Analyzing Periodically Occurring Patterns in Time Series

Shivam Sahai, Maitreya Natu, Vaishali Sadaphal

Tata Research Development and Design Centre,
Pune, India.
{shivam.sahai, maitreya.natu, vaishali.sadaphal}@tcs.com

Abstract

In this paper, we address the problem of identifying periodically occurring patterns in a time series. The domain of data-center management is the primary focus. Data here comprises of request latencies, resource utilization of servers, data center workload etc. to name a few. Although periodicity detection has been researched, the past work does not address the challenges presented by such data-sets. The major challenges include time scaling, time shifting, amplitude scaling, amplitude shifting and noise. We propose an innovative solution to cater to the new challenges. In this paper, we address the problem of identifying the shape of the periodically occurring pattern and the time-series regions which exhibit periodic behavior. We also present a crisp definition of a periodic pattern in the face of such challenges. In addition, we present experimental evaluation of the proposed technique on various data-sets to evaluate its robustness.

1 Introduction

There is a need for large-scale data-analysis in various domains such as data-center management, weather forecasting, bio-informatics, among many others. An important component of this analysis is the analysis of periodic behavior in such data-sets. In this paper, we focus on the domain of performance and capacity management in data-centers. Data here consists of monitored request latencies, workloads, resource utilization etc. to name a few. Analyzing periodic behavior in such data-sets can lead to very useful insights. Some examples are as follows:

Signature identification: Many events in data centers such as garbage collection, disk backups, etc. show periodic behavior. Such events easily get highlighted in the behavior of workloads, disk writes, available memory etc. to name a few. Periodicity analysis can provide signatures of these events.

Forecasting and Prediction: Periodicity analysis of various performance measures such as workload and latencies can be used in forecasting the likely system workload and latencies in future.

Objective: In this paper, we address the problem of analyzing the periodic behavior in a time-series to understand its properties. Informally the problem can be defined as follows; Given a time-series, (a) identify the shapes of periodically occurring patterns (b) identify the regions of occurrence of these shapes.

Challenges: Various challenges make the above defined problem difficult to solve. These challenges are mainly related to non-identical occurrences of the periodic patterns. Below we list some of the major challenges:

Time scaling: The periodically repeating patterns at times exhibit expansion or shrink in the shape. We refer to this behavior as *time scaling*. Figure 1 (a) shows an example of a periodic time-series showing time-scaling.

Time shifting: Time shifting refers to scenarios where the repeating occurrences of a pattern exhibit a lag or lead in time. Figure 1 (b) shows an example of time shifting.

Amplitude scaling: Amplitude scaling refers to the scenario where the periodic pattern demonstrate a jump or a fall in the amplitude. This can be considered as the y-axis equivalent of time-scaling. The shape of the pattern exhibits an overall expansion or shrink at the amplitude scale. Figure 1 (c) shows an example of amplitude scaling.

Amplitude shifting: Amplitude shifting refers to a scenario where the periodic pattern shows a trend in the amplitude of the subsequently repeating patterns. Figure 1 (d) shows an example of amplitude shift.

Presence of noise: Like many other problems in the domain of time-series analysis, noise presents a challenge in periodicity analysis.

Contributions: The main contributions of this paper are as follows: The periodically occurring patterns tend to demonstrate various properties even in the presence of scaling and shifting in time and amplitude axes. For instance, a periodically occurring pattern can be identified through some pivot points that are present even in the presence of time and amplitude variations. Furthermore, each manifestation of the periodic pattern has high similarity with other occurrences of the same pattern. We present these observations in Section 3. We exploit these observations together with various time-series analysis and pattern-matching techniques such as dynamic time warping, clus-

International Conference on Management of Data
COMAD 2010, Nagpur, India, December 8–10, 2010

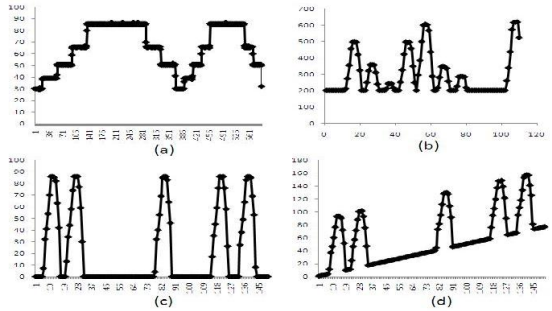

Figure 1: Example of patterns with (a) Time scaling, (b) Amplitude scaling, (c) Time shifting, (d) Amplitude shifting.

tering, etc. in Section 4 to identify the shape and regions of occurrence of the periodically occurring pattern.
1.*Shape estimation:* We present a solution to detect the shapes of periodically occurring patterns. The proposed solution caters to new challenges like scaling and shifting at both amplitude and time axes.
2.*Region determination:* We present a solution to detect regions of occurrences of periodic patterns. The proposed solution caters to new challenges like scaling and shifting at both amplitude and time axes. 3.*Application:* We demonstrate the application of the proposed solution in the domain of performance and capacity management in data-centers. We show how shapes and regions of periodic patterns can be used to derive time-series signature which can be used in a variety of ways.

2 Related work

In the past, a lot of work has been in analyzing the periodic behavior in a time-series. However, most of the work primarily estimates the length of the periodic cycle. Such attempts could be found in [10], [4]. Work done by [1] investigates the utility of the Lomb-Scargle periodogram for the analysis of biological rhythms which also show periodic behavior. In addition, [8] attempts to determine the period value in non-stationary time series by tracking the candidate periods using a Kalman filter.

Another related area of research in this context has been the area of similarity search between two sequences. Euclidian distance can be considered as the simplest similarity measure. More complex techniques include [11] which employ dynamic time-warping methodology. [12] employs a technique for sub-sequence matching to search for a pattern in a large sequence.

It is important to note that these techniques establish their utility owing to the fact that periodicity analysis requires a similarity measure that can compare time series regions.

Some work has been done in detecting shapes and occurrences of periodically occurring patterns [5], [3]. However, they do not address most of the challenges mentioned in the previous section. Current literature, in general, lacks a comprehensive solution to analyze periodic behavior in presence of these new challenges.

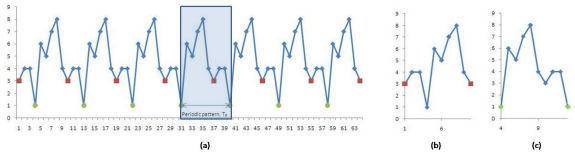

Figure 2: Definition of periodic pattern, T_p.

In this paper, we contribute such a comprehensive solution that analyzes periodic behavior while addressing the new challenges.

3 Design rationale

3.1 Definitions

We first define the various terms that we use in this paper.

- Time-series: A uni-variate time-series T of length N is defined as a finite sequence of N data-points:

  $$T = (v_1, \ldots, v_N).$$

  For the sake of clarity, the data-points are assumed to be sampled at uniform time intervals and have no missing values. We refer to the time and value of a data-point v_i as $Time(v_i)$ and $Value(v_i)$ respectively.

- Time-series region: A time-series region T_p of length p is a subsequence of p contiguous points in the time-series.

3.2 Properties of periodically occurring patterns

We next present various observations that we use to capture the periodic behavior of a time-series. We observe that in a time-series showing periodic behavior, the periodically repeating pattern demonstrates various properties. These properties can be classified as *local* and *global* properties. The local properties are limited to the specific region of occurrence, while the scope of global properties is over the entire time-series.

3.2.1 Property 1 - The periodic pattern can be defined to be bound by a pair of data-points that have minimum value

Consider an ideal scenario of absence of noise, time variations, and amplitude variations. For example, consider the pattern shown in Figure 2(a). Since the pattern is periodic, the start and end points that bound the pattern could be anywhere within a distance of period, p. For example, the pattern can be defined to be bound by a pair of points *(1-10)*, *(10-19)*, *(19-28)* and so on. These points are identified by squares and the resulting pattern is shown in Figure 2(b). The pattern could also be defined to be bound by other pair of points, say, *(2-11)*, *(11-20)*, *(20-29)* and so on. To remove this variability, we define the bounds of the pattern by a pair of data-points, say v_i and v_j that have minimum value.

A region $T_p = (v_i, \ldots, v_j)$ is periodic, if:

1. $Length(T_p) = Time(v_j) - Time(v_i) = p$
2. $Value(v_i) = Value(v_j) = min(Value(v_i), \ldots, Value(v_j))$, and
3. $Value(v_k) > Value(v_i), \forall k \in (i+1, \ldots, j-1)$

The points meeting the above property are shown by circles at points *(4-13), (13-22), (22-31)* and so on in Figure 2(a). The resulting pattern is shown in Figure 2(c). With this observation, a periodic pattern is always bound by a pair of minimum data-points. All other data-points in T_p are hence, assumed to have a greater value than both $Value(v_i)$ and $Value(v_j)$.

It is important to note that this property is limited to an ideal case scenario. However, it provides an intuitive idea on the end-point constraints that we apply on any periodic region. In order to accommodate variations such as amplitude scaling, etc., we now relax these constraints to apply to non-ideal scenarios.

3.2.2 Property 2 - In the presence of time scaling, the constraint on the length of the periodic pattern needs to be relaxed

Time-scaling results in stretching or compression of the pattern. For example, see Figure 1(a). The figure shows two patterns. Both the patterns have same shape but have different lengths owing to the time-scale factor. In such cases, the length property needs to be relaxed as follows.

Given a time series region $T_p = (v_i, \ldots, v_j)$, $Length(T_p) = Time(v_j) - Time(v_i) = p \pm \delta$

where p is the period of the pattern. The variable δ provides the scope of the supported compression and stretch in T_p. This aspect, as we will see later, will control the algorithm's sensitivity to time scaling.

3.2.3 Property 3 - In the presence of amplitude shift, both the end-points of the pattern may not be minimum of all the data points in the pattern

An example of patterns with amplitude shift has been shown in Figure 1(d). In the presence of amplitude shift, the bounding end-points may not be smaller than all other data-points in the region as expected in *Property 1*. Hence, the conditions 2 and 3 of *Property 1* need to be modified as follows:

A region $T_p = (v_i, \ldots, v_j)$ is periodic, if:

1. $Length(T_p) = Time(v_j) - Time(v_i) = p \pm \delta$, and
2. $\forall v_k \in T_p, where i < k < j$, following conditions must not hold true together:
   - v_k is a local minima, and
   - $Value(v_k) < max(Value(v_i), Value(v_j))$

The observation states that there does not exist a local minima, in between the two end-points v_i and v_j, that has a value lesser than either of the two end-points. Thus, even in the presence of amplitude shifts, the end-points of the periodic region are the two smallest local minima and the condition 2 and 3 in *Property 1* holds true.

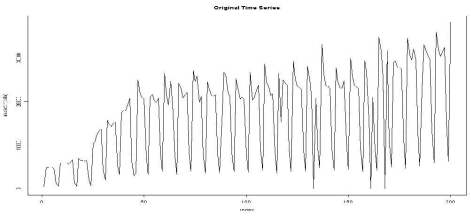

Figure 3: Time series used as running example.

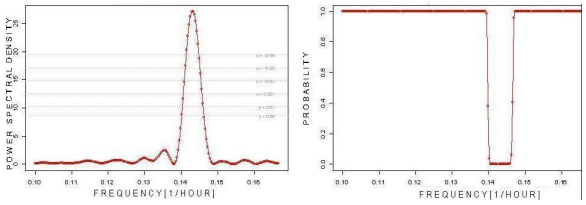

Figure 4: Periodogram for time-series shown in Figure

Although we attempt to cater to different types of amplitude and time variations in a pseudo-periodic time series, this property crisply defines the limits of these variations in the context of a periodic region.

Along with above properties that define a specific region of occurrence, we next present following global properties that hold across multiple occurrences of the periodically occurring pattern.

3.2.4 Property 4 - The pattern should repeat multiple times and the repeating patterns should have a similarity in shape

A time series T should consist of a set of multiple patterns $S_{tp} = \{T_{p1}, \ldots, T_{pm}\}$ such that

- $\forall T_{pi} \in S_{tp}$, the above mentioned local properties hold.
- All $T_{pi} \in S_{tp}$ should be similar in shape. We later systematically define a measure in Section 4 to compute similarity of shape.
- The set S_{tp} must contain at least k number of regions.

4 Proposed solution

In this section, we present a step by step description of the proposed algorithm. The example time-series used for this purpose is shown in Figure 3 which represents the daily workload pattern observed over a period of few months.

1. *Estimate the length of the periodic cycle, p:*

A lot of work has been done in the past [10] in this regard. However, due to the pseudo-periodic nature of time-series, the period value derived by these techniques is only approximate. We propose to use one of the standard techniques, viz. *Periodogram Analysis* [10] to estimate this value.

The left plot in Figure 4 shows the Periodogram for time-series shown in Figure 3. The peak in this case gives the period value of 7 data-points. The right plot in the same

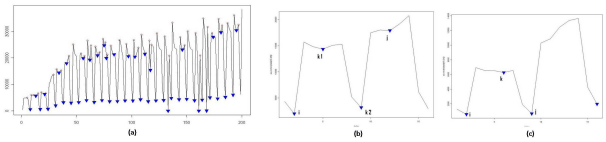

Figure 5: (a) Points of local-minima (solid triangles) for time-series shown in Figure. (b) A region (t_{30}, t_{45}) from the example time-series shown in Figure 3. (c) A region (t_{16}, t_{31}) from the example time-series shown in Figure 3.

figure, refers to the p-value, which in this case happens to be almost zero, thereby exhibiting high confidence.

2. *Smoothen the time-series, T_s:*
Noise is an integral part of any real-world time-series. In this regard, we estimate the LOESS[1] curve of the original series to decipher the hidden behavior in the series. For a large $p/length(T_p)$ ratio, the smoothing is more aggressively done and vice-versa.

In the example time-series, however, no smoothing is carried out owing to a very low $p/length(T_p)$ ratio.

3. *Identify the locations of local minima in T_s:*
This step identifies the locations of local minima in T_s. Local minima are identified in a moving window of r datapoints. This step partially fulfills the objective of identifying the end-points of the periodic patterns. The upcoming steps filter the local minima identified in this step on the basis of various properties discussed in the previous section.

In this regard, we construct a set LM to store these points.

$$LM = \{m_1, \ldots, m_n\}$$

The points of local minima for the example time-series are shown in Figure 5(a).

4. *Pair local minima based on Property 2:*
In this step we identify the pairs (m_i, m_j) in LM, such that m_i and m_j are $p \pm \delta$ time-units apart, where $Time(m_i) < Time(m_j)$. The value p is the length of the periodic cycle computed in Step 1. The variable δ is derived from Property 2 that addresses time-scaling in a time-series. Thus,

> $\forall (m_i, m_j) \in LM$ such that $Time(m_i) < Time(m_j)$: pair $(m_i, m_j) \in LMP$ if, $(p-\delta) \leq (Time(m_j) - Time(m_i) \leq (p+\delta)$

This ensures that $\forall (m_i, m_j) \in LMP$, the pattern T_p will be bound by minima points m_i and m_j, partially in line with the inferences presented in Property 1.

5. *Retain pairs in LMP obeying Property 3:*
In accordance to the constraints mentioned in Property 3, in this step, we remove any pair $(m_i, m_j) \in LMP$, for which there is a local minima $m_k \in LM$ such that,

1. m_k lies between m_i and m_j, i.e., $Time(m_i) < Time(m_k) < Time(m_j)$, and
2. the value of m_k is smaller than the value of either m_i or m_j, i.e., $Value(m_k) < (Value(m_i)\ Or\ Value(m_j)$

[1]LOESS (locally weighted regression) is a fitting technique or a function for which the value at a particular location t_i is determined only by the points in its vicinity.

This step, hence, will remove any such pattern that violates the limits applied on the pseudo-periodic definition.

Figure 5(b) shows a region (t_{30}, t_{45}) from the time-series of the running example shown in Figure 3. In accordance to the discussed property, the region (t_i, t_j) will not be a potential pattern bearing region. The reason being the presence of the local minima's t_{k1} and t_{k2}, both of them satisfying the constraints mentioned in *Property 3*. However, the region (t_i, t_{k2}) will be considered as a potential pattern bearing region. Due to time-scale factors discussed in *Property 1*, the region (t_i, k_1) is also discarded since its length does not fall in $p \pm \delta$ range.

6. *Apply Property 4 to set LMP to retain pairs with similar shape:*
As explained in *Property 4* of Section 3, the periodic pattern must demonstrate a similarity in shape with other patterns of the same series. In this regard, we compute a measure to estimate the similarity among potential patterns in LMP. For this purpose, *1-D Euclidean distance* is computed as follows;

1. for any pair (m_i, m_j) in LMP;
   - Compute 1-D *Euclidean distance* between $max(Value(m_i), Value(m_j))$ and all points $Value(p_k) \forall i < k < j$
   - Add this distance to a set E.
   - Normalize all distances in the set E.
2. Discard any pair (m_i, m_j) in LMP if $E_{ij} \leq \theta$ where $0 \leq \theta \leq 1$ is the *Euclidean threshold* for similarity. This threshold controls how much distortion in the shape of the periodic pattern is accepted by the algorithm.

7. *Address overlapping pairs in LMP:* The filters applied until now drastically reduce the number of pairs that represent potential pattern bearing regions. However, at this stage there is a possibility of overlapping pairs defined on the same region. More formally, a given minima m_i, might pair with many minima, say (m_i, m_j) and (m_i, m_k), that satisfy all the filters mentioned until now. We address such cases by retaining the minima pair that covers the largest region in time and has smallest difference in value. Thus, identify the set S of minima pairs in LMP that have common begin or end minima. Of all the minima pairs $(m_i, m_j) \in S$, retain the minima pair (m_i, m_j) in LMP that has the following property:

- $(Value(m_i) - Value(m_j)) \leq min(\forall_{(m_x, m_y) \in S}(Value(m_x) - Value(m_y)))$
- $(Time(m_i) - Time(m_j)) \leq max(\forall_{(m_x, m_y) \in S}(Time(m_x) - Time(m_y)))$

A scenario of overlapping regions is illustrated in Figure 5(c). The minima m_i forms two pairs viz. (m_i, m_k) and (m_i, m_j). This property chose (m_i, m_j) as a potential pattern bearing regions owing to constraints defined in this property.

8. *Filter out pairs based on Amplitude Shifts discussed in Property 2:*
We now attempt to find a subset of pairs of the form (m_i, m_j) from LMP, which can be clustered based on the displacements between $Value(m_i)$ and $Value(m_j)$. This

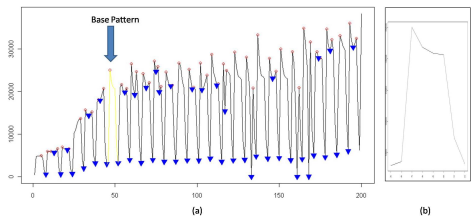

Figure 6: The base pattern that best represents the shape of the periodically occurring patterns in the time series.

filters out any pair from LMP having a relatively unusual displacement between the minima. The retained pairs, in the next step, undergo a similarity check based on the technique called *Dynamic Time Warping (DTW)* [11].

It is important to note here that, this step rather performs an aggressive filtering on the minima pairs. Owing to the computationally expensive nature of DTW, this step ensures that only a minimal set of pairs are processed, all of which are also potentially very similar owing to the collection of filters applied until now.

9. *Determine the base pattern:*
This step attempts to determine the base pattern among the pairs retained in LMP. The base pattern, as already discussed, is derived by estimating similarity among the potential regions bounded by pairs in LMP. The technique *DTW* [11, 2, 6, 9] is employed for this purpose.

Any pair $(m_i, m_j) \in LMP$ which represents the region say T_p[i,j], contains the base pattern, if;

- $\Sigma Distance(DTW(T_p[i,j], T_p[x,y]))\forall(m_x, m_y) \in LMP$ is minimum

We will denote the base pattern identified in this step as $T_p[base]$. Figure 6 shows the base pattern derived for the running example time-series shown in Figure 3.

10. *Identify regions exhibiting periodic behavior:*
Once the base pattern is determined, the algorithm now attempts to determine if a similar pattern exists in other potential locations. The set of locations to search for in this step, is more than the set of locations currently retained in LMP. As mentioned earlier, in order to determine the base pattern, the set LMP undergoes an aggressive filter at Step 8. However, since the base pattern is already identified, the search must be carried out on more potential locations. This is necessary to avoid missing out on any pattern bearing location which might have been filtered at Step 8.

At this stage, the search for periodic behavior is carried on all the pairs that went into Step 8. Let this set of pairs be represented by LMP_{s8}. As a similarity measure at this step, we re-employ DTW as follows:

Any pair $(m_i, m_j) \in LMP_{s8}$ which represents a region say T_p[i,j], is similar to $T_p[base]$, if;

- $distance(DTW(T_p[i,j], T_p[base]))\forall(m_x, m_y) \in LMP \leq \Delta$, where Δ is the similarity threshold.

Figure 7 shows the regions of occurrences of the periodic regions found by the algorithm.

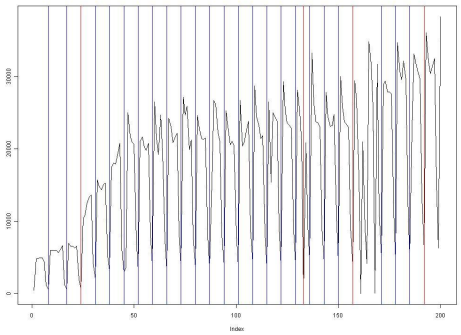

Figure 7: Regions of occurrences of the periodic time series

.

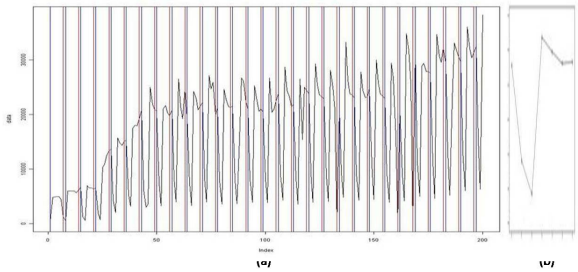

Figure 8: (a) Regions of occurrences of the periodic time series using algorithm in [5], (b) Base pattern identified by [5]

.

5 Experimental evaluation

We next present the experimental evaluation of the proposed algorithm. We present the sensitivity analysis of the proposed algorithm on synthetic data where we evaluate the correctness of the algorithm while varying different data properties such as noise, level of time variations in the periodic patterns, level of amplitude variations in the periodic patterns, etc.

5.0.5 Experiment setup

We use a discrete-event simulator CSIM [7] to generate periodic data. We then systematically insert different time and amplitude variations in this pattern. The objective of these experiments is to test the sensitivity of the proposed algorithm to the time and amplitude variations and to identify its effective region of operation. We hence take each parameter and generate data with increasing order of variation. We quantify each of these variations, amplitude scale, amplitude shift, time scale, time shift on a scale of 1 to 10 such that 1 refers to least variation and 10 refers to very high variation.

5.0.6 Evaluation criteria

In the above simulation setup the actual shape of the pattern is known a-priori. We refer to the actual pattern as P_{Act}.

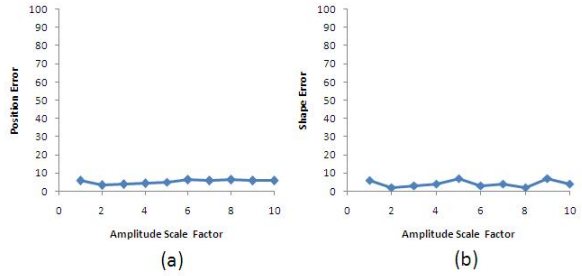

Figure 9: Effect of Amplitude Scale on Position Error or Shape Error (No. of patterns = 10, Period = 10).

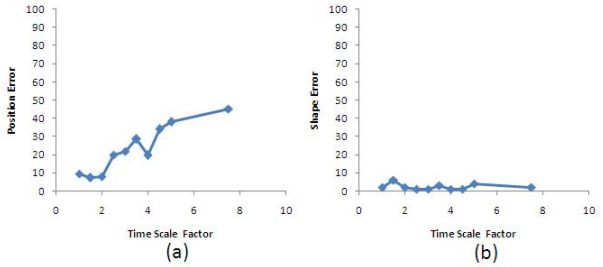

Figure 10: Effect of Time Scale on Position Error or Shape Error (No. of patterns = 10, Period = 50).

Let the estimated pattern be P_{Est}. We compare the actual pattern with the pattern estimated by the algorithm over two metrics viz. the shape error and the position error.

Shape error: We calculate this error by calculating difference in the actual shape P_{Act} and the shape estimated by the algorithm P_{Est}. We compute the difference in shape using dynamic time warping (DTW) and using the warp distance as an error metric. We use normalized shapes of P_{Act} and P_{Est} to take care of amplitude variations. Thus,

$$ShapeError = DTW(P_{Act}, P_{Est}).nonumber \quad (1)$$

Position error: We define position error between two patterns as the absolute difference between the start time of the actual and estimated pattern. For each pattern in the actual P_{Act} in the actual time-series, we find an estimated pattern P_{Est} derived by our algorithm such that difference in the begin time of P_{Act} and P_{Est} is minimal. We compute the difference in start time of P_{Act} and P_{Est} and refer to this distance as the position error in detecting P_{Act}. We average this over all instances of actual patterns in the original time series.

$$PositionError = Mean_{\forall P_{Act}}(|BeginTime(P_{Act}) - BeginTime(P_{Est})|) \quad (2)$$

In the following sections, we present effect of change in various factors viz. amplitude, time scale and amplitude, time shift on the performance of proposed algorithm.

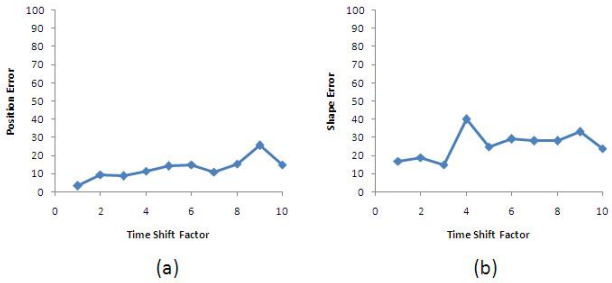

Figure 11: Effect of Time Shift on Position Error or Shape Error (No. of patterns = 10, Period = 50).

5.0.7 Addressing amplitude scaling

Figure 9(a), (b) show the effect of change in amplitude on position error and shape error respectively. There are 10 patterns of period 10 in the time series. The amplitude factor changes from 1 to 10. It can be observed from Figure 9(a) that the position error is almost nil even if the amplitude scale factor increases to 10. For any amplitude scale factor, the position error is less than 1 time unit. The position error increases slightly when the amplitude scale factor increases. From Figure 9(b) it can be observed that the *ShapeError* is between 0-10% for amplitude scale factor of 1 to 10. Amplitude scale does not affect detecting shape of the pattern. The amplitude variations are taken care of by a proper definition of the pattern as in Section 3.2.1.

5.0.8 Addressing time scaling

Figure 10(a), (b) show the effect of change in time scale factor on position error and shape error respectively. The position error increases significantly as the time scale factor increases from 1 to 10. For instance, if a time scale factor is 10, the period of the pattern may be anywhere between 1 to 10 times of the period of the original pattern. The position error increases with the increase in time scale since the patterns scale in time and this results in patterns occurring non-periodically at unexpected time intervals. From Figure 10(b), it can be observed that the change in time scale factor does not affect the *ShapeError*. The error in shape is based on dynamic time warping. As a result, even if the pattern in time-scaled, DTW is able to identify the base pattern in it. This results in a small *ShapeError*.

5.0.9 Addressing time shifting

Figure 11(a), (b) show the effect of change in time shift factor on the position error and shape error respectively. The time shift factor increases from 1 to 10. The effect of time shift is small on *PositionError*. The algorithm is fairly robust against different factors of time shifts and can detect patterns occurring at different locations. The *PositionError* does not go beyond 10%. The maximum *ShapeError* is around 30% for time shift factors 1-10. Both *PositionError* and *ShapeError* increase with increasing time shift factor.

5.0.10 Addressing amplitude shifting

Figure 12(a), (b) show the effect of change in amplitude shift factor on the position error and shape error, respec-

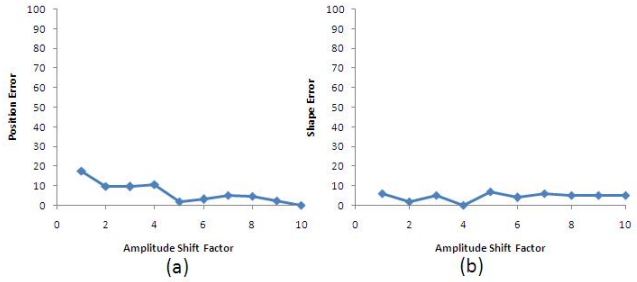

Figure 12: Effect of Amplitude Shift on Position Error or Shape Error.

tively. The amplitude shift factor increases from 1 to 10. The effect of amplitude shift on *PositionError* is small. *PositionError* reduces slightly as the amplitude shift factor increases. This is because as the amplitude increases, the effect of noise reduces and the algorithm is able to detect the patterns more accurately. The effect of change in amplitude shift factor on shape error is also not large. While *ShapeError* slightly increases with increasing amplitude factor, *PositionError* does not go beyond 20% for amplitude shift factors between 1 to 10. *ShapeError* is less than 10% for amplitude shift factor between 1 to 10.

6 Comparison with existing work

In this section, we compare the results of the proposed algorithm with the results of algorithm proposed in [5] to identify periodically occurring pattern and its region of occurrences. The algorithm in [5] proposes to derive the period value, p, using periodogram. It then computes the base pattern as the average of all the patterns occurring in the time series at a distance of period, p. Figure 8(a) shows the pattern identified by this algorithm and Figure 8(b) shows its regions of occurrences. Note that the region of occurrences identified by the algorithm proposed in [5] are skewed from actual location of the pattern. Further, the average base pattern identified by this algorithm (shown in Figure 8 (b)) is not representative of the actual pattern that is repeating.

7 Application of periodicity analysis in data centers

In this section, we present various scenarios where the proposed approach of analysis of periodic patterns provides many interesting insights into the functioning of the data centers.

Today's data centers are monitored to keep track of the overall system health. We demonstrate various scenarios observed in real-life case-studies where capturing the periodic behavior of these metrics provided very useful insights for the data center operators.

7.1 Detection of signature of an event

Various events in a data center demonstrate a periodically occurring signature. These events could be specific operations such as periodic data backup jobs or periodic operating system process such as disk cleanup.

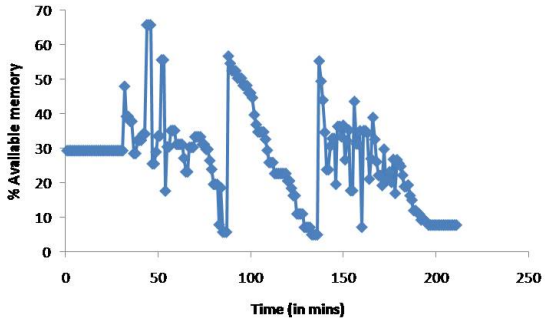

Figure 13: Time series of available memory showing signature of periodic garbage collection.

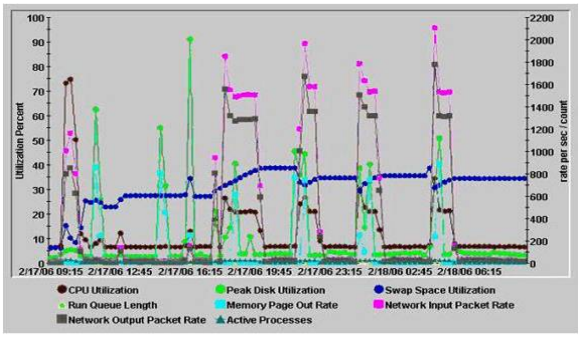

Figure 14: Time series of various system metrics observed during events of system restart.

We demonstrate the concept using an example of the event of *garbage collection*. The time-series of available memory of the system showed a particular behavior between subsequent garbage collection operations. Figure 13 presents the time-series of the available system memory. We observed that the JVM settings affect the length of this pattern. Increasing the active memory size of the JVM from 256MB to 512MB results in expansion of the pattern. An automatic detection of such patterns can provide direct insights into appropriate JVM settings for the observed workload.

7.2 Root-cause analysis

Due to the use of data centers for more and more performance-critical applications, it has become very important to have automated techniques to detect failures in the data center and to quickly identify the root-cause of the observed failure.

We observed a scenario in a data-center where a particular process was observing periodic restarts and the objective was to find the root-cause behind these restarts. Figure 14 shows time-series of CPU utilization, swap space utilization, network input and output packet rate, etc. of the server on which the process was running. A periodic pattern was identified in swap space, CPU utilization, and

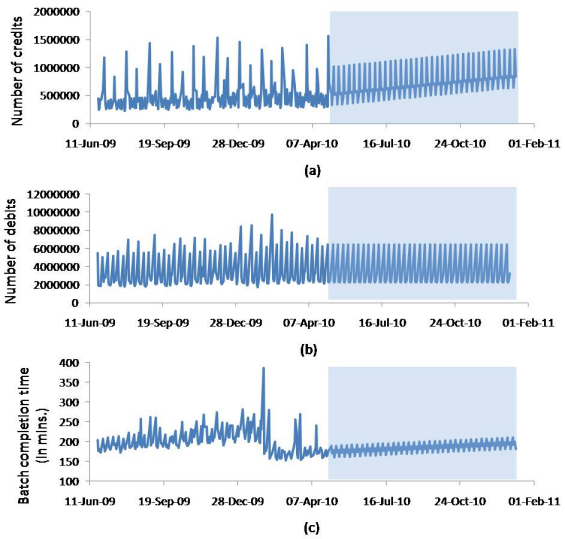

Figure 15: (a, b) Time series of workload (number of credits and debits) and their forecast, (c) Time series of the batch completion time and its prediction.

network input and output rate. These patterns were co-occurring with the events of process restart. It was deduced that as the swap space reached a certain limit, the process restarted making the CPU utilization and network input and output rate to drop.

7.3 Forecasting

We present a scenario of a batch processing system that is monitored to collect workload information, batch completion time, etc.

Figure 15 shows a scenario where we predict the workload and batch-completion time of a batch-processing system for the next 6 months. We forecast the number of credits and number of debits for the next 6 months as shown in Figure 15a and Figure 15b. We then build a correlation model to predict the batch completion time based on the forecasted number of credits and debits. We show the predicted batch completion time in Figure 15c. It can be seen that over a period of 6 months the batch completion time is predicted to increase from 180 mins to 200 mins.

8 Conclusion

In this paper, we addressed the problem identifying periodically occurring patterns in a time series. In this regard, we presented a technique that is robust against various challenges like time scaling, time shifting, amplitude scaling, amplitude shifting and noise.

Although periodicity detection has been an area of research, the past work did not address the challenges mentioned in this paper. We presented a crisp definition of a periodic pattern in the presence of time scales and shifts and amplitude scales and shifts. We then presented an innovative solution to analyze such data for periodic behavior.

We evaluated the accuracy of the proposed algorithm on data collected from a production data-center. We also performed sensitivity analysis of the proposed algorithm on simulation data generated in an controlled environment. Further, we demonstrated the application of the proposed solution in the domain of performance and capacity management in data-centers.

References

[1] M. Ahdesmki, H. Lhdesmki, R. Pearson, H. Huttunen, and O. Yli-Harja. Robust detection of periodic time series measured from biological systems. In *BMC Bioinformatics*, 2005.

[2] A. W. chee Fu, E. Keogh, L. Y. H. Lau, and C. A. Ratanamahatana. Scaling and time warping in time series querying. In *VLDB*, 2005.

[3] J. H. Fan, F. Zhang, and P. Shahabuddin. Characterizing normal operation of a web server: Application to workload forecasting and problem detection. In *Computer Measurement Group*, 1998.

[4] E. F. Glynn, J. Chen, , and A. R. Mushegian. Detecting periodic patterns in unevenly spaced gene expression time series using lombscargle periodograms. In *Bioinformatics, Volume 22, Number 3 Pp. 310-316*, 2005.

[5] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper. Workload analysis and demand prediction of enterprise data center applications. In *IEEE 10th International Symposium on Workload Characterization*, 2007.

[6] L. Guiling, W. Yuanzhen, L. Min, and W. Zongda. Similarity match in time series streams under dynamic time warping distance. In *CSSE*, 2008.

[7] http://www.mesquite.com/. Csim 20 development toolkit for simulation and modeling.

[8] S. Parthasarathy, S. Mehta, and S. Srinivasan. Robust periodicity detection algorithms. In *CIKM*, 2006.

[9] J. Paulus and A. Klapuri. Measuring the similarity of rhythmic patterns. In *IRCAM*, 2002.

[10] T. Ruf. The lomb-scargle periodogram in biological rhythm research: Analysis of incomplete and unequally spaced time-series. In *Biological Rhythm Research, Volume 30, Issue 2 April 1999 , pages 178 - 201*, 1999.

[11] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE TRANSACTIONS ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING*, ASSP-26(1), 1978.

[12] H. Wu, B. Salzberg, G. C. Sharp, S. B. Jiang, H. Shirato, and D. Kaeli. Subsequence matching on structured time series data. In *SIGMOD*, 2005.