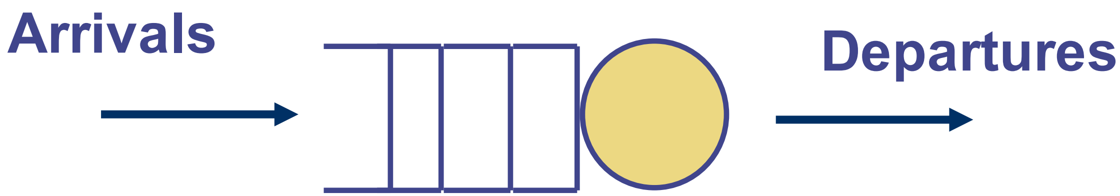
COMP9334 Capacity Planning for Computer Systems and Networks

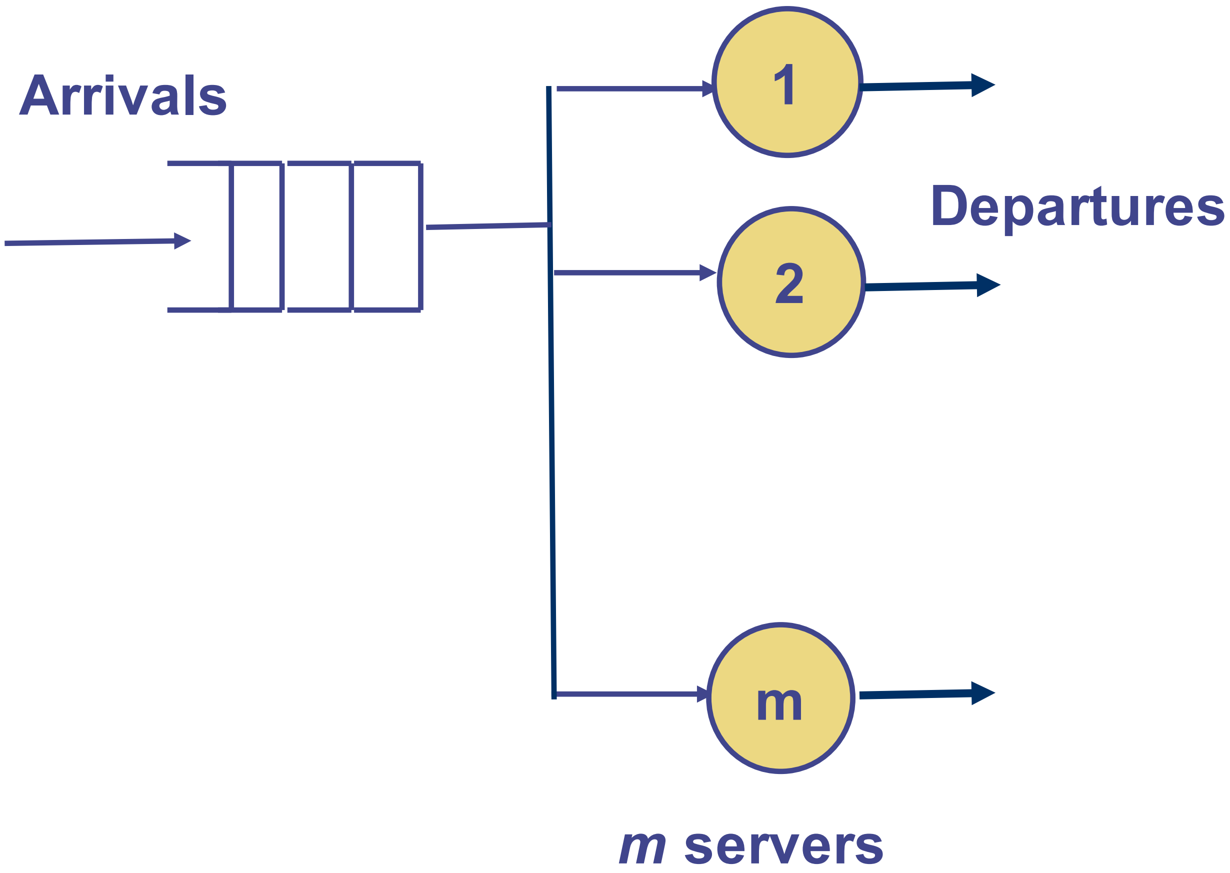
Week 4: Markov Chain

Last week: Queues with Poisson arrivals

- Single-server

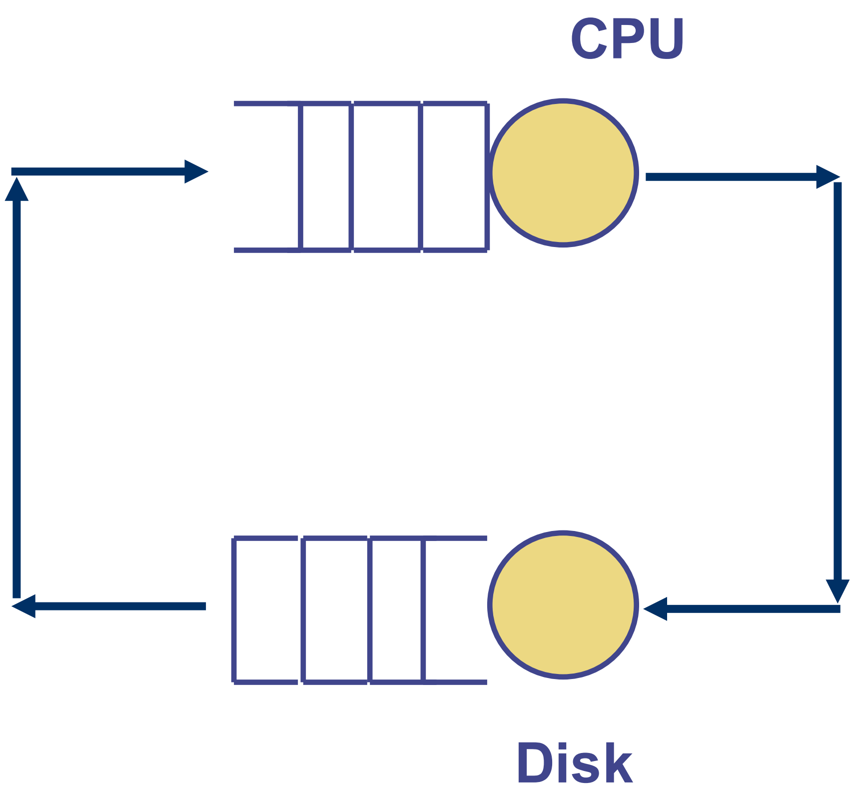


- Multi-server



This week: Markov Chain

- You can use Markov Chain to analyse
  - Closed queueing network (see example below)
  - Reliability problem



- There are *n* jobs in the closed system
- What is the response time of one job?
- What is the response time if we replace the CPU with one that is twice as fast?

This lecture: Road Map

- A recap on the methodology that we used to analyse Poisson queues last week
  - You were using Markov Chain without knowing it
- Analysing closed queueing networks
- Analysing reliability problem

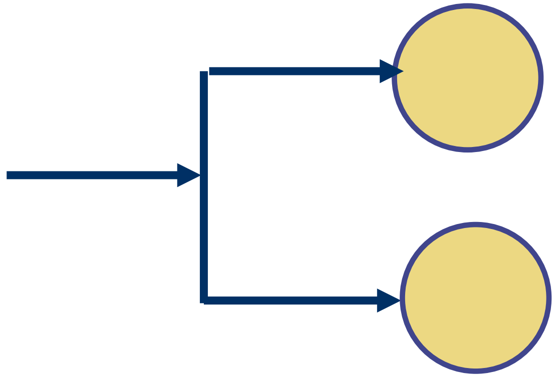
Recap: Properties of exponential distribution

- Exponential inter-arrival time and service time gives rise to the following two properties
- Inter-arrival time is exponential with mean rate λ,
  - Consider a small time interval δ
  - Probability [no arrival in δ] = $1 - \lambda \delta$
  - Probability [1 arrival in δ] = $\lambda \delta$
  - Probability [2 or more arrivals in δ] ≈ 0
- Service time distribution is exponential with mean rate μ
  - Consider a small time interval δ
  - Probability [0 job will finish its service in next δ seconds] = $1 - \mu \delta$
  - Probability [1 job will finish its service in next δ seconds] = $\mu \delta$
  - Probability [> 2 jobs will finish its service in next δ seconds] ≈ 0

Recap: M/M/2/2 queue

Exponential
Inter-arrivals (λ)

Exponential
Service time (μ)



No buffer.

Two servers

- A call centre analogy

Arrivals →

Call centre:

2 operators.
No holding slot.

- Calls are accepted as long as at least one operator is available.
- If both operators are busy, an arriving call is rejected.

- Let us recall how we can analyse this system

Recap: Analysing M/M/2/2

- The system can be in one of the following three states
  - State 0 = 0 call in the system (= both operators are idle)
  - State 1 = 1 call in the system (= one operator is busy, one is idle)
  - State 2 = 2 calls in the system (= both operators are busy)
- Define the probability that a certain state occurs

$$P_0 = \text{Probability in State 0}$$

$$P_1 = \text{Probability in State 1}$$

$$P_2 = \text{Probability in State 2}$$

Recap: The transition probabilities

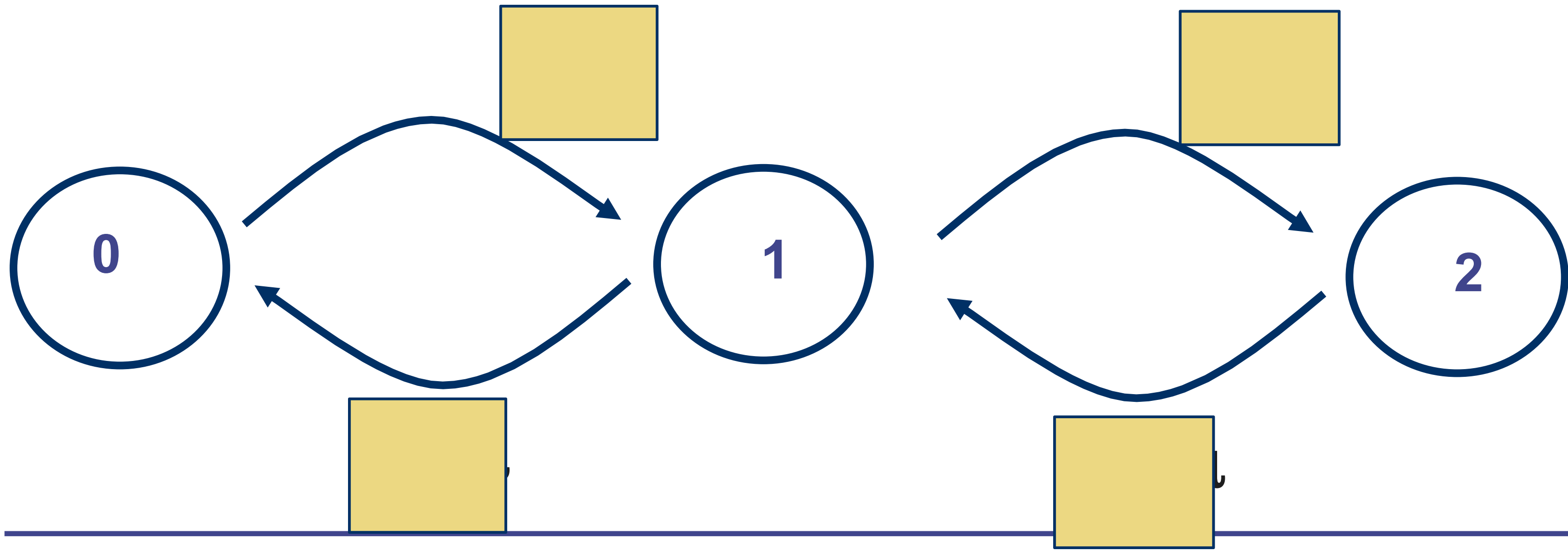
- Consider a small time interval δ
  - If the system is in State 1
    - What is the probability that it will move to State 0?
    - What is the probability that it will move to State 2?
- Transiting from *State 1* ➔ *State 0*
- Transiting from *State 1* ➔ *State 2*
-
-

Exercise: The transition probabilities

- Can you work out the following transition probabilities
  - Prob [*State 0* → *State 1*] = ?
  - Prob [*State 0* → *State 2*] = ?
  - Prob [*State 2* → *State 0*] = ?
  - Prob [*State 2* → *State 1*] = ?

Recap: The state transition diagram

- Given the following transition probabilities (over a small time interval δ)
  - Prob [*State 0* ➔ *State 1*] =
  - Prob [*State 0* ➔ *State 2*] =
  - Prob [*State 1* ➔ *State 0*] =
  - Prob [*State 1* ➔ *State 2*] =
  - Prob [*State 2* ➔ *State 0*] =
  - Prob [*State 2* ➔ *State 1*] =
- We draw the following state transition diagram
  - Note 1: We label the arc with transition rate = transition probability / δ
  - Note 2: Arcs with zero rate are not drawn



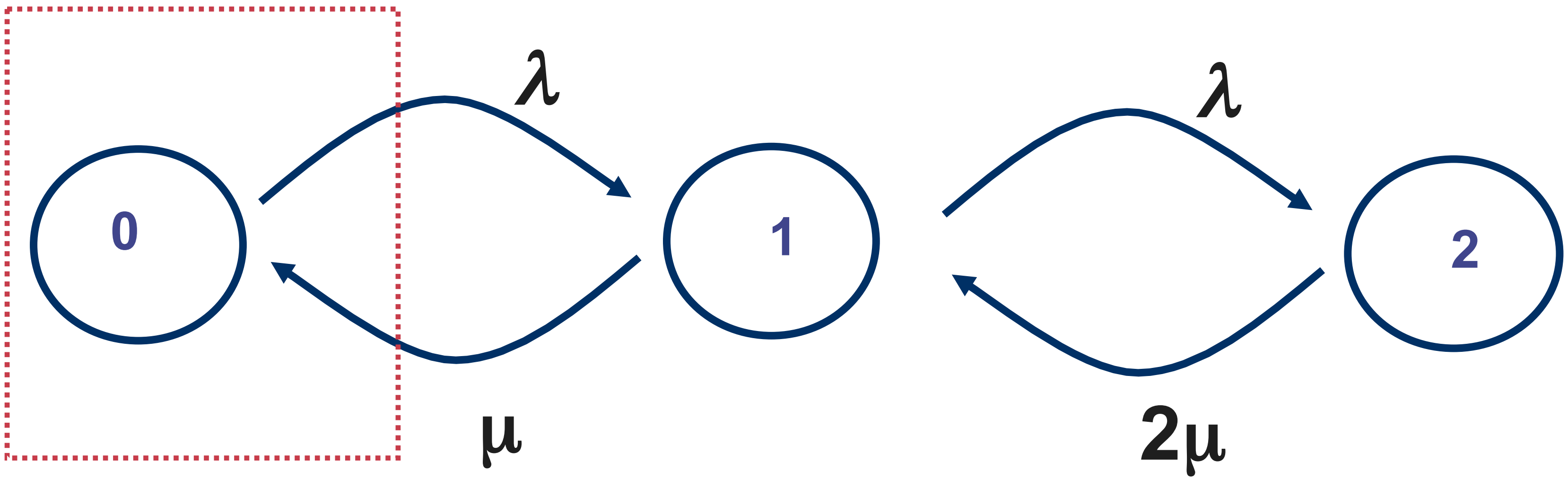
Recap: Setting up the balance equations (1)

- For steady state, we have
  - Probability of transiting into a “box” = Probability of transiting out of a “box”
  - Rate of transiting into a “box” = Rate of transiting out of a “box”
- Note a “box” can include one or more state
- The “box” is the dotted square shown below

$$\text{Prob out of ''box''} = P_0 \lambda \delta$$

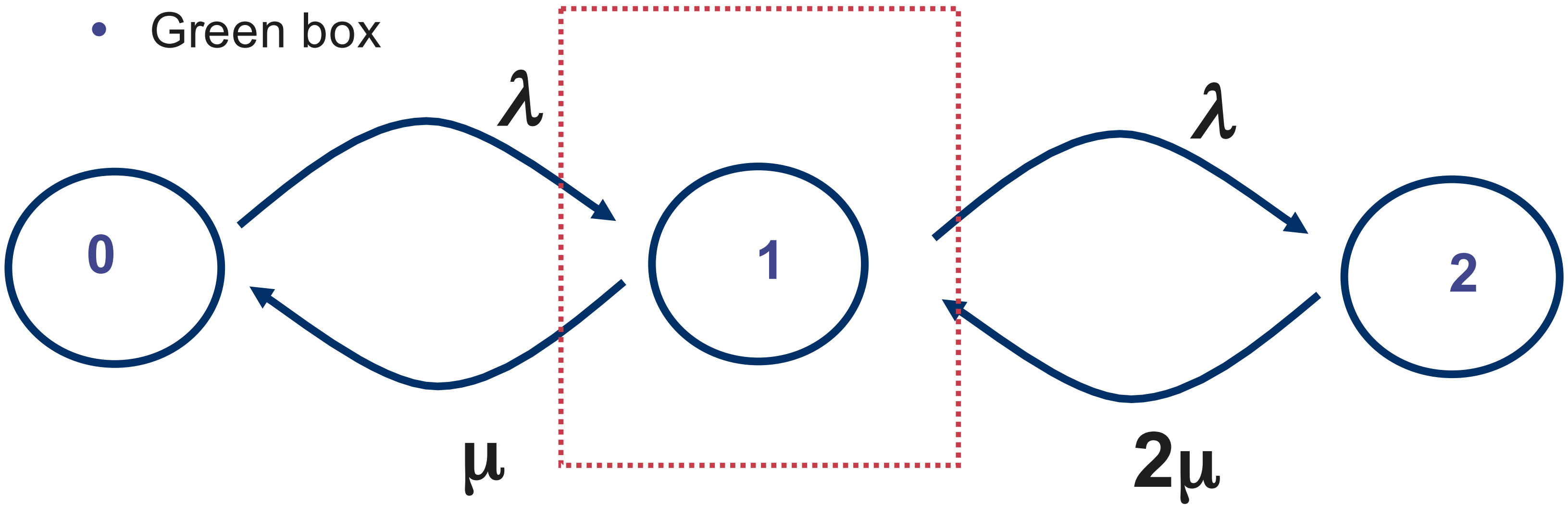
$$\text{Prob into ''box''} = P_1 \mu \delta$$

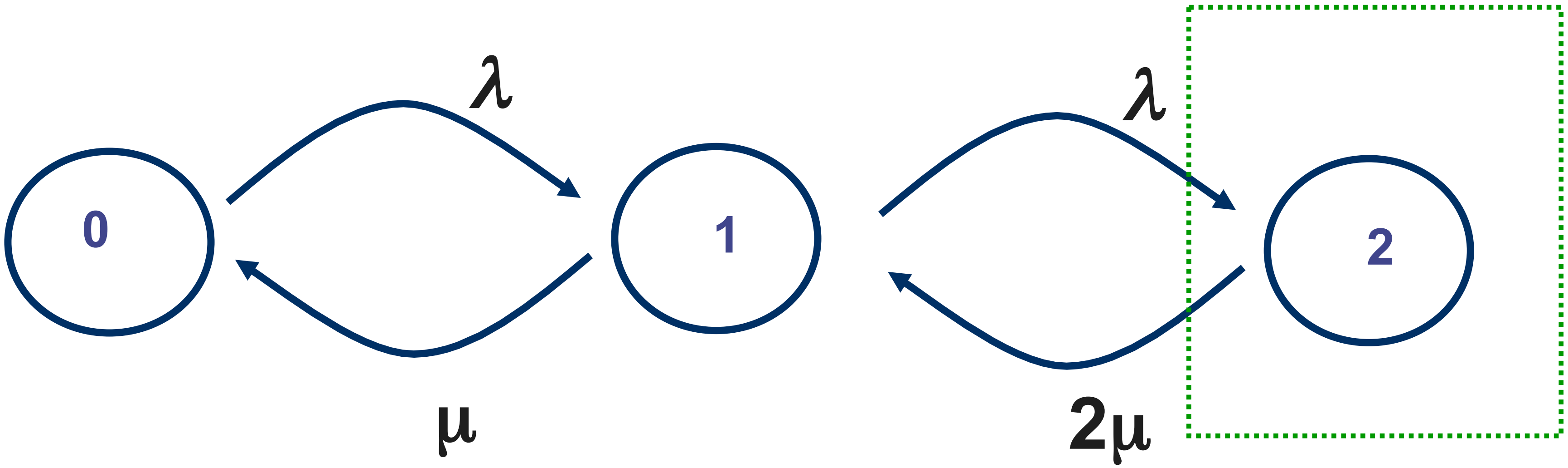
$$\lambda P_0 = \mu P_1$$



Exercise: Setting up the balance equations (2)

- Set up the balance equations for the
  - Red box
  - Green box





Recap: The balance equations

- There are three balance equations

$$\lambda P_0 = \mu P_1$$
$$\lambda P_0 + 2\mu P_2 = (\mu + \lambda) P_1$$
$$2\mu P_2 = \lambda P_1$$

- Note that these three equations are not linearly independent
  - First equation + Third equation = Second equation
- There are 3 unknowns (P_0, P_1, P_2) but we have only 2 equations
- We need 1 more equation. What is it?

Recap: Solving for the steady state probabilities

- An addition equation: Sum(Probabilities) = 1
- Solve the following equations for the steady state probabilities P_0, P_1, P_2 :

$$\lambda P_0 = \mu P_1$$

$$2\mu P_2 = \lambda P_1$$

$$P_0 + P_1 + P_2 = 1$$

- By solving these 3 equations, we have

Recap: Steady state probabilities

- By solving the equations on the previous slide, we have the steady state probabilities are:

$$P_0 = \frac{1}{1 + \frac{\lambda}{\mu} + \frac{\lambda}{\mu}\frac{\lambda}{2\mu}}$$

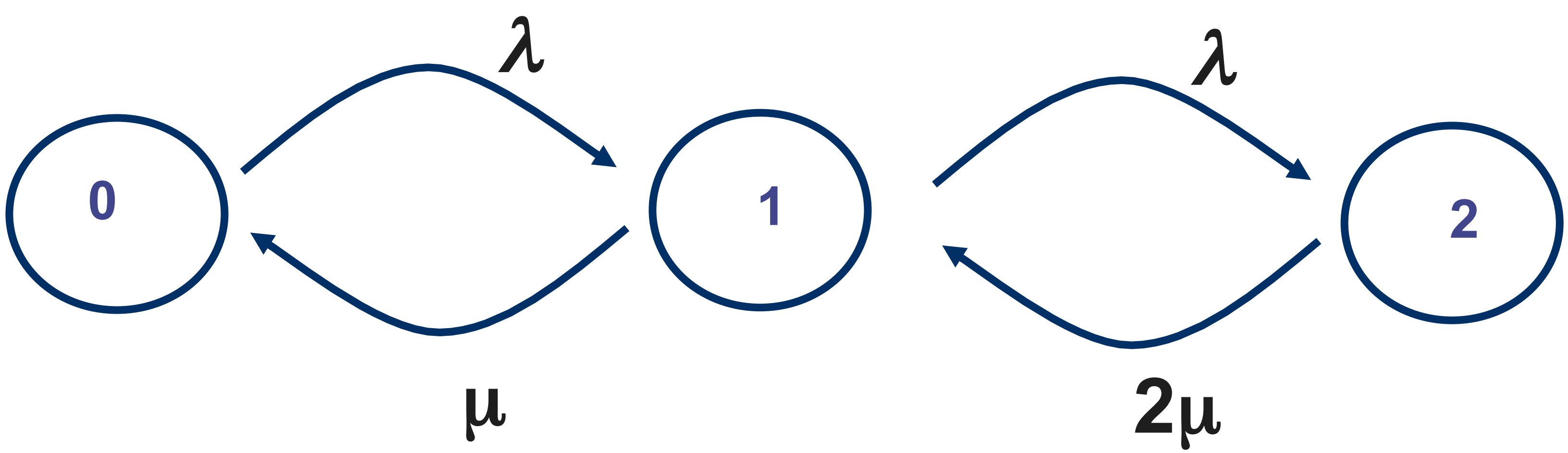
$$P_1 = \frac{\frac{\lambda}{\mu}}{1 + \frac{\lambda}{\mu} + \frac{\lambda}{\mu}\frac{\lambda}{2\mu}}$$

$$P_2 = \frac{\frac{\lambda}{\mu}\frac{\lambda}{2\mu}}{1 + \frac{\lambda}{\mu} + \frac{\lambda}{\mu}\frac{\lambda}{2\mu}}$$

- If we know the values of λ and μ, we can find the numerical values of these probabilities

Markov chain

- The state-transition model that we have used is called a continuous-time Markov chain
  - There is also discrete-time Markov chain
- The transition from a state of the Markov chain to another state is characterised by an exponential distribution
  - E.g. The transition from State p to State q is exponential with rate r_{pq}, then consider a small time interval δ
  - Probability [Transition from State p to State q in time δ] = $r_{pq}\,\delta$

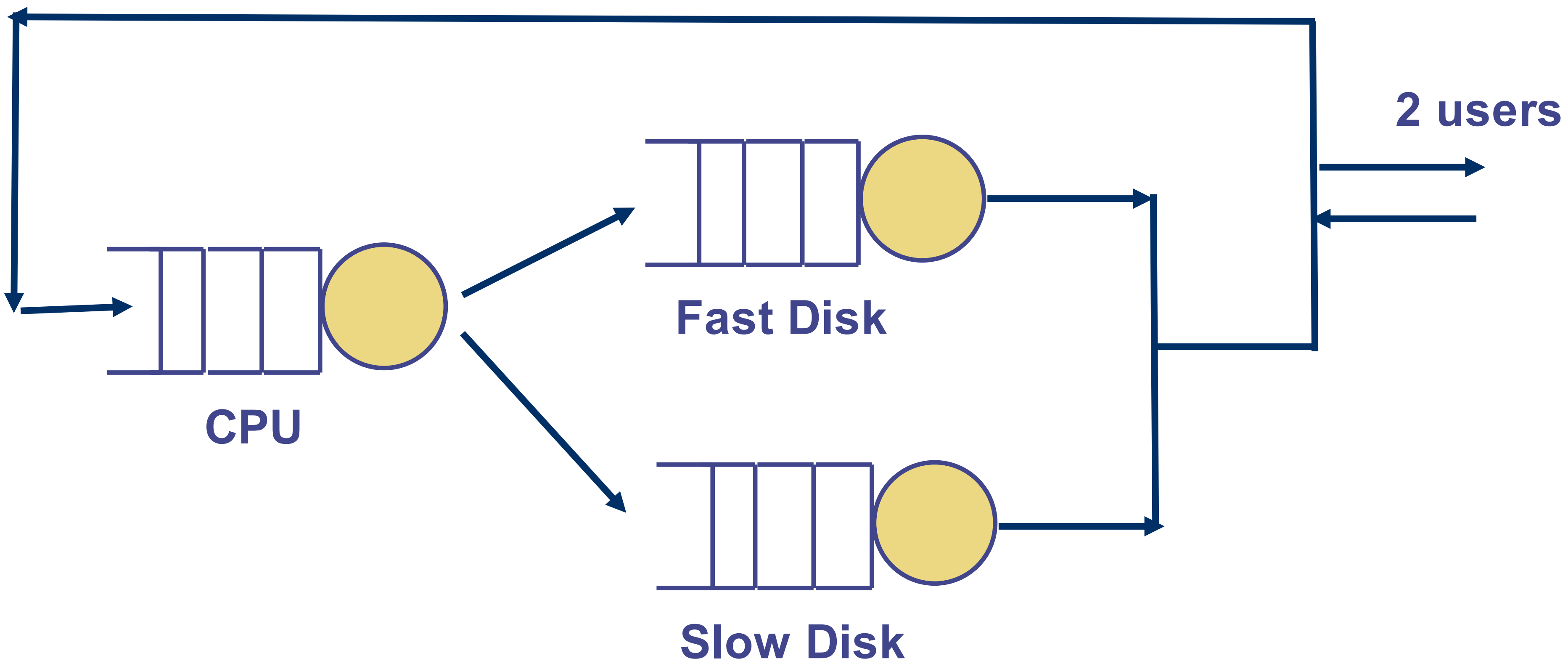


Method for solving Markov chain

- A Markov chain can be solved by
  - Identifying the states (may not be easy)
  - Find the transition rate between the states
  - Solve the steady state probabilities
- You can then use the steady state probabilities as a stepping stone to find the quantity of interest (e.g. response time etc.)
- We will study two Markov chain problems in this lecture:
  - Problem 1: A Database server
  - Problem 2: Data centre reliability problem

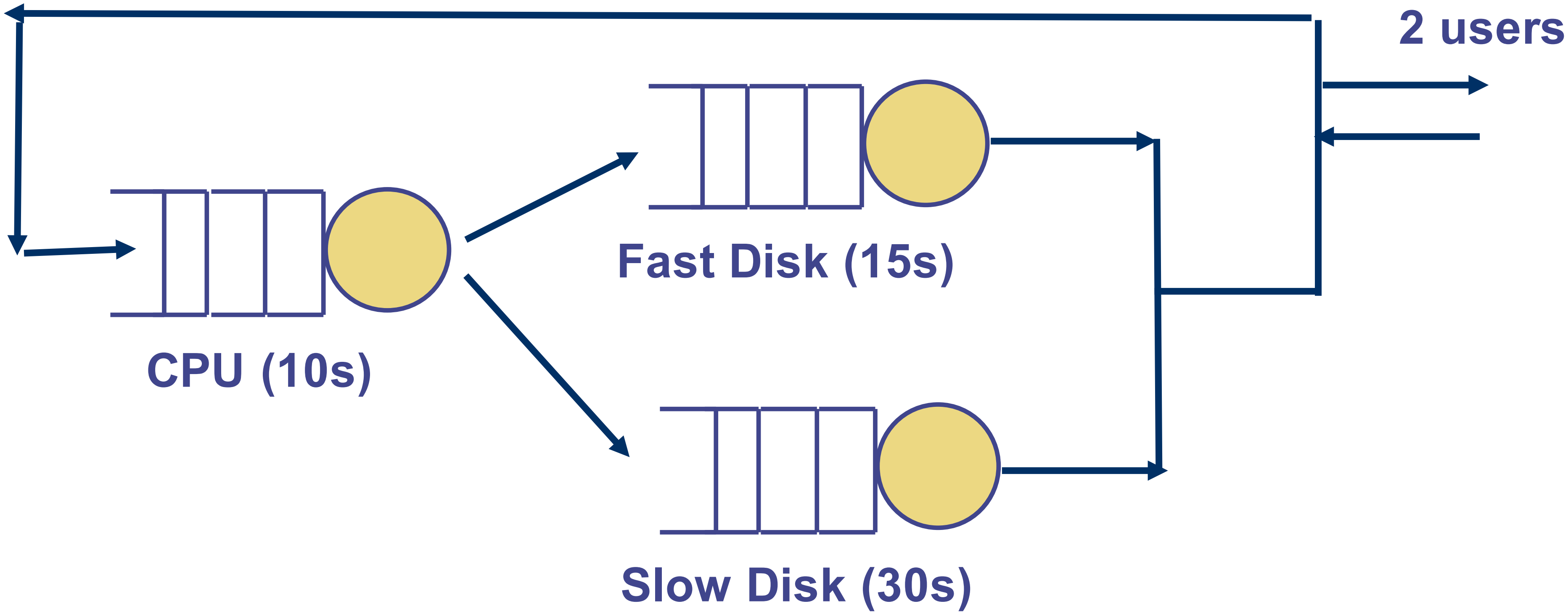
Problem 1: A DB server

- A database server with a CPU, a fast disk and a slow disk
- At peak demand, there are always two users in the system
- Transactions alternate between the CPU and the disks
- The transactions will equally likely find the file on either disk



Problem 1: A DB server (cont'd)

- Fast disk is twice as fast as the slow disk
- Typical transactions take on average 10s CPU time
- Fast disk takes on average 15s to serve all files for a transactions
- Slow disk takes on average 30s to serve all files for a transactions
- The time that each transaction requires from the CPU and the disks is exponentially distributed



Typical capacity planning questions

- What response time can a typical user expect?
- What is the utilisation of each of the system resources?
- How will performance parameters change if number of users are doubled?
- If fast disk fails and all files are moved to slow disk, what will be the new response time?

Choice of states #1

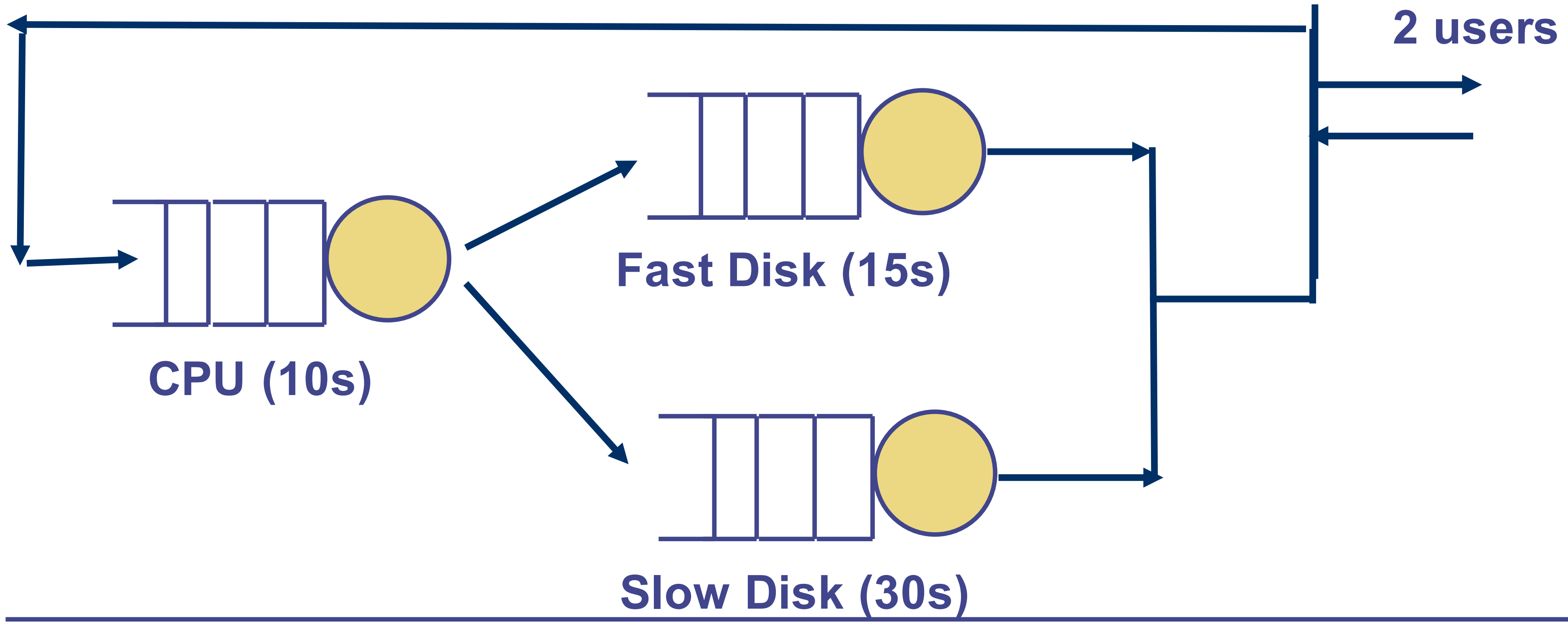
- Use a 2-tuple (A,B) where
  - A is the location of the first user
  - B is the location of the second user
  - A, B are drawn from {CPU,FD,SD}
    - FD = fast disk, SD = slow disk
  - Example states are:
    - (CPU,CPU): both users at CPU
    - (CPU, FD): 1st user at CPU, 2nd user at fast disk
  - Total 9 states
- If there are *n* users, how many states will you need?

Choice of states #2

- We use a 3-tuple (X,Y,Z)
  - X is # users at CPU
  - Y is # users at fast disk
  - Z is # users at slow disk
- Examples
  - (2,0,0): both users at CPU
  - (1,0,1): one user at CPU and one user at slow disk
- Six possible states
  - (2,0,0) (1,1,0) (1,0,1) (0,2,0) (0,1,1) (0,0,2)
- If there are n users, how many states do you need?

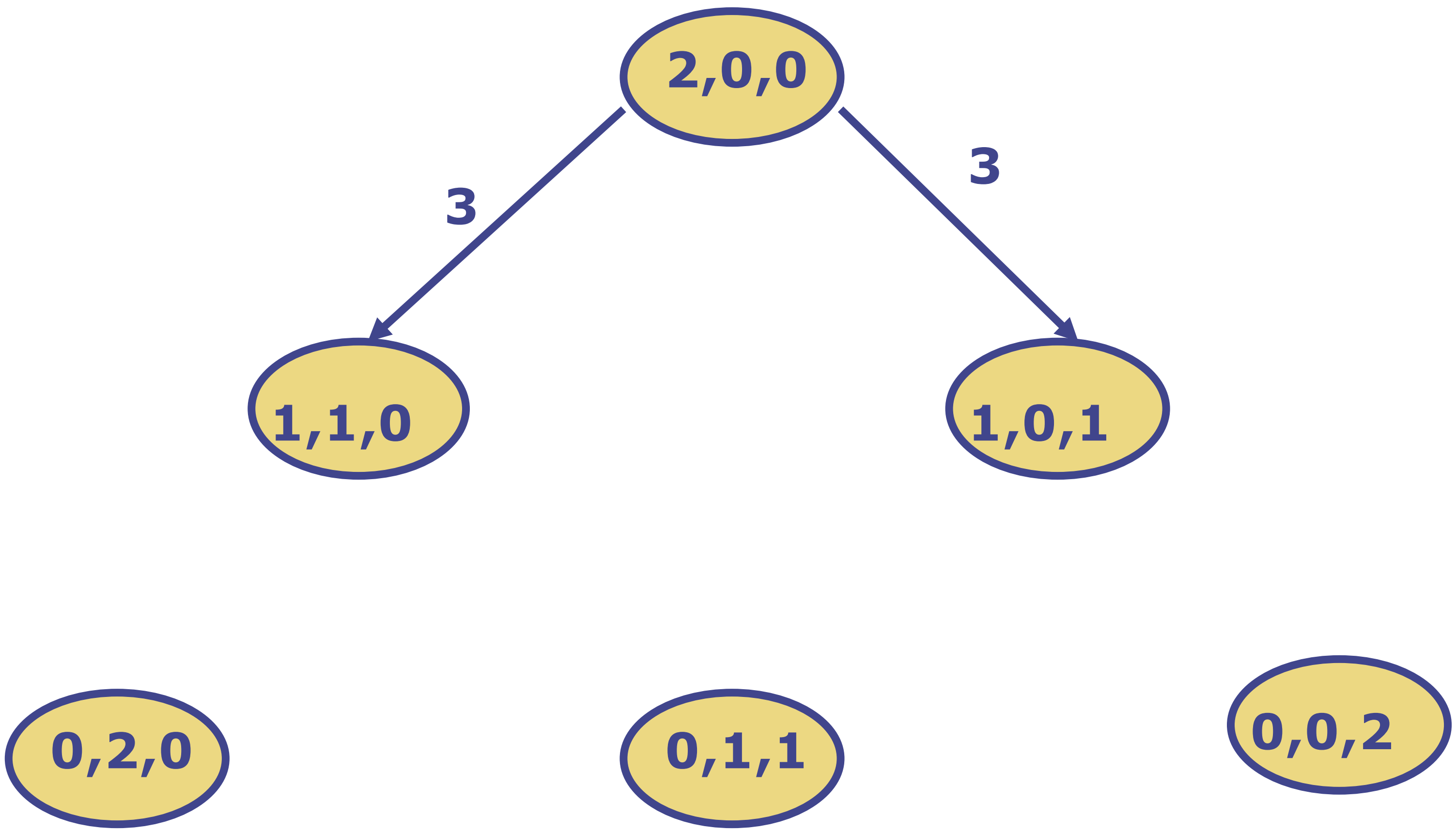
Identifying state transitions (1)

- A state is: (#users at CPU, #users at fast disk, #users at slow disk)
- What is the rate of moving from State (2,0,0) to State (1,1,0)?
  - This is caused by a job finishing at the CPU and move to fast disk
  - Jobs complete at CPU at a rate of 6 transactions/minute
  - Half of the jobs go to the fast disk
- Transition rate from (2,0,0) ➔ (1,1,0) = 3 transactions/minute
- Similarly, transition rate from (2,0,0) ➔ (1,0,1) = 3 transactions/minute



State transition diagram (2)

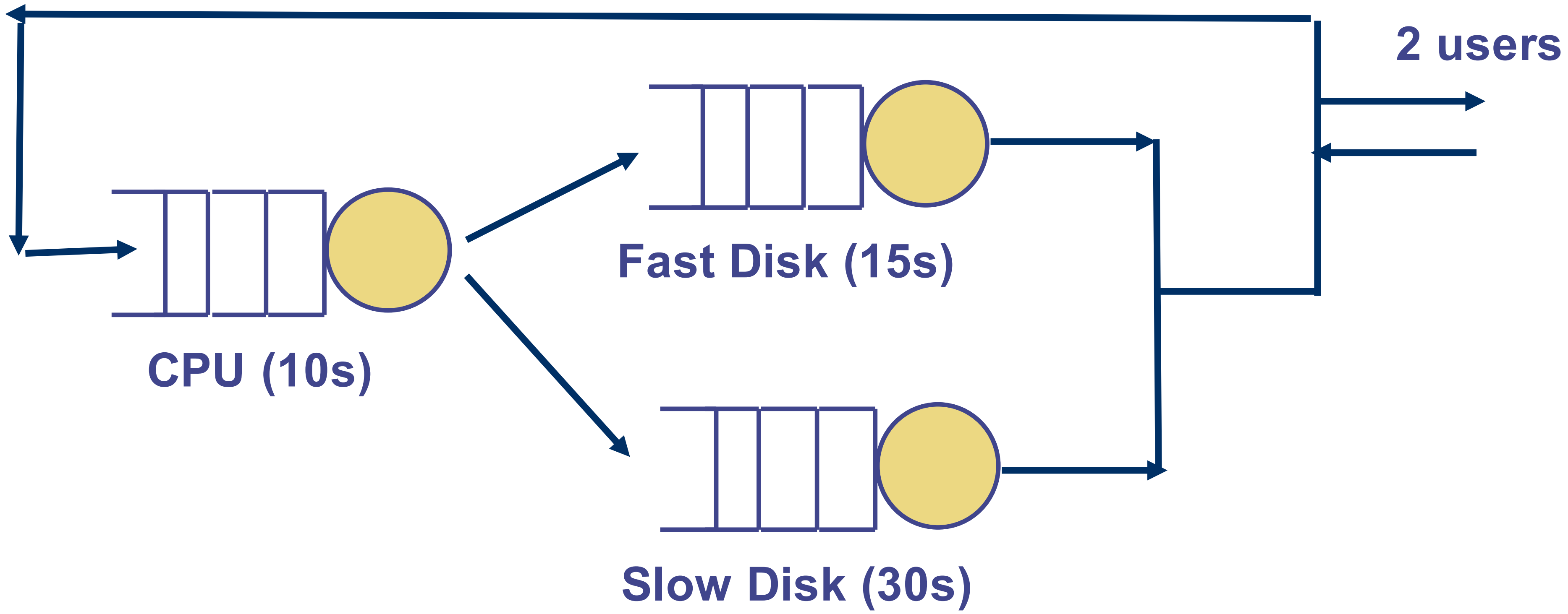
- Transition rate from (2,0,0) ➔ (1,1,0) = 3 transactions/minute
- Transition rate from (2,0,0) ➔ (1,0,1) = 3 transactions/minute



- Question: What is the transition rate from (2,0,0) ➔ (0,1,1)?

Identifying state transitions (2)

- From (1,1,0) there are 3 possible transitions
  - Fast disk user goes back to CPU (2,0,0)
  - CPU user goes to the fast disk (0,2,0), or
  - CPU user goes to the slow disk (0,1,1)
- Question: What are the transition rates in number of transactions per minute?



Complete state transition diagram

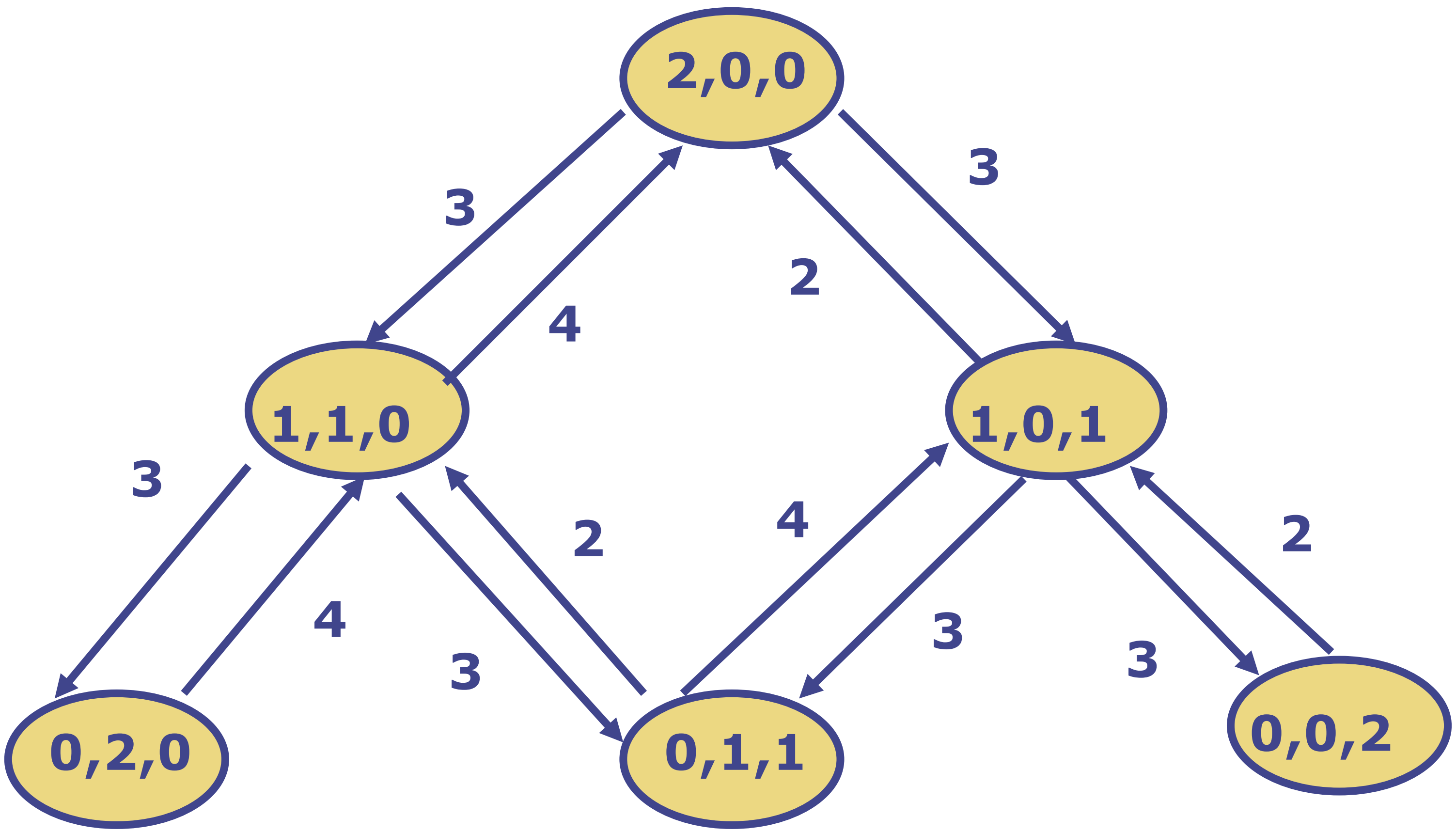
Balance Equations

Define

$P_{(2,0,0)}$ = Probability in state (2,0,0)

$P_{(1,1,0)}$ = Probability in state (1,1,0) etc.

Exercise: Write down the balance equation for state (2,0,0)



Flow balance equations

- You can write one flow balance equation for each state:

$$6\,P_{(2,0,0)} - 4\,P_{(1,1,0)} - 2\,P_{(1,0,1)} + 0\,P_{(0,2,0)} + 0\,P_{(0,1,1)} + 0\,P_{(0,0,2)} = 0$$

$$-3\,P_{(2,0,0)} + 10\,P_{(1,1,0)} + 0\,P_{(1,0,1)} - 4\,P_{(0,2,0)} - 2\,P_{(0,1,1)} + 0\,P_{(0,0,2)} = 0$$

$$-3\,P_{(2,0,0)} + 0\,P_{(1,1,0)} + 8\,P_{(1,0,1)} + 0\,P_{(0,2,0)} - 4\,P_{(0,1,1)} - 2\,P_{(0,0,2)} = 0$$

$$0\,P_{(2,0,0)} - 3\,P_{(1,1,0)} + 0\,P_{(1,0,1)} + 4\,P_{(0,2,0)} + 0\,P_{(0,1,1)} + 0\,P_{(0,0,2)} = 0$$

$$0\,P_{(2,0,0)} - 3\,P_{(1,1,0)} - 3\,P_{(1,0,1)} + 0\,P_{(0,2,0)} + 6\,P_{(0,1,1)} + 0\,P_{(0,0,2)} = 0$$

$$0\,P_{(2,0,0)} + 0\,P_{(1,1,0)} - 3\,P_{(1,0,1)} + 0\,P_{(0,2,0)} + 0\,P_{(0,1,1)} + 2\,P_{(0,0,2)} = 0$$

- However, there are only 5 linearly independent equations.
- Need one more equation:

Steady State Probability

- You can find the steady state probabilities from 6 equations
  - It's easier to solve the equations by a software packages, e.g
    - Matlab, Scilab, Octave, Excel etc.
    - See "Software" under course web page
- The solutions are:
  - $P_{(2,0,0)} = 0.1391$
  - $P_{(1,1,0)} = 0.1043$
  - $P_{(1,0,1)} = 0.2087$
  - $P_{(0,2,0)} = 0.0783$
  - $P_{(0,1,1)} = 0.1565$
  - $P_{(0,0,2)} = 0.3131$
- I used Matlab to solve these equations
  - The file is "dataserver.m" (can be downloaded from the course web site)
- How can we use these results for capacity planning?

Model interpretation

- Response time of each transaction
  - Use Little's Law $R = N/X$ with $N = 2$
    - For this system:
      - System throughput = CPU Throughput
    - Throughput = Utilisation x Service rate
      - Recall Utilisation = Throughput x Service time (From Lecture 2)
    - CPU utilisation (using states where there is a job at CPU): $P_{(2,0,0)} + P_{(1,1,0)} + P_{(1,0,1)} = 0.452$
    - Throughput = 0.452 x 6 = 2.7130 transactions / minute
    - Response time (with 2 users) = 2 /2.7126 = 0.7372 minutes per transaction

Sample capacity planning problem

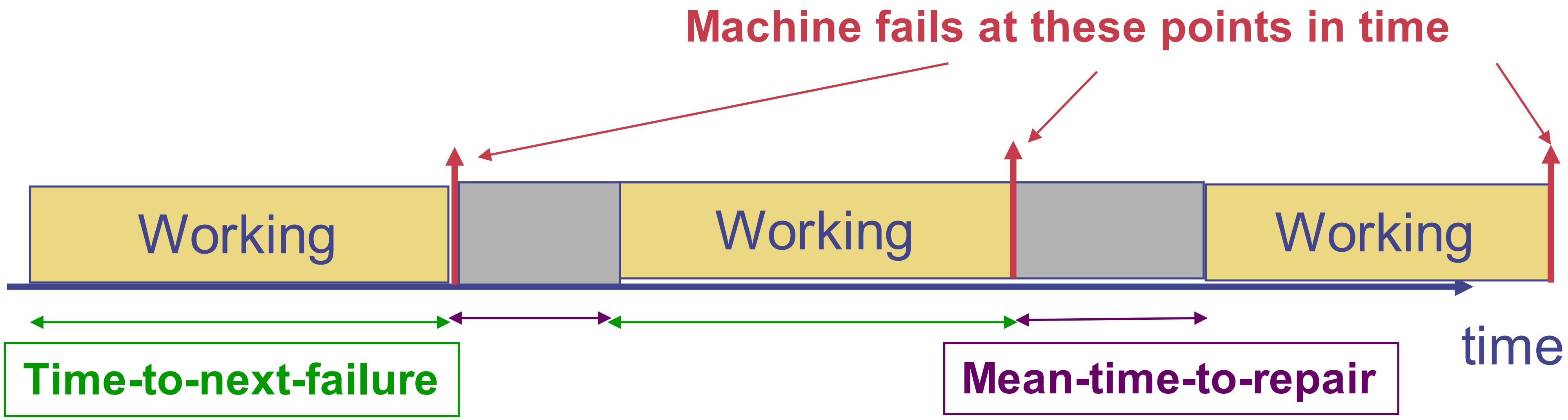
- What is the response time if the system have up to 4 users instead of 2 users only?
  - You can't use the previous Markov chain
  - You need to develop a new Markov chain
    - The states are again (#users at CPU, #users at fast disk, #users at slow disk)
    - States are (4,0,0), (3,1,0), (1,2,1) etc.
    - There are 15 states
    - Determine the transition rates
    - Write down the balance equations and solve them.
    - Use the steady state probabilities and Little's Law to determine the new response time
    - You can do this as an exercise
    - Throughput = 3.4768 (up 28%), response time = 60.03 seconds (up 56%)

Computation aspect of Markov chain

- This example shows that when there are a large number of users, the burden to build a Markov chain model is large
  - 15 states
  - Many transitions
  - Need to solve 15 equations in 15 unknowns
- Is there a faster way to do this?
  - Yes, we will look at Mean Value Analysis in a few weeks and it can obtain the response time much more quickly

Reliability problem using Markov chain

- Consider the working-repair cycle of a machine
- "Failure" is an arrival to the repair workshop
- "Repair" time is the service time to repair the machine
- Let us assume
  - "Time-to-next-failure" and "Repair time" are exponentially distributed



- Note: Mean-time-to-repair includes waiting (or queueing) time for repair and actual time under repair

Data centre reliability problem

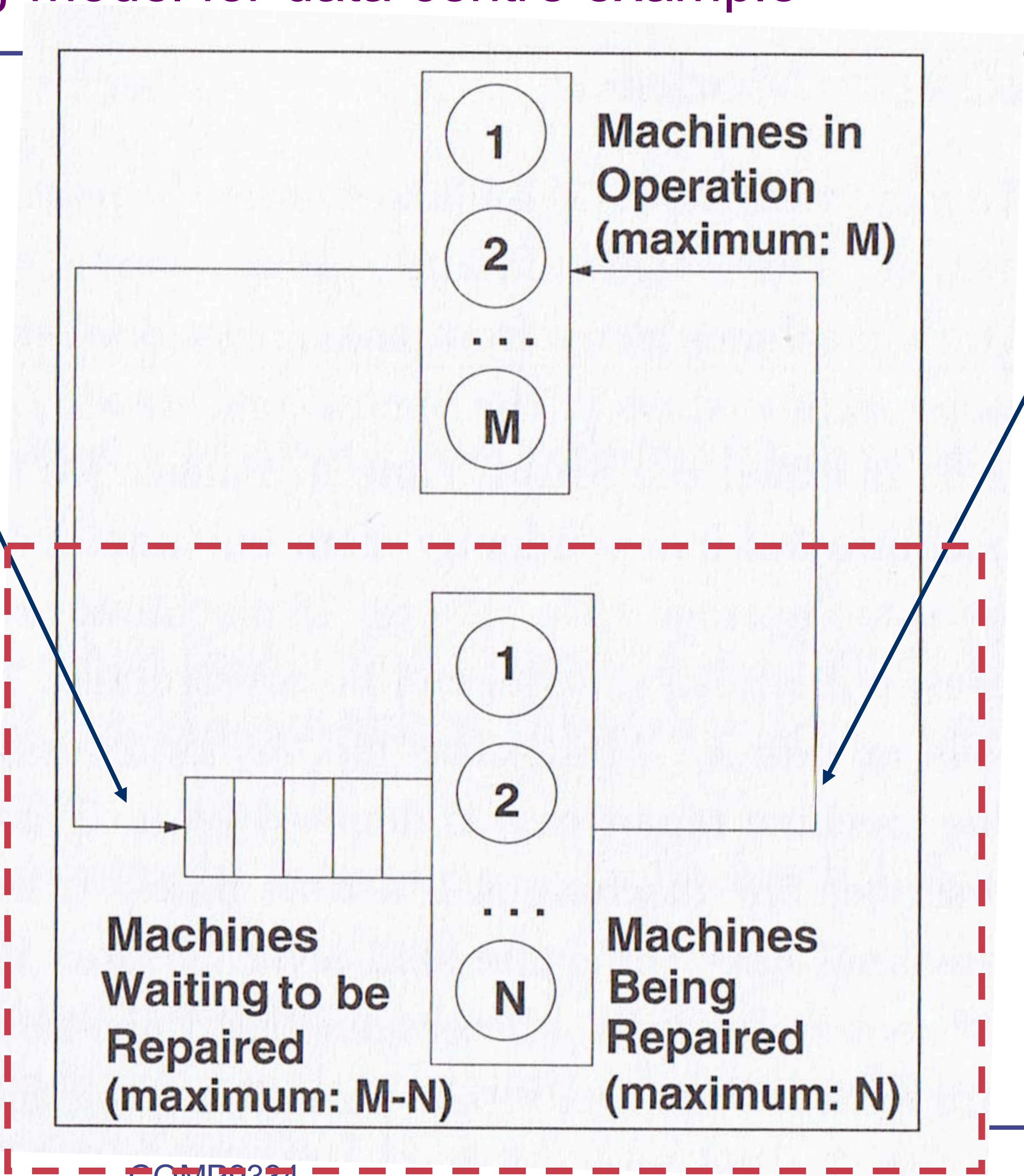
- Example: A data centre has 10 machines
  - Each machine may go down
    - Time-to-next-failure is exponentially distributed with mean 90 days
  - Repair time is exponentially distributed with mean 6 hours
- Capacity planning question:
    - Can I make sure that at least 8 machines are available 99.9999% of the time?
    - What is the probability that at least *6* machines are available?
    - How many repair staff are required to guarantee that at least *k* machines are available with a given probability?
    - What is the mean time to repair (MTTR) a machine?
      - Note: Mean-time-to-repair includes waiting time at the repair queue.

Data centre reliability - general problem

- Data centre has
  - *M* machines
  - *N* staff maintain and repair machine
  - Assumption: $M > N$
- Automatic diagnostic system
  - Check "heartbeat" by "ping" (Failure detection)
  - Staff are informed if failure is detected
- Repair work
  - If a machine fails, any one of the idle repair staff (if there is one) will attend to it.
  - If all repair staff are busy, a failed machine will need to wait until a repair staff has finished its work
- This is a queueing problem solvable by Markov chain!!!
- Let us denote
  - ***λ = 1 / Mean-time-to-failure***
  - **μ = 1/ Mean repair time**

Queueing model for data centre example

An arrival is due to a machine failure.

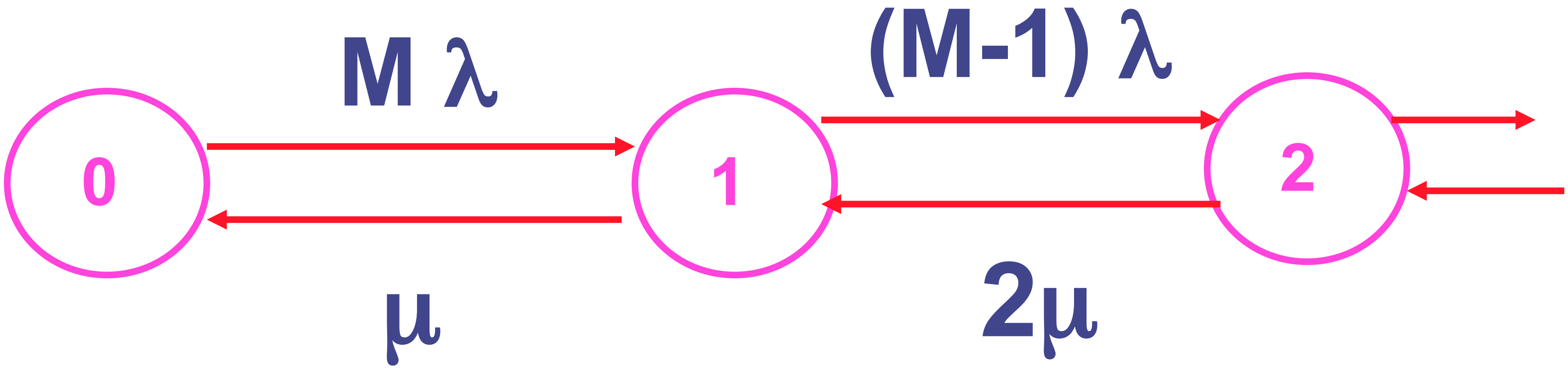


A departure occurs when a machine has been repaired.

We build a Markov chain for this box.

Markov model for the repair queue

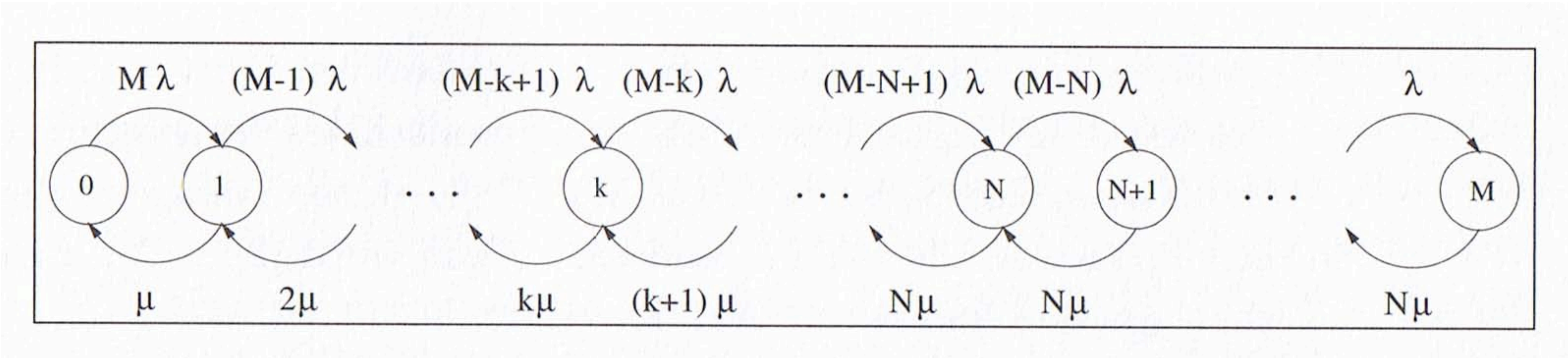
- State *k* represents *k* machines have failed
- Part of the state transition diagram is showed below



The rate of failure for one machine is λ. In State 0, there are M working machine, the failure rate is Mλ.

The same argument holds for other state transition probability.

Markov Model for the repair queue



Note: There are only (M+1) states.

Why is it Nμ?

Why not (N+1)μ?

Solving the model

- We can solve for P(0), P(1), …, P(M)

$$P(k) = \begin{cases} P(0)(\frac{\lambda}{\mu})^k C_k^m & k = 1, \ldots, N \\ P(0)(\frac{\lambda}{\mu})^k C_k^m \frac{N^{N-k} k!}{N!} & k = N+1, \ldots M \end{cases}$$

Where

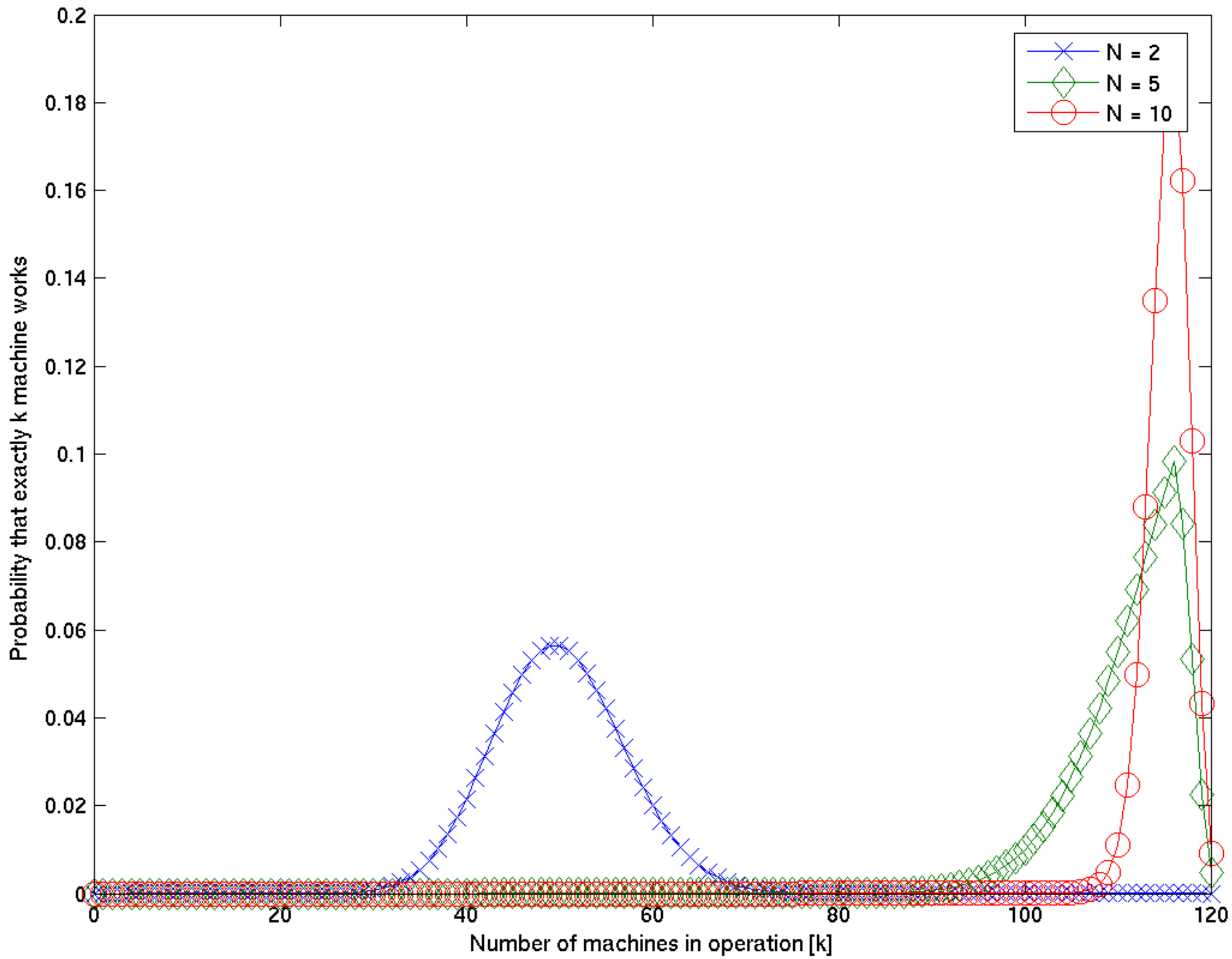
$$P(0) = \left[\sum_{k=0}^{N} (\frac{\lambda}{\mu})^k C_k^m + \sum_{k=N+1}^{M} (\frac{\lambda}{\mu})^k C_k^m \frac{N^{N-k} k!}{N!} \right]^{-1}$$

Using the model

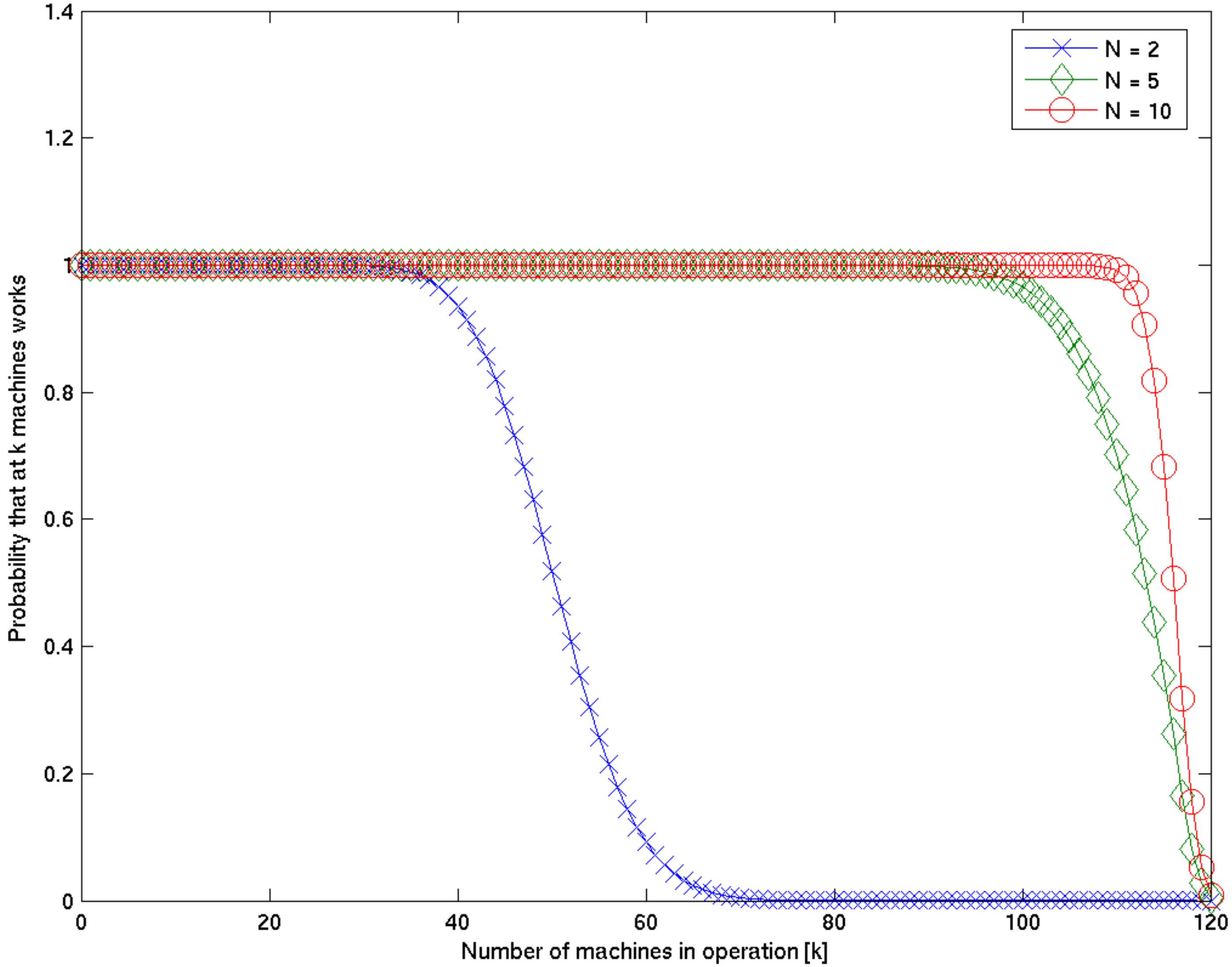
- Probability that exactly *k* machines are available =
- Probability that at least *k* machines are available
  =
- But expression for P(*k*)'s are complicated, need numerical software

- Example:
  - M = 120
  - Mean-time-to-failure = 500 minutes
  - Mean repair time = 20 minutes
  - N = 2, 5 or 10
  - The results are showed in the graphs in the next 2 pages
    - I used the file "data_centre.m" to do the computation, the file is available on the course web site.

Probability that exactly k machines operate



Probability that at least k machines operate

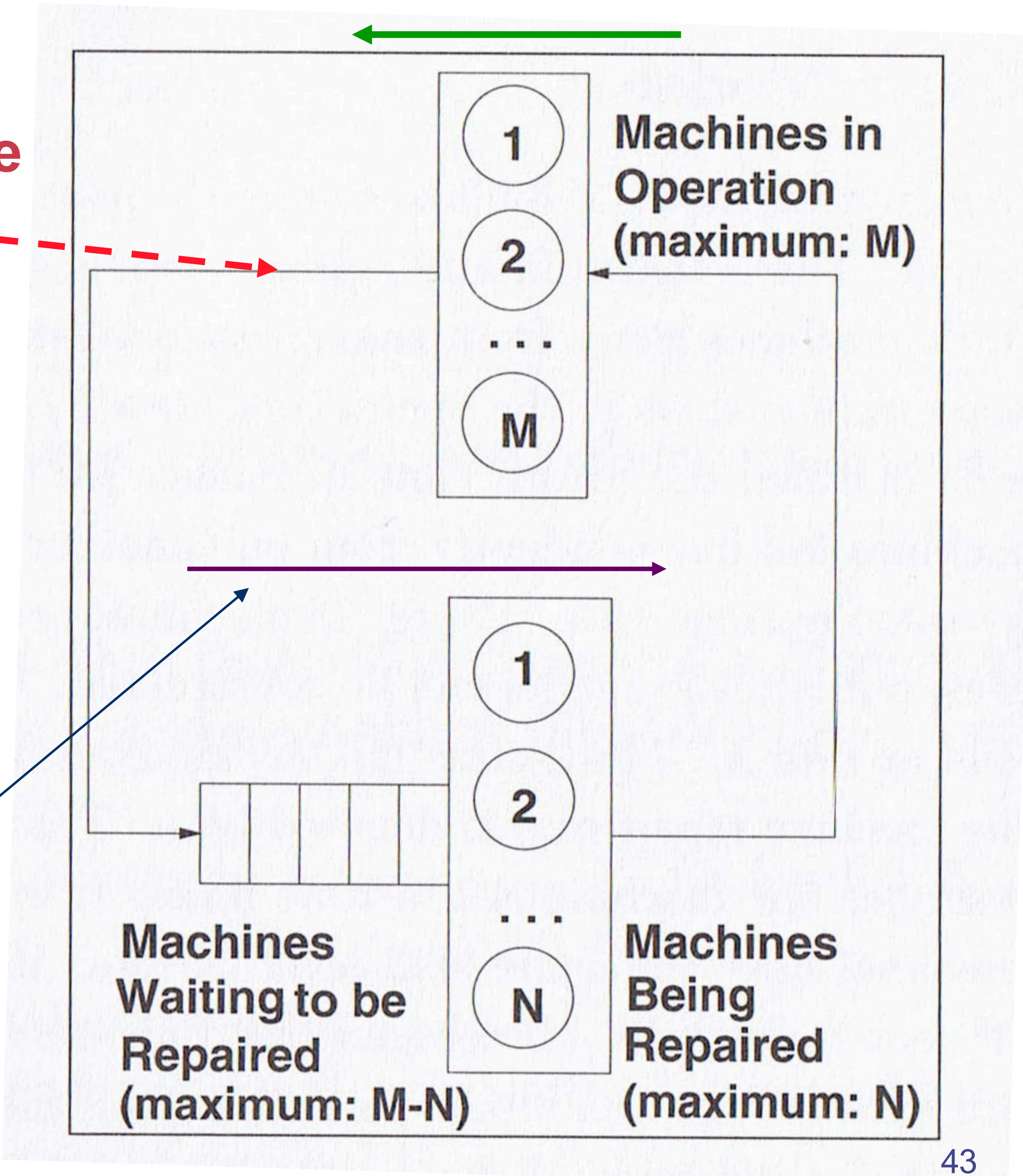


Think time ~ Mean-time-to-failure (MTTR) = 1 / λ

Throughput
~ Mean machine failure rate
(see next page)

Mean time to repair (MTTR)
= Queueing time for repair + actual repair time

Can compute MTTR using Little's Law.



Mean machine failure rate

State	Probability	Failure rate
0	P(0)	$M\lambda$
1	P(1)	$(M-1)\lambda$
2	P(2)	$(M-2)\lambda$
…	…	
k	P(k)	$(M-k)\lambda$
…	…	
M	P(M)	0

$$\bar{X}_f = \sum_{k=0}^{M-1} (M-k)\lambda P(k)$$

Continuous-time Markov chain

- Useful for analysing queues when the inter-arrival or service time distribution are exponential
- The procedure is fairly standard for obtaining the steady state probability distribution
  - Identify the state
  - Find the state transition rates
  - Set up the balance equations
  - Solve the steady state probability
- We can use the steady state probability to obtain other performance metrics: throughput, response time etc.
  - May need Little's Law etc.
- Continuous-time Markov chain is only applicable when the underlying probability distribution is exponential but the operations laws (e.g. Little's Law) are applicable no matter what the underlying probability distributions are.

References

- Recommended reading
  - The database server example is taken from Menasce et al., "Performance by design", Chapter 10
  - The data centre example is taken from Mensace et al, "Performance by desing", Chapter 7, Sections 1-4
- For a more in-depth, and mathematical discussion of continuous-time Markov chain, see
  - Alberto Leon-Gracia, "Probabilities and random processes for Electrical Engineering", Chapter 8.
  - Leonard Kleinrock, "Queueing Systems", Volume 1
- For mathematical software that you can use to solve a set of linear equations or do numerical calculations, go to the course web site and click on "Software".