



PowerPC G4 for Engineering, Science, and Education

**By Richard E. Crandall
Apple Distinguished Scientist
Advanced Computation Group**

October 2000



Contents

- 3 The PowerPC G4 as a new species of machine**
- 3 Velocity Engine and vectorization**
- 4 What ever happened to the dinosaurs?**
- 4 What is a supercomputer?**
- 5 That tired and homely old friend, the mainframe**
- 5 Desktop supercomputing**
- 6 How to envision a gigaflop rate**
- 7 Of galaxies and things**
- 8 In the blink of an eye**
- 9 Cryptography and “big arithmetic”**
- 9 Of birdsong and alien life**
- 10 Network and cluster supercomputing**
- 10 Power Mac G4 successes**
- 11 At last, a good “opportunity vector” for schools**



PowerPC G4 for Engineering, Science, and Education

The PowerPC G4 as a new species of machine

What could be more exciting to a botanist than the emergence of a new species of flower? Or to an ichthyologist, a brand-new species of fish? How about a new black-hole configuration, to an astronomer? As a career scientist and educator, I find myself joining the league of excited professionals, and this time the new animal is the PowerPC G4 processor technology.

There is no question that certain problems from both the academic and industrial sectors can now, finally, be solved because of the vector-processing capabilities of the PowerPC G4. This is a boon not only for industry, but also for schools and research laboratories. Of course, everyone can benefit from the vector-processing advantage—for example, through new and better website graphics, video effects, rendering, encryption, and compression, and generally enhanced system performance. But this paper focuses on how the G4, with its unprecedented desktop speed, presents itself as an emergent species, to the benefit of intellectual pursuit.

Velocity Engine and vectorization

It will be useful to start with an elementary description of what it means to do vector processing and why such processing is special. Simply put, the PowerPC G4 processor is enhanced with extra, very special silicon circuitry called the Velocity Engine. This engine is a full-fledged vector processor; included on-chip in an intimate, symbiotic fashion. For the tremendous processing advantages described herein, the relevant feature of the Velocity Engine is that several calculations—say, four or more—can occur all at once.

Here is a way to envision the tremendous advantage inherent to vector arithmetic. The Velocity Engine performs typical, elementary algebra operations, such as

$$a * b + c$$

(read “ a times b , plus c ”), but on *vectors* instead of plain numbers. You can think of a vector as a multicomponent number, such as the (floating-point) quad of numbers: $a = \{1, 3, 2, 5\}$; and imagine multiplying this by another vector $b = \{2, 12, 4, 40\}$ and then adding yet another vector $c = \{1, 1, 10, 3\}$ to get an answer “all at once,” if you will:

$$\{1, 3, 2, 5\} * \{2, 12, 4, 40\} + \{1, 1, 10, 3\} = \{3, 37, 18, 203\}$$

You can see how the last component becomes 203, namely $5 * 40 + 3$. Clearly, eight operations (four multiplies and four adds) are actually performed here, all at once.



There are, of course, many other vector operations, including ones that shuffle vector elements around, mix up the bits of elements, and so on. Moreover—and this is important in regard to competing processors such as Pentium—the Velocity Engine has an especially efficient ability to “schedule” the vector work so that maximum benefit is taken of the CPU clock. It is thus no surprise that in some isolated cases, G4 technology can sometimes move eight (or even more) times faster than competing PC hardware, with a more typical speed advantage of two or four times, depending on the precise application. Such talent on the part of the Velocity Engine leads to advanced software with unprecedented power, and this is how Power Mac G4 computers emerge as a new species in the world of computation.

What ever happened to the dinosaurs?

The age of supercomputing—the decade of the 1980s in particular—saw the rise and fall of multimillion-dollar supercomputers. These “dinosaurs” of that decade were able to reach beyond megaflops (millions of floating-point operations per second) into the gigaflops (billions of such operations per second). These hallowed machines stood majestic, alone, as the dominant power animals of the era. Just about the time people began to confer on the supercomputers some kind of magical capability (not to mention a new form of humor: “Did you hear about the forthcoming Cray-XYZ? It will do an infinite loop in about 7 seconds.”), the pressure of eight-digit price tags, coupled with the rise of network-based and distributed supercomputing on collections of smaller processors, seems to have brought down those dinosaurs from their once-dominant role. Of course, really impressive supercomputers operating in the teraflops (trillions of operations per second) are still being designed today, yet these most modern variants usually depend profoundly on the parallelization of smaller processors.

We shall see how Apple’s Power Mac G4 amounts to something new: a *commonly available* supercomputer, a new animal that transcends previous categories, in being very fast yet extremely cost-effective. But first, let us turn to a more precise notion of supercomputing, in order to underscore the innovative character of the G4.

What is a supercomputer?

Many definitions of *supercomputer* have come and gone. Some favorites of the past are (a) any computer costing more than ten million dollars, (b) any computer whose performance is limited by input/output (I/O) rather than by the CPU, (c) any computer that is “only one” generation behind what you really need, and so on. I prefer the definition that a supercomputer performs “above normal computing,” in the sense that a superstar is—whether in sports or in the media—somehow “above” the other stars.

Note that “superanything” should not be construed to mean the absolute fastest or best. Because almost all modern personal computers perform in the tens or hundreds of megaflops (more details later on such specifications), and yet these modern PCs cannot solve certain problems sufficiently rapidly, it is reasonable to define the supercomputing region, for today, as the gigaflops region. What this means is that you can do things at a few gigaflops that you cannot conveniently do at 100 megaflops. That is, you can do things in a convenient time span or sitting, or most important, you can do in *real time* certain tasks that were heretofore undoable in real time on PCs. In this sense, the Power Mac G4 is a personal supercomputer: obviously personal (the latest ergonomic and efficient designs for G4 products amplify this) and, due to their ability to calculate in the gigaflops region, clearly above the pack of competing PCs.



On the issue of cost: Some of us still recall the days when Cray time was \$1,000 per *hour*. (This was a common industrial rate; still, academic use sometimes cost in the hundreds of dollars per hour.) It goes without saying that any personal supercomputer that costs in the region of a few thousand dollars—or less—is not only competing with the past, it is rendering the past noncompetitive.

Of course, the state-of-the-art supercomputers of today reach into the hundreds of gigaflops, yet surprisingly, they are typically built from a host of gigaflop-level devices. A system like the Cray T3E ranges from about 7 gigaflops to the teraflop (1,000 gigaflops) region, depending on the number of individual processors. One way to envision the power of a desktop Power Mac G4 is that the low-end Cray T3E at 7.2 gigaflops has the equivalent speed of two G4 processors, each calculating at 3.6 gigaflops (an actual achieved benchmark for certain calculations). By the way, the advertised U.S. list price for the bottom-end Cray T3E is \$630,000 (www.cray.com/products/systems/crayt3e/1200e.html; October 11, 2000). On this basis, one can conclude that the Power Mac G4 is “between” the supercomputers of the 1980s and the very highest-end supercomputers of year 2000, and in fact is about as fast as the low-end versions of the latter, most modern supercomputers.

That tired and homely old friend, the mainframe

Many of us also recall the days when college campuses in general sustained a centralized paradigm simply described as a bunch of “dumb” terminals orbiting around a central mainframe. In the early 1980s, mainframe power was on the order of a few MIPS (or call it megaops, for integer operations, as opposed to megaflops, for floating-point operations). In fact, a standard unit of computing power—called a MIPS-YEAR—is supposed to be based on what an old DEC VAX 11/780 did in one year.

As of year 2000, however, not only can a student have something like Apple’s Power Mac G4 Cube sitting silently and aesthetically on a dormitory desk, but the G4 Cube has the power of *hundreds* of “old mainframes.” Look at it this way: A G4 performs one MIPS-YEAR of computational effort between breakfast and dinner. Thus, the G4 Cube is the equivalent of a “bushel of old mainframes.”

Desktop supercomputing

Let’s talk about the “feel” of a desktop supercomputer experience. When I first used a Power Mac G4 for a serious computation—the dreaded “F24 Problem” described later—it felt very much like the Cray-YMP days. In fact, the response of the Power Mac G4 system’s keyboard and monitor is, if you will, “zippier” than it used to be for a terminal connection to the Cray-YMP (of course, in the old days the closest thing to a “desktop” was a dumb terminal on one’s desktop). To put it another way: Recall the old characterization of a supercomputer, that it would be limited by I/O—say, long I/O delays punctuated by brief “hiccups” of billions of CPU operations each. For many research and engineering problems, that is also the feel of the Power Mac G4: One senses the CPU actions as mere “hiccups,” so fast is the internal machinery.

Sure enough, the performance numbers verify this subjective experience: A good program on a Cray-YMP would run at 300 megaflops (say, 2.4 gigaflops then, on an eight-headed Cray), and now we have G4 functions sometimes running well above the 3-gigaflop threshold. It is true that the dinosaurs had massive bus and memory bandwidth, vast storage, and so on. Nevertheless, for many realistic computation problems the PowerPC G4 with its Velocity Engine conveys a newer and sleeker variant of the feel of—the joy in—commanding a supercomputer.

**How to envision a gigaflop rate**

The wonderful news is that the PowerPC G4 can perform certain tasks—and I speak not of curious, isolated, prejudiced computations but of practical, widely needed tasks—in 1, 2, or even above 3 gigaflops. The peak rating of a 500-megahertz G4 is 4 gigaflops, and a good rule of thumb is the following: One can come near 4 gigaflops for some functions, yet one can generally exceed 1 gigaflop on most functions.

So it is convenient to concentrate on the question, “How do we envision 1 gigaflop?” One way is this: Light travels about 300 million meters per second, which means light would travel, if directed to, entirely around the Earth’s equator seven times per second; the front of the beam would pass your position in a “buzz” of 7 hertz. This means that at a gigaflop rate, the PowerPC G4 performs over 100 million operations on every equatorial pass of the light beam. To put it another way, light travels about one foot in one nanosecond (a billionth of a second). This means that one G4 operation occurs as light travels “about one foot.” Imagine—during one G4 operation at, say, 2 gigaflops, the light having left your monitor has not quite reached your face. It goes without saying that at the practical G4 peak of just under 4 gigaflops, one CPU operation has light traveling a distance more like the width of your hand.

Of course, there are less recreational, more meaningful ways to envision a gigaflop. One way is to consider a very large (full-screen, say) video clip, 1000 by 1000 pixels, which needs to run at 30-frame-per-second movie speed. That’s 30,000,000 pixels per second. So at a gigaflop, we are allowed about 33 operations *per pixel* to meet the prescribed movie rate. Sure enough, many of the modern, transform-based compressors—MPEG, JPEG2000 (motion), experimental wavelet schemes, and so on—involve a few dozen algebraic operations per pixel. What this tells us is the important fact that somewhere near 1 gigaflop, certain practical performance issues become resolved. That is, there are enough operations allowed per unit element to perform in *real time*. It is this kind of thinking that leads one to the conclusion that the gigaflop super-computing boundary has a hard relevance for engineering.

Incidentally, there is an interesting measure of CPU efficiency that, for the purposes of this paper, I will call “V-Factor;” where “V” stands for vectorization (or just as conveniently it can stand for velocity). You can think of the V-Factor as a measure of vectorization efficiency, calculated as the ratio flops/clock. For example, a vector processor having a 200-megahertz clock but performing 1 gigaflop would have a V-Factor of 5. For the PowerPC G4, the V-Factor is computed against, say, a 500-megahertz clock, and for Pentium III we might calculate against a 1-gigahertz clock.

It is well known that Pentium III programmers can likewise invoke some vectorization, and in some isolated cases similar speedups to those of the G4, but it is important to observe that the Pentium III enhancements such as MMX/SSE do not constitute an identical species to the G4. One major difference is that G4 vectors can be used more naturally for multiprecision arithmetic (as in cryptography applications), various algebraic manipulations (as in matrix algebra), and signal processing (as in filters and transforms). Other big differences include the efficiency of G4 scheduling, the availability of registers, and data load/store bandwidth. The way actual G4 routines can approach 4 gigaflops is that a “fused” multiply/add instruction acting on four-vectors, for a total of eight operations per vector, can be scheduled in some cases such that these eight operations occur on every clock cycle. So at 500-megahertz clock, that is what gives the 4-gigaflop peak.



No matter how you look at it, the PowerPC G4 tends to be faster than the Pentium III, even when the former has a lower CPU clock rate (say, 500 megahertz compared with 1 gigahertz). Technically speaking, such comparisons are most favorable to the G4 when computations are cache-bound. Here's yet another way to look at it: The older PowerPC G3 processor's performance is sometimes bested by a factor of four, eight, or even more on the G4. Since many routines on the G3 are roughly comparable to Pentium routines, you can see how dominant is the current G4 performance over that of all other PC hardware platforms.

It turns out that for a great many fundamental routines, one can sustain typically 1 gigaflop to 3.6 gigaflops on a 500-megahertz G4. This results in the following rule of thumb:

$$\text{V-Factor (PowerPC G4)} = \text{about } 2 \text{ to } 7$$

obtained by dividing the number of gigaflops by the CPU clock rate. On the other hand, a typical, optimized Pentium III function yields the ratio

$$\text{V-Factor (Pentium III)} = \text{about } 1 \text{ to } 1.5$$

This result can be obtained from publicly available performance figures on Intel's website (www.intel.com). For example, using Intel's signal-processing library code, a length-1024 complex FFT on a 600-megahertz Pentium III runs at about 850 megaflops (giving a V-Factor of about 1.4), while Intel's 6-by-6 fast matrix multiply performs at about 800 megaflops (V-Factor about 1.3). (Note that one might be able to best these Pentium III figures; I cite here just one of the publicly available code library results. These results do exploit the Pentium III processor's own MMX/SSE vector machinery, and I caution the reader that these rules of thumb are approximations.)

On the other hand, a PowerPC G4 FFT of the same style can be made to run with fairly straightforward vector code at 1 gigaflop, and with more optimization work at up to 2.5 gigaflops, giving a V-Factor of between 2 and 4. As I intimated, PowerPC G4 functions have been written that perform in the region of 2 to 3-plus gigaflops. These include fast matrix operations, convolution, digital filters, and so on. Readers versed in graphics and signal processing will recognize these kinds of functions as not arbitrarily listed, but rather pragmatically important and commonly used. A brief, explicit, consolidated table of the best-known, sustainable performance for a 500-megahertz G4 is the following, obtained from actual laboratory measurements:

Signal function	Gigaflops	V-Factor
1024-point FFT	2.5	5
32/32 matrix multiply	2	4
256-by-2048 convolution	3.6	7

Of galaxies and things

Perhaps you would like to know more precisely how this vector-processing Velocity Engine advantage can actually be used by, say, a scientific programmer. Here is a mental picture of the detailed manner in which G4 vectors might come into play. Imagine a galaxy—essentially a group of a great many stars—spinning on the computer screen. With the G4, the software can treat every star's position in space (x, y, z) *and* luminosity L as a so-called “four-vector.”



We define a star, then, by a four-vector (x, y, z, L) , the point being that the whole construct (x, y, z, L) can be rotated, warped, and shifted around using vector instructions on the G4. It is intriguing that star parameters can be fused in this way, so that operations can act all at once on three-dimensional space and brightness together. But there is yet more opportunity for vectorization in such a model. In scientific terms, one can invoke a so-called “perturbation series” for the gravity calculations, such a series being replete with the fused multiply/add operations at which the G4 excels. For example, the perturbation on a star at radius r from galactic center might take the form

$$a + b * r + c * r * r + d * r * r * r + \dots$$

which can be calculated using something called the Horner rule, which in turn exploits fused multiply/add. In other words, not just the vector design itself but the detailed vector arithmetic can be combined to give gigaflop performance for such a model. The net effect, of course, is that the galaxy spins *much* faster with the vector processing than without.

One of the very first gigaflop examples of the PowerPC G4 was a galaxy model in which 30,000 stars were propagated according to the usual Newtonian gravitational laws; these laws in perturbation form require about 80 operations per individual star. The 500-megahertz G4 implementation processes that entire galaxy in less than 0.002 second, giving over a gigaflop. The net effect is a spinning galactic ensemble, which on an already impressive G3 would stutter a tad, yet on the G4 the galactic rotation of the entire ensemble appears smooth. Such is the advantage of a physical scenario that allows modeling via multicomponent vectors and special logic architecture.

In the blink of an eye

There are further picturesque variants of the whole four-vector theme, variants important to both academia and industry. In the world of image processing, there are modern algorithms called wavelet algorithms. Wavelet literally means “little wave,” and the idea is to break an image down into its wavelet components. In fact, the wavelet transform is a recipe for building the image back up from a kind of “hierarchical dust,” a recipe that lends itself to compression and other important image operations. Here again, the PowerPC G4 is radically superior to “scalar” (that is, nonvector) machinery. Think of a color pixel in an image as a four-vector $RGBA = \{\text{Red, Green, Blue, Alpha}\}$, where Alpha is the opacity channel, used for image overlay. You can imagine, then, performing algebra on this color four-vector. Indeed, a G4 wavelet transform proceeds five times faster than a scalar G3 one. The reason for the greater-than-four speedup is that there are, as we have said, more advantages to the G4 than just the fourfold vectorization.

Incidentally, the upcoming image standard JPEG2000 will involve wavelets, so this G4 advantage will soon find its way into all sorts of scenarios. For one thing, future website images will likely use the newer and sleeker wavelet methods, for very rapid image decoding. And the G4 will make things all the more efficient. One wavelet transform that has been implemented will perform (for 500-megahertz clock) an entire wavelet-inverse transform followed by a traditional YUV-to-RGB conversion in 1/300 of a second, for a 320-by-240 image. That is an order of magnitude faster than one would need for real-time decompression for that wavelet scheme. This performance measures out at about 1 gigaop (1 billion integer operations per second).

**Cryptography and “big arithmetic”**

Again on the subject of vectorized integer processing, there is the burgeoning world of cryptography. It is well known these days that cryptography involves large numbers, like prime numbers. It is not an exaggeration to say that most people have used prime numbers, at least indirectly, in web activities such as mail and credit card transactions. For example, when you order garden tools over the web, your credit card transaction may well involve a pair of prime numbers somewhere in the ordering chain.

There is also a whole field of inquiry, called computational number theory, that is a wonderful academic field, attracting mathematicians, engineers, and hobbyists of all ages. Whether the interest is academic or commercial, it turns out that the G4 is quite good at this “big arithmetic.” Here is an example: When timed against any of the public arithmetic packages for Pentium (there are a dozen or so good packages), the G4 beats every one of them at sheer multiplication. In case you are familiar with timing for big arithmetic, a 500-megahertz G4 will multiply two numbers each of 256 bits, for a 512-bit product, in well under a microsecond. And here is a truly impressive ratio: In doing 1024-bit cryptography, as many crypto packages now do, the G4 is eight times faster than the scalar G3. That huge speedup ratio is due to many advantages of the vector machinery.

But there is more to this happy tale: The G4 can perform arbitrary vector shifts of long registers, and these too figure into cryptography and computational number theory. There are some macro vector operations that run even 15 times faster than their scalar counterparts, because of the vector shifts. In such an extreme case of vector speedup, it is not just the vector architecture but also the unfortunate hoops through which the scalar programmer must sometimes jump—hoops that can happily evaporate in some G4 implementations. (Technically, arbitrary shifting on scalar engines must typically use a logical-or-and-mask procedure on singleton words, which is a painful process.)

Of birdsong and alien life

There is a 35-year-old algorithm construct that is so ubiquitous that literally thousands of papers have been published about it, hundreds of books are devoted to it, and some university students and faculty are studying it, while engineers *use it* as you read this. We speak of the fast Fourier transform, or “FFT.” The ubiquitous FFT essentially generates the frequency spectrum of a signal, and is used to analyze birdsong or whalesong, provide voiceprints, process data, look for patterns, and on and on, with applications even to the search for extraterrestrial intelligence (SETI). The well-known SETI@home Project allows volunteers to process radio telescope data at home, searching for telltale patterns that might arise from living beings out there in the cosmos. At the very core of the processing software is, yes, an FFT. It turns out that the PowerPC G4 is well-suited for certain variants of the FFT, and can provide speedups of five times (sometimes even more than that) over the scalar G3. This is a fortunate circumstance, not only for such endeavors as the SETI@home Project, but also for the myriad academic uses of FFT. (On a humorous note, when chief astronomer D. Werthimer of the SETI@home Project heard of the G4 processor’s advantages, he suggested a commercial tome: “When ET phones, the G4 will be listening.”)

On the quantitative side, we have noted that the G4 and Pentium III FFT speeds differ considerably. (Technically, the difference is especially evident for commonly used medium signal lengths such as 1024 or thereabouts; the speed ratio of the G4 over the Pentium III depends



somewhat on signal length because of memory constraints.) Again we echo the theme: FFTs in the gigaflop region give rise to new applications in real time. Speech recognition, for example, is often FFT-based in part, and we all know how difficult are recognition and related artificial intelligence problems.

Network and cluster supercomputing

As has been said, one form of selection pressure giving rise to the extinction of the old supercomputers was the advent of network-based supercomputing. One can build great networks of desktop computers. In that mode, many (these days that could mean literally thousands of) desktop computers work together on a problem. Computational problems to solve in this way include the well-known prime number searches; cryptographic code cracking; “large” scientific problems in biology, chemistry, and physics; and the already famous SETI@home Project.

We have all heard by now of the Genome Project, which involves the detailed mapping appropriate to human genetics. Needless to say, that is a profound computational task; and even though success has recently been reported, it is evident that such computations will never end. For one thing, new and faster computers will always be needed to resolve, track, and model isolated gene defects for individuals and groups. Every new Genome milestone of the coming century will give rise to more, not less intense computational requirements.

The point of these observations is that the G4 will boost, several times over, the inherent processing power of these vast networks. To convey an idea of scale: A network of 300,000 volunteered G4 computers would have a peak computational power of 1 petaflop (1 quadrillion floating-point operations per second). Such power could in principle, if harnessed in proper parallel fashion, calculate the entire, synthetic Disney/Pixar movie *Toy Story II* in a few minutes. That kind of intuition aside, it is true that many of the hardest computations ever performed fall into the petaflop region, meaning that in a convenient time span for research (one year, say), one needs to move at a petaflop clip.

Power Mac G4 successes

Around the country, among national laboratories, universities, and developer sites and also within Apple’s Advanced Computation Group (ACG), there have been performance successes on real computational problems (as opposed to fundamental benchmarking). The research papers of the ACG, showing results on FFTs, wavelets, multiprecision, and matrix algebra, are available on the web at developer.apple.com/hardware/altivec/acgresearch.html.

The following are some brief descriptions of exemplary G4 successes from various user sectors. Note that, even as I write this brief summary, new applications are pouring in, as more and more deep problems begin to be assailed with the PowerPC G4.

- One of the university successes is the interesting AppleSeed project being run at University of California at Los Angeles. Reported by D. Dauger from that effort is a performance—on a graphical fractal generation experiment of over 20 gigaflops from 16 Power Mac G4 computers (at the reduced clock of 400 megahertz, in fact)—that amounts to well over 1 gigaflop per unit in their cluster mode. The details, along with comparisons to Pentium III, IBM, and Cray benchmarks, are provided at exodus.physics.ucla.edu/appleseed.
- As for the engineering sector, more precisely the engineering/academic interface, an interesting project at the University of Minnesota uses Power Mac G4 computers for high-performance analysis of seismic data. Seismic data pertains not only to earthquakes and volcanic activity,



but also to the search for natural resources. In the words of Dr. N. Coult of UMN, “High [oil] prices at the pump have an enemy in the G4.” His technical writings are available at www.ima.umn.edu/~coult.

- From the Advanced Photon Source of Argonne National Laboratory, D. Mancini reports the use of G4 systems for research into high-throughput X-ray analysis. Their general effort in tomography is described at www.itd.anl.gov:80/techtour/microtomography.html.
- From Sandia National Laboratories, S. Spiers reports that the G4 is being used for research involving “computation with decentralized collectives of intelligent agents, for applications where robustness is paramount and the environment may be hostile—like on the Internet.” One way to envision such research is that it essentially moves a great deal of Internet security issues into the “deep computation” domain. This is a great example of how deep computations can enjoy a practical importance well beyond, say, the cultural allure of sheer depth. As we have pointed out, the G4 excels in integer arithmetic, especially the multiprecision variety, which is of paramount relevance to cryptography and security applications. An overview of the research effort is at www.sandia.gov/media/NewsRel/NR2000/agent.htm.
- According to C. Hunter at NASA Langley Research Center, the Power Mac G4 runs vector computations several times faster than do traditional scalar workstations. During evaluation of various systems for simulations in computational fluid dynamics, the G4 was found to enjoy a potential cost effectiveness (calculated as speed divided by cost) up to eight times higher than other workstations. General information on the NASA Langley effort is at www.larc.nasa.gov.
- In the sector of so-called pure academic research, here is a rather unique example. It is especially pleasing to me that the Power Mac G4 was used as an integral part of the deepest computation in history, for a 1-bit (yes/no) answer. This dreaded “Problem of F24” involves the determination of whether a certain gargantuan number—called F24, possessed of about five million decimal digits—is prime or not. Various researchers around the country participated in this massive computation, which by the way was about as extensive as the rendering of one of those Disney/Pixar movies. (One can joke that you can get either a full-length synthetic movie or, for the same essential effort, this 1-bit answer.) One of the participating machines was a Power Mac G4, which technically speaking was doing a very large integer convolution at unprecedented speed. I hope this kind of “pure” computational example, although it does have its allure, will not detract from the other observations about the kind of supercomputer performance that is truly needed in this day and age.

At last, a good “opportunity vector” for schools

I would like to lay out a notion of what I will call an “opportunity vector.” Such a vector describes the opportunity offered to, say, a student or teacher. All through the 1970s, ’80s, and ’90s, we professors were forced to say, “Well, you can have this much performance, but it will cost you this much; and you can have this much graphics resolution/speed but the equipment will be this big,” and so on. In other words, we have always had to hedge, to offer solutions saddled with handicaps. It has been as though our opportunity vector were a sum of highly orthogonal components, adding up to a disappointing total vector. The insufficient educational efficacy of previous computing resources has, for some of us professors, long been a cause for embarrassment or at least regret.

Anyone who has been involved with higher education knows that in the liberal arts sector, and in many universities, the problem of offering supercomputing power—meaning at the rough measure of 1 gigaflop—to students has been problematic, if not impossible. Surely in the 1980s



a budget of \$10 million would generally, perhaps naturally, go to other causes. But computation affects all of our lives more and more—witness the examples of advanced medicine design via supercomputing, or the profound analysis of economic trends—so it is a wonderful thing that something like the G4 is now available to students generally, not only around the world but at most educational levels. Soon enough, parents will be able to send their children not just to schools that have supercomputers, but to schools that have something approaching one super-computer *per student*.

I recall a prominent plasma physicist of the mid-1980s who said his “life had changed” when he “finally had available 25 megaflops.” He meant that the complex models of fluid flow could not be fathomed in a convenient time span until he had 25-megaflop machinery on which to model the physics. Given that sentiment, imagine the opportunity for a student in the coming millennium, who can have at her fingertips forty or more times that physicist’s “dream” power.

There is a fascinating cultural aspect of this “desktop supercomputing” paradigm. As soon as PCs became widely available, they were very slow, because of course the first machines to break through any low price threshold will likely be slower machines. But with a new machine species like the G4, a decade-long tradition of “low-cost, sluggish” equipment is in many ways canceled, and that is a happy thing. That happy thing yields, at last, a good opportunity vector. Now we can say, “Well, you can have gigaflop-supercomputing power, at low cost, with efficient design, aesthetic and not bulky.”

We might pose the question, “Why would every student need a supercomputer?” The answer is that, as soon as the sluggish-cheap paradigm is fully blunted, some will be surprised at how often very deep computation will be required, even of individuals. Many of us professors have found out over the years (of disappointment, of insufficient opportunity vectors) that students of today—whenever “today” is—need as much computational power as we can offer. We have mentioned cultural breakthroughs (involving medicine, industry, the Internet, pure research) to which supercomputing applies, but we have also pointed out the need for real-time solutions (in sound, video, conferencing, general communications). It goes without saying that whatever the future of deep computation, it is the current student generation that will be the deep-sea divers in that future.

Allow me to cast a final point in metaphorical terms. I have a crazy dream pertaining to that dreaded day when I am rushed into an X-ray ward for whatever reason. I envision the signal-processing programmers, radiologists, and physicians, working with their as-yet unimaginable computational-medical apparatus. My hope is this: That those professionals will have been trained, back in their student days in the early 2000s, on desktop supercomputers such as the Power Mac G4. Why such a dream? Because those professionals will have learned, early in their careers, how to deduce important and telling things—facts that really do, after all, matter in life—by way of deep computations.