

**NAME**

`pslice` – display a 2-dimensional slice of a QNET policy

**SYNOPSIS**

`pslice` [*OPTIONS*] *POLICY*

**DESCRIPTION**

Display a specified slice of the QNET policy data stored in the file *POLICY*. The program needs no additional parameters to show two-dimensional policy data. If the dimension of the policy data is greater than two then a two-dimensional slice must be specified using the **--slice** option. By default the actions from all servers are shown but the user can request the action data for a single server.

**OPTIONS**

**-c, --csv-file=FILE**

write two-dimensional policy data to *FILE* in comma separated value format, suitable for viewing with a spreadsheet program.

**-N, --truncation=N**

use *N* as the truncation for each class. Overrides all larger truncation values given in *POLICY*.

**-n, --narrow**

do not insert spaces between actions. This allows for a more compact display.

**-s, --server=N**

compute and show policy data only for server *N*.

**-S, --slice=SPEC**

compute and show policy data for the two-dimensional slice specified by *SPEC* (see EXAMPLE below).

**-q, --quiet**

suppress output

**-v, --verbose**

show ALPS data

**--version**

display program version.

**--help** display usage information.

**EXAMPLE**

The `[0..10]x[0..10]` portion of the two-dimensional policy stored in `p2.p` can be displayed with the command

```
pslice --truncation=10 p2.p
```

or the shorter version

```
pslice -N 10 p2.p
```

will display output similar to

```
Server: all
Slice is x(1) vs. x(2)
```

```
(0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2)
(0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2)
(0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2)
(0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2)
(0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2)
(0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2)
```

```
(0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2)
(0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2)
(0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2) (0|2)
(0|2) (1|2) (1|2) (1|2) (1|2) (1|2) (1|2) (1|2) (1|2) (1|2) (1|2)
(0|0) (1|0) (1|0) (1|0) (1|0) (1|0) (1|0) (1|0) (1|0) (1|0) (1|0)
```

This shows the action data for both servers. To see only data from server two use either

```
pslice --server=2 --truncation=10 p2.p
or
pslice -s2 -N 10 p2.p
```

The two lines above the state space show that information for all servers is displayed and the slice shown is for  $x(1)$  (increasing in the horizontal direction) vs.  $x(2)$  (increasing in the vertical direction). The bottom-left entry corresponds to the state  $x(1)=0$  and  $x(2)=0$ . An entry of (0|2) in a particular state space location means that in that state server 1 will idle (serves no class) while server serves class 2.

If the state space is three or higher dimensional then the **--slice=SPEC** option must be used to specify a two-dimensional slice of the policy data to display. *SPEC* is a slice specification. A state is fully specified by  $x(1)$ ,  $x(2)$ , ...,  $x(n)$  if there are  $n$  classes in the network. All but two of these values must be supplied to describe a two-dimension slice of the state space. Assuming  $n=4$ , some examples of valid slice specifications are

```
x1:3,x3:2 (x(1)=3,x(2)=free,x(3)=2,x(4)=free)
x1:0,x2:0 (x(1)=0,x(2)=0,x(3)=free,x(4)=free)
x3:5,x4:9 (x(1)=free,x(2)=free,x(3)=5,x(4)=9)
```

Whitespace is not allowed in the specification but '=' may be used in place of ':'.

For example, if `p4.p` contains policy data for a 4 class network, the command

```
pslice --slice=x2:0,x3:2 --truncation 10 p4.p
or
pslice -S x2:0,x3:2 -N 10 p4.p
```

will create a policy slice corresponding to  $x(1)=\text{free}$ ,  $x(2)=0$ ,  $x(3)=2$ , and  $x(4)=\text{free}$ :

Server: all Slice is  $x(1)$  (horizontal) vs.  $x(4)$  (vertical): indexing begins with 0 Fixed state values:

```
x(2)=0
x(3)=2
```

```
(4|0) (4|0) (4|0) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3)
(4|0) (4|0) (4|0) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3)
(4|0) (4|0) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3)
(4|0) (4|0) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3)
(4|0) (4|0) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3)
(4|0) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3)
(4|0) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3)
(4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3)
(4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3)
(4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3) (4|3)
(0|3) (1|3) (1|3) (1|3) (1|3) (1|3) (1|3) (1|3) (1|3) (1|3) (1|3)
```

If the truncations are larger than 5 or 10 it often becomes difficult to view policy slice information on a terminal or print out. To work around this, you can use the **--server** option to show only actions for a single server or **--csv-file=FILE** to save the policy slice information to a file in comma separated value format suitable for viewing with a spreadsheet program.

**SEE ALSO**

**qnetdp(1)**, **alps2policy(1)**

**AUTHOR**

Jonathan Senning

Copyright © 2007-2009 Department of Mathematics and Computer Science Gordon College, 255 Grapevine Road, Wenham MA, 01984