

Live Container Migration without Dropping TCP Connections

Jiayi Liu, Tommy Oliver, Ben Reeves

Container migration



Image source: <https://www.infoq.com/articles/container-live-migration>

Container migration

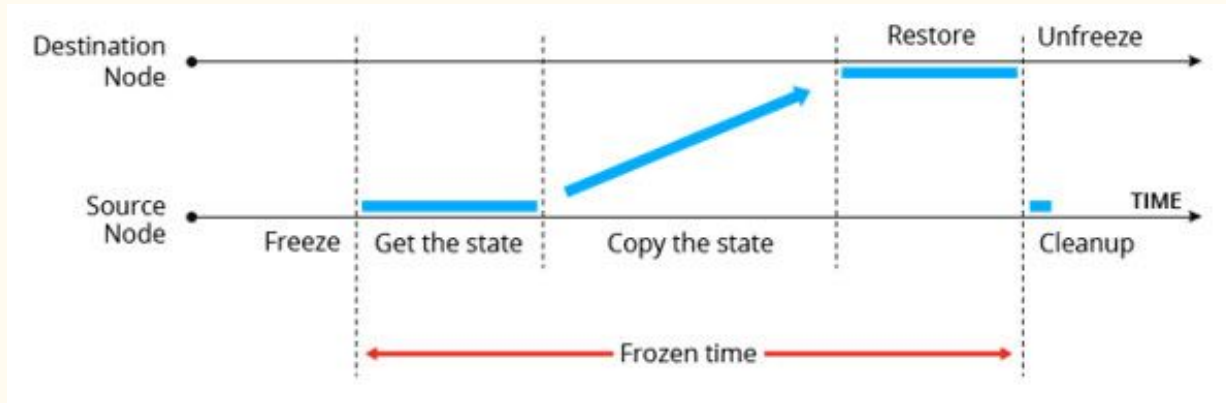


Image source: <https://www.infoq.com/articles/container-live-migration>

Container migration applications

- Server maintenance without downtime
- Seamless kernel upgrade
- Compute load balancing
- Change cloud vendor

Existing solution: CRIU

CRIU = Checkpoint/Restore In Userspace¹

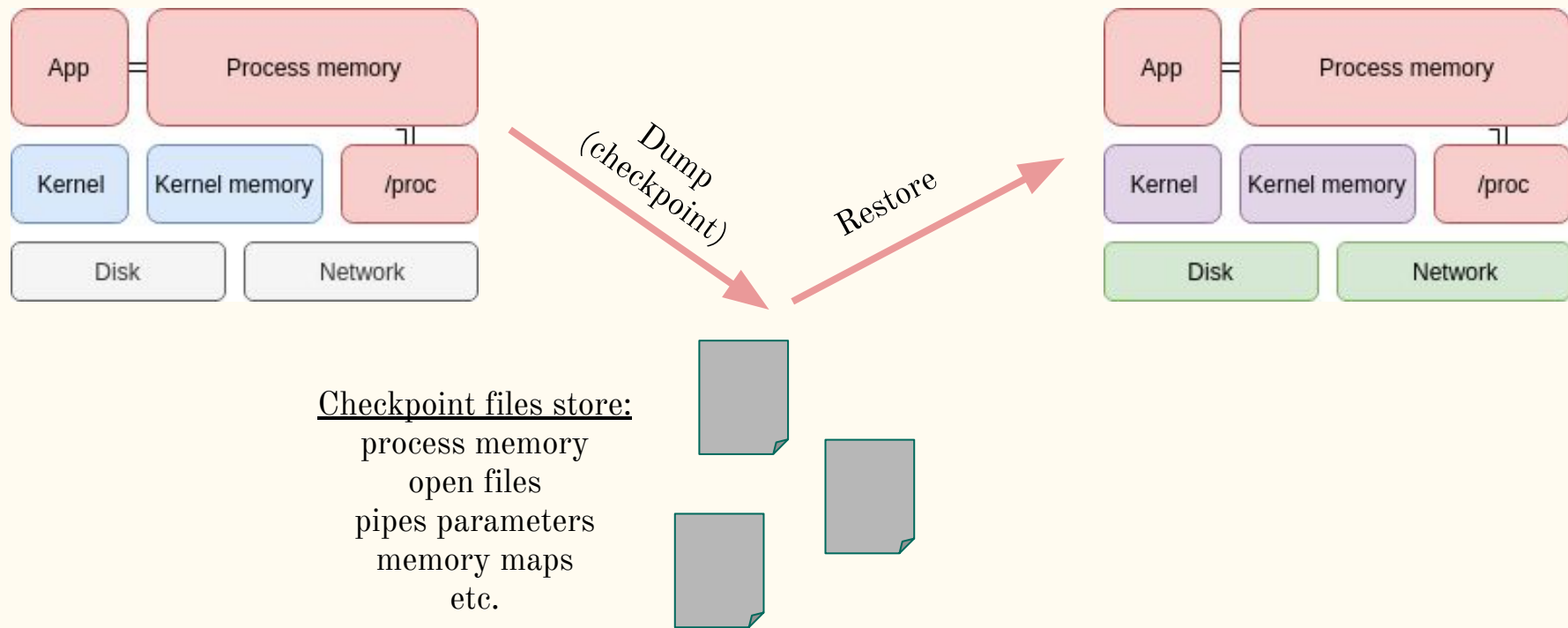
Provides generalized capabilities for checkpoint and restore.

Potential other uses:

- Snapshots
- Remote debugging
- Slow-boot services speed up



Checkpoint and restore

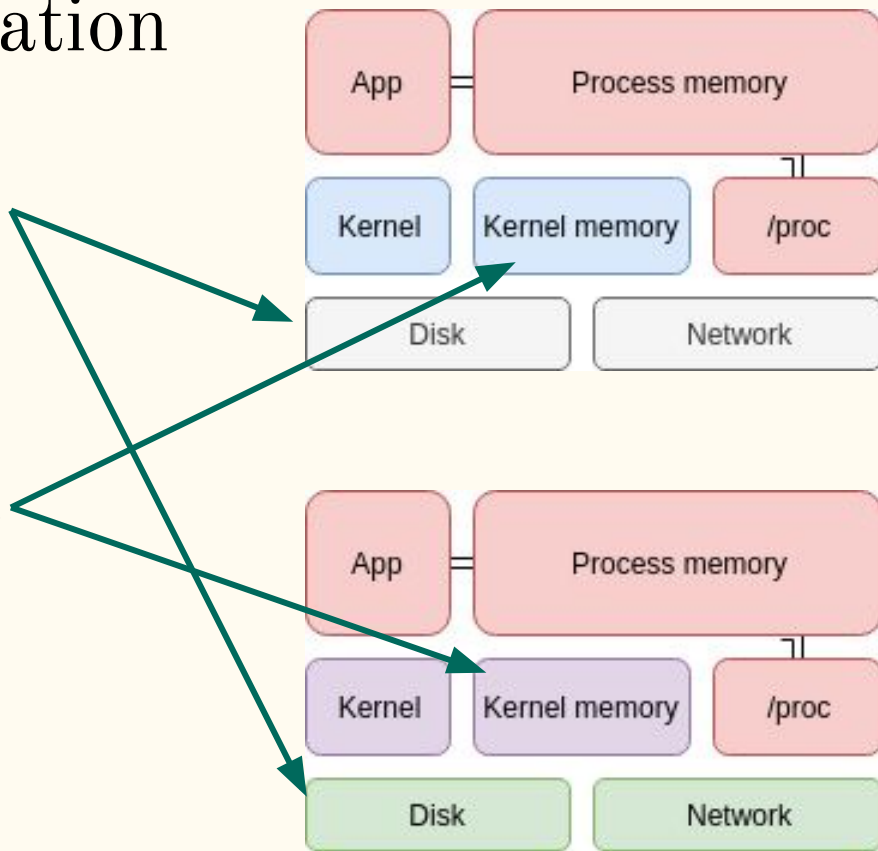


Limitations of live migration

Destination node may need to have the same file system.

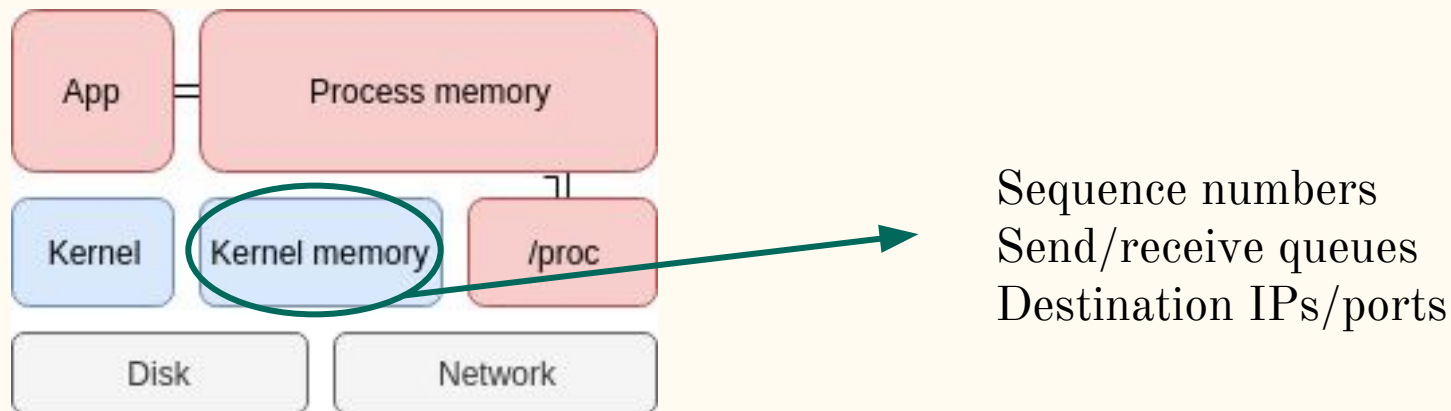
- Even when you think it shouldn't (hidden test.log)

Active TCP connections will be broken.



How to migrate TCP connections?

Theoretically you could make a kernel module to grab all relevant data out of the kernel and migrate it.



Actually CRIU can already do this!

The CRIU devs introduced **TCP repair mode** to the Linux kernel.

Put a socket into repair mode, then you can read and/or set:

- Sequence numbers
- Congestion window size
- Send and receive queues

Also, while in repair mode, you can do things without causing network I/O:

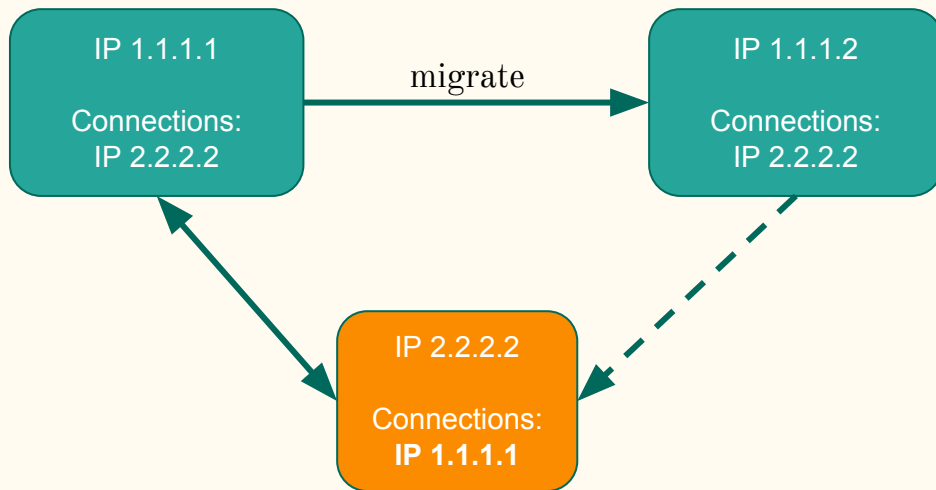
- `bind()`
- `connect()`

Still one issue: what about the client?

This is the core of our project.

We must figure out how to get the client to transfer their connection over.

In the end, all we want to do is demonstrate a seamless container migration from one physical machine to another.

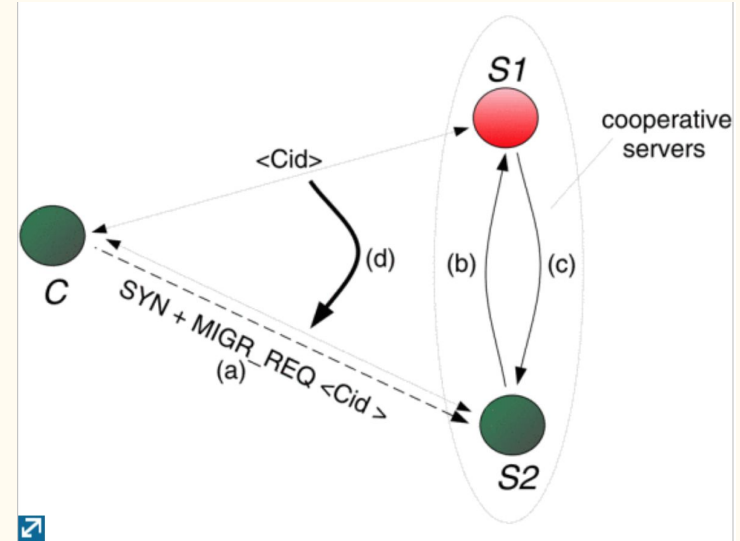


Past research: M-TCP

"Migratory TCP: connection migration for service continuity in the Internet"⁴

(Sultan et al., 2002)

- Client initiate the migration from server S1 to S2 once service delivery is interrupted.
- After receive migration certificate S2 sends request to S1 to get state.



Past research: MIGSOCK

"MIGSOCK: Migratable TCP Socket in Linux"²

(Kuntz, Rajan, 2002)

- Master thesis from CMU.
- It only works for sockets in the ESTABLISHED state.
- Puts client process to sleep during transfer.

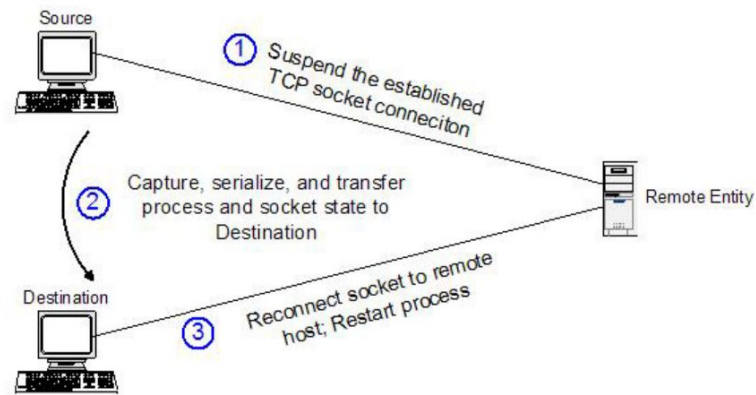


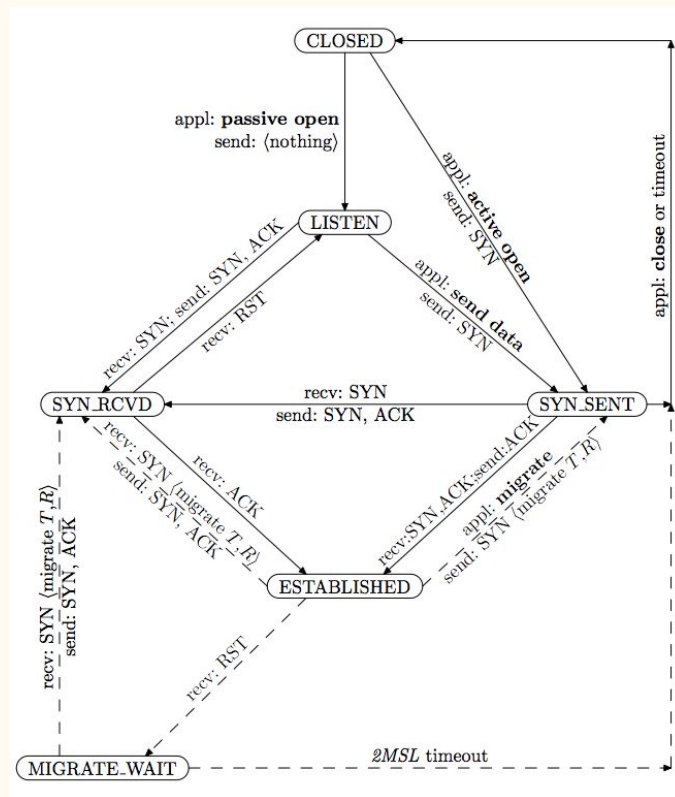
Figure 3.2: Socket Migration

Past research: TCP Migrate

"An End-to-End Approach to Host Mobility"³

(Snoeren, Balakrishnan, 2000)

- Introduces a TCP Migrate option directly into the kernel TCP stack.
- New MIGRATE_WAIT state for client.
- Inter-operable with machines that do not support TCP Migrate option.
- Made the kernel changes but we don't have their code :'(



Our approach

Our ideal approach:

Try to do what TCP Migrate did and add MIGRATE_WAIT state to client TCP.

Our initial approach:

Do everything in userspace with the help of a client daemon.

Client daemon idea

Any client process that would like to be "migratable" registers itself with the daemon.

When server migrates its application, it sends a message to its client(s) daemon processes.

The daemon process re-configures the open socket(s) on the client application to point to the new IP.

This should be possible in userspace. If CRIU can do it, it must be possible.

Next step: containers

First, we'll just get it working for regular processes.

Then we'll try containers.

Maybe try to integrate it with Docker and/or Kubernetes.

Progress

Checked how CRIU works uses TCP repair mode to recreate sockets.

Evaluated CRIU with processes that have open TCP sockets.

- Uses iptables to drop packets instead of reset.
- Fails when the system is restarted.
- Github issues about inconsistent IPs when moved from one host to another
 - As long as we update the client's socket, we believe (hope) this is not an issue.
- Learned extent of client shrinking CWND.

References

- [1] Pavel Emelyanov. CRIU: Checkpoint/Restore in Userspace, 2011. <https://criu.org/>
- [2] Kuntz, B. (2002). *MIGSOCK: Migratable TCP socket in Linux* (Doctoral dissertation, Carnegie Mellon University. Information Networking Institute).
- [3] Snoeren, A. C., & Balakrishnan, H. (2000, August). An end-to-end approach to host mobility. In *Proceedings of the 6th annual international conference on Mobile computing and networking* (pp. 155-166). ACM.
- [4] Sultan, F., Srinivasan, K., Iyer, D., & Iftode, L. (2002). Migratory TCP: Connection migration for service continuity in the Internet. In *Proceedings 22nd International Conference on Distributed Computing Systems* (pp. 469-470). IEEE.