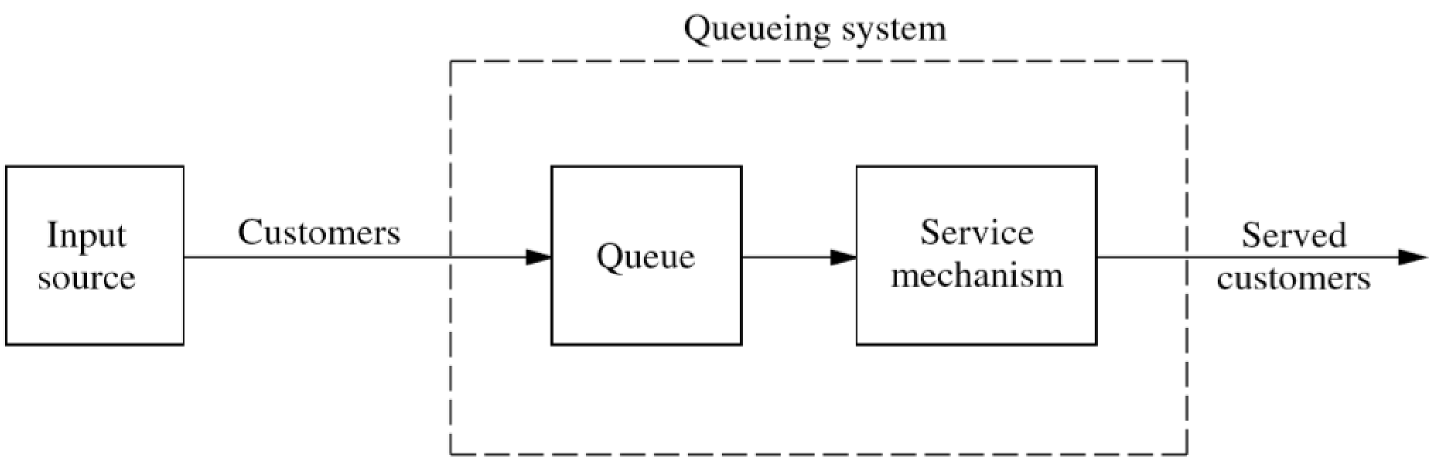
команда **МетаЛЛург**:

- Илья Седунов
- Вадим Альперович
- Анастасия Кислицына
- Максим Куклин

Single-Server-Queueing-System-With-Limited-Capacity

I. Математическая модель M/M/1/N



Математическая модель $M/M/1/N$

N - размер очереди,

λ - интенсивность входного потока,

μ - интенсивность обслуживания

Вероятностное пространство

$(\Omega, \mathcal{F}, \mathcal{P}(\bullet))$, *где*

$$\mathcal{F} = 2^{\Omega}$$

$\Omega = \{\omega = (\omega_1, \omega_2, \omega_3, \dots)\}$, *где*

$\omega_i \in \{0, 1, 2, \cdots, N\}$ *- состояние системы (кол-во требований в системе)*

в i-тый момент изменения системы

$$P(\omega_{i+1} = \xi_{i+1} | \omega_i = \xi_i, \omega_{i-1} = \xi_{i-1}, \dots, \omega_1 = \xi_1) = P(\omega_{i+1} = \xi_{i+1} | \omega_i = \xi_i)$$

$$P(\omega_{i+1} = 1 | \omega_i = 0) = 1$$

$$P(\omega_{i+1} = N-1 | \omega_i = N) = 1$$

$$P(\omega_{i+1} = a+1 | \omega_i = a) = \frac{\lambda}{\lambda+\mu},\ 0 < a < N$$

$$P(\omega_{i+1} = a-1 | \omega_i = a) = \frac{\mu}{\lambda+\mu},\ 0 < a < N$$

Так как:

Пусть $X \sim exp(\lambda)$, $Y \sim exp(\mu)$

Тогда $P(X < Y) = \int_0^{\infty} P(X < y) f_y(y) dy =$

$$= \int_0^{\infty} (1 - e^{-\lambda y}) \mu e^{-\mu y} dy =$$

$$= \mu \int_0^{\infty} (1 - e^{-\lambda y}) dy - \mu \int_0^{\infty} e^{-(\lambda+\mu) y} dy =$$

$$= -\int_0^\infty (1 - e^{-\mu y}) d(-\mu y) - \frac{\mu}{\mu+\lambda} \int_0^\infty e^{-(\lambda+\mu)y} dy =$$

$$= 1 - \frac{\mu}{\lambda+\mu} = \frac{\lambda}{\lambda+\mu}.$$

Зададим случайные величины

$$X_0 = 0 \qquad X_i(\omega) = \omega_i$$

$$P(X_{i+1} = x_{i+1} | X_i = x_i, X_{i-1} = x_{i-1}, \ldots, X_1 = x_1, X_0 = x_0) = P(X_{i+1} = x_{i+1} | X_i = x_i)$$

$$P_0(X_0 = 0) = 1$$

$$\mathbb{P} = \begin{array}{cc} & \begin{array}{ccccccc} 0 & 1 & 2 & \cdots & \cdots & N-1 & N \end{array} \\ \begin{array}{c} 1 \\ 2 \\ \vdots \\ \vdots \\ N-1 \\ N \end{array} & \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ \frac{\mu}{\lambda+\mu} & 0 & \frac{\lambda}{\lambda+\mu} & 0 & \cdots & 0 & 0 \\ \cdots & \frac{\mu}{\lambda+\mu} & 0 & \frac{\lambda}{\lambda+\mu} & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{\lambda}{\lambda+\mu} \\ 0 & \cdots & \cdots & \cdots & \cdots & 1 & 0 \end{pmatrix} \end{array}$$

Схема состояний

Уравнения баланса (RateIn = RateOut)

$0 : p_1\mu = p_0\lambda$
$1 : p_0\lambda + p_2\mu = p_1\lambda + p_1\mu$
$2 : p_1\mu = p_0\lambda$
$\dots$
$N-1 : p_{N-2}\lambda + p_N\mu = p_{N-1}(\lambda+\mu)$
$N : p_{N-1}\lambda = p_N\mu$

$p_i = (\frac{\lambda}{\mu})^i p_0$

$\Sigma_{i=0}^{N} p_i = 1$

$\Sigma_{i=0}^{N} (\frac{\lambda}{\mu})^i p_0 = 1$

Получаем

При $\lambda \neq \mu$:

$$p_0 = \frac{1}{\Sigma_{i=0}^{N}(\frac{\lambda}{\mu})^i} = \frac{1-\frac{\lambda}{\mu}}{1-\frac{\lambda}{\mu}^{N+1}}$$

$$p_i = (\frac{\lambda}{\mu})^i \frac{1-\frac{\lambda}{\mu}}{1-(\frac{\lambda}{\mu})^{N+1}}, \ \ \forall i = \overline{0, N}$$

При $\lambda = \mu$:

$$p_0 = p_i = \frac{1}{N+1}, \ \ \forall i = \overline{0, N}$$

1. Вероятность отказа требованию

Вероятность отказа равна вероятности пребывания системы в состоянии N

***При** $\lambda \neq \mu$**:*

$$P_n = (\frac{\lambda}{\mu})^N \frac{1-\frac{\lambda}{\mu}}{1-(\frac{\lambda}{\mu})^{N+1}}$$

***При** $\lambda = \mu$**:*

$$P_n = \frac{1}{N+1}$$

2.Среднее количество требований в системе

***При** $\lambda \neq \mu$**:*

***пусть** $\rho = \frac{\lambda}{\mu}$*

$$E(X) = \Sigma_{i=0}^{N} i P_i = \Sigma_{i=0}^{N} i \frac{1-\rho}{1-\rho^{N+1}} \rho^i =$$

$$= \frac{1-\rho}{1-\rho^{N+1}} \rho \Sigma_{i=0}^{N} \frac{d}{d\rho}(\rho^i) = \frac{1-\rho}{1-\rho^{N+1}} \rho \frac{d}{d\rho}(\Sigma_{i=0}^{N} \rho^i) =$$

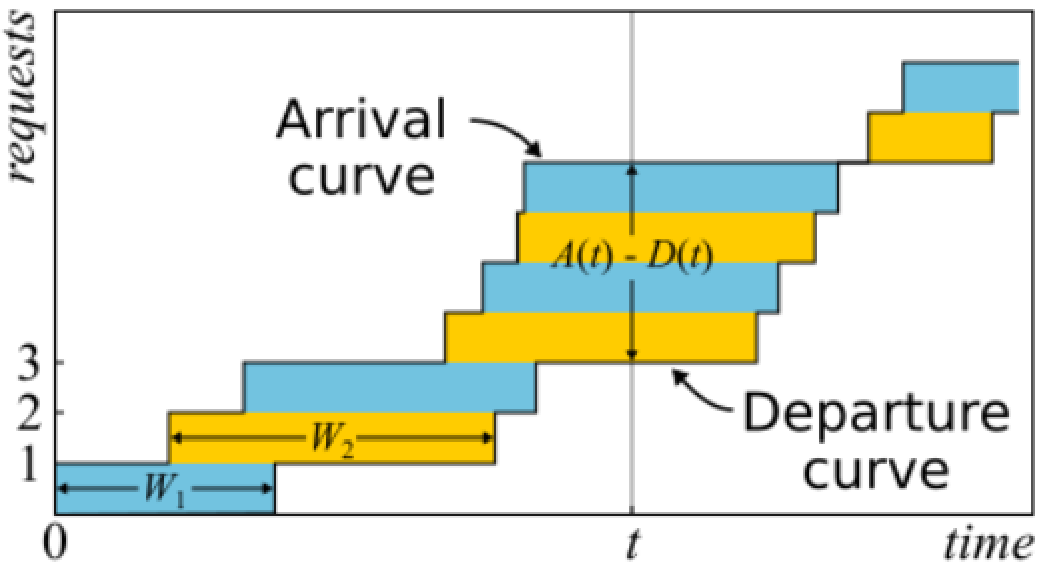
$$= \frac{1-\rho}{1-\rho^{N+1}} \rho \frac{d}{d\rho}(\frac{1-\rho^{N+1}}{1-\rho}) = \frac{1-\rho}{1-\rho^{N+1}} \rho \frac{-N\rho^{N-1}+N\rho^N-\rho^N+1}{(1-\rho)^2} =$$

$$= \rho \frac{-(N+1)\rho^N+N\rho^{N+1}+1}{(1-\rho^{N+1})(1-\rho)} = \frac{\rho}{1-\rho} - \frac{(N+1)\rho^{N+1}}{1-\rho^{N+1}}.$$

***При** $\lambda = \mu$**:*

$$E(X) = \Sigma_{i=0}^{N} i P_i = \frac{1}{N+1} \Sigma_{i=0}^{N} i = \frac{1}{N+1} \frac{(N+1)N}{2} = \frac{N}{2}$$

3. Среднее время пребывания одного требования в системе



$$\frac{\Sigma W_i}{\Delta t} = \frac{N_t}{\Delta t}\frac{\Sigma W_i}{N_t} \Rightarrow E(X) = \bar{\lambda} E(T)$$ ***, где***

$\bar{\lambda} = \lambda(1 - P_n)$***, т.к. очередь ограниченной длины***

Получаем, что среднее время пребывания в системе = среднее кол-во требований в системе / интенсивность входного потока (Закон Литтла)

Следовательно,

при $\lambda \neq \mu$***:***

$$E(T) = \frac{1}{\bar{\lambda}} E(X) = \frac{1}{\lambda(1-P_n)}\left(\frac{\rho}{1-\rho} - \frac{(N+1)\rho^{N+1}}{1-\rho^{N+1}}\right)$$

при $\lambda = \mu$***:***

$$E(T) = \frac{1}{\bar{\lambda}} E(X) = \frac{1}{\lambda(1-P_n)}\frac{N}{2} = \frac{1}{\lambda\frac{N}{N+1}}\frac{N}{2} = \frac{N+1}{2\lambda}$$

4. Вероятность ожидания требования

Вероятность ожидания требования равна тому, что в системе находится как минимум 1 требование

При $\lambda \neq \mu$***:***

$$P_{wait} = 1 - P_0 = 1 - \frac{1-\frac{\lambda}{\mu}}{1-(\frac{\lambda}{\mu})^{N+1}}$$

При $\lambda = \mu$***:***

$$P_{wait} = 1 - P_0 = 1 - \frac{1}{N+1} = \frac{N}{N+1}$$

II. Реализация модели

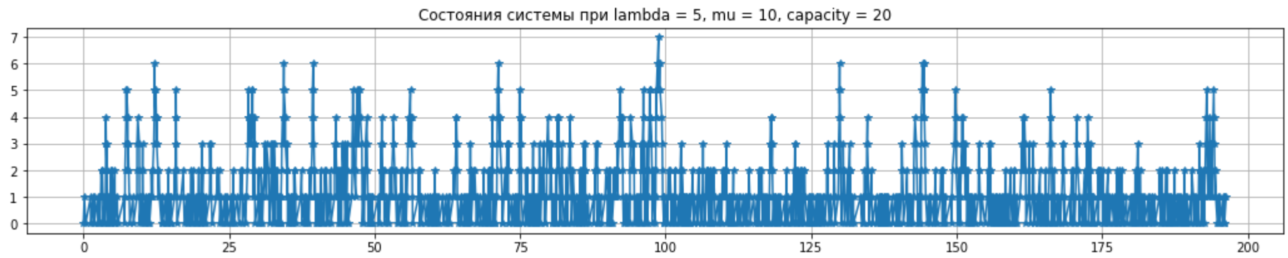
Имплементация [*https://github.com/VirtualRoyalty/Single-Server-Queueing-System*](https://github.com/VirtualRoyalty/Single-Server-Queueing-System) *(https://github.com/VirtualRoyalty/Single-Server-Queueing-System)*

In [18]:
```
import importlib

import matplotlib.pyplot as plt
import numpy as np

import prototype_v2 as simulator
from scripts import TheoryScripts as theory

import warnings
warnings.filterwarnings('ignore')

import importlib
plt = importlib.reload(plt)

simulator = importlib.reload(simulator)
theory = importlib.reload(theory)
```

In [4]:
```
lambd = [5, 15, 10, 5]
mu = [10, 10, 10, 100 ]
capacity = [20, 100, 400, 600]
iterations = 1000
```

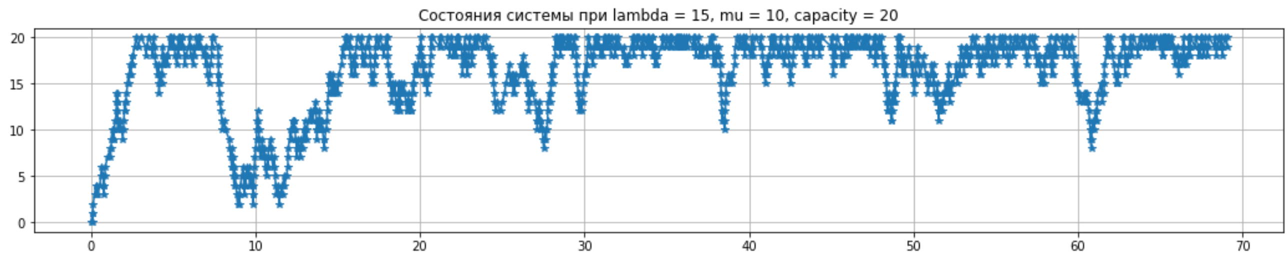
1. Состояния системы

In [151]:
```
i_case = 0
plt.figure(figsize=(18,3))
simulator.vis_num_of_clients_in_time(iterations=iterations,
                                     lambd=lambd[i_case],
                                     mu=mu[i_case],
                                     capacity=capacity[0])
```



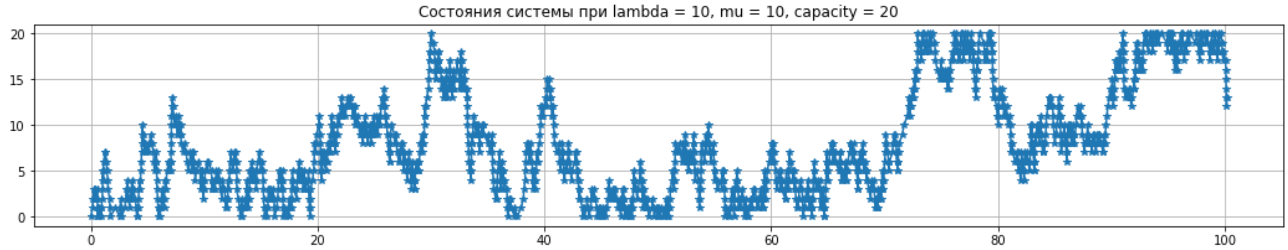

```
Lost = 0
Mean time inside = 0.17325213379430943
Mean waiting time  = 0.07830223035807928
Mean lost = 0.0
1.345
Mean state = 1.345
```

In [152]:
```
i_case = 1
plt.figure(figsize=(18,3))
simulator.vis_num_of_clients_in_time(iterations=iterations,
                                     lambd=lambd[i_case],
                                     mu=mu[i_case],
                                     capacity=capacity[0])
```




```
Lost = 265
Mean time inside = 1.5639542664699753
Mean waiting time  = 1.4687090649582248
Mean lost = 0.265
15.966230186078567
Mean state = 15.966230186078567
```

In [154]:
```
i_case = 2
plt.figure(figsize=(18,3))
simulator.vis_num_of_clients_in_time(iterations=iterations,
                                      lambd=lambd[i_case],
                                      mu=mu[i_case],
                                      capacity=capacity[0])
```

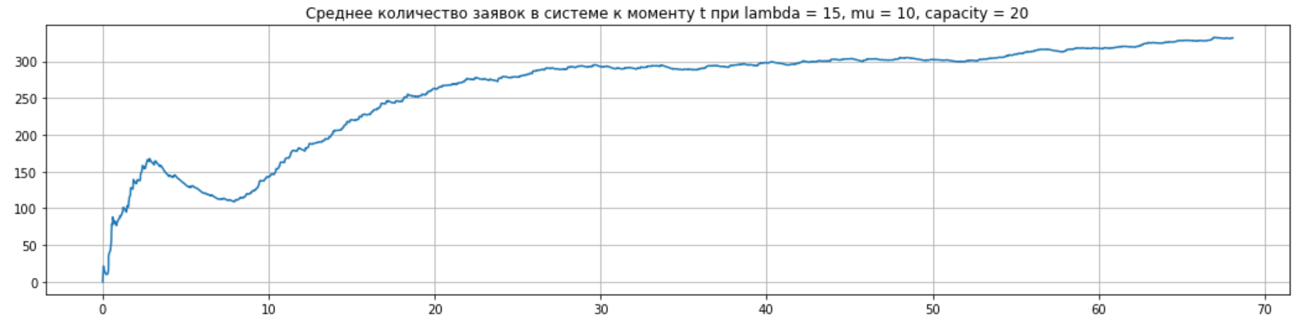



```
Lost = 52
Mean time inside = 0.8519540242911713
Mean waiting time  = 0.7519324986508873
Mean lost = 0.052
8.2484076433121
Mean state = 8.2484076433121
```

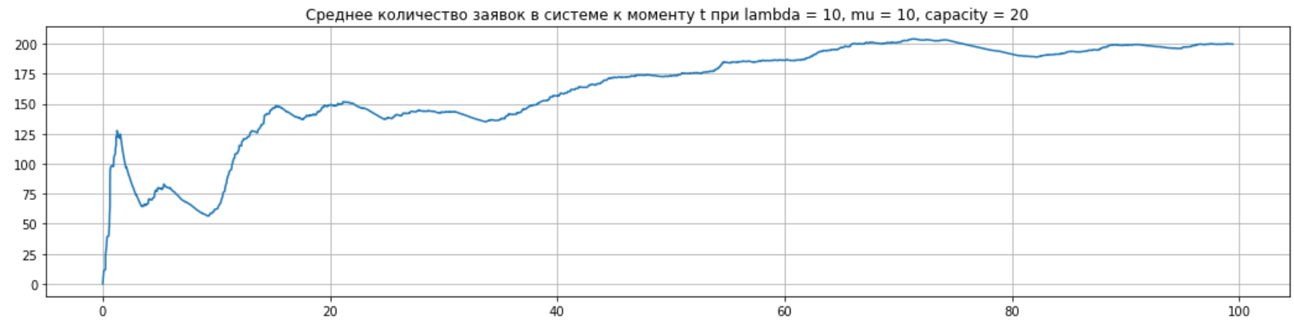
2. Среднее количество заявок в системе к времени t

In [155]:
```
i_case = 0
plt.figure(figsize=(18,4))
ox, oy = simulator.vis_mean_of_clients_in_time(iterations=iterations,
                                                lambd=lambd[i_case],
                                                mu=mu[i_case],
                                                capacity=capacity[0])
```

In [156]:
```
i_case = 1
plt.figure(figsize=(18,4))
ox, oy = simulator.vis_mean_of_clients_in_time(iterations=iterations,
                                               lambd=lambd[i_case],
                                               mu=mu[i_case],
                                               capacity=capacity[0])
```



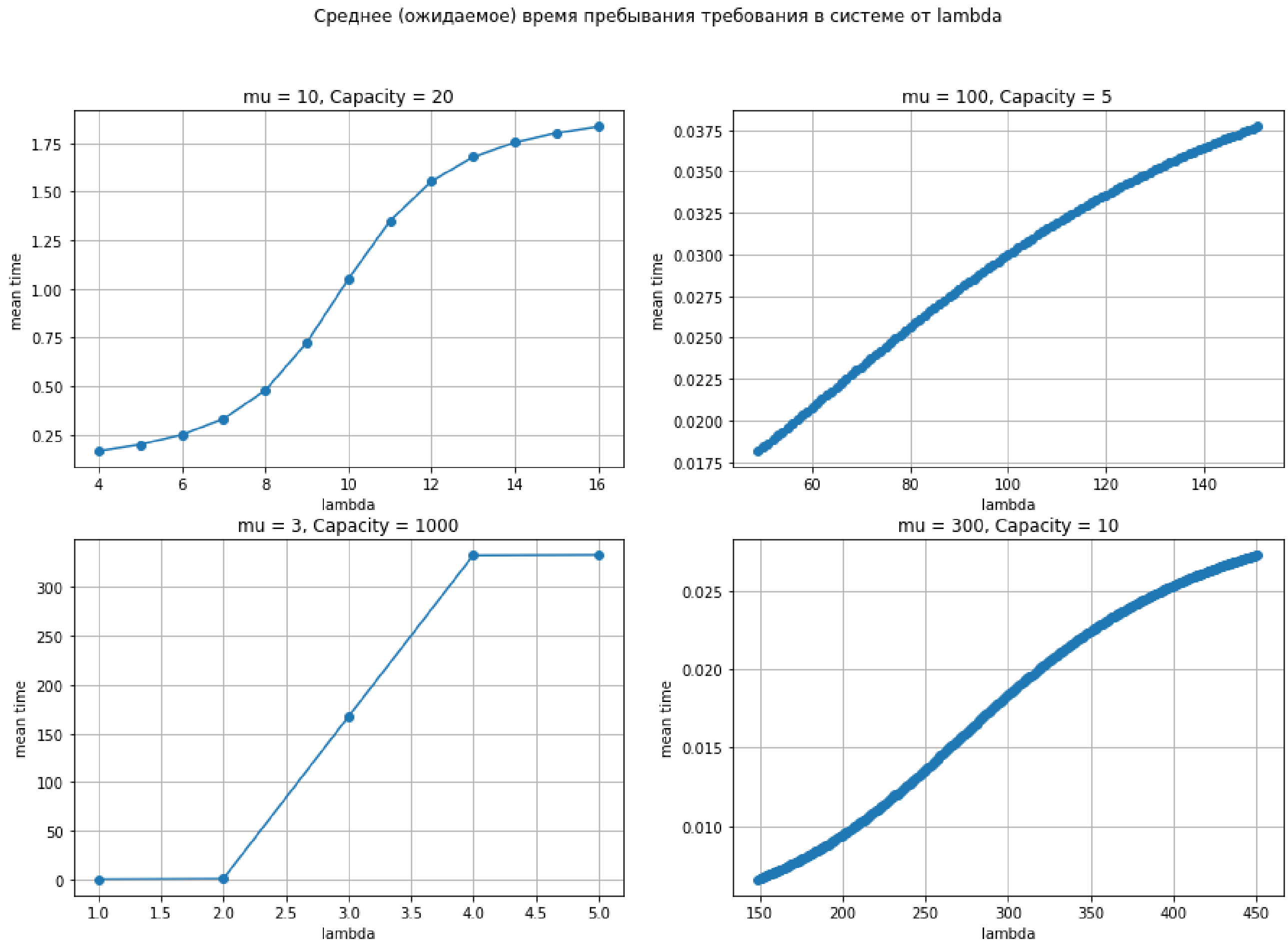
In [157]:
```
i_case = 2
plt.figure(figsize=(18,4))
ox, oy = simulator.vis_mean_of_clients_in_time(iterations=iterations,
                                               lambd=lambd[i_case],
                                               mu=mu[i_case],
                                               capacity=capacity[0])
```



3. Среднее время пребывания требования в системе

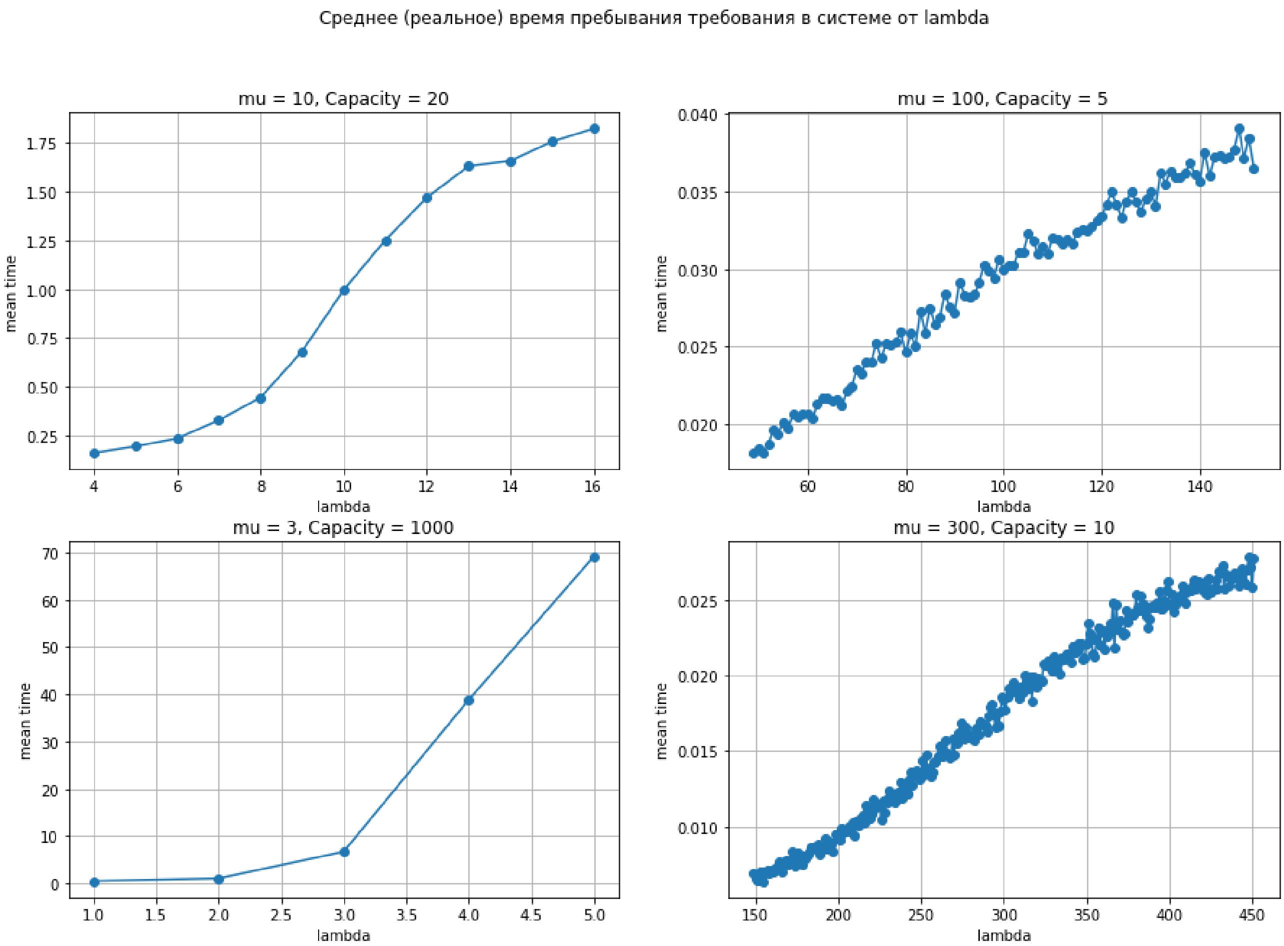
3.1 Теоретическое ожидание

In [197]:
```python
simulator.draw(message='Среднее (ожидаемое) время пребывания требования в системе от lambda',
               func_from_class=theory.get_average_time,
               ytitle='mean time')
```



3.2 Эксперимент

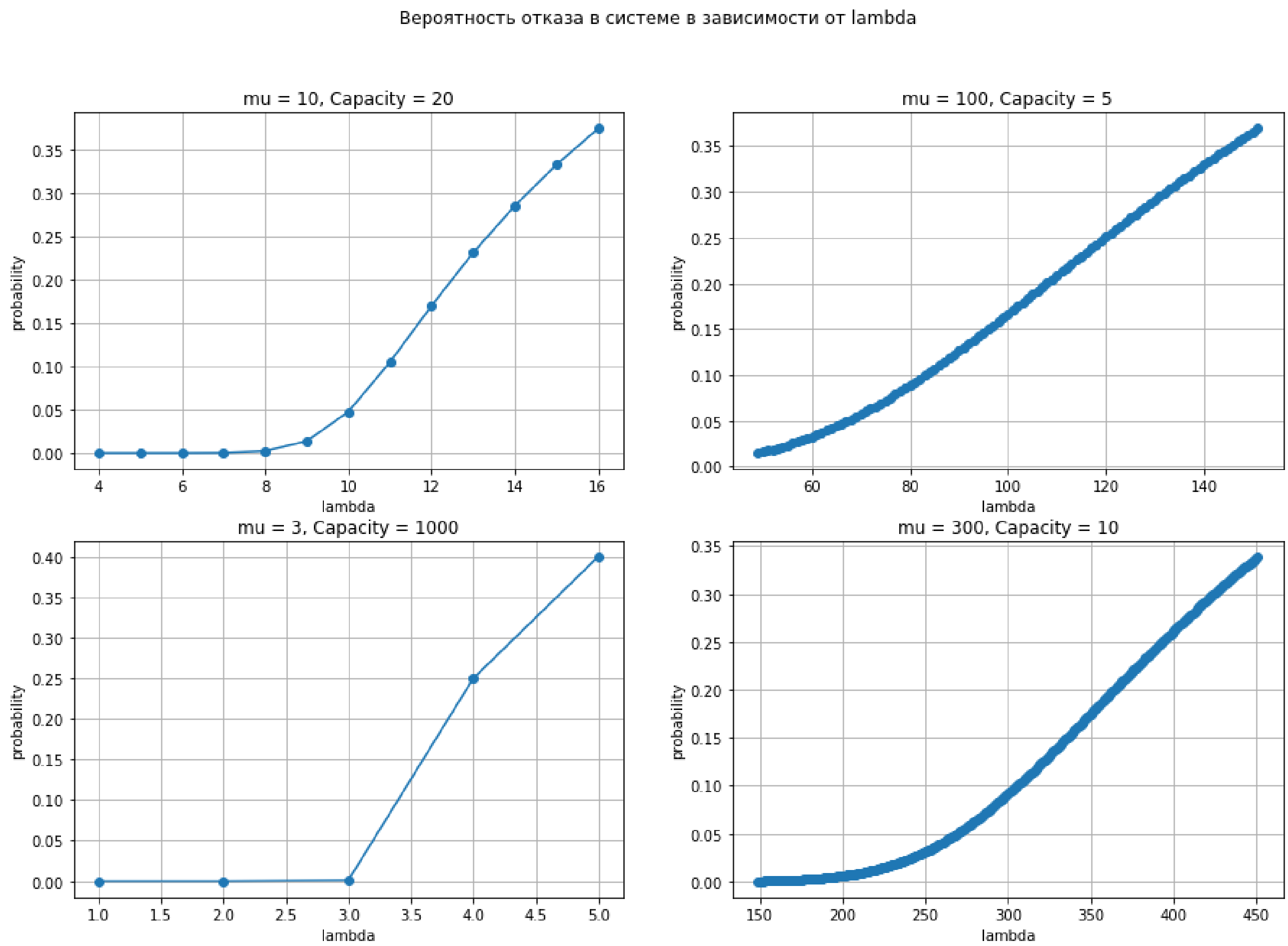
In [196]:
```
simulator.draw(message='Среднее (реальное) время пребывания требования в системе от lambda',
               func_from_class=simulator.mean_sojourn_time,
               ytitle='mean time')
```



4. Вероятность отказа требованию в системе

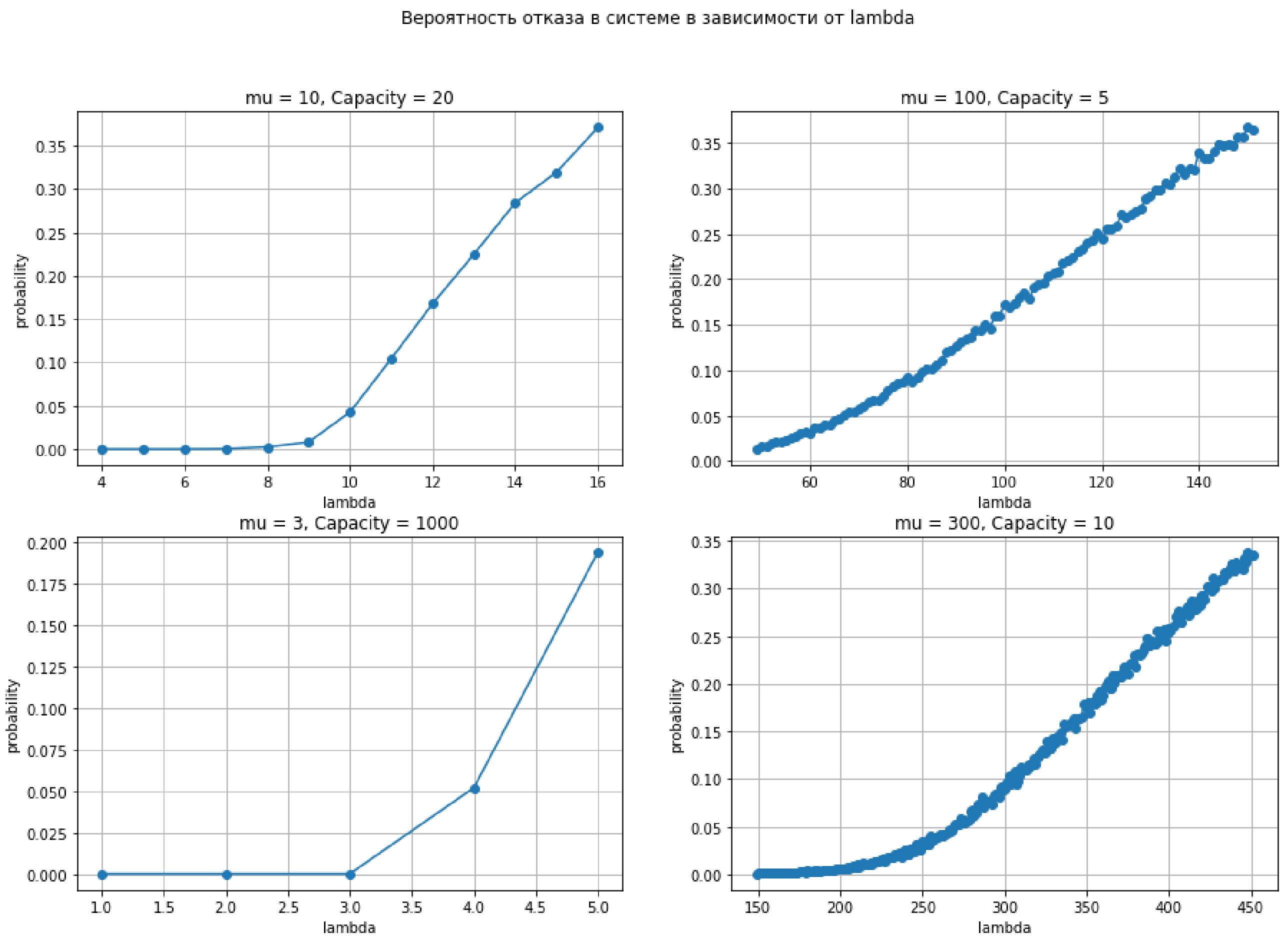
4.1 Теоретическое ожидание

In [202]:
```
simulator.draw(message='Вероятность отказа в системе в зависимости от lambda', itera
tions=5000,
               func_from_class=theory.get_decline_probability,
               ytitle='probability')
```



4.2 Эксперимент

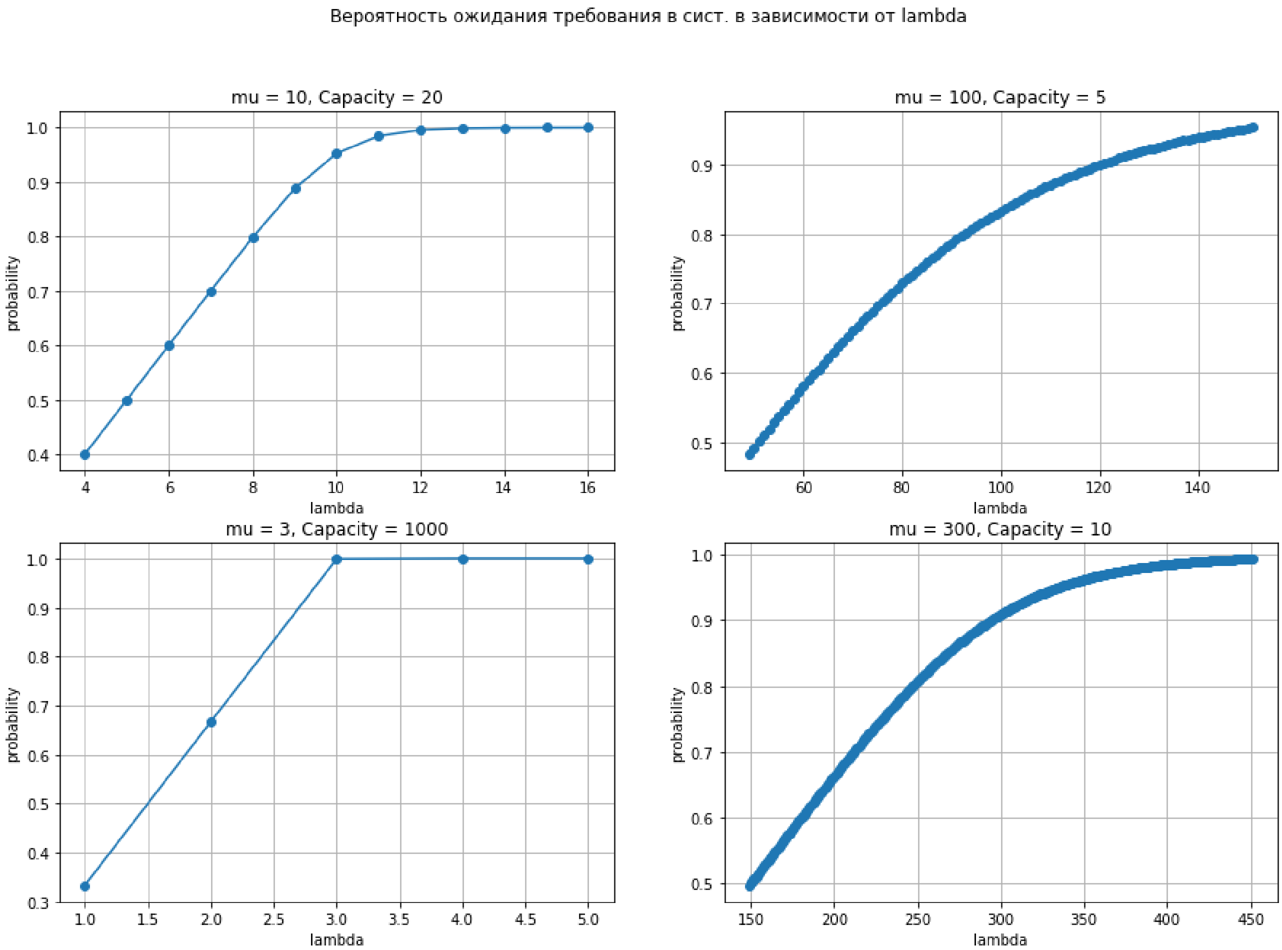
In [195]:
```
simulator.draw(message='Вероятность отказа в системе в зависимости от lambda', itera
tions=5000,
               func_from_class=simulator.prob_of_non_serving,
               ytitle='probability')
```



5. Вероятность ожидания требования в сист. в зависимости от lambda

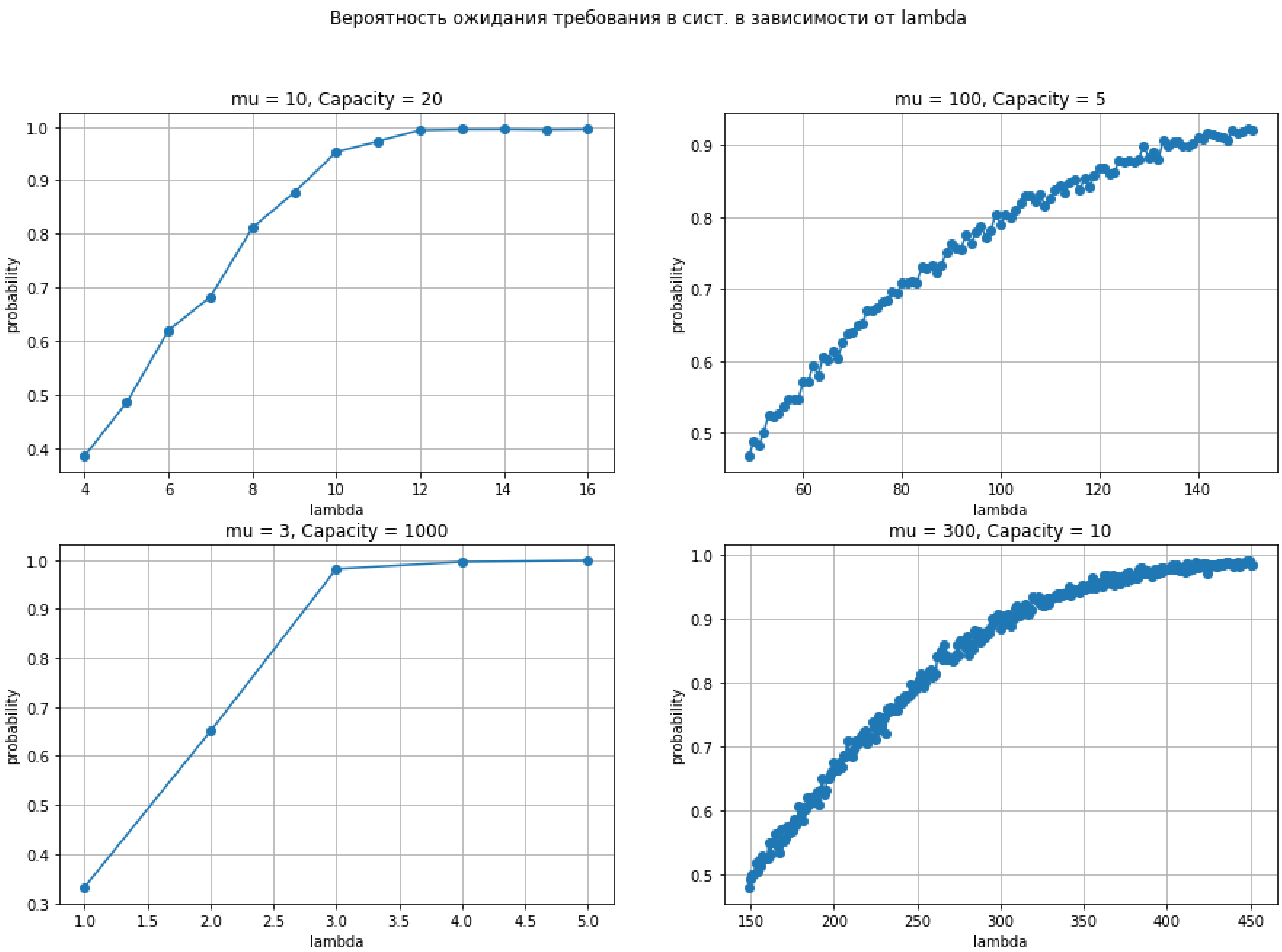
5.1 Теоретическое ожидание

In [206]:
```
simulator.draw(message='Вероятность ожидания требования в сист. в зависимости от lambda', iterations=1000,
               func_from_class=theory.get_waiting_probability,
               ytitle='probability')
```



5.2 Эксперимент

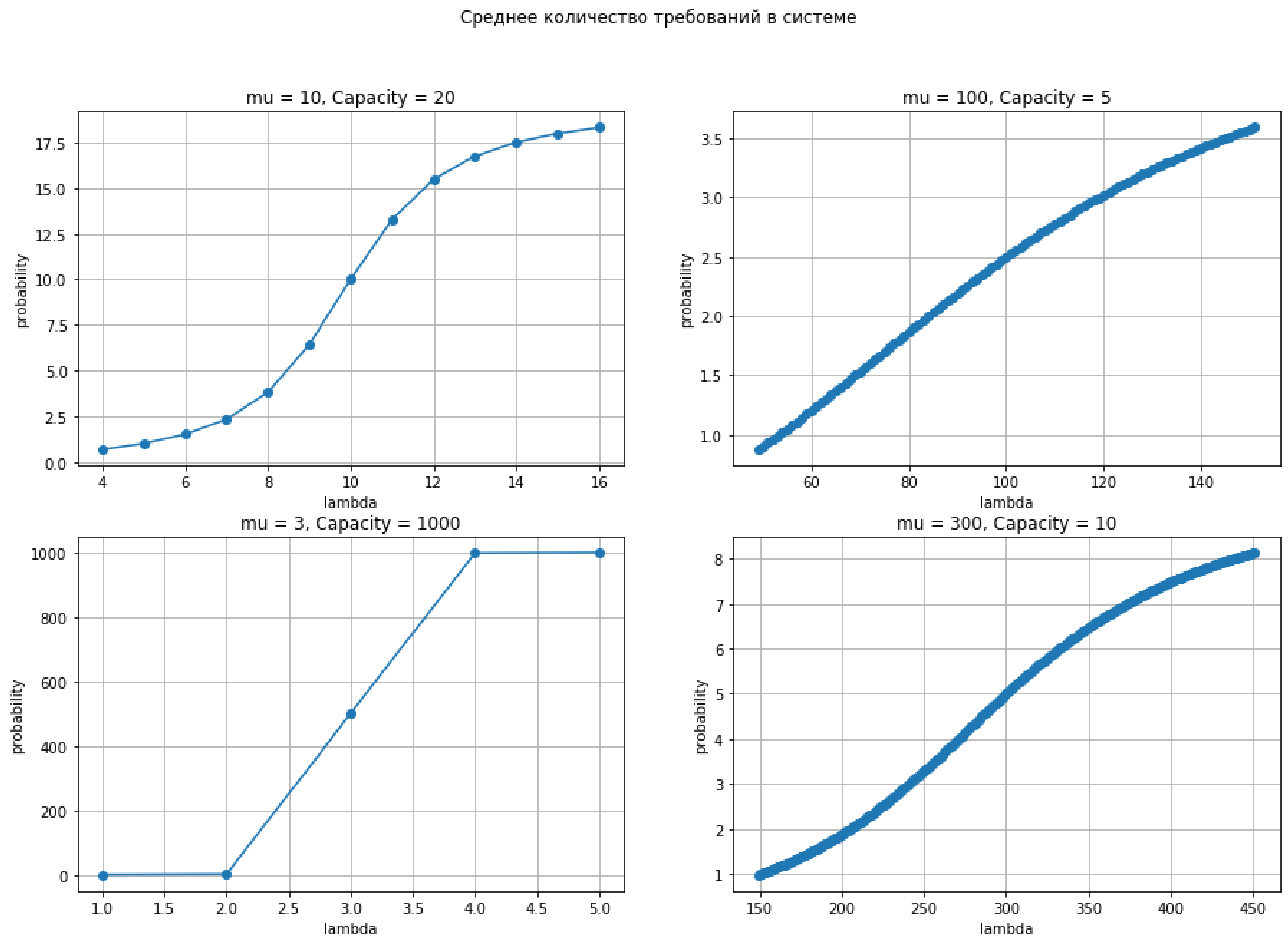
In [213]:
```
simulator.draw(message='Вероятность ожидания требования в сист. в зависимости от lambda', iterations=1000,
               func_from_class=simulator.prob_of_waiting,
               ytitle='probability')
```



6. Среднее количество заявок в очереди

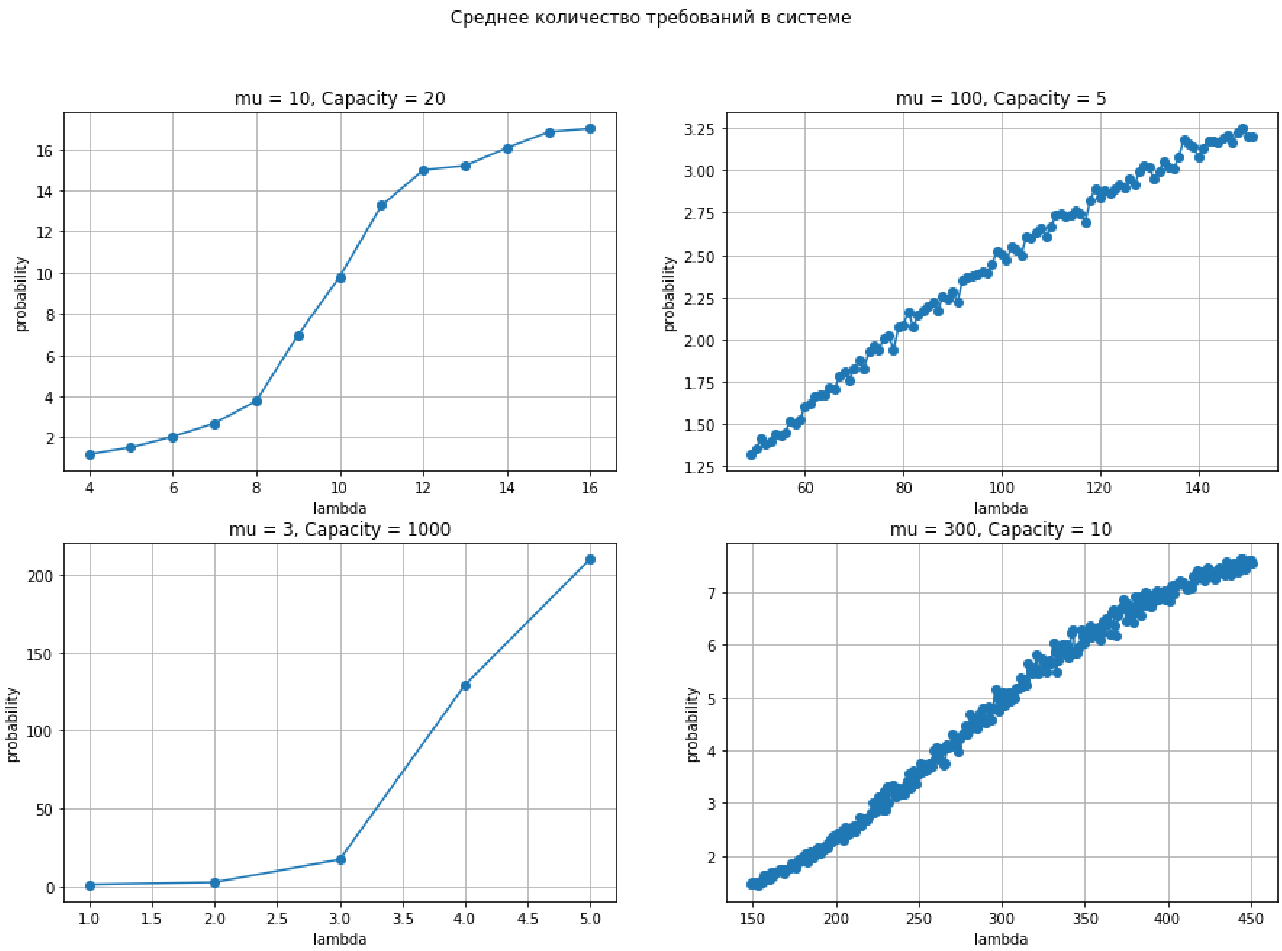
6.1 Теоретическое ожидание

In [215]:
```
simulator.draw(message='Среднее количество требований в системе', iterations=1000,
               func_from_class=theory.get_expectation,
               ytitle='probability')
```



6.2 Эксперимент

In [218]:
```
simulator.draw(message='Среднее количество требований в системе', iterations=1000,
               func_from_class=simulator.get_mean,
               ytitle='probability')
```



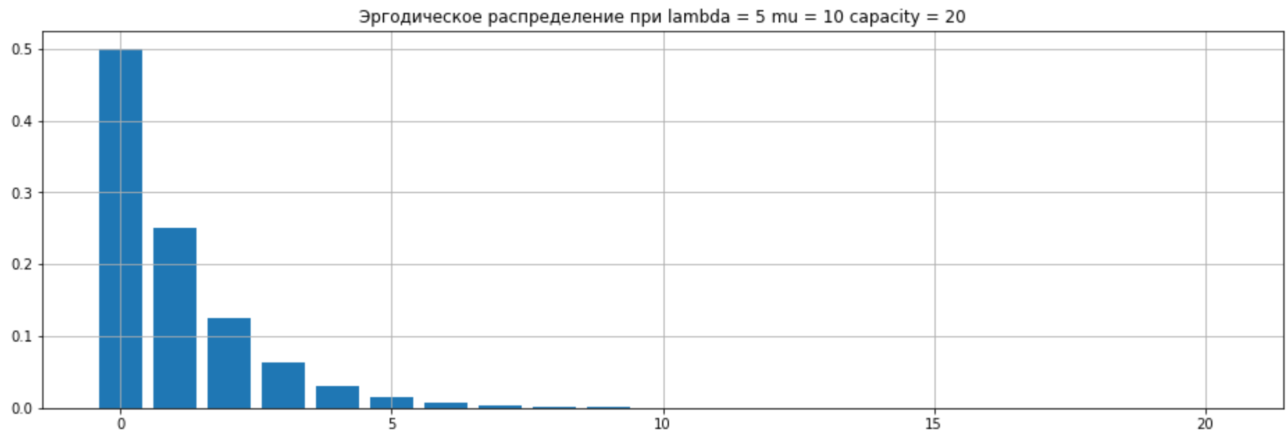
7. Эргодическое распределения

7.1 Теоретическое ожидание

In [5]:
```
def get_prob_distribution(i_case):
    bar_probs = []
    for state in range(capacity[i_case] + 1):
        prob = theory.get_probability(i=state, lambd=lambd[i_case],
                                      mu=mu[i_case], capacity=capacity[i_case])
        bar_probs.append(prob)
    return bar_probs
```

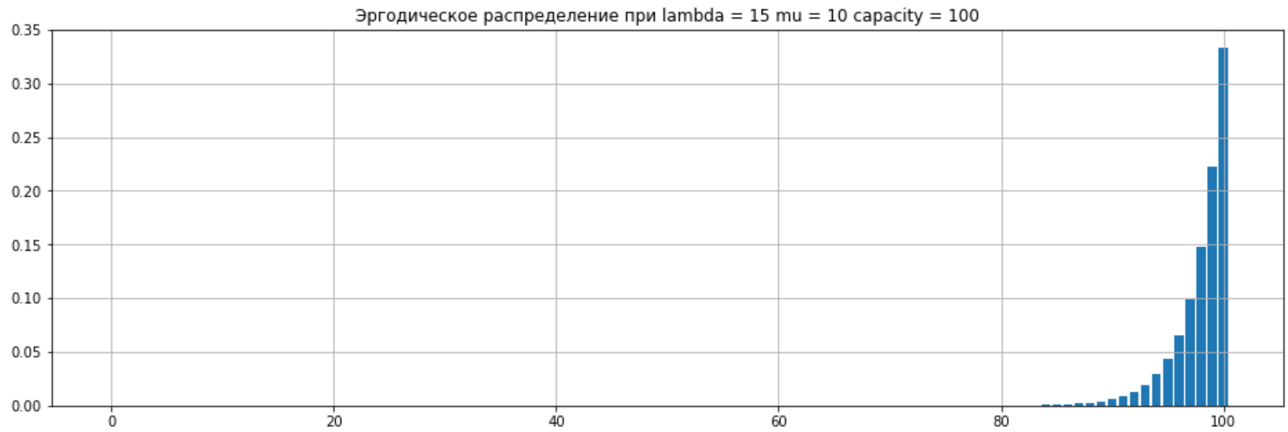
In [8]:

```
i_case = 0
bar_probs = get_prob_distribution(i_case)
plt.figure(figsize=(16,5))
plt.bar(range(capacity[i_case] + 1), bar_probs)
plt.title('Эргодическое распределение при lambda = ' + str(lambd[i_case]) + \
                                              ' mu = ' + str(mu[i_case]) + \
                                              ' capacity = ' + str(capacity[i_case]))
plt.grid()
plt.show()
```

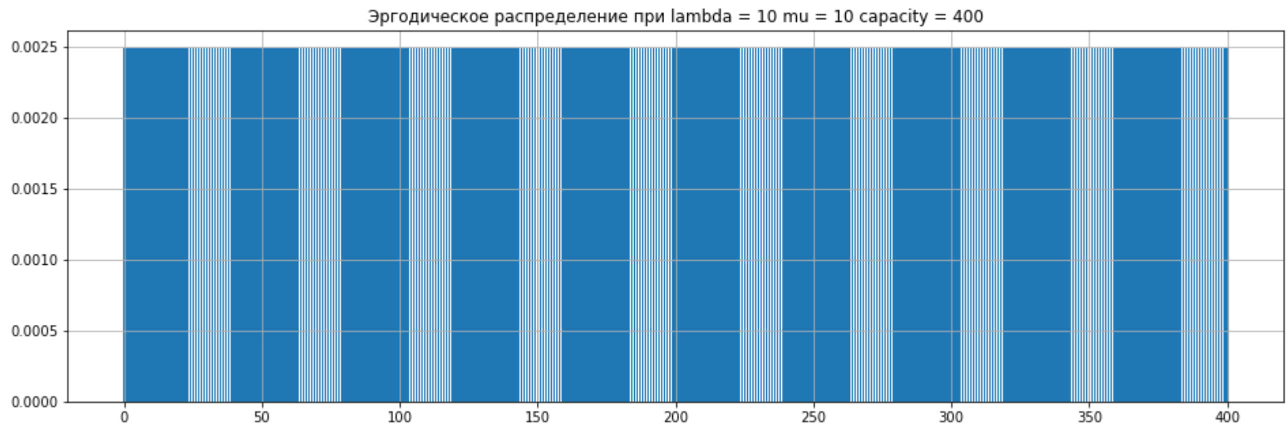


In [12]:

```
i_case = 1
bar_probs = get_prob_distribution(i_case)
plt.figure(figsize=(16,5))
plt.bar(range(capacity[i_case] + 1), bar_probs)
plt.title('Эргодическое распределение при lambda = ' + str(lambd[i_case]) + \
                                              ' mu = ' + str(mu[i_case]) + \
                                              ' capacity = ' + str(capacity[i_case]))
plt.grid()
plt.show()
```



In [14]:
```python
i_case = 2
bar_probs = get_prob_distribution(i_case)
plt.figure(figsize=(16,5))
plt.bar(range(capacity[i_case] + 1), bar_probs)
plt.title('Эргодическое распределение при lambda = ' + str(lambd[i_case]) + \
                                          ' mu = ' + str(mu[i_case]) + \
                                          ' capacity = ' + str(capacity[i_case]))
plt.grid()
plt.show()
```



7.2 Эксперимент

In [23]:
```python
i_case = 0
st_probs = simulator.get_probs(1000, lambd[i_case], mu[i_case], capacity[i_case])
plt.figure(figsize=(16,5))
plt.bar(range(capacity[i_case] + 1), st_probs)
plt.title('Эргодическое распределение при lambda = ' + str(lambd[i_case]) + \
                                          ' mu = ' + str(mu[i_case]) + \
                                          ' capacity = ' + str(capacity[i_case]))
plt.grid()
plt.show()
```

In [24]:
```python
i_case = 1
st_probs = simulator.get_probs(1000, lambd[i_case], mu[i_case], capacity[i_case])
plt.figure(figsize=(16,5))
plt.bar(range(capacity[i_case] + 1), st_probs)
plt.title('Эргодическое распределение при lambda = ' + str(lambd[i_case]) + \
                                          ' mu = ' + str(mu[i_case]) + \
                                          ' capacity = ' + str(capacity[i_case]))
plt.grid()
plt.show()
```

In [42]:
```python
i_case = 2
st_probs = simulator.get_probs(1550000, lambd[i_case], mu[i_case], capacity[i_case])
plt.figure(figsize=(16,5))
plt.bar(range(capacity[i_case] + 1), st_probs)
plt.title('Эргодическое распределение при lambda = ' + str(lambd[i_case]) + \
                                          ' mu = ' + str(mu[i_case]) + \
                                          ' capacity = ' + str(capacity[i_case]))
plt.grid()
plt.show()
```

In [ ]: