

Задание 5.1

In [8]:

```
%matplotlib inline
import scipy.stats as st
import numpy as np
import matplotlib.pyplot as plt
import math
import statsmodels.api as sm
```

In [9]:

```
from numpy import genfromtxt
my_data = genfromtxt('6.csv', delimiter=',', skip_header = 3)
N = len(my_data)
print (len(my_data))
```

1000

In [10]:

```
file = open('6.csv', 'r')
mylambda = 1 / float(file.readline()[9:])
t_0 = float(file.readline()[6:])
t = float(file.readline()[4:])
```

Запишем матожидание и воспользуемся его линейностью $E(N_t|N_s) = E(N_t - N_s|N_s) + E(N_s|N_s)$. По условию $(N_t - N_s) \sim \text{Pois}(\lambda(t - s))$ а также $(N_t - N_s)$ независимо с N_s . Тогда $E(N_t|N_s) = E(N_t - N_s) + N_s = \lambda(t - s) + N_s$

В промежутках времени от 0 до t с шагом $\frac{t_0}{teta}$ запишем количество серверов вышедших из строя, а также посчитаем уточненное значение предсказания $E(N_t|N_{kt_0})$

In [11]:

```
teta = 1
time = np.linspace(0, t, teta*(int(t/t_0))+1) # время выхода
false = np.array(0) # Вышедшие из строя сервера
predic = np.array(0) # Предсказание в данной модели
for i in range(teta*(int(t/t_0))):
    temp = len(my_data[my_data <= time[i+1]])
    false = np.append(false, temp)
    predic = np.append(predic, (t-time[i+1])*mylambda + temp)
```

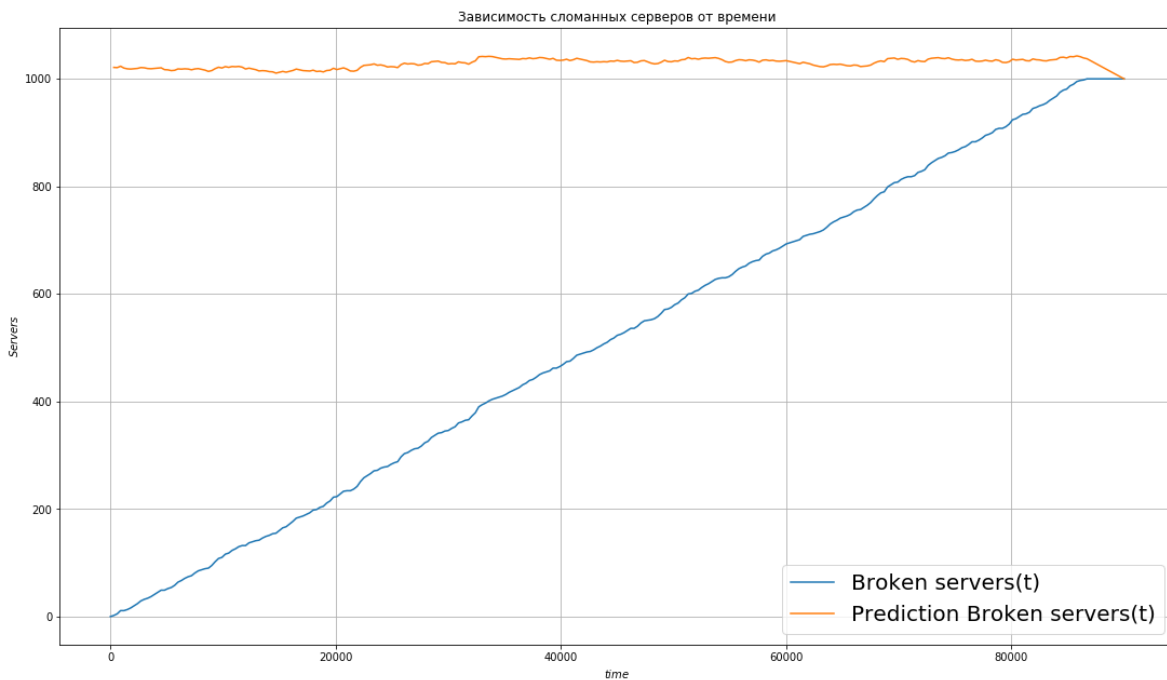
In [12]:

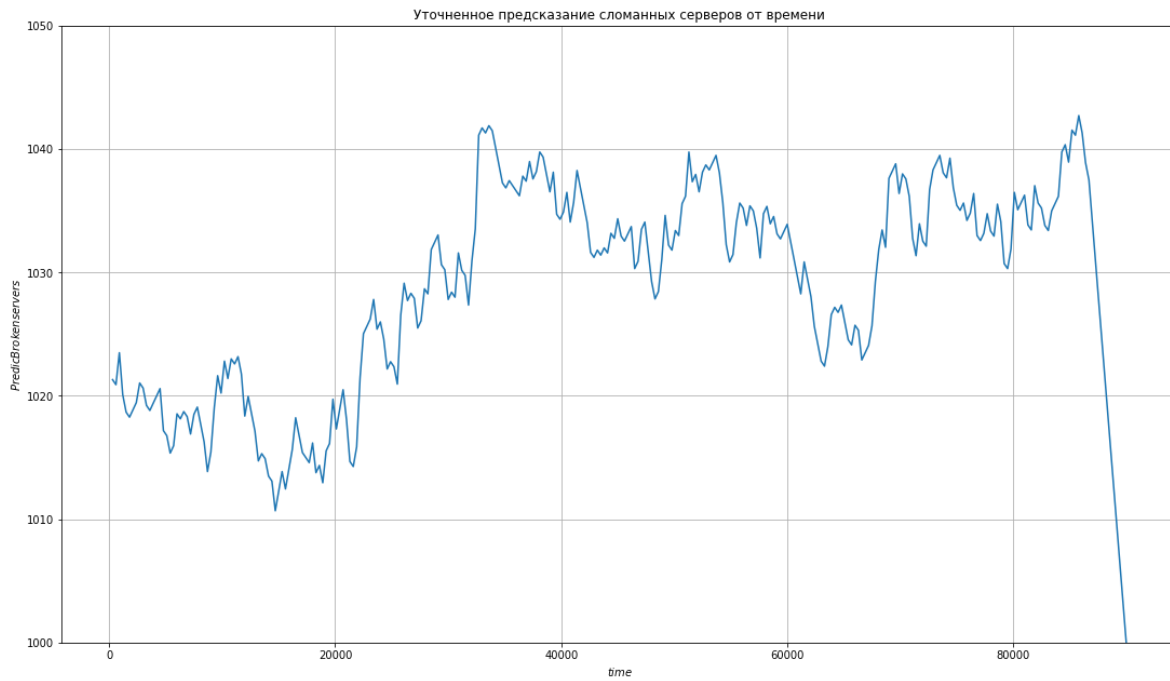
```

plt.plot(time, false, label = r'Broken servers(t)')
plt.plot(time[1:], predic[1:], label = r'Prediction Broken servers(t)')
plt.legend(fontsize=20, loc=4)
plt.title(r'Зависимость сломанных серверов от времени')
plt.xlabel(r'$time$')
plt.ylabel(r'$Servers$')
fig = plt.gcf()
plt.grid(True)
fig.set_size_inches(18.5, 10.5)
plt.show()

plt.plot(time[1:], predic[1:])
plt.title(r'Уточненное предсказание сломанных серверов от времени')
plt.xlabel(r'$time$')
plt.ylabel(r'$Predic Broken servers$')
fig = plt.gcf()
plt.grid(True)
fig.set_size_inches(18.5, 10.5)
plt.ylim(1000.0, 1050.0)
plt.show()

```





Вывод

Получили значение величины $E(N_t|N_s)$, которая предсказывает количество серверов, которые выйдут из строя к моменту t . Из графиков видно, что со временем, при росте числа уже сломавшихся серверов, а значит, и при росте информации о процессе, предсказание количества поломок к моменту t постепенно становится точнее и сходится к истинному значению при t .

In [7]:

```
print (predic[1:])
```

```
[1022.69318182 1022.65909091 1022.625      ... 1000.06818182 1000.03409091
 1000.          ]
```

In []: