

Slicing in 5G networks

Update 17/11/2020

Alessandro Spallina

A slice simulator with MDP resolution

Issues

- Difficulties with costs less than 1 in config (workable problem)
- Slurm: no python3.8 module (workable problem: lab-ia staff)
- State Explosion (???)

A slice simulator with MDP resolution

State Explosion (past presentation)

ServerNum	Queue0	Queue1	Queue2	StateNum	TransNum	ActNum	RAM
1	1	1	1	32	1024	4	ok
5	5	5	5	12096	146M	56	68GB
10	10	10	10	380666	144B	286	302TB

- Switch to interval for states:

$S0=\{(0,0),(0,0),(0,0)\}$; $S1=\{(0,\Delta_{server}),(0,0),(0,0)\}$; $S2=\{(\Delta_{queue0},0),(0,0),(0,0)\}$; ...

A slice simulator with MDP resolution

Assumptions for following tables

- We'll use float16 (2 bytes, $[-1e-07, +1e-07]$) instead of float64 (8 bytes)
- TransMatrix is a matrix $\text{actNum} * \text{StateNum} * \text{StateNum}$
- Size is calculated as `"np.zeros((9,756,756), dtype=np.float16).nbytes"`

A slice simulator with MDP resolution

State Explosion (multiple actions) with TWO slices.

ServerNum	Queue0	Queue1	StateNum	ActNum	RAM for TransMatrix
1	1	1	12	3	900 B
5	5	5	756	21	24MB
10	10	10	7986	66	8.4GB

A slice simulator with MDP resolution

State Explosion (multiple actions) with THREE slices.

ServerNum	Queue0	Queue1	Queue2	StateNum	ActNum	RAM for TransMatrix
1	1	1	1	32	4	8kB
5	5	5	5	12096	56	16GB
10	10	10	10	380666	286	75.4TB

A slice simulator with MDP resolution

State Explosion (unitary actions) with TWO slices.

ServerNum	Queue0	Queue1	StateNum	ActNum	RAM for TransMatrix
1	1	1	12	9	2.5kB
5	5	5	756	9	10MB
10	10	10	7986	9	1.1GB

A slice simulator with MDP resolution

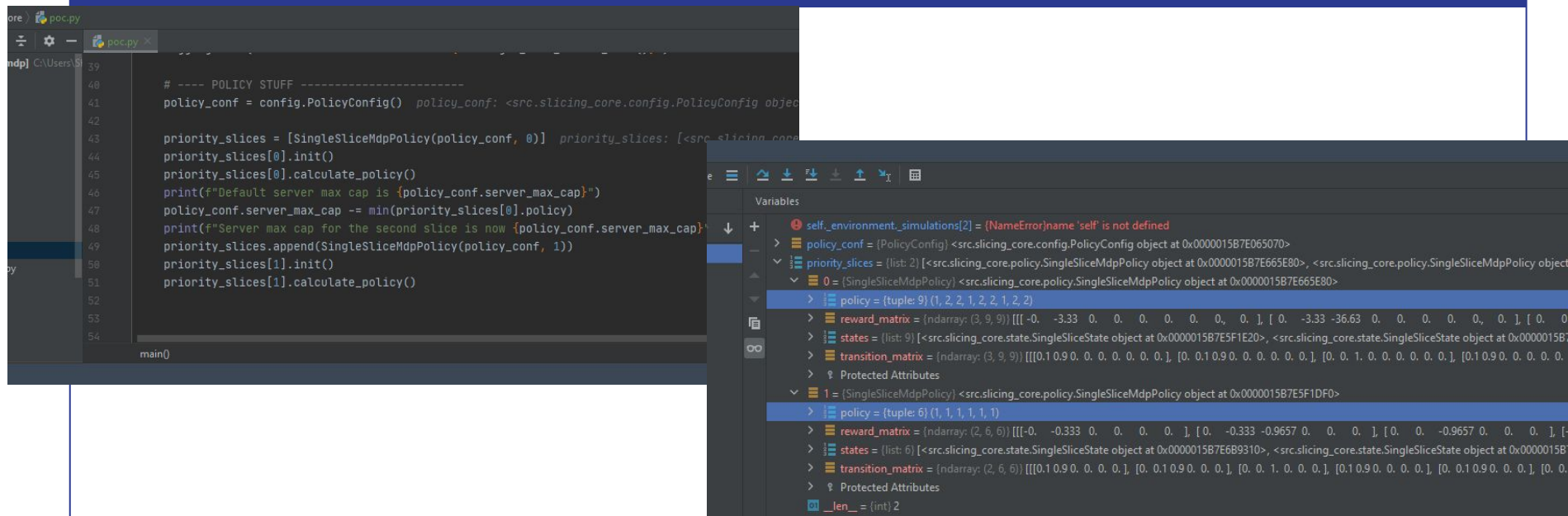
State Explosion (unitary actions) with THREE slices.

ServerNum	Queue0	Queue1	Queue2	StateNum	ActNum	RAM for TransMatrix
1	1	1	1	32	9	18kB
5	5	5	5	12096	9	2.6GB
10	10	10	10	380666	9	2.37TB

Even if this solution optimize a lot (from 302TB to 2.4TB), this method is still non-scalable

A slice simulator with MDP resolution

PriorityMultiSliceMdpPolicy proof of concept



The image shows a Python IDE with a code editor on the left and a variable explorer on the right. The code editor displays the following Python code:

```
39
40
41 # ---- POLICY STUFF -----
42 policy_conf = config.PolicyConfig() policy_conf: <src.slicing_core.config.PolicyConfig object>
43
44 priority_slices = [SingleSliceMdpPolicy(policy_conf, 0)] priority_slices: [<src.slicing_core.policy.SingleSliceMdpPolicy object>]
45 priority_slices[0].init()
46 priority_slices[0].calculate_policy()
47 print(f"Default server max cap is {policy_conf.server_max_cap}")
48 policy_conf.server_max_cap -= min(priority_slices[0].policy)
49 print(f"Server max cap for the second slice is now {policy_conf.server_max_cap}")
50 priority_slices.append(SingleSliceMdpPolicy(policy_conf, 1))
51 priority_slices[1].init()
52 priority_slices[1].calculate_policy()
53
54
55 main()
```

The variable explorer on the right shows the following variables:

- `self_environment_simulations[2]`: `(NameError)name 'self' is not defined`
- `policy_conf`: `<src.slicing_core.config.PolicyConfig object at 0x0000015B7E065070>`
- `priority_slices`: `(list: 2) [<src.slicing_core.policy.SingleSliceMdpPolicy object at 0x0000015B7E665E80>, <src.slicing_core.policy.SingleSliceMdpPolicy object at 0x0000015B7E665E80>]`
- `0`: `(SingleSliceMdpPolicy) <src.slicing_core.policy.SingleSliceMdpPolicy object at 0x0000015B7E665E80>`
- `policy`: `(tuple: 9) (1, 2, 2, 1, 2, 2, 1, 2, 2)`
- `reward_matrix`: `(ndarray: (3, 9, 9)) [[[-0. -3.33 0. 0. 0. 0. 0. 0. 0.], [0. -3.33 -36.63 0. 0. 0. 0. 0. 0.], [0. 0. 0. 0. 0. 0. 0. 0. 0.]]]`
- `states`: `(list: 9) [<src.slicing_core.state.SingleSliceState object at 0x0000015B7E5F1E20>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E5F1E20>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E5F1E20>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E5F1E20>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E5F1E20>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E5F1E20>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E5F1E20>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E5F1E20>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E5F1E20>]`
- `transition_matrix`: `(ndarray: (3, 9, 9)) [[[[0.1 0.9 0. 0. 0. 0. 0. 0. 0.], [0. 0.1 0.9 0. 0. 0. 0. 0. 0.], [0. 0. 1. 0. 0. 0. 0. 0. 0.], [0.1 0.9 0. 0. 0. 0. 0. 0. 0.], [0. 0. 0. 1. 0. 0. 0. 0. 0.], [0. 0. 0. 0. 1. 0. 0. 0. 0.], [0. 0. 0. 0. 0. 1. 0. 0. 0.], [0. 0. 0. 0. 0. 0. 1. 0. 0.], [0. 0. 0. 0. 0. 0. 0. 1. 0.]]]]`
- `Protected Attributes`
- `1`: `(SingleSliceMdpPolicy) <src.slicing_core.policy.SingleSliceMdpPolicy object at 0x0000015B7E5F1DF0>`
- `policy`: `(tuple: 6) (1, 1, 1, 1, 1, 1)`
- `reward_matrix`: `(ndarray: (2, 6, 6)) [[[-0. -0.333 0. 0. 0. 0. 0.], [0. -0.333 -0.9657 0. 0. 0. 0.], [0. 0. -0.9657 0. 0. 0. 0.], [0. 0. 0. -0.9657 0. 0. 0. 0.], [0. 0. 0. 0. -0.9657 0. 0. 0.], [0. 0. 0. 0. 0. -0.9657 0. 0. 0.]]]]`
- `states`: `(list: 6) [<src.slicing_core.state.SingleSliceState object at 0x0000015B7E6B9310>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E6B9310>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E6B9310>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E6B9310>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E6B9310>, <src.slicing_core.state.SingleSliceState object at 0x0000015B7E6B9310>]`
- `transition_matrix`: `(ndarray: (2, 6, 6)) [[[[0.1 0.9 0. 0. 0. 0.], [0. 0.1 0.9 0. 0. 0.], [0. 0. 1. 0. 0. 0.], [0.1 0.9 0. 0. 0. 0.], [0. 0. 0. 1. 0. 0.], [0. 0. 0. 0. 1. 0.]]]]`
- `Protected Attributes`
- `_len_`: `(int) 2`