Queueing Networks and Markov Chains analysis with the Octave Queueing Package

Moreno Marzolla
moreno.marzolla@unibo.it
https://www.moreno.marzolla.name/

Dept. of Computer Science and Engineering (DISI) University of Bologna, Italy

TOSME21, nov 12, 2021

The `queueing` package

- Analysis of (product-form) Queueing Networks and continuous/discrete Markov chains
- Written in GNU Octave (a free Matlab clone)
- Free software (GPLv3+)

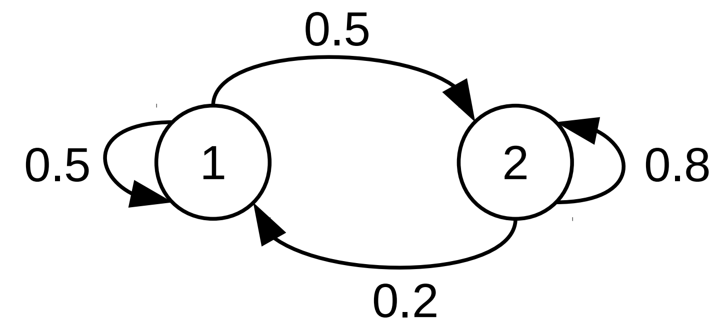
`https://octave.sourceforge.net/queueing/`

The `queueing` package

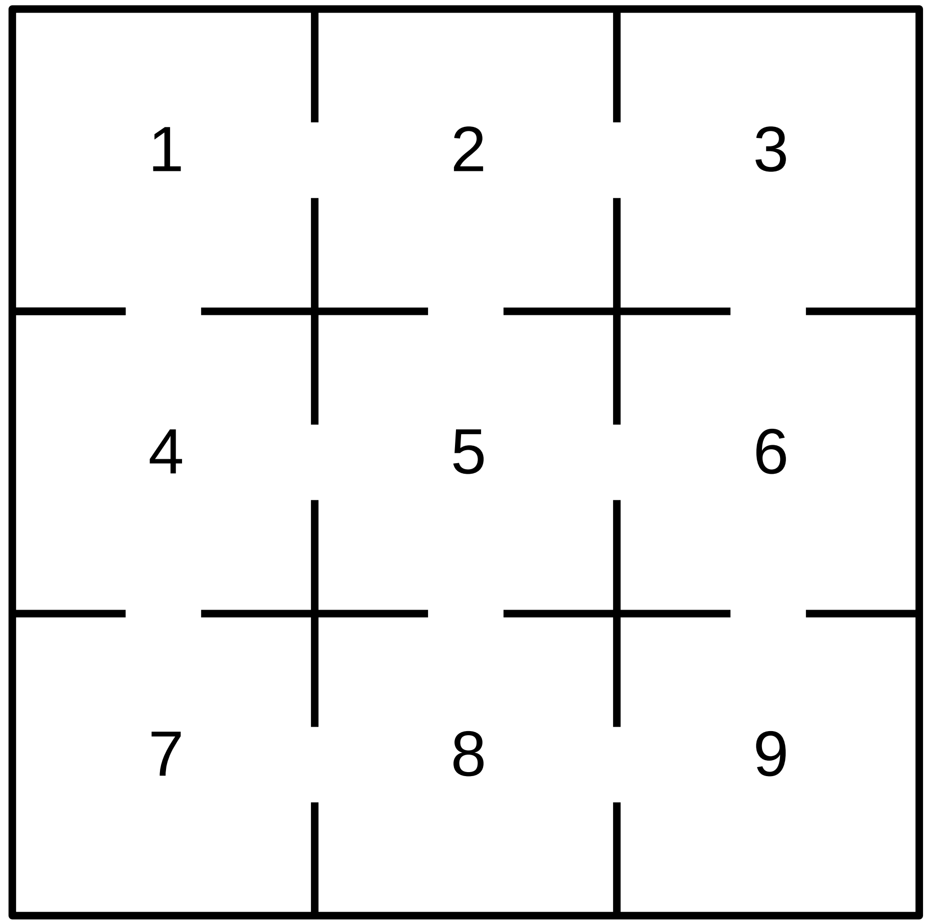
- Discrete- and Continuous-time Markov Chains
  - Transient/stationary state occupancy probabilities
  - Expected sojourn times
  - Mean Time to Absorption
  - First Passage Times
- Single-Station Queueing Systems
  - M/M/1, M/M/m, M/M/∞, M/M/1/K , M/M/m/K , ...
- Product-Form Queueing Networks
  - MVA and Convolution algorithms
  - Steady-State analysis of Open, Closed and Mixed networks
  - Multiple job classes
  - Performance bounds (Asymptotic, Balanced, Geometric)

Installation and usage

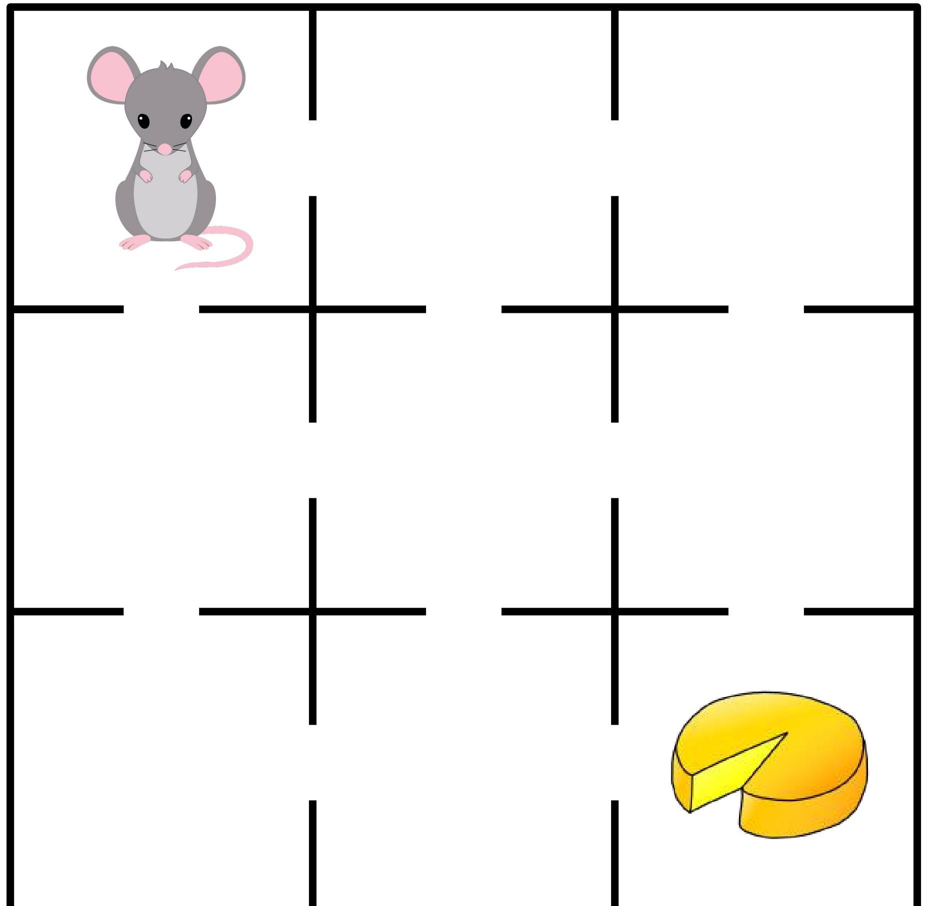
```
octave> pkg install -local -forge queueing

octave> pkg load queueing

octave> dtmc([0.5 0.5; 0.2 0.8])
ans =
   0.28571  0.71429

octave> help dtmc
-- Function File: P = dtmc (P)
-- Function File: P = dtmc (P, N, P0)
   Compute stationary or transient state occupancy
   probabilities for a discrete-time Markov chain.
   ...
```



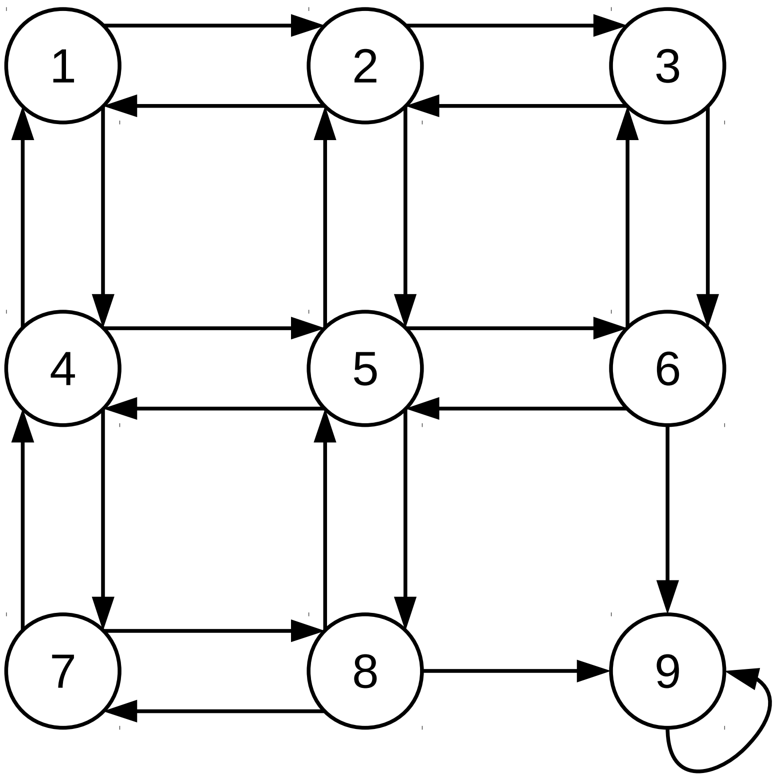
Mouse in a maze



Mouse in a maze

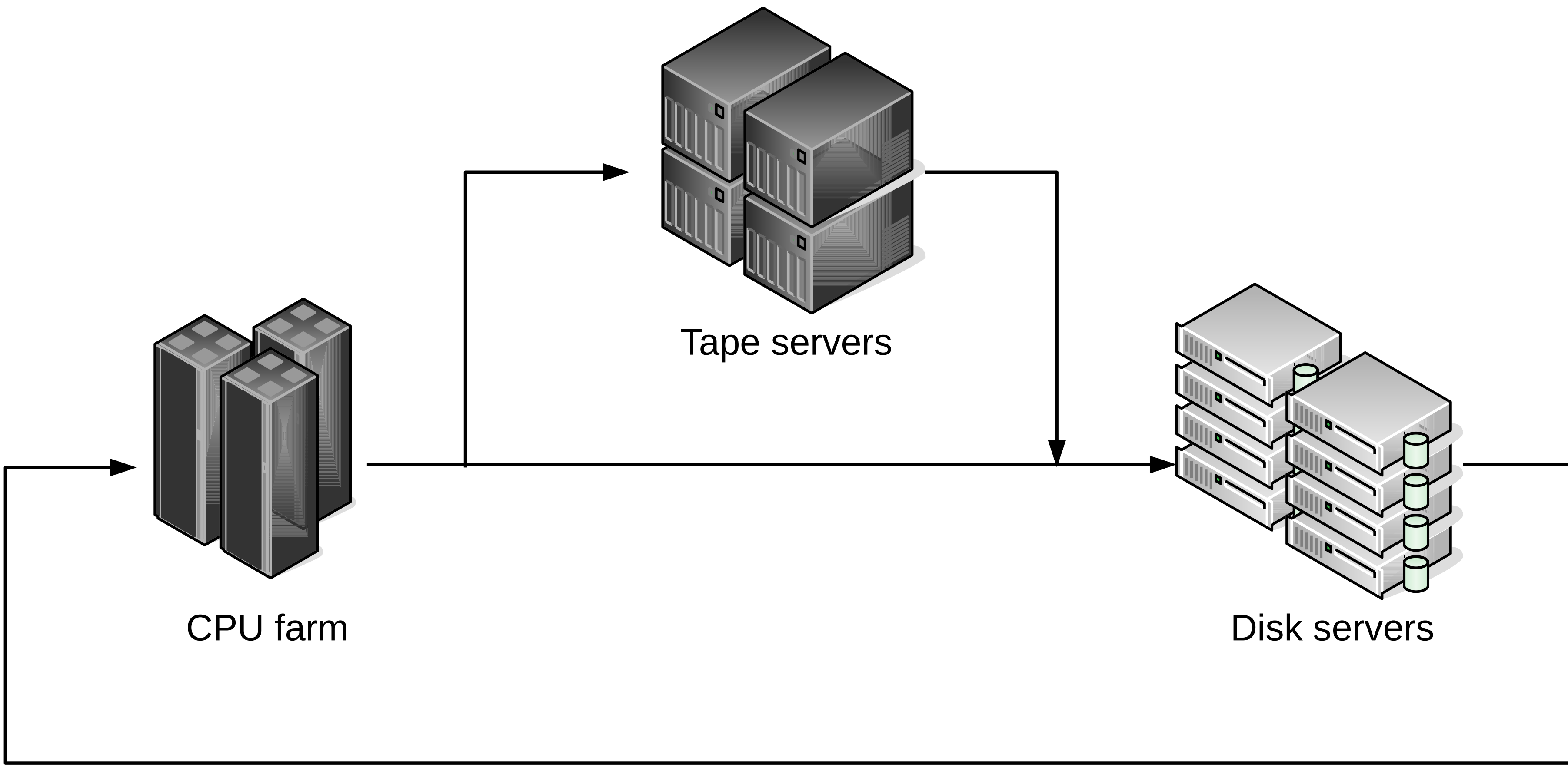


Mouse in a maze

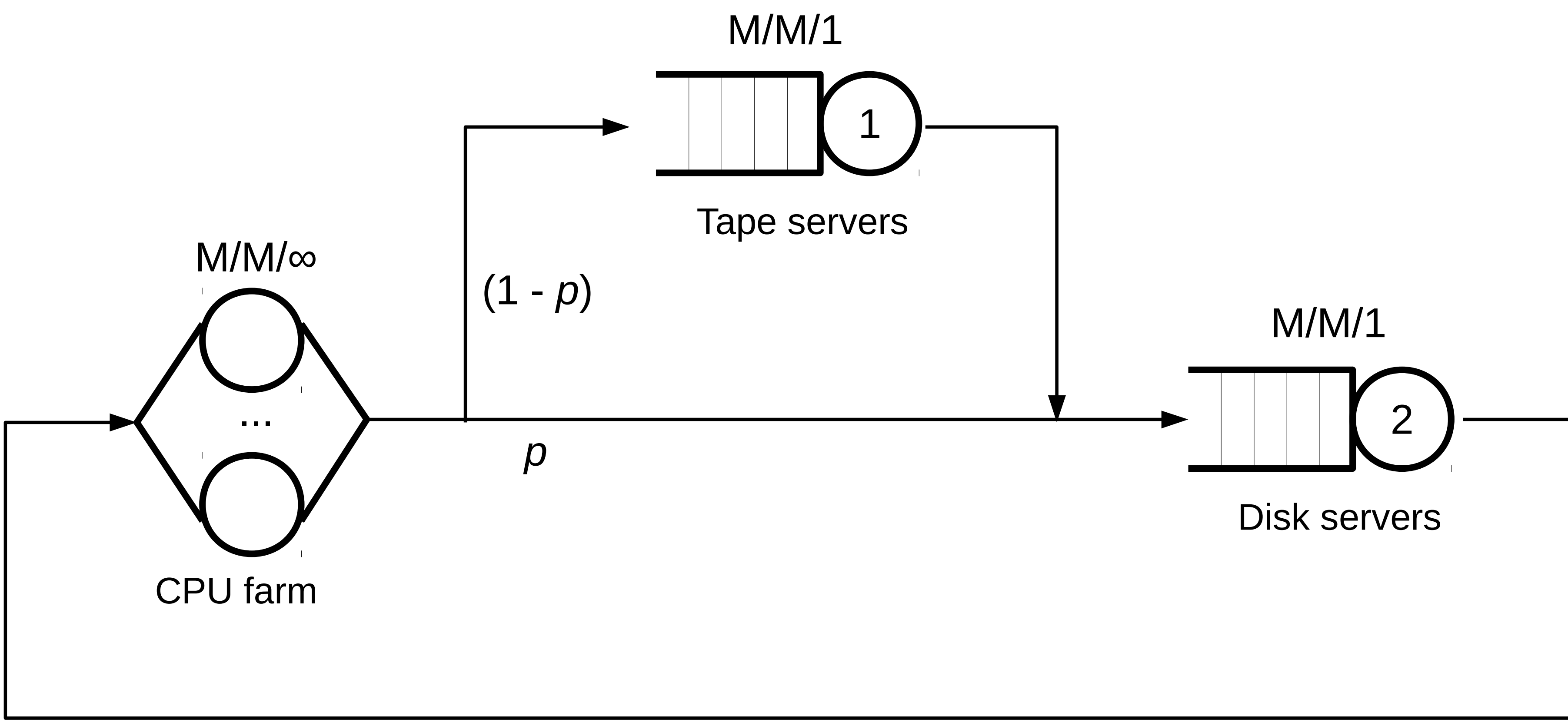



```
P = zeros(9,9);
P(1,[2 4]     ) = 1/2;
P(2,[1 5 3]   ) = 1/3;
P(3,[2 6]     ) = 1/2;
P(4,[1 5 7]   ) = 1/3;
P(5,[2 4 6 8]) = 1/4;
P(6,[3 5 9]   ) = 1/3;
P(7,[4 8]     ) = 1/2;
P(8,[7 5 9]   ) = 1/3;
P(9,9         ) = 1;
p0 = [1 0 0 0 0 0 0 0 0];
dtmcmtta(P, p0)

    ans = 18.000
```

Example: Compute Farm

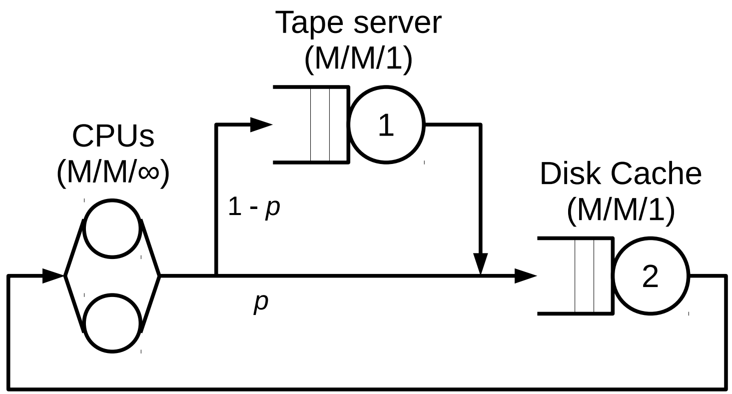


Example: Compute Farm



Example: Compute Farm

- CPU burst $Z = 1800\text{s}$
- Tape service time $S_1 = 300\text{s}$
- A) Larger cache, slow disks
  - $S_2 = 40\text{s}$
  - Cache hit rate $p = 0.9$
- B) Smaller cache, fast disks
  - $S_2 = 30\text{s}$
  - Cache hit rate $p = 0.75$



Example: Compute Farm

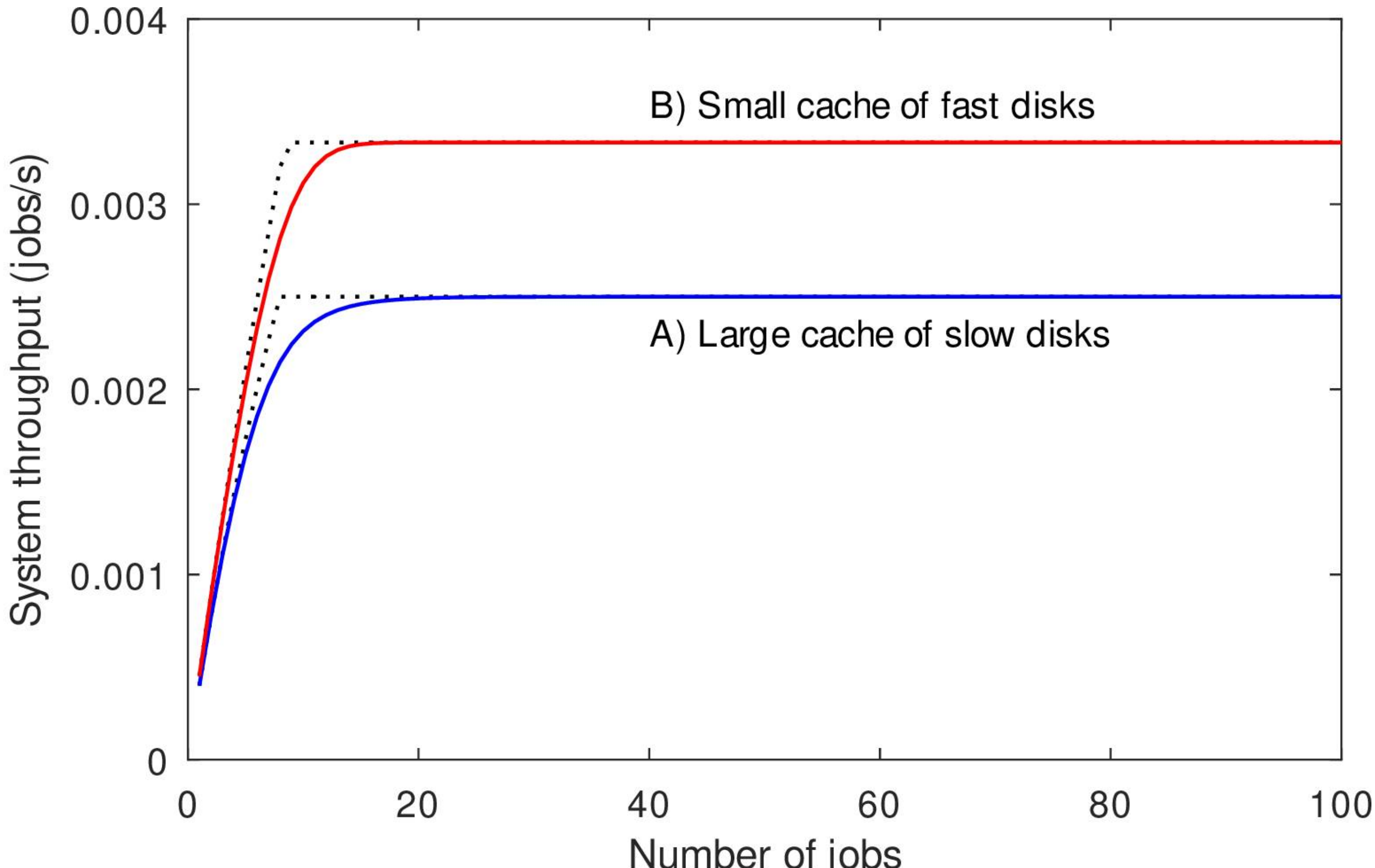
```
SA = [300 40];
pa = .9;
P = [  0   1;
     1-pa  pa ];
VA = qncsvisits(P);
DA = SA.*VA
DA =
   300   400
XA = 1/max(DA)
XA =
   2.5000e-03
```

```
SB = [300 30];
pb = .75;
P = [  0   1;
     1-pb  pb ];
VB = qncsvisits(P);
DB = SB.*VB
DB =
   300   120
XB = 1/max(DB)
XB =
   3.3333e-03
```

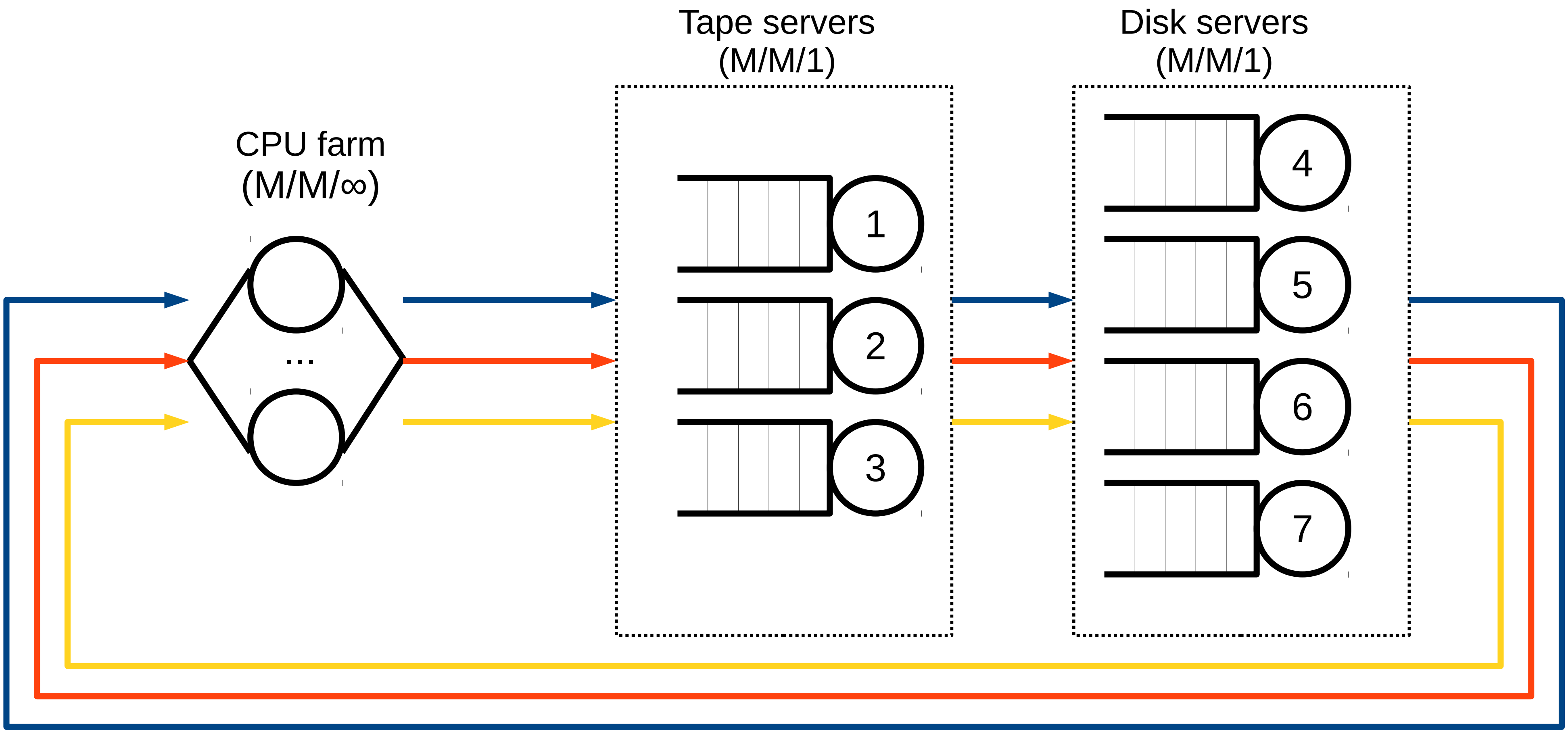
Example: Compute Farm

```
Z = 1800;
NN = 1:100;
XA = XB = XA_mva = XB_mva = zeros(size(NN));
for n=NN
  [nc XA(n)] = qncsbsb(n, SA, VA, 1, Z);
   [U R Q X] = qncsmva(n, SA, VA, 1, Z);
   XA_mva(n) = X(1)/VA(1);
  [nc XB(n)] = qncsbsb(n, SB, VB, 1, Z);
   [U R Q X] = qncsmva(n, SB, VB, 1, Z);
   XB_mva(n) = X(1)/VB(1);
endfor
```

Example: Compute Farm



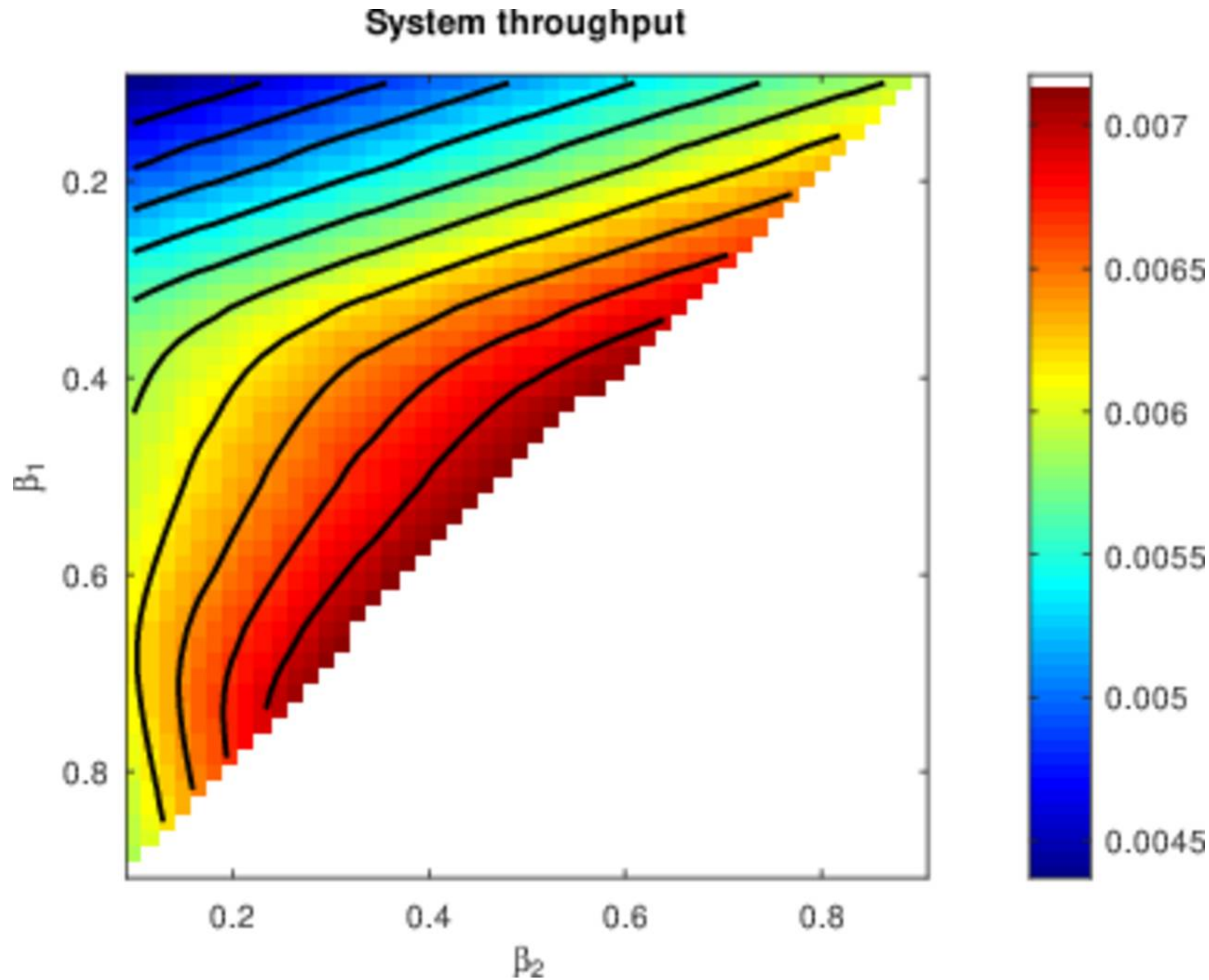
Example: Multiclass QN



Example: Multiclass QN

```
D = [100 140 200  30  50  20  10;     # service demands
     180  10  70  10  90 130  30;
     280 160 150  90  20  50  18];
Z = [2400 1800 2100];                 # mean duration of CPU burst
V = ones(size(D));                    # number of visits
m = [1 1 1 1 1 1 1];                  # number of servers
N = [100, 150, 50];

[U R Q X ] = qncmmva( N , D , V , m, Z );
X_sys = sum( X(:,1) ./ V(:,1) )

   X_sys = 6.5900e-03
```

Example: Multiclass QN



Why?

- Modeling Environment
  - The **`queueing`** package and GNU Octave can be used for rapid prototyping and iterative refinement of models
  - Parametric studies are possible since models are defined programmatically
- Reference implementations
  - The **`queueing`** package provides implementations of some common QN/MC algorithms, so that people do not have to reimplement the wheel
- Teaching
  - **`queueing`** is used in some Universities to teach performance modeling courses
  - Students can immediately put algorithms at work to solve practical problems

Why not?

- No GUI (*yet*)
  - Steep learning curve: models must be defined programmatically
- No support (*yet*) for extended QNs
  - Blocking, priorities, fork/join, passive resources...
- Known issues with some algorithms
  - Multiclass MVA can be *slow*
  - Numerical instability with load-dependent service centers

Thanks for your attention!

Questions?

`https://octave.sourceforge.net/queueing/`