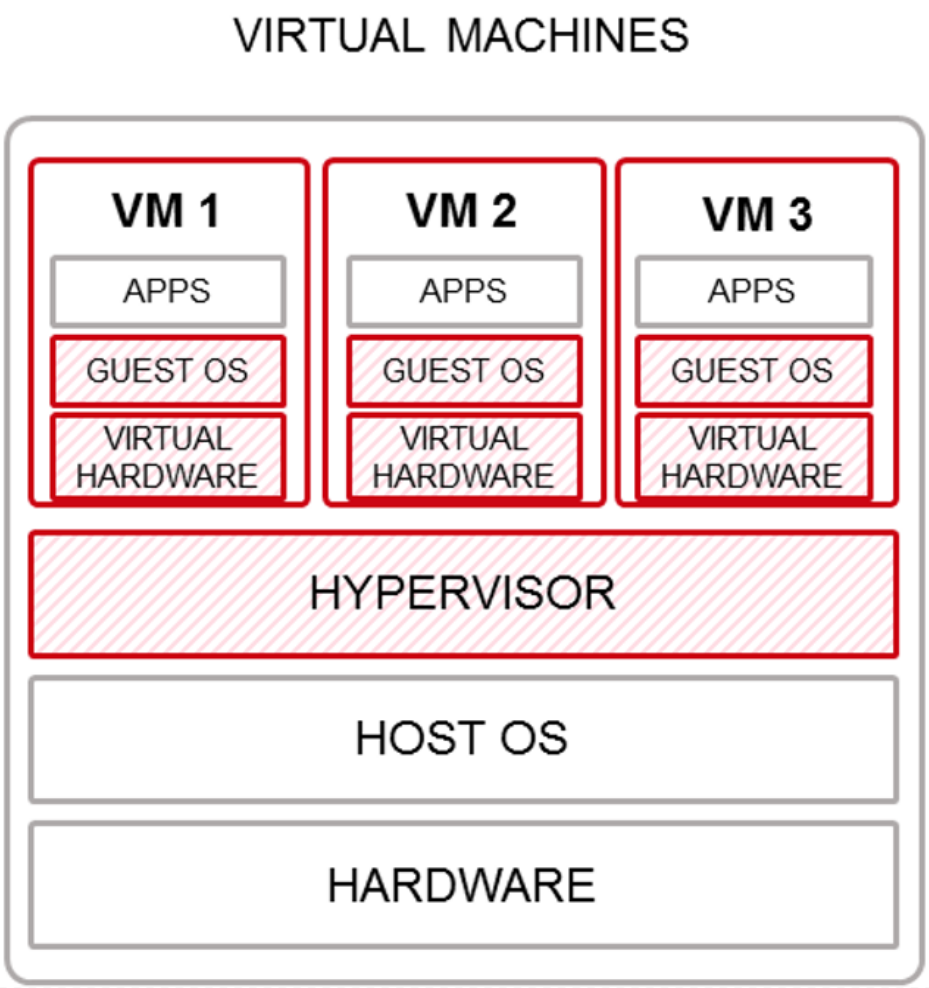
Efficient Live Migration of Linux Containers

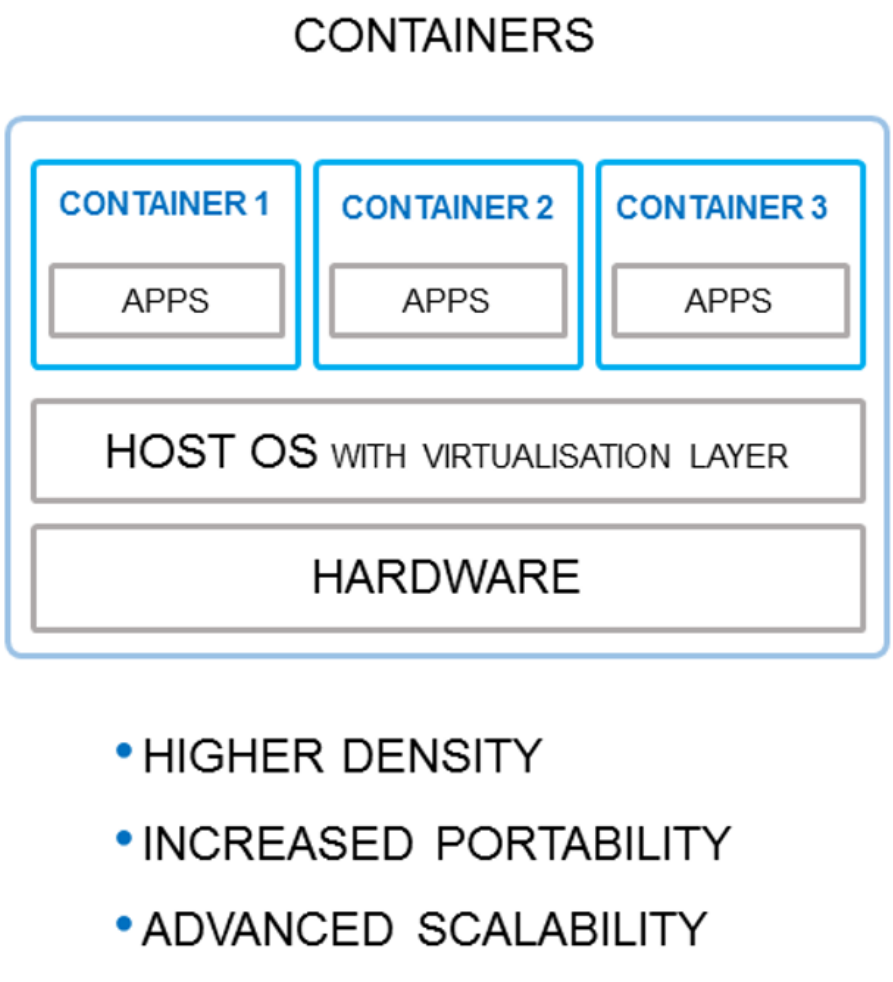
13th Workshop on Virtualization in High-Performance Cloud Computing VHPC ‘18

ISC-HPC, June 28, 2018, Frankfurt, Germany

Radostin Stoyanov < r.stoyanov.14@aberdeen.ac.uk >

Motivation

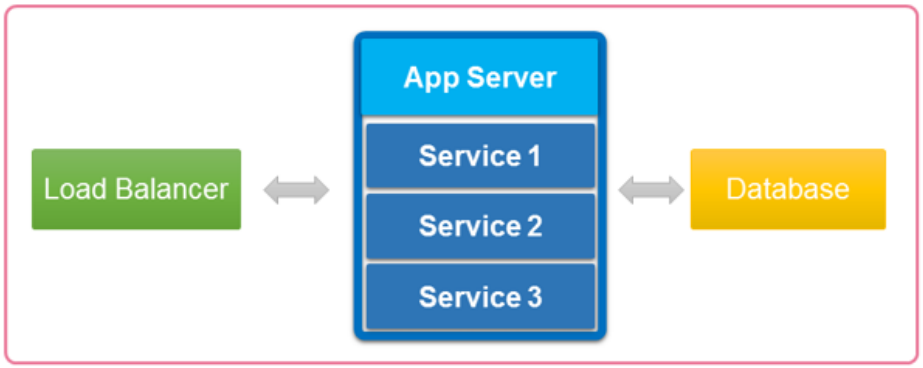




- HIGHER DENSITY
- INCREASED PORTABILITY
- ADVANCED SCALABILITY

Motivation

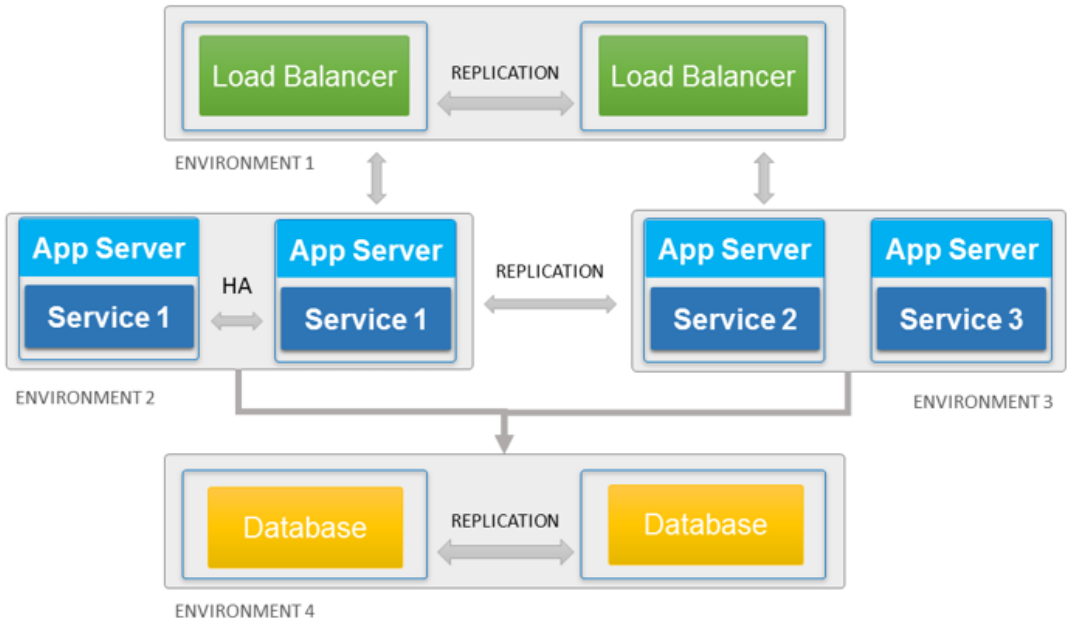
MONOLITHIC APPLICATION



Single monolithic VM

- Often redeploy everything
- Mutual dependencies slow down development
- Long QA cycle leads to less often updates
- High risk of failure or VM overload
- Difficult to scale

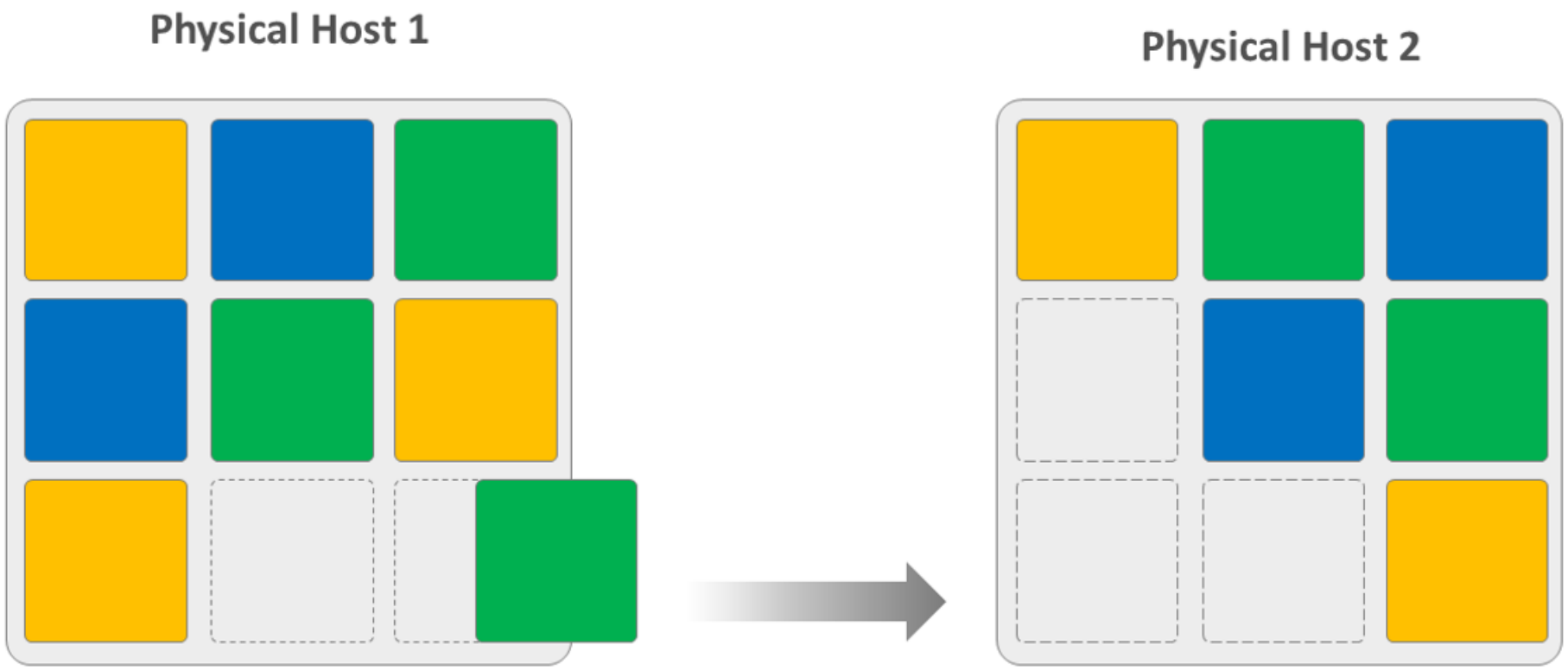
MICROSERVICES



Multiple Containers

- Modularity
- Deployed and updated independently
- Easy to scale and maintain
- Flexible architecture

Motivation

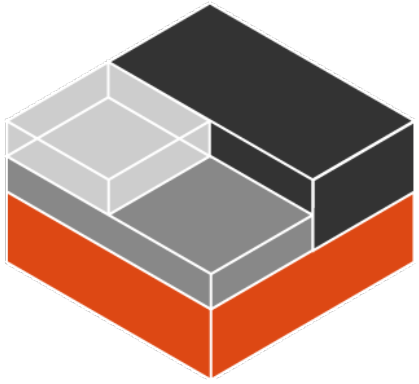

Physical Host 1
Physical Host 2
Live Migration of Containers

Virtualisation Technologies

Container-based Virtualisation

Checkpoint and Restore in Userspace

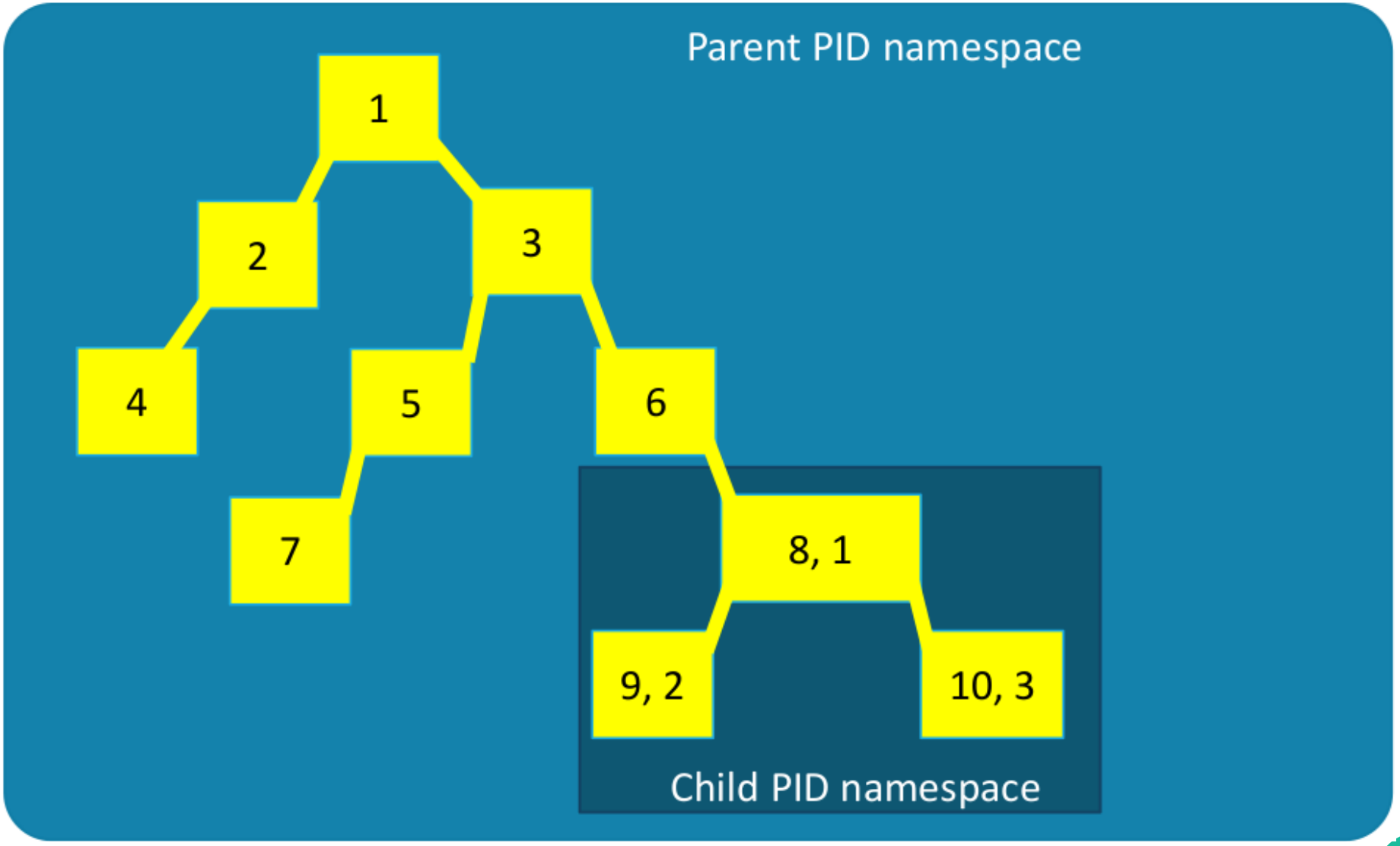
OPEN CONTAINER INITIATIVE



CRIU

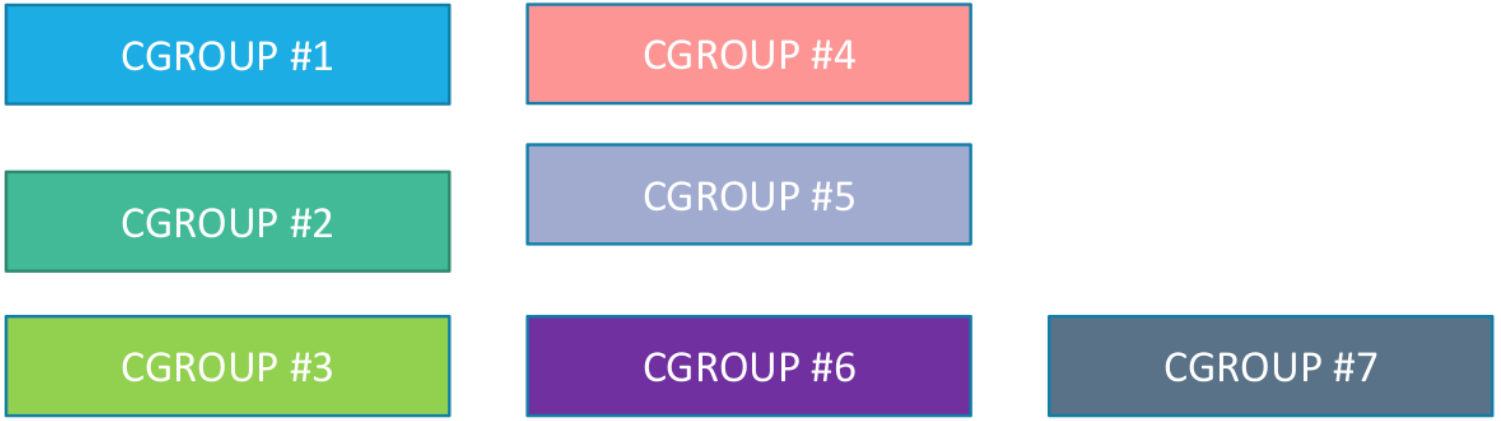
docker

Linux Namespaces

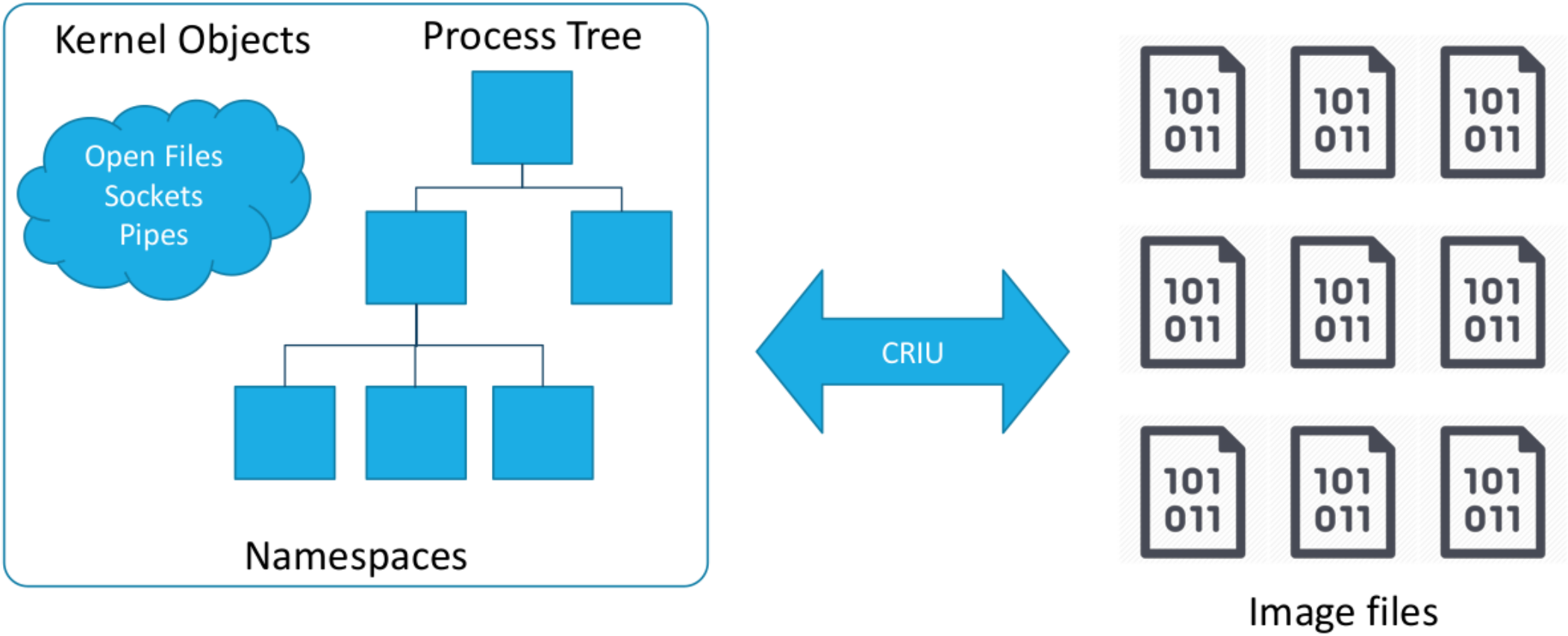


Linux Control Groups

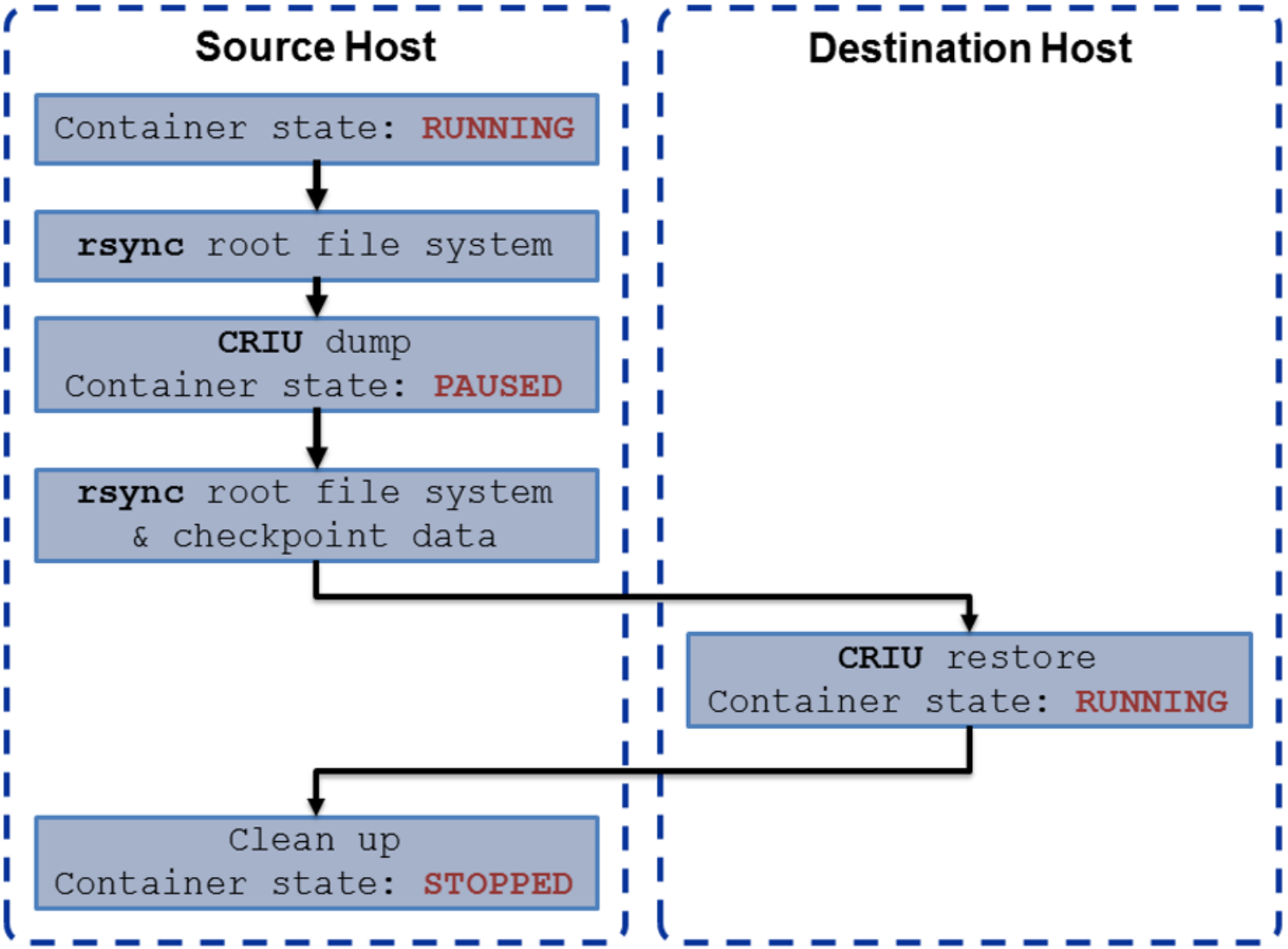
- Resource limits
- Prioritisation
- Control



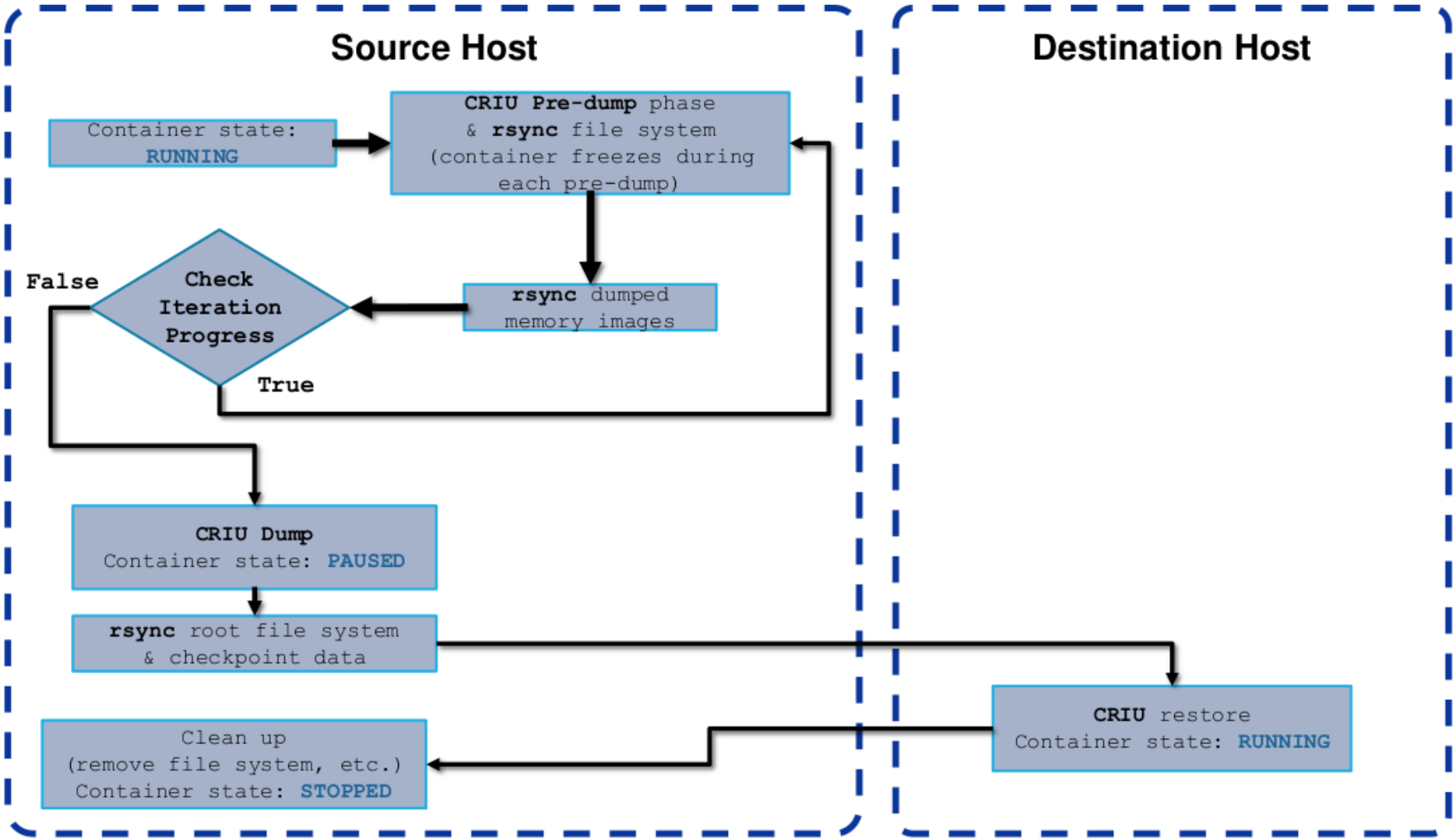
Checkpoint and Restore in Userspace



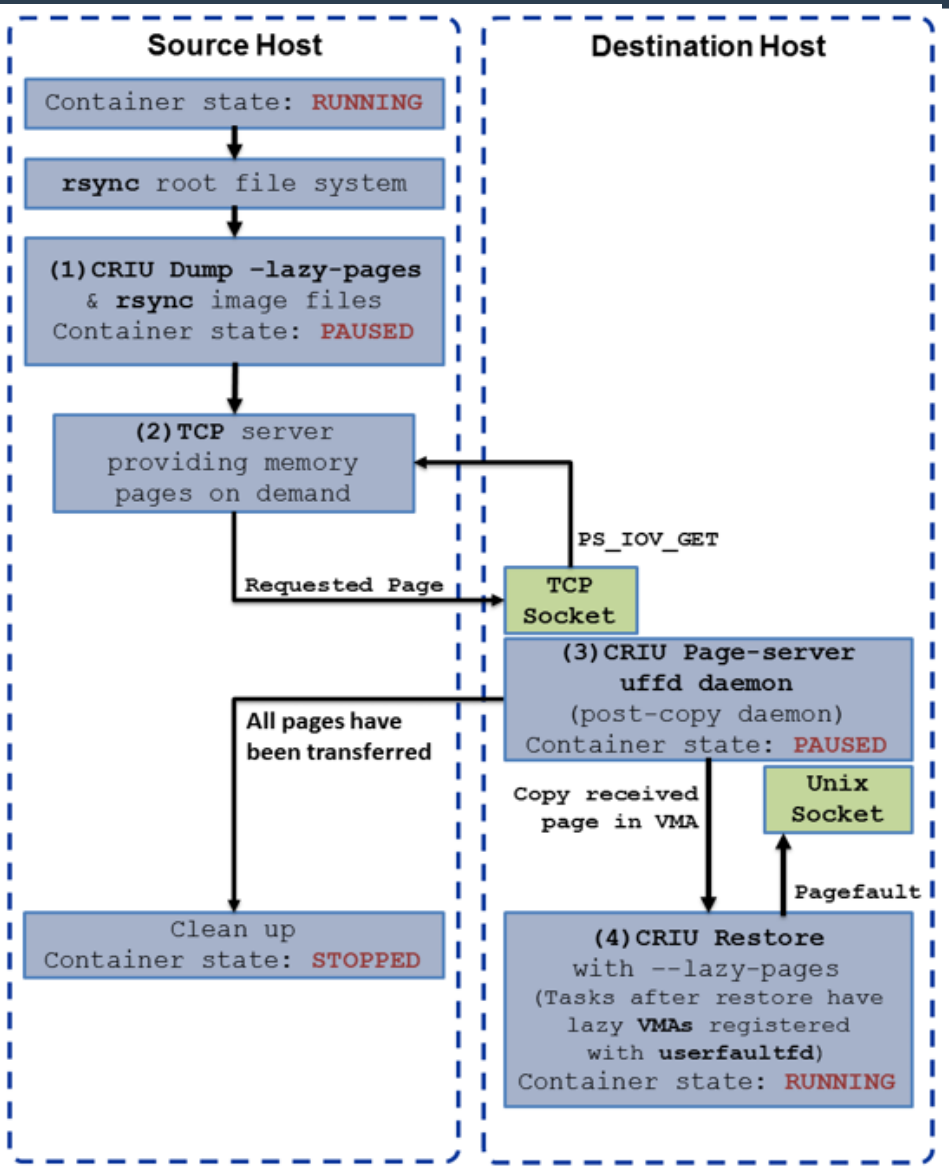
Native Live Migration



Pre-copy Live Migration



Post-copy Live Migration



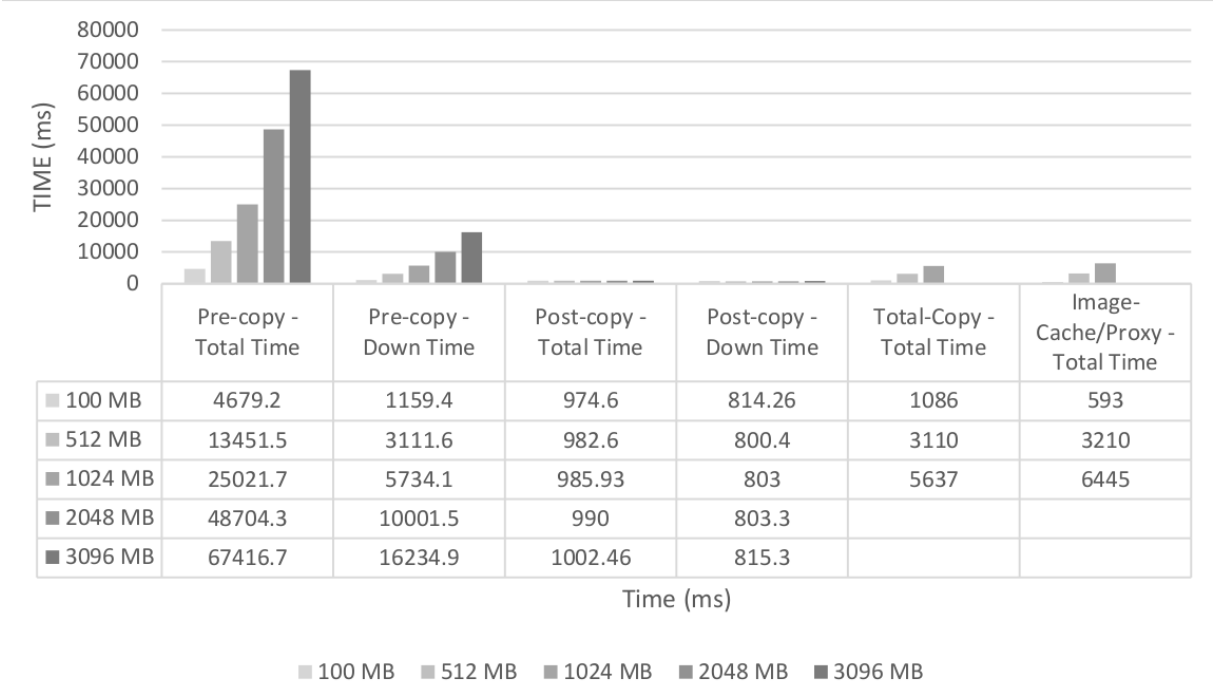
Evaluation - Experimental Design

Algorithm

- Pre-copy
- Post-copy
- Native
- Image-Cache/Proxy

Application

memhog



	Pre-copy - Total Time	Pre-copy - Down Time	Post-copy - Total Time	Post-copy - Down Time	Total-Copy - Total Time	Image-Cache/Proxy - Total Time
100 MB	4679.2	1159.4	974.6	814.26	1086	593
512 MB	13451.5	3111.6	982.6	800.4	3110	3210
1024 MB	25021.7	5734.1	985.93	803	5637	6445
2048 MB	48704.3	10001.5	990	803.3		
3096 MB	67416.7	16234.9	1002.46	815.3		

Conclusion & Future work

- Identifying Writable Working Set
- Resource Usage Control During Live Migration
- Delta Compression Based Memory Transfer
- Post-copy Bubbling with Multiple Pivots

Thank you!