# Self-Management of Systems through Automatic Restart

Katinka Wolter

Humboldt-Universität zu Berlin
Institut für Informatik
Unter den Linden 6, 10099 Berlin, Germany
wolter@informatik.hu-berlin.de

**Abstract.** Modern complex information systems require management mechanisms that operate to a large extent independently and autonomously. One such mechanism is the restart of components or transactions in case a failure in the system occurs. In this paper we introduce a pragmatic algorithm to determine close to optimal restart times on-line. We present a method for choosing best restart times based on empirical data, if no theoretical distribution is known. The best restart time is determined based on the empirical hazard rate. We study the sample size required to come to a reasonably good estimate, the effect of the failure probability of a job and issues of parameter selection for the hazard rate estimation. The application considered in this paper is the connection setup time in HTTP GET necessary for the download of web pages.

## 1 Introduction

In various situations in computer systems a restart of system components, a re-issuing of a request, or a re-establishment of a network connection improves the performance or availability of the component under consideration significantly. Not always is it known why precisely restart of a process or job becomes necessary or beneficial. Most Internet users, however, are familiar with the fact that clicking the reload button often helps in speeding up the download of a page, although we understand only to a limited extent what is happening exactly in the Internet. Another example is software 'aging', for which rejuvenation - the restart of the software environment - helps in preventing application failures hence also improves the completion time. But little understanding exists about the causes of aging and we are not usually able to identify the source of the problem and remove it. In practical situations, therefore, we will not be able to come to the required understanding to remove the problems, instead we want to optimise the deployment of 'black-box' restart to improve system availability or performance.

The use of restart has first been proposed for optimising Internet agent activities in [2], and further experiments have been carried out in [6]. [3] presents mathematics to optimise the expected download time, and based on this, [5] introduces a proxy server based architecture for restart including a software module for the computation of the optimal timeout value. Our objective is to automate restart, building on the above work. We decide on-line whether restart will be beneficial and when to do it. In this paper we simulate an on-line procedure by using increasingly more data from measurements taken earlier [5], but the applied methods can easily be included in a software module like the proxy server in [5] to be executed in real-time.

The shape of the hazard rate of a probability distribution indicates whether restart is beneficial. For empirical data the correct theoretical distribution is unknown and the hazard rate therefore needs to be estimated based on observations. Estimating the hazard rate is not a straightforward task, since it needs numerical computation of the derivative of the cumulative hazard rate. In this paper we derive and implement a new and simple rule based on the hazard rate that allows us to find the optimal restart time to maximise the probability of making a deadline. This rule approximates the optimal restart time independent of the exact value of the deadline, and is asymptotically exact (when the deadline increases). Moreover, the rule is very simple, making it a likely candidate for run-time deployment. Not in all cases doe the optimal restart time exists. Restart is applicable to a system if (and only if) the rule finds an optimal restart time. So, our simple rule actually serves a two-fold purpose: it enables us to decide whether restart will be beneficial in the given situation, and if so, it provides us with the optimal restart time.

We apply the rule to data sets we collected for HTTP, thus mimicking the on-line execution of the algorithm. We explore how much data is required to arrive at reasonable estimates of the optimal restart time. We also study the effect of failed HTTP requests by artificially introducing failures in the data sets. Based on these explorations we provide engineering insights useful for run-time deployment of our algorithm.

Finally, an important technical detail when using the hazard rate is the value of the bandwidth in the required smoothing algorithm. Based on many experiments, we obtain a reasonably robust rule for setting the bandwidth based on the variance of observations. This greatly speeds up the execution of the algorithm, thus improving its on-line performance.

## 2 The restart model

To automate restart, we need to decide the metric of interest, and postulate a mathematical model. In our earlier work, we use restart to minimise the expected download time of a web page in an algorithm that does not make use of the hazard rate [5]. But restart can also be used to increase the probability of making a deadline and for a finite deadline and a finite number of restarts algorithms based on the theoretical distribution and lognormally distributed completion times have been presented in [4]. In our experiments we measured different variables involved in the download of a web page. In this paper we only use the connection setup time from data sets already studied in [5]. We again study the probability of making a deadline, but unlike the formulation in [4] here we use an approximation to estimate the optimal restart time. Using the approximation we can formulate a very simple rule based on the hazard rate, which in fact is independent of the deadline to be met.

Our mathematical model assumes statistical independence of consecutive preemptive tries. We found this very often to be a realistic assumption in HTTP downloads from one URL [5]. Let the random variable $T$ denote the completion time of a job, with probability distribution $F(t), t \in [0, \infty)$. Assume $\tau$ is a restart time, and introduce the random variable $T_\tau$ to denote the completion time when an unbounded number of retries is allowed. That is, a retry takes place periodically, every $\tau$ 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 $T_\tau$. A distribution can equally well be described by the hazard rate

$$h(t) = \frac{f(t)}{1 - F(t)}$$

and the cumulative hazard

$$H(t) = \int_{s=0}^{t} h(s) ds$$

which both are very important throughout our analysis. One useful relation between the cumulative hazard rate and a distribution function is given by

$$H(t) = -\log(1 - F(t)).$$

Restart at time $\tau$ is beneficial only if the probability $F_\tau(t)$ of making the deadline $t$ under restart is greater than the probability of making the deadline without restart, i.e.

$$F_\tau(t) > F(t). \tag{1}$$

As we have shown in [4], one can intuitively reason about the completion time distribution with restarts as Bernoulli trials. At each interval between restarts there is a probability $F(\tau)$ the completion 'succeeds.' Hence, if the time $t$ is a multiple of the restart time $\tau$, we can relate the probability of missing the deadline without and with restart through:

$$1 - F_\tau(t) = (1 - F(\tau))^{\frac{t}{\tau}}. \tag{2}$$

Eqn. (2) is correct only for values of $t$ and $\tau$ such that $t$ is an integer multiple of $\tau$. But if we ignore this fact, or simply accept (2) as an approximation, we can find the optimal restart time in a straightforward way. Surprisingly, it turns out that the approximation gives us a restart time independent of the deadline $t$, which is optimal in the limit $t \to \infty$. That is, it optimises the tail of the completion time distribution under restarts, and is therefore beneficial for many other metrics as well, such as higher moments of the completion time.

**Theorem 1.** *If the restart time $\tau^*$ is an extreme (in $\tau$) of $(1-F(\tau))^{\frac{t}{\tau}}$ for* any *deadline $t$ then $\tau^*$ is a point where $\tau^*.h(\tau^*) = -\log(1-F(\tau^*))$;*

*Proof.* We use that

$$\frac{d}{dx}(g(x))^x = (g(x))^x \left( \frac{x \frac{d}{dx} g(x)}{g(x)} + \log(g(x)) \right). \tag{3}$$

$\tau^*$ is an extreme when the derivative of $(1-F(\tau))^{\frac{t}{\tau}}$ equates to 0:

$$\frac{d}{d\tau}(1-F(\tau))^{\frac{t}{\tau}} = (1-F(\tau))^{\frac{t}{\tau}} \left( \frac{f(\tau)\tau}{1-F(\tau)} + \log(1-F(\tau)) \right) = 0. \tag{4}$$

Irrespective of the value of $t$ it immediately follows that

$$\frac{f(\tau)}{1-F(\tau)} = \frac{-\log(1-F(\tau))}{\tau}, \tag{5}$$

and thus the conclusion holds if and only if the premiss holds. □

Eqn. (5) can be rewritten as

$$\tau \cdot h(\tau) = H(\tau) \tag{6}$$

where $H(\tau)$ can be interpreted as the surface under the hazard rate curve up to point $\tau$. We can therefore reason that (5) expresses the fact that if (1) holds there exists a point on the hazard rate curve such that the rectangle defined by x- and y-value of this point equals the integral under the hazard rate curve up to this point. We will refer to 6 as the *rectangle equals surface rule.* This very appealing and simple rule is used in this paper for an empirical hazard rate to find an empirical optimal restart time that maximises the probability of completion, that is the probability of making an infinite deadline.

It should be noted that if the hazard rate is monotonously increasing, no value of $\tau$ exists, such that (6) holds. In that case restart will not help increasing the probability of completion. Only if the hazard rate decreases after some point a value of $\tau$ exists, such that (6) holds. Only then restart can be applied successfully.

## 3 Estimating the hazard rate

It follows from (6) that an estimate $\hat{h}(t)$ of the hazard rate curve is needed to determine the optimal restart time following the *rectangle equals surface* rule. We will in this section provide the main steps of how to estimate the hazard rate and implement the rule (6) in an algorithm. Some details are shifted to the appendix. We use the theory on survival analysis in [1].

The hazard rate $h(t)$ can not be estimated directly from a given data set. Instead, first the cumulative hazard rate $H(t)$ is estimated and then the hazard rate itself is computed as a numerical derivative.

Let us consider a sample of $n$ individuals, that is $n$ completions in our study. We sample the completion times and if we order them, we obtain a data set of $D$ distinct times $t_1 \leq t_2 \leq \ldots \leq t_D$ where at time $t_i$ there are $d_i$ events, that is $d_i$ completions take time $t_i$. The random variable $Y_i$ counts the number of jobs that need more or equal to $t_i$ time units to complete. We can write $Y_i$ as

$$Y_i = n - \sum_{j=1}^{i-1} d_j$$

All observations that have not complete at the end of the regarded time period, usually time $t_D$, are called *right censored*. There are $Y_n - d_n$ right censored observations. The experimental data we use falls in that category, since Internet transactions commonly use TCP, which aborts (censors) transactions if they do not succeed within a given time.

The hazard rate estimator $\hat{h}(t)$ is the derivative of the cumulative hazard rate estimator $\hat{H}(t)$, which is defined in Appendix A. It is estimated as the slope of the cumulative hazard rate. Better estimates are obtained when using a kernel function to smooth the numerical derivative of the cumulative hazard rate. The smoothing is done over a window of size $2b$. A bad estimate of the hazard rate will yield a bad estimate of the optimal restart time and the optimised metric is very sensitive to whether the restart time is chosen too short. Therefore a good estimate of the hazard rate is needed.

Let the magnitude of the jumps in $\hat{H}(t)$ and in the estimator of its variance $\hat{V}[\hat{H}(t)]$ at the jump instants $t_i$ be $\Delta\hat{H}(t_i) = \hat{H}(t_i) - \hat{H}(t_{i-1})$ and $\Delta\hat{V}[\hat{H}(t_i)] = \hat{V}[\hat{H}(t_i)] - \hat{V}[\hat{H}(t_{i-1})]$. Note that $\Delta\hat{H}(t_i)$ is a crude estimator for $\hat{h}(t_i)$.

The kernel-smoothed hazard rate estimator is defined separately for the first and last points, for which $t-b < 0$ or $t+b > t_D$. For inner points with $b \leq t \leq t_D - b$ the kernel-smoothed estimator of $h(t)$ is given by

$$\hat{h}(t) = b^{-1} \sum_{i=1}^{D} K\left(\frac{t - t_i}{b}\right) \Delta\hat{H}(t_i). \tag{7}$$

The variance of $\hat{h}(t)$ is needed for the confidence interval and is estimated by

$$\sigma^2[\hat{h}(t)] = b^{-2} \sum_{i=1}^{D} K\left(\frac{t-t_i}{b}\right)^2 \Delta\hat{V}[\hat{H}(t_i)]. \tag{8}$$

The function $K(.)$ is the Epanechnikov kernel defined in Appendix B.

A $(1-\alpha)\cdot 100\%$ point wise confidence interval around $\hat{h}(t)$ is constructed as

$$\left[\hat{h}(t)\exp\left[-\frac{z_{1-\alpha/2}\sigma(\hat{h}(t))}{\hat{h}(t)}\right], \hat{h}(t)\exp\left[\frac{z_{1-\alpha/2}\sigma(\hat{h}(t))}{\hat{h}(t)}\right]\right]. \tag{9}$$

where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$ quantile of the standard normal distribution.

The choice of the right bandwidth $b$ is a delicate matter, but is important since the shape of the hazard rate curve greatly depends on the chosen bandwidth (see figure 2) and hence a badly chosen bandwidth will have a serious effect on the optimal restart time. One way to pick a good bandwidth is to use a cross-validation technique of determining the bandwidth that minimises some measure of how well the estimator performs. One such measure is the *mean integrated squared error* (MISE) of $\hat{h}$ over the range $\tau_{\min}$ to $\tau_{\max}$. The mean integrated squared error can be found in Appendix C. To find the value of $b$ which minimises the MISE we find $b$ which minimises the function

$$g(b) = \sum_{i=1}^{M-1}\left(\frac{t_{i+1}-t_i}{2}\right)(\hat{h}^2(t_i)+\hat{h}^2(t_{i+1})) - 2b^{-1}\sum_{i\neq j} K\left(\frac{t_i-t_j}{b}\right)\Delta\hat{H}(t_i)\Delta\hat{H}(t_j). \tag{10}$$

Then $g(b)$ is evaluated for different values of $b$. Each evaluation of $g(b)$ requires the computation of the estimator of the hazard rate. The optimal bandwidth can be determined only in a trial-and-error procedure. We found in our experiments that the optimal bandwidth is related with the size of the data set and the variance of the data. We use the standard deviation to determine a starting value and then do a simple step-wise increase of the bandwidth until $g(b)$ takes on its minimal value. In case the hazard rate is increasing in the first steps, we decrease $b$ and start again, since then we are obviously beyond the minimum already. In our experiments and in the literature we always found a global minimum, never any local minima. Advanced hill-climbing algorithms can be applied to find the minimum more quickly and more accurately than we do here.

Once the best estimate of the hazard rate is found we need to determine the point $i^*$ that satisfies the *rectangle equals surface* rule (6).

The following simple algorithm determines the optimal restart time $\tau^*$ by testing all observed points $t_i, i = 1, \ldots, n$ as potential candidates.

**Algorithm 1 (Optimal restart time)**

```
Input ĥ, Ĥ and t;
i = 1; #(t = t_1, ..., t_n)
While((i < n) and (t_i · ĥ(t_i) > Ĥ(t_i)) ) {
    i++;
}
return t_i;
```

This algorithm returns in the positive case the smallest observed value that is greater than the estimated optimal restart time $\tau^*$.

In many cases, however, the studied data set does not contain observations large enough to be equal or greater than the optimal restart time. Then we extrapolate the estimated hazard rate to find the point where the rectangle equals the surface under the curve. Assuming we have a data set of $n$ observations $t_i, i = 1, \ldots, n$, at first the slope of the estimated hazard rate at the end of the curve is determined as the difference quotient

$$\text{slope} \; = \frac{\hat{h}(t_n) - \hat{h}(t_{n-1})}{t_n - t_{n-1}}. \tag{11}$$

Then $t_\tau = t_n + \Delta t$ is determined such that for $t_\tau$ eqn. (5) holds.

$$\begin{aligned} (t_n + \Delta t) \cdot (\hat{h}(t_n) + \text{slope} \cdot \Delta t) &= \hat{H}(t_n) \cdot \text{slope} \;\; \cdot \Delta t \cdot t_n \\ \Longleftrightarrow \qquad \Delta t &= \frac{\hat{H}(t_n) - t \cdot \hat{h}(t_n)}{\hat{h}(t_n) - 2 \text{ slope } t_n - \hat{H}(t_n) - \text{slope}}. \end{aligned} \tag{12}$$

### 3.1 Complexity

The computational complexity depends in first place on the number of iterations needed to find the optimal bandwidth for the hazard rate estimator. In our experiments we used a heuristic based on the standard deviation of the data set that gave us the optimal bandwidth often in less than 5 iterations, but sometimes took up to 20 iterations.

The second important parameter is the number of observations considered. Each iteration on the bandwidth requires the computation of the estimated hazard rate, which in turn needs traversing all observations and uses for each point a window of size $2b$. Complexity of the hazard rate estimator is therefore at most $O(n^2)$. Improving on the heuristic for the bandwidth, so that in all cases only few iterations are needed is certainly worth while.

## 4 Experiments

We have implemented the algorithm to estimate the hazard rate and determine the optimal restart time as defined in theorem 1. The implementation is done in Mathematica and has been applied to the HTTP connection setup data studied in [5]. This data in fact consists of the time needed for TCP's three-way handshake to set up a connection between two hosts.

In our experiments we investigate various issues. One is the uncertainty introduced by small sample sizes. The available data sets consist of approximately one thousand observations for each URL, that is thousand connection setup times to the same Internet address. We use these data sets and take subsets of first one hundred then two hundred observations etc. as indicated in the caption of the figure and in the table. We do not use data of different URLs in one experiment since we found that very often different URLs have different distributions or at least distribution parameters. Furthermore, the application we have in mind is web transactions between two hosts.

The data we study is data set '28' consisting of the connection setup times to http://nuevamayoria.com, measured in seconds. This data set shows characteristics such as a lower bound on all observation and a pattern of variation which we found in many other data sets as well, even though usually not with the same parameters. The chosen data set is therefore to be seen as one typical representative of a large number of potential candidates. The considered connection setup times are shown in figure 1. The largest observation in this data set is 0.399678 seconds.

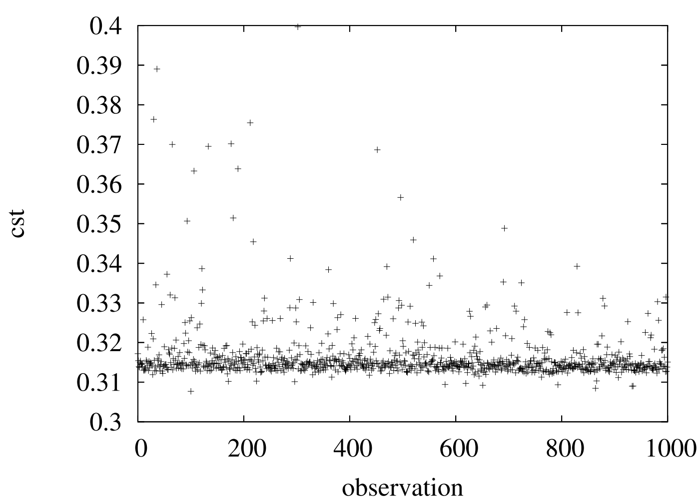


**Fig. 1.** Data set No. 28; connection setup times (in seconds).

For each of the mentioned subsamples the optimal smoothing factor, or bandwidth, is computed by evaluating (10) several times, finding the minimum in a simple search. Figure 2 shows estimates of the hazard rate for different values of the bandwidth. Parameter $b1$ is too large, whereas $b2$ is too small, $b3$ is the one that minimises the error and is therefore the optimal bandwidth. One can see that too large a bandwidth leads to an extremely smooth curve, whereas too small a bandwidth produces over-emphasised peaks. From the figure one might conclude that rather too large a bandwidth should be chosen than one that is too small, but more experiments are needed for a statement of this kind. Using the optimal bandwidth, the hazard rate and its 95% confidence interval

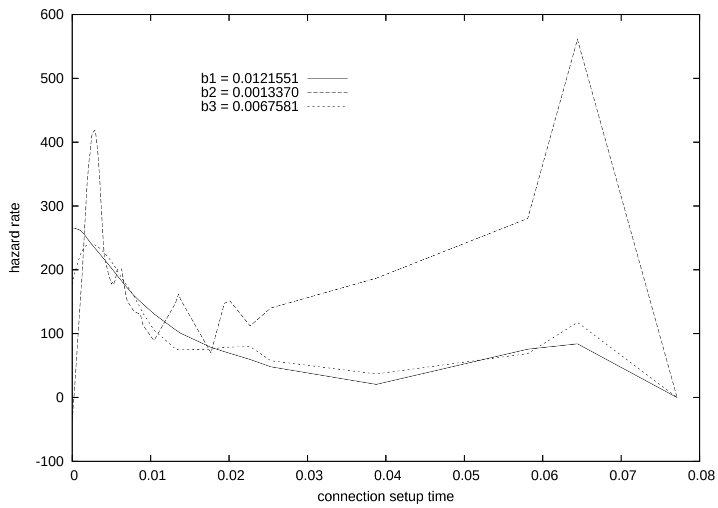


**Fig. 2.** Hazard rate for data set No. 28 and different values of the bandwidth $b$.

are estimated according to (7) and (9). Finally, for each estimated hazard rate the optimal restart time $\tau^*$ is computed using algorithm 1. In some cases, the algorithm finds the optimal restart time, since the data set includes still an observation greater than the optimal restart time. If the data set has no observation large enough to be greater than the optimal restart time, we extrapolate according to (12). The optimal restart times are drawn as vertical bars in the plots in figures 3 and 4. Note that in figure 3 although it looks like all optimal restart times are extrapolated in fact none of them is. The extrapolated optimal restart times are indicated by an asterisk in table 1.

The hazard rate curve has no value at the point of the largest observation, since for the numerical derivation always two data points are needed. Furthermore, because of the limited amount of data in the tail, it is not surprising that the confidence interval at the last observations grows rapidly.

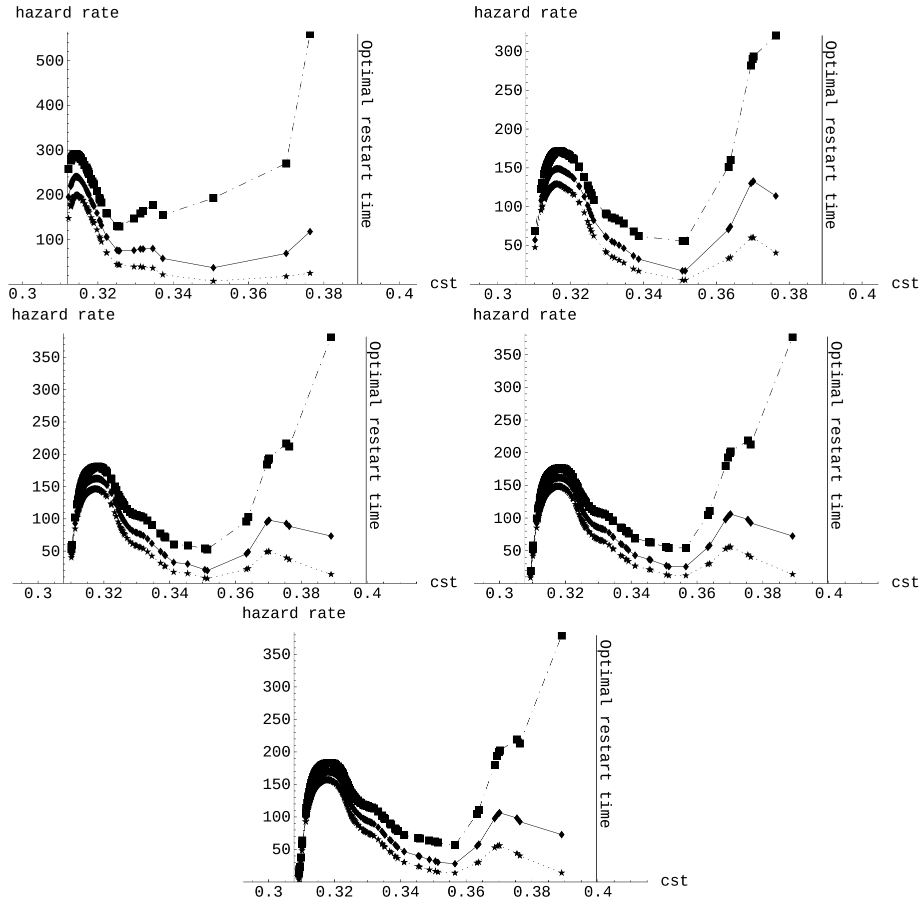


**Fig. 3.** Estimated hazard rates and confidence intervals for the estimates for increasing sample size (top row $n = 100$ and $n = 200$, middle row $n = 400$ and $n = 600$, bottom row $n = 800$) and failure probability 0.0

Table 1 shows some characteristics obtained in the program runs for data set 28. Each block of the table belongs to a subset of size $n$ with corresponding standard deviation. The standard deviation changes as more observations come into consideration. For each subsample three different cases are studied. In the first one only the $n$ observations are used and the failure probability equals either zero, or the relative fraction of observations that are greater than 3.0. This threshold is the first retransmission timeout of TCP and hence observations greater 3.0 are (somewhat arbitrarily) censored and retried. We treat them as censored observations and all censored observations contribute to the failure probability. Data set '28' does not have any such censored observations, but

| $n = 100$, StdDev = 0.0121551 | | | $n = 200$, StdDev = 0.0117341 | | |
|---|---|---|---|---|---|
| failure prob. | bw | $\tau^*$ | failure prob. | bw | $\tau^*$ |
| 0.0 | 0.006758 | 0.389027 | 0.0 | 0.011557 | 0.389027 |
| 0.666667 | 0.001779 | 0.597251* | 0.666667 | 0.001398 | 0.674306* |
| 0.8 | 0.001779 | 0.554513* | 0.8 | 0.001271 | 0.638993* |
| $n = 300$, StdDev = 0.0106746 | | | $n = 400$, StdDev = 0.010383 | | |
| failure prob. | bw | $\tau^*$ | failure prob. | bw | $\tau^*$ |
| 0.0 | 0.011742 | 0.389027 | 0.0 | 0.010226 | 0.399678 |
| 0.666667 | 0.001272 | 0.333271 | 0.666667 | 0.001124 | 0.333271 |
| 0.8 | 0.001156 | 0.333271 | 0.8 | 0.001124 | 0.333271 |
| $n = 500$, StdDev = 0.00997916 | | | $n = 600$, StdDev = 0.00941125 | | |
| failure prob. | bw | $\tau^*$ | failure prob. | bw | $\tau^*$ |
| 0.0 | 0.010977 | 0.399678 | 0.0 | 0.010352 | 0.399678 |
| 0.666667 | 0.001081 | 0.333271 | 0.666667 | 0.001138 | 0.333271 |
| 0.8 | 0.001081 | 0.333271 | 0.8 | 0.001019 | 0.333271 |
| $n = 700$, StdDev = 0.00895504 | | | $n = 800$, StdDev = 0.00851243 | | |
| failure prob. | bw | $\tau^*$ | failure prob. | bw | $\tau^*$ |
| 0.0 | 0.009850 | 0.309209 | 0.0 | 0.0103 | 0.399678 |
| 0.66667 | 0.000970 | 0.333271 | 0.66667 | 0.000922 | 0.332014 |
| 0.8 | 0.000970 | 0.333271 | 0.8 | 0.000922 | 0.332014 |
| $n = 900$, StdDev = 0.00816283 | | | $n = 1000$, StdDev = 0.00784583 | | |
| failure prob. | bw | $\tau^*$ | failure prob. | bw | $\tau^*$ |
| 0.0 | 0.009877 | 0.308456 | 0.0 | 0.009493 | 0.308456 |
| 0.6667 | 0.000884 | 0.332014 | 0.6667 | 0.000949 | 0.333271 |
| 0.8 | 0.000884 | 0.332014 | 0.8 | 0.000850 | 0.332014 |

**Table 1.** Optimal restart time ($\tau^*$) and optimal bandwidth (bw) for different subsample sizes of data set 28 and different failure probabilities

many other data sets do. The second group consists of the $n$ observations plus $2n$ censored ones and has therefore failure probability $2/3$, or a little higher if there are additional censored observations present in the data set. Analogously, the third group has $n + 4n$ observations and a failure probability of $n/5n = 0.8$ (or more if there are censored observations in the data set).

If we look at the results for failure probability zero, also plotted in figure 3 for $n = 100, 200, 400, 600, 800$ we see that the small data sets lead to an overestimated optimal restart time (if we assume that the full 1000 observations give us a *correct* estimate), but the 'correct' value is overestimated by less than 5%.

We used such high, and perhaps unrealistic, failure probabilities in our study since a failure probability of e.g. 0.1 does not show in the results at all. Looking at the results for the different sample sizes in the group with high failure probability, we also find that with the small samples the optimal restart time gets overestimated.

We also investigate the impact of the failure probability within a group of fixed sample size. The failure probability is increased by subsequently adding more failed (and hence censored) observations and then estimates for the hazard rate and optimal restart time are computed. The failed attempts of course increase the sample size. We notice (as can be seen in table 1) that the bandwidth used for estimating the hazard rate decreases for increasing failure rate, while the sample standard deviation is computed only from non-failed observations and hence does not change with changing failure probability. We found in [4] that for theoretical distributions the optimal restart time decreases with increasing failure probability. Typically our experiments agree with this property, which, however, is not true for some subsets of data set '28'.

An additional purpose of the experiments was to find out whether we can relate the optimal bandwidth to any characteristic of the data set. In the literature no strategy is pointed out that helps in finding the optimal bandwidth quickly. In our implementation we set the standard deviation as a starting value for the search. If we have no censored observations (failure probability zero) we always find the optimal bandwidth within less than five iterations. If the data set has many censored observations the optimal bandwidth roughly by factor 5 and we need more iterations to find that value, since our heuristic has a starting value far too large in that case.

Figure 4 compares two hazard rates using another data set for data with identical sample size, the first has zero failure rate and the second has failure rate 0.8. It can be seen that the high number of added censored observations leads to a much more narrow hazard rate, with lower optimal restart time. Note that this figure is based on a different data set than the ones above.

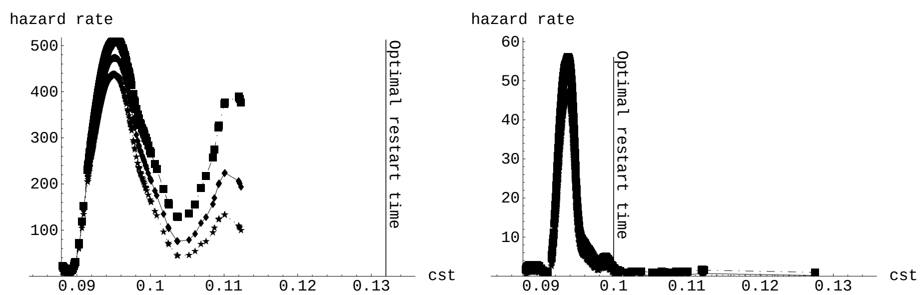


**Fig. 4.** Estimated hazard rates and confidence intervals for sample size $n = 1000$, failure probability 0.0 (left) and 0.8 (right)

In summary, we have provided an algorithm that gives us an optimal restart time to maximise the probability of meeting a deadline only if restart will indeed help maximising that metric. So if the algorithm returns an optimal restart time we can be sure that restart will help. We found a heuristic based on the variance of the data that helps in finding quickly the bandwidth parameter needed for the hazard rate estimator. We found that small data sets usually lead to an overestimated optimal restart time. But we saw earlier (in [4]) that an overestimated restart time does much less harm to the metric of interest than an underestimated one and we therefore willingly accept overestimates.

## 5 Discussion and Conclusions

Self-management of modern, complex systems can include among others the automatic restart of jobs, or transactions if they are performing badly. As the considered metric we chose in this paper the probability of completion before an arbitrary deadline under unlimited number of allowed restarts. We derived a *surface equals rectangle* rule for the optimal restart time that is based on the hazard rate. We implemented an algorithm to estimate the hazard rate from a given data set and to determine the optimal restart time. The *surface equals rectangle* rule provides an answer to the question of whether restart makes sense in a given scenario. If an optimal restart time is found at all we can be sure that the shape of the hazard rate is such that restart makes sense and (1) holds. A very simple heuristic was used to quickly find the best bandwidth for the hazard rate estimation. The benefit of our algorithm is that it gives reasonably good estimates on small data sets and can hence be used for fast estimates in on-line automatic restart.

The run-time of the algorithm depends on the considered number of observations and on the number of iterations needed to find a good bandwidth for the hazard rate estimation. We found that for our smaller data sets with up to 400 observations less than 5 iterations are needed and the algorithm is very fast. We did not evaluate CPU time and the Mathematica implementation is not run-time optimised, but a suggestion for an optimal restart time in the above setting can be provided within a few seconds. If, however, the data set grows large, has e.g. more than 800 observations, each iteration on the bandwidth takes in the order of some one or two minutes. The polynomial complexity becomes relevant and the method is no longer applicable in an on-line algorithm.

A good heuristic for choosing the optimal bandwidth is a key part in the whole process. The better the first guess, the less iterations are needed and the faster we obtain the optimal restart time. We cannot compare our heuristic to others since in the literature nothing but pure 'trial and error' is proposed. But we can say, that for small data sets and failure probability zero the optimal restart time is obtained very fast since the heuristic provides a good first estimate of the bandwidth.

In our experience the smallest data sets were usually sufficient for a reasonably good estimate of the optimal restart time. The optimal restart time will always be placed at the end of the bulk of the observations and some few hundred observations are enough to get a notion of 'bulk' and 'end of the bulk'. If we consider that some web pages consist of up to 200 objects a data set of 100 samples is not hard to obtain or unrealistic. In Internet transactions some hundred samples are very quickly accumulated. Furthermore, small samples seem to overestimate the optimal restart time, which does the maximised metric much less harm than underestimation.

In practical applications the required number of observations is no limitation to the applicability of our method and having not too much data has a positive effect on the run-time while it does not deteriorate the obtained result. The proposed method is well-suited as an on-line restart module.

One may argue that if everybody applies restart networks become more congested and response times will drop further. And in fact restart changes the TCP timeout - for selected applications. In our measurements we found that less than 0.5% of all connection setup attempts fail. Our method tries to detect failures faster than the TCP timeout and to restart failed attempts, since for slow connections restart typically does not lead to improved response time, whereas for failed connections in many cases it does. Failed attempts, however, are so rare that restarting those does not impose significant extra load on a network, while potentially speeding those up enormously.

## References


1. J. P. Klein and M. L. Moeschberger. *Survival Analysis, Techniques for Censored and Truncated Data.* Springer, 1997.
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," in *Proc. 1st International Conference on the Quantitative Evaluation of Systems (QEST)*, pp. 195-204, Twente, Netherlands, Sept. 2004.
4. A. van Moorsel and K. Wolter, "Making Deadlines through Restart," in *Proc. 12th GI/ITG Conference on Measuring, Modelling and Evaluation of Computer and Communication Systems (MMB 04)*, pp. 155–160, Dresden, Germany, Sept. 2004.
5. P. Reinecke, A. van Moorsel and K. Wolter, "A Measurement Study of the Interplay between Application Level Restart and Transport Protocol," in *Proc. International Service Availability Symposium*, Munich, Germany, May 2004.
6. M. Schroeder and L. Buro, "Does the Restart Method Work? Preliminary Results on Efficiency Improvements for Interactions of Web-Agents," in T. Wagner and O. Rana, editors, *Proceedings of the Workshop on Infrastructure for Agents, MAS, and Scalable MAS at the Conference Autonomous Agents 2001,* Springer Verlag, Montreal, Canada, 2001.

# Appendix

## A Cumulative hazard rate

The cumulative hazard rate is estimated using the Nelson-Aalen estimator, which has especially good small sample performance. The Nelson-Aalen estimator is

$$\hat{H}(t) = \begin{cases} 0 & \text{if } t \leq t_1 \\ \sum_{t_i \leq t} \frac{d_i}{Y_i} & \text{if } t_1 \leq t. \end{cases} \tag{13}$$

The estimated variance of the Nelson-Aalen estimator is

$$\sigma_H^2(t) = \sum_{t_i \leq t} \frac{d_i}{Y_i^2}. \tag{14}$$

## B Epanechnikov kernel

For the kernel $K(.)$ the Epanechnikov kernel is used

$$K(x) = 0.75(1 - x^2) \qquad \text{for } -1 \leq x \leq 1 \tag{15}$$

as it is shown in [1] to be often more accurate than other kernel functions. When $t - b < 0$ or $t + b > t_D$ the symmetric kernel must be transformed into an asymmetric one, which is at the lower bound with $q = t/b$

$$K_q(x) = K(x)(\alpha + \beta x), \quad \text{for } -1 \leq x \leq q, \tag{16}$$

where

$$\alpha = \frac{64(2 - 4q + 6q^2 - 3q^3)}{(1+q)^4(19 - 18q + 3q^2)} \tag{17}$$

$$\beta = \frac{240(1-q)^2}{(1+q)^4(19 - 18q + 3q^2)} \tag{18}$$

For time-points in the right-hand tail $q = (t_D - 1)/b$ the kernel function is $K_q(-x)$.

## C Bandwidth estimation

The *mean integrated squared error* (MISE) of the estimated hazard rate $\hat{h}$ over the range $\tau_{\min}$ to $\tau_{\max}$ is defined by

$$\begin{aligned} MISE(b) &= E\left(\int_{\tau_{\min}}^{\tau_{\max}} [\hat{h}(u) - h(u)]^2 \, du\right) \\ &= E\left(\int_{\tau_{\min}}^{\tau_{\max}} \hat{h}^2(u) \, du\right) - 2E\left(\int_{\tau_{\min}}^{\tau_{\max}} \hat{h}(u)h(u) \, du\right) \\ &\quad + E\left(\int_{\tau_{\min}}^{\tau_{\max}} h^2(u) \, du\right). \end{aligned} \tag{19}$$

This function depends on the bandwidth $b$ used in the Epanechnikov kernel. The last term does not contain $b$ and can be ignored when finding the best value of $b$. The first term is estimated by $\int_{\tau_{\min}}^{\tau_{\max}} \hat{h}^2(u)\, du$. We evaluate $\hat{h}(u)$ at a not necessarily equidistant grid of points $\tau_{\min} = u_1 < u_2 < \ldots < u_M = \tau_{\max}$ and apply the trapezoid rule. The second term we approximate by a cross-validation estimate suggested by Ramlau-Hansen where we sum over the event times between $\tau_{\min}$ and $\tau_{\max}$.