Picocenter: Supporting long-lived, mostly-idle applications in cloud environments

Liang Zhang[*] James Litton[†] Frank Cangialosi[†]
Theophilus Benson[‡] Dave Levin[†] Alan Mislove[*]

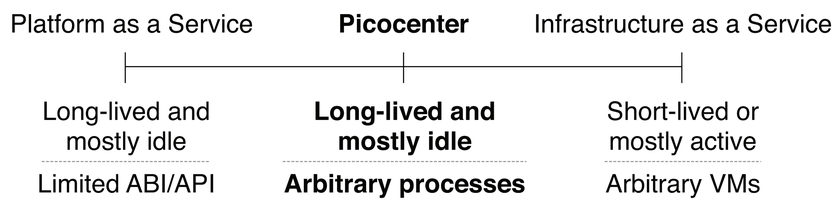
[*]Northeastern University [†]University of Maryland [‡]Duke University

Motivation

End-users wish to run wide range of applications in the cloud
- E.g., iRedMail, ownCloud, GitLab, Rocket.Chat

These applications are long-lived but mostly idle (LLMI) apps
- Long-lived: Users wish them to be available for a long time
- Mostly-idle: Personalized services are likely to be idle

Problem:
- Current cloud computing models are not suited for LLMI apps



Options for running LLMI apps in the cloud today:
1. Platform as a Service (PaaS): limited programming environment
2. Infrastructure as a Service (IaaS): can be expensive to run

Goal: Support LLMI apps in cloud environments
- Lower cost by allowing provider to run many LLMI apps
- Leverage workload by swapping idle apps to cloud storage

Picocenter

A hosting infrastructure designed to run lots of LLMI apps in the cloud

Provide a process-like environment and arbitrary network protocols
- Support wide variety of applications

Swap idle applications to cloud storage
- Use cloud resources efficiently, thus dramatically reduce cost

Key challenge: swap in application quickly on request
- Reactive page faulting and prefetching with ActiveSet

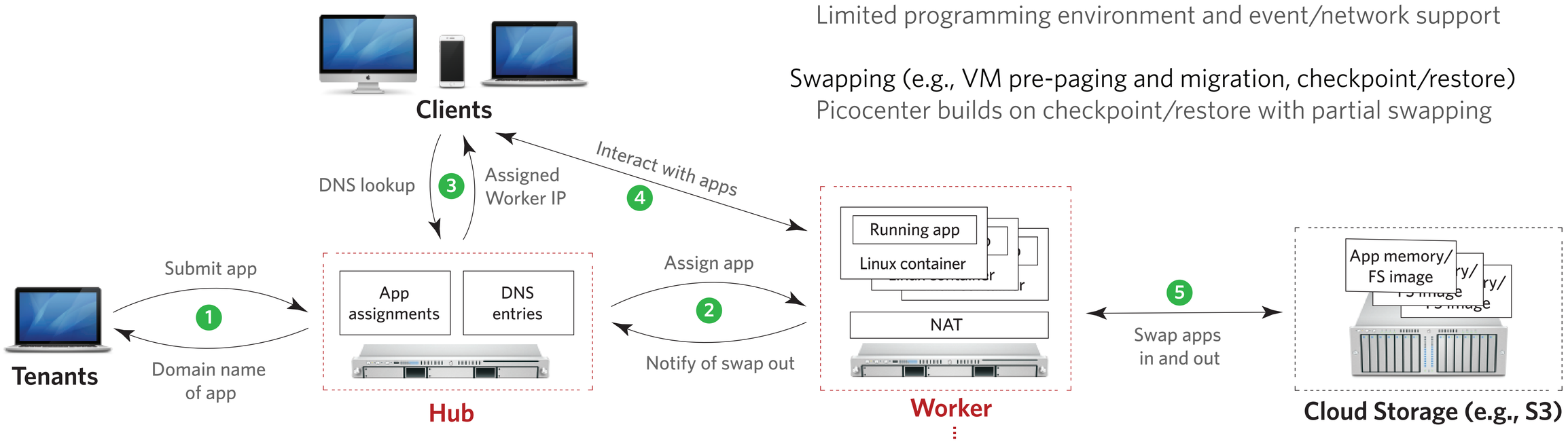
ActiveSet: predicted pages that are needed for the request
- Reduce total download size compared to full checkpoint
- Minimize fetching pages compared to reactive paging only

Related Work

Operating system containers (e.g., VServer, Docker, BSD Jail)
- Do not support checkpoint/restore and partial swap ins

Dedicated runtime (e.g., AppEngine, Lambda, Azure Functions)
- Limited programming environment and event/network support

Swapping (e.g., VM pre-paging and migration, checkpoint/restore)
- Picocenter builds on checkpoint/restore with partial swapping

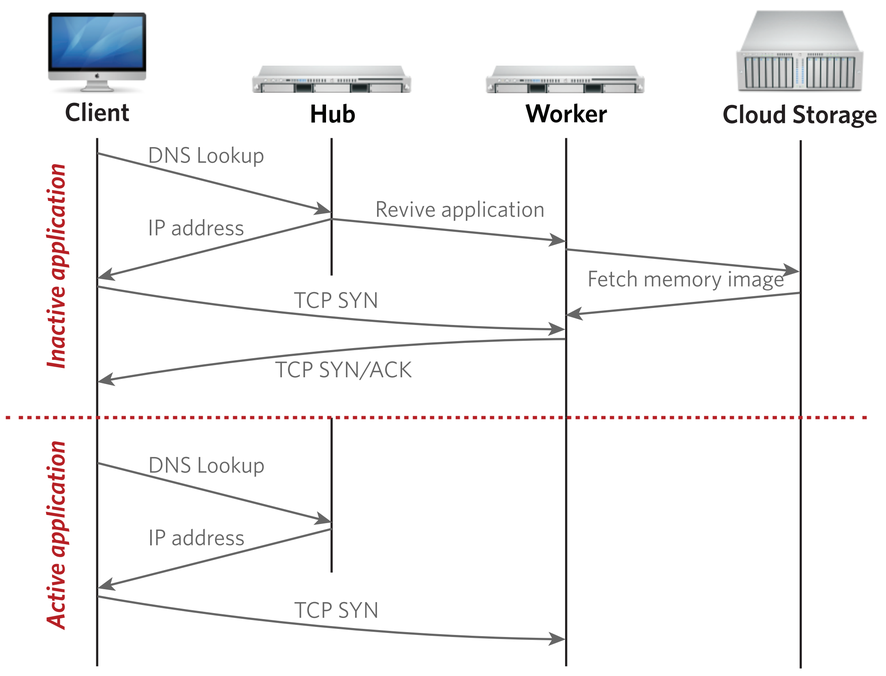


Design and Implementation

Process-like environment with LXC
- Users submit Docker-like app images and get back a DNS name
- An extended CRIU supports partial swap ins and ActiveSet
- A FUSE application catches page faults and builds ActiveSet

The Hub: Serves DNS requests; assigns apps to workers

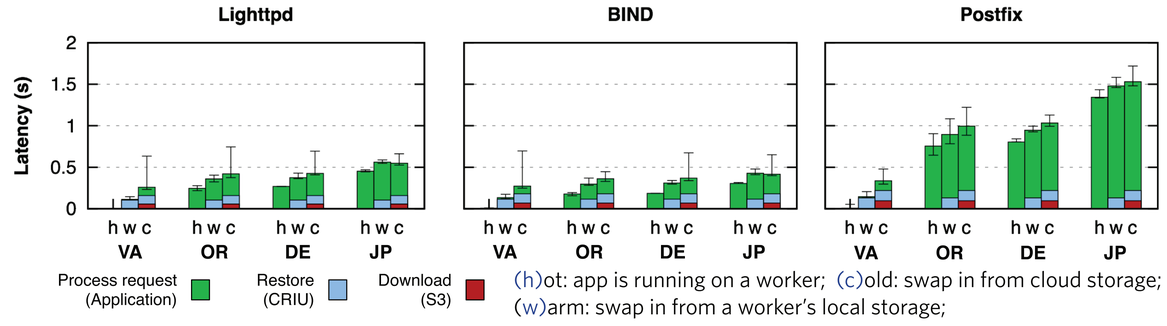
The Workers: Host and run applications; provide NAT for IPv4



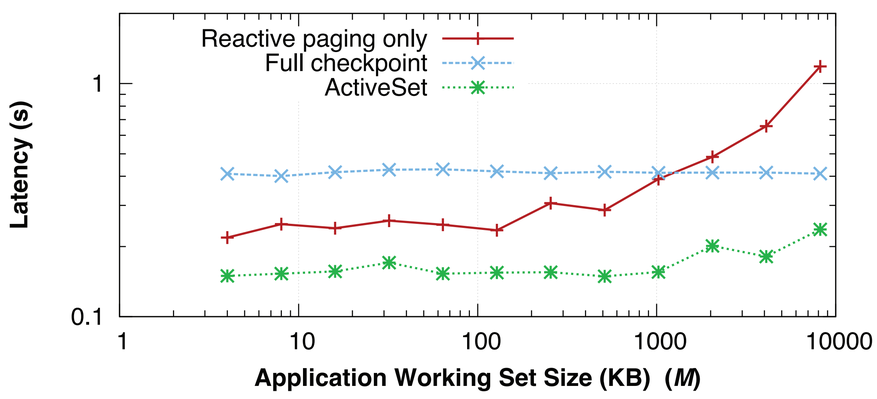
Evaluation

We deploy Picocenter on Amazon Virginia datacenter

How quickly can Picocenter restore real-world applications?
- Picocenter restores real-world applications in under 250 ms



How much does ActiveSet help to reduce the time to restore?
- ActiveSet reduces latencies by a factor of 1.5x – 5x



Source code: https://github.com/LeoLiangZhang/Picocenter