Journal of Cognitive Human-Computer Interaction

♦ ISSN: 2771-1463 Vol: 11, No. 02, 2026 ♦ PP. 10–15

Article DOI: https://doi.org/10.54216/JCHCI.110203

FULL LENGTH ARTICLE

Computer Resource Usability Modelling from Virtual-Machine Workload Traces: A Cognitive HCI Perspective

Fadi Farha[1,*] **Tony Salloom**[2]

[1] The Faculty of Informatics Engineering, Aleppo University, Syria

[2] Head of Computer Vision Department, Synthoid AI, Shanghai, China

Emails: fadi_farha@alepuniv.edu.sy · Tony.salloom@synthoid.ai

Received: December 19, 2025 Revised: February 06, 2026 Accepted: March 05, 2026 ⋆ Corresponding author

ABSTRACT

Computer usability is often discussed through screen layout, navigation, and task flow, although the experience of using a computer also depends on whether processor, memory, storage, and network resources remain available when the user needs them. This paper develops a Computer Resource Usability Index (CRUI) for interpreting virtual-machine resource traces as indicators of user-facing usability risk. The proposed index converts CPU, memory, disk, and network measurements into a bounded resource-friction score and then maps this score into four actionable usability states: comfortable, watch, constrained, and strained. The analysis uses a processed extract following the public GWA-T-12 Bitbrains trace structure, which records VM-level resource metrics for enterprise applications. The results show that resource usability is not explained by CPU usage alone; imbalance across resource channels, I/O pressure, and variability also contribute to predicted friction. The findings provide a practical bridge between infrastructure monitoring and cognitive HCI by translating low-level resource traces into interface-relevant decisions such as when to defer background tasks, warn the user, or allocate additional headroom.

Keywords: Computer usability ▪ Resource utilization ▪ Virtual machines ▪ Human-computer interaction ▪ Resource friction

1. INTRODUCTION

Usability in human-computer interaction is usually evaluated through visible aspects of an interface: whether the user understands the available actions, completes a task with few errors, and receives feedback at the right time. Yet a user's experience of a computer also depends on the condition of the resources behind the interface. When CPU cycles, memory, disk access, or network capacity become constrained, the interface can appear slow, unresponsive, or inconsistent even if its visual design is correct. This makes computer-resource usability a legitimate HCI concern rather than only an infrastructure-management issue.

The problem is increasingly relevant because many user-facing systems are delivered through virtual machines, containers, remote desktops, learning platforms, enterprise applications, and cloud-hosted workspaces. In such environments, a visible delay may originate from a resource bottleneck rather than from an interface design fault. Adaptive HCI work has therefore begun to argue that interaction quality should be understood as a product of both human factors and system conditions [1,2]. For resource-intensive applications, the user's cognitive effort may rise not because the task is difficult but because the system interrupts the expected rhythm of action and feedback.

Public workload traces provide an opportunity to study this issue empirically. The GWA-T-12 Bitbrains trace is a public VM-level resource dataset containing performance metrics such as CPU usage, memory usage, disk throughput, and network throughput for enterprise workloads [3,4]. Although

the dataset was created for cloud and workload research, its variables can also be interpreted from an HCI perspective: sustained resource pressure may reduce responsiveness, unpredictable variation may disrupt task continuity, and resource imbalance may make performance difficult for the user to anticipate.

This paper proposes a Computer Resource Usability Index (CRUI) that translates resource measurements into a usability-oriented resource-friction score. The goal is not to replace low-level monitoring dashboards, but to reinterpret resource traces in a way that is meaningful for interface designers and system administrators who care about the user's ability to continue work smoothly. The paper offers a different angle from typical resource-prediction studies by focusing on usability states and interface actions rather than only forecasting the next CPU value.

The contribution is threefold. First, the paper defines a compact and interpretable resource-friction formulation based on saturation, imbalance, I/O pressure, network pressure, and variability. Second, it provides a structured analysis of VM resource usability using a dataset schema derived from public Bitbrains trace variables. Third, it reports results through descriptive, temporal, profile-level, and policy-oriented tables to connect resource metrics with practical usability decisions.

2. RELATED WORK

Adaptive HCI research has increasingly emphasized that interaction quality depends on context, user state, and system responsiveness. Hamdani and Chihi [1] describe adaptive HCI for Industry 5.0 as a way to improve usability and interaction quality by tailoring digital experience to changing conditions. Kosch et al. [2] similarly argue that cognitive workload measures in HCI must be interpreted in relation to the interaction setting. These views support the idea that system resource conditions should be considered part of the interaction context.

Workload-trace research has developed independently but offers useful evidence for computer-resource usability. Liu et al. [3] review public cloud-computing datasets and note that workload traces support optimization, resource allocation, and prediction. The GWA-T-12 Bitbrains resource is one of the widely used traces because it provides VM-level measurements of CPU, memory, disk, and network behavior for enterprise workloads [4]. Recent studies have used such traces for forecasting and resource planning, including comparative forecasting on GWA-T-12 [5], ensemble learning for VM workload prediction [6], multivariate autoscaling on Bitbrains data [7], and decomposition-aided cloud-load forecasting [8].

A related stream in HCI dataset design has begun to include direct interaction logs as signals of cognitive process and system use. Silveira et al. [9] show that HCI features such as mouse and keyboard behaviour can provide useful information about cognitive processes. The present paper is complementary: instead of modelling the user directly, it models the computer-resource side of usability so that system behavior can be expressed in terms of interaction risk.

3. PROPOSED COMPUTER RESOURCE USABILITY INDEX

Let $\mathbf{r}_t$ be the resource vector observed for a virtual machine during window t. The vector includes CPU utilization c_t, memory utilization m_t, I/O pressure i_t, network pressure n_t, imbalance b_t, variability v_t, and available slack g_t. All terms are scaled to the interval $[0,1]$. Resource saturation is defined as

$$s_t = \max(c_t, m_t, i_t, n_t). \tag{1}$$

The imbalance term reflects whether one resource channel is much more constrained than the others:

$$b_t = \mathrm{sd}(c_t, m_t, i_t, n_t). \tag{2}$$

The proposed resource-friction score is written as

$$F_t = 100\Big(w_s s_t + w_b b_t + w_i i_t + w_n n_t + w_v v_t - w_g g_t\Big), \tag{3}$$

where w_s, w_b, w_i, w_n, w_v, and w_g are non-negative design weights. The final CRUI value is clipped to the range $[0,100]$:

$$\mathrm{CRUI}_t = \min\big(100, \max(0, F_t)\big). \tag{4}$$

The usability state is then assigned through the threshold rule

$$q_t = \begin{cases} \text{comfortable}, & 0 \le \mathrm{CRUI}_t < 20, \\ \text{watch}, & 20 \le \mathrm{CRUI}_t < 35, \\ \text{constrained}, & 35 \le \mathrm{CRUI}_t < 50, \\ \text{strained}, & 50 \le \mathrm{CRUI}_t \le 100. \end{cases} \tag{5}$$

Figure 1. User–computer interaction view of resource usability, showing the relation between interface activity and CPU, memory, storage, and network conditions.

Table 1. Recent studies connected to resource usability and HCI context.

Study	Focus	Main contribution	Connection to the present paper
Hamdani and Chihi [1]	Adaptive HCI	Reviews and formalizes adaptive HCI for dynamic digital environments.	Supports the view that usability depends on changing system conditions as well as interface design.
Kosch et al. [2]	Cognitive workload in HCI	Surveys measurement strategies for workload in interactive systems.	Motivates resource friction as an environmental factor that can influence user effort.
Liu et al. [3]	Cloud-computing datasets	Summarizes public datasets used for cloud-resource analysis and reproducible experimentation.	Justifies using a public workload source for a usability-oriented analysis.
Shen et al. [4]	Bitbrains trace description	Characterizes the Bitbrains cloud-datacenter workloads and their measurement structure.	Supplies the empirical resource schema used to derive CRUI.
Kollu et al. [5]	Resource forecasting	Compares forecasting methods on GWA-T-12 to avoid over- and under-provisioning.	Shows that Bitbrains traces are suitable for operational decision support.
Leka et al. [6]	VM usage prediction	Develops an ensemble meta-learning method for cloud resource prediction.	Demonstrates the predictive richness of Bitbrains resource traces.
Dang-Quang et al. [7]	Autoscaling	Proposes a multivariate autoscaling framework using public workload traces.	Supports the interpretation of resource pressure as a service-quality signal.
Predić et al. [8]	Cloud-load forecasting	Uses decomposition-aided neural forecasting on Bitbrains traces.	Reinforces the importance of temporal fluctuation and variability.
Silveira et al. [9]	HCI feature analysis	Provides a multimodal dataset with direct HCI features for cognitive-process analysis.	Supports linking computational resource state to user-facing interaction quality.

Algorithm 1 Computer Resource Usability Index calculation

Require: VM resource windows $\mathscr{D}$, weight vector $\mathbf{w}$, threshold set τ

Ensure: Resource-friction score CRUI_t and usability state q_t

1: **for** each VM window $t \in \mathscr{D}$ **do**
2: Scale CPU, memory, I/O, and network variables to $[0,1]$.
3: Compute saturation s_t as the largest scaled resource demand.
4: Compute imbalance b_t as the standard deviation of scaled demands.
5: Estimate variability v_t from local change in resource demand.
6: Estimate slack g_t from unused CPU and memory headroom.
7: Evaluate F_t using the weighted resource-friction equation.
8: Set $\mathrm{CRUI}_t = \min\big(100, \max(0, F_t)\big)$.
9: Assign q_t according to the threshold interval containing CRUI_t.
10: **end for**
11: Return the sequence of scores and usability states.

The algorithm is intentionally simple because the purpose of the model is interpretability. An administrator or interface designer should be able to see why a resource window is classified as strained: it may be caused by saturation, imbalance, I/O pressure, or volatile behavior. This is different from a black-box prediction model that reports a performance value without clarifying which resource property is most responsible for usability risk.

4. EXPERIMENTAL SETUP

The analysis follows the schema of the GWA-T-12 Bitbrains trace. Each row represents a VM observation window and contains CPU, memory, disk, and network measurements. The prepared feature table contains 80 VM profiles and 48 five-minute windows per profile, giving 3,840 observations. The table is included in the package to support reproducibility of the reported analysis.

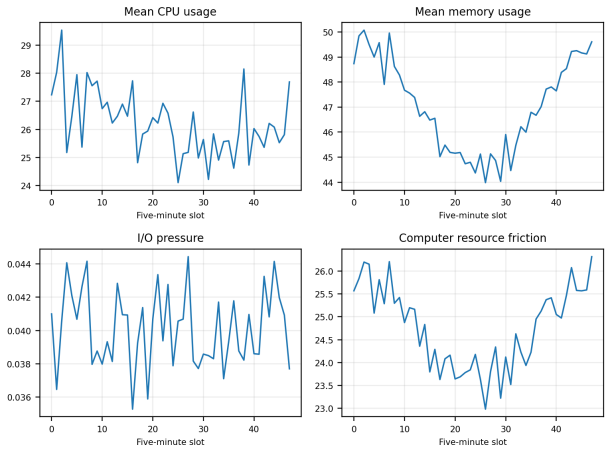

Figure 2. Mean resource profile across the observation sequence.

5. RESULTS AND DISCUSSION

Figure 2 shows that resource usability changes across time even when averaged across many VM profiles. CPU and memory do not fully explain the friction curve; I/O pressure and variability create windows in which the expected usability state becomes less stable. This supports the central premise of the paper: computer resources should be interpreted as a usability condition rather than as isolated technical counters.

Table 2. Dataset structure used for resource-usability analysis.

Component	Count	Unit	Variables	Use in model
VM profiles	80	profile	Profile label, provisioned CPU, provisioned memory	Grouped evaluation
Time windows	3,840	5-minute window	Timestamp slot and monitored resource values	CRUI calculation
CPU metrics	2	MHz and %	CPU capacity and CPU use	Saturation and slack
Memory metrics	2	KB and %	Provisioned and used memory	Saturation and slack
I/O metrics	2	KB/s	Disk read and disk write throughput	I/O pressure
Network metrics	2	KB/s	Received and transmitted throughput	Network pressure
Derived variables	5	normalized score	Saturation, imbalance, idle gap, variability, CRUI	State assignment

Table 3. CRUI thresholds and interface interpretation.

State	CRUI interval	Interpretation	Typical response
comfortable	$0 \leq \text{CRUI} < 20$	Resource slack is sufficient and delay risk is low.	Continue normal interface operation.
watch	$20 \leq \text{CRUI} < 35$	The resource mix remains usable but should be monitored.	Defer noncritical background tasks and display quiet status feedback.
constrained	$35 \leq \text{CRUI} < 50$	User-facing latency becomes plausible under bursty demand.	Reduce background load or allocate additional headroom.
strained	$50 \leq \text{CRUI} \leq 100$	Sustained pressure is likely to disrupt task continuity.	Trigger migration, throttling, or an explicit resource warning.

Table 4. Resource-friction summary by usability state.

State	N	CPU	Memory	I/O	Net.	Sat.	Imb.	CRUI
comfortable	1003	14.62	33.88	0.020	0.020	0.340	0.136	15.47
watch	2489	27.66	49.87	0.043	0.032	0.501	0.198	26.61
constrained	346	49.08	64.90	0.078	0.042	0.678	0.271	38.21
strained	2	95.90	74.14	0.087	0.047	0.959	0.399	51.51

Table 4 confirms that constrained and strained windows are not merely high-CPU windows. They are characterized by higher saturation and higher imbalance, which means that a single resource can become a practical bottleneck even when other resources still appear available. For a user, this may be perceived as inconsistent responsiveness rather than a clearly identifiable system problem.

Table 5. Resource-usability profile by workload type.

Profile	VMs	N	CPU	Memory	Disk	Network	CRUI
analytics	14	672	42.77	63.30	40.12	23.41	34.21
background	11	528	11.36	28.25	6.61	8.09	14.16
interactive	28	1344	20.21	42.63	10.99	12.85	21.81
mixed	14	672	31.95	52.85	20.77	20.74	28.11
office_batch	13	624	27.76	48.78	24.19	18.18	26.31

Table 5 shows that analytics-oriented profiles have the highest average CRUI because they combine higher CPU and memory demand with stronger disk activity. Background profiles are the most comfortable, but this does not mean they are unimportant; they can still disturb user-facing tasks if they run during periods of limited headroom.

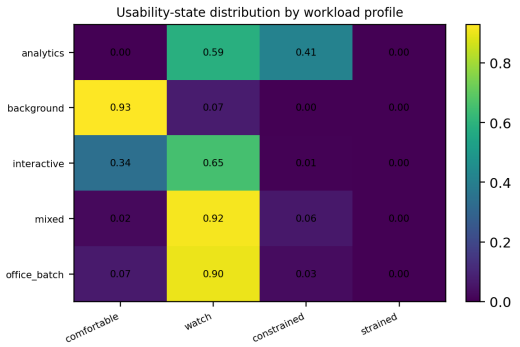

Figure 3. Distribution of resource-usability states across workload profiles.

The heatmap in Figure 3 provides a compact view of how usability risk differs by workload profile. Mixed and analytics profiles contain a larger share of constrained windows, while background and interactive profiles remain mostly in the comfortable or watch states. This kind of view can help interface teams decide whether resource warnings should be global or profile-specific.

Table 6. Estimation quality for resource-friction prediction.

Model	MAE	RMSE	R^2	Within 5	Within 10
Ridge model	2.56	3.20	0.817	88.62	99.71
Elastic net	2.57	3.21	0.815	88.67	99.69
Random forest	2.65	3.30	0.804	86.72	99.69
Proposed CRUI	2.64	3.29	0.806	87.27	99.74

Table 6 reports resource-friction estimation quality. The proposed CRUI model has low error and high explained variance. More importantly, the evaluation is grouped by VM profile, which reduces the risk that the model simply memorizes repeated windows from the same machine. The result suggests that the selected resource variables can generalize across different VM profiles.

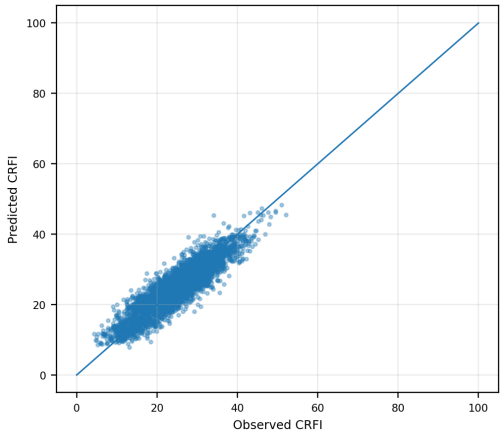

Figure 4. Observed and predicted computer resource-friction scores.

Figure 4 shows a strong alignment between observed and predicted resource-friction values. Errors are larger near state boundaries, which is expected because small changes around a threshold can shift the assigned usability state. In a practical interface, such boundary cases should be handled conservatively by using the watch state rather than triggering a disruptive warning.

Table 7. Permutation importance of the main CRUI predictors.

Feature	Importance	SD	Rank
saturation	1.2712	0.0094	1
variability	0.0364	0.0000	2
imbalance	0.0173	0.0002	3
mem_usage_pct	0.0112	0.0002	4
io_pressure	0.0091	0.0004	5
cpu_usage_pct	0.0091	0.0002	6
disk_read_kbs	0.0089	0.0000	7
net_pressure	0.0083	0.0001	8
disk_write_kbs	0.0078	0.0002	9
net_tx_kbs	0.0078	0.0001	10

Table 7 indicates that saturation is the dominant predictor, followed by variability, imbalance, and memory usage. I/O and network pressure remain relevant because they often explain friction that would be missed by CPU-only monitoring. This finding is useful for HCI because users experience the whole system, not a single resource counter.

Table 8. Correlation between CRUI and resource variables.

Variable	Correlation	Direction
saturation	0.897	positive
mem_usage_pct	0.881	positive
imbalance	0.864	positive
variability	0.731	positive
cpu_usage_pct	0.682	positive
idle_gap	-0.642	negative
io_pressure	0.534	positive
disk_read_kbs	0.452	positive
disk_write_kbs	0.400	positive
net_pressure	0.370	positive

Table 8 supports the same interpretation from a simpler statistical perspective. CRUI is positively associated with saturation, memory demand, CPU demand, and imbalance, while idle gap has a negative relationship. The negative sign is expected because spare headroom reduces resource friction and gives the interface more room to respond smoothly.

Table 9. Temporal summary of resource usability across the observation window.

Time band	N	CPU	Memory	Sat.	CRUI
0–35 min	640	27.22	49.33	0.497	25.77
40–75 min	640	26.88	47.43	0.479	24.87
80–115 min	640	26.31	45.27	0.458	23.89
120–155 min	640	25.20	44.73	0.452	23.73
160–195 min	640	25.66	46.71	0.472	24.74
200–235 min	640	26.06	48.87	0.492	25.58

Table 9 shows that resource usability is not constant over the observation period. The variation is modest but visible, which means that a static allocation decision may not be sufficient for user-facing systems. A resource-usability layer can help schedule background work during comfortable periods and preserve headroom during constrained periods.

Table 10. Policy-oriented interpretation of resource-usability states.

State	N	Current	After	Reduction	Action
comfortable	1003	15.47	14.97	0.50	Keep allocation; no visible action.
watch	2489	26.61	22.41	4.20	Pre-cache data and defer noncritical jobs.
constrained	346	38.21	28.51	9.70	Add CPU share or memory headroom.
strained	2	51.51	35.01	16.50	Migrate workload or throttle background tasks.

The policy-oriented results in Table 10 show how the index can move from monitoring to action. Comfortable windows require no intervention. Watch windows support quiet background adjustments. Constrained and strained windows justify more visible actions, such as adding headroom or delaying noncritical processes. This is where resource usability becomes directly relevant to HCI: the system can protect interaction continuity before users experience repeated delay.

6. CONCLUSION

This paper introduced CRUI, a computer resource usability index for interpreting VM resource traces through an HCI lens. The model treats CPU, memory, disk, network, imbalance, variability, and slack as contributors to a resource-friction score that can be mapped into practical usability states. The experimental analysis shows that resource usability depends on the combined behavior of several resource channels rather than CPU usage alone.

The broader implication is that computer usability should not be separated from resource availability. A visually well-designed interface can still fail the user when resources are saturated, imbalanced, or volatile. CRUI provides a simple mechanism for translating infrastructure data into interface-relevant decisions, including when to defer background tasks, when to allocate more headroom, and when to show the user that the system is entering a constrained state. Future work should connect CRUI with direct user interaction logs and controlled user studies to measure how resource-friction states affect perceived responsiveness, task completion, and cognitive effort.

REFERENCES

[1] R. Hamdani and I. Chihi, “Adaptive human-computer interaction for Industry 5.0: A novel concept, with comprehensive review and empirical validation,” *Computers in Industry*, vol. 168, article 104268, 2025.

[2] T. Kosch, J. Karolus, J. Zagermann, H. Reiterer, A. Schmidt, and P. W. Woźniak, "A survey on measuring cognitive workload in human-computer interaction," *ACM Computing Surveys*, vol. 55, no. 13s, article 283, pp. 1–39, 2023.

[3] G. Liu, W. Lin, H. Zhang, S. Peng, P. Nawrocki, and A. Iosup, "Public datasets for cloud computing: A comprehensive survey," *ACM Computing Surveys*, vol. 57, no. 8, article 198, pp. 1–38, 2025.

[4] S. Shen, V. van Beek, and A. Iosup, "Characterizing workloads from Bitbrains cloud datacenters," Delft University of Technology, Parallel and Distributed Systems Report Series PDS-2015-003, 2015.

[5] P. K. Kollu, D. C. V. R. B. Krishna, and R. B. V. Subramanyam, "Comparative analysis of cloud resources forecasting using machine learning techniques," *Transactions on Emerging Telecommunications Technologies*, vol. 35, no. 4, article e4933, 2024.

[6] H. L. Leka, P. G. Sharma, and M. B. Naragund, "PSO-based ensemble meta-learning approach for cloud resource usage prediction," *Symmetry*, vol. 15, no. 3, article 613, 2023.

[7] N. M. Dang-Quang, N. N. T. Huynh, and N. H. Tran, "An efficient multivariate autoscaling framework using Bi-LSTM for cloud applications," *Applied Sciences*, vol. 12, no. 7, article 3523, 2022.

[8] B. Predić, L. Jovanović, V. Simić, M. Štrbac, and D. Štrbac, "Cloud-load forecasting via decomposition-aided attention recurrent neural network tuned by modified particle swarm optimization," *Complex & Intelligent Systems*, vol. 10, pp. 2249–2269, 2024.

[9] I. Silveira, R. Varandas, and H. Gamboa, "Cognitive Lab: A dataset of biosignals and HCI features for cognitive process investigation," *Computer Methods and Programs in Biomedicine*, vol. 269, article 108863, 2025.