The Seductive Appeal of Thin Clients

"Those who cannot remember the past are condemned to repeat it."
George Santayana, *The Life of Reason, Volume 1, 1905*

Niraj Tolia, David G. Andersen, M. Satyanarayanan

February 2005
CMU-CS-05-151

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

Interest in thin clients is very high today because of frustration with the growing total cost of ownership of personal computers. Unfortunately, thin clients may not meet the usability goal of crisp interactive response. This paper shows that the adequacy of thin-client computing is highly variable, and depends on both the application and the available network quality. For intensely interactive applications, the tight control of end-to-end network latency required by thin clients may be hard to guarantee at large scale. The paper advocates the concept of *stateless thick clients,* and describes how they may reduce total cost of ownership while preserving good interactive performance.

This research was partially supported by the National Science Foundation (NSF) under grant numbers ANI-0081396 and CCR-0205266, and by the Intel Corporation. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF, the Intel Corporation or Carnegie Mellon University. All unidentified trademarks mentioned in the paper are properties of their respective owners.

Keywords: interactive response, network latency, network delay, virtual machines, thin client computing, total cost of ownership, TCO

1 Introduction

In his 1983 classic, "Hints for Computer System Design" [10], Lampson offers the following quote from Jim Morris: *"The nicest thing about the Alto is that it doesn't run faster at night."* This quote succinctly captures the joy of a timesharing user who has just discovered the crisp, unvarying performance of a personal computer on highly interactive tasks. Gone is the pain of sluggish interactive response during periods of peak load. By co-locating a processor with each user, personal computing offers a computing experience that is unaffected by the actions of other users. Low I/O latency and high display bandwidth make possible the tight user-machine coupling of today's GUIs and interactive applications. Never having experienced the frustration of poor interactive response, most users today cannot relate first-hand to Morris' quote — it is just a relic of an archaic world, long gone and never to return.

Alas, history has come full circle. Our collective timesharing experience is relevant to *thin client* computing, a hot topic today for reasons discussed in Section 3. A thin client consists of a display, keyboard and mouse combined with sufficient processing power and memory for graphical rendering and network communication with a compute server using a specialized protocol. All application and OS code is executed on the server. The client has no long-term user state and needs no disk. Many thin client protocols exist, and their relative merits have been explored by Lai and Nieh [9].

Thin-client computing is similar to timesharing in that *even trivial user-machine interactions incur queuing delay on a resource that is outside user control.* In both cases, queuing delay is acutely sensitive to the vagaries of the external computing environment. In timesharing, the shared resource is the processor. In thin-client computing, it is the network. Thin clients may incur additional queuing delay at a shared compute server. For example, some proposed designs use a small set of blade servers for a large number of thin clients, a modern realization of the processor pool timesharing design pioneered by Wilkes and Needham [18]. Server queuing delay can be eliminated by dedicating a compute server per client, but network queuing delay cannot be eliminated — it is intrinsic to the thin client model.

2 Position Statement

This paper shows that the adequacy of thin-client computing is highly variable, and depends on both the application and the available network quality. Thin clients may be inappropriate for some common network environments.

If near-ideal network conditions (low latency and high bandwidth) can be guaranteed, thin clients offer a good computing experience. As network quality degrades, interactive performance suffers. It is latency, not bandwidth, that is the greater challenge. Tightly coupled tasks such as graphics editing suffer more than loosely coupled tasks such as web browsing. The combination of the worst anticipated network quality and the most tightly coupled task determines whether a thin client is satisfactory. The extreme case of network disconnection cannot be tolerated by thin clients.

Note that it is the worst case, not the average case, that determines whether a thin client approach will be satisfactory. An organization that adopts thin client computing must also invest in system management resources to ensure adequate network quality at all times for its most demanding interactive tasks.

Thick clients offer an easy way to cope with uncertain network quality. Certain problems inherent in enterprise-scale deployments of thick clients lead to interest in thin clients. Fortunately, these problems can be overcome using virtual machine technology, as discussed in Section 6.

3 Why Thin Clients are Attractive

Two distinct reasons motivate interest in thin clients. The first and more important reason is *concentration of personal computing state.* In a large organization, the physical dispersion of personal computing hardware complicates system administration. Isolating an infected machine, forcing certain security upgrades, or restarting a crashed machine are examples of actions that typically require physical access to the hardware. Concentrating all relevant state in compute servers simplifies this physical access. Rather than walking from machine to machine, a system administrator has them all at her fingertips in a server room.

These considerations are especially relevant at enterprise scale, where the *total cost of ownership* of personal computers is of growing concern. As hardware costs plummet, an increasing fraction of the total lifetime cost of owning a personal computer goes to its ongoing maintenance rather than to its initial purchase. Thin clients offer the hope that concentration of state will lead to reduced total cost of ownership.

The second reason for interest in thin clients is *user mobility.* A user can authenticate at any thin client and have immediate access to her unique computing environment. This anonymity of thin clients harkens back to timesharing, where one could login at any "dumb terminal." It enhances collaboration and spontaneity, and simplifies the logistics of hardware deployment.

4 Interactive Response and User Perception

What performance goals should thin client computing strive for so that it will be embraced by users? The most critical performance measure is the crispness of interactive response. For example, when a user presses a mouse button she expects the pop-up menu to appear with no perceptible delay; in free-hand drawing she expects the on-screen curve to track her mouse movements with no lag; when enlarging or shrinking an object, she expects the rubber-banding effect on screen to smoothly and precisely track her mouse. This is the standard of interactive performance today, and users will be loathe to settle for less.

Notice that it is the "trivial" interactions that are the most challenging for thin clients. Such interactions involve end-to-end communication (from user to application code and back to user), but involve negligible delay within the application. These interactions suffer the full queuing delay of the network, yet must meet the user's highest expectation of performance. This is in contrast, for example, to a click on a web link; in that case, the user is already expecting download delay.

Over a 40-year period, the HCI community has built up a substantial body of knowledge about the impact of interactive response times on user satisfaction and task productivity. Examples of work in this area include those by Card et al [1], Guynes [4], Martin and Corl [11], Miller [12], Rushinek and Rushinek [14], Shneiderman [17], and Youmans [19]. From these studies, a broad consensus has emerged on acceptable response times for trivial interactions:

- Response times below 150ms are imperceptible to the user. This is therefore a good quantitative definition of "crisp response."
- In the range from 150ms to 1s, users become increasingly aware of response time. They strongly prefer response times well below 1s.
- Above 1s, users are unhappy. When forced, users are able to adapt to response times over 1s. However, this is accompanied by frustration with the system and a drop in productivity.
- Good response time is the key to overall satisfaction with an interactive session. User anxiety is positively correlated with poor response time.

Also relevant to this discussion is the finding from psychology and economics [2, 6, 5] that negative experiences have much greater impact than positive experiences on judgment and risk taking. The implication for thin clients is that incidents of poor response times will be overweighted in users' memories. Even a few sluggish interactions in an otherwise acceptable interactive session may be sufficient to turn off a user.

5 Evaluating the Adequacy of Thin Clients

This section answers two questions: How well do thin clients perform under different network conditions, and for what kinds of applications are they best suited? We answer these questions by applying the above HCI guidelines to a set of three application traces under a variety of network conditions. How often is each application's response time crisp, annoying, or unacceptable?

5.1 Data Collection

The following evaluation uses the open-source VNC thin client [13]. To model user activity, we captured three different user input traces using the Xnee record and replay tool [15]. These traces, described in Section 5.3, capture the keyboard and mouse actions of a user working with a photo editing application, a presentation creator, and a word processor.

These user input traces were replayed on a VNC client connected to the VNC server on a 100 Mbit/s switched Ethernet network with sub-ms round-trip time. The server exported a desktop of size 1600×1200 with a depth of 16 bits-per-pixel. During replay, an intermediary machine captured the packet traces for all client-server interaction. We analyzed the packet traces to identify the VNC protocol packets containing client user input and screen update requests, and the replies from the server.

5.2 Simulation and Analytical Method

The packet traces drove a simulator, built using *ns-2* [3], that had a high-level understanding of the VNC protocol. The simulator deterministically modeled the impact of different network characteristics on user performance. We simulated a wide variety of network conditions, but in the interest of space, this section presents only the results of 10 and 100 Mbit/s networks with round-trip latencies of 1, 66, and 100ms.

The simulator preserves quality—it does not drop updates on slower networks, because doing so could also degrade the user experience—but it batches updates more aggressively than the real VNC system might. This batching ensures that the simulated performance meets or exceeds that of the real system, making an aggressive assumption in favor of thin clients.

The packet trace records the user action packets and the subsequent series of screen updates and requests for updates from the client. As shown in Figure 1, these requests each release additional screen updates from the server. When simulating a slower network, an incoming request for updates releases *all* screen updates that were ready on the server. In other words, actions and updates are causally related, but the subsequent updates become available as a function of server time.

The simulation assumes that screen updates following action 1 are causally related to action 1. It conservatively assumes that the updates are *not* causally related to the intervening "request for update," but become available a fixed time after the user action was received by the VNC server. During replay to a slower client, the last screen update could be sent *before* the request for update, improving the interactive performance of the simulated VNC system.

We define an operation as beginning at the time the client sent a user action and terminating with the last screen update from the server before a subsequent user action. Each operation captures a single user event—such as a keystroke or mouse click. The analysis only counts the response time for user actions that generate a screen update. Based upon the discussion in Section 4, we place response times into one of several bins:

Latency	Subjective impression
$<$ 150ms	Crisp
150ms - 1s	Noticeable to Annoying
1s - 2s	Annoying
2s - 5s	Unacceptable
$>$ 5s	Unusable

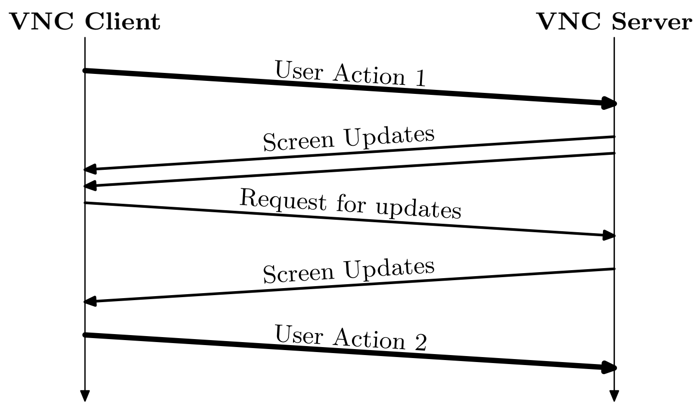

Figure 1: Simulation of VNC Protocol

5.3 Trace Descriptions

GIMP, similar to Adobe Photoshop, is a popular application for photo editing. The trace captured an experienced GIMP user separating headshots from a group photograph, removing red eyes from another image, and sharpening another blurry image. The trace was about five minutes long.

Impress, similar to Microsoft PowerPoint, is a part of the OpenOffice suite. The trace captures a user creating a new slide for a presentation. Examples of user actions captured include importing external images, creation of various shapes on the slide and the addition of text to them, addition of animation, and previewing and tweaking the animation. The trace was about ten minutes long.

Writer, similar to Microsoft Word, is the word processor in the OpenOffice suite. To distinguish between different usage scenarios, this trace was divided into two parts. One part consisted of a user transcribing a page of text into Writer. The other part captured the actions of tracking changes made to a large document and either accepting or rejecting these changes. The combined duration of this trace was approximately nine minutes.

5.4 Results

The interactive performance of a thin client setup varies greatly with two parameters: the degree of interactivity of the application and the latency of the network. Highly interactive applications such as GIMP perform poorly with even moderate latency. High latency impairs all but the simplest applications. Table 1 shows the number of operations at 100 Mbit/s that fell into each time bin for the applications described above. The chart should be read in two directions—within an application as latency increases, and, for a given latency, between applications. Table 2 shows the same data for a 10 Mbit/s network. The similarity of the two tables suggests that, above 10 Mbit/s, bandwidth does not limit thin client performance for these applications.

The data in Table 1 shows that on a 100 Mbit/s network with 1 ms latency, all of the traced applications perform well. Few operations require more than 1s to complete, and most complete before the 150ms threshold. This finding agrees with that of previous work [9] and with our subjective observation that VNC performed very well on a local network.

As latency increases, however, the picture does not remain rosy. While the *mean* response time remains low, as shown in Figure 2, the number of operations above the “annoying” and “unacceptable” thresholds increases dramatically. GIMP was the most demanding application. When run over a 100 Mbit/s network with 100ms of round-trip latency, 29% of the operations required more than 150ms to complete, and 1.5%

	Response Time Bin				
	Crisp	Noticeable	Annoying	Unacceptable	Unusable
Latency	< 150ms	150ms - 1s	1s - 2s	2s - 5s	> 5s
		GIMP			
1ms	3278	40	0	0	0
66ms	2710	572	12	3	21
100ms	2296	973	20	6	23
		Impress			
1ms	5879	15	1	0	0
66ms	5621	269	2	1	2
100ms	5520	363	9	1	2
		Writer - Tracking Changes			
1ms	270	15	0	0	0
66ms	201	82	0	0	2
100ms	182	99	2	0	2
		Writer - Typing			
1ms	2911	56	0	0	0
66ms	2899	66	0	0	2
100ms	2887	77	1	0	2

Table 1: Operations Response Time - 100 Mbit/s

required more than 1 second, compared to 1.2% and 0%, respectively, with a 1ms round-trip latency. These figures also matched our experience running the application over such a network: response time was poor, and at times, the lag was severe enough to make using the program difficult.

The application mix also makes a large difference to perceived performance: certain applications appear much better suited for thin-client computing than others. Looking down the columns in Table 1 shows how the performance changes between applications. Typing in a text editor, for instance, involves only small screen updates; its performance barely changes between the 1ms case (56 operations $>$ 150ms) and the 100ms case (80 operations $>$ 150ms). The presentation creation script occupies a middle ground, while the GIMP and Tracking Changes traces update large sections of the screen. These tightly coupled applications perform measurably worse at high latencies.

6 "Stateless Thick Client": Not an Oxymoron

The previous section exemplifies the kind of quantitative information that must be obtained when adopting thin client computing. After an organization determines acceptable limits of network latency and bandwidth for its unique mix of interactive applications, it must ensure that network quality never falls below those limits. Adding bandwidth is relatively easy, but reducing latency is much harder. In addition to physical layer transmission delays and end-to-end software path lengths, technologies such as firewalls, overlay networks, and lossy wireless networks add latency and other hurdles.

Vigilance by an organization will be required to ensure network adequacy in spite of user growth, network changes, and new or changing applications. Of course, this is not an impossible task. But it does imply

	Response Time Bin				
Latency	Crisp < 150ms	Noticeable 150ms - 1s	Annoying 1s - 2s	Unacceptable 2s - 5s	Unusable > 5s
GIMP					
1ms	3244	51	3	20	0
66ms	2552	730	12	3	21
100ms	2129	1140	20	6	23
Impress					
1ms	5869	23	1	2	0
66ms	5614	276	2	1	2
100ms	5514	368	10	1	2
Writer - Tracking Changes					
1ms	267	16	1	1	0
66ms	196	87	0	0	2
100ms	181	100	2	0	2
Writer - Typing					
1ms	2909	56	1	1	0
66ms	2899	66	0	0	2
100ms	2887	77	1	0	2

Table 2: Operations Response Time - 10 Mbit/s

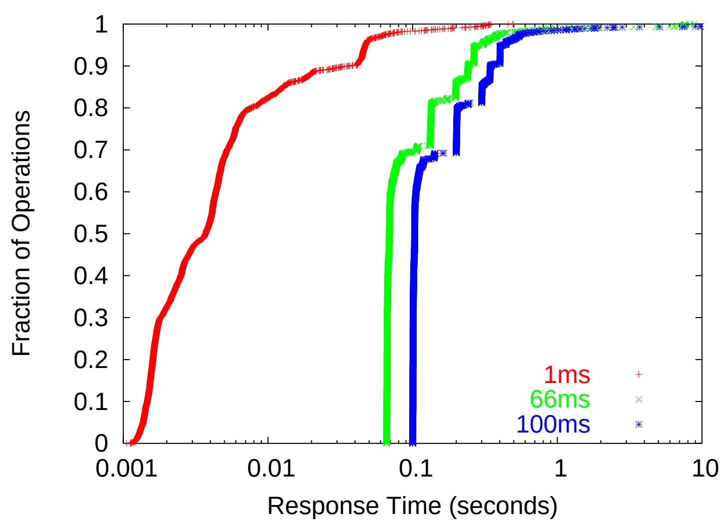

Figure 2: GIMP Operations - 100 Mbit/s

greater attention to system management than is apparent from a first glance at thin-client computing. It is not yet known how this will impact the hoped-for reduction in total cost of ownership.

Is there a more robust solution? Can the benefits discussed in Section 3 be achieved without the brittleness of a thin client solution? Virtual machine (VM) technology today enables clean encapsulation of a user's entire personal computing state. VM state can be stored in a distributed file system and delivered on demand to any thick client. This approach trades off startup delay for crisp interaction: once execution begins, all interaction is local.

Our work on *Internet Suspend/Resume(ISR)* [7, 8] confirms the feasibility of this approach, while Saputzakis et al [16] present relevant performance optimizations. An ISR client is stateless from the viewpoint of long-term user state. A user's entire "personal computer" (that is, his VM including guest OS, applications, and user files) is stored on file servers and delivered on demand. ISR clients can thus be identical to each other, and offer the same benefits of state concentration and user mobility as thin clients.

7 Conclusion

Crisp response is taken for granted today. The building blocks of modern-day GUIs such as scrolling, highlighting and pop-up menus are built upon this key assumption. Over time, we have learned how to use these building blocks to create applications that are well matched to human cognition. These applications, ranging from spreadsheets and word processors to CAD tools and medical imaging aids, are the backbone of personal computing today. Based on the assumption of excellent interactive response, an entire ecosystem of OSs, GUIs, applications, user expectations, and practices has co-evolved over two decades.

Thin clients pose an inadvertent threat to this world. Born of frustration with high total cost of ownership, interest in thin clients is very high. Unfortunately, dependence on thin clients may hurt the hard-won goal of crisp interactive response. An alternative approach is to continue using thick clients but to encapsulate user state using VM technology and to deliver it on demand from servers. A hybrid approach could start as a thin client to mask state transfer delay, and then switch to VM execution to provide crisp response. Regardless of the best final solution, care must be taken to ensure good user experience for demanding interactive applications under adverse network conditions.

References

[1] Card, S. K., Robertson, G. G., and Mackinlay, J. D. The Information Visualizer, An Information Workspace. In *CHI '91: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (1991), ACM Press, pp. 181–186.

[2] Coombs, C.H., Lehner, P.E. Conjoint Design and Analysis of the Bilinear Model: An Application to Judgements of Risk. *Journal of Mathematical Psychology 28*, 1 (May 1984), 1–42.

[3] Fall, K., and Varadhan, K. *The ns Manual*. The VINT Project, December 2003.

[4] Guynes, J. L. Impact of System Response Time on State Anxiety. *Communications of the ACM 31*, 3 (1988), 342–347.

[5] Kahneman, D. Maps of Bounded Rationality: A Perspective on Intuitive Judgement and Choice. In *Nobel Prizes 2002*. Almqvist & Wiksell International, 2003.

[6] Kahneman, D., Tversky, A. Prospect Theory: An Analysis of Decision under Risk. *Econometrica 47*, 2 (March 1979), 263–292.

[7] Kozuch, M., Satyanarayanan, M. Internet Suspend/Resume. In *Proceedings of the Fourth IEEE Workshop on Mobile Computing Systems and Applications* (Calicoon, NY, June 2002).

[8] Kozuch, M., Satyanarayanan, M., Bressoud, T., Helfrich, C., Sinnamohideen, S. Seamless Mobile Computing on Fixed Infrastructure. *IEEE Computer 37*, 7 (July 2004).

[9] Lai, A., Nieh, J. Limits of Wide-Area Thin-Client Computing. In *Proceedings of the 2002 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems* (Marina Del Rey, CA, June 2002).

[10] LAMPSON, B.W. Hints for Computer System Design. In *Proceedings of the 9th ACM Symposium on Operating Systems Principles* (Bretton Woods, NH, October 1983).

[11] MARTIN, G. L., AND CORL, K. G. System Response Time Effects on User Productivity . *Behaviour and Information Technology 5*, 1 (1986), 3–13.

[12] MILLER, R. B. Response Time in Man-Computer Conversational Transactions. In *Proceedings of the AFIPS Fall Joint Computer Conference* (1968), pp. 267–277.

[13] RICHARDSON, T., STAFFORD-FRASER, Q., WOOD, K. R., AND HOPPER, A. Virtual Network Computing. *IEEE Internet Computing 2*, 1 (1998), 33–38.

[14] RUSHINEK, A., AND RUSHINEK, S. F. What Makes Users Happy? *Communications of the ACM 29*, 7 (1986), 594–598.

[15] SANDKLEF, H. Testing Applications with Xnee. *Linux Journal 2004*, 117 (2004), 5.

[16] SAPUNTZAKIS, C., CHANDRA, R., PFAFF, B., CHOW, J., LAM, M., ROSENBLUM, M. Optimizing the Migration of Virtual Computers. In *Proceedings of the 5th Symposium on Operating Systems Design and Implementation* (Boston, MA, Dec. 2002).

[17] SHNEIDERMAN, B. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 2nd ed. Addison-Wesley, Reading, MA, 1992.

[18] WILKES, M.V., NEEDHAM, R.M. The Cambridge Model Distributed System. *SIGOPS Operating Systems Review 14*, 1 (1980).

[19] YOUMANS, D. M. User Requirements for Future Office Work-stations With Emphasis on Preferred Response Times. Tech. Rep. HF058, IBM United Kingdom Laboratories, Hursley Park, September 1981.