriteDone:4793: 1386700947274651 D: 65536 byte IO (1 element)
to disk 0 took 19785ms to complete (0ms blocked, 0 retries).
2013-12-10T10:39:30.742Z cpu14:1000035113)WARNING: XVMotion:
AsyncIOWriteDone:4793: 1386700947274651 D: 65536 byte IO (1 element)
to disk 0 took 18704ms to complete (0ms blocked, 0 retries).
```

This still does not mean guaranteed failure, but it's reporting unusual slowness

You could see messages like these in destination kernel log:

```
2012-04-20T10:47:35.128Z cpu5:58469)XVMotion:
HandleConflictingIO:4653: 1334918329520774 D: Found real conflict
between 6985759744, 6985760256 and 6985756672, 6985760768
2012-04-20T10:47:44.312Z cpu0:58469)XVMotion:
HandleConflictingIO:4653: 1334918329520774 D: Found real conflict
between 3155169280, 3155173376 and 3155169280, 3155173376
2012-04-20T10:47:44.434Z cpu0:58469)XVMotion:
HandleConflictingIO:4653: 1334918329520774 D: Found real conflict
between 3155165184, 3155169280 and 3155165184, 3155169280
2012-04-20T10:47:44.467Z cpu0:58469)XVMotion:
HandleConflictingIO:4653: 1334918329520774 D: Found real conflict
between 3155169280, 3155173376 and 3155169280, 3155173376
```

They are benign messages and do not indicate any slowness or unusual activity by themselves. Paired with other (already mentioned above) logs, it could mean something. Note that they occur in bulk, so it's highly possible to blame them for anything wrong that might be happening.

In the kernel logs, the messages are appended with a "D" or "S" along with the migration ID to indicate **D**estination or **S**ource. This is important for storage vMotion when the kernel logs are in the same

logfile.

Timeouts:

- 90 seconds: Between prepareSource and initiateSource
- 120 seconds: Wait for data on destination. Starts after 4 step setup is over
- IO timeout: [XVM\\_IO\\_TIMEOUT](#) = VMK\_SCSI\_CMD\_TIMEOUT \* 3 = 120 seconds
- 20 seconds: Failed to allocate buffer for disk transfer in XvMotion. Failed to flush data into network.
- 100 seconds: Switchover timeout. Starts with source stun, ends at destination resume.