Dieter Hertweck, Christian Decker (Eds.): Digital Enterprise Computing 2016,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2016 201

Log-Data analysis through Tick-Discretisation

Christian Knoedler[1], Bernhard Moessner, Ilia Petrov [2]

Abstract: Nowadays almost every major company has a monitoring system and produces log data to analyse their systems. To perform analysation on the log data and to extract experience for future decisions it is important to transform and synchronize different time series. For synchronizing multiple time series serveral methods are provided so that they are leading to a synchronized uniform time series. This is achieved by using discretisation and approximation methodics. Furthermore the discretisation through Ticks is demonstrated, as well as the repsectivly illustrated results.

Keywords: discretisation, data analysation, time series, synchronisation, correlation

1 Introduction

Nowadays companies collect increasing amounts of data. Collecting log-data is becoming increasingly important to monitor the performance of IT -infrastructures, security flaws, hardware or system failures, user behaviour. Nevertheless, not all log-data are suitable for correlations, analyses, and actionable analytics.

In this paper, a real set of log-data generated by the performance monitoring infrastructure of a large hosting company is used. The infrastructure generates a log-entry if an error or warning event in one of the monitored systems occurs. Each produced log entry contains the SystemID as well as cpu usage, memory usage, priority levels and more details for hard- and software usage as well as responsibilities. Furthermore, each log entry contains a timestamp, indicating the system time of the error occurrence. As error conditions occur unpredictably on different systems, the timestamps are not set in a *periodic* manner, but rather as the event occurs. The log-data also contains information about each individual system/priority/type.

Since forecasts as to when, how often and why the problem will occur next are required, it is important to analyse the produced records over a period of time for each system to discover statistical causalities. In this paper, regression is used as standard statistical analysis method [FLS09]. *A prerequisite for applying regression analysis is a uniform time series, whereas the underlying log-data are asynchronous discontinuous time series. Therefore a uniform time series must be generated, from multiple time series of the collected log-data.* The contribution of this paper is the following: statistical analyses applied on log-data time series synchronised after discretisation yield an unacceptably high inaccuracy.

[1] Corresponding author: Christian.Knoedler@student.reutlingen-university.de
[2] Reutlingen University, Informatics, Alteburgstraße 150, 72762 Reutlingen

2 Log-data as synchronous time series

Consider two time series $X^k = (t_i^k, x_i^k)\ k = 1,2$, represented by their timestamps $t_i^k : i \in \mathbb{N}$ and their corresponding values x_i^k. For example, these values can be CPU utilisation in [%], or memory usage as illustrated in Figure 2. Two such time series are not time-synchronous, therefore $t_i^1 \neq t_i^2$.

If the monitoring system did not produce a log entry, clearly no error has occurred and it might seem reasonable to assume that zero-values can be used for the time series value (Figure 1). Yet, this is only possible if the value represents the number of occurrences. Otherwise, if the value represents measurements (as shown in Figure 2) it can be assumed that the missing values between two event occurrences are undefined, but can be interpolated.

In fact this is only a subset of the possibilities for interpreting missing values in time series [Ha94]. There are other alternatives such as: (a) all missing values are substituted with zeros; (b) for each missing value between x_i^k and x_{i+1}^k use value x_i^k; (c) for each missing value between x_i^k and x_{i+1}^k use value x_{i+1}^k; (d) interpolate values between x_i^k and x_{i+1}^k.

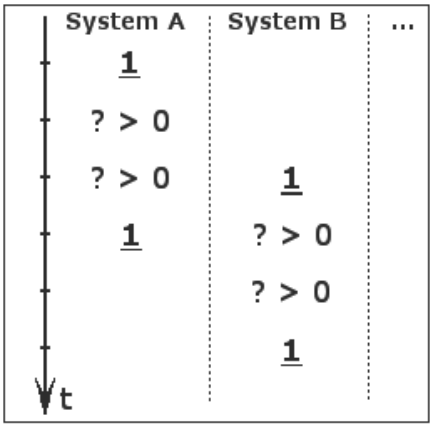

Fig. 1: Zero values for missing log data

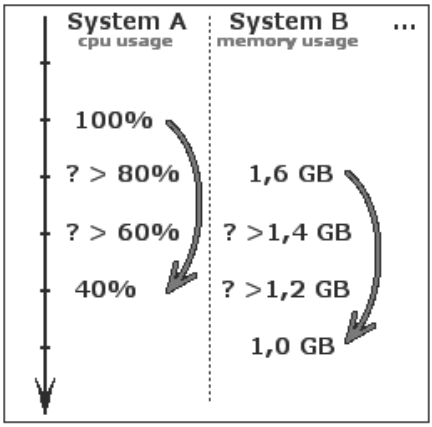

Fig. 2: Interpolate missing log data

3 Methods

To synchronize two time series it is needed to fill missing values in each time series to generate a synchronized time series for further analysis. This can be accomplished in two possible ways by applying: (i) approximation[Po81]; or (ii) discretisation.

3.1 Approximation

First, all the records of the two time series to be synchronised are combined and brought in an order so that:

$$V = \{t_1^1, t_2^1, ..., t_n^1, t_1^2, t_2^2, ..., t_m^2\} \quad (1)$$

$$= \{v_1, v_2, ..., v_e\}, \qquad v_i < v_{i+1} \quad (2)$$

As a result, two synchronised series of entries $\hat{X}^k : (v_i, \hat{x}_i^k)$ are produced, for which the following conditions apply:

$$\hat{x}_i^k = \begin{cases} x_j^k & \text{falls } v_i = t_j^k \\ 0 & \text{sonst} \end{cases} \quad (3)$$

3.2 Discretisation

Discretisation[KK06] was used as means for log-data preparation. First, a set of Ticks $D = d_1, d_2, ..., d_e$ are defined, for instance equidistant. Consequently, a discrete point in time (tick) d_j is assigned to each point in time t_i:

$$\alpha^k(i) = min\left\{ j \in \{1, ..., e\} \,\middle|\, |t_i^k - d_j| = \min_{\mu=1...e} |t_i^k - d_\mu| \right\} \quad (4)$$

$$I_j^k = \left\{ i = \{1, \ldots, n\} \,\middle|\, \alpha^k(i) = j \right\} \quad (5)$$

Figure 3 shows an example of such an assignment.

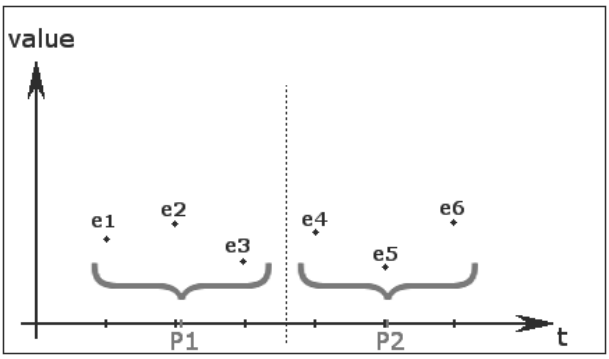

Fig. 3: Example of tick discretisation

Different types of analysis can be performed on the resulting discretised time series. For example, the average value (6) or use the number of occurrences (7) can now be computed for the discretised time series $\hat{X}^k : (d_j, \hat{x}_j^k)$:

$$\hat{x}_j^k = \frac{1}{|I_j^k|} \sum_{i \in I_j^k} x_i^k \tag{6}$$

$$\hat{x}_j^k = \sum_{i \in I_j^k} x_i^k \tag{7}$$

$\hat{X}^1$ and $\hat{X}^2$ are automatically synchronous.

4 Result and Conclusion

After synchronising the time series with discretisation a synchronous time series is produced that is suitable for different types of analysis. In the present scenario, different query analyses for the number of occurrences for different systems were performed. Those resulted in (Figure 4) strongly correlating results.

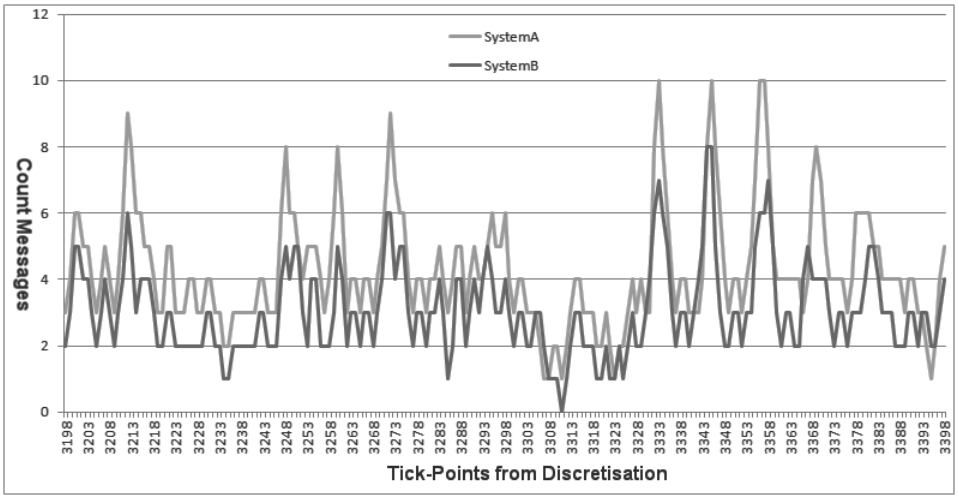

Fig. 4: Results of tick discretisation

In fact, this correlation results from the distance between the points P_i and P_{i+1}. To choose the right distance between these points you have to consider on the one hand the performance and resources used by the discretisation algorithm and the other hand the accuracy of the resulting data. The smaller the distance between two points P_x and P_{x+1} is chosen, the more accurate the result. Consequently, if the distance is increased the correlation of the result data is increased and thus the inaccuracy is increased too.

Because of the high inaccuracy of discretisation during the synchronisation of time series the log-data analysis in the present scenario produces suboptimal results.

References

[FLS09] Fahrmeir L., Kneib T.; S., Lang: Regression: Modelle, Methoden und Anwendungen (Statistik und ihre Anwendungen). Springer Verlag Auflage: 2, 2009.

[Ha94] Hamilton, James D.: Time Series Analysis. Princeton Univers. Press, 1994.

[KK06] Kotsiantis, S.; Kanellopoulos, D.: Discretization Techniques: A recent survey. In (GESTS, ed.): GESTS International Transactions on Computer Science and Engineering Vol 32 (1). Department of Mathematics, University of Patras, Greece, pp. 47–58, 2006.

[Po81] Powell, M. J. D.: Approximation Theory and Methods. Cambridge University Press, 1981.