Repeatable experiments in the cloud resources management domain with use of Reinforcement Learning

Włodzimierz Funika[1], Paweł Koperek[1], Jacek Kitowski[1,2]

[1] AGH, Faculty of Computer Science, Electronics and Telecommunication, Dept. of Computer Science,
al. Mickiewicza 30, 30-059, Kraków, Poland,
[2] AGH, ACC CYFRONET AGH, ul. Nawojki 11, 30-950, Kraków, Poland

Agenda

- Research motivation
- Proposed solution
- Related work
- Experiment setup
- Results
- Conclusions
- Further work

Research motivation (1/3)

- **Reinforcement Learning combined with Deep Learning - successes in multiple domains**
  - Computer games

    Mnih, Volodymyr & Kavukcuoglu, Koray & Silver, David & Rusu, Andrei & Veness, Joel & G Bellemare, Marc & Graves, Alex & Riedmiller, Martin & K Fidjeland, Andreas & Ostrovski, Georg & Petersen, Stig & Beattie, Charles & Sadik, Amir & Antonoglou, Ioannis & King, Helen & Kumaran, Dharshan & Wierstra, Daan & Legg, Shane & Hassabis, Demis. (2015). **Human-level control through deep reinforcement learning.** Nature. 518. 529-33. 10.1038/nature14236.
  - Game of Go

    Silver, David & Schrittwieser, Julian & Simonyan, Karen & Antonoglou, Ioannis & Huang, Aja & Guez, Arthur & Hubert, Thomas & Baker, Lucas & Lai, Matthew & Bolton, Adrian & Chen, Yutian & Lillicrap, Timothy & Hui, Fan & Sifre, Laurent & van den Driessche, George & Graepel, Thore & Hassabis, Demis. (2017). **Mastering the game of Go without human knowledge.** Nature. 550. 354-359. 10.1038/nature24270.
  - Robot control

    Gu, Shixiang & Holly, Ethan & Lillicrap, Timothy & Levine, Sergey. (2017). **Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates**. 3389-3396. 10.1109/ICRA.2017.7989385.
- **Attempts to use the Deep Learning techniques for Reinforcement Learning (Deep Reinforcement Learning) for cloud resources management**
  - Power management in data center

    Yuanlong Li & Yonggang Wen & Kyle Guan and & Dacheng Tao (2017), **Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning**, arXiv 1709.05077
  - Cloud resources management

    Hongzi Mao, Mohammad Alizadeh, Ishai Menache, Srikanth Kandula (2016), **Resource Management with Deep Reinforcement Learning** In Proceedings of the 15th ACM Workshop on Hot Topics in Networks - HotNets XV, pp. 50-56, doi:10.1145/3005745.3005750

Research motivation (2/3)

- **Reproducing results in cloud resources management is not trivial**
  - Costs of infrastructure
    - Required to train the model
    - Used as training environment
  - Models hard to replicate:
    - Different interfaces
    - Different Deep Learning frameworks
    - Different assumptions (e.g. available actions)
- **Sample architecture**
  - Every element is custom made

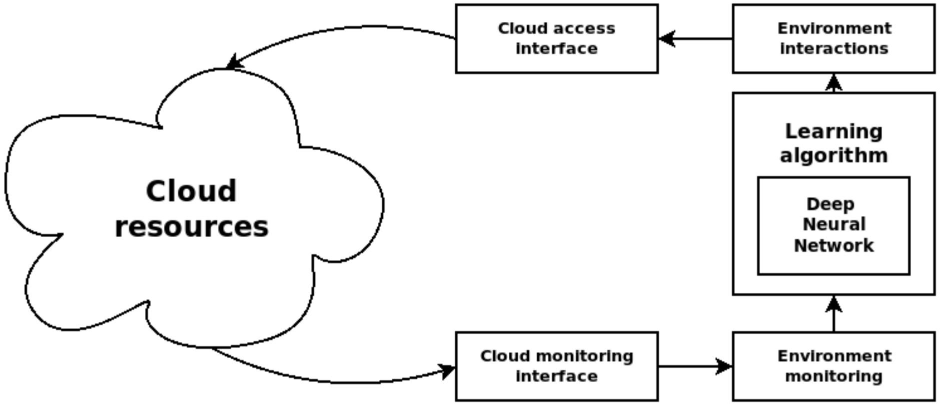

Fig. 1. Sample architecture of Reinforcement Learning managing a cloud application

Research motivation (3/3)

Problem statement

– Is it possible to apply Deep Reinforcement Learning to optimize cloud resources management of a specific application?

– How to make the results comparable and easy to reproduce?

Proposed solution (1/2)

- **Solution features**
  - Leverage existing framework with *uniform interfaces* for different environments
    - Extend widely adopted Open AI Gym
  - Provide a foundation for *reusable cloud environments* and *repeatable* research
    - Use simulation as the environment for training
      - Simplifies training infrastructure setup
      - Guarantees reproducibility of results
      - Lower training costs
      - Training time speed up
      - Faster research through parallelization of experiments
  - Simplify setting up the infrastructure and *speed up the research process*
    - Ready-to-use deployment descriptors for `docker-compose` and `kubernetes`
      - *https://hub.docker.com/r/pkoperek/dqn-manager/*
      - *https://hub.docker.com/r/pkoperek/cloudsimplus-gateway/*
- **Open Source source code**
  - https://github.com/pkoperek/gym_cloudsimplus
  - https://github.com/pkoperek/cloudsimplus-gateway
  - Sample DQN model accessing the simulation: https://github.com/pkoperek/dqn_cloudsimplus

Proposed solution (2/2)

- **System elements**
  - Sample RL model - Deep Q Network
    - Uses Open AI Gym to access simulation
    - Performs training
    - Separated from simulation – enabled to deploy to separate compute resources
  - Open AI Gym
    - Common, uniform interface
    - New CloudSimPlus environment – proxy to Cloud Sim Plus Computation Process
    - Exposes easy to understand methods: *step, render, reset, close*
    - Available actions: create, remove VM, do nothing
    - New actions can be added with backward compatibility
  - Cloud Sim Plus Computation Process
    - Each step corresponds to a second of data center time
    - Pre-configured set of jobs stored in Standard Workload Format
      - http://www.cs.huji.ac.il/labs/parallel/workload/swf.html

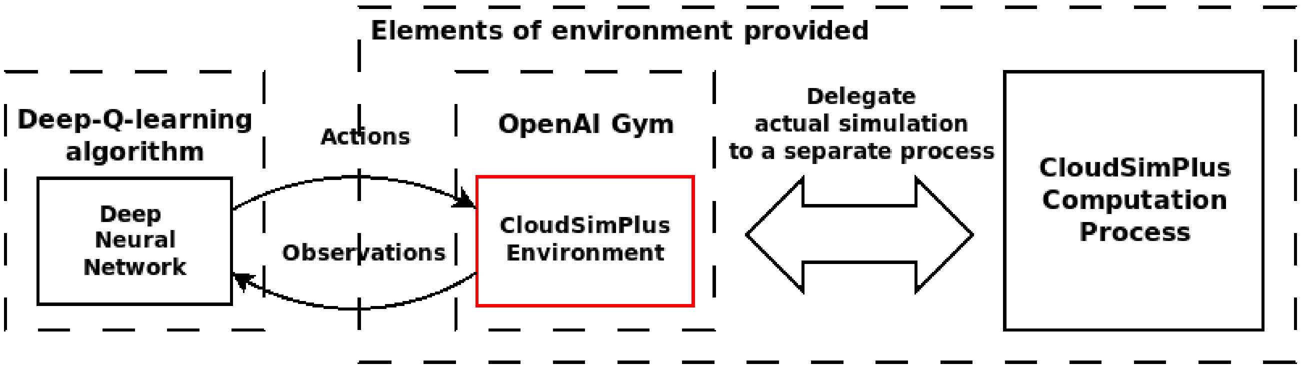

Fig. 2. Deep-Q-learning interacting with a CloudSim Plus simulation through Open AI Gym interface.

Related work (1/2)

- **Reinforcement learning**
  - Focuses on **agents** which take **actions** in an **environment** to maximize **reward**
  - Active research topic for long time

    Kaelbling, L. P., Littman, M. L., Moore, A. W., 1996. **Reinforcement learning: A survey**. Journal of artificial intelligence research 4, 237–285.

    Sutton, R. S., Barto, A. G., 1998. **Introduction to reinforcement learning**. Vol. 135. MIT press Cambridge.
  - Multiple algorithms available
    - Online and offline
    - Model-based and model-free (e.g. Q-learning)
- **Recently combined with Deep Learning:**
  - Deep Q Networks (DQN)

    Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D. & Riedmiller, M. (2013). **Playing Atari with Deep Reinforcement Learning** (arxiv:1312.5602, NIPS Deep Learning Workshop 2013)
  - Asynchronous advantage actor-critic (A3C)

    Mnih, Volodymyr & Puigdomènech Badia, Adrià & Mirza, Mehdi & Graves, Alex & Lillicrap, Timothy & Harley, Tim & Silver, David & Kavukcuoglu, Koray. (2016). **Asynchronous Methods for Deep Reinforcement Learning.**

Related work (2/2)

- **Open AI Gym**
  - „*Toolkit for developing and comparing reinforcement learning algorithms*" (https://gym.openai.com)
  - Provides environments with uniform access interface
    - Atari games
    - Physics engine
    - Algorithmic
  - Open Source project

    Greg Brockman and Vicki Cheung and Ludwig Pettersson and Jonas Schneider and John Schulman and Jie Tang and Wojciech Zaremba (2016), **Open AI Gym**, arxiv: arXiv:1606.01540
- **Cloud Sim Plus**
  - „*CloudSim Plus is a full-featured, highly extensible simulation framework enabling modeling, simulation, and experimentation of Cloud computing infrastructures and application services."* (http://cloudsimplus.org/)
  - Support for:
    - Large scale network topologies
    - Data centers
    - Virtualized server hosts with customizable provisioning policies
  - Open Source project

    M. C. Silva Filho, R. L. Oliveira, C. C. Monteiro, P. R. M. Inácio, and M. M. Freire. **CloudSim Plus: a Cloud Computing Simulation Framework Pursuing Software Engineering Principles for Improved Modularity, Extensibility and Correctness**, in IFIP/IEEE International Symposium on Integrated Network Management, 2017, p. 7.

Experiment setup (1/2)

- **Objective**
  - Validate correctness of running an experiment in the environment
  - Create a dynamic resource allocation policy using Deep Reinforcement Learning to optimize resource use in a sample scenario
- **Simulated environment**
  - Single datacenter with 1000 hosts
  - Single host can run a single VM
    - VM configuration: 4 core processor with 16 GB of RAM (*.xlarge Amazon EC2 instances)
  - Initial number of Virtual Machines was pre-configured
  - Running until no more tasks left to process
  - Metrics: number of running VMs, P99 latency, P90 latency, Avg CPU utilization, P90 CPU utilization, queue wait time
- **Simulated workload**
  - The Swedish Royal Institute of Technology (KTH) IBM SP2
  - Target system: 96-node IBM SP2 (POWER2 66MHz CPUs, Top500 rank 64 as of 06/1996)
  - Number of batch processing jobs: 28,490
  - Variable time of execution and required resources
  - Duration: October 1996 thru August 1997
  - Time flow speed increased 1000x
- **Reward function**
  - Negative cost of running the infrastructure including the SLA penalties
  - Cost of machines: flat $0.2 per hour of virtual machine work
  - $0.00001 penalty for a second of delay in task execution
- **Focus on CPU and RAM resources, ignore storage and network**

Experiment setup (2/2)

- **Reinforcement Learning model compared to cost of manually tuned infrastructure**
- **Sample Deep Neural Network used to evaluate RL approach**
  - 3 1D Convolution layers (number of kernels: 6, 16, 32, kernel size: 5)
  - Convolution followed by Batch Normalization
  - ReLU activation function
  - Final layer fully connected
  - Attempt to replicate model used in:

    Z. Wang, C. Gwon, T. Oates and A. Iezzi: **Automated Cloud Provisioning on AWS using Deep Reinforcement Learning**, 2017, arXiv: 1709.04305
- **Compared the total reward obtained through dynamic resource allocation to static, pre-configured number of VMs**
  - In both cases the same interface to the simulated environment was used

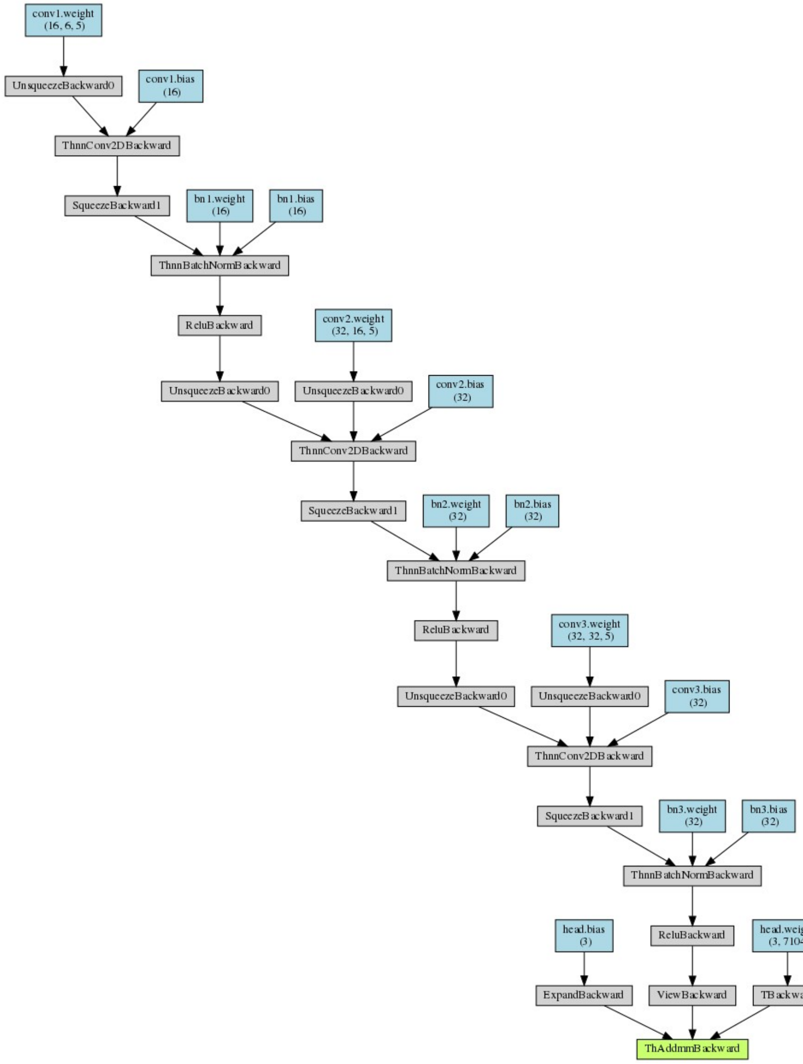

Fig. 3. DNN model visualization

Results

- **Single simulation of processing all jobs on static infrastructure executed within approx. 39 minutes.**
  - Dynamic resource changes significantly slow down the process (15-20x – depending on actions taken by the model)
- **Dynamic resource allocation by DQN model achieved better results comparing to statically configured pool of resources**
  - Total cost improved **4,07x** (comparing to best static configuration)
  - Total wait time improved **2,77x** (comparing to best static configuration)
- **Observed flexibility of the training method**
  - Model discovered and exploited bugs in simulation implementation

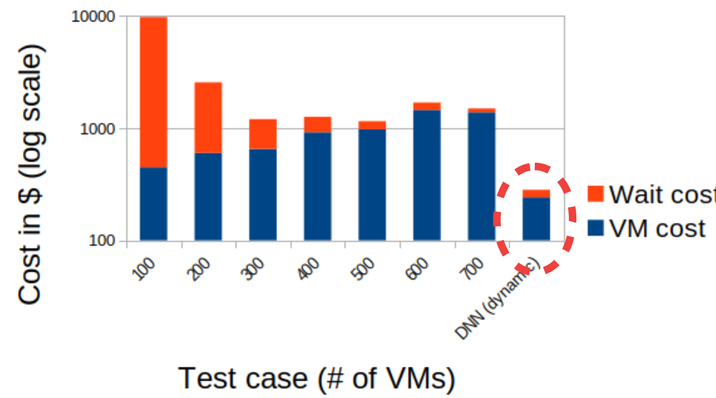

Fig. 4. Total reward in experiment test cases

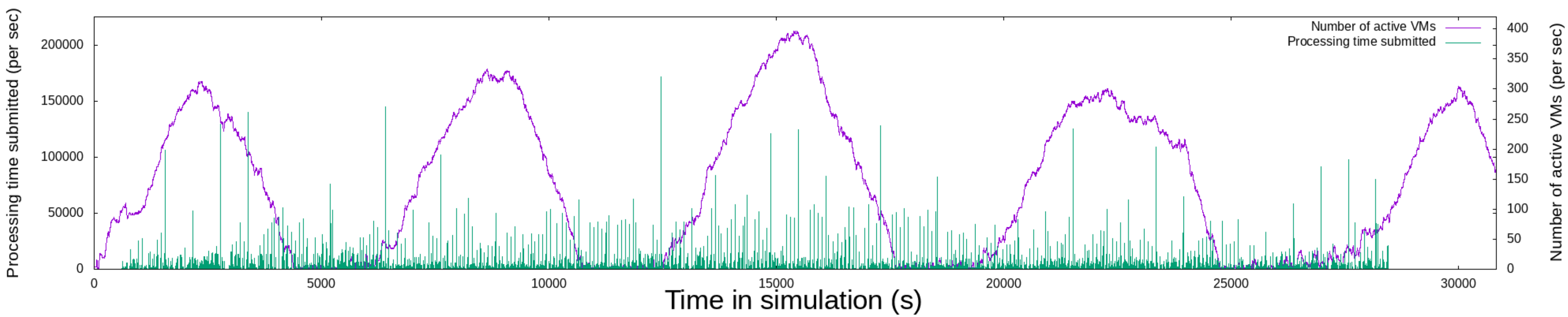

Fig. 5. Simulation progress

Conclusions

- **Contribution**
  - Cloud infrastructure simulation environment accessible through a *simple, uniform interface*
  - *Reusable infrastructure* enabling focusing on research
  - *Reproducible results* of running specific policy in simulation
- **Low infrastructure costs of the training environment**
- **Sample experiment: DQN improvement over static configuration**
  - Improved aggregated cost 4x while simultaneously reducing wait time by 2,8x.

Further work

- **Extend the approach under discussion**
  - Include more elements of real-world environments in the simulation, e.g.:
    - Multi-datacenter setups
    - Storage limitations
    - Network limitations
  - Introduce additional monitoring metrics as description of state, e.g.:
    - Amount of free/used RAM
    - Free storage
    - Free network bandwidth
  - Improve how reward function represents real cloud pricing models, e.g.:
    - Include various machine types
    - Bill CPU time per each started hour
  - Optimize simulation in dynamic resource allocation scenario

Thank you!