Parallel Reinforcement Learning
Parallel Design

team07: Sam DePaolo, Michael Kielstra, Manqing Liu, Xiaohan Wu

Refresher: Reinforcement Learning

- At each time t, the agent receives current **state** S_t and **reward** R_t
- The agent then choose **action** A_t
- The action is sent to the environment , which moves to a new state S_{t+1} and reward R_{t+1}
- Goal is to learn a **policy** π which maximizes the cumulative reward

$$\pi(a, s) = Pr\left(A_t = a \mid S_t = s\right)$$

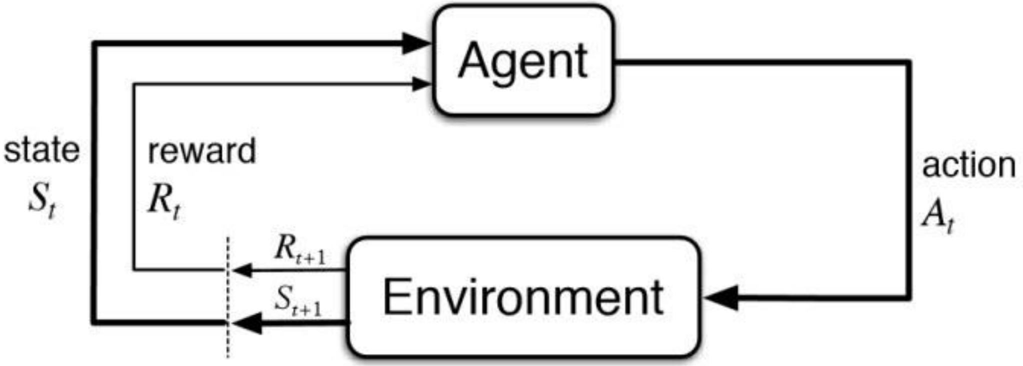


Image credit: https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html
Reference: https://en.wikipedia.org/wiki/Reinforcement_learning

Sequential Baseline Results

- Learning performance of a single RL agent with a 10-armed bandit
- We use an ε-greedy policy (ε = 0.1) to average 500 different experiments where each contain 100 trials
- n=10 arms are created randomly from N(1.0, 1.0)
- As the agent gains more **experience**
  - Reward for each arm approaches the **true mean**
  - The agent is more likely to select the **optimal action**

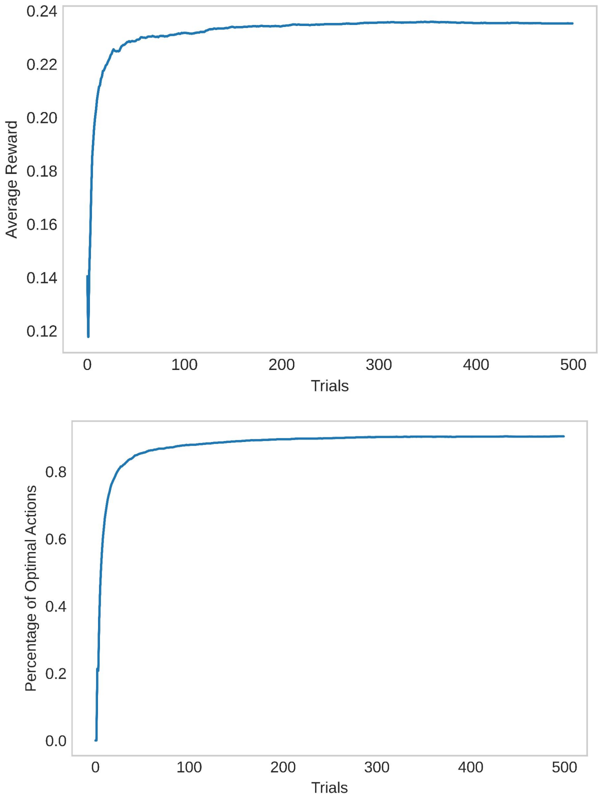
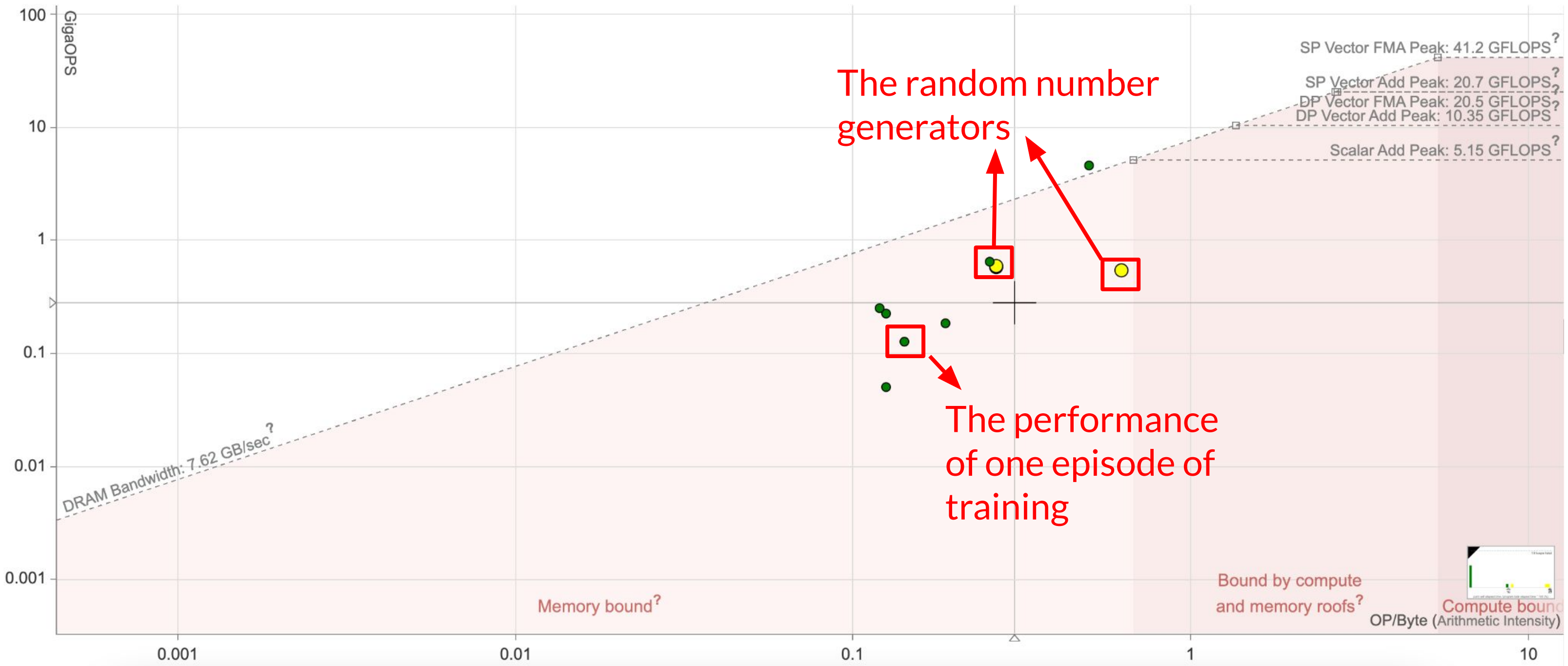


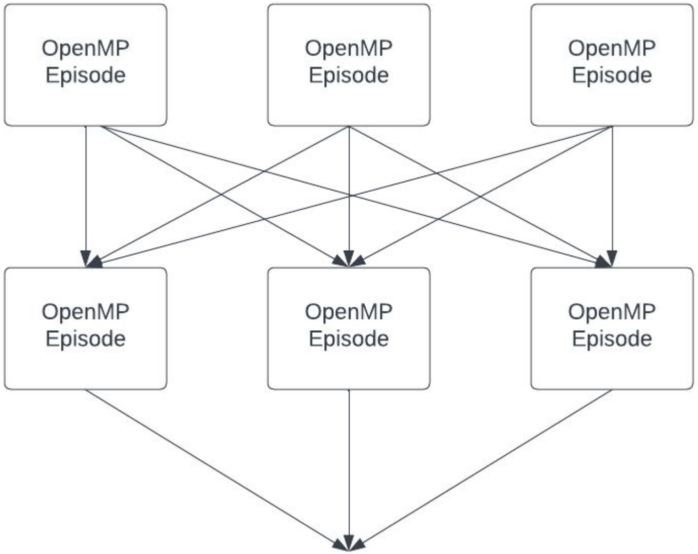
Figure: Single Agent in 10-armed Bandit Task

Roofline Analysis



Parallel programming models

- We cannot make the random number generators perform better
- Parallelize both the agents and the training within each episode
- Parallelize the agents with MPI; parallelize the episodes with OpenMP



Plan to implement parallel code

- Flow of computational steps:
  - Define an initial best action
  - Run the 500 episodes with 100 trials each
- Synchronization must occur:
  - Between agents, after a set number of episodes to share the information among agents and optimally update for learning
  - Within an agent, between each episode
- No load imbalance will be expected due to the random and repeated nature of the RL

Hiding Latency

- Latency will likely occur largely when sharing information between agents:
  - Small size of information makes latencies even more apparent
  - Must determine optimal frequency in which to share between agents to hide latencies
  - Want to send information between agents while calculations continue to happen