An Adaptive Utilization Accelerator for Virtualized Environments

LCCC Workshop in Cloud Control

David Breitgand, Zvi Dubitzky, Amir Epstein, Oshrit Feder, Alex Glikson, Inbar Shapira and Giovanni Toffetti
Cloud Operating Systems Technologies – IBM Haifa Research Lab

What is the problem? VM sprawl in private clouds

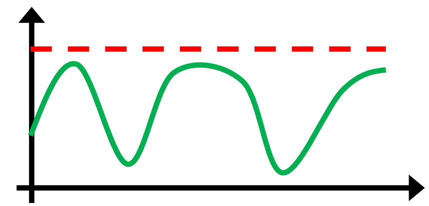
- VM sprawl
  - Proliferation of inactive / unused VMs in clouds
  - Stems from cloud provisioning model (and relative lack of control)
- In the absence of resource utilization models, VM provisioning is based on nominal resource demand

Looking at a VM:

2 VCPUs
4 GB RAM

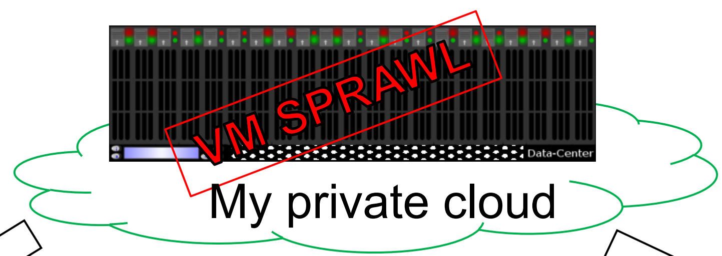
Nominal resource demand

!=

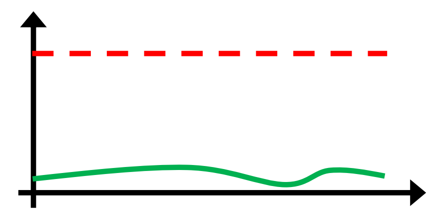


Resource utilization

Looking at a DC:



100%



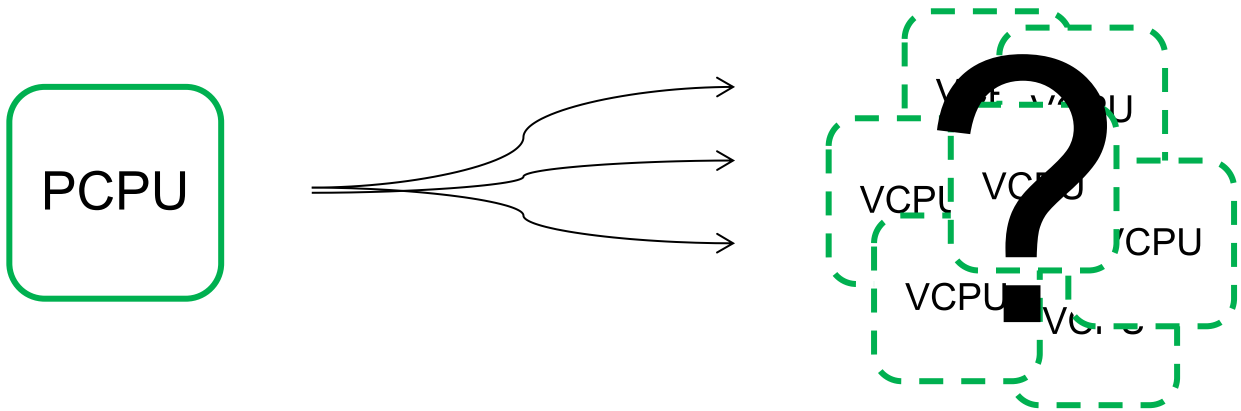
Fully utilized nominal capacity

Low Infrastructure utilization

VM provisioning based on nominal resource demand

Unsatisfied VM placement demand

Common solution: resource over-commit



Over-commit Ratio (OCR) = #VCPUs / #PCPUs

- OCR is a cloud-wide <u>configuration parameter</u>
- Default CPU OCR for Openstack: 16(!)

OCR => infrastructure utilization

OCR => risk of congestion

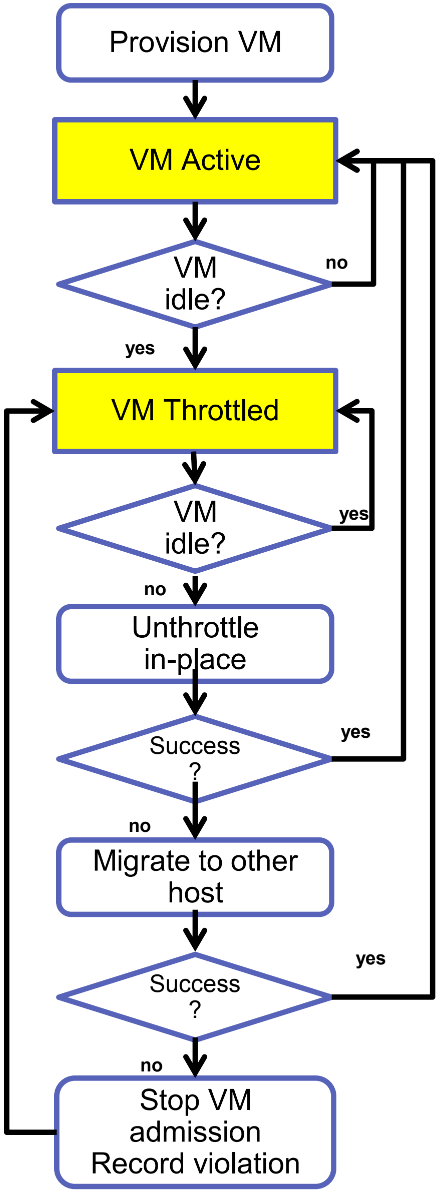
performance degradation

Our proposal – Adaptive over-commit

- RATIONALE:
  - There is **NO "right"** fixed OCR
  - VMs "activity" vs. "idleness" are application-dependent and vary over time
  - Need for automated solution
- GOALS/FEATURES:
  - Increase DC utilization
  - Minimize performance degradation
  - Transparent to VM tenants
  - No assumptions/forecast on VM resource consumption

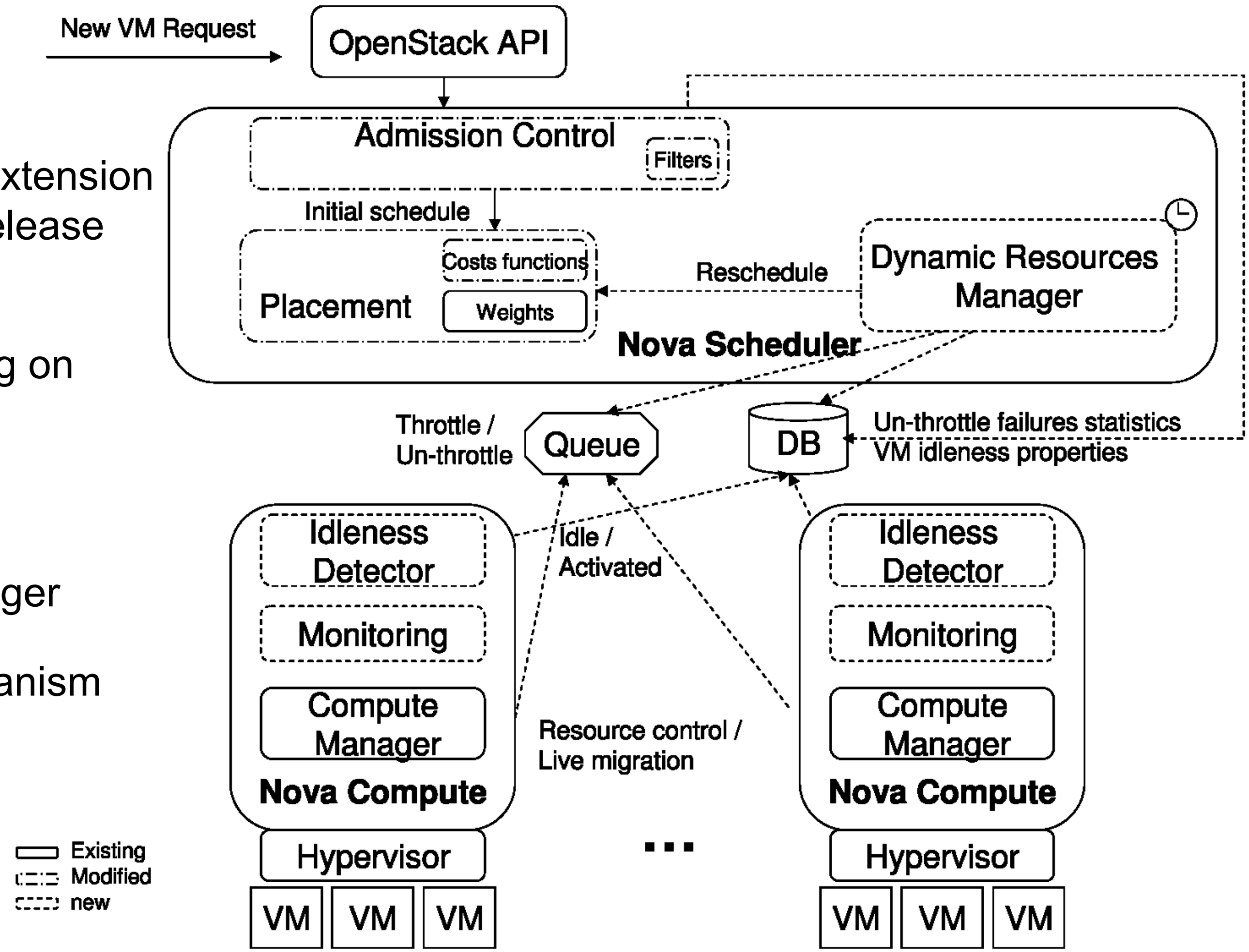
Pulsar high-level functioning

- **IBM Adaptive Utilization Accelerator for Virtualized Environments (PULSAR):**
  - Simple VM idleness detector (CPU util threshold)
  - Claim resources from idle VMs by 'throttling' them (reducing their resource reservation, **cgroups in KVM**)
  - Use **adjusted capacity** (considering throttling) to provision and place more VMs in the system



Implementation

- Full implementation as extension to OpenStack Havana release (soon IceHouse)
- Idleness detector running on each host
- Adjusted capacity filter
- Dynamic resource manager
- Admission control mechanism



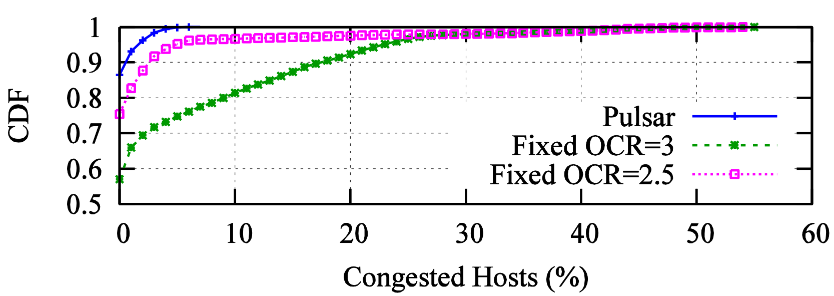
openstack™
CLOUD SOFTWARE

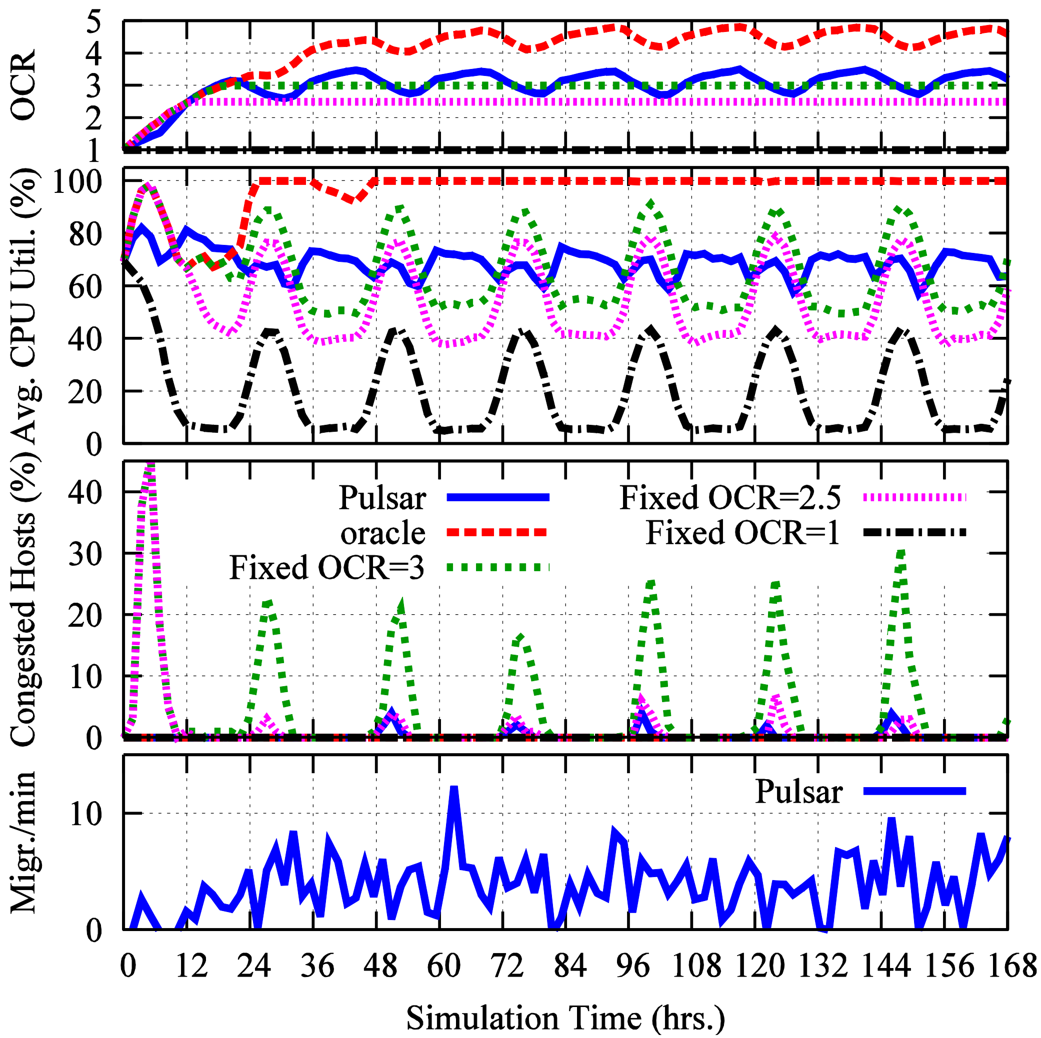
Pulsar evaluation

- Experiments
  - Smaller testbed
  - Full implementation
  - (Synthetic) trace-driven workload: boulders and sand model
  - Measure performance degradation
- Simulations
  - Large testbeds
  - Nova scheduler + testbed emulator (pymoc)
  - Synthetic and real datacenter trace-driven workload
  - Estimate performance degradation through host congestion
  - Compare with theoretical upper-bound “oracle” scheduler

Synthetic workload simulation

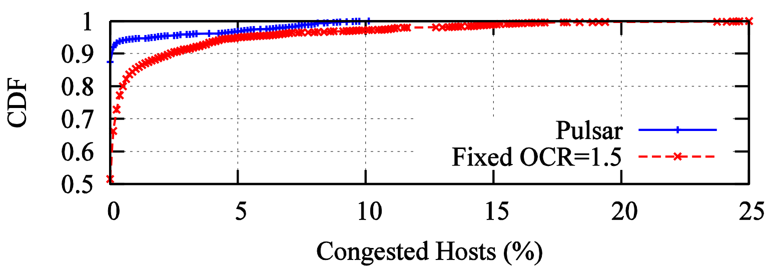
- Based on OpenStack code
- Medium-size scenario
  - 100 hosts, 24 PCPUs each (2400 total cores)
- 1 week synthetic workload using "boulders and sand" model
  - Boulders: long-living VMs with periodic demand pattern
  - Sand: short-lived VMs CPU-intensive jobs (dev-test, map-reduce) Markov-chain demand model
- Compare with fixed OCR (1,2.5,3) and Oracle (theoretical upper bound)



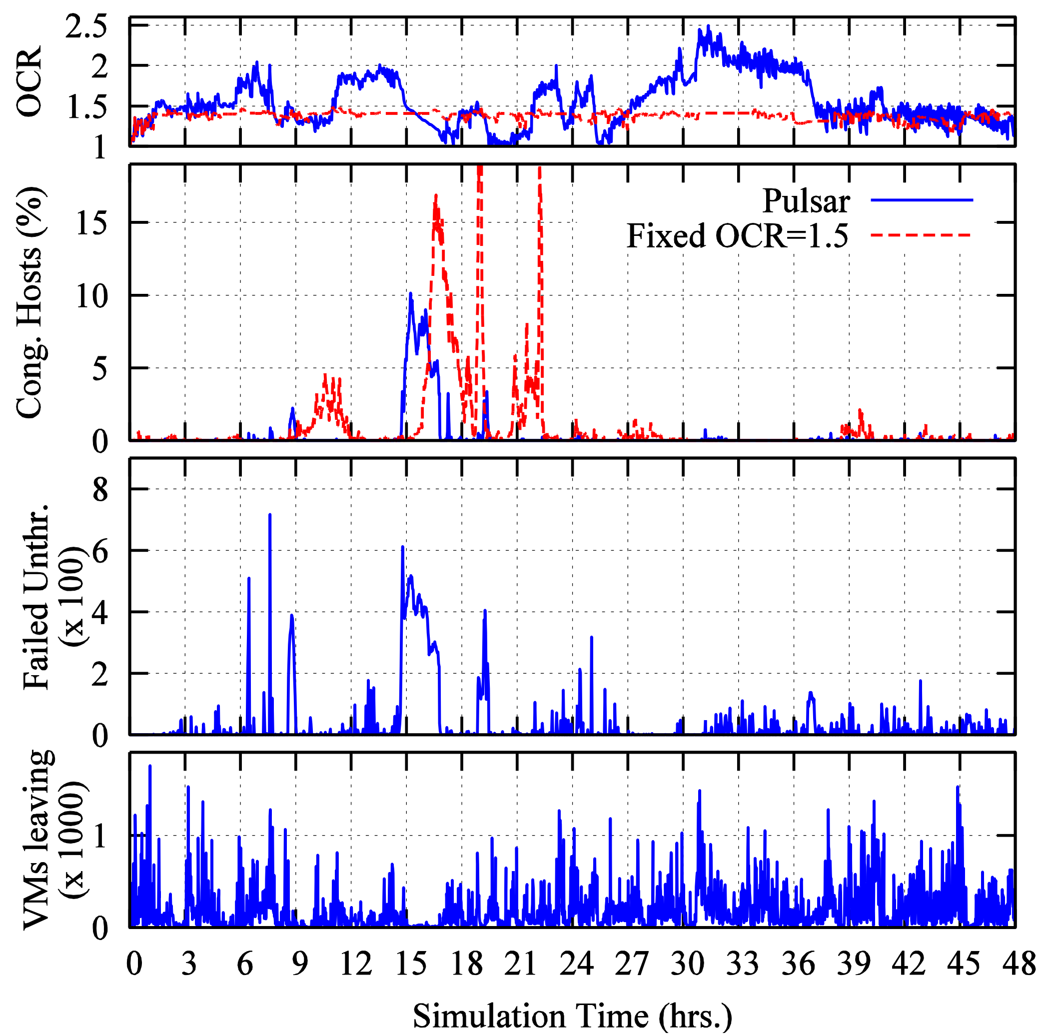


Trace-based simulation

- Pseudo-random sample from Google cluster traces [Wilkes 2011]
- 800 hosts, 48 cores each [38400 cores]
- Pulsar:
  - +20% admitted VMs
  - -50% congestion
- Limits of:
  - Reactive admission control
  - No VM preemption / prorities



10 runs avg.	Fixed OCR 1.5	PULSAR
Total admitted VMs	455k	548k
Avg. Congestion	7.27%	3.12%



Experimental evaluation

- Trace-driven experiment (trace generated with boulder/sand model)
  - Daytrader (DT) Web app [3VCPU, 30min period]
  - Sudoku solver (SD) [1VCPU, 90min avg lifetime, **5% probability of switching btw idle/active each minute**]
  - Very "active" workload, very low maximum achievable OCR [1.5 max from Oracle]
- Testbed
  - Openstack Controller node [Supermicro 8 Xeon E5420 2.5 GHz cores, 8GB RAM]
  - 2 Openstack Compute nodes [IBM System X3550 M3, 24 Xeon X5680 3.3 GHz cores, 28GB RAM]
- Runs (averaged over 20 executions):
  - R1: 4 DTs + 36 SDs (group A), OCR=1
  - R2: PULSAR with group A + SDs from a Poisson process with 2 minutes inter-arrival time (group B)
  - R3: fixed OCR=1.27 (average obtained by Pulsar) groups A+B

Run	DT avg RT (STD) [ms]	SD avg thr (STD) [Hz]	Host 1 avg util [%]	Host 2 avg util [%]
R1	28.8 (6.4)	61.2 (13)	57.8	62.3
R2	34.6 (9.7)	50.25 (14.1)	79.1	80.4
R3	41.6 (11.8)	46.95 (12.85)	85	84

Conclusion

From our evaluation:

- PULSAR is adaptive to changes in resource utilization
- It increases infrastructure utilization
- Limited host congestion
- Limited number of VM migrations
- Outperforming any fixed-OCR solution

Future work:

- Use improved idleness detector / load predictors
- Proactive admission control / VM priorities - preemption
- Larger experiments!

- **Questions?**

Backup slides

Related work

- Many papers on demand prediction for stable VM population [Breitgand 2012] [Chen 2011] [Meng 2010] [Gmach 2007]
  - We consider dynamic VM population, discrepancy between nominal and actual resource usage, adaptive over-commit
- [Gmach 2012] [Yanagisawa 2013] assume static VM population and no overcommit model, use past VM demand patterns to predict future demand
  - We left out prediction of future demand on purpose assuming dynamic VM population, albeit we could leverage this information
- [Carrera 2012] aim at fair placement decision by using a model of expected performance given a resource allocation for each workload
- [Blagodurov 2013] requires application performance monitoring instrumentation and knowledge of resource consumption profiles to classify applications as batch or interactive
- [Wuhib 2012] use average resource utilization over a sliding window to implement different placement policies (e.g., consolidation). The same solution can be applied for adaptive overcommit. However churn and high utilization variation cause number of required migrations to grow quickly

References

- [Wilkes 2011]: John Wilkes, “More Google cluster data”, Google research blog, Nov 2011
- [Breitgand 2012] D. Breitgand and A. Epstein, “Improving Consolidation of Virtual Machines with Risk-Aware Bandwidth Oversubscription in Compute Clouds,” in INFOCOM, 2012.
- [Chen 2011] M. Chen, H. Zhang, Y.-Y. Su, X. Wang, G. Jiang, and K. Yoshihira, “Effective VM Sizing in Virtualized Data Centers,” in IEEE/IFIP IM’11, Dublin, Ireland, May 2011.
- [Meng 2010] X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis, “Efficient Resource Provisioning in Compute Clouds via VM Multiplexing,” in The 7th IEEE/ACM International Conference on Autonomic Computing and Communications, Washington, DC, USA, Jun 2010.
- [Gmach 2007] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, “Workload Analysis and Demand Prediction of Enterprise Data Center Applications,” 2007 IEEE 10th International Symposium on Workload Characterization, pp. 171–180, Sep. 2007.
- [Gmach 2012] D. Gmach, J. Rolia, and L. Cherkasova, “Selling T-shirts and Time Shares in the Cloud,” Cloud and Grid Computing, 2012.
- [Yanagisawa 2013] H. Yanagisawa, T. Osogami, and R. Raymond, “Dependable Virtual Machine Allocation,” in Infocom, 2013, pp. 653–661.
- [Blagodurov 2013] S. Blagodurov, D. Gmach, M. Arlitt, Y. Chen, C. Hyser, and A. Fedorova,“Maximizing Server Utilization while Meeting Critical SLAs via Weight-Based Collocation Management,” in IFIP/IEEE IM’13, 2013.
- [Wuhib 2012] F. Wuhib, R. Stadler, and H. Lindgren, “Dynamic resource allocation with management objectives: Implementation for an OpenStack cloud,” in CNSM’12, 2012, pp. 309–315.