Erlang-C Formula Calculation

Author: Bo Bayles (bbayles@gmail.com)

The Erlang-C formula is defined as follows:

$$P_w = \frac{\frac{A^N}{N!}\frac{N}{N-A}}{\frac{A^N}{N!}\frac{N}{N-A} + \sum_{x=1}^{N-1}\frac{A^x}{x!}}$$

In the forumla, P_w is the probability of waiting in the queue, A is the total traffic offered by the system (in Erlangs), and N is the number of trunks in the system. It is assumed that the incoming traffic arrives in a Poisson process and that the times a trunk is occupied follow an exponential distribution (before using any formulas discussed below, you should verify these assumptions are applicable to your situation).

It's possible to calculate the waiting probability directly from the formula above, but that involves computing (N-1)!, which can become quite cumbersome with anything but a small N. For example, 171! is larger than the largest double-precision floating point number that can be represented in many applications (170! is about 7.26E+306. By comparison, the number of particles in the universe is estimated to be something like 1E+87.). Any calculation that tried to use a larger N would then fail.

Fortunately, we can derive a recursive definition of the Erlang-C formula that avoids have to calculate (N-1)!. First, we observe that the reciprocal of P_w is easier to handle than P_w itself, so we define C and simplify as follows:

$$C = \frac{1}{P_W} = \frac{\frac{A^N}{N!}\frac{N}{N-A} + \sum_{x=0}^{N-1}\frac{A^x}{x!}}{\frac{A^N}{N!}\frac{N}{N-A}} = 1 + \frac{N!}{A^N}\frac{N-A}{N}\sum_{x=0}^{N-1}\frac{A^x}{x!}$$

For N = 1 trunk, we can see that $C_{N=1}$ works out to be 1/A. Using wxMaxima, we can work out some of the next few as well:

$$C_1 = \frac{1}{A}$$

$$C_2 = \frac{1}{A} + \frac{2}{A^2}$$

$$C_3 = \frac{1}{A} + \frac{4}{A^2} + \frac{6}{A^3}$$

$$C_4 = \frac{1}{A} + \frac{6}{A^2} + \frac{18}{A^3} + \frac{24}{A^4}$$

Notice that we can write $C_{N=3}$ as:

$$C_3 = \left(\frac{1}{A} + \frac{2}{A^2}\right) + \left(\frac{2}{A^2} + \frac{6}{A^3}\right)$$
$$C_3 = C_2 + \left(\frac{3}{A^2} + \frac{6}{A^3}\right) - \frac{1}{A^2}$$
$$C_3 = C_2 + \frac{3}{A}\left(\frac{1}{A} + \frac{2}{A^2}\right) - \frac{1}{A}\left(\frac{1}{A}\right)$$
$$C_3 = C_2 + \frac{3}{A}C_2 - \frac{1}{A}C_1$$
$$C_3 = C_2\left(1 + \frac{3}{A}\right) - C_1\left(\frac{1}{A}\right)$$

This pattern holds for $C_{N=4}$, $C_{N=5}$, and so on... for N > 2, we can write:

$$C_N = C_{N-1}\left(1 + \frac{N}{A}\right) - C_{N-2}\left(\frac{N-2}{A}\right) \quad \text{for integer N > 2}$$

Or, we can write the following, which is a bit prettier:

$$C_N = \frac{1}{A}\left(C_{N-1}(A+N) + C_{N-2}(2-N)\right) \quad \text{for integer N > 2}$$

This can be proved using the definition of C_N with the following facts and (lots of – use a CAS) algebra:

$$\sum_{x=0}^{N-2} \frac{A^x}{x!} = \left(\sum_{x=0}^{N-1} \frac{A^x}{x!}\right) - \frac{A^{(N-1)}}{(N-1)!}$$

$$\sum_{x=0}^{N-3} \frac{A^x}{x!} = \left(\sum_{x=0}^{N-1} \frac{A^x}{x!}\right) - \frac{A^{(N-1)}}{(N-1)!} - \frac{A^{(N-2)}}{(N-2)!}$$

$$N! = N(N-1)!$$

$$N! = N(N-1)(N-2)!$$

Now that we have the recursive relationship for the inverse of the waiting probability, we can write a function in our favorite programming language to compute the Erlang-C formula without having to work out large factorials.

It is best to use a strictly typed programming language, since the number of trunks must be a positive an integer. In C++:

```
double ECBlocking(const double A, const unsigned int N)
{
  double C;
  double C_minus_1 = 1/A + 2/(A*A);
  double C_minus_2 = 1/A;

  if(A > N)
    C = 1;
  else if(N > 2)
  {
    for(int x = 3; x <= N; x++)
    {
      C = C_minus_1*(1+x/A)-C_minus_2*(x-2)/A;
      C_minus_2 = C_minus_1;
      C_minus_1 = C;
```

```
    }
  }
  else if(N == 2)
    C = C_minus_1;
  else if(N == 1)
    C = C_minus_2;
  else
    C = 1;

  return(1/C);
}
```