Grand Challenges Conference Application

Grand Research Challenge Problems

Todd C. Mowry
Computer Science Department, CMU
http://www.cs.cmu.edu/~tcm

Before I describe two specific scenarios, the overall theme that excites me for some grand research challenge problems in systems is finding new ways for computers to interact with the *physical* world. Much of what computer systems do today exists only in the virtual world. If we look at how many computer systems are designed, for example, I/O appears to be an afterthought. I would like to bring computing into the physical world of humans, rather than trying to immerse people in an online, virtual chat-room culture, etc. My first scenario is fairly far-fetched, and the second one is perhaps more practical.

1. A Computer-Enhanced Immune System

Many life-threatening diseases can be successfully treated if they are detected early enough. However, few adults probably see a doctor more than once a year. Imagine if we could augment our body's natural immune system with an "on-board" computer system that would continuously check for the presence of such diseases. Many vehicles today have systems like this that alert the driver that they should take their car in to be serviced. Imagine if we could be alerted to the earliest presence of cancer cells, heart disease, HIV, or other serious medical problems. Such a system might save many lives.

Now imagine taking this a step further. In addition to detecting a health problem, the computer-enhanced immune system might take active measures to correct the problem. Perhaps this would involve leveraging existing mechanisms within the bodies immune system, and directing them to the problem. Perhaps it would involve manufacturing certain chemicals or drugs within our bodies to treat the problems (e.g., insulin for diabetics). Or in a radical scenario, perhaps it would involve deploying nanotechnology robots to attack the problem in a way that our bodies cannot do so today. For example, rather than trying to coax our bodies own immune system into destroying cancer cells, perhaps these nanotechnology robots would physically destroy the cells themselves in a brute force method. (We might think of this as "nano-surgery.")

Obviously computer systems as we know them today are not equipped for such a purpose. Miniaturization would be needed on a much more ambitious scale than we see today. How would we network together the sensors and computers within this system without disturbing the natural functioning of our bodies? Perhaps the bloodstream or the central nervous system would be used as a form of a network. Security is clearly an important issue; we would not want other people to be able to access our on-board computers, especially if they could be compromised such that they could take active measures to hurt our bodies. How would we create a system that was reliable enough for peoples' lives to depend upon it? How would we manage the large database of information that this system would collect about how our bodies are functioning?

2. Augmented Reality for the Masses

The idea of "augmented reality" - i.e., integrating 3D graphics into what we actually see in reality - has been proposed before. Some motivating examples include "X-ray vision" for surgeons, and other specialized applications. We see a form of augmented reality in television broadcasts of pro football games with the yellow first-down stripe projected across the field.

What if we could make augmented reality technology ubiquitous? What if everyone could buy augmented reality glasses for $200? I think that this would radically change the way that we interact with computers in some very exciting ways. I think it would result in entirely new sets of applications that were based on making information useful within the physical world that surrounds us. A simple example of such an application would be a heads-up navigation system. Imagine that after specifying where you would like to go, a "yellow-brick road" would appear superimposed on the ground in front of you to guide you to your destination.

I think that there are many other more serious and useful applications of this technology, especially in education. Imagine conducting experiments in science classes where the underlying properties of what one was observing could be highlighted or emphasized visually. Perhaps a model of a system that a student had constructed could be superimposed upon the real object so that deviations between the model and reality would be more evident.

From a systems perspective, an enormous amount of computation would be needed to match the augmented content with what a person was actually observing, not to mention reasoning about what the content should be. The orientation of a person and possibly what their eyes were focusing upon would also be critical. In addition to research on new applications to exploit such a capability, we are a long way away systems technologies that would make this possible.

Biographical Statement:

I am currently an associate professor in the Computer Science Department at Carnegie Mellon University. I received my Ph.D. from Stanford in 1994, where my Ph.D. thesis advisors where Anoop Gupta and Monica Lam. While I was a Ph.D. student, I worked part-time in the MIPS architecture group on a number of different processor designs (including the MIPS R10000). From 1994 through 1997, I was an assistant professor in the ECE and CS departments at the University of Toronto prior to moving here to Carnegie Mellon in July 1997.

The goal of my research so far has been on improving the performance of commercial microprocessor-based systems without placing any additional burden on the programmer, usually through some combination of novel architectural, compiler, or OS support. One of my "secret weapons" in my research has been a strong emphasis on working across traditional systems boundaries in an effort to find new ways to match the strengths of one part of the system (e.g., the compiler's ability to analyze an entire program) with the strengths of other parts (e.g., the dynamic information that is available to the hardware or the OS).

There are two specific themes in the research problems that I have addressed so far. The first is how to cope with the ever-increasing relative latencies in accessing and communicating information (via DRAM, disks, and networks) that threaten to nullify any other improvements in processing efficiency. This work began in my PhD thesis on how to automatically insert data prefetch instructions into matrix-based codes to hide the cache-to-memory latencies on uniprocessors and multiprocessors. Since then, I have expanded this work to cover irregular memory access patterns (e.g., pointer-based codes) and larger latencies (e.g., to disk and across networks). More recently, I have been focusing specifically on applying these techniques to improving database performance. The second theme in my research so far is how to conquer the automatic parallelization problem so that we can exploit the full potential of parallel machines for general-purpose codes without placing this burden on the programmer. To accomplish this, we have a combination of new hardware and software support that allows the compiler to optimistically create parallel threads without for certain whether or not they are independent.