https://doi.org/10.3311/PPee.17671

Periodica Polytechnica Electrical Engineering and Computer Science, 66(1), pp. 1–11, 2022

Extreme Value Analysis for Time-variable Mixed Workload

Szilárd Bozóki[1*], András Pataricza[1]

[1] Department of Measurement and Information Systems, Faculty of Electrical Engineering and Informatics, Budapest University of Technology and Economics, H-1117 Budapest, Magyar tudósok krt. 2., Hungary
[*] Corresponding author, e-mail: bozoki@mit.bme.hu

Received: 10 December 2020, Accepted: 12 July 2021, Published online: 09 December 2021

Abstract

Proper timeliness is vital for a lot of real-world computing systems. Understanding the phenomena of extreme workloads is essential because unhandled, extreme workloads could cause violation of timeliness requirements, service degradation, and even downtime. Extremity can have multiple roots: (1) service requests can naturally produce extreme workloads; (2) bursts could randomly occur on a probabilistic basis in case of a mixed workload in multiservice systems; (3) workload spikes typically happen in deadline bound tasks. Extreme Value Analysis (EVA) is a statistical method for modeling the extremely deviant values corresponding to the largest values. The foundation mathematics of EVA, the Extreme Value Theorem, requires the dataset to be independent and identically distributed. However, this is not generally true in practice because, usually, real-life processes are a mixture of sources with identifiable patterns. For example, seasonality and periodic fluctuations are regularly occurring patterns. Deadlines can be purely periodic, e.g., monthly tax submissions, or time variable, e.g., university homework submission with variable semester time schedules.

We propose to preprocess the data using time series decomposition to separate the stochastic process causing extreme values. Moreover, we focus on the case where the root cause of the extreme values is the same mechanism: a deadline. We exploit known deadlines using dynamic time warp to search for the recurring similar workload peak patterns varying in time and amplitude.

Keywords

Extreme Value Analysis (EVA), capacity design, time series, dynamic time warp

1 Introduction

An extreme event occurs when interrelationships between complex and coinciding circumstances cause considerable deviations from the usual behavior.

However, extreme events (workload spikes and resulting abnormally long service times) are generally present in real-world computer systems where the correlated workload sources sharing computing resources result in occasional workload spikes [1]. For example, a seasonal promotion in a webshop can incur an extreme workload. Moreover, many humans triggered deadline-bound tasks, and as late as possible (ALAP) scheduling may also lead to such workload profiles.

In real-life computing systems, extreme values are usually rare; thus, in this case, rarity and being an outlier coincide. However, for critical applications, even a low rate of timeliness violations is usually unacceptable. Rarity makes statistical modeling (parameter identification, parametrization) difficult due to the moderate size of the data representing the extreme values. Moreover, traditional statistics usually suppress values significantly larger or smaller in amplitude than the majority of the dataset as outliers, leading to ill dimensioned systems.

Overall, we aim to provide a method for real-life computing systems with mixed workloads to meet their temporal and high availability (HA) requirements by adequately dimensioning for extreme workloads.

We focus on cases where extreme workload occurrences have a single root cause: a time-varying recurring deadline. We will use it to identify the extreme component and compensate for its time variance.

2 Modeling workloads with time-varying peaks

In Section 2, we present Extreme Value Analysis (EVA) and some of the issues and solutions of not perfectly *independent and identically distributed* (*iid*) datasets. Such datasets are typical in real life [1]. Examples are the service sector, banking, power grid, and transportation, where there are peak operating hours.

Cite this article as: Bozóki, S., Pataricza, A. "Extreme Value Analysis for Time-variable Mixed Workload", Periodica Polytechnica Electrical Engineering and Computer Science, 66(1), pp. 1–11, 2022. https://doi.org/10.3311/PPee.17671

2.1 Extreme Value Analysis

EVA is a particular area of statistics aimed at modelling extremely deviating, far from average values. The classic use case of EVA is in hydrology:

- *Analysis*: What is the probability of a given embankment surviving the floods of the next time period (for example 100 years)?
- *Dimensioning*: How tall embankment is needed to survive the floods of the next time-period (for example 100 years) of a given probability?

EVA has two main methods:

- the *block-maxima method* (aka Annual Maxima Series, AMS) based on the *Fisher-Tippett-Gnedenko theorem*,
- the *peak-over threshold* (POT) method based on the *Pickands-Balkema-de Haan theorem*.

The *block maxima* method searches the dataset for large representative values by slicing the dataset into equal length blocks, selecting only the maximum value from each block, and discarding all the other values. The block maxima method uses *Gumbel, Frechet, or Weibull* distribution to fit.

The *peak-over-threshold method* selects all the values above a threshold and discards the rest, and uses *Generalized Pareto Distribution* (*GPD*) to fit (Fig. 1).

2.1.1 Extreme Value Analysis background

The EVA mathematics is summarised based on [2–4].

Let $\chi = \langle X_1, X_2, ..., X_n \rangle$ be a sequence of *iid* random variables with *cumulative distribution function* (*cdf*) $F(x)$ and let $M_n = \max(X_1, X_2, \ldots X_n)$ denote their maxima. If $F(x)$ is known and variables X_i are still assumed to be *iid*, then the exact *cdf* of the maxima can be computed as the product of the *cdf*s.

$$P(M_n \le x) = P(X_1 \le x, X_2 \le x, \ldots X_n \le x) = \prod_{i=1}^{n} P(X_i \le x) = [F(x)]^n \quad (1)$$

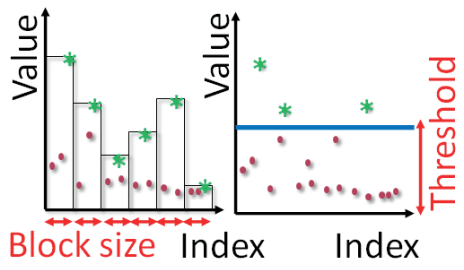

Fig. 1 EVA methods compared. Left: AMS Right: POT

If $F(x)$ is unknown, then a simple solution could be using a direct estimator of $\hat{F}(x)$ out of a sample sequence. However, when computing $[\hat{F}(x)]^n$, even small estimation errors in $\hat{F}(x)$ can cumulatively distort $[\hat{F}(x)]^n$.

Instead of using an estimator $\hat{F}(x)$, EVA approximates directly $[F(x)]^n$. Hence, the core mathematical question is: *How does the "n-th power" of cdfs behave? Are there any requirements? Does it converge? How, where?*

Intuitively, if $F(x)$ has an upper bound and the number of random variables n goes to infinity, then $P(M_n \le x) = [F(x)]^n$ converges to the upper endpoint with a probability of 1, which makes the converging *cdf* degenerate.

A *degenerate cdf* has all the weight concentrated in a single point at its *probability density function* (*pdf*), which renders associated random variables constant (producing the same constant value).

Moreover, a prerequisite of analytical EVA is a *"connection"* between $F(x)$ and $P(M_n \le x) = [F(x)]^n$ regardless of the number of random variables n. Several ideas revolve around the issue of *"connection"*.

A stochastic process is *ergodic* if a statistical property can be deduced from a single, sufficiently long, random sample of the process. For EVA, the stochastic process needs to be *tail-distribution-type-ergodic*, because if $[F(x)]^n$ was a randomly changing distribution type while n goes to infinity, then making an inference based on $[F(x)]^n$ would be random and impractical.

A *distribution is stable* if a *linear combination* of independent random variables copies has the same distribution up to location and scale (the result distribution is of the same type). If X is a non-degenerate random variable, and there exists $a_n > 0, b_n \in \mathbb{R}$ constant series such that $X_1 + X_2 + \ldots + X_n$ has the same distribution as $a_n X + b_n$ for all $n > 1$, then X and its *cdf* are stable.

The *location* parameter shifts the *pdf* along the X axis. The *scale* parameter controls how spread out the pdf is along the X axis. Larger scale means the pdf is more spread out along the X axis. As the integral of pdf over the entire space is equal to 1, a larger X axis spread means a smaller Y axis spread on average.

A *distribution is max-stable* if the *maximum value* of independent random variables copies has the same distribution up to location and scale (the result distribution is of the same type). If X is a non-degenerate random variable, and there exists $a_n > 0, b_n \in \mathbb{R}$ constant series such that $M_n = \max(X_1, X_2, \ldots, X_n)$ has the same distribution as $a_n X + b_n$ for all $n > 1$, then X and its *cdf* are max-stable. As EVA is concerned with taking the maximum value of

random variables, max-stability is a prerequisite property of EVA for the *cdf* under investigation.

Also, owing to the *iid* requirement, the stochastic process must be *stationary*: its unconditional joint *cdf* must not change when shifted in time.

Normalization can make $[F(x)]^n$ converge to a non-degenerate distribution function if possible:

$$M_n^{\text{normalised}} = \frac{M_n - a_n}{b_n}, \tag{2}$$

$$a_{1,2,\ldots,n}, b_{1,2,\ldots,n} > 0 \in \mathbb{R} \text{ approriately selected constants.} \tag{3}$$

$F(x)$ belongs to the *Maximum Domain of Attraction* (*MDA*) of $H(x)$ only if normalizing constant sequences of $a_n > 0, b_n \in \mathbb{R}$ exist such that $[F(x)]^n$ converges to $H(x)$:

$$\lim_{n\to\infty} P\left(\frac{M_n - b_n}{a_n} \le x\right) = \lim_{n\to\infty} F^n\left(a_n x + b_n\right) = H\left(x\right), x \in \mathbb{R}, \tag{4}$$

where $H(x)$ is a non-degenerate *cdf* and the normalized maxima $M_n^{\text{normalised}}$ converges in distribution to a random variable with distribution function *H*.

The *maximum domain of attraction is unique* concerning location and scale: If $K(X) \in$ MDA $[L(X)]$ and $K(X) \in$ MDA $[J(X)]$ then $L(X)$ and $J(X)$ are necessarily from the same type of distributions, meaning there exists $a > 0, b \in \mathbb{R}$ that $L(X) = J(aX + b)$.

2.1.2 Annual Maxima Series

The *Fisher–Tippett–Gnedenko theorem* states that if the probability distribution function of the maximum value with the increasing number of observations converges to a non-degenerate distribution function after normalizing, then it always belongs to one of the three Extreme Value Distribution (EVD) classes of *Gumbel*, *Fréchet*, and *Weibull* (Table 1). If normalization is impossible and $[F(x)]^n$ diverges, then that means that the systematic extremity of the random variables is beyond the modeling power of EVA. *The class of the extreme value distributions and the max-stable distributions coincide*.

Table 1 Extreme Value Distribution classes

Name	CDF Formula	Params	distribution of
Weibull	$\Psi_\alpha(x) = \exp\left(-(-x)^\alpha\right)$ $\Psi_\alpha(x) = 1$	$x \le 0, \alpha > 0$ $x > 0, \alpha > 0$	short tail with finite upper bound
Gumbel	$\Lambda(x) = \exp\left(-\exp(-x)\right)$	$x \in \mathbb{R}$	light tail (exponential tail)
Fréchet	$\Phi_\alpha(x) = 0$ $\Phi_\alpha(x) = \exp\left(-x^{-\alpha}\right)$	$x \le 0, \alpha > 0$ $x > 0, \alpha > 0$	heavy tail (incl. polynomial decay)

The standard EVDs differ by the convergence rate (speed) of their respective tail distributions. The MDA binds the class members: the EVD class members can be different while retaining the same asymptotic tail behavior.

The reference rate of convergence is the exponential-tailed *Gumbel* distribution family. Example class members are: normal, gamma, log-normal, exponential.

The slowest is the *Fréchet* family; these distributions have heavy-fat tails, where the complementary cumulative distribution function decreases as a power function. Example class members are Pareto, Student, Cauchy, Burr.

The *Weibull* distribution family has the fastest tail convergence with a finite right endpoint (thin tail). Example class members are: uniform, beta and reverse Burr.

2.1.3 Threshold exceedance

The threshold exceedance model separates the "normal" and extreme values by their respective amplitude with a threshold *u*. Selecting a proper threshold is cumbersome. However, several alternative methods are helping the threshold selection [5, 6].

The *Pickands–Balkema–de Haan theorem* states that for the approximation of the conditional distribution function F_u, with a large enough threshold *u*, the *Generalized Pareto Distribution* (*GPD*) is a right candidate if the unknown *F* is within the MDA of EVDs [2]. The conditional excess distribution F_u over the threshold *u* is:

$$F_u(x) := P\left(X - u \le x \mid X > u\right) = \frac{F(u+x) - F(u)}{1 - F(u)}, \tag{5}$$
$$0 \le x \le x_f - u,$$

where: *F*: unknown distribution function of random variable *X*; x_f: either the finite or infinite right endpoint of the underlying distribution.

As the threshold *u* converges to the right endpoint of the underlying distribution *F*, F_u converges to GPD if the normalized maxima of *F* converges to an EVD, which essentially links the two EVA methods together.

For reference, the GPD has the following CDF:

$$G_{\xi,\beta}(x) = \begin{cases} 1 - \left(1 + \frac{\xi x}{\beta}\right)^{\frac{-1}{\xi}} & \text{if } \xi > 0, \beta > 0, x \ge 0 \\ 1 - \exp\left(\frac{-x}{\beta}\right) & \text{if } \xi = 0, \beta > 0, x \ge 0 \\ 1 - \left(1 + \frac{\xi x}{\beta}\right)^{\frac{-1}{\xi}} & \text{if } \xi < 0, \beta > 0, 0 < x \le -\frac{\beta}{\xi} \end{cases}. \tag{6}$$

Parameter ξ is named "shape" while β is named "scale".

2.2 Local correlation and de-clustering

The locality is a particular case of correlation when a variable correlates with itself over short to medium ranges. For example, if a message is sent to a destination in a network application, it is more likely that follow-up messages will be sent there shortly [1].

When a time series (TS) is created, the real-world data is periodically sampled, for example, daily, monthly, and quarterly data. However, generally speaking, EVA models the amplitude domain (Y-axis) and does not consider the length of a peak (X-axis). Thus, if the duration of the peaks and the time series sampling period are ill aligned, then the longer peaks are at risk of being over-represented with multiple locally correlated values. This local correlation can distort the peak distribution and violate the *iid* assumption, and invalidate EVA results (Fig. 2).

For block maxima, if the data shows a significant autocorrelation with small lags, a standard way of handling this problem is selecting a larger block size, effectively suppressing bursts of locally correlated data.

For POT, a standard way of handling this problem is de-clustering around the following idea: non-extreme values must separate extremes. Extreme values that are not separated by at least N under the threshold values are clustered. Then from each cluster, the largest value is selected while the rest is moved under the threshold. This de-clustering uses that the under the threshold values are not used for fitting GPD. N is an input of the de-clustering. The de-clustering threshold can differ from the POT threshold, ideally, with the former being smaller (Fig. 3).

2.3 Global correlation and Time Series Decomposition

Real-world time series data usually contain regular identifiable patterns, which can distort the analysis. Although both AMS and POT can tolerate some level of correlation, as previously discussed, further investigation is advised for global/long-term correlation.

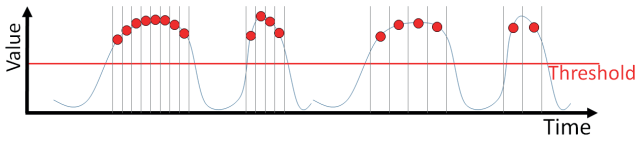

Fig. 2 Sampling issues due to the sampling interval

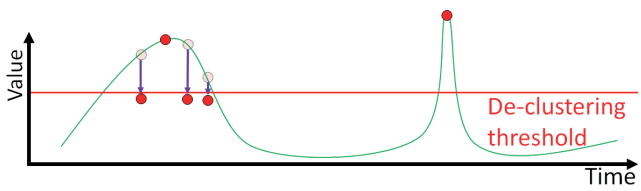

Fig. 3 De-clustering

Traditionally, time series *decomposition* is aimed at modeling the time series by extracting identifiable patterns using statistical methods. Usually, the following high-level components are used for time series analysis:

- *Trend*: long-term increase or decrease [T];
- *Seasonality*: periodicity with a fixed, known period length [S];
- *Cycles*: periodicity without a fixed, known period length [C];
- *Residual / Error Term / Irregular Component / Noise*: the remainder data that is not extracted by the other components [R].

While these components are typical for numerous real-world processes (natural, business, human, etc.), other ones could be used by using other apriori knowledge.

For EVA, the residual component with the random data is relevant because the data in the other components are dependent according to their respective models. Consequently, the more the non-random components model the data, the less correlation remains in the residual.

Time series decomposition has two often used models:

- Additive model: $Y = T + S + C + R$; $R = Y - T - S - C$;
- Multiplicative model: $Y = T * S * C * R$; $R = \frac{Y}{T * S * C}$.

The additive model is preferred when the seasonal fluctuations are independent of the trend level, or the data tends to shows a linear trend. In the opposite case, the multiplicative model is preferred.

The multiplicative decomposition can be deduced to an additive one by using a log transformation:

$$\log(Y) = \log(T) + \log(S) + \log(C) + \log(R). \tag{7}$$

There are several decomposition techniques: classical, X11, SEATS, and STL, just to name a few.

2.3.1 Classical decomposition

Classical decomposition is based on averages. For the trend line, a moving average is calculated with a fixed window length, e.g., seven days moving average. For the seasonal/cyclical component, first, the trend is removed from the data (de-trending), then the average is calculated for each season. For example, the seasonal effect for February is the average of all the February data. There

are several issues with the classical decomposition, which newer methods improve upon:

- The smoothing effect of averaging makes the classical decomposition method unable to capture rapid changes, especially if the moving average window is large, which can lead to a significant underestimation of extreme values.
- The methods assume constant seasonal effects; thus, they cannot correctly capture non-constant seasonality, making the technique inaccurate with real-world dynamically changing systems.
- Vulnerability to a burst of outliers if the burst size is comparable to the averaging window size.

2.3.2 STL

Seasonal Trend decomposition using Loess (STL) uses loess regression for modeling the trend and seasonal components [7]. Loess regression is a nonparametric technique that is a generalization of moving average and polynomial regression: loess uses locally weighted regressions to fit a smooth curve to points. Loess curves can reveal trends and cycles in data that might be difficult to model with a parametric curve, allowing non-linear relationships to be estimated [8]. Consequently, STL is:

- an additive decomposition method;
- able to handle any type of seasonality;
- can capture rapid changes because the smoothness of the trend can be controlled;
- able to model changes in seasonality, where the rate of change can be controlled;
- robust to outliers via parameter tuning.

There are various methods to forecast the remainder component when using STL. We will use Autoregressive integrated moving average (ARIMA) [9].

2.3.3 Facebook Prophet

Facebook Prophet is an open-source time-series analysis tool with an additive model at its core describing user-initiated workloads with main influence factors corresponding to trends (like the growing popularity of a service), typical periodicity, and social factors [10].

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \tag{8}$$

where:

$y(t)$: the predicted (forecasted) load at time t,

$g(t)$: the trend component,

$s(t)$: the seasonality component,

$h(t)$: the holiday component,

ϵ_t: is the error term.

The original aim of Prophet was resource utilization optimization of a vast infrastructure running a non-critical service that serves hundreds of thousands of users around the globe simultaneously.

Prophet offers two different trend models $g(t)$: limited and unlimited. The limited trend model applies an optionally time-dependent logistic growth model.

$$g(t) = \frac{C}{1+\exp\left(-k\left(t-m\right)\right)}, \tag{9}$$

where:

C: the constant carrying capacity (upper limit of the trend),

k: growth rate,

m: an offset parameter.

The unlimited trend model applies a piecewise linear regression model. The growth rate is the base rate and the sum of all the past rate changes.

$$g(t) = \left(k + \boldsymbol{a}(t)^T \boldsymbol{\delta}\right)t + \left(m + \boldsymbol{a}(t)^T \boldsymbol{\gamma}\right), \tag{10}$$

where:

k: growth rate;

S: number of rate change points;

s_j: the change in the rate occurring at time s_j; $j = 1 \ldots S$;

$\boldsymbol{a}(t) = \left[a_1(t) \ldots a_S(t)\right] \in \mathbb{B}^S$ is a selector vector of all change points before the actual time instance;

$$a_j(t) = \begin{cases} 1, & \text{if } t \geq s_j, \\ 0, & \text{otherwise.} \end{cases};$$

$\boldsymbol{\delta} = \left[\delta_1 \ldots \delta_S\right] \in \mathbb{R}^S$: vector of rate adjustments;

$\boldsymbol{\gamma}$: vector makes the function continuous. The offset parameter is needed to connect the segments at the point of rate change (to make the change more continuous), $\boldsymbol{\gamma} \in \mathbb{R}^S$.

Prophet uses Fourier series for seasonality modelling:

$$s(t) = \sum_{m=1}^{M}\left(a_m \cos\left(\frac{2\pi mt}{P}\right) + b_m \sin\left(\frac{2\pi mt}{P}\right)\right) \tag{11}$$

where:

P: period time;

M: number of components in the decomposition;

a_m and b_m: the estimated parameters.

In short, Prophet focuses on the typical and average cases; thus, it mainly suppresses extreme values (outliers) because dimensioning for outliers is contra-productive for utilization, and availability is not a primary concern.

2.4 Dynamic time warping

Uneven peak intervals can make the seasonality inference difficult. If the unevenly occurring peaks could be aligned to the same peak intervals without affecting EVA, then the accuracy of the seasonality could be enhanced.

POT is insensitive to the ordering of the data. Thus the precise location of the peaks is irrelevant, and accordingly, moving the peaks does not distort the GPD. We will exploit this insensitivity to ordering for time series preprocessing.

Contrarily, the order of data influences the way blocks are created in AMS. Hence, block maxima can change if peak values are moved between blocks (Fig. 4).

To better assure the representativeness of the peak values and the *iid* property, the peaks must be transformed (warped) to a more predictable periodical phenomenon.

When comparing two time series, an initial approach could be orderly matching (mapping) the elements and summing up the point-to-point distances. This *pointwise* orderly matching is also known as *linear* (*Euclidean*) *matching*. The problem with *linear matching* is that it does not exploit a set/cluster of very close points and cannot handle uneven random intervals (Fig. 5).

Non-linear matching allows the matching of non-orderly elements. Consequently, it can handle unevenness and potentially reduce the overall distance by exploiting data points where point-to-point distance is small (Fig. 5).

For point-to-point distance calculation, various functions can be used, with Euclidean being the usual.

Dynamic time warping (DTW) compares two time series, which may vary in speed, and measures their similarity by optimally matching the two time series [11]. The similarity is the minimum distance between the two time series given a set of constraints. The original use case for DTW is speech recognition, where the speaker might speak faster/slower than the comparison data. In other words, the matching problem is analogous to data clustering. Thus DTW resembles a form of clustering.

The core of DTW is an $N \times M$ matrix called the *distance matrix*. N and M are the numbers of elements in the compared time series. The distance matrix represents all the possible matchings between the time series (Fig. 6).

- The cell (i, j) represents that the j-th element is matched to the i-th element.
- The value in cell (i, j) is the sub-total minimum distance of the matching until elements i and j starting from the beginning.
- The sub-total is the point to point distance between i and j plus the sum of the distance from previous matchings on the optimal path to (i, j)
- The DTW algorithm starts at the bottom left (1, 1), fills the first row, then first column, then goes row by row, left to right until the top right (N, M) cell.

The path that minimizes the total distance is the *warping function*. Constraints can modify the warping function:

- *monotonicity*: the alignment cannot go back
- *continuity*: the alignment cannot skip elements
- *boundary conditions*: the alignment starts at the bottom left and ends at the top right to cover both time series entirely
- *warping window*: how far the warping path can go from the X-Y diagonal
- *slope constraint*: what is the maximum number of consecutive steps in the same direction before a step in the other direction must be taken

Additionally, the *step patterns* govern the traversal of the matrix: what are the allowed steps and their costs. The cost of stepping is added to the distance (Fig. 7).

DTW can be fine-tuned by configuring the step patterns, point-to-point distance functions, and constraints, making DTW a flexible, general-purpose tool.

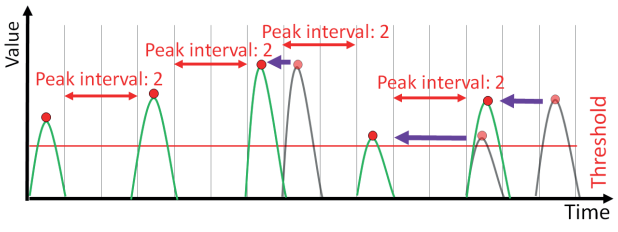

Fig. 4 Variable peak intervals and temporal regularization

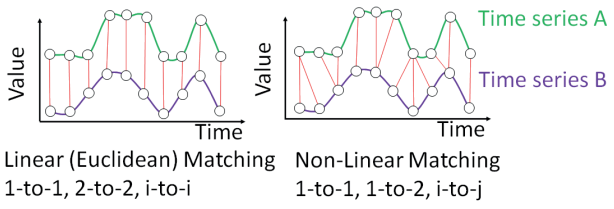

Fig. 5 Orderly one-to-one mapping and non-linear mapping

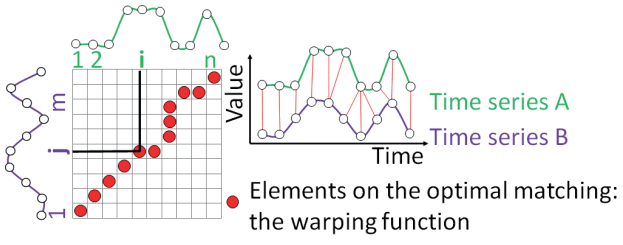

Fig. 6 Distance Matrix

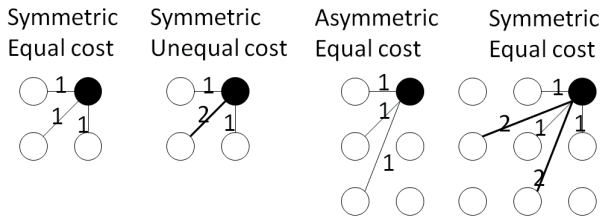

Fig. 7 Example step patterns

3 A method for variable peak workload modeling

As mentioned before, EVA necessitates peak representative and *iid* data. Accordingly, the combined algorithm performs temporal regularization of peak periodicity, time series decomposition, and POT de-clustering in the following way:

1. *Slice size determination* based on the observed high-level seasonality. A *slice* is a fixed number of consecutive points, similar to AMS blocks.
2. *Slicing*: divide the dataset into equal length blocks.
3. *Master slice selection*: the other slices will be compared to the master slice and transformed.
4. *Warping*: compare the master slice to all the other slices using DTW.
5. *Temporal regularization*: based on the *warping functions*, re-align (transform) the slices to *regularize the peak periodicity*. When a single point is matched to multiple points, take the maximum value and discard the rest.
6. *Reassembly*: orderly combine the slices to recreate the full-length time series.
7. *Time Series decomposition of* the combined dataset to *identifiable time-series patterns*.
8. *Separation of the extreme component*: extract the residual component to *eliminate patterns (and correlation) that can be modeled using the time series decomposition technique of the previous step.*
9. *POT De-cluster* the residual component to compensate for *local correlation*.
10. *EVA*: execute POT on the residual component.
11. *Quantitative aggregation*: combine the POT estimate of the residual component with the other components.

3.1 The applicability of the proposed method

The proposed method does not guarantee results. However, there are numerous ways to improve the results via tuning DTW and time series decomposition steps.

Since the proposed method uses the visible peak patterns for DTW-clustering-based temporal regulation, the method works if slicing makes sense: the recurring peak patterns vary in speed/time but otherwise similar.

Additionally, DTW can be omitted if the used time series decomposition method can accurately model the temporal variation of the peaks.

4 Case study

The pilot dataset originates in the workload records of a university virtual computing lab (VCL). The resultant workload comprised different non-iid workload sources: the students driven by homework submission deadlines and researchers conducting computation tasks (Fig. 8).

There is apparent seasonality between different semesters due to the yearly changing homework submission deadlines. Moreover, in each semester, there are high bursts of recurring workload peaks before the deadlines (Fig. 9).

A non-EVA approach, like Prophet, could underestimate the peaks by suppressing the extremes as outliers.

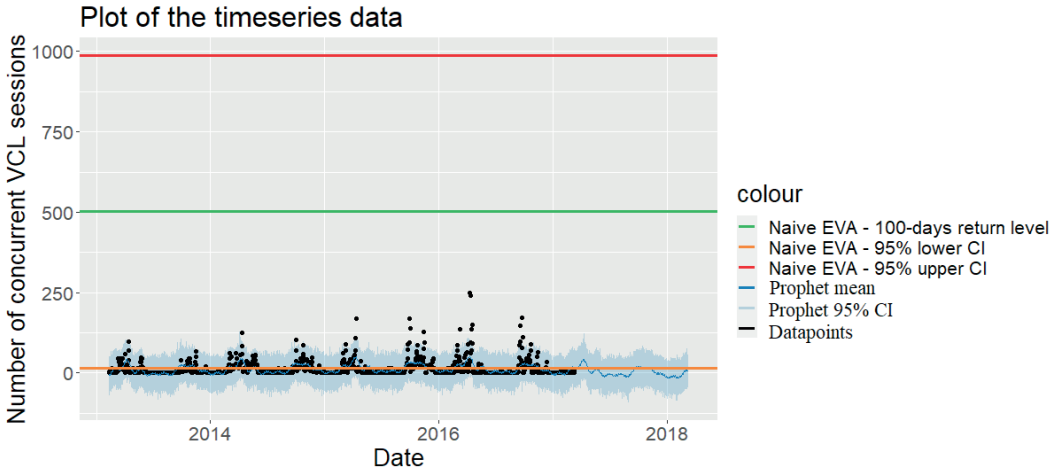

Fig. 8 The Original dataset with Prophet and a Naïve EVA prediction

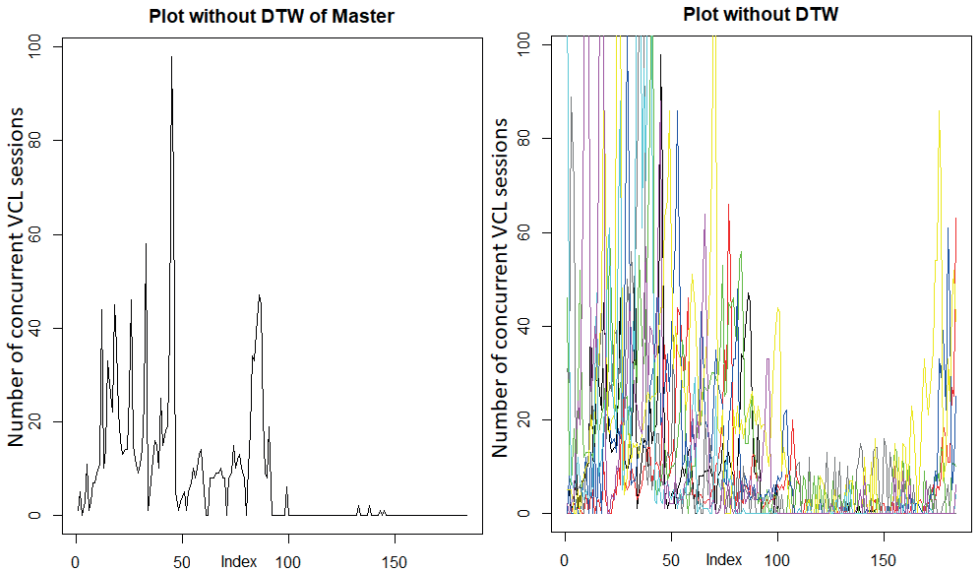

Fig. 9 The master slice, and all the slices compared

Meanwhile, an EVA-only approach could lead to wasteful overprovisioning by neglecting time series patterns: quasi periodicity, local correlation, and trend [12].

4.1 Peak re-alignment via DTW

There is a similarity in the semester workloads: varying peaks in the first half of the semester, a low workload section, and some peaks at the end of the semester (Fig. 9).

One of the significant problems in finding a proper master slice ("etalon" sample) is the appearance of bursty background noise like workload. Appropriate separation of their impact is a research task itself [13, 14].

Engineering background knowledge helped us to select the first semester as a master slice, because it had a cleaner waveform. The VCL was primarily introduced to serve the preparation of homework. Initially, other tasks were rare, which resulted in a cleaner waveform in the first semester. In the subsequent semesters, spare capacity usage became more popular, which led to increased utilization in the quiet period and relatively near the peak periods, resulting in an unclean, mixed waveform. However, the effect of other usage was limited near to the peaks, so no superposition-like interferences occurred. This way, the fundamental similarity of the waveforms around the peak loads remained highly undistorted. Only their temporal position did change according to the deadlines in the individual semesters.

DTW visibly improved the similarity of the slices for the high peaks, and distortions of the original values occurred primarily at low levels irrelevant for POT (Fig. 10).

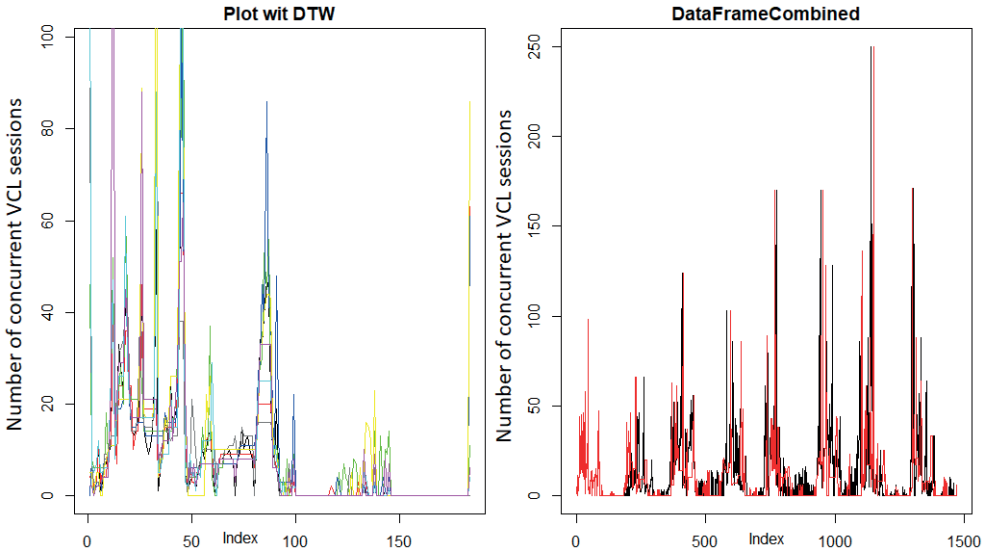

Fig. 10 All the slices and the whole dataset after DTW alignment, original dataset: black line, DTW re-aligned dataset: redline

4.2 Analyzing the results of DTW: (auto) correlation

Considering the Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), Periodogram and the greatly increased slice correlations, it can be concluded that DTW had the desired effect of aligning the peaks. Moreover, the correlation between the different slice pairs, 1–to–2, 2–to–3, …, 7–to–8 increased as expected (Figs. 11 to 14).

4.3 Discussion

We analyzed the dataset in different ways (Table 2). The 100 year return level is the level that is exceeded on average every 100 years, in our case, every 100 days. We calculated 95% confidence intervals (CI).

The maximum in the test dataset was 250 concurrent VCL sessions. As a reference, the mean of the naïve EVA estimation was 502. Compared to this, the closest mean estimate was 258 with DTW, Log transformation, and Prophet. The second closest mean estimate was 270 using STL with and without DTW, which shows the robustness of STL to the uneven peak intervals. For the DTW+LOG+STL case, we had outlier results because a proper threshold could not be selected. The threshold was either too low (GPD not fitting) or too large (too few data points).

5 Conclusions

Generally speaking, based on our results, we conclude that time series decomposition and DTW can increase the

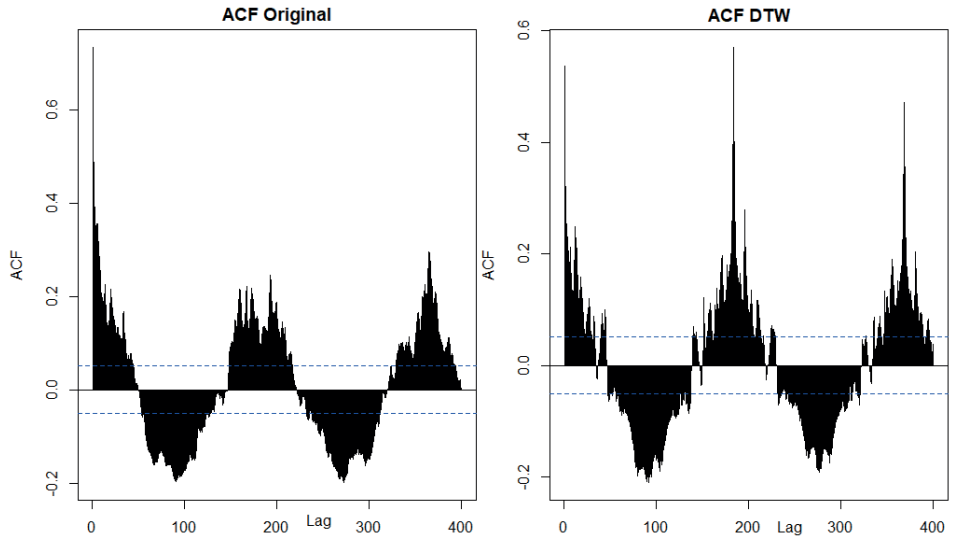

Fig. 11 ACF plot for the original and DTW aligned data

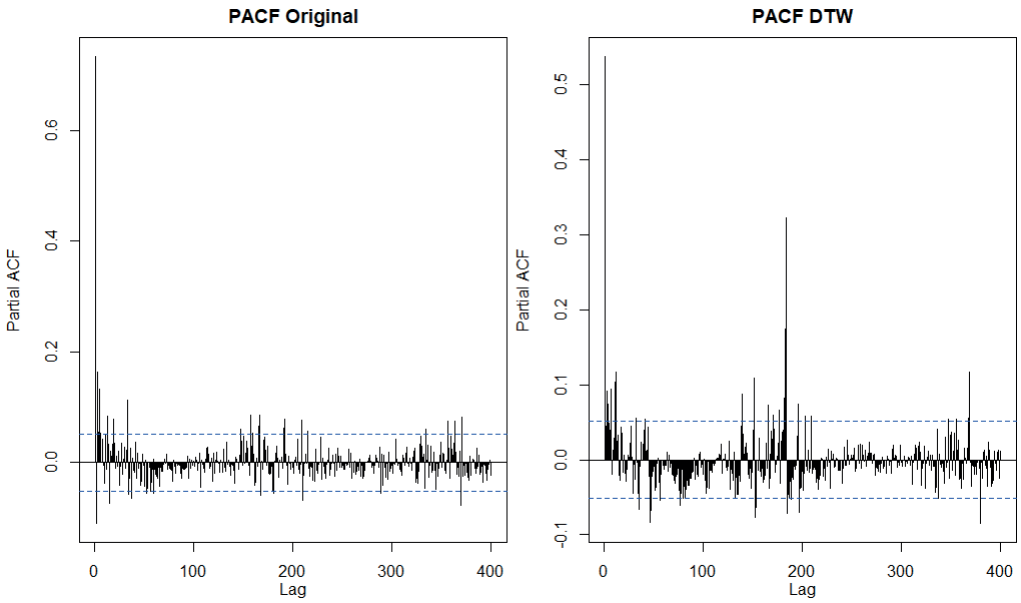

Fig. 12 PACF plot for the original and DTW aligned data

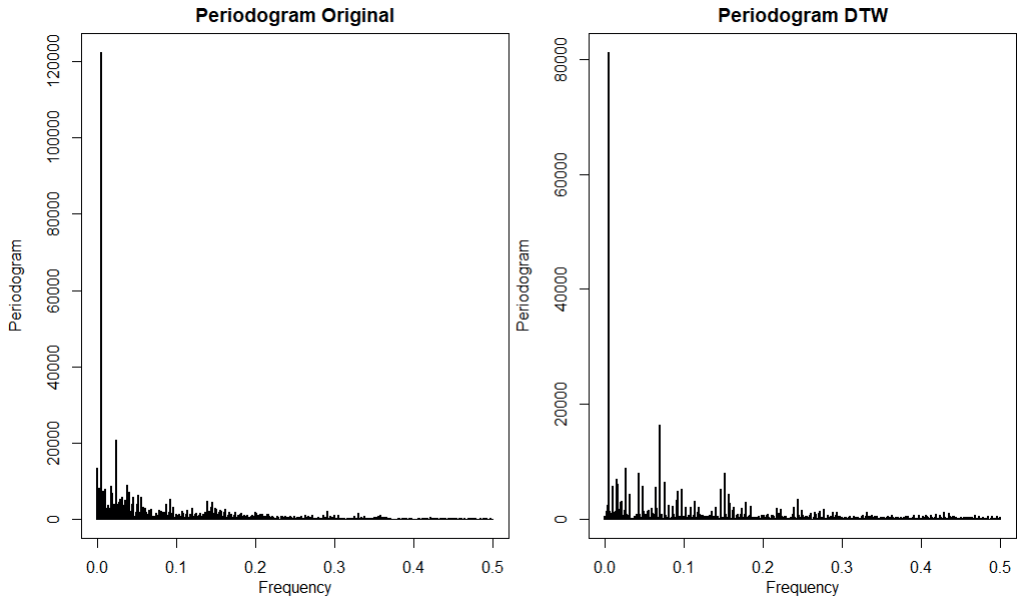

Fig. 13 Periodogram for the original and DTW aligned data

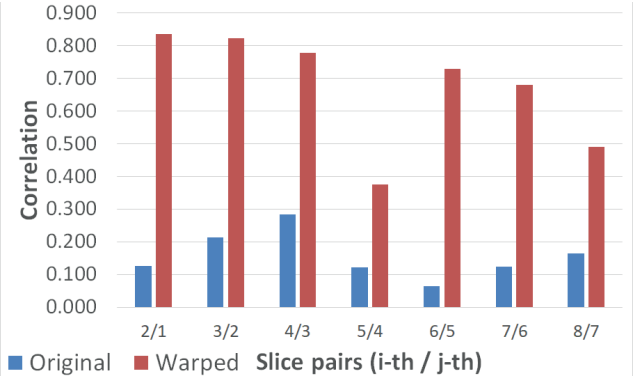

Fig. 14 Correlation between the slices for the original and the warped

Table 2 100 year return level of aggregated results

Ideas used	Method	LO CI	Mean	HI CI
NONE	Naïve EVA (reference)	16	**502**	988
TS	STL+ARIMA	23	**270**	517
TS	LOG+STL+ARIMA	227	497	767
TS	Prophet	-168	394	956
TS	Log Prophet	-1	299	599
DTW+TS	STL+ARIMA	14	**270**	525
DTW+TS	LOG+STL+ARIMA	**-17711**	**6502**	**30714**
DTW+TS	Prophet	-129	338	805
DTW+TS	Log Prophet	43	**258**	474

accuracy of POT in various cases. Time series decomposition was more effective when it could tolerate uneven peak intervals, such as STL. Meanwhile, DTW was more effective when the used decomposition method did not tolerate uneven peak intervals. Meanwhile, there are still open research questions:

1. how to select or artificially create the master slice,
2. how to tune DTW.

References

[1] Feitelson, D. G. "Workload Modeling for Computer Systems Performance Evaluation", Cambridge University Press, Cambridge, UK, 2015. https://doi.org/10.1017/CBO9781139939690

[2] McNeil, A. J., Rüdiger, F., Embrechts, P. "Quantitative Risk Management: Concepts, Techniques, and Tools", Princeton University Press, Princeton, NJ, USA, 2015.

[3] Rakonczai, P. "On Modeling and Prediction of Multivariate Extremes", [pdf] PhD Thesis, Lund University, 2009. Available at: http://web.cs.elte.hu/~paulo/pdf/RakonczaiLic.pdf [Accessed: 29 June 2021]

[4] Cízek, P., Härdle, W. K., Weron, R. "Statistical Tools for Finance and Insurance", Springer, Berlin, Germany, 2011. https://doi.org/10.1007/978-3-642-18062-0

[5] Caeiro, F., Gomes, M. I. "Threshold Selection in Extreme Value Analysis: Methods and Applications", In: Dey, D. K., Yan, J. (eds.) Extreme Value Modeling and Risk Analysis. Methods and Applications, Chapman and Hall/CRC, New York, NY, USA, 2016, pp. 69–86. https://doi.org/10.1201/b19721

[6] Scarrott, C., MacDonald, A. "A review of extreme value threshold estimation and uncertainty quantification", [pdf] REVSTAT - Statistical Journal, 10(1), pp. 33–60, 2012. Available at: https://www.ine.pt/revstat/pdf/rs120102.pdf [Accessed: 29 June 2021]

[7] Cleveland, R., Cleveland, W. S., McRae, J. E., Terpenning, I. "STL: A Seasonal-Trend Decomposition Procedure Based on Loess", [pdf] Journal of Official Statistics, 6(1), pp. 3–73, 1990. Available at: https://www.wessa.net/download/stl.pdf [Accessed: 29 June 2021]

[8] Cleveland, W. S., Devlin, S. J. "Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting", Journal of the American Statistical Association, 83(403), pp. 596–610, 1988.
https://doi.org/10.1080/01621459.1988.10478639

[9] Box, G. E. P., Jenkins, G. M., Reinsel, G. C. "Time Series Analysis: Forecasting and Control", John Wiley & Sons Inc., Hoboken, NJ, USA, 2008.
https://doi.org/10.1002/9781118619193

[10] Taylor, S. J., Letham, B. "Forecasting at Scale", The American Statistician, 72(1), pp. 37–45, 2018.
https://doi.org/10.1080/00031305.2017.1380080

[11] Sakoe, H., Chiba, S. "Dynamic programming algorithm optimization for spoken word recognition", IEEE Transactions on Acoustics Speech and Signal Processing, 26(1), pp. 43-49, 1978.
https://doi.org/10.1109/TASSP.1978.1163055

[12] Bozóki, Sz., Pataricza, A. "Extreme value analysis for capacity design", International Journal of Cloud Computing, 7(3/4), pp. 204–225, 2018.
https://doi.org/10.1504/IJCC.2018.095353

[13] Casale, G., Mi, N., Smirni, E. "Model-Driven System Capacity Planning under Workload Burstiness", IEEE Transactions on Computers, 59(1), pp. 66–80, 2010.
https://doi.org/10.1109/TC.2009.135

[14] Casale, G., Mi, N., Smirni, E., Cherkasova, L. "How to parameterize models with bursty workloads", ACM SIGMETRICS Performance Evaluation Review, 36(2), pp. 38–44, 2008.
https://doi.org/10.1145/1453175.1453182