Model-free resource management with Reinforcement Learning

Yue JIN, Makram BOUZID, Dimitre KOSTADINOV, and Armen AGHASARYAN
Causal Data Analytics department
Bell Labs, Paris

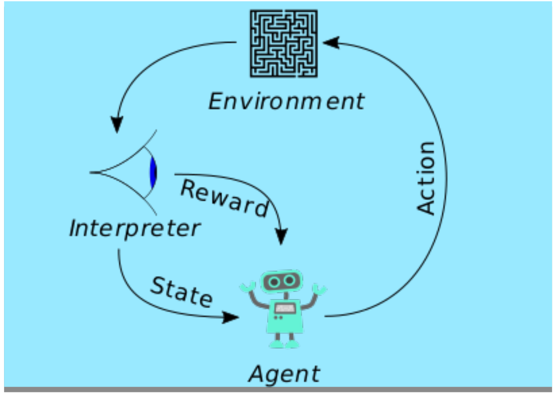
Model-free Resource Management with Reinforcement Learning

Management and Control in Hyper-connected society

The digital system of the future will face the growing challenge of controlling the system behavior in complex dynamically evolving environments

Learning in interaction

Live interaction with a partially observed system to provide it with autonomous management capabilities



Model-free management of Cloud resources

Automated control of Cloud-based applications using a model-free Reinforcement Learning approach

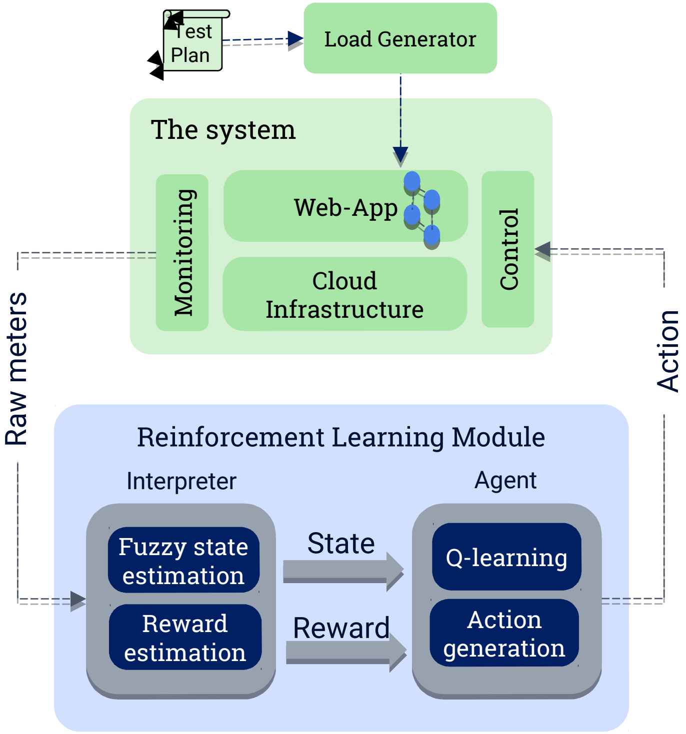
Accelerated Reinforcement Learning for model-agnostic resource management at scale

NOKIA

Reinforcement Learning: Q-learning approach

Automatically learn optimal resource management policies with limited prior information and adapt automatically to changes

Applied to an elastic cloud-based application



Action a:

[+/- Capacity (+/-VMs)]

Q-value update:

$Q(s,a) \leftarrow Q(s,a) + \alpha * error$

$error = [r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$

State s:

[*Workload*; *Capacity (#VMs)*]

Policy : the best action for each state

$\pi(s) = \operatorname{argmax}_a Q(s,a)$

Reward r: the intrinsic desirability of the state

$r = [Revenue - K_1 \cdot Capacity - K_2 \cdot \max(Response\ time - SLA, 0)]$

Experimental results

Automatically derive control decisions that maximize system efficiency

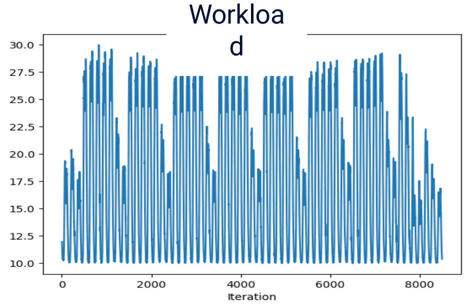
- Minimal system information (state and reward) with no system model
- Trade off between minimizing allocated capacity and maximizing customer responsiveness

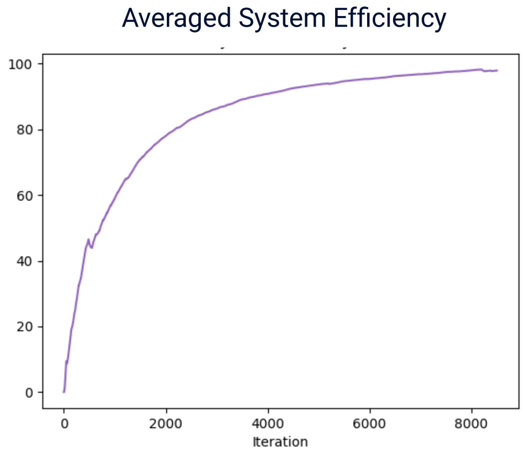
$$r = [Revenue - K_1 \cdot Capacity - K_2 \cdot \max(Response\ time - SLA, 0)]$$

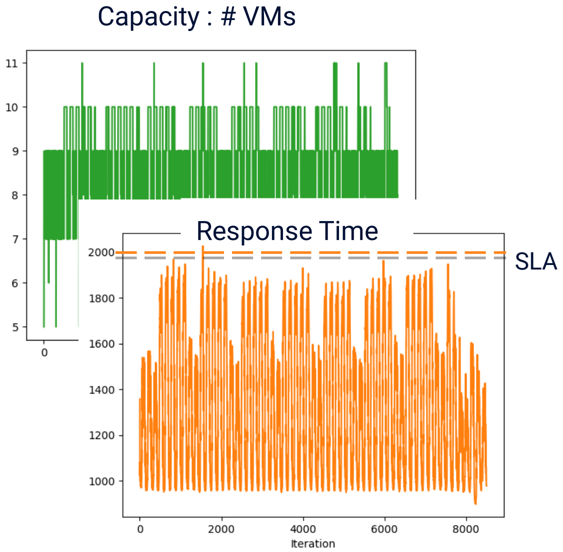
- Convergence of the averaged return (system efficiency)

Return: Sum of discounted rewards

$$R = \sum_{t=0}^{\infty} \gamma^t r_t$$







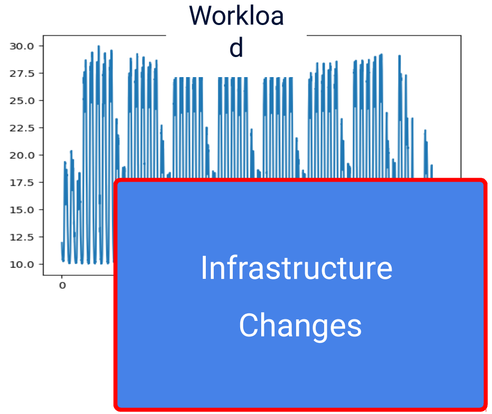
Experimental results (cont'd)

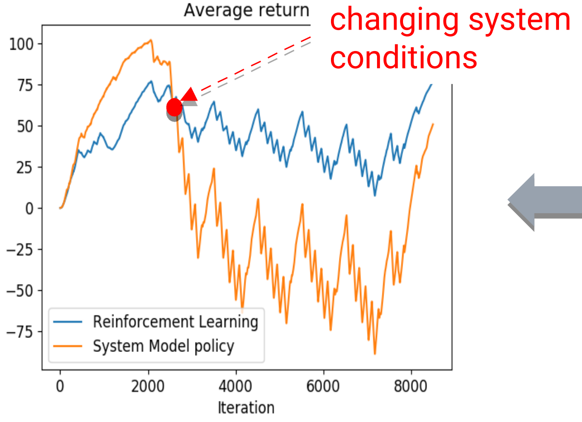
Maximize system efficiency in a <u>changing environment</u>

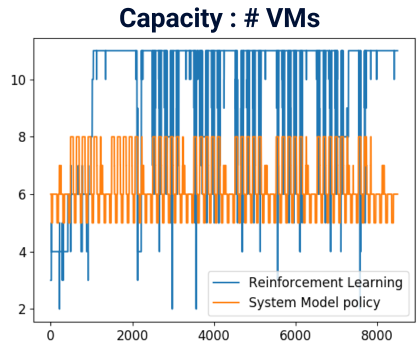
Fit a system model and analytically derive "optimal" policies which maximize the current reward

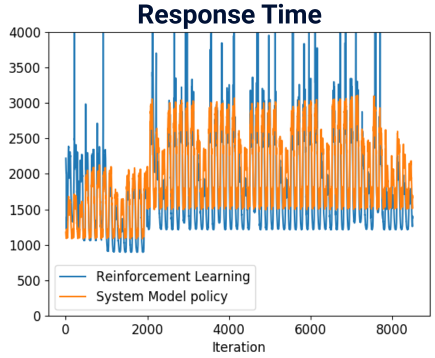
Introduce a system change which invalidates the system model

- CPU limitation, network bandwidth limitation, etc.









Conclusions

Principles for state and reward definition:

- Collect relevant system feedback by separating causes from effects (<workload, capacity> => response time),
- Design a meaningful reward, e.g. reward = revenue – cost, return = log term profit

Automatic derivation and adaptation of control policies

- Development of a simulator for speeding-up the tests (x10^4) and validation within a real Cloud infrastructure
- Demonstrating that RL derived policies beat a system model based approach under changing infrastructure conditions

Open Challenges

- Unknown reward latency
- Failed actions
- Slow convergence (“cold-start”)
- Abnormal environment changes / failures

Understand strength and challenges of Reinforcement Learning for Resource Management

NOKIA

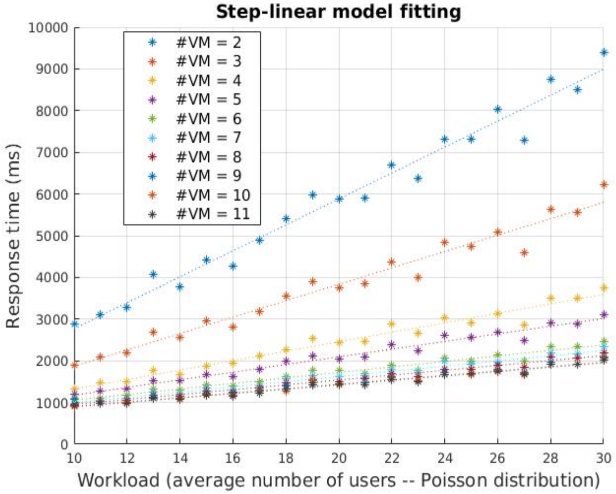
NOKIA

System model fitting and derivation of optimal policies

Derive the system's input-output behavior (Workload vs Response Time)

- Fit a linear model for a given system capacity N_{VM}

$$RT(N_{VM}) = \beta_{0,N_{VM}} + \beta_{1,N_{VM}} \cdot WL$$



- Then, derive scaling policies that maximize the current theoretical