Fill-in the Gaps: Spatial-Temporal Models for Missing Data

Ji Xue[*], Bin Nie[*], and Evgenia Smirni[*]
[*]College of William and Mary
Williamsburg, VA, USA
{xuejimic,bnie,esmirni}@cs.wm.edu

***Abstract*—Effective workload characterization and prediction are instrumental for efficiently and proactively managing large systems. System management primarily relies on the workload information provided by underlying system tracing mechanisms that record system-related events in log files. However, such tracing mechanisms may temporarily fail due to various reasons, yielding "holes" in data traces. This missing data phenomenon significantly impedes the effectiveness of data analysis. In this paper, we study real-world data traces collected from over 80K virtual machines (VMs) hosted on 6K physical boxes in the data centers of a service provider. We discover that the usage series of VMs co-located on the same physical box exhibit strong correlation with one another, and that most VM usage series show temporal patterns. By taking advantage of the observed spatial and temporal dependencies, we propose a data-filling method to predict the missing data in the VM usage series. Detailed evaluation using trace data in the wild shows that the proposed method is sufficiently accurate as it achieves an average of 20% absolute percentage errors. We also illustrate its usefulness via a use case.**

I. Introduction

Collection of performance measures is central to the success of long-running systems that serve performance-sensitive applications. From supercomputing systems [1], to data centers [2], to storage systems [3], collecting traces of performance measures is instrumental for effective system management and resource allocation, for meeting user service level objectives, and for enhancing system reliability. For long-running, large, distributed systems it is a common phenomenon that the tracing mechanism of some of its components may periodically fail, i.e., it is possible that for a period of time there may be gaps in the trace data due to failures either in the actual measurement instrumentation but also into the actual recording of the system-related events in log files (e.g., due to transient errors in communication or storage infrastructure) [4], [1], [5]. It is common that such failures are transient, i.e., after a period of time, trace recording is restored.

The focus of this paper is on the analysis of data gaps in traces and on mechanisms that can potentially compensate for such missing data. Specifically, we study the CPU usage data of virtual machines (VMs) hosted on a multitude of physical boxes in IBM cloud data centers and we observe that for almost over 50% of the physical boxes, the VMs traces have significant gaps in their usage series. Past work on managing data centers [2] is based on effectively characterizing the system workload and on accurately predicting the upcoming one, this allows proactive management of resources and contributes to improving the user quality of experience. Yet, if only a portion of the traces is usable for prediction, improvements as those described in [2], [6] are restricted to the portion of the system that complete trace data are available. In this paper, we present a methodology that can compensate for missing data using statistical characteristics of the data traces.

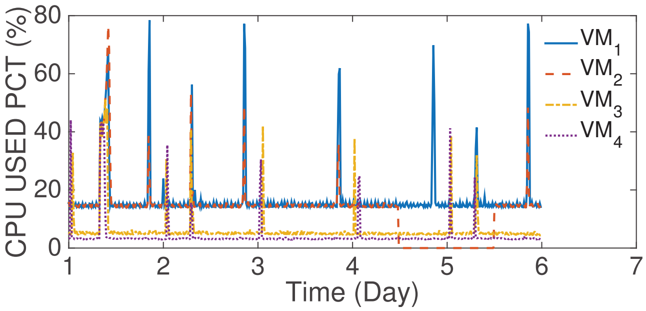


Fig. 1: CPU usage series of four co-located VMs within the same box. The traces show strong spatial dependency across six days. Here VM_2 has gaps in its observations from points 4.5 to 5.5 in the x-axis, essentially missing data for nearly 24 hours.

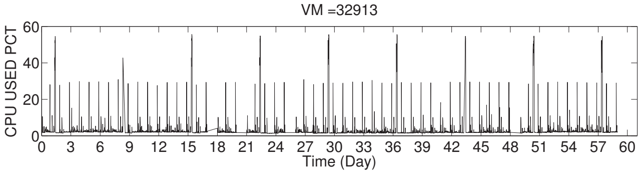


Fig. 2: CPU usage series of a certain VM in a cloud data center exhibits strong temporal dependency for more than two months.

Figure 1 presents the CPU usage series on six days for four VMs that are co-located on the same physical box. We notice that VM_2 does not have any CPU usage data recorded for one day period, from point 4.5 to point 5.5, possibly due a system failure. Previous works in data center management show that the resource usage series of VMs co-located on the same physical box exhibit strong *spatial* dependency [2], [6], this strong dependency between VM_1 and VM_2, as well as between VM_3 and VM_4 is visually clear in Figure 1. We propose to leverage spatial dependencies between co-located VMs to generate data to fill in the gaps using linear regression.

In addition to spatial dependency among VMs in co-located usage series, another interesting observation is that VM usage series exhibit strong *temporal* dependencies [7]. Figure 2 presents the CPU usage workload of a randomly selected VM for over two months. This usage series shows strong *daily* and *weekly* patterns, with distinct peaks and valleys. This observation intuitively suggests that it is possible to leverage *temporal* dependency of past observations within a VM usage

978-3-901882-98-2 © 2017 IFIP

series to potentially fill up missing data in the same series.

In this paper, we propose a model to fill up missing data in time series based on the spatial and temporal dependencies across and within different time series. We first conduct a detailed, workload characterization study on VM CPU usage series in IBM production data centers corresponding to 80K VMs hosted on over 6K physical servers and discover the statistical characteristics of data gaps in VM usage series. We then develop a spatial-temporal model that fills in the gaps based on the characteristics of the series. Our evaluation results show that the proposed spatial-temporal model achieves an average of 20% absolute percentage errors, and can be efficiently integrated with customized resource management policies.

The outline of this work is as follows. Section II provides a characterization study on the VM CPU usage series as well as the spatial and temporal dependencies across/within co-located VMs in the IBM data centers. We propose spatial-temporal filling-up methods for resource usage series in Section III. In Section IV, we evaluate the effectiveness of the proposed model. Section V presents related work, followed by summary and conclusion in Section VI.

II. Characteristics of VM workloads

We first perform statistical analysis on a real-world trace collected from IBM production data centers that serve diverse industries, including banking, pharmaceutical, IT, consulting, and retail. The trace includes CPU usage data from over 80K VMs (mostly VMware VMs) hosted on 6K physical boxes. On average, 10 VMs reside within one physical box. Each data point on the trace data corresponds to resource usage averages within a widow of 15 minutes. Ideally, there should be 96 data points per VM per day within the data, yielding to over 54 million data points across all VMs. In reality, there are several missing data points, this is either because the tracing mechanism or the recording of usage data periodically fails, which results in several “holes" in the time series. The phenomenon of missing data is quite common in other environments, including social science data [4], [8], HPC systems [1], [9], and medical data [10], [11]. For the specific problem in hand, the observed holes in the VM time series significantly impede the effectiveness data analysis and the development of methodologies for better data center management [6], [2]. Our thesis is that provided certain temporal and spatial trace characteristics, it may possible to accurately recreate the missing data with minimal errors.

A. Missing Data in the Wild

To begin, we present an overview of the prevalence of missing data in the IBM trace. Figure 3(a) shows the histogram of physical boxes with missing data in the VMs that they host. For each box, we calculate the percentage of VMs with missing data. We then partition boxes into 10 bins according to the percentage of VMs with missing data (see x-axis). The y-axis represents the percentage of boxes in each bin. The figure shows that more than 50% of boxes have more than 10% of their co-located VMs with missing data. At the most extreme case, for nearly over 20% of boxes, almost *all* of their co-located VMs have missing data (see bin $[0.9,1]$). Such observation shows clear evidence of the prevalence of missing data in the wild.

Next, it is natural to focus on the boxes that contain VMs with missing data and study the severity of such data gaps. Recall that CPU usage is collected every 15 minutes, this results in 96 data points per VM per day. Figure 3(b) illustrates the histogram of the percentage of boxes that have missing time windows. The x-axis shows the average number of missing data per VM on each box (organized in seven bins) while the y-axis shows the percentage of boxes in each bin. The figure confirms that more than 70% of boxes fall into bins $[32,64)$ and $[64,96]$, indicating that the VMs residing in these boxes experience more than 8 hours of missing usage logs. Such critical lack of data impedes the usefulness of data analysis of the VM usage data series.

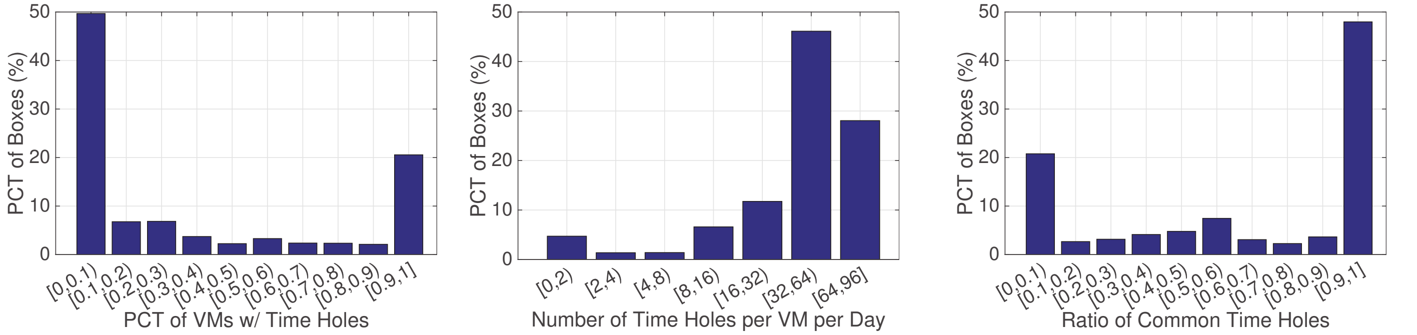
Since our intention is to use the data series of complete VM time series to recreate the data of the VM time series with missing data, we look into co-located VMs and at the percentage of common missing holes, see Figure 3(c). The x-axis represents the percentage of common time holes across all consolidated VMs in the same physical box, the y-axis gives the percentage of boxes in each bin. Note that for 20% of the boxes (see bin $[0,0.1)$), their missing time holes are quite spread across different time windows. This suggests that it is possible to exploit the similarity of data series of VMs in the same box to fill up missing data by exploiting potential spatial dependency of different VMs residing on the same box. On the contrary, for boxes in bin $[0.9,1]$ (around 50% of boxes), the majority of missing time holes occur at the same time among the VMs in the same box. Any spatial model on those boxes is not feasible. For such boxes, we look beyond spatial dependency and more specifically into possible temporal patterns.

B. Spatial Dependency Analysis

To quantify spatial dependency, we use the concept of correlation coefficient [12]. For each pair of co-located VMs on the same box, we calculate the Pearson correlation [12] of their CPU usage series. For a physical box with M VMs, there are $\frac{M\times(M-1)}{2}$ coefficient values. We use the mean and 90%ile values to represent to spatial dependency of each box and show the cumulative distribution function (CDF) of boxes for these two measures in Figure 4. We observe that co-located VM CPU usage series exhibit strong spatial dependency. The mean values of mean and 90%ile cross-correlation are 0.31 and 0.56, respectively. The distance between the two CDFs as well as their shapes show clearly that there is strong spatial dependency among VMs, as there is a significant percentage of VM pairs with high correlation coefficients as shown by the 90%ile graph and naturally fewer pairs that have lower coefficients, which contribute to reducing the mean.

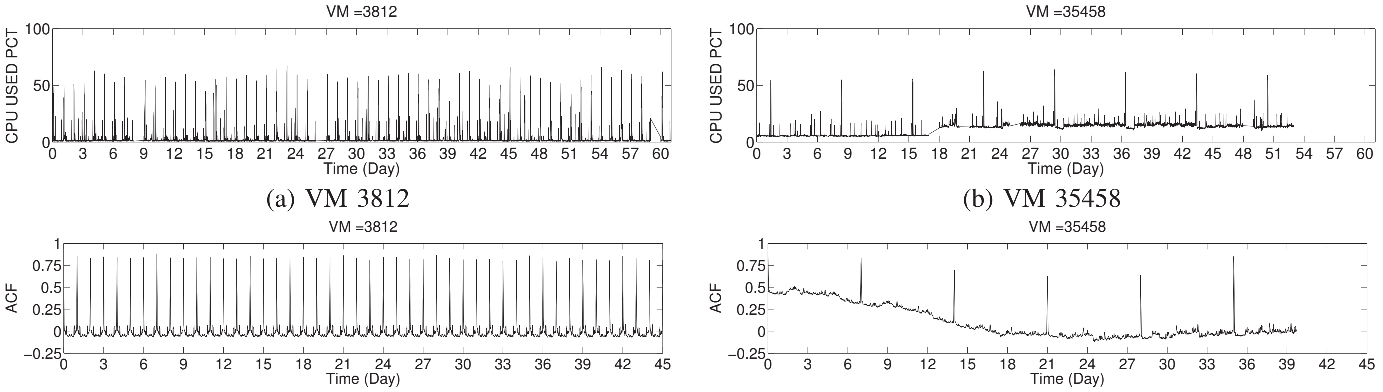
C. Temporal Dependency Analysis

Figure 3(c) illustrates that the missing data of around 50% of the boxes cannot be filled by spatial models. This leads us to explore whether temporal dependency within the time series of the same VM can be used as an alternative. We select two representative VMs and show their daily CPU usage over two months, see Figure 5(a) and 5(b). The graphs show



(a) PCT of VMs with holes per box (b) Length of holes per VM (c) Ratio of same holes for co-located VMs

Fig. 3: Overview of missing data in the VM CPU usage series across 6K boxes.



(a) VM 3812

(b) VM 35458

(c) Autocorrelation function of CPU Usage of VM 3812

(d) Autocorrelation function of CPU Usage of VM 35458

Fig. 5: CPU utilization over time for two representative VMs in (a) and (b), with their autocorrelation functions presented in (c) and (d), respectively.

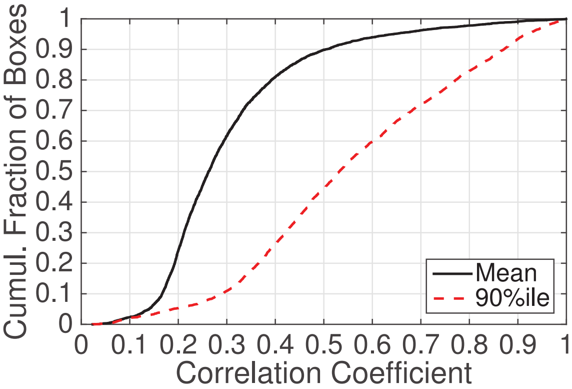


Fig. 4: CDFs of cross-correlation among co-located VM CPU usage series in terms of mean and 90%ile.

obvious periodical patterns, i.e., the CPU usage repeats every day and week for VM 3812 and VM 35458 (see Figure 5(a) and 5(b), respectively). We quantify this periodicity using the autocorrelation function. Autocorrelation shows the degree of similarity between a time series and its lagged version. It is commonly used to uncover periodical patterns in the time series [12]. The range of autocorrelation values is $[-1,1]$. High positive values imply strong similarity, while negative values indicate diametrical differences. Zero values mean no repeating patterns present in the time series. From Figures 5(c) and 5(d), we notice that the autocorrelation values peak either on a daily-basis or on a weekly-basis, which are in accordance with observations in Figures 5(a) and 5(b).

To summarize, the missing data phenomenon is commonly observed. Fortunately, we find that strong spatial and temporal dependencies is also present in the data trace, and can be exploited to compensate for the missing data.

III. Methodology

In this section, we propose a new prediction methodology that fills up the missing data in a usage trace, leveraging on the *spatial* and *temporal* dependency within/across co-located VMs in the same box. The high level description is as follows: given a set of VMs with missing data in their usage logs, we first calculate their spatial and temporal dependency levels, and then leverage either spatial or temporal models to fill up the missing data. The workflow of the proposed filling-up method is presented in Figure 6.

A. Step 1 - Dependency Comparison

As illustrated in Section II, usage series in data centers exhibit *spatial* dependency among co-located VMs in the same

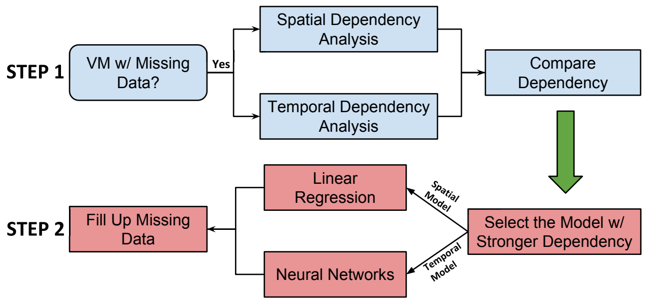


Fig. 6: Steps to fill up the missing data in usage series.

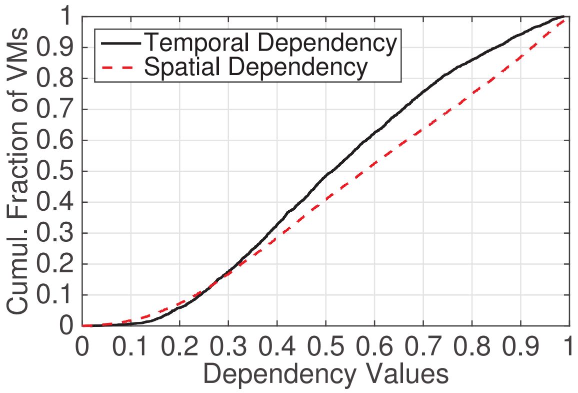


Fig. 7: CDFs of temporal and spatial dependencies across all VMs with missing data.

box, and *temporal* dependency within themselves across time. As a first step, we need to quantify the spatial and temporal dependencies of each VM usage series with missing data.

1) Quantification of Spatial Dependency: To measure spatial dependency, we use cross-correlation. For each box and each VM with missing data, we first compute the pairwise correlation coefficients [12], of the target VM with all other VM usage series in the same box that do not have the same periods of missing data. For a target VM i with missing data, there are at most $(M-1)$ pairs $\rho_{i,l}$, where M is the number of co-located VMs in the same box, and l is the index of a co-located VM with $l \neq i$. To find the most correlated VM with the target VM i, we select VM k with the largest absolute correlation coefficient $\rho_{i,k}$. $\rho_{i,k}$ expresses the strongest spatial dependency that we observe for VM k and the target VM i in the box. VM i will be presented by the usage series of VM k using linear regression.

2) Quantification of Temporal Dependency: As shown in Section II, it is possible that all VMs in the same box experience the same gaps in their time series, In such cases, temporal models could instead be used to fill up the missing data. In the following, we propose a method to quantify the predictive capability of the series using temporal models. We define a metric called goodness of temporal dependency (GTD):

$$GTD = \alpha * max\{ACF_{short}\} + (1-\alpha) * max\{ACF_{long}\}. \quad (1)$$

Here ACF_{short} is a list of autocorrelation coefficients for lags that correspond to time stretches of less than 1 day, and ACF_{long} consists of autocorrelation coefficients for lags that are more than 1 day. We capture the highest autocorrelations both for short- and long-term, i.e., $max\{ACF_{short}\}$ and $max\{ACF_{long}\}$, respectively. The term $\alpha \in [0,\ 1]$, is a weight that represents how important is the short-term behavior for filling up missing data in the usage series. This is inversely related to the prediction length, defined as:

$$\alpha = 1 - min\{\frac{Length\ of\ Missing\ Data}{Number\ of\ Observations\ per\ Day},\ 1\}. \quad (2)$$

The higher the GDT, the stronger the temporal dependency of the usage series is.

Having proposed measures that quantify spatial and temporal dependencies, it is natural to ask which dependency is stronger in the trace. We carry out statistical analysis across all VMs suffering from missing data, and present the CDFs of their spatial and temporal dependencies in Figure 7. The figure shows that more than 60% of VMs exhibit strong dependencies, as their spatial or temporal dependency values are greater than 0.4. This again confirms opportunities to fill up the missing data of VM usage series. It is also worth mentioning that across all VMs, around 85% of them have stronger spatial dependencies than temporal ones. This suggests that leveraging spatial models may be more effective than temporal ones. We note also that since spatial models are based on linear regression, they are much cheaper than temporal models that are based on more expensive neural networks.

B. Step 2 - Model Selection

Here, we provide technical details on the two prediction models.

1) Spatial Models: Given a pair of highly spatially correlated VMs, we use linear regression [13] to predict the missing data. Specifically, for the usage series $\mathbf{D}_i$ of VM i with missing data, let the co-located VM k be the one with the strongest correlation with VM i. Recall VMs k and i cannot have the same periods of missing data. For statistical significance, when the cross-correlation between VMs i and k is computed, there need to be at least 28 usage observations in the same time periods [13].

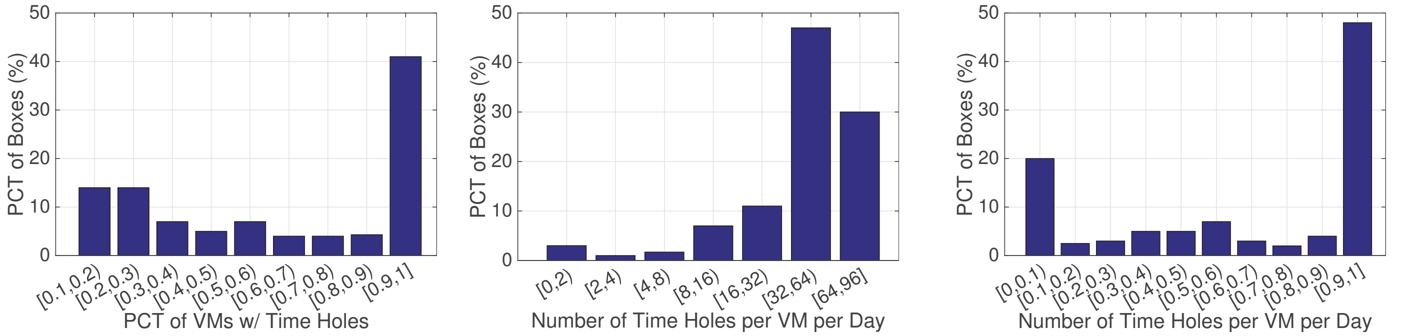
We express the usage series $\mathbf{D}_i$ of VM i by a linear regression model of the usage series $\mathbf{D}_k$ of VM k:

$$\mathbf{D}_i = a_{i,k} \times \mathbf{D}_j + b_{i,k}. \quad (3)$$

To calculate the coefficients $a_{i,k}$ and $b_{i,k}$, we first need to obtain the intersection between the usage series of VM i and VM k, denoted as $\mathbf{D}_i^k$ and $\mathbf{D}_k^i$ respectively, logged in the same time periods for both VMs. We train a linear regression model with $\mathbf{D}_i^k$ as target variable and $\mathbf{D}_k^i$ as predictor variable, to compute $a_{i,k}$ and $b_{i,k}$. Finally, to compute $\mathbf{D}_{i,t_m}$ for the missing periods $t_m \in \{t_{m1}, t_{m2}, t_{m3}, ...\}$ for VM i, we insert the observed usage, e.g., $\mathbf{D}_{k,t_m}$, of VM k into the Eq.(3). We continue until the prediction for all the missing usages of VM i is complete.

2) Temporal Models: To build a temporal model of the usage series $\mathbf{D}_i$ for VM i, assume that $\mathbf{D}_i$ is the target variable for prediction and $\mathbf{D}_{i,t}$ is the value of $\mathbf{D}_i$ at time t, then the goal is to create a model of the following form:

$$\mathbf{D}_{i,t} = f(\mathbf{D}_{i,t_1}, \mathbf{D}_{i,t_2}, \mathbf{D}_{i,t_3}, \cdots, \mathbf{D}_{i,t_n}) + \varepsilon_{i,t}, \quad (4)$$



(a) PCT of VMs with holes per box

(b) Length of holes per VM

(c) Ratio of same holes for co-located VMs

Fig. 8: Selected boxes for simulations follow the same characteristics of missing data in the original trace.

where f is the function of the temporal model, $t_1 < t_2 < t_3 < \cdots < t_n < t$, and $\varepsilon_{i,t}$ is an error term. To obtain the coefficients in the function f, we first need to insert the observed periods of usage series for VM i into Eq. (4). We then apply the temporal model and compute the value of $\mathbf{D}_{i,t_m}$ for missing periods $t_m \in \{t_{m1},\, t_{m2},\, t_{m3},\, \ldots\}$ for VM i.

Traditional temporal models such as ARMA/ARIMA [14] and Holt-Winters exponential smoothing [15] are usually limited by the linear basis function and poor in predicting bursty workloads [7]. Our previous work [7] has shown that temporal models based on neural networks achieve accurate prediction of usage series in data centers, especially peak values. Here, we use neural networks as the temporal models to predict the missing data in the VM usage series. To achieve *efficient* and *accurate* prediction, selecting appropriate temporal features (e.g., $\mathbf{D}_{i,t_j}$, where $j \in [1,\, n]$ in Eq. 4) is key, as it should reliably capture periodic behavior, changing trends, and repeating patterns. To identify informative features, we resort to the correlogram (e.g., Figure 5) because autocorrelation can provide quantitative and qualitative information on the above factors. Figure 5 shows that there can be several lags with high positive autocorrelation values. This indicates that there exist several good candidate features that represent short- and long-term correlation patterns. To automate the process, we use a local maximum detection function to identify the peak points in autocorrelations and use the respective lag values as features for neural network training. In this way, different correlation ranges from short- to long-term can all be captured, which improves the efficiency and accuracy of the temporal models.

IV. Evaluation

For evaluation of the proposed model it is not possible to directly use the real trace, as there is no ground truth for missing data. To solve this, we randomly select 400 physical boxes that contain only VMs with no missing data. As a second step, we create holes in the CPU usage series of the VMs on these boxes, by deliberately removing some data points aiming to result in a new trace that strictly follows the characteristics of the real trace. Given the histograms of VMs with holes and the length of missing data across the 6K boxes (see Figure 3), we determine the selection of VMs to create the time holes in each box, and the length of the holes for each VM. Figure 8 presents histograms of the generated data from the three perspectives presented in Figure 3. The strong similarity between Figures 3 and 8 confirms that the generated data follow the characteristics of the real trace. Consequently, using the created data to evaluate the proposed spatial-temporal model is sound because the missing data characteristics are similar to the real trace and because now we have a ground truth to evaluate the effectiveness of the method.

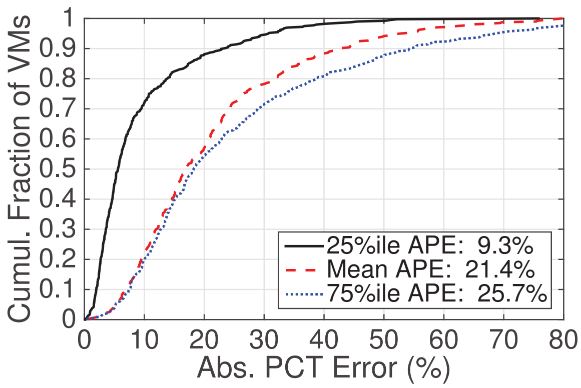


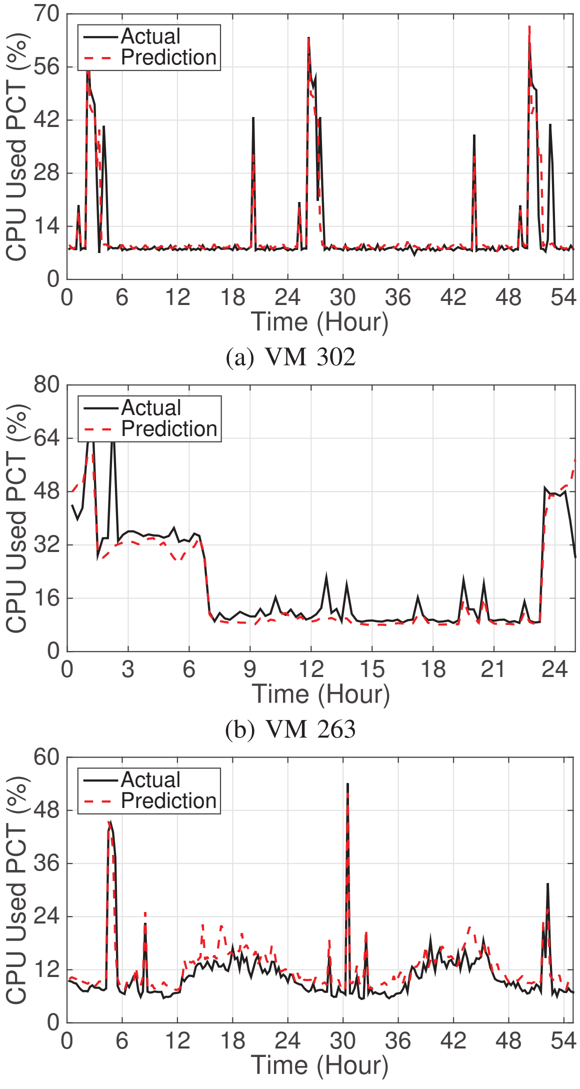
Fig. 9: Overall prediction accuracy across all tested VMs. We show the prediction errors in terms of 25%ile, mean, and 75%ile for each VM.

A. Prediction Accuracy

We use the commonly used absolute percentage error (APE), see Equation 5, to evaluate prediction accuracy. APE quantifies the relative difference between the prediction results and the actual data. Clearly, smaller APE values indicate better prediction quality.

$$APE = \frac{|Prediction - Actual|}{Actual} \times 100\% \qquad (5)$$

1) Prediction overview: For every tested VM, there is a list of time windows with missing CPU usage data, and the prediction of each of those missing data points corresponds to an APE value. Therefore, there is a set of APE values per VM. We use the mean APE value to represent the average prediction quality of the VM, together with the 25%ile and 75%ile values. Figure 9 shows the CDFs of the three APE representatives across all tested VMs. Looking at the mean APE line, an



(a) VM 302

(b) VM 263

(c) VM 209

Fig. 10: Representative VM usage series with the missing data filled up.

average APE of 21.4% is achieved for all VMs, suggesting good model effectiveness. The average of the 25%ile APE is the lowest (i.e., 9.3%), indicating that the model can deliver high prediction quality for VMs. Even for the 75%ile APE, we still observe a not-so-high average APE of 25.7%, which again confirms that the model is able to effectively fill up missing time holes.

2) Normal vs. Peak: Previous work [2] has illustrated that peak usages of VMs are key to resource allocation. Next, we evaluate prediction quality by separating normal and peak periods in the usage series of VMs, as accurately capturing peak usages is a lot more challenging than normal ones [7]. To demonstrate the performance of the proposed model, we randomly select three VMs with different spatial and temporal characteristics, and show their actual CPU usages along with the predicted ones with missing data filled-up, see Figure 10. We notice that mostly the prediction and ground truth are close to each other during both normal and peak periods.

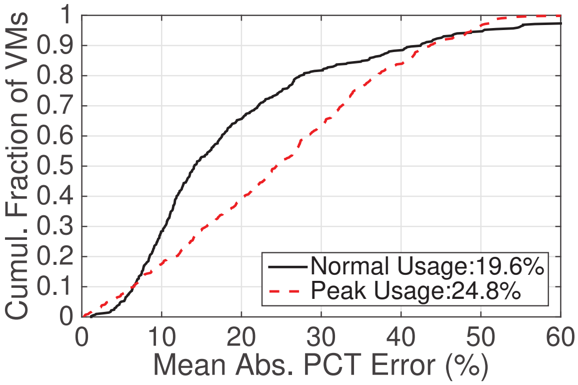


Fig. 11: Prediction accuracy for normal and peak usages across all the tested VMs. We show the mean prediction errors for each VM.

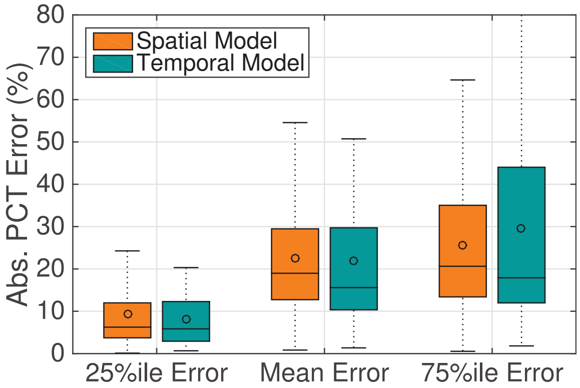


Fig. 12: Comparison on spatial and temporal models. Dots represent mean while horizontal lines represent median.

We also quantitatively compare the prediction quality for normal and peak usages. Normal and peak usages in the time series are detected using the k-means algorithm. Normal usages represent low levels of resource usage, while peak ones correspond to workload burstiness. Figure 11 shows the CDFs of the VM mean APEs for both usage types. We observe an average APE of 19.6% for normal usages, while the average APE of the more challenging peak usages is 24.8%. This last observation is especially encouraging since it demonstrates that the spatial-temporal model is effective for prediction of both usage levels.

3) Effects of various factors on accuracy: Key to selecting whether a spatial or temporal model are the relative value of their respective dependencies, see Figure 6. The model type could be one of the potential factors dominating the prediction quality. Figure 12 shows the boxplots of the values for the three APE representatives (i.e., 25%ile, mean, and 75%tile) of each VM for both spatial and temporal models. We notice that the pair-wise boxplots are quite similar to each other, indicating that prediction quality provided by either type of dependency is comparable with each other.

Next, we investigate the impact of various dependency levels, which are quantified by the correlation coefficients

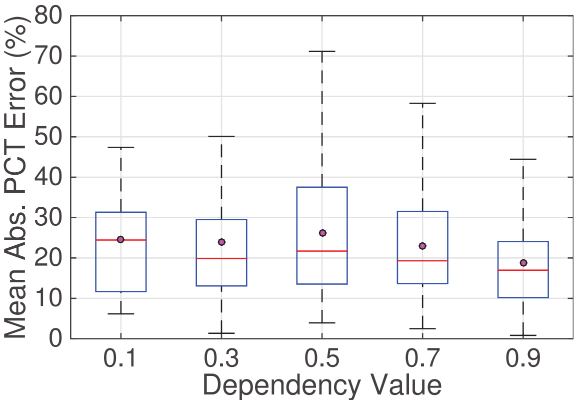


Fig. 13: Effect of various dependency level on prediction accuracy. Dots represent the mean while horizontal lines represent the median.

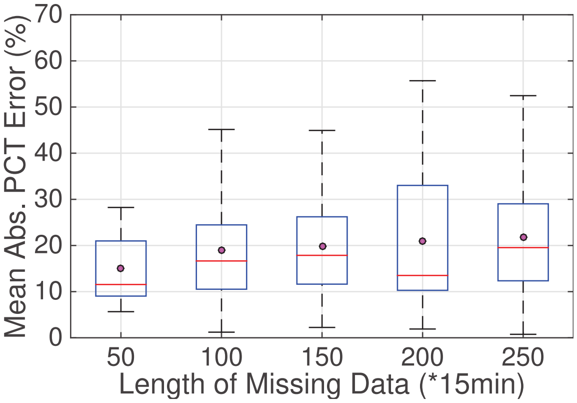


Fig. 14: Effect of the length of missing data on prediction accuracy. Dots represent the mean while horizontal lines represent the median.

for spatial dependency and GTD for temporal dependency, as described in section III. Intuitively, one may assume that the stronger the dependency, the lower the APE, thus the better the prediction quality. Figure 13 confirms this intuition by showing that the average of VM mean APE values decreases as the dependency level decreases. Even for VMs that exhibit low dependency levels (e.g., 0.1), the model is still able to exploit sufficient characteristics and results in predictions with an average error of 25%.

Next we explore the impact of the length of missing data on prediction quality, see Figure 14. We observe that the average of VM mean APE values is slightly higher for VMs with more missing data, which is understandable as longer periods of missing data introduces more difficulties to the spatial-temporal model. Even for the most difficult cases (i.e., with more than 2.5 days of missing data), the model provides quite accurate predictions (i.e., an average APE of 21%).

B. Use Case Scenario

In this section we illustrate a case study that shows the usefulness of the proposed spatial-temporal model to fill in the missing holes in VM CPU usage series for reducing performance tickets in data centers [2]. Performance ticketing systems provide the means to data centers to interactively improve user experience, maintain performance at tails, and guarantee smooth system operation. Typically, system monitoring and users issue tickets when the resource usage exceeds a predefined threshold, e.g., 60%. Ticket resolution is unfortunately very expensive [16], [17] as a significant amount of manual labor is required for root-cause analysis and to remedy the detected problem [18]. Previous work [2] has proposed a VM resizing algorithm to reduce the amount of performance tickets in data centers. Limited by the missing data problem on the trace, this resizing algorithm could only be implemented on a small fraction of boxes that had no missing data. Given the proposed data filling method, we are able to extend the feasibility of the VM resizing algorithm to all boxes.

To evaluate the efficiency of the spatial-temporal models, we use the VM resizing algorithm presented in [2] on the 400 physical boxes that have no missing data, and we can treat the result as ground truth since we do know the outcome. We introduce holes in the data trace and apply the spatial-temporal models to evaluate how effective VM resizing is. Results are presented in Table I and present the mean ticket reduction and its standard deviation. The reduction of tickets for the case of the trace with holes is 89.4%, just 7 percentage points less than the ground truth.

TABLE I: Ticket reductions across 400 boxes using the VM resizing algorithm. The original case corresponds to the trace without any missing data, while the filling-up method predicts values for the the created holes.

	Mean Ticket Reduction (%)	StD of Ticket Reduction(%)
Original	96.2	8.2
Filling-up method	89.4	12.1

We now apply the VM resizing algorithm to all boxes (6K boxes), with and without missing data. We first select all those that have no holes and apply resizing for ticket reduction and then to the entire set of boxes. For the entire set of boxes, if there are data missing, we fill the gaps using the spatial-temporal models. Results are presented in Table II and illustrate that the percentage of ticket reduction is very similar in both cases. Yet, for the case of all boxes, the sheer number of ticket reduction is significantly higher as ticket reduction is applied now to all nodes of the datacenter and is not restricted to the select few with no missing points.

TABLE II: Ticket reductions using the VM resizing algorithm across all boxes with or without missing data. Boxes with time holes are filled up by spatial-temporal models.

	Mean Ticket Reduction (%)	StD of Ticket Reduction(%)
Boxes w/o holes	95.1	10.7
All boxes	96.0	11.8

V. Related Work

A. Prediction under Missing Data

The missing data problem is common in many real-world traces and challenges researchers from diverse areas including statistics and mathematics [19], [20], social science [4], [8], data centers and HPC systems [1], [9], transportation systems [5], and medical science [10], [11]. Authors in [19], [20] summarize commonly used missing data estimation methods, i.e., Expectation-Maximization algorithm and Bayesian multiple imputation. In addition to those general methods to mitigate missing data challenges, researchers also design customized solutions based on the characteristics of their data traces. Ma et al. [4] predict missing data by leveraging user-based and item-based similarities among movie ratings and effectively improve the performance of their proposed collaborative filtering algorithms. Van Lint et al. [5] propose a specialized recurrent neural network that takes advantage of the freeway stretch lay-out and sufficiently reduce the impact of missing data on travel time prediction. In this work, we resolve the missing data problem with the help of the inherent spatial and temporal dependencies presented in the IBM data trace, and the proposed data-filling model is able to effectively and accurately predict missing time holes.

B. State-of-the-Art Time Series Prediction

As an important way to develop proactive system management polices, time series prediction and analysis have been studied extensively [21], [22]. ARIMA [14] is an effective temporal model that is able to learn the strong seasonality in time series. Livni et al. [23] take advantage of sophisticated neural network models to capture the characteristics in highly irregular time series at the expense of long training overheads. In addition, time series clustering algorithms are able to explore spatial dependency through original series (e.g., DTW [24]) or extracted features (e.g., moments [25]). The data-filling model proposed in this paper takes advantage of the low-overhead linear model derived from spatial dependencies, but also improves the predictability with the powerful neural network model that exploits temporal dependencies [7].

VI. Conclusion

In this paper, we focus on the commonly observed missing data phenomenon in real-world data traces. We study the CPU usage series of IBM data centers and discover strong spatial and temporal dependencies. We design ways of quantifying the strength of spatial and temporal dependencies in the VM CPU usage series, and propose a data-filling method to predict the missing data. We show that the spatial-temporal model is able to reach accurate predictions with around 20% absolute percentage error on the average, and is able to achieve reasonable performance even under challenging situations, such as boxes with low dependency and usage series with long lengths of missing data. Meanwhile, the model can be also efficiently integrated with customized resource management policies, such as ticketing systems in the IBM data centers.

Acknowledgment

This work is supported by NSF grant CCF-1649087. We thank our colleagues at IBM Research Zurich Lab for providing us with the data center traces.

References

[1] B. Nie, D. Tiwari, S. Gupta, E. Smirni, and J. H. Rogers, "A large-scale study of soft-errors on gpus in the field," in *High Performance Computer Architecture (HPCA), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 519–530.

[2] J. Xue, R. Birke, L. Y. Chen, and E. Smirni, "Managing data center tickets: Prediction and active sizing," in *DSN*, 2016.

[3] B. Schroeder, R. Lagisetty, and A. Merchant, "Flash reliability in production: The expected and the unexpected," in *14th USENIX Conference on File and Storage Technologies, FAST 2016, Santa Clara, CA, USA, February 22-25, 2016.*, 2016, pp. 67–80. [Online]. Available: https://www.usenix.org/conference/fast16/technical-sessions/presentation/schroeder

[4] H. Ma, I. King, and M. R. Lyu, "Effective missing data prediction for collaborative filtering," in *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2007, pp. 39–46.

[5] J. Van Lint, S. Hoogendoorn, and H. J. van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 5, pp. 347–369, 2005.

[6] J. Xue, R. Birke, L. Y. Chen, and E. Smirni, "Tale of tails: Anomaly avoidance in data centers," in *SRDS*, 2016.

[7] J. Xue, F. Yan, R. Birke, L. Y. Chen, T. Scherer, and E. Smirni, "PRACTISE: robust prediction of data center time series," in *CNSM*, 2015.

[8] P. D. Allison, "Missing data: Quantitative applications in the social sciences," *British Journal of Mathematical and Statistical Psychology*, vol. 55, no. 1, pp. 193–196, 2002.

[9] D. Tiwari, S. Gupta, J. Rogers, D. Maxwell, P. Rech, S. Vazhkudai, D. Oliveira, D. Londo, N. DeBardeleben, P. Navaux *et al.*, "Understanding gpu errors on large-scale hpc systems and the implications for system design and operation," in *High Performance Computer Architecture (HPCA), 2015 IEEE 21st International Symposium on*. IEEE, 2015, pp. 331–342.

[10] S. R. Browning and B. L. Browning, "Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering," *The American Journal of Human Genetics*, vol. 81, no. 5, pp. 1084–1097, 2007.

[11] J. A. Sterne, I. R. White, J. B. Carlin, M. Spratt, P. Royston, M. G. Kenward, A. M. Wood, and J. R. Carpenter, "Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls," *Bmj*, vol. 338, p. b2393, 2009.

[12] L. M. Leemis and S. K. Park, *Discrete-event simulation: a first course*. Pearson Prentice Hall Upper Saddle River, NJ, 2006.

[13] J. Neter, M. H. Kutner, C. J. Nachtsheim, and W. Wasserman, *Applied linear statistical models*. Irwin Chicago, 1996, vol. 4.

[14] C. Chatfield, *The analysis of time series: an introduction*. CRC press, 2013.

[15] P. Goodwin, "The holt-winters approach to exponential smoothing: 50 years old and going strong," *Foresight*, 2010.

[16] Y. Liang, Y. Zhang, M. Jette, A. Sivasubramaniam, and R. Sahoo, "Bluegene/l failure analysis and prediction models," in *DSN*, 2006.

[17] I. Giurgiu, J. Bogojeska, S. Nikolaiev, G. Stark, and D. Wiesmann, "Analysis of labor efforts and their impact factors to solve server incidents in datacenters," in *CCGrid*, 2014.

[18] I. Giurgiu, A.-D. Almasi, and D. Wiesmann, "Do you know how to configure your enterprise relational database to reduce incidents?" in *IM*, 2015.

[19] W. Wothke, "Longitudinal and multigroup modeling with missing data." 2000.

[20] J. W. Graham, "Missing data analysis: Making it work in the real world," *Annual review of psychology*, vol. 60, pp. 549–576, 2009.

[21] N. Tran and D. A. Reed, "Automatic ARIMA time series modeling for adaptive I/O prefetching," *IEEE Transactions on Parallel Distributed Systems*, vol. 15, no. 4, 2004.

[22] Z. Zhuang, H. Ramachandra, C. Tran, S. Subramaniam, C. Botev, C. Xiong, and B. Sridharan, "Capacity planning and headroom analysis

for taming database replication latency: experiences with linkedin internet traffic," in *ICPE*, 2015.

[23] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the computational efficiency of training neural networks," in *NIPS*, 2014.

[24] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16, 1994.

[25] B. D. Fulcher and N. S. Jones, "Highly comparative feature-based time-series classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, 2014.