Experimental Evaluation of SDN-Controlled, Joint Consolidation of Policies and Virtual Machines

Wajdi Hajji*  Posco Tso*  Lin Cui†  Dimitrios Pezaros‡

*Loughborough University, UK  †Jinan University, China  ‡Glasgow University, UK
1 Motivation

2 Sync algorithms and system architecture

3 Conclusion

4 Experimental Evaluation
Middlebox challenges
Middlebox challenges

- IPS
- LB
- VLAN1, VLAN2
- Unreachable
- VM Migration

Posco Tso
Sync
Middlebox challenges

- Link overloaded
- Overloaded link
- IPS1
- IPS2
- Policy Migration
- VM Migration
- S1
- S2
- v1
- v2
We have proposed Sync: *Synergistic Policy and Virtual Machine Consolidation in Cloud Data Centers*\(^1\).

Plain language: Sync migrates virtual machines and network policies at the same time.

We have proposed Sync: *Synergistic Policy and Virtual Machine Consolidation in Cloud Data Centers*\(^1\).

Plain language: Sync migrates virtual machines and network policies at the same time.

Does it scale in a real data centre environment?

---

1 Motivation

2 Sync algorithms and system architecture

3 Conclusion

4 Experimental Evaluation
How does *Sync* work? – Sync algorithms

**Get Communicating VM Groups**

The algorithm partitions all VMs into isolated groups in which VMs do not communicate with a VM outside their group. These VM groups will be the input of other algorithms.

**Policy Migration**

This algorithm focuses on migrating the policies, in other words defining again the MBs; replace them with the same type of MBs as the deployed ones.

**VM Migration**

The VM migration algorithm, for a given VM group, initialises and obtains the preference list (where no policy violation or overused server capacity) of all servers.
The topology and the controller communicate through OpenFlow to add rules to switches and via out-of-band control channel.
| Source code | Sync algorithms and system architecture | Conclusion | Experimental Evaluation |

**Source code available on GitHub**

https://github.com/wajdihajji
1 Movitation

2 Sync algorithms and system architecture

3 Conclusion

4 Experimental Evaluation
1 Motivation

2 Sync algorithms and system architecture

3 Conclusion

4 Experimental Evaluation
We have run our experiments on two identical servers, each has 8 Cores/1.2Ghz CPU and 8GB Memory. Both servers have Ubuntu 14.04 is running atop.

In server A, we have installed Mininet version 2.3.0d1, OpenFlow 1.35 and Python 2.7.6.

In server B, we have installed Ryu controller 4.10.

Two servers are connected through a 1Gbps switch.
Sync runtime with growing number of VMs – VM groups

![Graph showing CDF of groups runtime with growing number of VMs: 2k VMs, 6k VMs, 10k VMs. The graph indicates that as the number of VMs increases, the runtime also increases, with the 10k VMs taking the longest runtime.](image-url)
Sync runtime with growing number of VMs – Policy Migration
Sync runtime with growing number of VMs – VM Migration
Sync runtime with growing number of flows – VM groups

CDF of groups vs Runtime (s) for 20k, 60k, and 100k flows.
Sync runtime with growing number of flows – Policy Migration
Sync runtime with growing number of flows – VM Migration

CDF of groups vs. runtime for different flow counts:
- 20k flows
- 60k flows
- 100k flows

Runtime (s) range from 0 to 100 seconds.
Sync runtime with growing number of MBs – VM groups

The graph shows the Cumulative Distribution Function (CDF) of groups as a function of runtime (s) for different numbers of MBs: 20 MBs, 50 MBs, and 80 MBs. The CDF represents the probability that a random sample from the group is less than or equal to a given value. The graph indicates that as the number of MBs increases, the runtime for Sync algorithms also increases, with a higher CDF value at the same runtime compared to smaller MB sizes.
Sync runtime with growing number of MBs – Policy Migration

![Sync runtime with growing number of MBs](image)

- **Motivation**
- Sync algorithms and system architecture
- Conclusion
- Experimental Evaluation

Posco Tso
Sync runtime with growing number of MBs – VM Migration
Conclusions

- The number of VMs has a measurable effect on *Get communicating VM Groups* and *Policy migration* on one hand, and *VM migration* on the other hand.

- The three factors have a different impact on the *Sync* algorithms, flows impacts more *Get Communicating VMs* and *Policy Migration* algorithms, while the number of VMs can significantly alter the time needed by *VM migration* algorithm.

- The number of MBs has a known effect on *Get communicating VM Groups* and *Policy migration*, whereas, in *VM migration*, its impact becomes unpredictable because VM migration decision depends more on policy violation prevention strategy.

- Because of its fractional use of CPU resources, *Sync* is very resource efficient and has room to scale to much bigger topologies.
Thank you! Questions?