



December 16th, 2008

## [MiniFrame SoftXpand](#)

Posted by Dan Kusnetzky @ 3:00 am

**Categories:** [Virtual machine software](#), [Virtual processing software](#), [Access virtualization](#), [Managing virtualized environments](#), [Linux](#), [Operating system virtualization/partitioning](#), [Windows](#)

**Tags:** [Computer](#), [MiniFrame SoftXpand](#), [Productivity](#), [Microsoft Windows](#), [Tools & Techniques](#), [Remote Access](#), [Operating Systems](#), [Software](#), [Management](#), [Dan Kusnetzky](#)

After I posted on Ncomputing (see [Ncomputing: Sharing the power of a single computer](#)), [MiniFrame](#)'s PR agency contacted me and asked that I take a look at SoftXpand. At first glance, it appeared that MiniFrame was walking down the same path that many others had taken, creating technology in the operating system virtualization/partitioning category that would allow many people to share the computing power of a single computer. After a number of false starts, I was finally able to speak with MiniFrame's CEO, Eli Segal. I learned that my initial impression had been wrong. MiniFrame is doing something different. It has developed a software-only application that takes advantage of the both the multiuser, multitasking capabilities of the Windows operating system as well as both the CPU and graphics processing unit resources to create what is, in essence, a very cost-effective Minicomputer. Why go to that effort?

Well, the power of the computing engines (both central processing unit and graphics processor) have significantly overshot what a single individual typically requires by a rather wide margin. Depending upon how one evaluates this situation, organizations are either paying too much for a single person's system or they haven't yet found a way to fully utilize what they've been paying for.

Finding a way to better utilize all of that computational power is a goal some organizations have set for themselves. This has lead to the adoption of virtual machine or operating system virtualization/partitioning technology that allows the system to be

“loaded up” with work and, it is hoped, make better use of the available power. Since a single individual can only do so many things at a time, this approach still may not fully consume the power organizations have purchased.

Another line of thinking is that make those “single user” computers into multiuser systems by taking advantage of the great leap in hardware capabilities in a typical PC. The folks in the Intel/UNIX and Intel/Linux communities have been walking down that path for decades. It appears to me that MiniFrame is walking down the same path using a load balancing algorithm on a standard PC with Windows as the operating system. Since the Windows application portfolio is huge, they thought it was a fertile ground for investment in the development of this type of tricky system software.

MiniFrame is offering two different packages based upon their technology. One of these, [SoftXpand Education Suite](#), is designed to allow up to 8 students to share a single standard computer in a very cost-effective and manageable way. The other, [SoftXpand Professional Suite](#), is designed to do the same thing for business people. In the end, both products do the same thing — allow a group of people to share a single standard PC and have the same experience as if they each was using his/her own PC. This would most certainly reduce the cost of hardware, software, installation and a number of other IT costs.

As I thought about this, I was concerned that this would run into restrictions in the terms and conditions Microsoft and others have on the use of their software. After learning more, it appears that what MiniFrame is doing would not be restricted by those Ts and Cs since there is no remote thin client or computer device connected to the system. Only additional standard monitors, graphics adapters, keyboards and pointing devices are added to the basic system hardware. So, from the computer’s point of view, a single “user” (the SoftXpand software itself) has spawned a large number of processes in different user accounts rather than the system communicating with many remote access devices.

It’s a clever approach. What do you think?