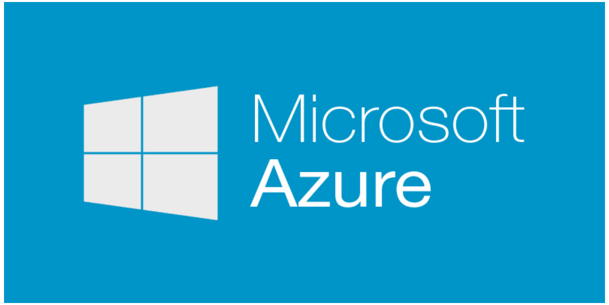
Virtual Middlebox Management for Cloud

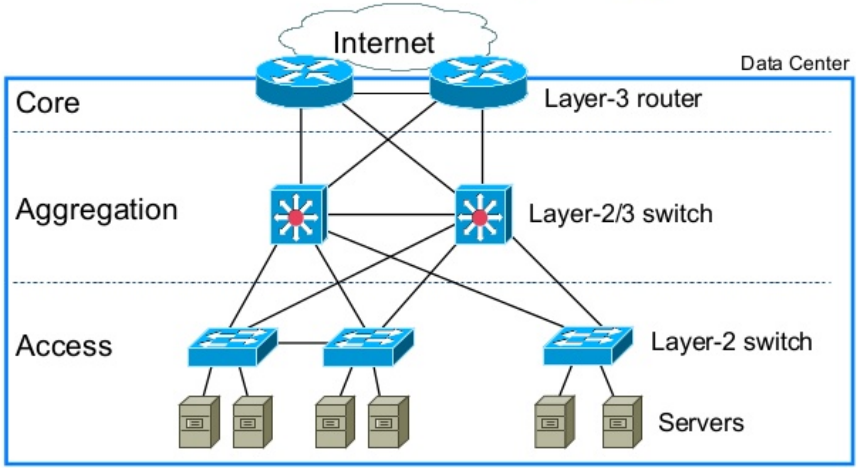
Peter Feifan Chen
Nodir Kodirov

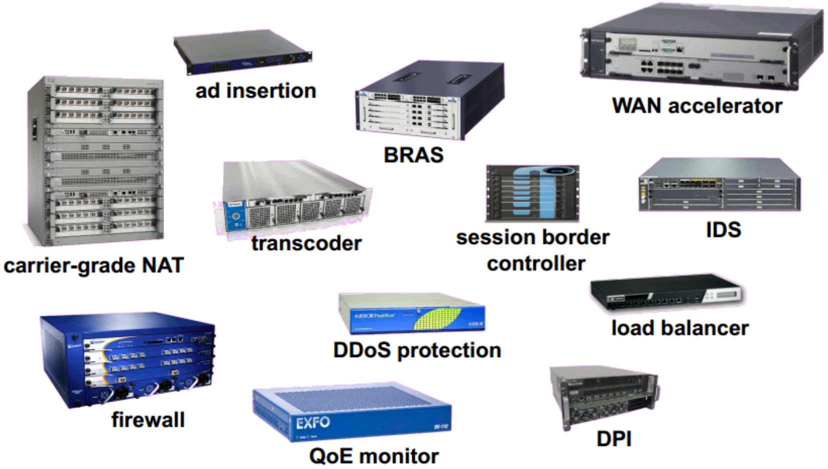
538B: Distributed Systems
Class project presentation
April 14, 2015

Motivation: Service providers

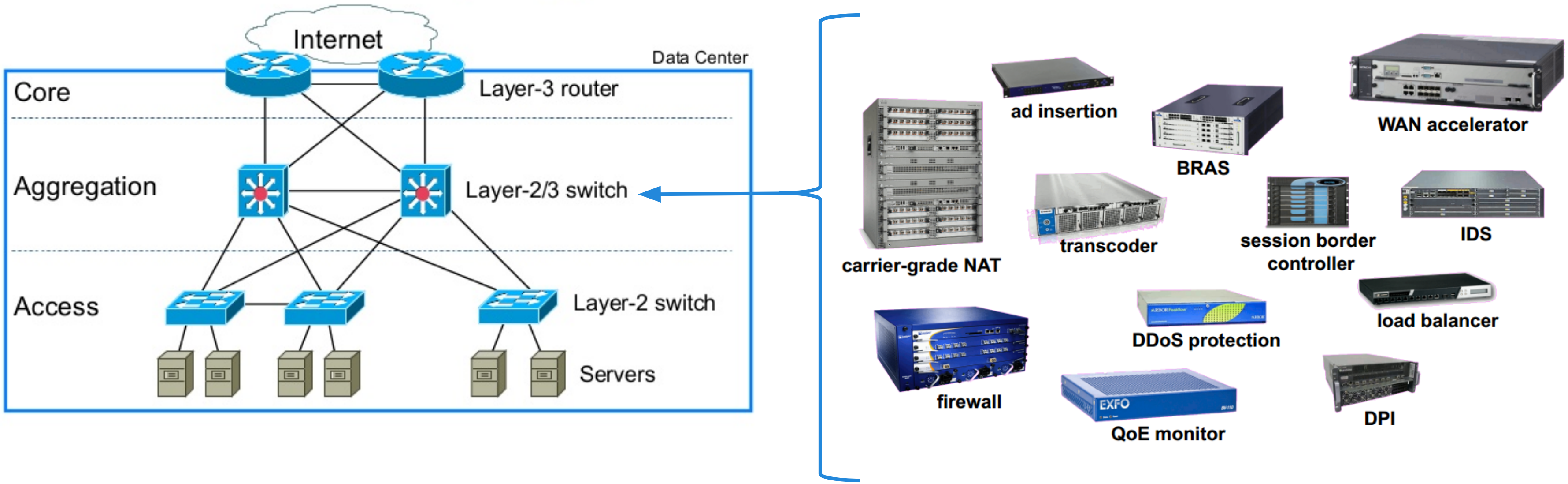


Motivation: Middleboxes in Data Center





Motivation: Middleboxes in Data Center



Motivation: Middlebox Virtualization

- Legacy: hardware middleboxes
- Recently: Virtual machine based
  - hypervisors: Xen, KVM

Motivation: Middlebox Virtualization

- Legacy: hardware middleboxes
- Recently: Virtual machine based
  - hypervisors: Xen, KVM
- More recently: containers

kubernetes

docker

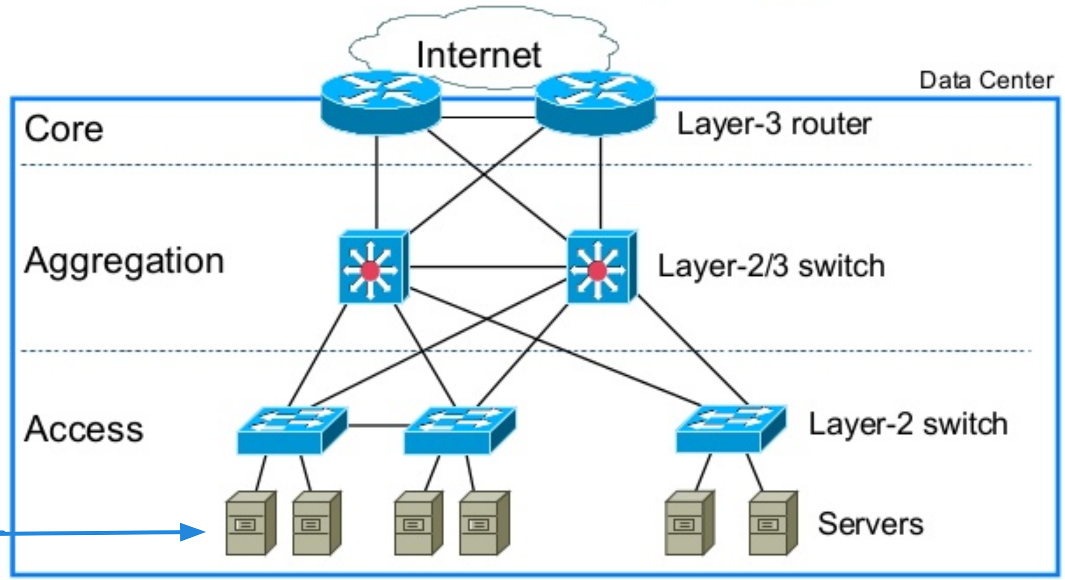
CoreOS

Motivation: Our scope

kubernetes

docker

CoreOS



Challenges

- Can we build container-based middleboxes
  - generalizable
  - scalable
  - correct

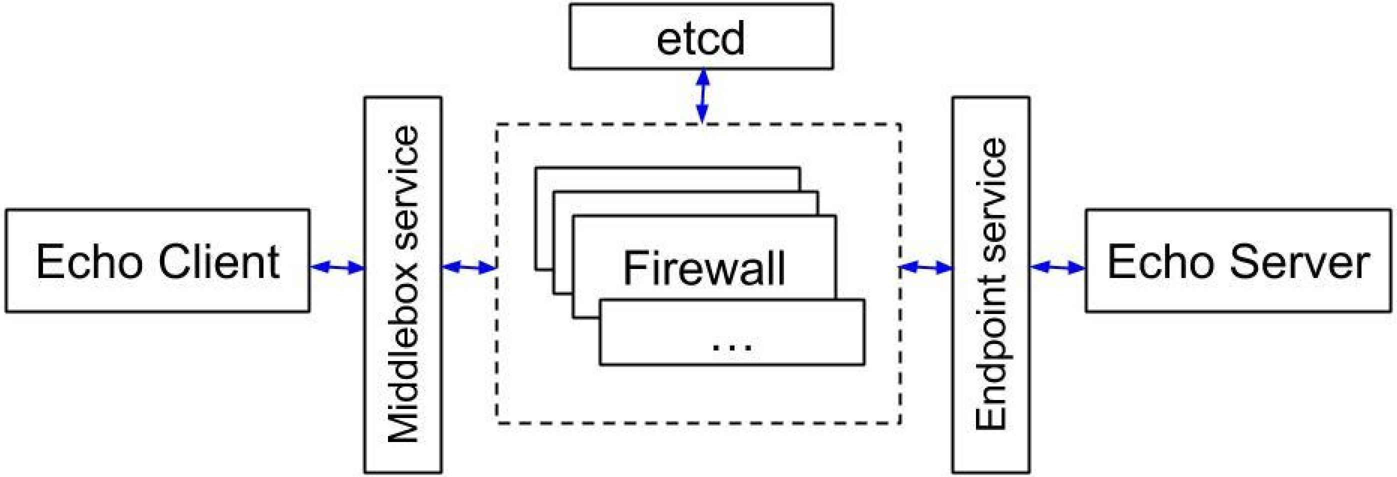
Kubernetes

- Kubernetes’ three abstractions
  - pod
  - replication controller
  - service
- Kubernetes pods are stateless, but most useful middleboxes have a shared state
  - etcd

System Design: sample middlebox

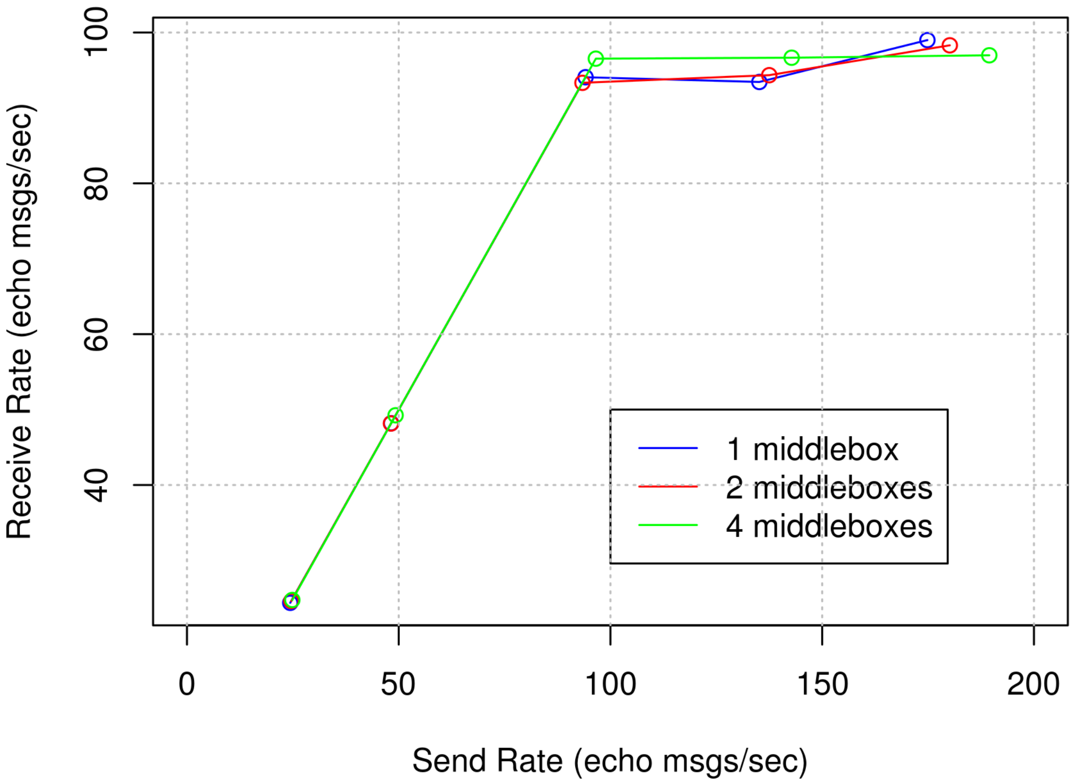
- Rate-limiting Firewall
  - simple shared counter state
  - run as a pod in Kubernetes
  - services as redirection mechanism

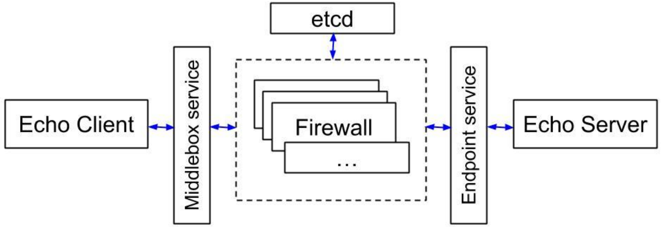
System Design



Results

Middlebox correctness with shared etcd state





etcd

Echo Client

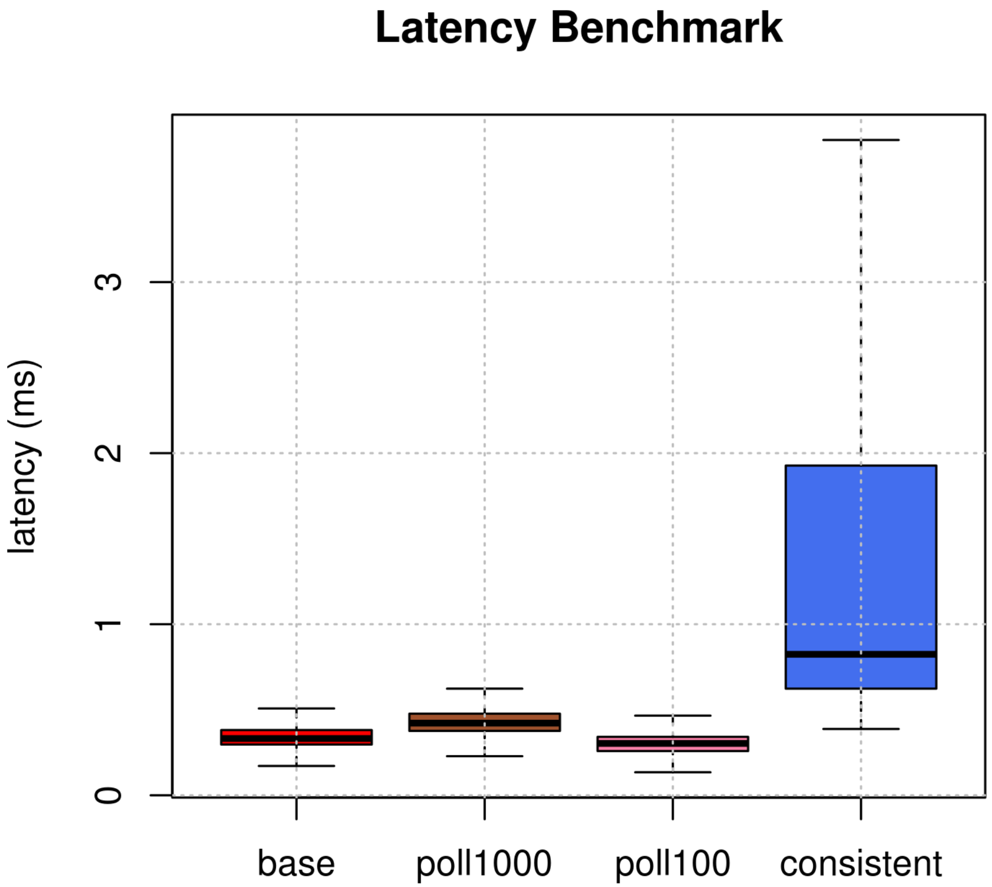
Middlebox service

Firewall

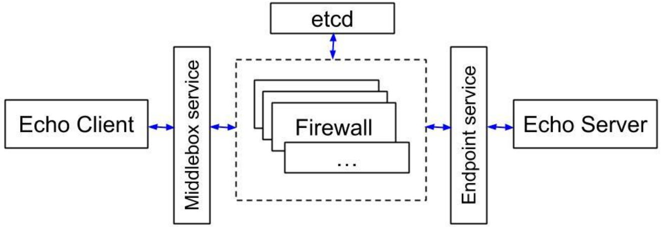
...

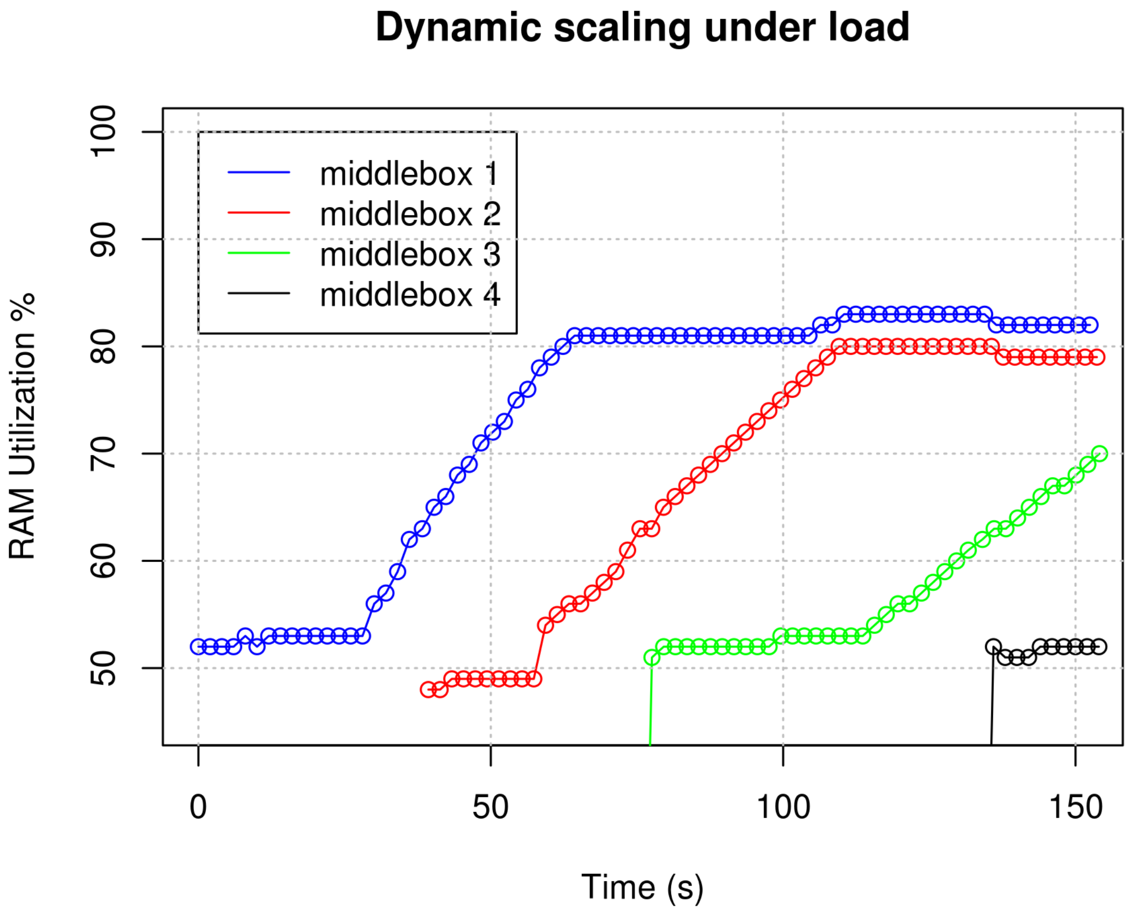
Endpoint service

Echo Server

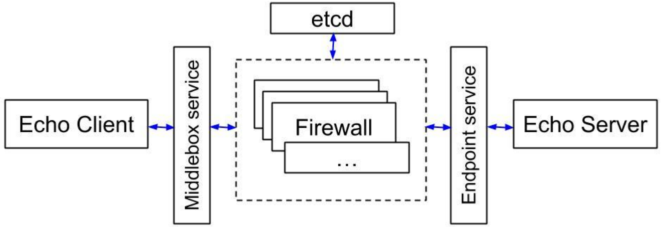


Latency Benchmark





Dynamic scaling under load



Conclusion

- We can implement scalable virtual middleboxes that trade-off between correctness and performance
- We can generalize to other middleboxes (e.g., NAT, VPN, firewall, load-balancers)
- Cloud providers can offer scalable and generalizable middleboxes as a value-add feature

Thank you!