

Thanks to

Caught on a planet of peril, he dared challenge its monster rulers... **PBS** presents

# I, CRINGELY THE PULPIT

DECEMBER 27, 2001

The Pulpit I Like It! Baloney Old Hat Tell Me When Pass It On Bob's

## What I Want for Christmas A Supercomputer in Every Garage!

By Robert X. Cringely

Reality has lately been a little too REAL for me. The economy is tanking, we're at war, the national and international situations simply cry out for escape, denial, and delusion. Why worry when you can nerd out, instead? That's when I decided that what I really wanted for Christmas was my very own supercomputer. Doesn't everyone?

Ignoring for the moment the pressing question of why any individual would actually need a supercomputer, I prefer to revel in the fact that it is even possible to have one. We live at a time when processor and memory prices are at historic lows allowing enormous amounts of computing power to be accumulated in the back bedrooms of houses like mine. The minute I realized that I could have a supercomputer, I had to have one. But would Santa come through? Given all the poor children in the world far more deserving of gifts, I decided to take the obligation from Santa's shoulders and build the supercomputer myself.

Building a supercomputer these days is pretty much a matter of throwing a lot of processors and memory into a big box or boxes, then finding some cheap clustering OS (usually Linux) to make it all work together. My role model in this venture is KLAT2, the Kentucky Linux Athlon Testbed 2, which in a recent ranking came up as the 200th most powerful supercomputer on the planet. I love the idea of a supercomputer from Kentucky, but I love even more the clever design of KLAT2, which is a cluster of 64 PCs each running a 700-MHz Athlon processor for a total of more than 64 gigaflops of number-crunching power. Built by University of Kentucky graduate students led by professor Hank Dietz, KLAT2 cost only \$41,000, which is



cheap for a supercomputer.

What's so clever about KLAT2 is the way the 64 separate PCs are linked together. This is the real bottleneck in building a high performance computer at low cost. Processors are cheap, memory is cheap, disk storage is cheap, but networking -- at least really fast networking -- is still expensive. To build a network capable of keeping up with those 64 Athlons, the obvious choice would have been to use gigabit Ethernet adapter cards. But gigabit Ethernet cards are expensive, and at the time would have cost more than the PCs in which they were installed. There had to be a cheaper way of connecting all those PCs together.

So Dietz came up with a whole new network topology that allows cheaper, slower network cards to perform as well or better than gigabit Ethernet. The solution was to put more cheap Ethernet cards in each PC, and then use "channel bonding" to make them all look like a single faster card. Dietz put four 100 megabit-per-second fast Ethernet cards in each PC. Each card has only one tenth the speed of a gigabit card, but not even gigabit Ethernet cards actually carry a billion bits per second. With channel bonding, it turns out that even using three cheap network cards per PC allows greater throughput than a single gigabit card. And fast Ethernet (10base-100) costs about three percent of gigabit Ethernet on a per-card basis, so using four cards per PC still saves 88 percent.

It is easy to use channel bonding for higher performance when the number of machines is low and they can all connect through the same switch, but when the number of processors goes up in the dozens or hundreds, the network topology requires a lot of calculating. This is called a Flat Neighborhood Network, and Dietz had to devise a genetic algorithm to calculate the switch configuration that made for the lowest latency connections.

The other KLAT2 advance is using the Athlon processors' 3Dnow! instructions to bring parallel execution inside each processor, not just to the Linux cluster. 3Dnow! is very similar to Intel's Multimedia Extension (MMX) instructions and can greatly speed up certain calculations. Using 3Dnow! in the KLAT2 required some C compiler changes that led to an across-the-board 3X speedup. Now I wanted to apply those KLAT2 design features to my little supercomputer.

Fortunately, I wouldn't be starting from scratch. In my office, I had the bones of an old Internet business venture gone sour -- six rackmount computers. They used antiquated Cyrix-233 processors, but I mainly valued the systems for their cases, fast Ethernet cards, and 9.1 gigabyte, 10,000 RPM Seagate Cheetah SCSI hard drives. I'd be replacing the ATX motherboards and adding network cards in every case. I would need six new motherboards and a lot of memory.

The process is moving forward only as I can afford it, but so far I have six new dual Athlon XP motherboards, each with a pair of 1.4-GHz processors, a gigabyte of DDR RAM for each machine, and a total of 24 network adapter cards. These will all plug

into a 24-port 10/100 network switch to create a flat network neighborhood. While my supercomputer won't be quite as fast as KLAT2, it will be a lot smaller and cheaper at right around 24 gigaflops, which is comparable to a top-of-the-line Cray T90 supercomputer from 1998. And while that Cray cost millions, my out-of-pocket supercomputer budget is \$6,000.

Unlike KLAT2, the operating system for my little supercomputer won't be Linux. It will be QNX, a real time OS that supports massive parallelism and has very low overhead. QNX is fast! QNX is also Posix compliant, so there is lots of software that almost works under it. And even though QNX is a commercial operating system, it is free for noncommercial purposes like mine.

Beyond using it to heat my office, I plan to keep the supercomputer busy with a video compression project I'm doing as well as further experiments in wireless communication. Having not learned any painful lessons from my long distance 802.11b experiments, I've decided to get even wackier in my attempts to improve Internet connectivity and will be looking into Ultra Wide Band networking. UWB is a form of wireless data communication that uses radio in a completely different way, sending short pulses of energy across the entire zero to 60 GHz frequency band. Not long ago, only spies and Secret Service agents used this stuff, but now there are many companies, including Intel, that are developing UWB chipsets. UWB could replace communications of all types, ending forever our dependence on wires and making worthless the ownership of radio frequencies.

UWB is like magic or quantum mechanics, whichever you prefer. It is immune to interference just as it doesn't interfere with traditional radio signals, so the FCC is considering UWB as an unlicensed service across all frequency bands -- even cellphones and broadcast frequencies. How could they regulate communications they can't even detect? UWB uses one ten thousandth the energy of networks like 802.11b, yet offers the prospect of greater range and greater privacy along with data rates that are presently around 60 megabits-per-second and might eventually hit one gigabit-per-second. UWB is virtually undetectable by traditional radios, since its signals are considered noise -- noise spread across such a wide band as to be beneath the threshold of traditional receivers. UWB uses multipath interference as a form of error correction! What was formerly considered bad is now good. In fact, UWB only works at all because we know precisely where and when to listen. It is based on a complex and very rigidly-structured encoding scheme, and that's where my little supercomputer comes in.

I've decided to call her Wendy.