Network Personal Computing For World Simulation

Peter A. Dinda
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
pdinda@cs.cmu.edu

The future of personal computing will be commensurate with its past history. Over the next decade, realistic, user–centric applications will demand ever increasing compute power and memory. High speed, commodity networks will let this demand extend beyond the confines of a single personal computer, and allow individual users to exploit many machines collectively as extensions of their desktop personal computers. We call this vision network personal computing.

The driving applications for network personal computing will be realistic world simulations. World simulations faithfully simulate real (or imagined) human environments, and so are characterized by a tight control loop, where the results of the user's current action must be computed and displayed with imperceptible delay, because they affect his next input. There are many examples of such applications, including laboratory simulations, tele–operation, data visualization, teleconferencing, architectural walk–throughs, virtual reality, distributed battlefield simulations, flight simulations, and video games.

Supporting world simulations tomorrow requires that significant research be done today. We will drive our research with an archetypical world simulation, a parallel simulation of a photographic darkroom. The research goals we will pursue include modeling user behavior, modeling network traffic, devising useful network quality of service guarantees, developing network mechanisms to support such guarantees, inventing techniques for fast collective communication in the context of limited compile–time knowledge or run–time locality, quantifying and understanding tradeoffs between display update quality, response time, and offered network load, developing predictive techniques to improve responsiveness, and abstracting this knowledge into a framework for building world simulations.

The impact of this work will be manifold. First, it will result in a useful tool for photographers and artists. At a deeper level, it will help us understand how best to build network–based world simulations, and how to design networks to support them. This will encourage the building of world simulations, and the sale of networks and personal computers on which to run them. Ultimately, this will lead to the commoditization of world simulation and make personal computers even more useful and approachable by making them behave according to the predictable, real–world rules that humans are accustomed to.

1. Introduction

High performance computing research has traditionally been driven by the needs of scientific computing. The largest segment of computing, personal computing, has largely been ignored. However, advances in semiconductor technology, storage technology, and the economies of scale inherent in the personal computer market have led to extremely powerful, inexpensive computing hardware that is widely used by the general public. The public has found many ways to exploit this hardware, and, in fact, demands ever more powerful hardware. One has only to glance in a general computing periodical to understand why — ever more realistic applications such as games, simulations, and even productivity applications demand incredible compute horsepower, graphics performance, and storage. Since this trend of ever increasing demand for computing power has held for the entire history of personal computing, it is reasonable to expect that it will continue into the future.

Over the next decade, a new dimension to personal computing will develop. High bandwidth, low latency, off–the–shelf networks will bring supercomputer class communication performance, performance on par with memory bandwidths and processor speeds, to personal computers, and the economies of scale of this market will make the technology inexpensive, widely available, and widely used. Further, metropolitan and wide area networks will follow the same path. By the end of the decade, communication bandwidth will be like compute cycles and disk storage are today — "too cheap to meter." The implication is that computer networks will be used in significantly different ways in the future.

An important use of these future networks will be *network personal computing*, exploiting multiple powerful personal computers to enable new levels of realism and responsiveness from user–centric applications through distributed and parallel computing. The driving applications will be *world simulations*, programs that interact with the user at a fine level by simulating an environment in the real (or an imagined) world. To prepare for this future, we must begin today to pursue research aimed at supporting these very different programs on tomorrow's networks. To do this, we are constructing an archetypical world simulation, a *parallel darkroom simulator*, which will be used to drive such research as (1) modeling the user, network, and computational behavior of world simulations; (2) modeling, developing, and exploiting network quality of service characteristics and technology for world simulations; and (3) understanding and exploiting display quality tradeoffs in world simulations. The remainder of this document explains these concepts and our research directions in more detail.

2. Network Personal Computing

Future personal computers will often ship computations to other computers to be performed remotely, and these computations will often be parallel. This prediction may seem surprising, especially considering that these future personal computers will likely be powerful desktop shared memory multiprocessors. One could argue that a single one of these machines will be all a user could ever need. However, that claim is suspiciously like Bill Gates's infamous claim that "640KB should be more than enough for anybody." The reality is that the public's thirst for computing power is unquenchable. Finally, historical evidence argues in support of the technological viability of the prediction. The communication bandwidths of machines like the Intel Paragon and the Cray T3D have caught up with and remain on par with the compute and memory bandwidths of their small scale shared memory multiprocessor compute nodes.[1]

Network personal computing will be fundamentally different from high performance scientific computing. Just like their sequential counterparts, network personal applications will be user–centric and will be controlled by the user at very fine level. This is in sharp contrast to scientific applications, which are typically batch programs with static or data–driven control. In the same vein, the goal of distributing or parallelizing scientific applications using networks is almost always to minimize total running time, while the goals of distributing or parallelizing personal applications using networks will be to drive down response time while enhancing functionality and enhancing the user experience, the traditional market–driven goals of personal computing.

3. World Simulations

The force behind network personal computing will be world simulations. World simulations are programs which interact with the user by simulating an environment in the real (or an imagined) world, possibly using sensor inputs from the real world. They are characterized by a tight control loop: the user's current action must generate a fast response because it feeds directly into determining his next action. The speed and quality of the response determine the utility of the simulation. Examples of world simulations include laboratory simulations, tele–operation, data visualization, shared electronic whiteboards, teleconferencing, architectural walk–throughs, virtual reality, distributed battlefield simulations, flight simulations, and video games. Even today, world simulations have significant financial, military, and technological importance. Consider, for example, video games. Games are of profound interest to the military because they provide realistic training. Many game genres, such as flight simulation, were developed in the military, and later crossed over to civilian entertainment use. Games are a driving force in the personal computer industry.

Complex games that stress existing hardware become popular, and the public demands faster hardware to play them well. The hardware is developed, and its successful adoption fuels the development and sales of more complex games. Outside the personal computer industry, dedicated games have made giants out of companies like Nintendo and Sega via this process of positive economic feedback. The dedicated video game a child might receive as a birthday gift today has the computing power of a Cray I and a graphics engine worthy of a Silicon Graphics workstation of not very long ago.

World simulations are also significant well outside of the entertainment and military domains. In a very real way, the heart and soul of personal computing is world simulation. People know how to interact with the real world. They learned the rules in childhood, and they want to apply those rules in using a personal computer. In the past fifteen years, we have seen a fantastic move to graphical direct manipulation user interfaces which consume vast amounts of computing and graphics power to make familiar rules applicable to interacting with a computer — to simulate aspects of the real world. Today, this simulation is both better and worse than the emulated world, but people want enhanced worlds, world simulations that capture all the subtlety of the worlds they simulate and then some. In other words, they want the simulation to be strictly better than what is simulated. Network personal computing will make this possible.

4. Darkroom Simulation

Consider an archetypical world simulation, simulating the "dry side" of a photographic darkroom. The dry side of a darkroom is the environment of the photographic enlarger, a device for projecting a negative (or positive) onto a flat easel that holds light sensitive paper. "Printing" the negative consists of exposing the paper, and then developing it chemically (in the "wet side") to bring out the latent image. The position of the easel and the height of the enlarger "head" are varied to crop the image, while the density (darkness) of the image is determined by the duration of the exposure. This is a very basic description of the process. In fact, the range of possible image manipulations is endless. For example, "dodging" and "burning" tools can be used to reduce, or extend the exposure of selective portions of the image, optical filters can be used to vary contrast and color balance, or even to introduce non–local effects, and the easel can be tilted, or even replaced with a non planar surface in order to correct distortions or to warp the image in spectacular ways.

The immediacy of using an enlarger to work with a pure full resolution image is largely responsible for its power as an artistic tool. The artist can see the effects of even her most complex adjustments, such as warping the image, immediately, and on the actual image, not a facsimile. This gives her significant freedom to explore the creative space. Partial simulations, such as image editors like Adobe Photoshop, simply do not compare. Image editors are both more and less than a real darkroom. They emulate only some of the operations possible in a darkroom, such as burning, dodging, and airbrushing, and depart from the immediate behavior of a real darkroom when confronted with full resolution photographic images. The reason why is that the memory and computational resources needed to achieve parity are enormous. To understand the scale of the memory demands, consider the following table, which shows the amount of memory needed to support some common film formats. The film resolution is 100 lines/mm (a typical value) sampled at twice that rate to satisfy the Nyquist criterion.

Film Format	*Typical Use*	*Size in Pixels*	*Memory Required (4 Bytes/Pixel)*
24mm by 36mm	Standard 35mm format	4800 by 7200	132 MB
6cm by 6cm	Portraiture, commercial	12000 by 12000	550 MB
6cm by 7cm	Portraiture, commercial	12000 by 14000	640 MB
4in by 5in	Commercial, artistic	20320 by 25400	1969 MB

Notice that these memory sizes (132 MB to almost 2 GB) are unlikely to be available on personal computers in the next decade. The computational demands are also enormous. Consider realistically simulating the process of changing the color balance. On the enlarger, this is typically done by turning a filtration knob and observing the change. To simulate this, each of the three color channels of every pixel of the image needs to be updated around thirty times per second. For an image comparable to a 6cm by 7cm negative, this is over **five billion** updates per second, and this is a pixel–wise operation! Applying a small convolution kernel to correspond to non–local filtration or achieving real–time image warping as one can under an enlarger requires even more compute power.

4.1 Parallel Darkroom Simulation

Clearly, the memory and computational requirements of an accurate darkroom simulation argue for parallelism. Let us consider the implementation of such a parallel darkroom simulation. The first observation we make is that the user can interact with only a single set of input and output devices. We don't yet know what form these will take in the future, but in today's terms, the user can only sit in front of one personal computer. This suggests that we dedicate this machine to the job of interaction and display client. It captures the user's input, converts it to a command, and broadcasts it over the network to a parallel simulation server, a cluster of personal computers. The server machines perform collective communication over the network in order to execute the command, and then collectively generate a display update. The collective communication will depend on the command, the affected region of the image, and the distribution of the image and the computation over the server machines. Finally, each server machine ships its portion of the display update to the client, where it is displayed. The size of the display update depends on the affected region, and, if any necessary downsampling and clipping is done before sending the update, the size of the update can be no larger than the user's display. The goal is for the entire round–trip to be imperceptible to the user, just as it is in the real world.

5. Research

A parallel darkroom simulation implemented as described above presents some interesting challenges. Due to its user–driven nature, no a priori knowledge of communication patterns and volumes is available. Without this information, it is unclear how to exploit the quality of service capabilities of future networks, or how we might build quality of service capabilities that such applications can use. Yet, it appears that some sort of guarantees are necessary if we are to meet the hard real–time constraint on the round–trip. Since network traffic will depend in large part on user actions, can we characterize and model user behavior sufficiently well to provide input into a quality of service mechanism? How would such a mechanism work? If user behavior is in some measure predictable, can we exploit the predictions to reduce network load and improve response? Can we reduce network load or improve response by imperceptibly exploiting human tolerance for degraded image quality? In another vein, supposing we knew the communication patterns and volumes, how could we optimally schedule the communication on general purpose networks, which have arbitrary topologies? In this section, research into these and other questions is outlined.

5.1 Characterizing and Modeling User Behavior

An important research goal is characterizing and modeling user behavior in world simulation programs using the parallel darkroom simulator described above as a testbed. By "user behavior", we mean the semantics of the user's actions: the operations he selects, their characteristics, and where and when they are to be applied to the image. To collect initial data, we are building a tool to instrument Adobe Photoshop. The tool captures a trace of Photoshop's interactions with the windowing system. Another tool, a "Photoshop simulator," will convert this information into a trace of the user's behavior. This data will be used to develop an initial characterization and model of user behavior to help in the design of the parallel darkroom simulator. Ultimately, the model will be refined with data collected using the parallel darkroom simulator. A good model for user behavior, combined with implementation characteristics, will help us to model and predict the computation and communication workloads of simulator. Having models for these workloads will be critical for mapping the computation onto a non–dedicated network with arbitrary topology.

5.2 Quality of Service for World Simulations

In current data networks, the bandwidth, delay, and loss rate of a connection can vary drastically over time due to other connections in the network. This behavior makes these networks unsuitable for many kinds of traffic. In particular, the hard real–time requirements of the parallel darkroom simulator will be impossible to meet under these conditions. An important goal of next generation networks is be able to guarantee quality of service (QoS) for connections. However, most of the research[2,3,4,5,6,7,8] into quality of service has concentrated on how to support time–based media streams, in particular compressed video streams[9,10], because it is perceived that these will be of supreme commercial importance. Some work, in our group and others[11], is concentrating on QoS to support traditional parallel computing on networks, but no one has yet explored QoS for user–driven parallel applications such as world simulations. Due to the future commercial importance of these programs, this is an important area of research.

There are three significant components to research into QoS for these applications. The first is modeling application traffic. Such a traffic model is essential in order to understand what the application's communication requirements are and how meeting those requirements will affect the network.[9] Clearly, the traffic on each of the three sets of connections (client to server, intra–server, and server to client) implied by the parallel darkroom simulation described above will be strongly dependent on the user's behavior, and will be correlated in time with it. If we assume that commands sent from the client are of some fixed size, then we can reasonably characterize the client to server traffic along each connection because human input speed is limited. The traffic along intraserver connections will depend on the actual command sent, and how the image data and computation is mapped across the server machines. Currently, we can say very little about it other than that traffic along each connection will be correlated with traffic along other connections, an observation which has also been made for traditional parallel programs.[11,12] The characteristics of traffic along connections from server nodes to the client is also not well understood, other than it depends on the actual command sent, the affected region of the image, and what portions of the image are visible on the client. Clearly, the volume must be bounded by the display characteristics of the client. An important research contribution will be to characterize and model the traffic of this application.

Another research opportunity is understanding how to specify the performance guarantees themselves. Which of bandwidth, delay, or loss rate is it useful to guarantee, and for which connections? Is a less constraining guarantee of some combination of these parameters sufficient? Should the guarantees be deterministic or will statistical guarantees suffice?[8] If we can use statistical guarantees, what are the right parameters? What is the right tradeoff between improved network utilization and improved application behavior? Should the user choose the tradeoff point, and, if so, how? Finally, how can we use predictive feedback from networks without guarantees to our advantage? Answers to these questions will be an important research contribution because looser guarantees mean potentially more network bandwidth will be available to other clients.

The most interesting research component is the QoS mechanisms themselves and their interaction with mapping the parallel simulation to the server machines. Given a traffic characterization and a specification of the needed performance guarantees, how can the network decide whether to accept or reject our connections? If it accepts them, how can it assure that the guarantees are met? It is important to note that the application can adjust its mapping of image data and computation to the server machines in order to adjust its communication patterns. For example, techniques have been developed to automatically map task and data parallel signal processing applications in order to satisfy throughput and latency constraints.[13] Applying these techniques in a network with QoS causes a significant insight to emerge. Namely, the application and the network are each trying to optimize communication independently of the other.[12] Consider a simple model of the server running on P processors. Each processor repeatedly communicates N items to a neighbor, then does W/P work. An example of this might be applying a small convolution kernel to a block distributed image. If the network allocates a bandwidth B to each of the connections, then the interval between bursts will be $W/P + N/B$. Notice that the burst interval, which is certainly plays into what B the network can commit to is itself a function of B. Further, the application can vary the number of connections it needs by varying the number of processors P, which also varies the interval between bursts, again affecting the bandwidth the network can provide. How can we capture and exploit this interdependence between the application mapping and the service the network can provide is an important research question.

5.3 Fast Collective Communication With Limited Compile–time Knowledge and Locality

Technology has been developed to achieve fast collective communication for regular data and computation mappings using compile–time knowledge[14] or run–time caching[15,16], and for irregular, slowly changing, data and computation mappings using caching techniques[17,18]. Collective communication in world simulations presents new challenges because compile–time knowledge is unavailable, and communication patterns may change very quickly, both in terms of the aggregate, many–to–many structure of communication, and in the microcosm of data transfers between individual nodes. Further, it will be necessary to quickly and efficiently map this communication onto networks with arbitrary topologies. Developing techniques for these situations is an important research direction.

5.4 Interface Latency and Image Update Quality Tradeoffs

In the parallel darkroom simulator, it will be possible to trade off between display update quality, user interface latency (round–trip time), offered network load, and other values. In particular, due to the forgiving nature of human perception, we may be able to reduce the size of display updates through downsampling or lossy compression techniques, and later fully update the display when the user pauses. How can we take advantage of this adaptability when running on a network which does not provide necessary QoS guarantees? How can we can take advantage of this adaptability in order to simplify QoS mechanisms and minimize impact on other network users? Where and how shall the tradeoff point be decided? What is the effect of the tradeoff point on the artistic utility of the simulator? If we elect to have the user decide the tradeoff point, how do we present the information to him? What is an appropriate metric of quality in world simulation? These are all interesting research questions induced by the adaptability of world simulation.

5.5 Predictive Optimization

Can we model user behavior in parallel darkroom simulation to a sufficient degree that we can frequently predict the user's next few actions? If so, then it will be possible to overlap computing the response to a predicted action with sending the results of the current action back to the user. This would allow us to relax the latency requirements because the critical path would no longer be a round trip.

5.6 Application Frameworks for Network–based World Simulation

Application frameworks such as the Microsoft Foundation Classes and Metrowerks PowerPlant have drastically simplified building typical applications for windowing systems. Can we sufficiently abstract from the parallel darkroom simulator to allow us to build an effective and usable application framework for network–based world simulation?

6. Conclusion

Our vision of the future of personal computing is commensurate with its past history. Personal computing is synonymous with realistic, *user-centric* applications that require significant, and ever increasing amounts of compute power and memory to run well. In the next decade, high speed, commodity networks will permit the demand for compute power and memory to expand beyond the levels conceivable in a single personal computer of that period. Individual users will exploit many machines collectively in the same ways they currently exploit their own desktop personal computer — to run realistic, user–centric applications. We call this vision network personal computing.

The driving applications for network personal computing will be realistic world simulations. World simulations attempt to faithfully simulate real (or imagined) human environments, and so are characterized by a tight control loop, where the results of a user's current action must be computed and displayed very quickly, ideally with imperceptible delay, because they affect his next input. An example of such an application is the parallel darkroom simulator which we will use to drive our research.

Supporting world simulations tomorrow requires that significant research be done today. The research goals we will pursue include modeling user behavior, modeling network traffic, devising useful network quality of service guarantees, developing network mechanisms to support such guarantees, inventing techniques for fast collective communication in the context of limited compile–time knowledge or run–time locality, quantifying and understanding tradeoffs between display update quality, response time, and offered network load, developing predictive techniques to improve responsiveness, and abstracting this knowledge into a framework for building world simulations.

The impact of this work will be manifold. First, it will result in a useful tool for photographers and artists. At a deeper level, it will help us understand how best to build network–based world simulations, and how to design networks to support them. This will encourage the building of world simulations, and the sale of networks and personal computers on which to run them. Ultimately, this will lead to the commoditization of world simulation and make personal computers even more useful and approachable by making them behave according to the predictable, real–world rules that humans are accustomed to.

---

[1] Stricker, T. and Gross, T., *Optimizing Memory System Performance for Communication in Parallel Computers*, Proceedings of the 22nd International Symposium on Computer Architecture, June, 1995, pp. 308–319.

[2] Ferrari, D., *Client Requirements for Real–Time Communication Services*, IEEE Communications Magazine, November, 1990, pp. 65–72.

[3] Ferrari, D., Verma, D., *A Scheme For Real–Time Channel Establishment In Wide–Area Networks*, IEEE Journal on Selected Areas of Communication, Volume 8, Number 3, April 1990, pp. 368–379.

[4] Clark, D., Shenker, S., and Zhang, H., *Supporting Real–Time Mechanisms In An Integrated Services Packet Network: Architecture and Mechanism*, Proceedings of SIGCOMM '92, August, 1992, pp. 14–26.

[5] Zhang, H., and Keshav, S., *Comparison of Rate–based Service Disciplines*, Proceedings of SIGCOMM '91, September, 1991, pp.113–122.

[6] Golestani, S., *A Stop–and–Go Queuing Framework For Congestion Management*, Proceedings of SIGCOMM '90, September, 1990, pp. 8–18.

[7] Jamin, S., Danzig, P., Shenker, S., and Zhang, L., *A Measurement–based Admission Control Algorithm For Integrated Services Packet Networks*, Proceedings of SIGCOMM '95, August, 1995, pp. 2–13.

[8] Kurose, J., *Open Issues and Challenges in Providing Quality of Service Guarantees In High Speed Networks*, ACM Computer Communication Review, Volume 23, Number 1, January, 1993, pp. 6–15.

[9] Garrett, M., and Willinger, W., *Analysis, Modeling, And Generation of Self–Similar VBR Video Traffic*, Proceedings of SIGCOMM '94, September, 1994, pp. 269–280.

[10] Zhang, H., and Knightly, E., *A New Approach To Support Delay–Sensitive VBR Video in Packet–Switched Networks*, Proceedings of NOSSDAV '95, April, 1995.

[11] Fernandez, J., and Mutka, M., *Model And Call Admission Control For Distributed Applications With Correlated Bursty* Traffic, Proceedings of Supercomputing '95.

[12] Dinda, P., Garcia, M., and Leung, K., *Traffic Characterization of Parallel Programs*, May, 1995, working paper.

[13] Subhlok, J., O'Hallaron, D., Gross, T., Dinda, P., Webb, J., *Communication And Memory Requirements As The Basis For Mapping Task And Data Parallel Programs*, Proceedings of Supercomputing '94, November, 1994, pp. 330–339.

[14] Stichnoth, J., O'Hallaron, D., and Gross, T., *Generating Communication For Array Statements: Design Implementation, and Evaluation*, Journal of Parallel and Distributed Computing, Volume 21, Number 1, April, 1994, pp. 150–159.

[15] Dinda P., and O'Hallaron, D., *The Performance Impact of Address Relation Caching*, Proceedings of LCR '95, pp. 213–226.

[16] Dinda P., and O'Hallaron, D., *Fast Message Assembly Using Compact Address Relations*, Proceedings of ACM SIGMETRICS '96, to appear.

[17] Saltz, J., Petiton, S., Berryman, H., and Rifkin, A., *Performance Effects of Irregular Communication Patterns On Massively Parallel Multiprocessors*, Journal of Parallel and Distributed Computing, Volume 13, 1991, pp. 202–212.

[18] Agrawal, G., Sussman, A., and Saltz, J., *An Integrated Runtime and Compile–Time Approach for Parallelizing Structured and Block Structured Applications*, IEEE Transactions on Parallel and Distributed Systems, Volume 6, Number 7, July, 1995, pp. 747–754.