Multi-Server Systems

Erlang-B – Application

Erlang-B- Application -Telephone Networks

We assume:

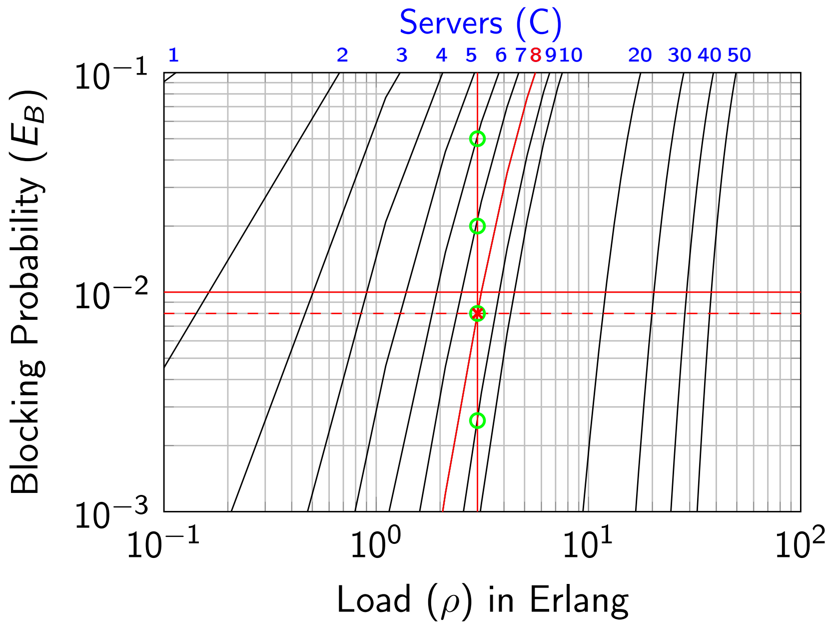
- Calls arrivals: Poisson Process
  - On average: 1 call arrives per minute $\Rightarrow \lambda = 1$ calls/minute
- Call duration: Random variable, exponentially distributed $\Rightarrow \rho = \frac{\lambda}{\mu} = 3$
  - On average: calls last 3 minutes $\Rightarrow \mu = 1/3$ minute^{-1}

> How many phone lines do we need to have in the telephone exchange in order to avoid blocking calls with a probability lower than 1%?

System model: M/M/C/C $\Rightarrow$ C such that $E_B(\rho, C) = \dfrac{\frac{\rho^C}{C!}}{\sum_{i=0}^{C} \frac{\rho^i}{i!}} \leq 0.01$?

- We want to dimension a telephone network over a certain region, in other words, we want to determine the number of lines that are needed in a telephone exchange or switching station
- We assume that calls arrive according to a Poisson process, with a rate of 1 call per minute on average. We also assume that call durations are random and exponentially distributed, and on average, they last three minutes. Our problem is to determine the number of phone lines to install in order to guarantee that the blocking probability is going to be lower than 1 per center
- In order to do so, we note that thanks to the Poissonian arrivals and exponentially distributed calls' duration we can model the system as an M/M/C/C system where the parameters are $\lambda = 1$ call per minute and $\mu = 1/3$ minutes^{-1}. We thus have an offered load which is equal to 3 Erlang. Our problem boils down to finding the value of C such that the blocking probability, given by Erlang's formula, is smaller than or equal to 0.01
- But, we can see that using this formula can be rather cumbersome, so we'll use a graphic method instead

Erlang-B – Application – Using Graphs



In this graph we can see the blocking probability as a function of the load of the system, for several values of C, in a logarithmic scale. We know that we are working at a load which is equal to 3 Erlang, that is 3 times 10 exponent 0, and that we need a blocking probability of less than 1 percent, that is, less than 10 to the exponent minus two. We then look at the intersections of our working load with the different curves for different values of C, and look for the intersection where the blocking probability is immediately below the red line. So we keep the solution with 8 servers, which provides a blocking probability around 0.8%.

Erlang-B– Application – Using Tables

- Maximum load (ρ) versus blocking probability (E_B) and number of servers (C).

C\ E_B	1%
6	1.909
7	2.501
8	3.128
9	3.783
10	4.461

Another highly used method are pre-calculated tables. In these tables, we have the supported load for different system capacities and blocking probabilities, one per cent in this case. In our problem we need to support at least 3 Erlang with a blocking probability of 1%, so we see that we need at least 8 servers. And we find the same answer as before, namely, that we need to install 8 phone lines