On Optimizing Operational Efficiency in Storage Systems Via Deep Reinforcement Learning

SAMSUNG SDS AMERICA

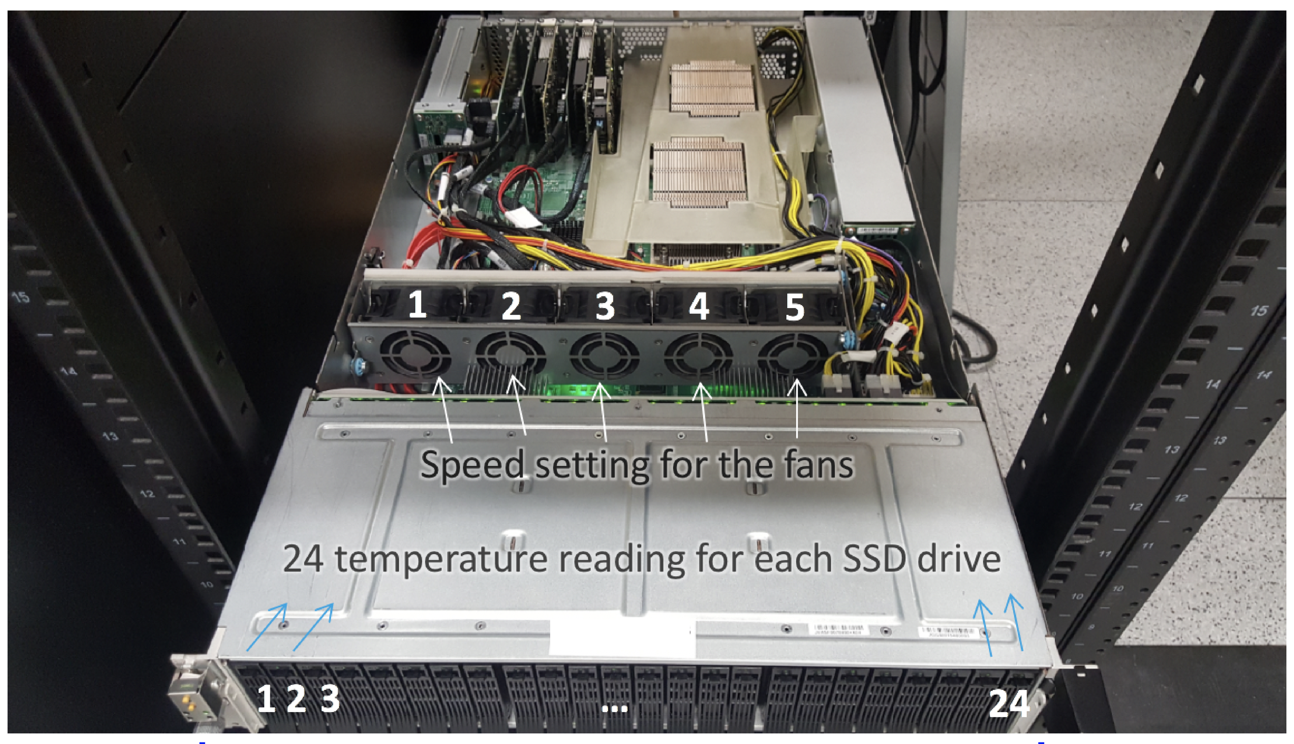
Sunil Srinivasa*, Girish Kathalagiri*, Julu Subramanyam Varanasi**, Luis Carlos Quintela*, Mohamad Charafeddine* and Chi-Hoon Lee*

*Samsung SDS Research America, **Samsung Semiconductors Inc.

{ssunil, girish.sk, sv.julu}@gmail.com, {l.quintela, mohamad.c, lee.chihoon}@samsung.com

SAMSUNG SEMICONDUCTORS

Problem Setup



24 SSDs solid state drives for storage

A Samsung SSD Storage Rack

- Rack temperature varies based on the workload: read or write (how many KB, and how often).
- High temperatures lower the performance of reading/writing and shorten the lifespan of the SSDs.
- Almost impossible to use rules in order to model the dynamics of workload and temperature.
- Use **deep reinforcement learning** to represent the state of operation and learn the optimum policy for controlling the fan speeds, as a proxy of energy usage for cooling.

Contributions

- **Model-free approach**, no need to understand to SSD server dynamics of workload and temperature
- We trained on the **real environment** (the server), not through a simulator
- We designed a **reward function** for this problem to guide the algorithm towards the desired operation behavior (servers temperature and fans speeds)

States, Actions and Rewards

State =

1. Γ(**# of I/O requests** to the server **per second**)
2. Γ(**# of Kbytes read**, averaged over all SSDs per sec)
3. Γ(**# of Kbytes written**, averaged over all SSDs per sec)
4. Γ(**# of Kbytes read in previous time slot**, averaged over all SSDs)
5. Γ(**# of Kbytes written in previous time slot**, avg. over all SSDs)
6. Γ(**Mean Temperate**: averaged over all SSDs)
7. Γ(**Mean Fan Speed**: averaged over all 5 cooling fans)

Read via `iostat` (1–5)

Read via `ipmi-sensors` (6–7)

Where normalization operation per field is $\Gamma(x) = \frac{x - minX}{maxX - minX}$

Raw Actions

Size of Actions space = 7

A = {6, 8, 10, .., 18}

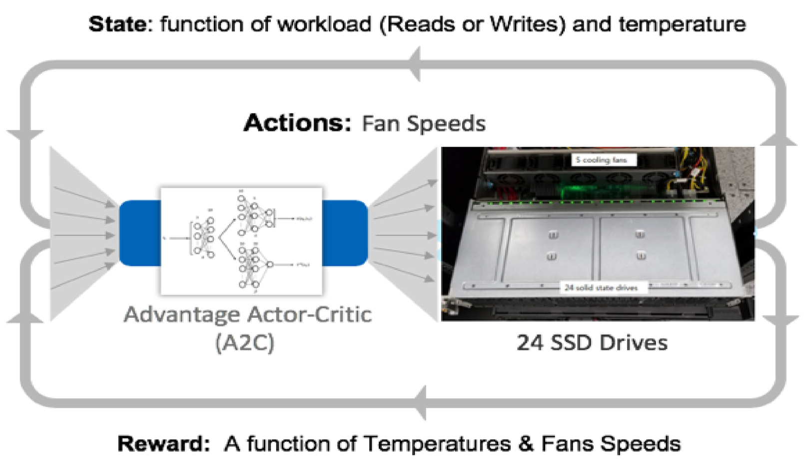
Kilo rpm (revolutions per minute)

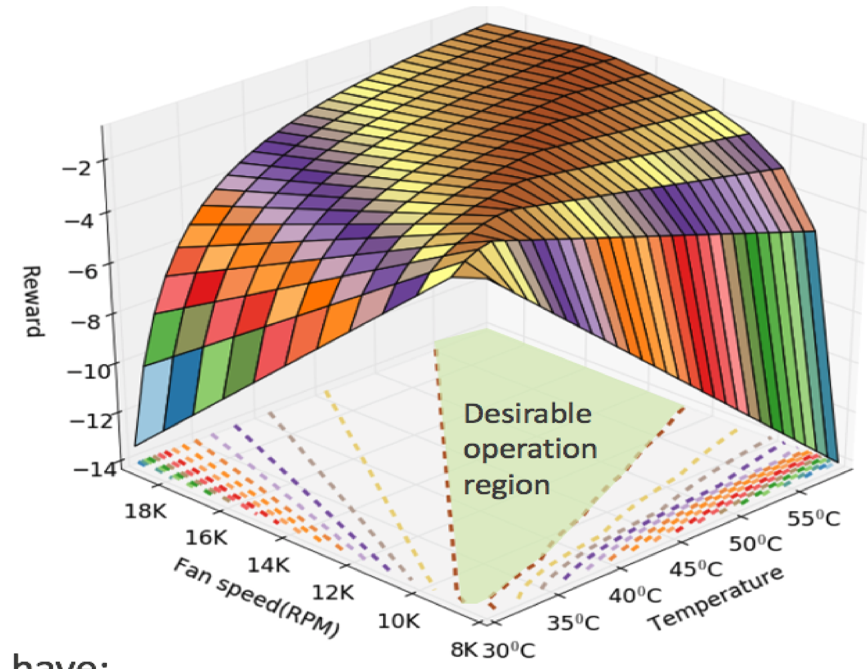
Incremental Actions

Size of Actions space = **3**

A = {Decrease by 1000, No change, Increase by 1000} rpm

Allows for smoother transitions





Reward shaping is an important design step which requires domain knowledge and guides the RL agent towards its optimal behavior

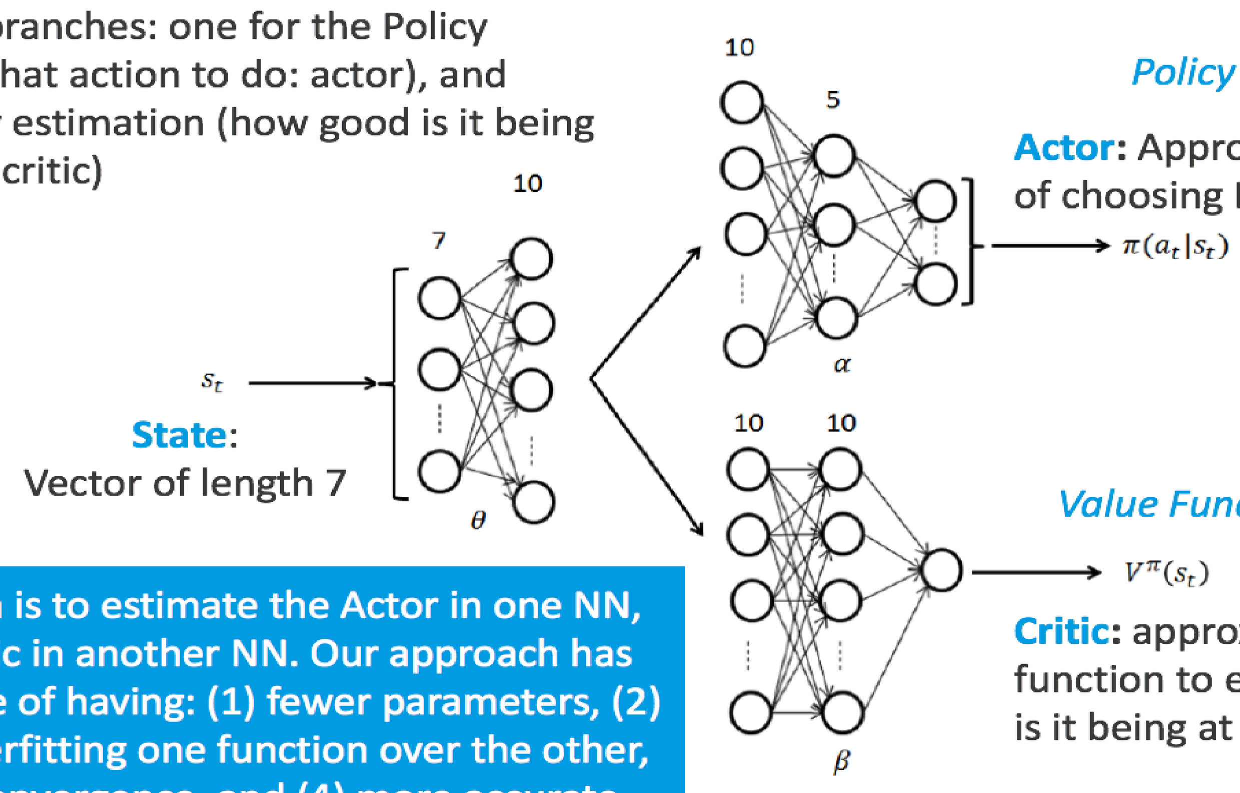
For this RL application, a good reward function should have:

1) High reward for low fan speeds and low temperature
2) Low reward for high fan speeds and low temperature – as this is wasting energy
3) Low reward for low fan speed and high temperature – as this will damage the SSDs

$$R = -max\left(\frac{\Gamma(T)}{\Gamma(F)}, \frac{\Gamma(F)}{\Gamma(T)}\right)$$

We chose this design above.
T is the mean temp across all SSDs.
F is the mean speed across fans.

A NN with 2 branches: one for the Policy evaluation (what action to do: actor), and one for policy estimation (how good is it being in such state: critic)



Policy Neural Network

Actor: Approximates optimal policy, of choosing Best action for state ***s***

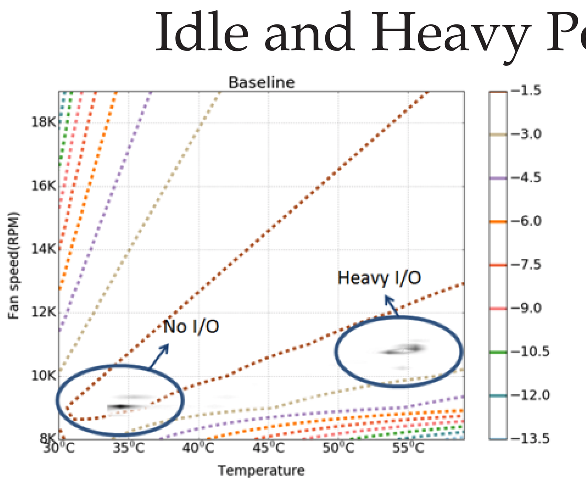
Value Function Neural Network

Critic: approximates the state-value function to estimate how attractive is it being at state ***s***

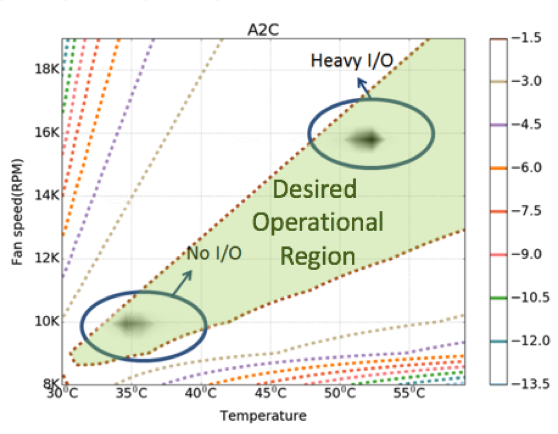
Other option is to estimate the Actor in one NN, and the Critic in another NN. Our approach has the advantage of having: (1) fewer parameters, (2) mitigates overfitting one function over the other, (3) faster convergence, and (4) more accurate approximations for the actor & the critic.

Experiment Results (Baseline vs A2C)

Idle and Heavy Periodic Load

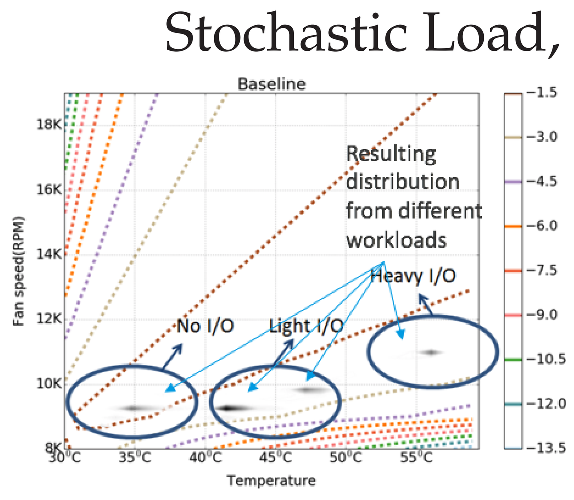


Status Quo controller

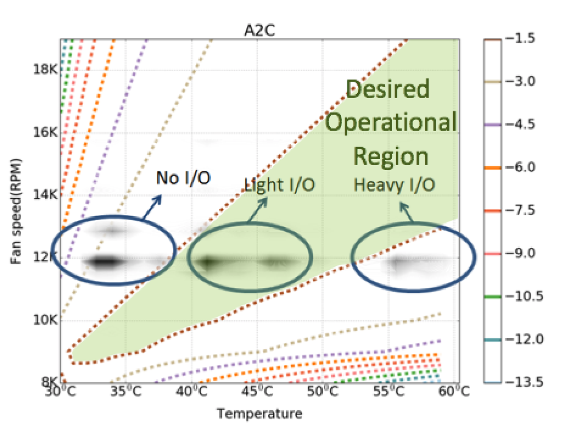


Using Deep Reinforcement Learning with Raw Actions

Stochastic Load, Raw Actions

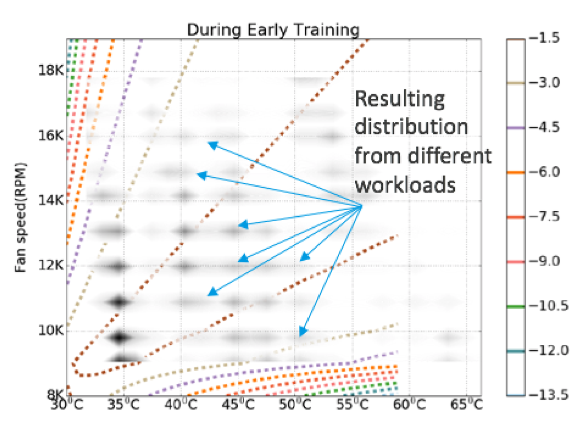


At the beginning of training, algorithm is exploring and learning

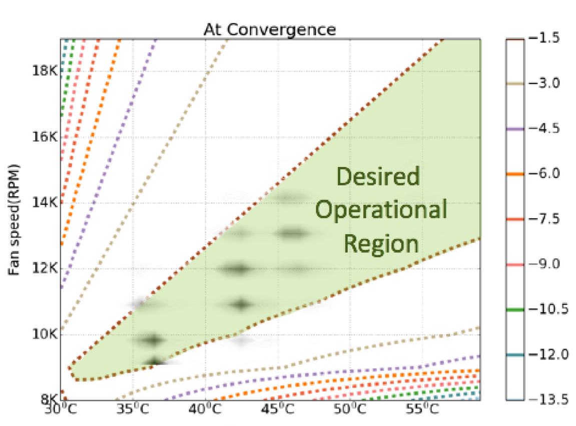


Using Deep Reinforcement Learning with Raw Actions – Suboptimal performance, likely due to insufficient exploration

Stochastic Workload, Incremental Actions



At the beginning of training, algorithm is exploring and learning



Once finished learning right policy, operational behavior is within desired region. Used DRL with incremental actions.