12th GI/ITG CONFERENCE ON MEASURING, MODELLING AND EVALUATION OF COMPUTER AND COMMUNICATION SYSTEMS

3rd POLISH-GERMAN TELETRAFFIC SYMPOSIUM

MEETING DEADLINES THROUGH RESTART

Aad P. A. van Moorsel[1] and Katinka Wolter[2]

[1] University of Newcastle
School of Computing Science
Newcastle upon Tyne, NE1 7RU, United Kingdom
aad.vanmoorsel@ncl.ac.uk
[2] Humboldt-Universität zu Berlin,
Institut für Informatik,
Unter den Linden 6, 10099 Berlin, Germany
wolter@informatik.hu-berlin.de

Abstract. This short paper reports on algorithms to determine restart times that maximise the probability that a deadline is met. These algorithms are tailored to lognormal distributed completion times, as found in various Internet applications. The algorithms are particularly efficient, so they can be applied on-line. This work therefore represents an intermediate step towards implementing on-line adaptive restart mechanisms.

1 INTRODUCTION

Retrying tasks is an obvious thing to do if one suspects a task has failed. However, also if a task has not failed, it may be faster to restart it than to let it continue. Whether restart is indeed faster depends on the completion time distribution of tasks, and on the correlation between the completion times of consecutive tries. In this paper we assume that the completion times of consecutive tries are independent and identically distributed, an assumption that has been shown to be not unreasonable for Internet applications [4]. Furthermore, we analyse algorithms in this paper that are tailored to lognormal distributions, which we (and others) have found to be representative for various Internet applications [1,4]. Our metric of interest is the probability that a pre-determined deadline is met, and we want to find the restart times that maximise this metric. Note that the metric of meeting deadlines corresponds to points in the completion time distribution, a metric often harder to obtain than moments of completion time (analysed in [3]).

We derive two very efficient algorithms to determine the optimal time for restart. The 'equi-hazard' algorithm finds all restart intervals with equal hazard rates, which corresponds to all local extrema for the probability of making the deadline. It turns out that among the equi-hazard restart intervals, in all cases we applied the algorithm to lognormal distributed completion times, equi-distant points are optimal. Therefore, a practical engineering approach is to only consider equi-distant points, which we do in our second algorithm. The equi-hazard algorithm finds each local extreme in logarithmic time, the equi-distant algorithm takes a constant time to do the same, and finds the globally optimal solution in a few iterations. Hence, these algorithms are excellent candidates for on-line deployment in potential future adaptive restart implementations.

2 DEADLINE PROBABILITIES OF TASKS WITH RESTART

To analyse and optimise the time at which to restart a job, we start from a simple model that lends itself to elegant analysis. We assume that the restart of a task terminates the previous attempt. This is for instance the case when we click the reload button in a web browser: the connection with the server is terminated and a new download attempt is tried. We then assume that successive tries are statistically independent and identically distributed. This assumption has been found realistic in a measurement study of HTTP Get [4].

In mathematical terms, the problem formulation is as follows. Let the random variable T denote the completion time of a job, with probability distribution $F(t), t \in [0, \infty)$, and let d denote the deadline we set out to meet. Obviously, without restart, the probability that the deadline is met is $F(d)$. Assume τ is a restart time, and introduce the random variable T_τ to denote the completion time when an unbounded number of retries is allowed. That is, a retry takes place periodically, every τ time units, until completion of the job or until the deadline has passed, which ever comes first. We write $f_\tau(t)$ and $F_\tau(t)$ for the density and distribution of T_τ, respectively, and we are interested in the probability $F_\tau(d)$ that the deadline is met.

One can intuitively reason about the completion time distribution with restarts as Bernoulli trials. At each interval between restarts there is a probability $F(\tau)$ that the completion 'succeeds.' Hence, if the deadline d is a multiple of the restart time τ, we can relate the probability of missing the deadline without and with restart through:

$$1 - F_\tau(d) = (1 - F(\tau))^{\frac{d}{\tau}}. \tag{1}$$

This can simply be extended to the case that d is not a multiple of τ, or that a time penalty is associated with restarts, and also to the case that restart times are not all identical to τ. Some additional notation is required, and we omit it here.

For a single retry during the finite interval $[0, d)$, we obtain that when the retry is at time τ, $\tau < d$, then the probability of completion before d is:

$$F_\tau(d) = 1 - (1 - F(\tau))(1 - F(d - \tau)). \tag{2}$$

By equating the derivative with respect to τ to zero, we obtain for the extrema of $F_\tau(d)$ that:

$$\frac{f(\tau)}{1 - F(\tau)} = \frac{f(d - \tau)}{1 - F(d - \tau)}. \tag{3}$$

The function $h(t) = \frac{f(t)}{1-F(t)}$ is known as the hazard rate, and is key throughout our analysis and algorithms. The above result shows that minima and maxima for the probability that a deadline is met with restarts are found at *equi-hazard* restart

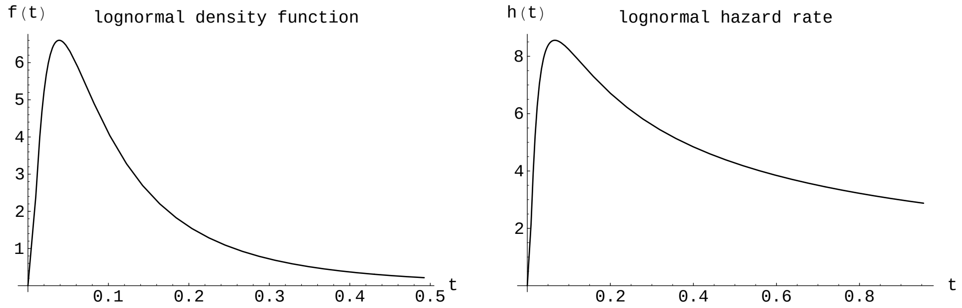

Fig. 1. The probability density (left) and hazard rate (right) of a lognormal distribution ($\mu = -2.3, \sigma = 0.97$).

intervals. Moreover, the *equi-distant* restart intervals $\tau = \frac{d}{2}$ are a special case of equi-hazard intervals, and form thus also a local extreme.

For multiple retries before the deadline, we can do similar mathematics. This time we take derivatives with respect to each restart interval $\tau_i, i = 1, \ldots, N$. (Note, the restarts take place at times $\tau_1, \tau_1 + \tau_2, \ldots, \sum_{n=1}^{N} \tau_n$, and we assume without loss of generality that $\sum_{n=1}^{N} \tau_n = d$.) Then we obtain that an optimum with respect to all retry intervals $\tau_1, \ldots, \tau_N$ is found when:

$$\frac{f(\tau_1)}{1 - F(\tau_1)} = \frac{f(\tau_2)}{1 - F(\tau_2)} = \ldots = \frac{f(\tau_N)}{1 - F(\tau_N)}. \tag{4}$$

Again, the extrema are at equi-hazard intervals, with as special case the equi-distant restart intervals $\tau_n = \frac{d}{N}$.

2.1 Deadline Probabilities for Lognormal Distribution

Very often, completion times for Internet tasks have a distribution function that can be closely fit by a lognormal distribution [1,4]. A lognormal distribution relates closely to the normal distribution: one obtains a normal distribution if one takes the logarithm of samples of a lognormal distributed random variable. To define a lognormal distribution uniquely, we need two parameters, and usually one takes the parameters μ and σ that correspond to the mean and standard deviation of the normal distribution constructed as explained above. Figure 1 shows the density function and the hazard rate of a lognormal distribution.[3]

The lognormal shape of the hazard function can be exploited by optimisation algorithms, since it has at most two points with the same hazard function value. This allows us to quickly identify all potential solutions of the optimisation problem. The

[3] A lognormal distributed random variable with parameters μ and σ has density $f(x) = \frac{1}{\sigma x \sqrt{(2\pi)}} e^{\frac{-(ln(x)-\mu)^2}{2\sigma^2}}$.

following algorithm finds the two restart interval lengths τ_a and τ_b for which holds:

$$h(\tau_a) = h(\tau_b), \tag{5}$$
$$n_a\tau_a + n_b\tau_b = d, \tag{6}$$

where n_a and n_b denote the number of intervals of each length. The parameters n_a and n_b are input to the algorithm, and to find the optimal restart strategy, one needs to call the algorithm for all relevant combinations of n_a and n_b, and then select from all the equi-hazard solutions the one that optimises the probability of meeting the deadline.

Algorithm 1 (Equi-hazard Restart Intervals)

```
Input n_a and n_b;
top=d/n_b; bottom=d/(n_a + n_b);
τ_b = top; τ_a = (d − n_b τ_b)/n_a;
Repeat {
   top = (top+bottom)/2;
   τ_b = top;
   τ_a = (d − n_b τ_b)/n_a; (so interval lengths sum to d)
   If( SignChanged(h(τ_b) − h(τ_a)) ) {
     bottom=top;
     top=PreviousValue(top);
   }
}
Until (top-bottom≈ 0)
```

To explain the working of Algorithm 1, first note that one solution to (6) is the equi-distant restart strategy $\tau_a = \tau_b = \frac{d}{N}$. The algorithm will end up with that solution, unless there exists a second solution. For this solution, it cannot be that τ_a and τ_b are both smaller or both larger than $\frac{d}{N}$, since then the intervals would not sum to d. Therefore, we can choose $\tau_b > \frac{d}{N}$ and $\tau_a < \frac{d}{N}$. Furthermore, it also must hold that $\tau_b \le \frac{d}{n_b}$. The algorithm utilises these facts to initialise an interval between `bottom` and `top` in which τ_b lies, and then breaks the interval in two at every iteration, until top $\approx$ bottom. At every iteration, it sets τ_b to the guess `top` and computes the belonging $\tau_a = \frac{d-n_b\tau_b}{n_a}$. It then tests if the sign of $h(\tau_b) - h(\tau_a)$ changes, to decide if τ_b lies in the upper or lower half. This test works correctly thanks to the particular shape of the lognormal hazard function. Note that since the algorithm divides the considered interval in two in every iteration, it takes logarithmic time to find the optimum for every pair n_a, n_b for which the algorithm is run.

We applied Algorithm 1 to the lognormal distribution with parameters $\mu = -2.3$ and $\sigma = 0.97$, and deadline $d = 0.7$. The parameters fit data collected in [4], but are otherwise arbitrary. Figure 2 shows typical behaviour if one considers a single restart. The equi-distant restart (at $\tau = 0.35$) is optimal, while the other equi-hazard points turn out to be minima ($\tau = 0.013$ or $\tau = 0.687$). The improvement in probability

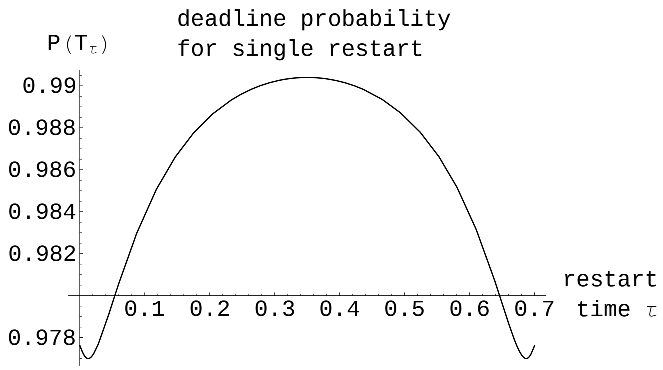

Fig. 2. Probability of meeting deadline for one restart ($d = 0.7, \mu = -2.3, \sigma = 0.97$).

of making the deadline is from 0.977 to 0.990. Table 1 shows results for increasing number of restarts, displaying all sets of equi-hazard intervals that are extrema. We see from the table that for this example equi-distant hazard rates always outperform the other equi-hazard points, and that the optimum is for three equi-distant restarts (and thus four intervals).

It turns out that equi-distant restarts are optimal in all experiments with log-normal distributions. Although we can construct examples in which for instance two non-equi-distant points outperform equi-distant points, for the lognormal distribution this only seems to be possible if no restart performs even better. Unfortunately, we have no proof for this phenomenon, but it gives us ground to use an algorithm that limits its search for optima to equi-distant points, which can be done even faster than Algorithm 1 for equi-hazard points. In the following algorithm we increase the number of equi-distant restart points (starting from 0), consider the probability of making the deadline for that number of restarts and stop as soon as we see no more improvement. This is a very advantageous stopping criterion since one needs not to set an arbitrary maximum on the number of restart points. We do not give the derivation of the correctness of this stopping criterion here, but instead close with the algorithm.

Algorithm 2 (Equi-distant Restart Intervals)

```
n=1; prob[1]=F(d);
Do{
  n++;
  prob[n] = 1 − (1 − F(d/n))^n;
}
Until (prob[n] < prob[n-1])
Return(d/(n − 1))
```

# restarts	equi-hazard intervals	$P(T_{\{\tau\}} < d)$
0	—	0.978
1	0.35, 0.35	0.990
1	0.013, 0.687	0.977
2	0.23, 0.23, 0.23	0.993
2	0.019, 0.34, 0.34	0.990
2	0.013, 0.013, 0.674	0.976
3	0.175, 0.175, 0.175, 0.175	0.99374
3	0.024, 0.225, 0.225, 0.225	0.993
3	0.019, 0.019, 0.331, 0.331	0.989
3	0.013, 0.013, 0.013, 0.660	0.976
4	0.14, 0.14, 0.14, 0.14, 0.14	0.99366
⋮	⋮	⋮

Table 1. Equi-hazard restart intervals and associated probability of meeting the deadline ($d = 0.7, \mu = -2.3, \sigma = 0.97$).

3 CONCLUSION AND OUTLOOK

This paper presents some basic algorithms to optimise the deployment of restart when one wants to maximise the probability a pre-defined deadline is met. These algorithms are tailored to probability density and hazard functions with lognormal shape, which we have found to be representative for simple Internet applications (HTTP Get [4]). The ultimate goal is to use the algorithms at run-time, but to do so, we must carefully research various issues, related to data analysis, learning, algorithms and implementation. This short paper offers a promising approach to one of those issues, namely algorithms.

References

1. B. Krishnamurthy and J. Rexford, *Web Protocols and Practice,* Addison Wesley, 2001.
2. S. M. Maurer and B. A. Huberman, “Restart strategies and Internet congestion,” in *Journal of Economic Dynamics and Control,* vol. 25, pp. 641–654, 2001.
3. A. van Moorsel and K. Wolter, “Analysis and Algorithms for Restart,” accepted for publication in *Quantitative Evaluation of Systems (QEST)*, Twente, The Netherlands, Sep. 2004.
4. P. Reinecke, A. van Moorsel and K. Wolter, “A Measurement Study of the Interplay between Application Level Restart and Transport Protocol,” to be published in *Springer Verlag LNCS,* at *International Service Availability Symposium,* Munich, May 2004.