CAFE: Adaptive VDI Workload Prediction with Multi-Grained Features

Yao Zhang,[1,2] **Wen-Ping Fan,**[2] **Xuan Wu,**[1] **Hua Chen,**[2] **Bin-Yang Li,**[2] **Min-Ling Zhang**[1,3,*]
[1]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[2]VMware Information Technology (China) Ltd.
[3]Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China
{yaoz, wfan}@vmware.com, wuxuan@seu.edu.cn, {huac, binyangl}@vmware.com, zhangml@seu.edu.cn* (corr. author)

Abstract

Virtual desktop infrastructure (VDI) is a virtualization technology that hosts desktop operating system on centralized server in a data center of private or public cloud. Effective resource management is of crucial importance for VDI customers, where maintaining sufficient virtual machines helps guarantee satisfactory user experience while turning off spare virtual machines helps save running cost. Generally, existing techniques work in passive manner by either driving available capacity reactively or configuring management schedules manually. In this paper, a novel proactive resource management approach is proposed which aims to predict VDI pool workload adaptively by utilizing CoArse to Fine historical dEscriptive (CAFE) features. Specifically, aggregate session count from pool end users serves as the basis for workload measurement and predictive model induction. Extensive experiments on real VDI customers data sets clearly validate the effectiveness of multi-grained features for VDI workload prediction. Furthermore, practical insights identified in our VDI data analytics are also discussed.

Introduction

Virtual Desktop Infrastructure (VDI) is a technology that separates personal computer desktop environments from physical machines using a client-server architecture. It provides the ability to virtualize personal desktops in the data center and access them via a remote client through a display protocol. As shown in Figure 1, VDI uses a client-server architecture where all the virtual desktops (and applications) run remotely on top of the physical infrastructure in the data center. As the physical infrastructure is managed centrally, the virtual desktops can be managed centrally as well. In this way, the desktop end users can access their desktops on any devices including PC, laptop, smartphone, tablet and thin clients. According to a recent analysis report (ReportLinker 2017), VDI market is expected to reach 15.3 billion dollars by 2023.

One key feature of VDI lies in the non-persistent desktop pool in which any virtual desktop can be used by any user. The non-persistent pool is highly efficient as it can shrink or expand its size dynamically. The pool needs to understand the required pool capacity at any time in order to determine the number of desktops that should be powered on. Today in VDI systems, this is dealt with by either driving available capacity reactively or configuring management schedules manually. The passive management scheme would result in many issues in VDI resource management, such as poor user experience, unnecessary cost, time-consuming and error-prone. To enable proactive VDI resource management, it is desirable to predict the pool workload adaptively. In this way, the VDI system can power on enough desktops for sudden workload climb and shut down unnecessary desktops once the workload starts dropping in a timely manner. Therefore, adaptive VDI pool workload prediction can help generate suitable resource management schedule automatically, and thus reduce infrastructure/administration cost and improve user experience.

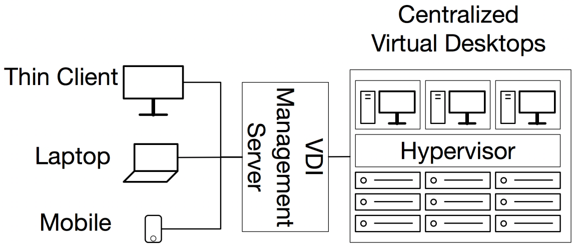


Figure 1: A typical VDI architecture.

In this paper, we introduce CAFE, an adaptive prediction model with multi-grained features which can predict VDI workload with high accuracy. Specifically, we consider the aggregate session count from pool end users which is the most evident indicator on VDI workload. For each pool, its aggregate session count is recorded once per minute unless the pool terminates. On every minute, CAFE creates multi-grained feature representation by utilizing coarse to fine historical descriptions of the workload sequence. Accordingly, the pool workload in forthcoming time span are regarded as the target for prediction.[1] Regression model for workload

[1] In this paper, workload in forthcoming 30 or 60 minutes is predicted which serves as feasible time span for adaptive VDI resource provisioning.

prediction is built by learning from examples extracted from historical sequence with multi-grained features. Comparative studies on real-world data sets clearly validate the effectiveness of multi-grained features for VDI workload prediction.

The rest of this paper is organized as follows. Section 2 gives technical details of the proposed CAFE model. Section 3 discusses related researches. Section 4 introduces the collected benchmark data sets for VDI workload prediction as well as experimental results of comparative studies. Section 5 summarizes contributions, lesson learnt and future works.

The CAFE Model

Model Structure

A VDI system is comprised of a full stack of components from underlying infrastructure to desktops and applications. Proper provisioning of resources has a high impact on the cost and performance of a VDI system so the workload knowledge is very important (Casalicchio, Iannucci, and Silvestri 2015). There are different perspectives to measure the workload of VDI system. In this paper, we use the VDI pool session count which corresponds to the basic metric of VDI workload. Pool is a logical concept which plays an important role in the VDI system with two major advantages. Firstly, it is the basic unit of desktop assignment where VDI administrator assigns a group of users to a pool. The users will either share the desktops (in non-persistent pool) or have dedicated desktops (in persistent pool). Secondly, it makes the desktop resource optimization easier where the desktops in a pool are provisioned from the same template. Thus, the non-persistent pool can expand and shrink in a fairly easy way of cloning desktops from the template and deleting unused desktops.

In the current mainstream VDI solutions, the system handles this either reactively or by scheduling. The reactive strategy uses thresholds to reactively drive the available capacity upward or downward, while the scheduling strategy requires the administrator to manually inform VDI system the expected capacity at specific times. If the session count can be accurately predicted in advance, the pool can shrink or expand its size in a more timely and precise manner. In this way, both the management cost and end user's time can be significantly saved by keeping only necessary desktop powered on and reducing waiting time for desktop login. At the same time, manual scheduling is no longer needed.

The workload is an aggregate value which reflects the global behaviors of all end users within the same pool. In CAFE, we choose to use the aggregate session count as VDI workload indicator for the following reasons. Firstly, the aggregation of a large number of individual variables can help mitigate the impact of noise statistically. Therefore, the pattern of aggregate workload data can be better fitted by the learning model, where similar characteristics have been observed in time-series analysis such as smart meter data (Wijaya et al. 2014; Laurinec and Lucká 2017) and tourism data (Song and Li 2008). Secondly, aggregate model is more efficient, where only one model needs to be built for one VDI pool. This will greatly help reduce the infrastructure cost for model training and deployment.

VDI workload prediction is the process of using historical workload data to predict the future workload. Formally speaking, let $\boldsymbol{x}_{t,n} = (x_t; \ldots ; x_{t-n+1})$ be the pool workload history sequence, where x_t is the pool workload value at time t (in minute) and n is the sequence length. Let $x_{t+\Delta t}$ be the workload we want to predict at time $t + \Delta t$. Our aim is to learn the predictive model $f_{\Delta t} : \boldsymbol{x}_{t,n} \mapsto \mathbb{R}$, where the future workload $\hat{x}_{t+\Delta t}$ at time $t + \Delta t$ can be obtained as follows:

$$\hat{x}_{t+\Delta t} = f_{\Delta t}(\boldsymbol{x}_{t,n}) \tag{1}$$

Furthermore, the predictive model $f_{\Delta t}$ is instantiated as:

$$f_{\Delta t} = g(\boldsymbol{\mu}_t) \tag{2}$$

where $\boldsymbol{\mu}_t$ is the feature vector generated from $\boldsymbol{x}_{t,n}$ and $g(\cdot)$ is a regression function. Similar to (He et al. 2014), $\boldsymbol{\mu}_t$ is composed of components from different categories including *historical*, *seasonal* and *contextual* features:

$$\boldsymbol{\mu}_t = (\boldsymbol{h}_t; \boldsymbol{s}_t; \boldsymbol{c}_t) \tag{3}$$

Here, $\boldsymbol{h}_t$ is the representation of historical workload changes. In CAFE, a coarse to fine description transformation $\lambda : \boldsymbol{x}_{t,n} \mapsto \boldsymbol{h}_t$ is introduced for multi-grained representation. $\boldsymbol{s}_t$ denotes the seasonal historical features retrieved from $\boldsymbol{x}_{t,n}$ and $\boldsymbol{c}_t$ is the contextual feature vector.

Let $\boldsymbol{S} = \{x_1, x_2, \ldots , x_T\}$ be the full-length workload sequence of one VDI pool, which in turn leads to the training data set $\mathcal{D} = \{(\boldsymbol{\mu}_t, x_{t+\Delta t}) \mid l \leq t \leq T - \Delta t\}$. Here, l is the minimum feasible starting time of $\boldsymbol{x}_{t,n}$ and T is the ending time of the sequence. Based on $\mathcal{D}$, the regressor $g : \boldsymbol{\mu}_t \mapsto \mathbb{R}$ can be derived by invoking some regression learning method $\mathcal{L}$ on $\mathcal{D}$, i.e. $g \leftarrow \mathcal{L}(\mathcal{D})$.

Multi-Grained Features

Following the above explanations, $\boldsymbol{h}_t$ is defined as the data representation of pool workload. The model $\lambda : \boldsymbol{x}_{t,n} \mapsto \boldsymbol{h}_t$ describes the pool workload sequence $\boldsymbol{x}_{t,n}$ in a coarse to fine manner.

Let k_i be the length of granule, m_i be the the number of granules that are taken into account, and $\gamma_i = k_i * m_i$ be the corresponding action scope with $\gamma_i \leq n$. The two vectors $\boldsymbol{k} = (k_1, \ldots , k_\alpha)$ and $\boldsymbol{\gamma} = (\gamma_1, \ldots , \gamma_\alpha)$ specify the granularities and corresponding action scopes in multi-grained representation, where the feature vector $\boldsymbol{h}_t$ is generated from $\boldsymbol{x}_{t,n}$ as follows:

$$\begin{aligned} \boldsymbol{h}_t &= \lambda(\boldsymbol{x}_{t,n}, \boldsymbol{k}, \boldsymbol{\gamma}) \\ &= (\boldsymbol{L}_1, \boldsymbol{L}_2, \ldots , \boldsymbol{L}_\alpha)^T \boldsymbol{x}_{t,n}, \\ &\text{where } \boldsymbol{L}_i = [l^i_{pq}]_{n\times m_i},\ m_i = \frac{\gamma_i}{k_i}. \end{aligned} \tag{4}$$

where $\boldsymbol{L}_i = [l^i_{pq}]_{n\times m_i}$ denotes the i_{th} grain-layer transformation performed on the input sequence $\boldsymbol{x}_{t,n}$, with granularity k_i and action scope γ_i. Specifically, we utilize mean aggregation with elements of $\boldsymbol{L}_i$ specified as follows.

$$l^i_{pq} = \begin{cases} \frac{1}{k_i}, & (q-1)k_i + 1 \leq p \leq qk_i, \\ & 1 \leq q \leq m_i. \\ 0, & otherwise. \end{cases} \tag{5}$$

Table 1: $\boldsymbol{k}$ and $\boldsymbol{\gamma}$ in coarse-granularity description

i	1	2	3	4
k_i	n	$\frac{n}{2}$	$\frac{n}{6}$	$\frac{n}{8}$
γ_i	n	$\frac{n}{2}$	$\frac{n}{6}$	$\frac{n}{8}$

Table 2: $\boldsymbol{k}$ and $\boldsymbol{\gamma}$ in fine-granularity description

i	5	6	7
k_i	$\frac{\Delta t}{3}$	$\frac{\Delta t}{15}$	1
γ_i	$2\Delta t$	Δt	1

To instantiate the multi-grained representation of pool workload sequence, we employ seven coarse to fine grain-layer transformations ($\alpha = 7$).

For coarse-granularity description, CAFE applies multiple hour-level aggregations. Here, we aim to retrieve the global trend and filter out minor fluctuation, avoiding unnecessary noise for subsequent regression model learning. Accordingly, we set $k_i = \gamma_i$ in coarse-grained description so that the workload trend is reflected by the global average obtained from the whole action scope. Configurations of k_i and γ_i for coarse-granularity workload description are summarized in Table 1.

For fine-granularity description, CAFE applies multiple minute-level aggregations. Intuitively, for the more recent period, we prefer to describe the pool workload in a finer granularity such that the prediction could catch up with the sudden change in workload. Hence, the value of granularity k_i decreases when a smaller scope scope γ_i is selected. Here, γ_i is determined by the prediction time space Δt ($\Delta t \geq 30$), Configurations of k_i and γ_i for coarse-granularity workload description are summarized in Table 2.

Besides the coarse to fine historical description, CAFE also makes use of the long-term seasonal features and contextual information. For instance, the day and week seasonal pattern can be clearly observed in Figure 2. In fact, the pool workload is not solely reflected by the value presented in last day or last week. Taking customers from education industry as an example, the workload of VDI supplying teaching software fluctuates in a cycle of several weeks, and the cycle length is variant as determined by the course schedule. For this reason, we extend the daily and weekly seasonal features to d days and w weeks[2]. Let s_t be the workload seasonal value on time t, the seasonal feature vector $\boldsymbol{s}_t$ is specified as:

$$\begin{aligned}\boldsymbol{s}_t = (&s_{t-1440*1}; \dots, s_{t-1440*d};\\ &s_{t-1440*7*1}; \dots; s_{t-1440*7*w}),\\ \text{where } s_a = &\frac{\sum_{i=1}^{\Delta t} x_{a+i}}{\Delta t}.\end{aligned} \tag{6}$$

Contextual features is used to describe the contextual information along with the workload sequence. For CAFE, we choose to select two static features representing the day ($\{1, 2, \dots, 1440\}$) and the week ($\{1, 2, \dots, 7\}$) which the current time (minute) belongs to:

$$\boldsymbol{c}_t = (minute\_of\_the\_day, day\_of\_the\_week). \tag{7}$$

Related Work

To the best of our knowledge, the CAFE approach presents as the first dedicated attempt towards adaptive VDI workload prediction. As per the definitions given in (Esling and Agon 2012) and (Tan et al. 2014), VDI workload prediction can be regarded as a predictive analysis task for time-series data. Holt-Winters (Chatfield 1978; Chatfield and Yar 1988) is one of the most commonly-used statistical model for seasonal time-series data prediction. In (Burkom, S, and Shmueli.G 2007), it is shown that Holt-Winters outperforms non-adaptive regression and adaptive regression in forecasting syndromic data streams of biosurveillance. Facebook introduced Prophet in (Taylor and Letham 2017), a large-scale time-series data prediction algorithm based on additive model where non-linear trends are fit with seasonalities. In recent years, stream data (continuous time-series data) data analysis based on ensemble learning has also received significant attentions. Due to the merits of robustness, ease of parallelization and incremental model training (Crmanová et al. 2016), ensemble learning shows significant advantages to deal with the high volume, high velocity and non-stationary characteristics of stream data (Krawczyk et al. 2017).

From industrial perspective, real-world modeling tasks similar to VDI workload prediction include electricity consumption forecasting (Hernandez et al. 2014), tourism demand forecasting (Burger et al. 2001) and finance modeling (Taylor 1986). For instance, smart grid data is collected from the individual smart meter of resident and aggregated to be the overall electricity consumption at transmission or sub-transmission system level (Hernandez et al. 2014). According to (Wijaya et al. 2014), aggregate forecasting can lead to better predictive accuracy than disaggregate forecasting as aggregated data can neutralize the noises of individual data. Furthermore, clustering based forecasting methods have been discussed in several literatures (Tidemann et al. 2013; Laurinec and Lucká 2016; 2017; Liao 2003). For instance, model-based representation of smart grid time-series data are generated for k-means clustering. Centroid of each cluster is used for predictive model training, where experimental results showed notable accuracy improvement over the disaggregate model (Laurinec and Lucká 2017).

Experiments

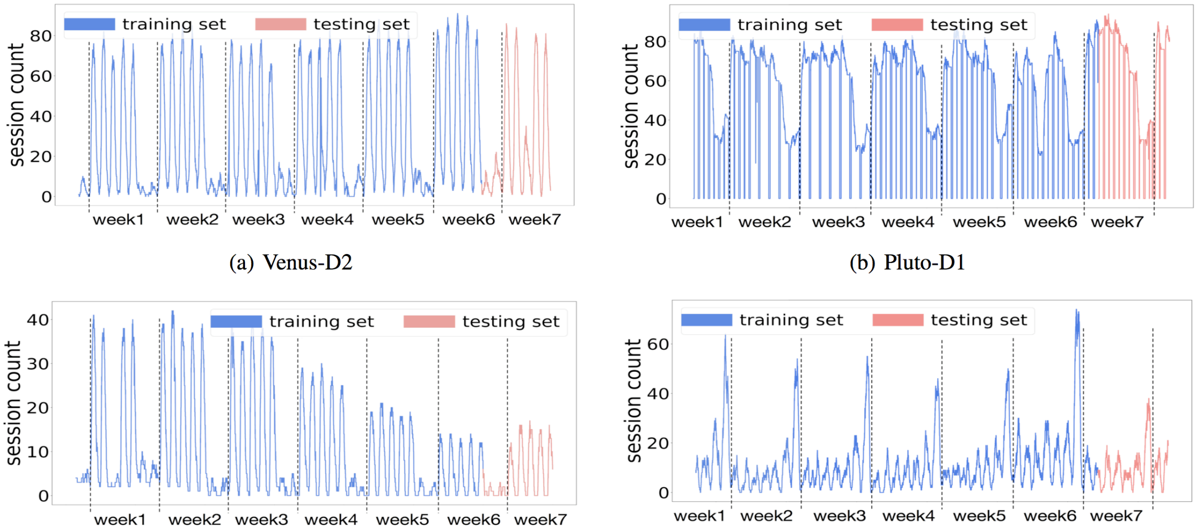
Experimental Setup

Data Sets The experimental data sets are collected from four pools of four different VDI customers over a period of 11 weeks from June 1st to August 17th, 2018. For each pool, the data is divided into two divisions. Division D1 uses data from June 1st to July 10th, 2018 as the training set and data from July 11th to July 17th, 2018 as testing set. Division D2 uses data from July 1st to August 10th, 2018 as the training set and data from August 11th to August 17th, 2018 as testing set. The eight data sets are named

[2] In this paper, we set $d = 6$ and $w = 5$.

Table 3: Characteristics of the real-world VDI data sets.

Division	Data sets	Training set			Testing set		
		Period	min-max	mean±std	Period	min-max	mean±std
D1	Venus	1/06 - 10/07, 2018	0-82	26.26±26.66	11/07 - 17/07, 2018	0-85	27.09±27.35
	Uranus		0-74	11.63±11.81		0-38	9.59±7.61
	Pluto		0-91	51.15±27.88		0-94	56.01±32.29
	Mars		0-110	17.81±19.36		0-39	12.26±13.51
D2	Venus	1/07 - 10/08, 2018	0-91	28.35±28.09	11/08 - 17/08, 2018	0-86	26.09±26.83
	Uranus		0-74	13.09±12.24		0-61	10.76±11.46
	Pluto		0-99	52.01±30.98		0-85	46.14±29.35
	Mars		0-42	9.92±11.41		0-17	4.72±5.41



(a) Venus-D2

(b) Pluto-D1

(c) Mars-D2

(d) Uranus-D1

Figure 2: Real-world VDI workload data sets.

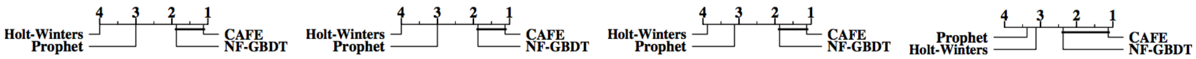
as Venus-D1, Venus-D2, Uranus-D1, Uranus-D2, Pluto-D1, Pluto-D2, Mars-D1 and Mars-D2. Table 3 shows details of the eight data sets, where min-max refers to minimum/maximum session count of VDI pool and mean±std refers to the mean value and standard deviation of session counts.

Figure 2 illustrates four pools of the data sets, where obviously different patterns can be observed. For the Uranus pool, a weekend pattern can be observed whose workload has a much higher peak on weekend (especially on Sunday) and is relatively low on working days. For the Venus, Pluto and Mars pools, a working day pattern can be observed whose workload is higher on Monday to Friday. Specifically, Venus exhibits a regular working day pattern with the session count climbing and dropping in a smoother way, Pluto is special for its regular sudden workload drop and sudden climb several times in a day, and Mars exhibits typical working day pattern while its average session count shows a long-term declining trend. Furthermore, the testing sets also has notable variance than the training set. In terms of testing set, Venus-D2 has a much lower peak on Wednesday, Uranus-D1 has a lower peak on Sunday, and Mars-D2 has lower average workload than previous weeks. Due to the diverse properties, those real-world data sets serve as a solid basis for evaluating the effectiveness of comparing approaches.

Comparing Approaches As discussed in previous section, there have been various approaches for time-series data prediction. In this paper, we compare CAFE with three approaches including Prophet (Taylor and Letham 2017), Holt-Winters (Chatfield and Yar 1988) and NF-GBDT. Prophet is good at large scale time-series forecasting with the ability of detecting seasonalities automatically. Holt-Winter algorithm is a traditional statistical method for seasonal time-series data prediction using a exponential smoothing model. NF-GBDT is the combination of naive time-series features (Laurinec and Lucká 2017) with GBDT (Friedman 2001; 2002) as regression model. Here, naive features correspond to use single granularity and action scope to generate the historical features. Therefore, NF-GBDT can be regarded as

Table 4: Performance on the real-world VDI data sets in term of NMAE, NRMSE, OPR and UPR ($\Delta t = 30$).

Comparing	NMAE							
Methods	Venus-D1	Venus-D2	Uranus-D1	Uranus-D2	Pluto-D1	Pluto-D2	Mars-D1	Mars-D2
CAFE	**0.0625**	**0.0953**	**0.1224**	**0.1043**	**0.0859**	0.0775	**0.0714**	**0.1260**
Prophet	0.2981	0.4131	0.4991	0.4030	0.2324	0.2464	0.2611	0.3331
Holt-Winters	0.8134	0.7693	0.6282	0.8191	0.4404	0.3490	1.2227	0.9367
NF-GBDT	0.0969	0.1764	0.2595	0.1864	0.1284	**0.0640**	0.1564	0.3081
Comparing	NRMSE							
Methods	Venus-D1	Venus-D2	Uranus-D1	Uranus-D2	Pluto-D1	Pluto-D2	Mars-D1	Mars-D2
CAFE	**0.0765**	**0.1189**	**0.1473**	**0.1048**	**0.1321**	0.1152	**0.0728**	**0.1332**
Prophet	0.2702	0.3704	0.4836	0.3767	0.2791	0.2791	0.2721	0.3251
Holt-Winters	0.6779	0.6869	0.7010	0.7211	0.4607	0.3696	1.0293	0.8201
NF-GBDT	0.1185	0.2268	0.2679	0.1709	0.1757	**0.0747**	0.1913	0.3100
Comparing	OPR							
Methods	Venus-D1	Venus-D2	Uranus-D1	Uranus-D2	Pluto-D1	Pluto-D2	Mars-D1	Mars-D2
CAFE	**0.0287**	**0.0699**	**0.0843**	**0.0607**	**0.0267**	**0.0211**	**0.0514**	0.0567
Prophet	0.1705	0.2902	0.3360	0.2398	0.1140	0.1232	0.2231	0.2090
Holt-Winters	0.5877	0.2045	0.5258	0.6906	0.3634	0.2873	1.1949	0.6985
NF-GBDT	0.0381	0.1287	0.2110	0.1280	0.0651	0.0350	0.0570	**0.0281**
Comparing	UPR							
Methods	Venus-D1	Venus-D2	Uranus-D1	Uranus-D2	Pluto-D1	Pluto-D2	Mars-D1	Mars-D2
CAFE	**0.0338**	**0.0255**	**0.0381**	**0.0435**	**0.0592**	0.0564	**0.0199**	**0.0693**
Prophet	0.1277	0.1229	0.1631	0.1632	0.1184	0.1232	0.0380	0.1242
Holt-Winters	0.2257	0.5648	0.1024	0.1285	0.0770	0.0617	0.0278	0.2382
NF-GBDT	0.0588	0.0477	0.0485	0.0584	0.0633	**0.0290**	0.0994	0.2800



(a) CD-NMAE-30 (b) CD-NRMSE-30 (c) CD-OPR-30 (d) CD-UPR-30

Figure 3: Comparison of CAFE (control algorithm) against other comparing approaches with the Bonferroni-Dunn test ($\Delta t = 30$). Approaches not connected with CAFE in the CD diagram are considered to have significantly different performance from the control algorithm (CD=1.5453 at 0.05 significance level).

a degenerated version of CAFE without considering multi-grained features.[3] Comparison between CAFE and NF-GBDT help prove the usefulness of multi-grained features for VDI workload prediction.

The experiments are performed with two different time spans: $\Delta t = 30$ minutes and $\Delta t = 60$ minutes. From VDI domain perspective, smaller Δt focuses more on real time changes and immediate mitigation actions whilst bigger Δt is more valuable on boot storm prevention and capacity planning. For CAFE, the maximum action scope n is 1440 (24 hours). Furthermore, the granularity vector $\boldsymbol{k}$ is $(1440, 720, 240, 180, 10, 2, 1)$ when $\Delta t = 30$ minutes and is $(1440, 720, 240, 180, 20, 4, 1)$ when $\Delta t = 60$ minutes. Correspondingly, the action scope vector $\boldsymbol{\gamma}$ is $(1440, 720, 240, 180, 60, 30, 1)$ when $\Delta t = 30$ minutes and is $(1440, 720, 240, 180, 120, 60, 1)$ when $\Delta t = 30$ minutes. For Holt-Winters, we specify the seasonal parameter as 168 (24*7 hours) due to the weekly seasonal pattern in the data. For the NF-GBDT, we use the 168-dimensional naive time-series features $(max_{t-168*60+1 \leq i \leq t-167*60}\, x_i,\, \dots,\, max_{t-60+1 \leq i \leq t}\, x_i)$, where the max operator is used as maximum workload is more important for VDI administrator.

Evaluation Metrics Four metrics are used for performance evaluation, including *Normalized Mean Absolute Error* (NMAE), *Normalized Root Mean Square Error* (NRMSE), *Over Prediction Rate* (OPR), and *Under Prediction Rate* (UPR). Let $S = (s_1, \dots, s_m)$ and $\hat{S} = (\hat{s}_1, \dots, \hat{s}_m)$ denote the ground-truth and predicted workload sequence with length m, definitions on the four metrics are as follows:

$$\mathrm{NMAE}(S, \hat{S}) = \frac{\mathrm{MAE}(S, \hat{S})}{\|S\|_1} = \frac{\sum_{t=1}^{m} |s_t - \hat{s}_t|}{\sum_{t=1}^{m} |s_t|}$$

$$\mathrm{NRMSE}(S, \hat{S}) = \frac{\mathrm{RMAE}(S, \hat{S})}{\|S\|_2} = \sqrt{\frac{\sum_{t=1}^{m} (s_t - \hat{s}_t)^2}{\sum_{t=1}^{m} {s_t}^2}}$$

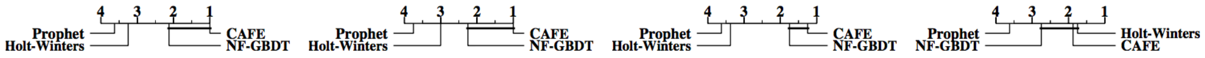
$$\mathrm{OPR}(S, \hat{S}) = \frac{\sum_{t=1}^{m} |s_t - \hat{s}_t| \times \frac{sign(\hat{s}_t - s_t) + 1}{2}}{\sum_{t=1}^{m} |s_t|}$$

$$\mathrm{UPR}(S, \hat{S}) = \frac{\sum_{t=1}^{m} |s_t - \hat{s}_t| \times \frac{sign(s_t - \hat{s}_t) + 1}{2}}{\sum_{t=1}^{m} |s_t|}$$

[3] For CAFE, GBDT is also utilized as the regression method to learn from the training examples.

Table 5: Performance on the real-world VDI data sets in term of NMAE, NRMSE, OPR and UPR ($\Delta t = 60$).

Comparing	NMAE							
Methods	Venus-D1	Venus-D2	Uranus-D1	Uranus-D2	Pluto-D1	Pluto-D2	Mars-D1	Mars-D2
CAFE	**0.0760**	**0.1589**	**0.1639**	**0.1463**	**0.1244**	**0.0885**	**0.0982**	**0.1550**
Prophet	0.2835	0.4062	0.5111	0.4074	0.2255	0.2232	0.2270	0.5351
Holt-Winters	0.1011	0.2435	0.4064	0.2426	0.1705	0.2776	0.2296	1.8100
NF-GBDT	0.1085	0.2045	0.3321	0.2187	0.1647	0.0961	0.1823	0.3767
Comparing	NRMSE							
Methods	Venus-D1	Venus-D2	Uranus-D1	Uranus-D2	Pluto-D1	Pluto-D2	Mars-D1	Mars-D2
CAFE	**0.0758**	**0.2005**	**0.1833**	**0.1558**	**0.1687**	**0.1018**	**0.0957**	**0.1583**
Prophet	0.2600	0.3716	0.4935	0.3839	0.2711	0.2728	0.2418	0.5392
Holt-Winters	0.0969	0.3053	0.4529	0.2092	0.2567	0.3655	0.2165	1.9407
NF-GBDT	0.1140	0.2635	0.3440	0.2008	0.2108	0.1053	0.2324	0.3721
Comparing	OPR							
Methods	Venus-D1	Venus-D2	Uranus-D1	Uranus-D2	Pluto-D1	Pluto-D2	Mars-D1	Mars-D2
CAFE	**0.0305**	**0.1101**	**0.1237**	**0.0864**	**0.0396**	**0.0330**	0.0686	0.0679
Prophet	0.1568	0.3024	0.3820	0.2626	0.0881	0.1274	0.1748	0.4378
Holt-Winters	0.0654	0.1608	0.3310	0.1987	0.1127	0.2448	0.1425	1.7545
NF-GBDT	0.0441	0.1508	0.2821	0.1368	0.0802	0.0469	**0.0438**	**0.0611**
Comparing	UPR							
Methods	Venus-D1	Venus-D2	Uranus-D1	Uranus-D2	Pluto-D1	Pluto-D2	Mars-D1	Mars-D2
CAFE	0.0455	**0.0489**	**0.0402**	0.0599	0.0848	0.0555	**0.0296**	0.0871
Prophet	0.1267	0.1038	0.1291	0.1448	0.1374	0.0958	0.0522	0.0973
Holt-Winters	**0.0357**	0.0827	0.0754	**0.0439**	**0.0578**	**0.0329**	0.0872	**0.0554**
NF-GBDT	0.0644	0.0537	0.0499	0.0819	0.0844	0.0492	0.1385	0.3156



(a) CD-NMAE-60 (b) CD-NRMSE-60 (c) CD-OPR-60 (d) CD-UPR-60

Figure 4: Comparison of CAFE (control algorithm) against other comparing approaches with the Bonferroni-Dunn test ($\Delta t = 60$). Approaches not connected with CAFE in the CD diagram are considered to have significantly different performance from the control algorithm (CD=1.5453 at 0.05 significance level).

Here, $sign(\cdot)$ corresponds to the signed function.

Generally, NMAE and NRMSE are better choices than MAE, RMSE since they are robust to the scale of predicted values. OPR measures the rate that the predicted value is larger than the ground-truth value, which indicates the case that more desktops are powered on than necessary. UPR measures the rate that the predicted value is smaller than the ground-truth value, which indicates the case that desktops powered on are not enough for usage and some users will have to wait for extra desktops resources. In real-world deployment, both OPR and UPR should be considered and well balanced according to customers' requirements.

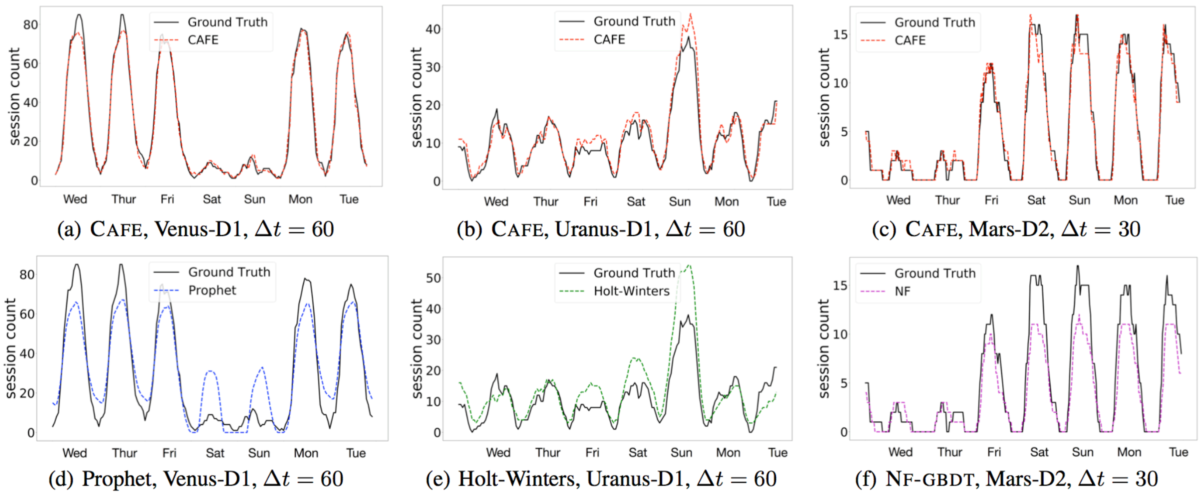
Experimental Results

Detailed experimental results are reported on Table 4 and Table 5 respectively, where the best performance on each data set is shown in boldface. To analyze the relative performance among the comparing approaches in a systematic manner, *Friedman test* (Demšar 2006) is employed as the statistical test for performance comparison.

Table 6 and Table 7 report the Friedman statistics F_F and the corresponding critical values in terms of each evaluation metric for $\Delta t = 30$ and $\Delta t = 60$ respectively. It is obvious that the null hypothesis of equal performance is rejected at 0.05 significance level. Accordingly, post-hoc Bonferroni-Dunn test (Dunn 1961) is performed to compare the relative performance among the comparing approaches. The CD diagrams are presented in Figure 3 and Figure 4 for $\Delta t = 30$ and $\Delta t = 60$ respectively, where the average rank of each approach is marked along the axis (the smaller the better).

Based on the reported experiment results, the following observations can be made:

- Among the 64 configurations (16 data sets × 4 evaluation metrics), CAFE ranks 1st and 2nd in 82.8% and 14.1% cases respectively. Specifically, for the shorter time span prediction ($\Delta t = 30$), CAFE ranks 1st and 2nd in 87.5% and 12.5% cases respectively. For the longer time span prediction($\Delta t = 60$), CAFE ranks 1st in 78.1% cases and ranks 2nd in 15.6% cases, and achieves best performance against all comparing approaches in terms of NMAE and NRMSE.
- It is remarkable that CAFE achieves the lowest average rank in terms of all evaluation metrics, except on UPR with $\Delta t = 60$.



(a) CAFE, Venus-D1, $\Delta t = 60$ (b) CAFE, Uranus-D1, $\Delta t = 60$ (c) CAFE, Mars-D2, $\Delta t = 30$

(d) Prophet, Venus-D1, $\Delta t = 60$ (e) Holt-Winters, Uranus-D1, $\Delta t = 60$ (f) NF-GBDT, Mars-D2, $\Delta t = 30$

Figure 5: Illustrative prediction results of CAFE and comparing approaches on several data sets over one week.

Table 6: Friedman statistics F_F in terms of each evaluation metric and the critical value at 0.05 significance level (# comparing algorithms $k = 4$, # data sets $N = 8$, $\Delta t = 30$).

Evaluation metric	F_F	critical value
NMAE	153.0000	3.0725
NRMSE	153.0000	
OPR	73.0000	
UPR	11.0645	

Table 7: Friedman statistics F_F in terms of each evaluation metric and the critical value at 0.05 significance level (# comparing algorithms $k = 4$, # data sets $N = 8$, $\Delta t = 60$).

Evaluation metric	F_F	critical value
NMAE	37.8000	3.0725
NRMSE	33.0000	
OPR	34.4815	
UPR	5.8736	

- It is worth noting that CAFE demonstrates desirable generalization performance on data sets with different properties: a) On the `Venus-D1` data set (Figure 5(a) and 5(d)), CAFE achieves much better generalization performance against Prophet in predicting the workload of Saturday and Sunday; b) On the `Uranus-D1` data set (Figure 5(b) and 5(e)), CAFE fits very well on Sunday whose workload peak does not conform to normal weekly pattern, while Holt-Winters severely over-predicts the workload on Sunday; c) On the `Mars-D2` data set (Figure 5(c) and 5(f)), CAFE shows its capability of adapting to both long term trend and short term change, while NF-GBDT severely under-predicts the workload from Saturday to Tuesday.

Conclusion

In this paper, an adaptive learning model named CAFE is proposed for predicting VDI pool workload based on multi-grained features. CAFE generates coarse to fine historical features from raw workload time-series data, which are fed to the GBDT regressor for model induction together with useful seasonal and contextual features. Extensive experimental studies show that CAFE achieves superior performance against comparing approaches and demonstrates desirable performance on data sets with varying properties.

The following practical insights have been identified in our design and evaluation of the CAFE approach: 1) For VDI data analytics, building aggregation model by considering the global behavior of all end users in the pool is feasible, which can help ease model training and reduce the deployment cost; 2) Feature engineering is rather important where specific domain knowledge can help understand the underlying regularities and design better feature representation than naive time-series features; 3) Long term pattern and short term characteristics need to be well balanced in feature extraction, where multi-grained features do benefit the generalization ability of induced predictive model.

In the future, it is also interesting to explore the possibilities of designing customized predictive model for individual end user. Furthermore, in VDI production environment, it is desirable to enable the predictive model with the ability of online training and testing.

Acknowledgement

This work was completed while Xuan Wu is conducting her internship at VMware Information Technology (China) Ltd. The authors wish to thank the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Key R&D Program of China (2018YFB1004300), the National Science Foundation of China (61573104), the Fundamental Research Funds for the

Central Universities (2242018K40082), and partially supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

Burger, C.; Dohnalb, M.; Kathradab, M.; and Lawc, R. 2001. A practitioners guide to time-series methods for tourism demand forecasting - a case study of durban, south africa. *Tourism Management* 22:403–409.

Burkom, H.; S, M.; and Shmueli.G. 2007. Automated time series forecasting for biosurveillance. *Statistics in Medicine* 26(22):4202–4218.

Casalicchio, E.; Iannucci, S.; and Silvestri, L. 2015. Cloud desktop workload: A characterization study. In *Proceedings of the 3rd IEEE International Conference on Cloud Engineering*, 66–75.

Chatfield, C., and Yar, M. 1988. Holt-winters forecasting: Some practical issues. *Journal of the Royal Statistical Society* 37(2):129–140.

Chatfield, C. 1978. The holt-winters forecasting procedure. *Journal of the Royal Statistical Society* 27(3):264–279.

Crmanová, G.; Laurinec, P.; Rozinajová, V.; Ezzeddine, A. B.; Lucká, M.; Lacko, P.; Vrablecová, P.; and Návrat, P. 2016. Incremental ensemble learning for electricity load forecasting. *Acta Polytechnica Hungarica* 13(2):97–117.

Demšar, J. 2006. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7(Jan):1–30.

Dunn, O. J. 1961. Multiple comparisons among means. *Journal of the American Statistical Association* 56:52–64.

Esling, P., and Agon, C. 2012. Time-series data mining. *ACM Computing Surveys* 45(1):Article 12.

Friedman, J. H. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29:1189–1232.

Friedman, J. H. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38:367–378.

He, X.; Pan, J.; Jin, O.; Xu, T.; Liu, B.; Xu, T.; Shi, Y.; Atallah, A.; Herbrich, R.; Bowers, S.; and Candela, J. Q. 2014. Practical lessons from predicting clicks on Ads at facebook. In *Proceedings of the 3rd IEEE International Conference on Cloud Engineering*, 1–9.

Hernandez, L.; Baladrón, C.; Aguiar, J. M.; Carro, B.; Sanchez-Esguevillas, A. J.; Lloret, J.; and Massana, J. 2014. A survey on electric power demand forecasting: Future trends in smart grids, microgrids and smart buildings. *IEEE Communications Surveys & Tutorials* 16:1460–1495.

Krawczyk, B.; Minku, L. L.; Gama, J.; Stefanowski, J.; and Woźniak, M. 2017. Ensemble learning for data stream analysis: A survey. *Information Fusion* 37(C):132–156.

Laurinec, P., and Lucká, M. 2016. Comparison of representations of time series for clustering smart meter data. In *Proceedings of the World Congress on Engineering and Computer Science*, 458–463.

Laurinec, P., and Lucká, M. 2017. New clustering-based forecasting method for disaggregated end-consumer electricity load using smart grid data. In *Proceedings of the 14th International Scientific Conference on Informatics*, 210–215.

Liao, T. W. 2003. Clustering of time series data - a survey. *Pattern Recognition* 38:1857–1874.

ReportLinker. 2017. Global desktop virtualization market analysis (2017-2023). https://www.reportlinker.com/p05207394.

Song, H., and Li, G. 2008. Tourism demand modelling and forecasting-a review of recent research. *Tourism Management* 29(2):203–220.

Tan, Y.; Liudmila, U.; Ye, O.; and Fengyuan, X. 2014. Data mining in time series: Current study and future trend. *Journal of Computer Science* 10(12):2358–2359.

Taylor, S. J., and Letham, B. 2017. Forecasting at scale. *The American Statistician* 72:37–45.

Taylor, S. J. 1986. *Modelling Financial Time Series*. Wiley, New York, NY.

Tidemann, A.; Høverstad, B. A.; Langseth, H.; and Öztürk, P. 2013. Effects of scale on load prediction algorithms. In *Proceedings of the 22nd International Conference and Exhibition on Electricity Distribution*, 1–4.

Wijaya, T. K.; Vasirani, M.; Humeau, S.; and Aberer, K. 2014. Individual, aggregate, and cluster-based aggregate forecasting of residential demand. Technical report, School of Computer and Communication Sciences, EPFL.