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

Liang Zhang  
Northeastern University  
liang@ccs.neu.edu

Frank Cangialosi  
University of Maryland  
frank@cs.umd.edu

Theophilus Benson  
Duke University  
tbenson@cs.duke.edu

Dave Levin  
University of Maryland  
dml@cs.umd.edu

Alan Mislove  
Northeastern University  
amislove@ccs.neu.edu

**Motivation**

End-users wish to run long-lived but mostly-idle (LLMI) apps  
E.g., web/email servers, distributed social networks

But running them in today’s cloud (e.g., AWS) is inefficient:  
User pays for an entire VM, even when the app is idle  
Provider reserves resources for idle VMs  
Idle apps need not stay in memory

Options for running LLMI apps in cloud today:  
Sharing VMs with users (Privacy and accounting challenges)  
PaaS, such as AppEngine (Limited programming environment)

Goal: **Support LLMI apps in cloud environments**  
Pay-by-usage billing, not charge by time  
Swap idle apps off to secondary storage

**Picocenter**

A hosting infrastructure designed to support LLMI apps  
Can be deployed on top of today’s cloud  
Run today’s apps efficiently (e.g., Nginx, Python apps)

Key challenge: VM is not designed for fast swapping

Solution: **Process-based model for computation consolidation**

Extend Picoprocess [NSDI’13] for LLMI apps in the cloud  
Fine-grained control of paging, networking and timer  
Transparent checkpoint and restore

**ActiveSet**: predictive page fetching  
Prefetch active working pages  
Predictive page loading based on packet metadata

**Design**

Each app runs inside its own picoprocess  
Fetched from cold storage on DNS request  
Moved back to cold storage when not active

The Hub: Manages DNS mappings and app assignments  
The Workers: Host picoprocesses and provide NAT network

**Evaluation**

We built a proof-of-concept implementation  
Deployed on AWS (Ravello) and local cluster

How fast can Picocenter swap picoprocesses?  
On the order of 100 ms, even for large working pages  
Cost for loading from S3 is dominated by network latency  
ActiveSet optimization provides a significant speedup

**Related Work**

Hardware virtualization (e.g., Xen, KVM)  
Performance and management overhead of running OS

Operating system containers (e.g., Docker, BSD jail)  
Bound to particular hosting operating system kernel

Process or VM migration  
Heavy operations; inspect kernel or hardware states