Common Structures in Resource Management as Driver for Reinforcement Learning

MOTIVATION

Growing digitalization is rapidly expanding the complexity in computing and cyber-physical systems. These systems in the future will connect to end-users through billions of sensors and IoT devices, and will rely on cloud-enabled telecommunication networks. Moreover, they will operate in permanently evolving environment. The swollen complexity creates a major challenge for system management and control. Traditional approaches to resource management based on system model specification, off-line behaviour learning and traffic prediction will be increasingly defied due to the emergence of these complex and dynamically evolving systems.

To deal with this challenge, a new paradigm of continuous learning in interaction brings a strong promise for highly adaptive control mechanisms. Starting with little or no knowledge about system characteristics, the control agents start taking actions and learn on-the-fly about their efficiency through the observed feedback from the environment. This approach, referred to as Reinforcement Learning (RL), allows to hide the inherent complexity of the environment and to adapt dynamically to its changing conditions.

However, current RL methods still struggle to learn rapidly in the incremental, online settings for many practical problems. Many iterations are needed before the RL in tabular form converges. Methods based on deep learning and artificial neural networks rely on batch training with large data sets, extensive off-line self-play, or learning asynchronously from multiple simultaneous streams of agent-environment interactions. They don't learn well online. The reason may be that standard RL methods are developed for general problems. They don’t exploit structural properties specific to practical problems they are solving. Exploiting the structural properties and specializing RL methods may be the key to make RL faster and break down barriers for practical application of RL.

MODEL-FREE RESOURCE MANAGEMENT WITH REINFORCEMENT LEARNING

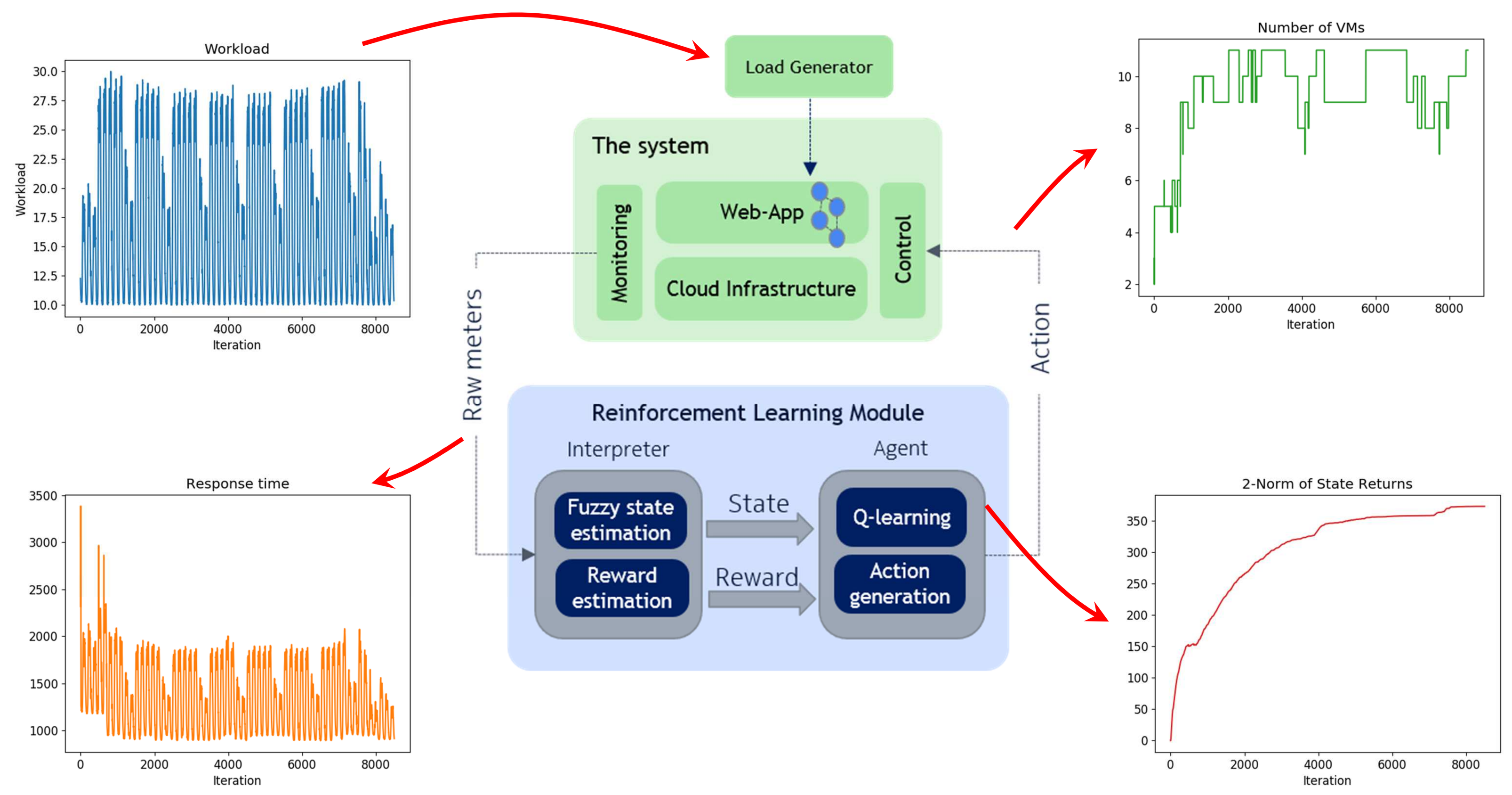
RL for resource management

A RL agent

- Observe the ***state*** of the resource management system: the current workload and the number of virtual machines in use.
- Takes an ***action***: adding or removing up to 2 virtual machines.
- Observe the raw meters of response time and receives the ***reward*** for his action.

His objective is to find a ***policy*** that maps actions to states and maximizes the ***return***, the discounted long-term sum of rewards.

He estimates the return from the rewards he receives by updating the return of the state after receiving the reward



Open challenges

- ***Slow convergence*** (“cold-start”)
- Unknown reward latency
- Failed actions
- Abnormal environment changes / failures
- Distributed RL for multi-dimensional distributed applications resource management

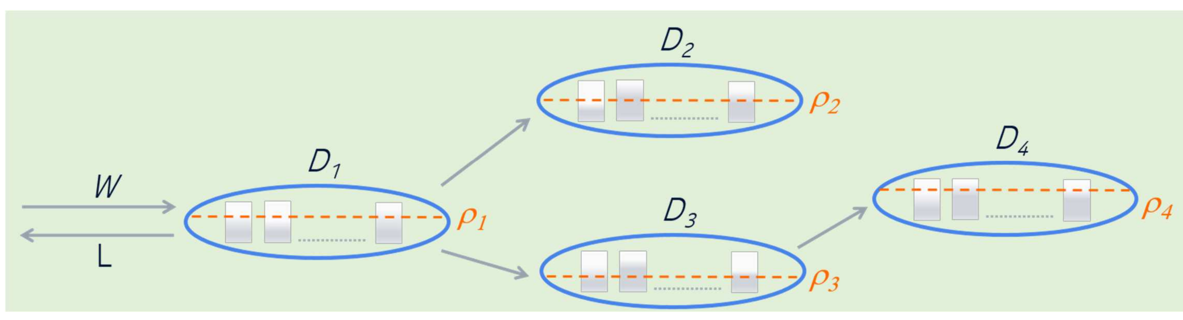
COMMON STRUCTURES IN RESOURCE MANAGEMENT SYSTEMS

Resource management, and more specifically, capacity management systems exhibits many common attributes across different application domains such as Cloud computing, telecommunication, or service systems.

- Incoming ***demands*** usually change stochastically and temporally,
- Demands are processed by ***servers*** with stochastic processing time.
- A higher ***capacity***, i.e. a higher number of servers, reduces the sojourn time of the demands in the system, but increases the cost.
- Depending on the complexity of the systems, the demands may go through multiple processing ***stages*** before leaving the system.

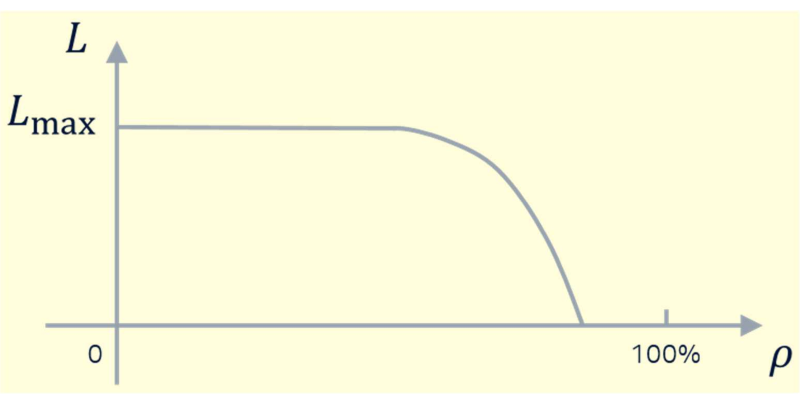
Exploiting these commonalities and specializing RL methods may be the key to make RL faster and break down barriers for practical application of RL.

General Attribute of a Capacity Management system



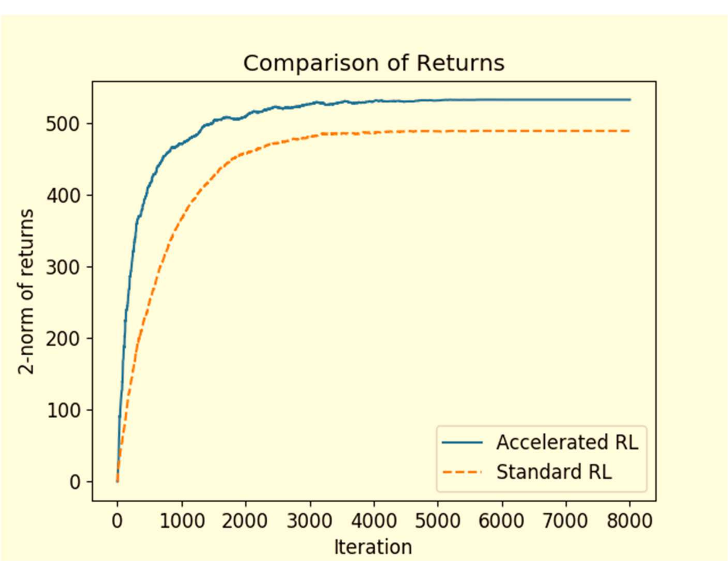
IN-STAGE FUNCTIONAL DEPENDENCIES

Some of the common structural properties are dependencies between attributes at a single processing stage.



Dependency between service level and utilization

- At a given demand, a higher capacity increases the service level and reduces the utilization.
- The rate the service level decreases is often increasing in the utilization.

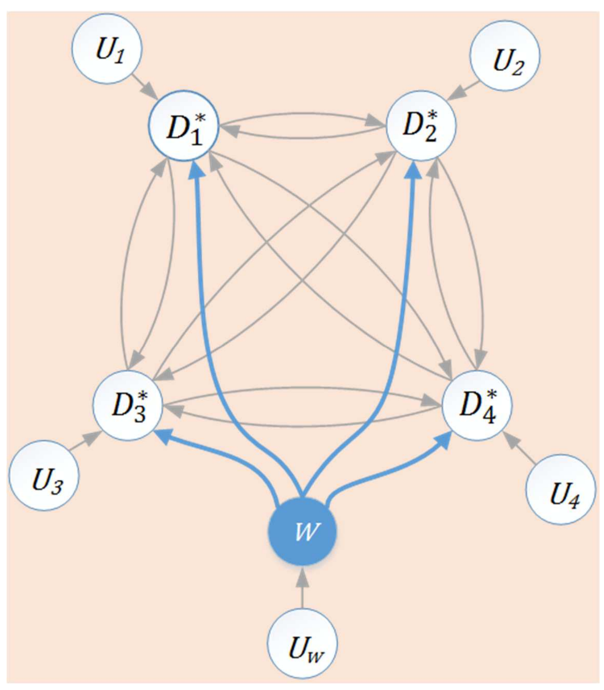


These dependencies can be exploited to derive functional properties between RL concepts. For instance:

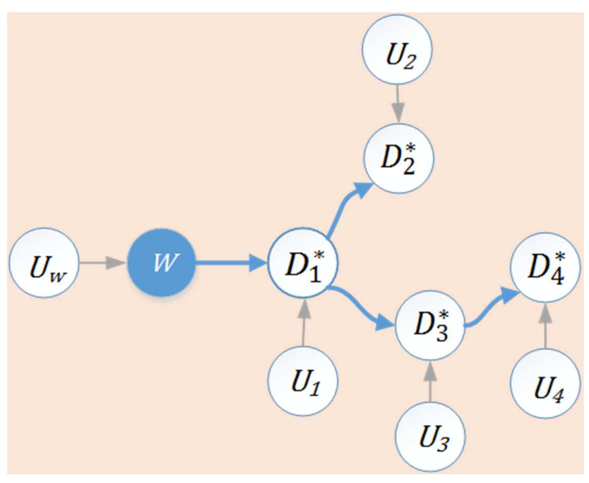
- Under mild conditions, the return can be shown to be monotonic in the state.
- Additional updates of the state returns can be made to accelerate RL

CROSS-STAGE (CAUSAL) DEPENDENCIES

Other structural properties come from topologies of the capacity management systems and emerge as simplified dependencies between attributes at different stages. This simplification can be exploited to accelerate RL.



Dependencies between attributes at different stages when no structure is exploited



Dependencies between attributes at different stages simplified to an acyclic graph by exploiting the topology

Contacts:

Yue Jin, Yue.1.Jin@nokia-bell-labs.com
Makram Bouzid, Makram.Bouzid@nokia-bell-labs.com
Dimitre Kostadinov, Dimitre.Kostadinov@nokia-bell-labs.com
Armen Aghasaryan, Armen.Aghasaryan@nokia-bell-labs.com

NOKIA Bell Labs