

A Visual Analytics Approach for Peak-Preserving Prediction of Large Seasonal Time Series

M. C. Hao¹, H. Janetzko², S. Mittelstädt², W. Hill¹, U. Dayal¹, D. A. Keim², M. Marwah¹, R. K. Sharma¹

¹Hewlett Packard Laboratories and EB IT Services, Palo Alto, CA

²University of Konstanz, Germany

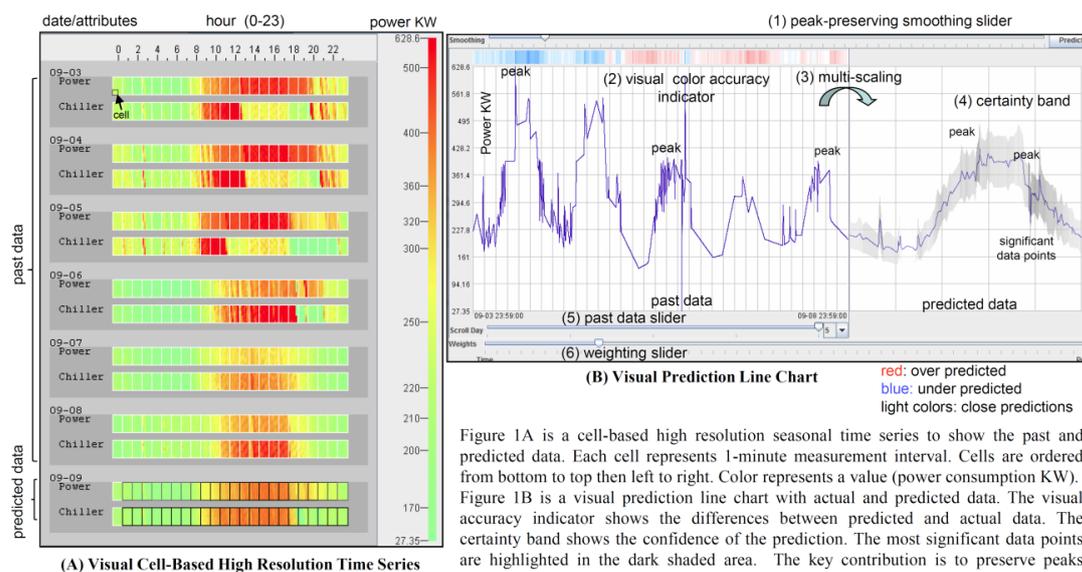


Figure 1: Visual prediction of next day's power consumption KW from historical data in a data center.

Abstract

Time series prediction methods are used on a daily basis by analysts for making important decisions. Most of these methods use some variant of moving averages to reduce the number of data points before prediction. However, to reach a good prediction in certain applications (e.g., power consumption time series in data centers) it is important to preserve peaks and their patterns. In this paper, we introduce automated peak-preserving smoothing and prediction algorithms, enabling a reliable long term prediction for seasonal data, and combine them with an advanced visual interface: (1) using high resolution cell-based time series to explore seasonal patterns, (2) adding new visual interaction techniques (multi-scaling, slider, and brushing & linking) to incorporate human expert knowledge, and (3) providing both new visual accuracy color indicators for validating the predicted results and certainty bands communicating the uncertainty of the prediction. We have integrated these techniques into a well-fitted solution to support the prediction process, and applied and evaluated the approach to predict both power consumption and server utilization in data centers with 70-80% accuracy.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image Generation – Display Algorithms; H.5.0 [Information Systems]: Information Interfaces and Presentation – General

1. Introduction

The motivation for analyzing time series data is to find patterns and trends and predict the likelihood of future events based on historical data. Data center administrators, for example, want to predict the next day's power consumption from the previous months' data. A retailer needs to predict the number of products to be stored in the warehouse this month using last year sales data.

Classic statistical methods for time series prediction are model-based (e.g., ARIMA and Holt Winters [AH94], G-TSFE [CNC00]) and smoothing, trend, or similarity-based [BAPKS05]. Each method has its own characteristics and applications. For example, Holt Winters is used for seasonal data, while ARIMA is used for non-seasonal data. Most methods are heavily dependent on the application used and require the user's expertise to drive the prediction.

We introduced a visual time series prediction idea in an IEEE VAST09 poster paper [HHRUK09]. In this paper, we present an extension of that work and propose peak-preserving smoothing and prediction algorithms to enable data centers to budget their resources without exceeding their capacities. An illustrative example predicting the next day's power consumption in a data center is shown in Figure 1.

Our contributions

In this paper we introduce peak-preserving smoothing and prediction methods which allows a reliable long term prediction for seasonal data. An advanced visual interface enables the users to steer the process and produce better prediction results, as shown in the visual prediction line chart (Figure 1B); prediction accuracy color indicators and certainty bands show the quality of the prediction; Brushing and linking between past and future data allows to compare their differences; a peak-preserving smoothing slider enables the users to reduce data points without missing peaks; and a weighting slider helps users to adjust the weight between peaks and time distances (e.g., recent data points are more important than those further in the past). This iterative process helps to refine the predictions.

To predict over large volumes of time series data, a cell-based high-resolution time series [HDKS07] is provided to discover patterns (Figure 1A). We have applied the above techniques successfully to resource consumption (e.g.; power and chiller) and server utilization predictions in data centers with a good accuracy.

This paper is organized as follows: Section 2 introduces related work. Section 3 describes our approach. Section 4 explains the peak-preserving smoothing and

prediction methods. Section 5 introduces the visual analytics techniques. Section 6 presents applications to real-world datasets and section 7 evaluates our approach. Section 8 concludes and outlines future work.

2. Related Work

Time series prediction methods are important for making business decisions. A large number of prediction techniques have been developed over the past years. Some of these techniques are closely related to our work. The previously developed techniques can be classified into two categories.

Prediction Algorithms

ARIMA and Holt Winters are two well-known algorithms used in many different applications, such as sales, ATM and traffic prediction. ARIMA is mostly used for trend analysis. To enable ARIMA to capture long-range-dependences for high speed network traffic prediction, Sadek [SKC04] introduced Fractional ARIMA for predicting traffic values at different time scales. F-ARIMA is able to capture both the short- and long-range characteristics of the underlying data.

Holt [HOLT52] and Winters [WIN60] both used exponentially weighted moving averages to forecast seasonal sales data. Their forecast is a function of past and current sales using exponential smoothing. Taylor [TAY07] applied the Holt Winters techniques to predict daily supermarket sales using exponentially weighted quantile regression for inventory control. Taylor extended exponential smoothing based forecast to cumulative distributed function level forecast for better predictions. We applied Holt-Winters algorithms to predict the next day's temperature for the data center. The results from Holt Winters are very close to our prediction results (Figure 8), but peaks are missing from their prediction.

Prediction visualization techniques

Broberg [BLH02] applied Kalman Filter algorithms [KAL60] to predict the behavior and execution of multithreaded programs using line charts. Ichikawa [IT02] introduced a visualization environment which allows users to view a large number of stock price predictions using different types of line charts, texture, color, and 3D graphs. Croker [CRO07] de-cluttered the forecast plot by using colored regions for showing forecast prediction and confidence level. Masse [MAS08] proposed a visual approach for the U.S. presidential election prediction. SAS Forecasting System [SAS11] provides automatic model fitting and forecasting with confidence limits on time series data. SAS compares actual and forecasted values with

prediction errors using different colored time series. For visualizing large complex time series, many layers of time series may clutter the display. Our technique uses accuracy color indicators to show the difference between the past and predicted data and can be scaled up for visualizing large time series.

3. Our approach

All the above related algorithms and methods are advanced prediction solutions and some provide visualization support. However, none of the related work considers peaks and the time distances (e.g., how recently measurements have been recorded). Often, peaks are removed along with the noise during smoothing operations. In many seasonal applications peaks play an important role. For economic reasons data centers should use their resources as efficiently as possible. Without integrating expected peaks in the prediction, there is a risk of exceeding the provided power or cooling capacities that can lead to damage or unavailability of equipment.

However, the peak-preserving techniques cannot be used in all applications. In sales applications, for example, one time peaks are usually not relevant. Also, in signal processing a peak generally is regarded as an unwanted noise that needs to be filtered out.

In our prediction method, recent data points are weighted higher than the earlier data points. But we also need to preserve peaks, even if they occurred much earlier. To achieve our goals, we introduce peak-preserving smoothing to retain the peaks. Furthermore, we propose peak-preserving prediction to particularly consider the peaks in the prediction process. We also provide a weighting slider for administrators with their domain knowledge to balance the importance between the time distance and peak preservation.

The overall process of our visual peak-preserving prediction technique is described in an iterative three-step solution (Figure 2):

1. Select an interesting seasonal pattern from a cell-based high resolution time series.
2. Use the *peak-preserving smoothing* algorithm to remove noise from the time series but retain the peaks before applying the prediction methods. Users can use *multi-scaling* to adjust to the best view and use the *visual accuracy color indicator* to get an impression of the quality of the past predictions without cluttering the display. In addition, users can move the *smoothing slider* to analyze the effects of smoothing.
3. Generate the *peak-preserving prediction*. A *visual certainty band* is provided for users to validate their prediction results using the certainty proportional to the width of the band.

The user can also move the *weighting slider* to balance the weights between peak preservation and time distance depending on the application. Also, users can use *brushing and linking* techniques to examine the predicted data and compare it to the historical data.

4. Peak-Preserving Smoothing and Prediction

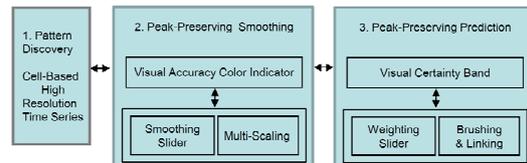


Figure 2: Visual Peak-Preserving Prediction Process.

Both peak-preserving smoothing and peak-preserving prediction are automated methods for generating better predictions, which highlight peaks and their patterns.

4.1. Peak-preserving smoothing

In our experiments with resource consumption in data centers, in many cases the existing prediction methods do not provide sufficiently good results. One reason is that the data is usually very noisy. Smoothing can be used to reduce the negative effects of noise on the prediction, but it is always a compromise between removing noisy data and retaining potentially valuable information. The original shape of a time series must not be changed during the noise reduction process. The shape not only shows the global trend, but also the local extremes, which have high frequency characteristics.

In the VAST 2009 poster [HHRUK09], we used weighted moving averages for the prediction. For every data point, the corresponding past data points are weighted on the basis of a Gaussian distribution and included in the average according to their weights. This technique performs very well in removing clutter, but behaves like a low-pass filter. Thus, significant high frequency peaks are lost for prediction.

In order to retain peaks we derive a variant version from the well-known Douglas-Peucker [DP72] algorithm, which reduces a graph to its most significant data points (see Figure 3). Our approach simplifies the calculation of the original version. Due to the fact that time series are simple graphs, we speed up the smoothing by computing the distance along the vertical axis which leads to the same results, but significantly higher efficiency. The algorithm starts with creating a connecting line (blue), which connects the first and the last value. Then, it searches for the highest or lowest data point in between these values with respect to the connecting line. If the absolute height of the data point exceeds a certain threshold, this data point is tagged as

a peak (Figure 3a). At this point, the graph is divided into two with both parts containing the peak value. The algorithm performs these steps recursively again on both parts, trying to find the next peak in each part (Figures 3b and 3c). If no more peaks can be found, the process ends (Figures 3c and 3d). To create the smoothed result, all peak points (including the first and the last value of the original graph) are connected. Figure 3e shows the smoothed result of the example.

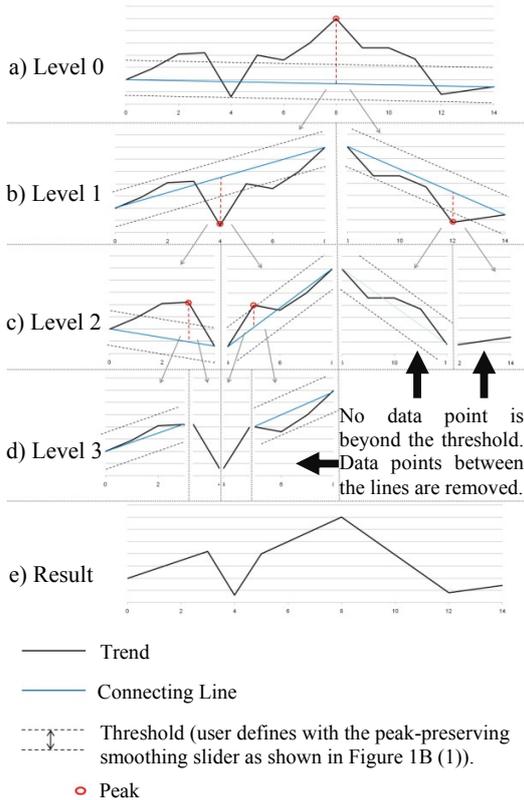


Figure 3: A variant version of Douglas-Peucker algorithm.

A proper threshold is crucial for producing desirable results with this algorithm, which varies from application to application. Since users, aided by visual feedback are more adept at picking a good value for the threshold than automated estimates, we allow the users to modify the parameter with the peak-preserving smoothing slider (Figure 1 B (1)) and represent the result of the selection in real time. The final graph represents the shape of the original data very precisely and without clutter.

Figure 4 shows the difference between weighted moving average smoothing and peak-preserving

smoothing. While both approaches remove noise very well, the outlined significant peak is missing in (B). In subfigure (C) the peak is preserved and thus, it influences the prediction.

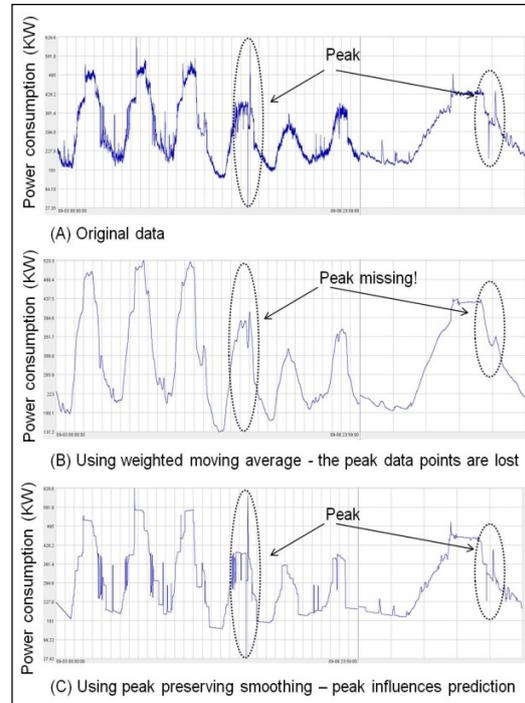


Figure 4: Comparison of Peak-Preserving Smoothing to Weighted Moving Average Smoothing.

4.2. Peak-preserving prediction

In many application domains, e.g., server utilization or temperature time series, the prediction should serve two purposes:

- Predict the overall trend and make the analyst aware of possible future developments.
- Predict a probable time distance for peak points where a critical value has been reached, which are usually minimum and maximum peaks

In the server utilization application (Section 6.2), it is important to know whether it is possible to shutdown some servers to save power and cooling cost, because these servers could be temporarily unused. Additionally, it is not only important to take the peaks into account, but in the case of trend analysis it is also important to know how recently measurements were made. Values that are one month old are not likely to

be as influential as more recent measurements (without knowing more about external events). Both goals, preserving peaks and considering the distance in time, can be contradictory, as peaks could be distant in time from the present measurements. Therefore, we provide a way to enable the users to influence the balance between both these goals via a weighting slider as shown in Figure 1B (6). The slider can be moved to the left to increase the time-distance weight and to the right to increase the peak-preserving weight.

```
double[] doPrediction(double[] pastValues,
                    Date[] dateOfPastValues,
                    double[] importancePeakWeights) {
    // create temporary storage:
    double valueForEachMinuteOfTheDay[]
    = new double[60 * 24];
    int counterForEachMinuteOfTheDay[] = new int[60 * 24];

    double c = calculateConstant(numberOfDays);
    for (int i = 0; i < pastValues.length; i++) {
        Date d = dateOfPastValues[i];
        int minuteOfTheDay = d.getHours() * 60 + d.getMinutes();
        counterForEachMinuteOfTheDay[minuteOfTheDay]++;
        // Add the current value multiplied with a computed weight to
        // the right slot, as we are calculating a weighted average
        valueForEachMinuteOfTheDay[minuteOfTheDay] +=
            values[i] * combinedWeights(
                counterForEachMinuteOfTheDay[minuteOfTheDay] * c,
                importancePeakWeights[i], userSetValue);
    }
    return valueForEachMinuteOfTheDay;
}
```

Figure 5: Algorithm for prediction based on daily patterns.

As depicted in Figure 5, our peak-preserving prediction algorithm generates the predicted data points based on the time period of the historical data. The predicted data points are computed in hours, days, weeks, and months across the entire dataset instead of by the traditional moving average method. Which level of time to group is highly application dependent and in our application we have to use a daily grouping. If we want, for instance, to predict the data point for the time 0:00 o'clock, we look for all measurements made at 0:00. We assign to each of these data points different weight factors and afterwards aggregate the values according to the weights. In particular, we first initialize some temporary storage for each minute of the day and also compute a constant, which is described below. Afterwards, we iterate through the data of the past and determine for each value, on which minute of the day the measurement has been made. Lastly, we add the currently seen data value to the temporary storage slot by multiplying it with a combined weight, which will be explained below. A single predicted value in Figure 1 (a daily grouping) can be therefore calculated as follows, where all value v are measured during the same time interval (minute) of the day:

$$pred(\text{min Of Day}) = \sum \text{combinedWeights} \cdot v$$

The weights regarding the distance in time are linearly decreasing with increasing time distance. Furthermore, the sum of the weights is set to be 1, as we are using them for a weighted average. The weights should therefore look like $1 * c, 2 * c, 3 * c \dots$ where c is a constant, normalizing the weights of the result. The following equation describes the calculation necessary to compute the constant and weights:

$$\sum_{i=1}^n i * c = c * \sum_{i=1}^n i = c * \frac{n * (n + 1)}{2} = 1$$

$$\Rightarrow c = \frac{2}{n * (n + 1)}$$

In this equation n is the number of weights we need to calculate. Using these weights, the recent data will influence the prediction more than the older values.

The second goal, preserving peaks, is also achieved by computing weights. In this case we are using a side outcome of the smoothing algorithm described in section 4.1. As the peak-preserving smoothing algorithm recursively partitions the data set according to the highest or lowest peak, we can use the recursion depth to estimate the importance of a peak (Figure 3). For this purpose we normalize the recursion depth according to the maximum occurring recursion level and use the inverse of the level as a weight. To avoid division by 0, we add a constant number (in our case 1) to all recursion levels before normalization.

As stated previously, we allow the user to influence the prediction. The weights are adapted to the user's choice by using a weighted average of the two weights. The combined weights are used for the prediction, allowing a user-controlled prediction for large time series. The *combinedWeights* method calculates a weighted average of two values v_1 and v_2 by using the userSetValue (abbreviated to α):

$$\text{combinedWeights}(v_1, v_2, \alpha) = v_1 \cdot \alpha + v_2 \cdot (1 - \alpha)$$

5. Visual Analytics

5.1. Visual cell-based high resolution time series

Time series used for prediction are often very long. In data center applications, analysts commonly use one month's historical thermal data to predict the next day's power and chiller utilization. Because of the limited screen size, we use a high-resolution cell-based time series visualization to show the data [HDKS07]. In Figure 1(A), cells (pixels) of each time series are arranged from bottom to top and left to right according to time. The size of the cells automatically scales down as more cells are displayed. The color of a cell represents the value in a measurement interval.

5.2. Visual prediction line charts

Figure 1(B) shows a visual prediction line chart for analyzing power consumption in a data center. It contains the following new visual interaction techniques for prediction.

Accuracy color indicators

An important component of our system is the *visual accuracy color indicator* shown in Figure 1B (2). This component is used to show the accuracy quality obtained by our prediction method for all time frames in the past compared to the actual value for that time interval. We normalize the differences between the predicted and actual values using the standard deviation. Then, we map these values to a color map. Dark colors indicate larger differences; light colors indicate smaller differences. The red/blue lines indicate that the actual values are higher/lower respectively than the predicted value. Figure 1B (2) shows many very close predictions (light blue/under, pink/over, and white lines/exact).

Multi-Scaling

Multi-scaling is an optional function used as needed. It allows the user to adjust the time scale between the actual and predicted data as shown in Figure 1B (3). For better visibility of the predicted data, users often assign a small portion of the screen to display the history data in order to have enough space to view the prediction results. This is helpful since the predicted time interval (one day) is usually quite small compared to the history time frame (e.g., a month).

Certainty Band and Significant Data Points

In order to give the analyst further help in interpreting the prediction, the predicted values are enclosed within a certainty band as shown in Figure 1B (4). The width of this band shows the range in which the values are to be expected. A narrow certainty band indicates that the predicted values will match the real value with high probability. The width of the band can be calculated by using the standard deviation of the past data. If the values of the past at the corresponding time points differ significantly then we have wide confidence bands. On the other hand, if the values of the same time points in the past are nearly the same then we have narrow confidence bands. Thus, the lower and upper bounds of each value can be calculated by adding or subtracting half of the standard deviation of its occurrences.

The shading of certainty band is used to indicate the significance of the associated data points. Darker areas have been more heavily influenced by peak values of the past. The probability of the corresponding values becoming significant is very high. Therefore, the user can find the peaks in these areas. The light areas indicate a stable or gradual curve, where peaks are unlikely. By using the variation of saturation, the user is led to focus on important areas of the predicted line graph and can perform further analysis by applying different weights for the prediction.

As illustrated in Figure 1A, the predicted data points are shown in the last two rows on date 09-09. To visualize the certainty of the predicted values in the

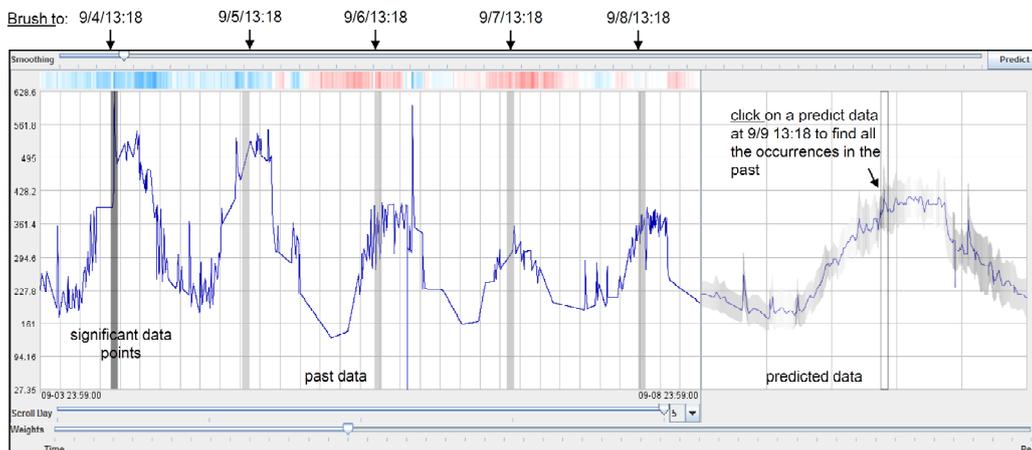


Figure 6: Brushing and linking predicted data to their past occurrences
Different shades of gray indicate the degree of significance
(dark: high significance; light: low significance)

cell-based time series, we change saturation (less saturated for low certainty, more saturated for higher certainty) of each cell. Additionally, we adapt the border color to the average certainty of each group of cells (light borders in case of low certainty, dark borders in case of high certainty).

Brushing & Linking between Past and Future

To determine the source of a peak or answer the question why a data point is significant in prediction, it is important to know the corresponding past values. Therefore, we apply brushing & linking in our advanced visual interface as illustrated in Figure 6. First, the user clicks on a predicted data value which needs to be examined. The value itself will be marked by an unfilled rectangle. The linked values in the past time series will be marked with a vertical rectangle which is shaded by a grayscale value. The shading of each rectangle corresponds to the importance of the value in the graph. In the same way, the user can select a point of interest in the past and see the predicted value which is affected and the related past occurrences. Thus, the user can easily detect the influencing values and their importance. This information can be used for further analysis.

Past Data Scrolling

In our application our tool needs to handle very large time series. To visualize this amount of information appropriately in the limited space available, we do not show the whole time series, but only reasonable parts of it. Due to the fact that our data follows daily patterns, we show a certain number of days of the past data as seen in Figure 1B (5), which are adjustable by the user. The users are able to scroll through the whole data stream for examination and, during this process they obtain the same feedback information mentioned in the sections above. Thus, the analyst can choose whether he wants an overview or follow the details very precisely. Figure 7(B) shows an example, where three days are shown.

6. Applications

We have applied the peak-preserving prediction techniques to a number of data center resource consumption datasets. The results show the wide applicability and usefulness of this visual analytics technique. For example, given the visual cell-based high resolution time series and visual prediction line charts, the administrator can quickly compare daily resource consumption patterns and prediction from each time series (e.g., power and chiller) for planning the next day's resource usages as shown in Figure 1. From Figure 7, the administrator notices the low application utilization during late evening, early morning, and weekends; and can switch off some servers and distribute applications to partially idle

servers to save the overall power consumed by applications (e.g., SAP).

The following two examples show prediction for daily periodicity. However, daily periodicity is not hardcoded. The periodicity is determined during the preprocessing step. In addition, the data can be separated into weekdays and weekends to obtain a better prediction.

6.1. Resource consumption prediction

A common question [PMSR09, SHA08] from data center administrators is: What resource consumption can we expect tomorrow from our previous usage pattern? To answer this question, we applied our peak-preserving techniques to a large data center covering

70,000 square feet with 2000 racks of IT equipment. The data has been collected from July to December 2008. A partial result is shown in Figure 1.

In Figure 1A, analysts can simultaneously perform predictions on both power and chiller cooling consumption using the peak-preserving prediction. Both patterns are seasonal and the prediction results are shown for day 2008-09-09 in Figures 1A and 1B. The chiller and power will be higher during the day (hours 10 to 17) and early evening (more red and orange), than early morning and late evening (mostly green and yellow). The administrator can navigate on both graphs and compare the differences between the real and predicted data points and drilldown for the detailed information. This helps the administrators configure their system for the next day. With the prediction, the data center is able to save about 40% resource consumption from switching off low utilization chillers [PMSR09 and BPS06].

6.2. Server utilization prediction

Optimizing the utilization of servers is a main cost factor in IT-Services center. The basic power consumption of an idle server is significant – approximately 50% of peak power usage. This leads to the conclusion that a server is utilizing power best when it is fully loaded and idle servers should be turned off. To reduce the risk of performance degradation, administrators have to analyze the server utilization patterns and relocate applications from servers which have a few applications running. For getting a reasonable high utilization, administrators are required to consolidate applications to a fewer servers.

Figure 7 shows a server's daily utilization on two common attributes (SAP utilization = normalized SAP application resource consumption and #of Users = number of application users) of 36,338 measurements. Figure 7A shows the SAP utilization and #of User are following cycles of high and low patterns. Figure 7B

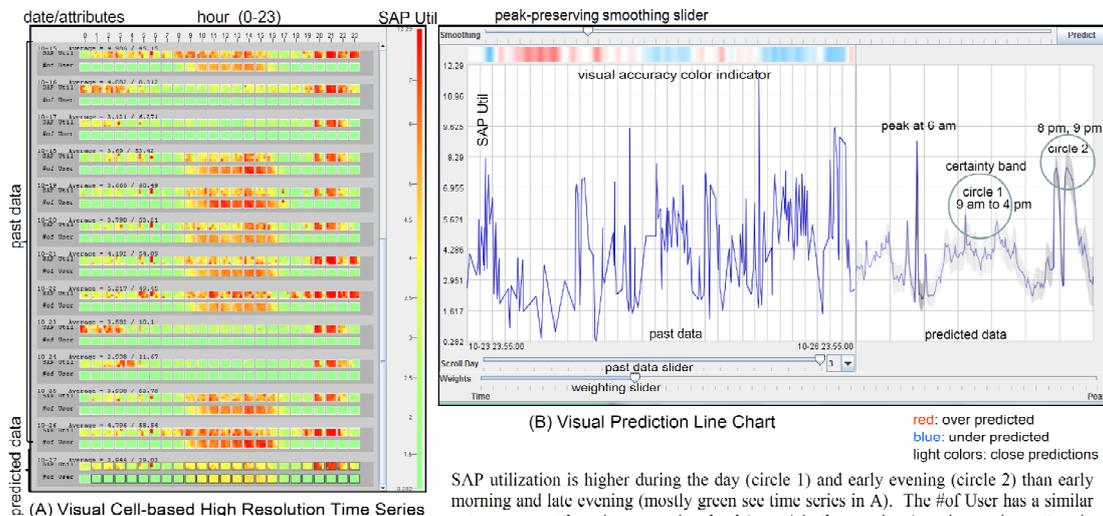


Figure 7: Visual prediction of daily SAP server utilization on Server A in a IT-Services center

shows the predicted SAP server utilization on the analyzed server.

From the visual accuracy color indicators, the quality of the prediction for all time intervals in the past is close to the real values (very few dark red/blue colors). Also, the narrow certainty band indicates that it is safe to move the applications to another server and power down this server from hour 22 in the late evening to 8 am in the early morning. The peak time is during the day for SAP applications and hours 20 to 21 for system

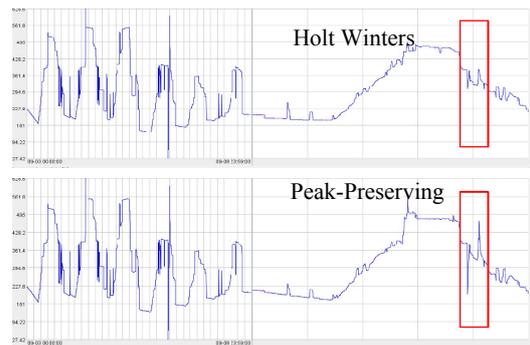


Figure 8: Comparison of Holt Winters method with our peak-preserving prediction. Peak removed in Holt Winters. Our prediction handles the peak better than Holt Winters method.

work. From our experiments, a power savings of up to 30% seems realistic.

7. Evaluations

In this section we discuss the peak preservation and evaluate the accuracy of our prediction approach.

7.1. Comparison with other prediction methods

As described above, the prediction algorithm uses weights to calculate the predicted future values. To show the peak-preserving abilities of our algorithm we only use the weights resulting from applying the improved Douglas-Peucker algorithm (Figure 3).

Figure 8 compares the existing prediction method Holt Winters, which is suitable for periodical data, with our proposed approach. The red box highlights that the proposed technique performs better in terms of peak preservation. As shown in Figure 8, the fast switch from no power consumption to high power consumption is not unusual in data centers.

This situation usually happens when the administrator exchanges the chillers (cooling units). But this sudden change in power consumption has to be considered in the planning activities, and it is therefore important to include these peaks in the prediction process.

7.2. Accuracy comparison between actual and predicted data

The server utilization from 10/06 to 10/26 (details in Section 6.2) has been used to measure the accuracy of our prediction approach. The values of every single day are predicted and compared to the observed real data. The result of this comparison shows an accuracy of 70% - 80% with an average accuracy of 75%. Figure 9 shows two predictions of different days (10/14 and 10/22). The upper figure shows the predicted values for Thursday 10/14 with an accuracy of about 76%. The lower figure shows Friday 10/22 with an accuracy of 74%.

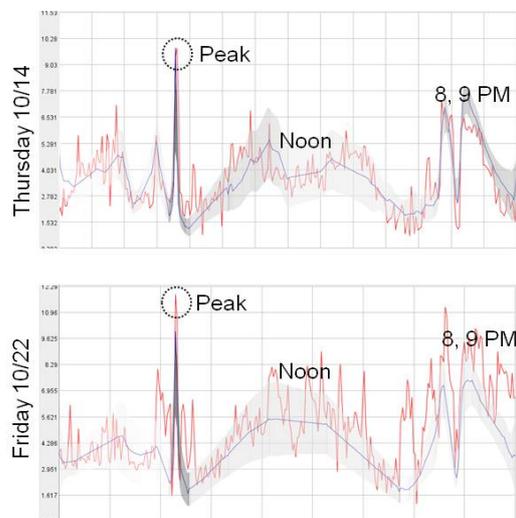


Figure 9: Prediction accuracy comparison between actual and predicted data (blue: predicted values / red: actual values)

8. Conclusion

In this paper, we presented a visual analytics approach for peak-preserving prediction of large seasonal time series. Our prediction technique preserves peaks and seasonal patterns. These methods are different from the previous work based on weighted moving averages. We combine our new prediction method with an advanced visual interface for analysts to quickly refine their predictions using domain knowledge. These new techniques have been successfully deployed and evaluated in data centers and IT-services centers. In our data center application, we were able to save the resource consumption of about 40% by switching off low utilization chillers and water pumps. In the IT-services application, we were able to redistribute applications and balance the workload among servers with a savings of up to 30%. By considering the peaks in the prediction, the life span of the equipments can

also be increased. Currently, we are exploring the prediction of network communication lines, which also have significant peaks and a potential overload problem. Our future work will proceed to analyze local trends and add external event influence, such as exceptions, holidays, and weather conditions.

ACKNOWLEDGEMENT

The authors wish to thank Mei Chun Hsu of HP Laboratories for her encouragement, Cullen E. Bash, for many discussions and suggestions, and for Shen-You Chen providing suggestions and data.

References

- [AH94] ARIMA and Holt Winters prediction models are described in the Chatfield, C. The analysis of time series: An Introduction, 6th ed., CRC Press, Boca Raton, USA. 2004.
- [BAPKS05] Buono, P., Aris, A., Plaisant, C., Khella, A., Shneiderman, B., Hochheiser, H., Schneiderman, B. Interactive Pattern Search in Time Series. Proceedings of Conference on Visualization and Data Analysis, VDA 2005, SPIE. CA.
- [BLH02] Broberg, M., Lundberg, L., Grahm, H. Visualization and performance prediction of multithreaded Solaris programs by tracking kernel threads. IEEE Xplore, 2002.
- [BPS06] Bash, C. Patel, R. Sharma Dynamic Thermal Management of Air Cooled Data Centers, IEEE Itherm06.
- [CNC00] Castellanos, M., Norman Salazar, N., Casati, F., Dayal, U., Shan M. TSFE Time Series Automated Forecasting Engine. Int. J. Computation Science and Engineering, 2000.
- [CRO07] Croker, S. T. Effective Forecast Visualization with SAS/GRAPH. SAS Global Forum, 2007.
- [DP72] Douglas D. & Peucker, T. "Algorithms for the reduction of the number of data points required to represent a digitized line or its caricature", The Canadian Cartographer 10(2), 112-122 (1973).
- [HHRUK09] Hao, M., Janetzko, H., Sharma, R., Dayal, U., Keim, D. A., Castellanos. Visual Prediction of Time Series. IEEE VAST09, Poster, Best Poster Award October, 2009.
- [HDKS07] Hao, M., Dayal, U., Keim, D. A., Schreck, T. Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. Proceedings: IEEE VGTC Symposium on Visualization, EuroVis 2007.

[HOLT52] Holt C. C., "Forecasting seasonal and trends by exponentially weighted moving averages. ONR Research Memorandum, Carnegie Institute 52.

[IT02] Ichikawa, Y., Tsunawaki, T. A Visualization environment for multiple Daytime Stock Price Predictions, 2002. IEIC Technical Report.

[KAL60] Kalman, R. E. "A new approach to linear filtering and prediction problems", Transactions of ASME, Journal of Basic Engineering, vol. 82, 1960.

[MAS08] Masse, C. F. "2008 US Presidential Election Prediction – A Visual Approach of InTrade's Prediction Markets. InTrade, 2008.

[PMSR09] Patnaik, D., Marwah, M., Sharma, R., Ramakrishnan, N. Sustainable Operation and Management of Data Center Chillers using Temporal Data Mining. In the proceedings of KDD'09, 6/09, France.

[SAS11] Time series forecasting system offered by SAS. <http://support.sas.com>.

[SHA08] Sharma, R. et al. On building next generation data centers: Energy flow in the information technology stack. In Proceedings of Compute 2008, 2008.

[SKC04] Sadek, N., Khotanzad, A., Chen T. "ATM Dynamic Bandwidth Allocation Using F-ARIMA Prediction Model" IEEE Xplore, 2004.

[TAY07] Taylor, J. W. "Forecasting Daily Supermarket Sales using Exponentially Weighted Quantile Regression" European Operational Research, 2007, Volume 207.

[WIN60] Winters, P. R. Forecasting sales by exponentially weighted moving averages, Management Science 6, 1960.