Enabling Secure Container Checkpointing for Distributed Model Training

Radostin Stoyanov - PhD Student, Scientific Computing Group

Collaboration with Viktória Spišaková, Behouba Manassé, Adrian Reber

Supervisors: Prof. Rodrigo Bruno, Prof. Wes Armour

Challenges with Distributed Training

Single GPU failure may require restarting the entire training job

- 54 days training — 466 job interruptions [1]
  - ~78% of unexpected interruptions attributed to hardware issues
- 3-23 hours MTBF on older GPUs [2]
- Estimated monthly cost up to few million dollars, depending on job size [3]

Inefficient Checkpointing — Low GPU Utilization [4]

- Checkpointing larger models leads to longer GPU idle times

[1] Aaron Grattafiori, et al. "*The Llama 3 Herd of Models*". 2024
[2] Myeongjae Jeon, et al. "*Analysis of Large-Scale Multi-Tenant GPU Clusters for DNN Training Workloads*". USENIX ATC '19.
[3] Tanmaey Gupta, et al. "*Just-In-Time Checkpointing: Low Cost Error Recovery from Deep Learning Training Failures*". EuroSys '24.
[4] Yanjie Gao, et al. "*An Empirical Study on Low GPU Utilization of Deep Learning Jobs*". ICSE '24

Error Recovery Tradeoffs

Model Checkpoints — Implement error recovery in user code

- Requires restarting training jobs & re-running application code
- Involves potentially large overheads of job initialization

Infrastructure Checkpoints — Transparent error recovery at system-level

- Enables checkpointing for all jobs without any user code changes [1]
- Supports transparent job migration with existing cluster schedulers [2, 3, 4]

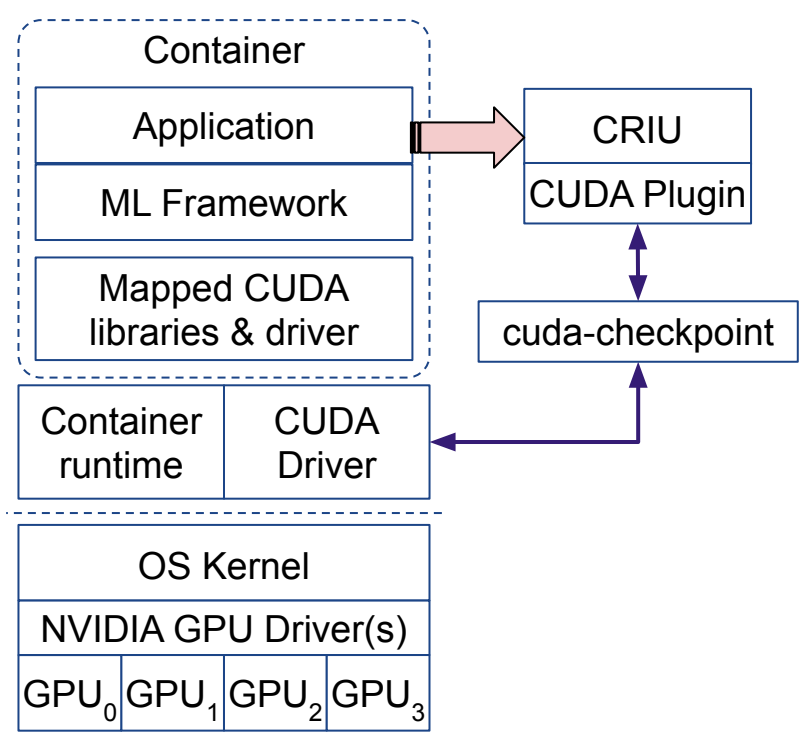
[1] Tanmaey Gupta, et al. "*Just-In-Time Checkpointing: Low Cost Error Recovery from Deep Learning Training Failures*". EuroSys '24.
[2] Dharma Shukla, et al. "*Singularity: Planet-Scale, Preemptive and Elastic Scheduling of AI Workloads*". (2022)
[3] Victor Marmol, et al. "*Task Migration at Scale Using CRIU*". Linux Plumbers Conference 2018
[4] Andy Tucker, et al. "*Task Migration at Google Using CRIU*". Linux Plumbers Conference 2018

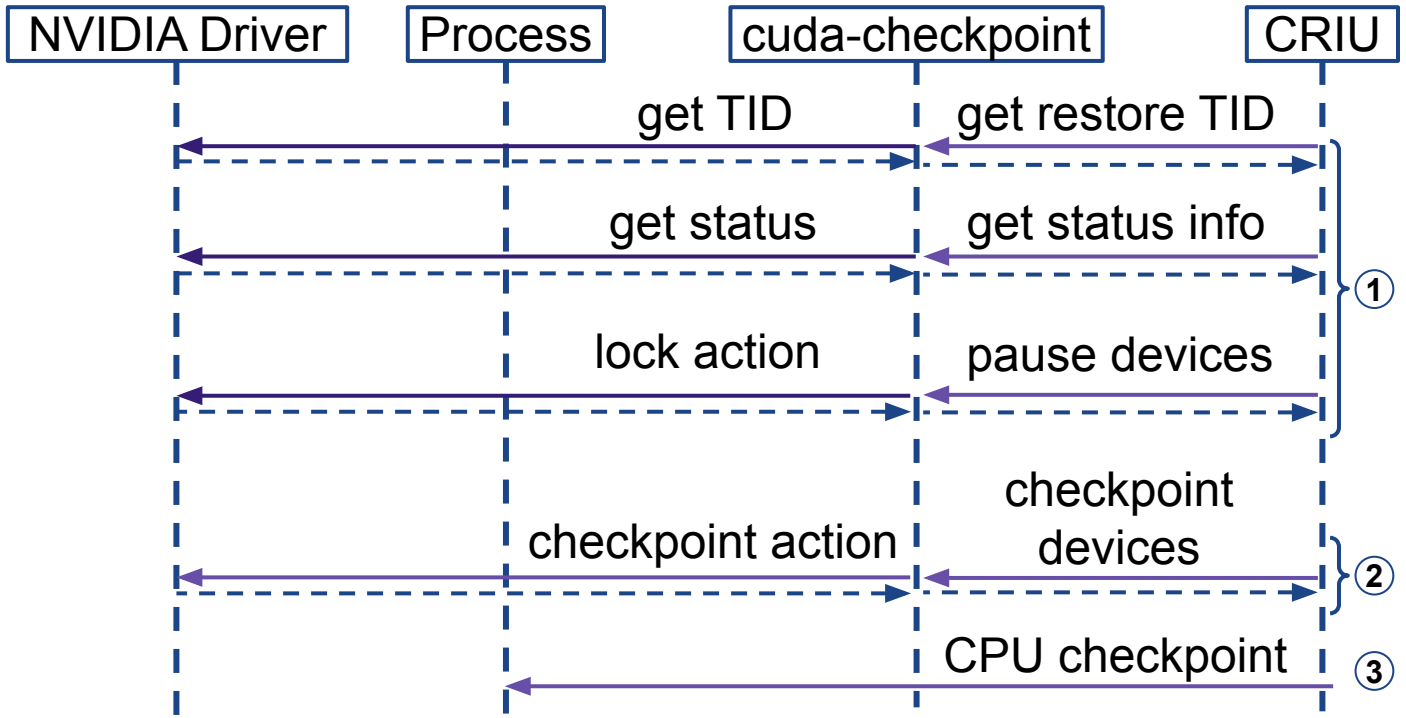
Transparent Checkpointing of GPU Workloads

Enabling transparent fault-tolerance for training jobs

Transparent GPU Checkpointing

CRIU with CUDA Plugin





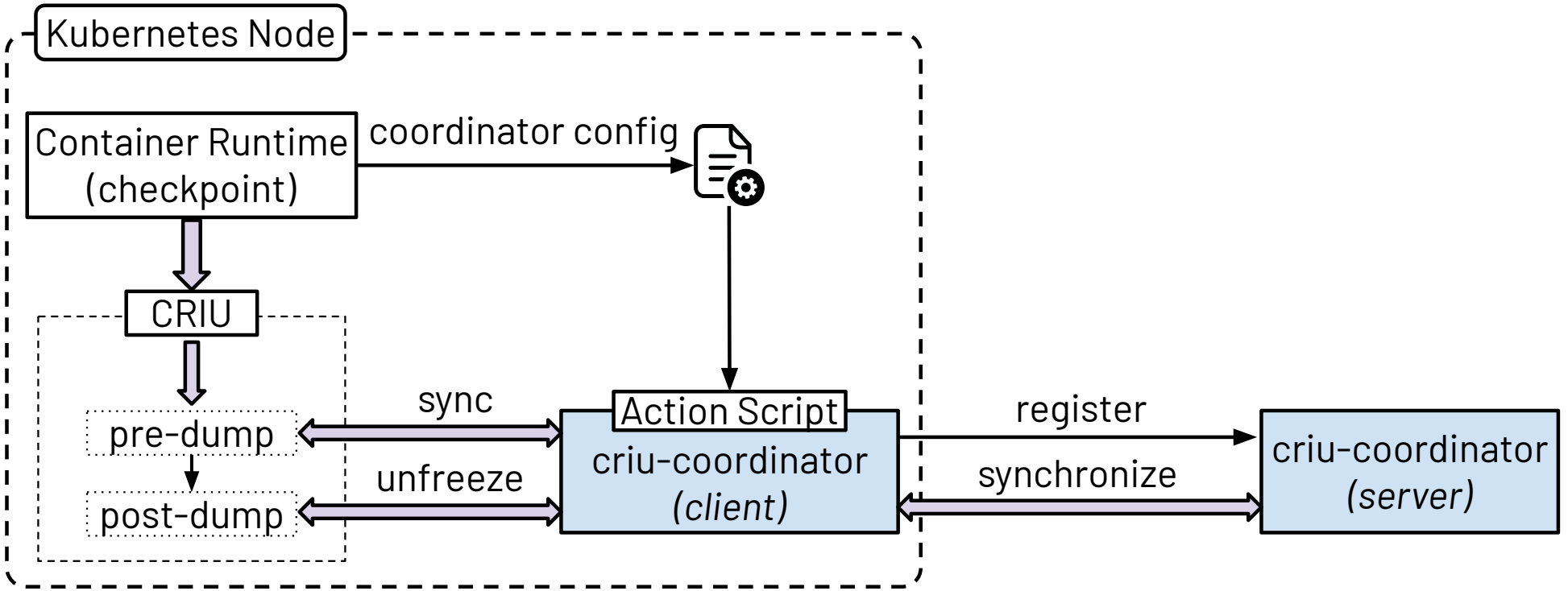
"*CRIUgpu: Transparent Checkpointing of GPU-Accelerated Workloads*". Radostin Stoyanov, Viktória Spišaková, Jesús Ramos, Steven Gurfinkel, Andrei Vagin, Adrian Reber, Wesley Armour, Rodrigo Bruno. (2025).

Coordinated Checkpointing for Distributed Training

Enabling low-cost error recovery in distributed training

Coordinated Checkpointing

CRIU Coordinator



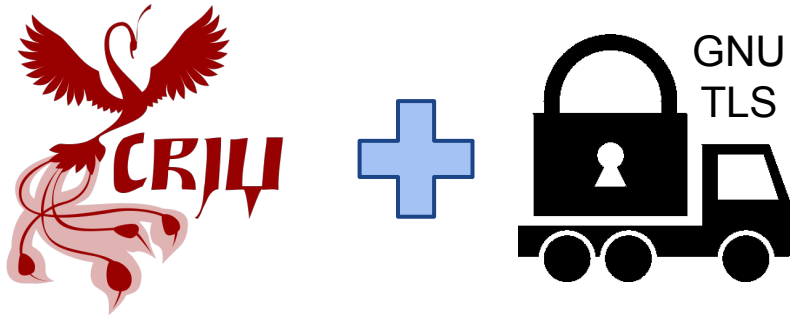
"Checkpoint Coordination for Distributed Containerized Applications". Linux Plumbers Conference (2024)
"Enabling Coordinated Checkpointing for Distributed HPC Applications". KubeCon Europe (2024)

DEPARTMENT OF ENGINEERING SCIENCE

UNIVERSITY OF OXFORD

End-to-end Encryption for Container Checkpoints

Adding support for end-to-end checkpoint encryption



Checkpoint Encryption Methods

Local Encryption [1]

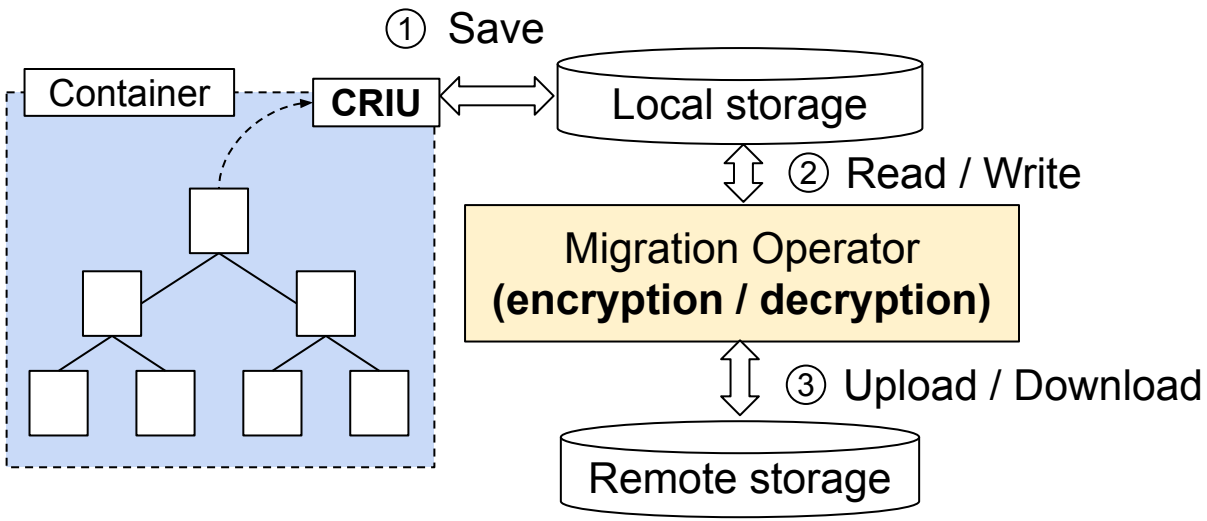
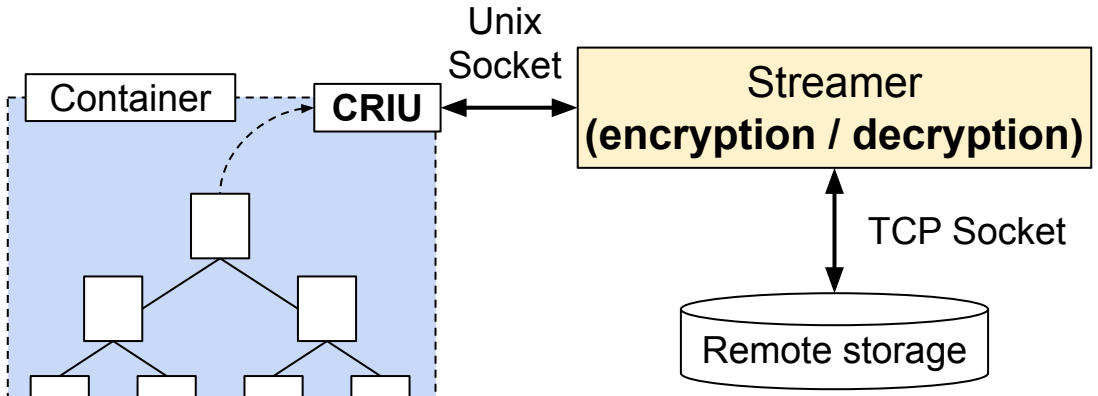


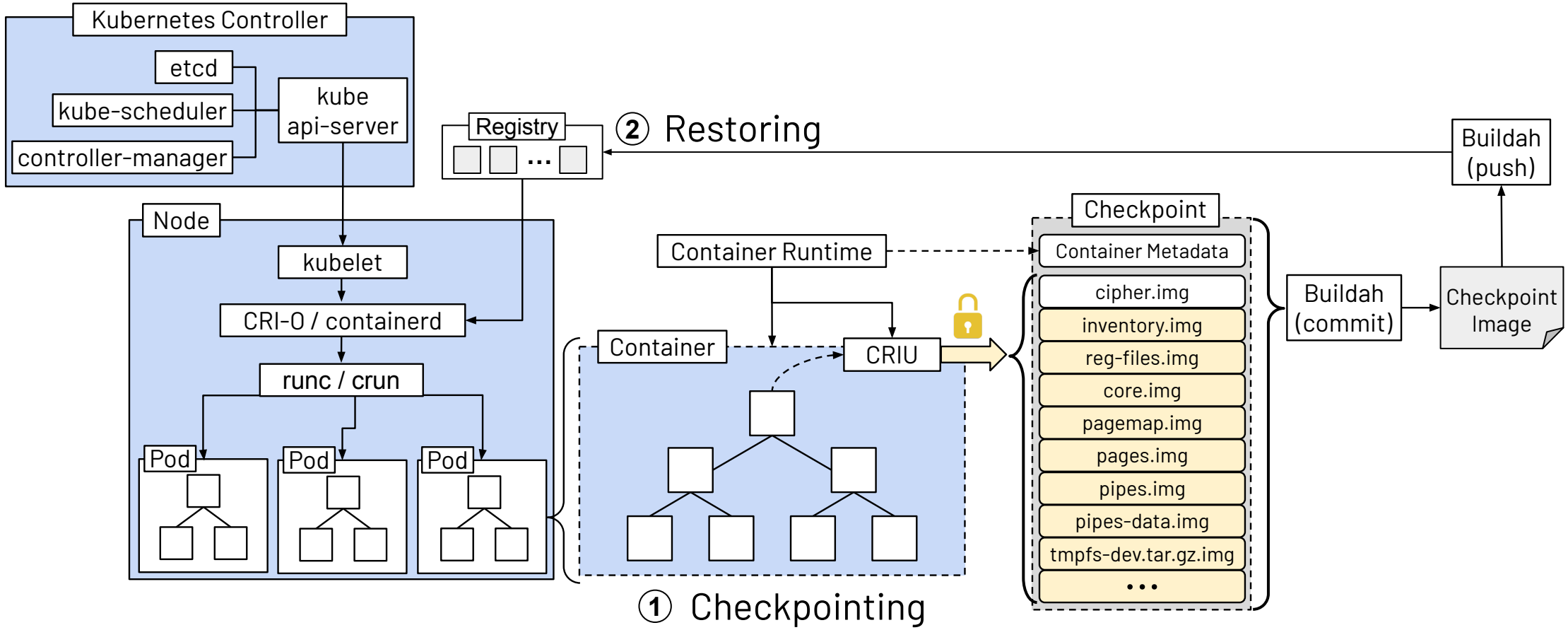
Image Streaming [2]



[1] Victor Marmol, et al. "*Task Migration at Scale Using CRIU*". Linux Plumbers Conference 2018
[2] Nicolas Viennot. "*Fast checkpointing with criu-image-streamer*". Linux Plumbers Conference 2020

End-to-end Checkpoint Encryption

CRIU with Built-in Encryption

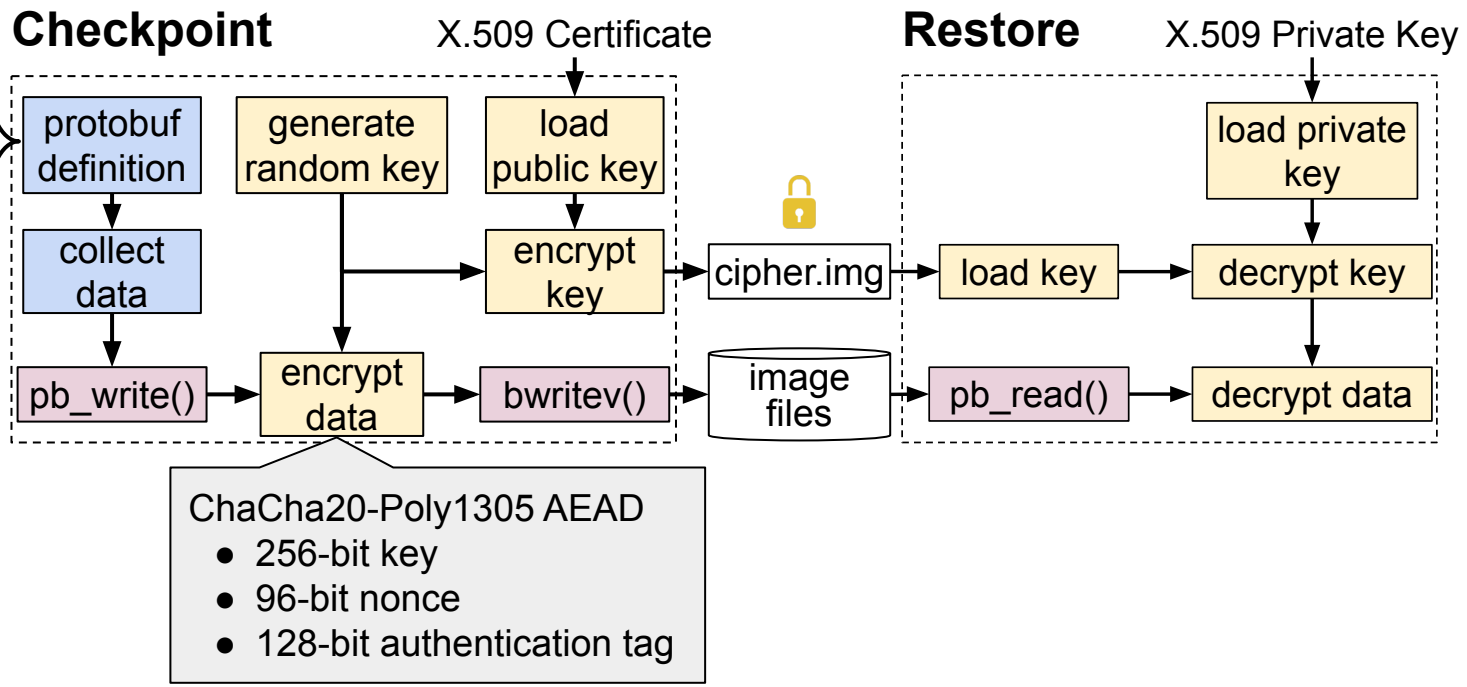


"End-to-End Encryption for Container Checkpointing in Kubernetes". CloudNativeSecurityCon North America 2024
"Towards Efficient End-to-End Encryption for Container Checkpointing Systems". Asia-Pacific Workshop on Systems 2024
"Protecting Sensitive Data in Container Checkpoints". Linux Plumbers Conference 2023

Built-in Encryption for CRIU Images

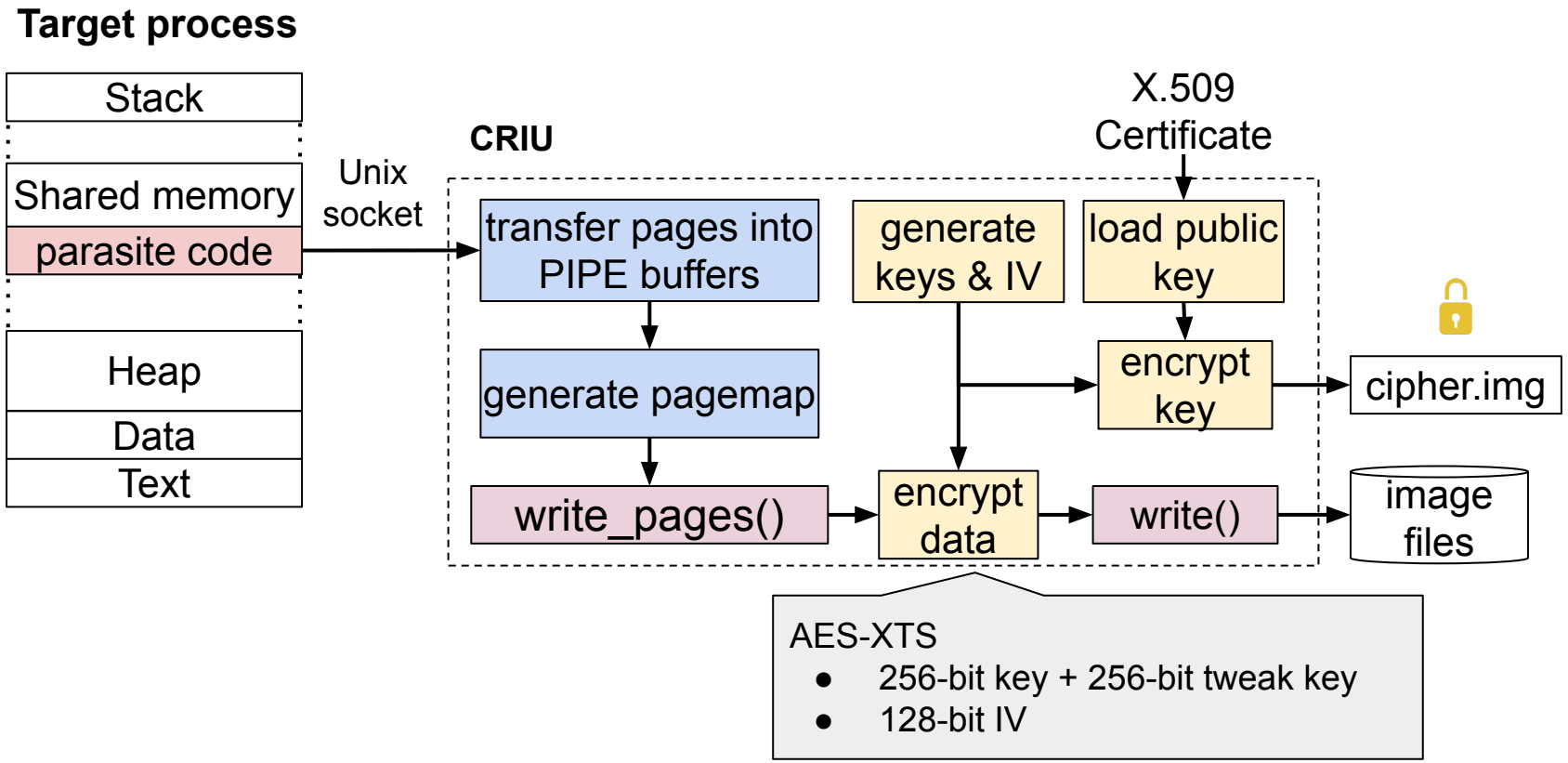
CRIU Images

```
syntax = "proto2";
message inventory_entry {
    required uint32 img_version = 1;
    optional bool fdinfo_per_id = 2;
    optional task_kobj_ids_entry root_ids = 3;
    optional bool ns_per_id = 4;
    optional uint32 root_cg_set = 5;
    optional lsmtype lsmtype = 6;
    optional uint64 dump_uptime = 8;
    optional uint32 pre_dump_mode = 9;
    optional bool tcp_close = 10;
    optional uint32 network_lock_method = 11;
}
```



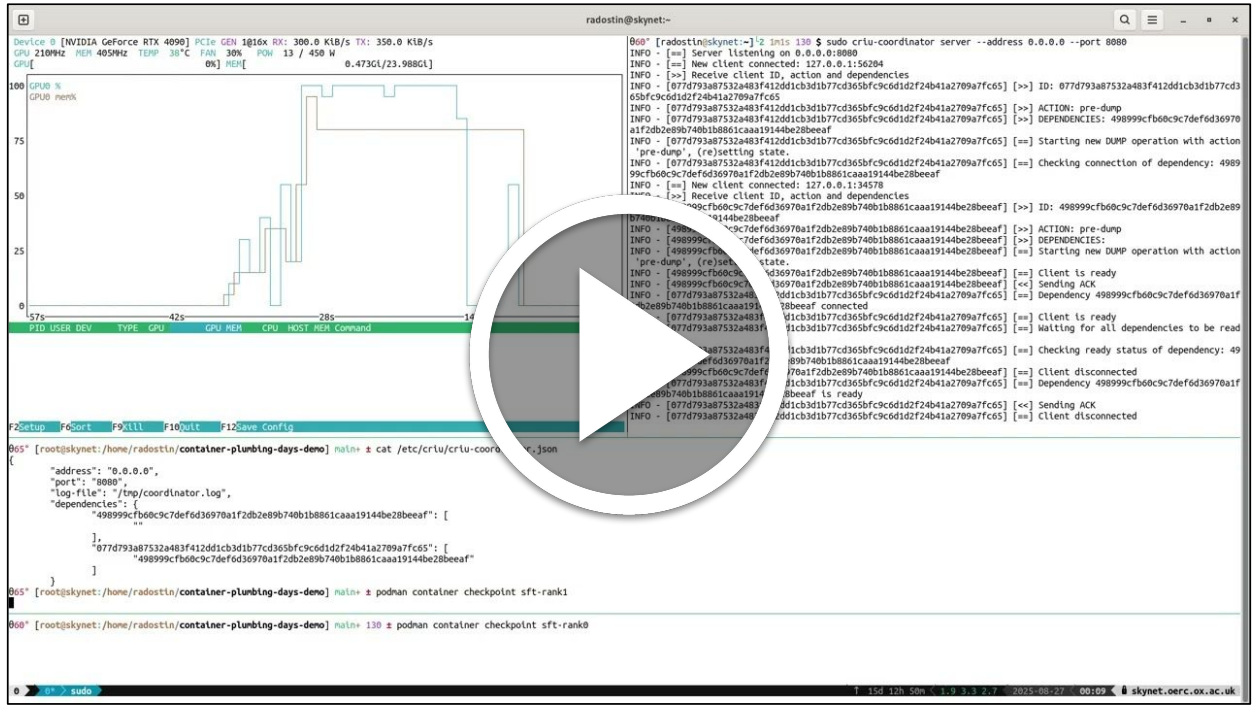
"End-to-End Encryption for Container Checkpointing in Kubernetes". CloudNativeSecurityCon North America 2024
"Towards Efficient End-to-End Encryption for Container Checkpointing Systems". Asia-Pacific Workshop on Systems 2024
"Protecting Sensitive Data in Container Checkpoints". Linux Plumbers Conference 2023

Encryption of Memory Pages



"End-to-End Encryption for Container Checkpointing in Kubernetes". CloudNativeSecurityCon North America 2024
"Towards Efficient End-to-End Encryption for Container Checkpointing Systems". Asia-Pacific Workshop on Systems 2024
"Protecting Sensitive Data in Container Checkpoints". Linux Plumbers Conference 2023

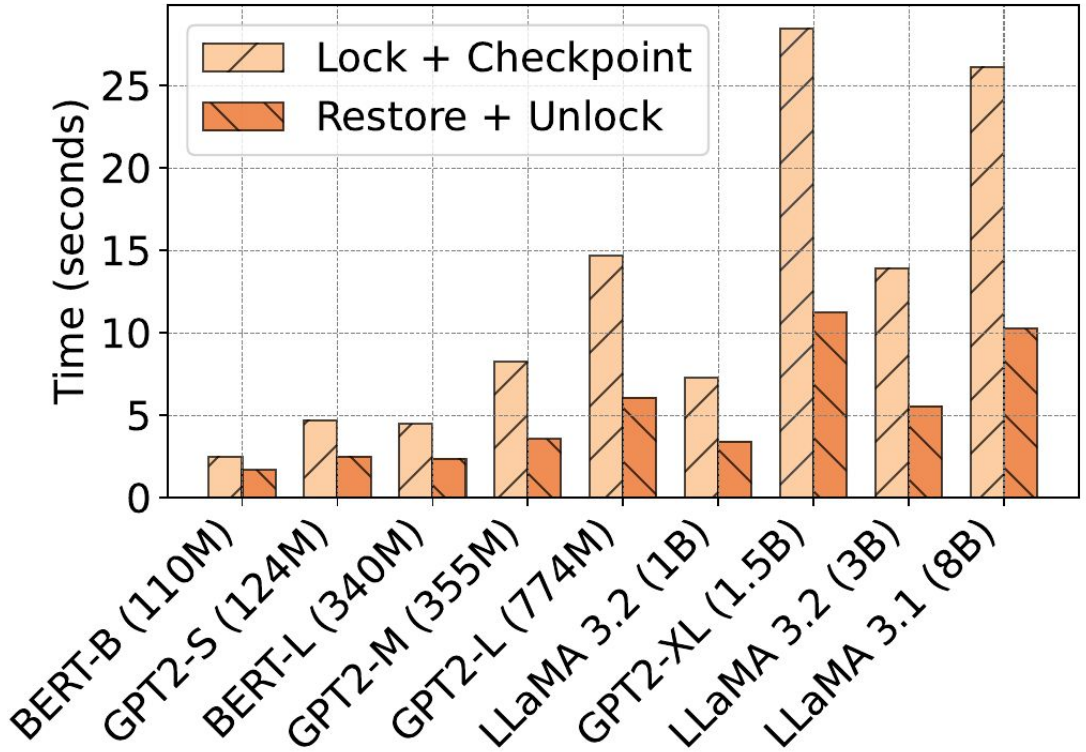
Coordinated Checkpointing Demo

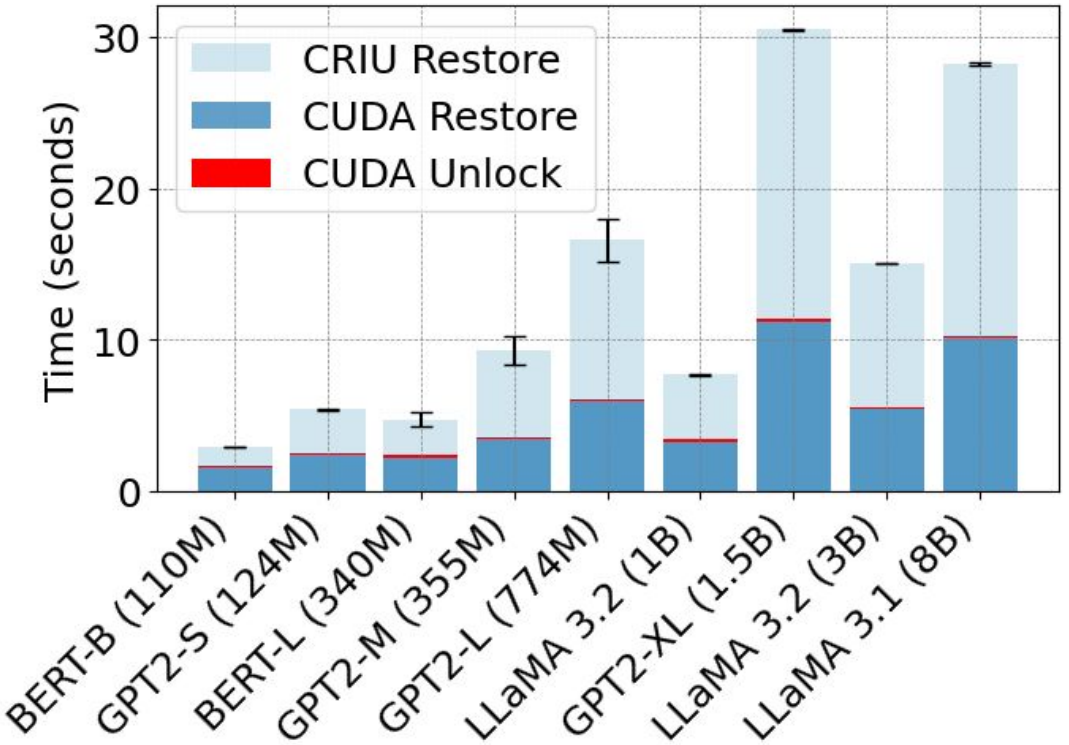


Checkpointing for Distributed Training

- Fine-tuning LLaMA 3.2-1B on Guanaco dataset
- 2 GPU containers running PyTorch DDP
  - `torchrun` with 2 nodes, 1 process/node
- NVIDIA GeForce RTX 4090 (24GB)

Evaluation with Training Workloads

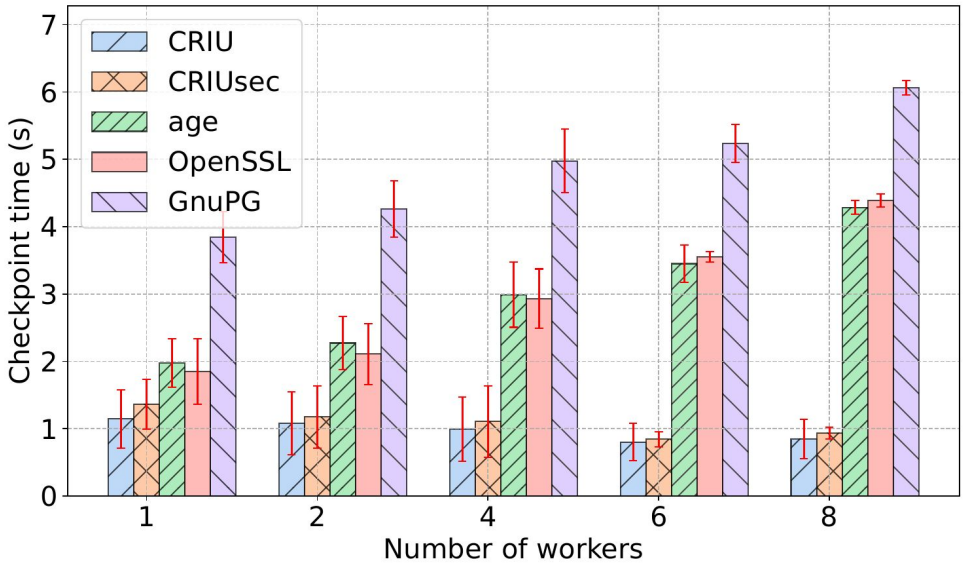




NVIDIA H100 (PCIe 5.0 80GB HBM3)

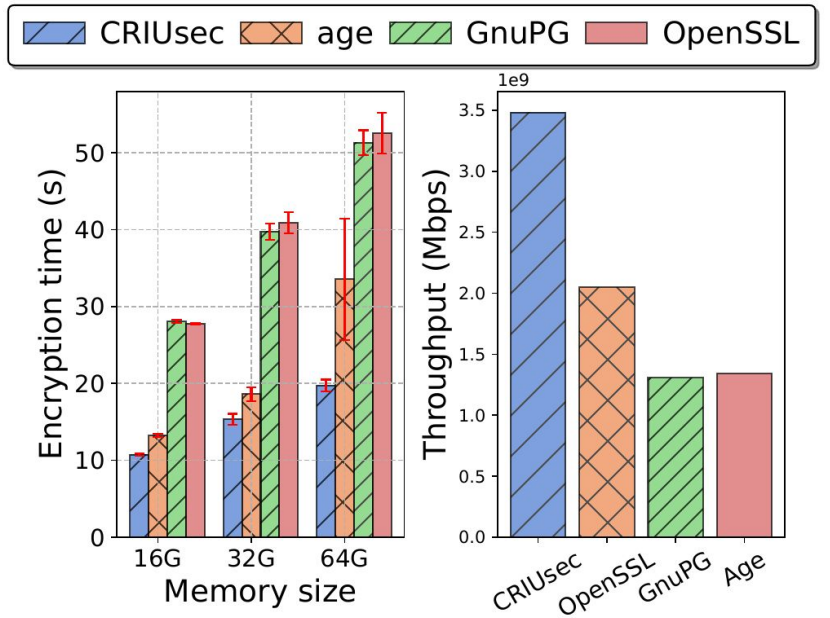
Evaluation of End-to-End Encryption

Checkpoint time for compute-intensive workloads (stress-ng)



Up to two orders of magnitude faster checkpoint creation

Encryption throughput for memory-intensive workloads (memhog)



Up to 62% reduced encryption overhead

Next Steps and Future Work

- Kubernetes Checkpoint/Restore Working Group
  - https://github.com/kubernetes/community/pull/8508

- Enabling Transparent GPU Checkpointing of TrainJobs in Kubeflow Trainer
  - https://github.com/kubeflow/trainer/issues/2777

DEPARTMENT OF ENGINEERING SCIENCE
UNIVERSITY OF OXFORD

Summary & Questions

- Transparent Checkpointing of Distributed Training Jobs
- End-to-end Encrypted Container Checkpoints
- Out-of-the-box Integration with Container Platforms

github.com/nvidia/cuda-checkpoint
github.com/checkpoint-restore/criu
github.com/checkpoint-restore/criu-coordinator