Описание счета формулы Эрланга Б и Энгсета

Велькович Серджан

Ноябрь 2010.

1 Введение

Эрланг Б формула определяет потери некоторых из предлагаемых трафика для заданное число линий.

Практические, Эрланг Б формула для определенного числа (телефон) вызова, которая должна пройти через ряд голос путей (каналов) дает вероятность того, что звонок не будет в состоянии пройти (но будет “ падения”).

Эрланг Б формула предполагает бесконечное число источников трафика, и назвать, что упал не влияет на дальнейший ход. Оба предположения грубо.

Первое предположение используется, когда число источников значительно больше, чем количество каналов (далее “ более 8 раз выше,”, хотя мы и не известно что математически это производный рисунок). Если нет, Энгсет формула применяется, были решения в разделе 3.

Второе предположение уместно, если потери малы. Если потери не мала, то она применяется “ расширенной Эrlang B формула”, то есть., алгоритма. Дисплей этого алгоритма приведены в разделе 2.5.

Для более подробное описание можно прочитать некоторые статьи о теории трафика.

1.1 Если есть Б, то есть также Ц

Существует также формуле Эрланг Ц, которая относится к случаю, когда вы звоните, если вам не удастся, подождите, пока он освобождает некоторые канала. Грубой думаю, ждать “бесконечно”, если не выполнено, применимы и к другим формулам, которые учитывают “ Сколько времени ожидания вызова” или “сколько звонков можно ждать в то же время”, или обоих.

Есть еще дело не с Эрланг Ц формула и ее родственников.

2 Эргланг Б формула

К сожалению, Есть некоторые теги приняты концепции в теории трафика, поэтому мы будем использовать ее. Таким образом, формулы Эрланга Б является:

$$p_g = \frac{\frac{s^k}{k!}}{\sum_{i=0}^{k} \frac{s^i}{i!}} \tag{1}$$

где:

p_g вероятность потери вызова

s предложил трафика в Эргланг (количество звонков в одно время)

k количество каналов для передачи трафика (звонков)

2.1 Счет

Поскольку Эрланг Б формула используется факториал, ясно, что использование Формула 'непрактично в современных компьютерах для более высокие значения k, потому что значение факториала или вне диапазона, или слишком медленная обработка.

Кроме того, формулы Эрланга Б дает соотношение между тремя параметрами. Часто интересно определить некоторые из двух входных параметров Erlang B Формула, основанная на другой вход и выход. Так что, если мы знаем предложил трафика, мы можем определить количество необходимых каналов определенной вероятностью потери. Такође, ако нам је наметнут број канала, можемо за одређену вероватноћу губитака да одредимо колики понуђени саобраћај можемо да прихватимо.

2.2 Рекурсивные счет Эрланг Б формуле

Этот счет довольно хорошо известны. Идея заключается в том, за те же трафика, мы значение для заданного числа каналов для определения на основе стоимости предыдущий номер канала (один меньше). Для удобства мы вычислить обратное значение.

$$\frac{1}{p_g} = \frac{\sum_{i=0}^{k} \frac{s^i}{i!}}{\frac{s^k}{k!}} = \frac{k!}{s^k} \sum_{i=0}^{k} \frac{s^i}{i!} = \sum_{i=0}^{k} \frac{k! s^k}{s^i i!}$$

$$\frac{1}{p_g} = \sum_{i=0}^{k} \frac{k!}{s^{k-i} i!} \tag{2}$$

$$\frac{1}{p_g(k+1)} = \sum_{i=0}^{k+1} \frac{(k+1)!}{s^{k+1-i} i!} = \frac{(k+1)!}{(k+1)!} + \sum_{i=0}^{k} \frac{(k+1)!}{s^{k+1-i} i!} = 1 + \frac{k+1}{s} \sum_{i=0}^{k} \frac{k!}{s^{k-i} i!}$$

$$\frac{1}{p_g(k+1)} = 1 + \frac{k+1}{s} \frac{1}{p_g(k)} \tag{3}$$

Главная ценность - за $k =$, получается простой формулы расчета 2:

$$\frac{1}{p_g(0)} = \frac{0!}{s^0 0!} = 1$$

Выраженный алгоритмически, в языке программирования Руби:

```
def erlangb(s, k)
    r = 1.0
    1.upto(k) { |j| r = 1 + r*j/s }
    return 1.0 / r
end
```

Это просто для иллюстрации алгоритма, а не для промышленного использования.

2.3 Определение числа каналов с бисекции

Чтобы определить количество каналов, мы замечаем, что для инвариантной трафика p_g равномерно убывающая функция (при k растет, p_g уменьшается). Это интуитивно понятно, так как многие каналы, то он будет потерь же трафик будет меньше. Однако, это может быть доказано математически.

Вот как выполнить Киао и Киао, в работе, что было (до этого письменной форме) можно ознакомиться на Интернете по адресу:

`http://www.cas.mcmaster.ca/~qiao/publications/erlang/newerlang.html`

Интересно отметить, что Киао Киао и дают различные определения рекурсивные Эрланг Б формулу, которая дает другой результат от “наши”. Как не дать подробную процедуру, это не ясно, будет ли ошибка в процедуре, или различия незначительны в той мере, интерес к применению.

Однако, их доказательство того, что p_g равномерно уменьшается неизменной трафика является действительным. Здесь он “рассказывает”:

$$p_g(k+1) = \frac{\dfrac{s^{k+1}}{(k+1)!}}{\displaystyle\sum_{i=0}^{k+1} \frac{s^i}{i!}} = \frac{s}{k+1} \frac{\dfrac{s^k}{k!}}{1 + \displaystyle\sum_{i=1}^{k+1} \frac{s^i}{i!}} < \frac{\dfrac{s^k}{k!}}{\dfrac{k+1}{s} \displaystyle\sum_{i=1}^{k+1} \frac{s^i}{i!}} = \frac{\dfrac{s^k}{k!}}{\dfrac{k+1}{s} \displaystyle\sum_{i=0}^{k} \frac{s^{i+1}}{(i+1)!}} =$$

$$= \frac{\dfrac{s^k}{k!}}{\displaystyle\sum_{i=0}^{k} (\frac{k+1}{i+1} \frac{s^i}{i!})} < \frac{\dfrac{s^k}{k!}}{\displaystyle\sum_{i=0}^{k} \frac{s^i}{i!}} = p_g(k)$$

Короче говоря:

$$p_g(k+1) < p_g(k)$$

Если мы знаем, что это функция монотонно убывает, то мы можем использовать метод бисекции. Нам нужно только два значения $p_g(k)$, одно маленькое и один больше, чем вход P_g - то есть, два значения k, для которых действительным.

Логично начать с 0 и предложил трафика в качестве начального значения. Но если $p_g(s) < p_g$, то мы должны быть такими, $k > s$, для которых $p_g(k) > P_g$.

Киао и Киао предлагается, без объяснения причин (за исключением “Мы пришли к выводу, что такие лучшие”), что мы предложили трафика увеличивается с 32, а не найти то, что мы ищем. Потому что не ясно, как “ одно и то же число” быть хорошим для различных значений предлагается трафика, до 32 в этом случае делает магическое число.

Мы считаем, что смысл применять фактор, а не, как это будет поиски, чтобы сделать предметом такой торговли. Мы считаем, что в 1,5 раза необходимости. В самом деле, потери должны быть очень малы, что он принимает 50 % больше каналов, чем предлагаемые трафика, что на практике редко счастья. Если они действительно случаются, и, после нескольких проходов должны мы закончим наш поиск.

Теперь мы покажем, алгоритм в Руби:

```
def kanala_erlangb(p_g, s)
    l, r = 0, s.ceil
    while erlangb(s, r) > p_g
        l = r
        r = 1 + (r * 3) / 2
    end
    mid = 0
    while (r-l) > 1
        mid = (l+r+1)/2
        if erlangb(s, mid) > p_g
            l = mid
        else
            r = mid
        end
    end
    return r
end
```

Это просто для иллюстрации алгоритма, а не для промышленного использования.

2.4 Определение трафика с бисекции

Для определения трафика, обратите внимание, что при неизменной количество каналов, p_g равномерно возрастающая функция (в s увеличивается, p_g растет). Это интуитивно понятно, выше трафик, это будет потери за тот же номер канала будет выше. Однако, это может быть доказано математически. број канала бити већи. Ипак, ово може и математички да се докаже.

Здесь мы применим классический метод, используя первое различие. Доказать:

$$p_g'(s) > 0 \iff k = const \tag{4}$$

Используя хорошо известное тождество:

$$f'(\frac{u}{v}) = \frac{f'(u)f(v) - f(u)f'(v)}{v^2}$$

по формуле 1, предполагая, что k постоянная, мы получим:

$$p'_g(s) = \frac{\frac{ks^{k-1}}{k!}\sum_{i=0}^{k}\frac{s^i}{i!} - \frac{s^k}{k!}\sum_{i=0}^{k}\frac{is^{i-1}}{i!}}{(\sum_{i=0}^{k}\frac{s^i}{i!})^2}$$

Для наших целей достаточно отметить, что доля ниже линии квадрат выражение, которое означает, что ≥ 0, и может быть 0 только если $s = 0$ (нет трафика), которые не интересны для нас. Таким образом, мы можем ограничить анализ знака выше дробных линии.

$$\frac{ks^k}{k!s}\sum_{i=0}^{k}\frac{s^i}{i!} - \frac{s^k}{k!}\sum_{i=0}^{k}\frac{is^{i-1}}{i!} = \frac{s^k}{k!}(\sum_{i=0}^{k}\frac{ks^{i-1}}{i!} - \sum_{i=0}^{k}\frac{is^{i-1}}{i!})$$

Как $\frac{s^k}{k!} > 0$, мы можем ограничить характер, что остаются в скобках:

$$\sum_{i=0}^{k}\frac{ks^{i-1} - is^{i-1}}{i!} = \sum_{i=0}^{k}\frac{s^{i-1}(k-i)}{i!}$$

Отсюда ясно, что знак зависит только от знака выражения:

$$\sum_{i=0}^{k}(k-i) = (k+1)k - \sum_{i=0}^{k} i = k(k+1) - \frac{k(k+1)}{2} > 0$$

Кроме того, это как порядок определения количества каналов, так что не дальнейших объяснений мы даем Руби код.

```
def saob_erlangb(p_g, k)
    l, r = 0, k.ceil
    while erlangb(s, r) < p_g
        l = r
        r = 1 + (r * 3) / 2
    end
    mid = 0
    while (r-l) > 1
        mid = (l+r+1)/2
        if erlangb(mid, k) < p_g
            l = mid
        else
            r = mid
        end
    end
    return r
end
```

2.5 Алгоритм расширенной Эрланг Б формуле

Если p_d обозначает вероятность того, что называвать это не удается, быть (справа) и еще раз, алгоритм:

1. Рассчитать Эрланг Б формуле 1
2. Увеличить предложил трафика толко: sp_dp_g
3. Рассчитать Эрланг Б формуле с трафика, полученного на предыдущем шаге
4. Повторите предыдущие два шага до p_g становится стабильной.

При стабильных предназначена для изменения результатов (p_g) за счет реализации этого алгоритма стала незначительной.

В Руби:

```
def erlangb_prosxiren(s, k, p_d, epsilon)
    r, p_g = 0.0, erlangb(s, k)
    delta = epsilon
    while delta >= epsilon
        r = erlangb(s, k)
        delta = (r - p_g).abs
        p_g = r
        s += s * p_d * p_g
    end
    return p_g
end
```

В код `epsilon` пограничной разница между двумя последовательными результаты Эрланг Б формула, согласно которой считается, что алгоритм выполняется.

Счет числа каналов и максимальный трафик по расширенной Эрланг Б формулу (алгоритм) в основном же, как и счет (основной) Эрланг Б формуле. Только виджет, что при расчете формулы Эрланга Б во время применения метода бисекции использовать и p_d.

3 Формула Энгсета

В смысле Энгсет формула является более общим Эрланг Б формуле, что не игнорирует призыв ряда источников.

$$p_g = \frac{s^k \begin{pmatrix} v \\ k \end{pmatrix}}{\sum\limits_{i=0}^{k} s^i \begin{pmatrix} v \\ i \end{pmatrix}} \tag{5}$$

В связи с Эрланг Б формуле 1), здесь у нас есть новый размер v которая является источником вызовов ($v \geq k$).

Эквивалент записи этой формулы:

$$p_g = \frac{\frac{s^k v!}{(v-k)!k!}}{\sum_{i=0}^{k} \frac{s^i v!}{(v-i)!i!}} = \frac{\frac{s^k}{(v-k)!k!}}{\sum_{i=0}^{k} \frac{s^i}{(v-i)!i!}}$$

В настоящее время очевидно, что Энгсет и Эрланг Б формула очень похожа. Как результат, во многом таким же образом мы приходим к рекуррентной формулы для вычисления.

$$\frac{1}{p_g(k)} = \frac{(v-k)!k!}{s^k} \sum_{i=0}^{k} \frac{s^i}{(v-i)!i!}$$

$$\frac{1}{p_g(k+1)} = \frac{(v-k-1)!(k+1)!}{s^{k+1}} \sum_{i=0}^{k+1} \frac{s^i}{(v-i)!i!} =$$

$$= \frac{(v-k-1)!(k+1)!}{s^{k+1}} \left(\sum_{i=0}^{k} \frac{s^i}{(v-i)!i!} + \frac{s^{k+1}}{(v-k-1)!(k+1)!} \right)$$

$$= \frac{(k+1)}{s(v-k)} \frac{(v-k)!k!}{s^k} \sum_{i=0}^{k} \frac{s^i}{(v-i)!i!} + 1 = \frac{(k+1)}{s(v-k)} \frac{1}{p_g(k)} + 1$$

Очевидно, что мы получим очень аналогичная формула, но мы мало что изменилось чтобы быть более понятным для кодирования:

$$\frac{1}{p_g(k+1)} = \frac{(k+1)}{s(v+1-(k+1))} \frac{1}{p_g(k)} + 1 \tag{6}$$

Что касается формулы Эрланга Б, а вот прямой подстановки получим:

$$p_g(0) = \frac{s^0 v!}{s^0 v!} = 1$$

Оттуда легко добраться Руби код:

```
def engset(s, k, v)
    r, v_1 = 1.0, v+1
    1.upto(k) { |j| r = 1 + r*j/(s*v_1) }
    return 1.0 / r
end
```

Здесь интересно отметить, что формулы для счета Энгсет другой популярный метод, который даже не решить проблему больших факториал. Вполне возможно, что это связано с тем, что Энгсет формула обычно используется для небольших количествах, для которых это не проблема. Однако мы считаем, что это решение лучше, потому что она позволяет применение Энгсет формулы для больших чисел, при принятии вызова пружин большого объема трафика. Она подходит для расчета худшем случае при проектировании системы.

Счет числа каналов, источников и максимальный трафик, когда другие известные параметров аналогично формуле Эрланга Б. Соображения о предмет может быть впоследствии добавил при этом, если возникнет такая необходимость.