

Container Migration using CRIU and LXC

Optimizing Migration Downtime

Adrian Reber <areber@redhat.com>

June 11, 2016

Docker!

Docker!

Docker!

Containers

- ▶ Advantages
 - ▶ Almost no overhead
 - ▶ Fast startup times
 - ▶ Portability
 - ▶ Lightweight separation
- ▶ Limitations
 - ▶ Operating system depends on the host
 - ▶ Still limited management options

CRIU

- ▶ Checkpoint/Restore In Userspace
- ▶ Developed by OpenVZ/Virtuozzo/Parallels/Odin
- ▶ Required features merged in upstream Linux kernel
- ▶ Provides transparent checkpoint and restore
- ▶ Available in many Linux distributions

Process Migration

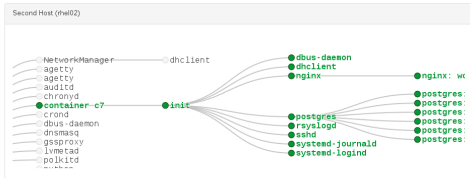
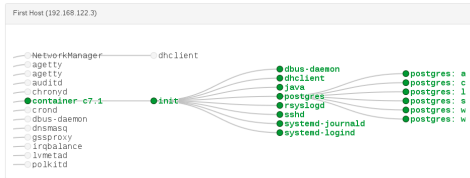
- ▶ Checkpoint Process
- ▶ Transfer Checkpoint
- ▶ Restore Process

Container Migration

- ▶ Checkpoint Container
- ▶ Transfer Checkpoint
- ▶ Restore Container
- ▶ Container \Rightarrow Process Tree

CRIU Process Migration GUI

Process Trees



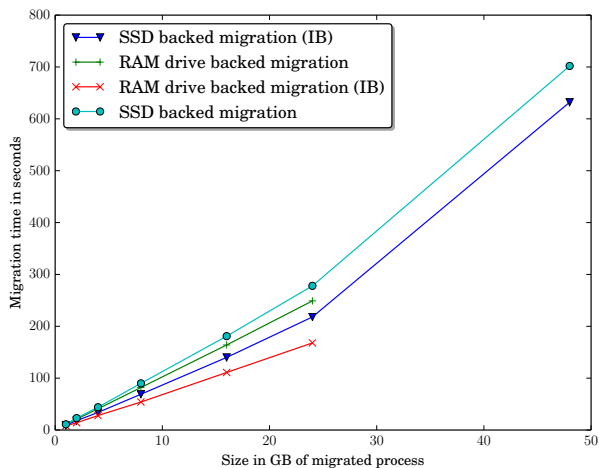
Information

Process Info

Hover over a tree node to see more information about that process.

Name	ID	Children
polkitd	601	none

Process Downtime During Migration



Process Downtime During Migration

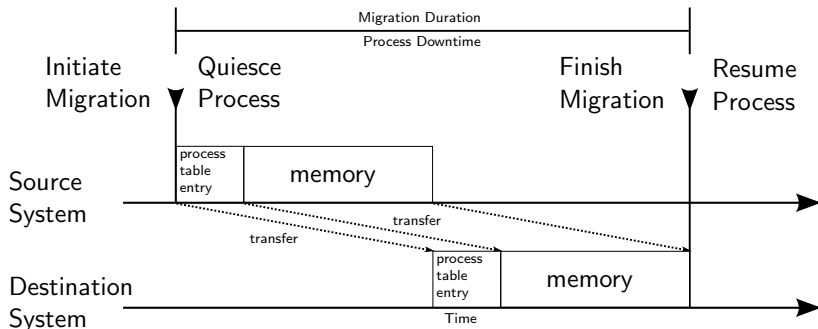


Figure: Process Migration

Optimizations - Pre-Copy

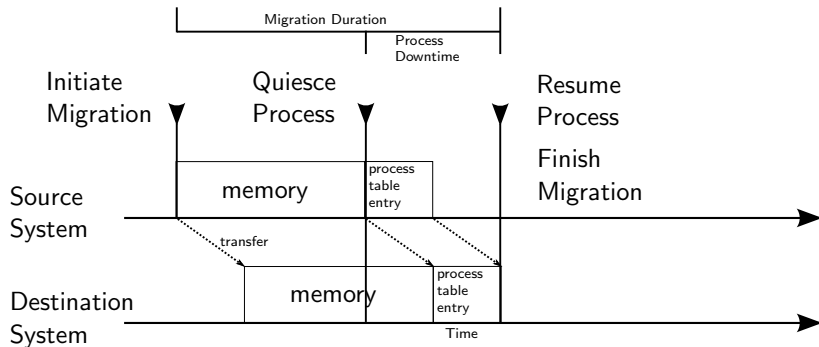
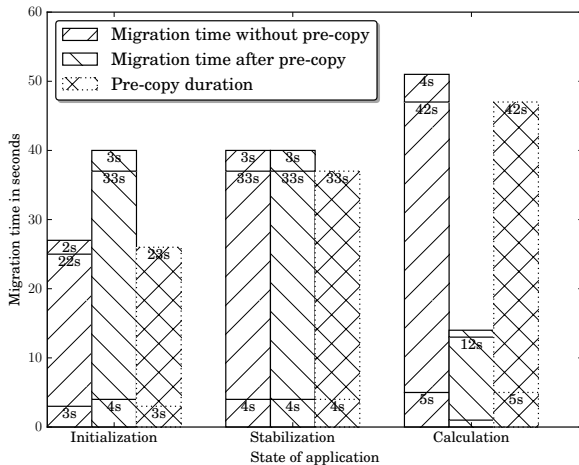
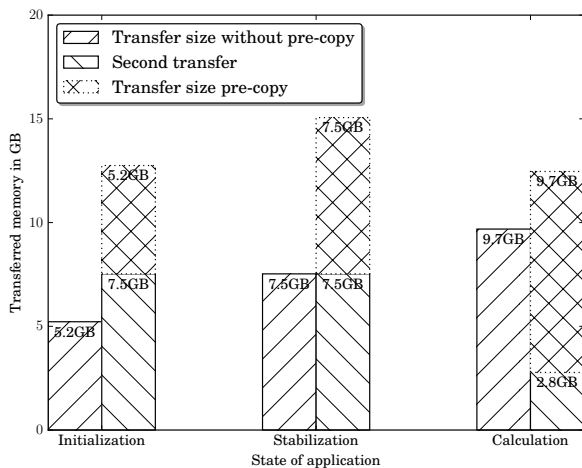


Figure: Pre-Copy Migration

Possible Drawback Using Pre-Copy



Possible Drawback Using Pre-Copy



Optimizations - Post-Copy

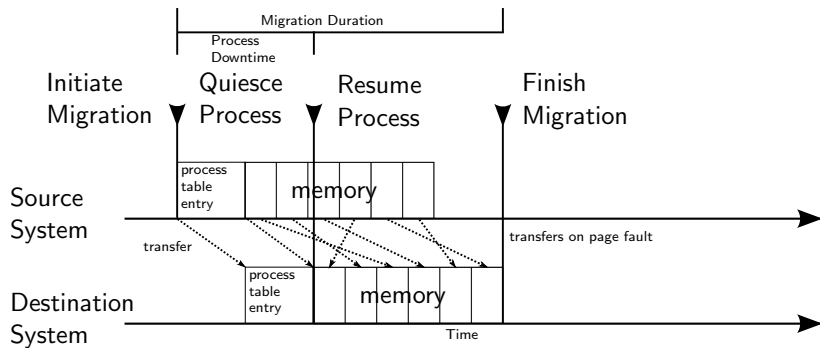


Figure: Post-Copy Migration

CRIU And Userfaultfd

- ▶ Userfaultfd (UFFD) integration into CRIU
- ▶ Most pages can be handled by UFFD
 - ▶ No need to transfer those pages to restore the process
- ▶ Process downtime can be increased
- ▶ To restore a 200MB process
 - ▶ transfer 200MB without Post-Copy
 - ▶ transfer 116KB with Post-Copy

The end.

Thanks for listening.