

What Is Massively Parallel Computing, and Why Is It Important?

WHY SHOULD THERE BE AN ISSUE OF *Dædalus* devoted to computation? Because there is a significant technology transition that is taking place in computers, and since it radically changes the cost and capabilities of information processing, it is likely to change our lives. The new technology is called *massively parallel* computing. On one hand, it is just a set of engineering ideas: a way of reorganizing the structure of the computer so that it can do many things at once. These ideas are simple and, in retrospect, obvious. On the other hand, the implication of these ideas is that the computer revolution has not yet begun to reach its limits. By the end of this article I will be writing about strange and unlikely sounding things, like home robots and virtual worlds. These things sound unlikely because significant technical advances—such as the automobile, television, the computer—generally have surprising consequences. It is always easiest to believe in a future that is a minor extrapolation of current-day trends. Such an extrapolated present is unlikely to happen in a time of rapid technological change. To help make this point and to set the ground for explaining the technology of parallel processing, I will begin with a story of how some simple ideas about reorganizing the flow of processing had a profound effect on our present-day lives.

In the 1790s, just before Thomas Jefferson was elected president, the United States was in danger of getting into a war with the French. At that time, most of the rifles used by the US militia were made in France. In hopes of encouraging the development of an American

W. Daniel Hillis is Chief Scientist at Thinking Machines Corporation.

2 W. Daniel Hillis

arms industry, Congress allocated funds for the purchase of twenty thousand domestically manufactured rifles. In those days rifles were made by hand, one at a time. Each rifle was slightly different, with its own individually crafted parts. The largest manufacturers were able to build only a few hundred rifles per year.

A gentleman in New England claimed that he could build and deliver ten thousand of the highest quality rifles, although he had never built a rifle in his life. He was Eli Whitney, the inventor of the cotton gin. Whitney planned to build the rifles out of interchangeable parts manufactured with the aid of water-powered machinery. Instead of building the rifles one at a time, parts would be produced from standard templates in batches of hundreds. By organizing his “manufactory” to take advantage of the economies of scale, Whitney believed that he could produce rifles at a rate and quality never before imagined.

Whitney’s competition was understandably skeptical, but Thomas Jefferson, who was familiar with the latest innovations in French arms manufacture, was able to see the potential benefits of the scheme. Whitney was awarded a large federal contract and advanced much of the necessary capital. His methods worked and the arms that he eventually delivered were widely acknowledged as the highest quality rifles made in America.¹

The methods that Whitney pioneered were later called *mass production*. Today they are applied to the manufacture of most high-volume products. Whitney’s methods now seem obvious. It is hard to imagine that almost all manufactured goods at the time, from wagons and clocks to Whitney’s own cotton gin, were not mass produced; yet they were not.² It is difficult for us to imagine how such seemingly simple ideas, like the use of interchangeable parts or the specialization of workers in different stages of assembly, could have had such a profound effect on our society. Yet mass production was important. It was important because it created an *economy of scale* in the processing of material goods, which in turn caused fundamental changes in the manufacturing sector of our economy and in the type of products it produced.

We are now seeing the emergence of a technology which will change the economics of information processing in much the same way that mass production changed the economies of manufacturing. This new technology is massively parallel computing. Mass comput-

ing is analogous in many ways to mass production. They are each a collection of techniques for organizing processing to take advantage of economies of scale. In the case of mass production, it is physical material that is processed; in the case of computing, it is information. Because both mass production and mass computation create economies of scale, they change the economics of production and encourage the centralization of certain functions; this in turn leads to changes in the types of products manufactured and the methods of distribution utilized. In both cases the range of applicability turned out to be much wider than initially expected.³ Most important, both technologies have created the possibility and economic practicality of fundamentally new types of services and products.

The other critical thing about massively parallel computers is that they are *scalable*: it is always possible to build a more powerful machine by simply adding more processors. This means that there is no obvious upper limit on the computational power of the machines that we can build. This opens some possibilities.

Before speculating on what some of these new possibilities are likely to be, I'd like to first describe some of the methods of massively parallel computing. I will use the analogy to manufacturing because each of the principle methods of mass information processing has an analogy in mass production. In both a computer and a factory the basic method of achieving high throughput is to do more than one thing at once. There are basically three different techniques of organizing concurrent processing: pipelining, functional parallelism, and data parallelism. Each of these have their counterparts in a factory and in a computer.

One of the simplest methods of coordinating parallel activities is *pipelining*. It is exemplified by the sequence of operation on an assembly line. In an automobile factory, for example, the wheels of a car are attached at one stage of the assembly line, while the windshield is inserted at the next. The operations are scheduled sequentially and balanced in such a way that each takes approximately the same amount of time. A large number of vehicles advance through different stages of the sequence simultaneously.

The technique of pipelining is applied in most modern computers. For example, one stage of the pipeline may be preparing the data, while another is performing an addition, while another is storing the previous result. Some computers have dozens of operations "in the

pipe” simultaneously. So-called vector computers work according to this principle.

Another technique of parallelism that can be used in conjunction with pipelining is *functional parallelism*. In an automobile factory the engine is assembled on one assembly line while the frame and body are assembled on another. The two lines proceed independently until the engine is placed into the body. Functional parallelism is also used in computers. For example, many modern computers have separate addition and multiplication units which can operate concurrently. Even single chip microprocessors can incorporate this type of parallelism.

Pipelining and functional parallelism are both useful techniques, but they are limited in the degree of parallelism they can achieve. They are not scalable. In an operation that has only ten steps, not more than ten things can be done simultaneously. In both the factory and the computer, there is far more to be gained from performing the same operation to many things concurrently than in doing many different operations at once. When performing the same operation on many items at once, the potential for parallelism increases with the amount of material being processed, rather than with the number of steps in the processing.

In computers, performing similar operations on many elements at the same time is called *data parallelism*, because it exploits parallelism in proportion to the quantity of data. This has the potential of improving processing rates not by just factors of ten, but by factors of ten thousand. The material counterpart of data parallelism can be found in high volume manufacturing. A machine that molds plastic forks forms hundreds at a time. A printing press draws all the letters on a page in a single operation. The millions of transistors on the microchip are fabricated and wired simultaneously. This type of coordinated parallel processing leads to the dramatic cost reductions of high-volume manufacturing.

Data parallelism is also the greatest source of economy of scale in computing. For example, a massively parallel computer may have sixty-four thousand processors that can add together sixty-four thousand numbers in one step. Like the letters on a printed page, not all the numbers need to be identical as long as the operations being performed are similar. If the operations to be performed are different on each element then sequences of *masking* steps may be used. This is like printing a page with some letters in red and some in black,

using one step for each color of ink. Similar techniques are used to process data within a parallel computer.

The beautiful thing about data parallelism is that the opportunity for doing more than one thing at once increases with the amount of data being processed. The power of the solution scales with the size of the problem. A conventional computer takes twice as much time to process twice as much data. A massively parallel computer can often process twice as much data in the same amount of time by applying twice as many processors.

One of the most important design principles of mass production is the integration of storage and processing. In the workshop of an individual craftsman, the storage of materials is usually physically separated from the area of manufacture. The worker moves to the storage area to pick up a piece of wood or metal and takes it to the bench or the machine to work it into shape. As long as the operations that are being performed on the individual pieces are relatively time consuming, the time spent traveling to fetch the materials is not a limiting factor. If by some magic the operations of fabrication were sped up by a thousandfold, then the time traveling to fetch the material would become the limiting factor in the speed of processing. This is exactly what has happened in computers.

A conventional nonparallel computer can be divided into two parts, a *memory* for holding the data and a *processor* for transforming it. During normal operation of a computer, data is repeatedly moved from memory to processors, and the results are returned to memory. All this moving of data back and forth between memory and processor creates a significant bottleneck for the operation of a conventional computer, putting an upper limit on the speed at which it can process data.

The solution adopted in a massively parallel computer is the same as that in a modern factory: integrate the facilities for storage and processing. Instead of physically separating memory and processor (store room and workbench), bring the material that is to be operated upon near to where it is processed. A modern factory accomplishes this through a system of conveyer belts and locally distributed bins of parts. Large quantities of materials are moved at a time and held locally while being processed. A parallel computer accomplishes a similar effect by distributing the memory physically throughout the computer near the places where the data is being processed. In

computer parlance, this is called a *distributed memory*. A massively parallel computer contains tens of thousands of simple processors, each with its own local memory.

Distributing the data to individual processors means that parallel computers must implement one significant function that has no real counterpart in mass production: communication. When large numbers of data elements are processed they usually cannot be processed independently; the computations interact. For example, in calculating the flow of air over an airplane wing, the calculations describing the flow of the air in one part of the wing must interact with others because one part of the wing affects another. To model the interactions between various parts of the wing the processors must communicate.

Communication is the most difficult part of parallel computation, and much of the initial skepticism about the applicability of the technology was based on concerns that many patterns of data interaction would force sequential processing of the data. As it turned out, the worry proved unfounded. Most patterns of interaction that occur in practice can be done in parallel. Still, the part of a massively parallel computer responsible for communication is the most expensive and complex part of the machine.

Most of the engineering in a massively parallel computer is not in the processors but in the switching system that allows them to communicate. In a Connection Machine, for example, a five-foot cube contains 65,536 processors and associated memory. Each silicon chip contains sixteen tiny processors and the memory associated with the chip is sitting in the same board a few centimeters away. Since 65,536 different memory locations can be accessed simultaneously, the processor to memory bottleneck is eliminated. The cube also contains a communication system that is the equivalent of a telephone system with sixty-five thousand subscribers, each of whom places thousands of calls per second. Miniature switching stations in the same chips as the processors steer messages through a web of hundreds of thousands of wires connecting various parts of the machine.⁴ Massively parallel computers of the future will require even greater network capacity.

What impact will massively parallel technology have on our lives? Like mass production, mass computation dramatically changes the economics of large-scale processing. Mass production changed the world; so will mass information processing. The changes brought

about by the adoption of mass production took place over a long period of time. After Whitney and others proved the success of mass production methods on rifles, it was more than fifty years before these methods were applied to the manufacture of sewing machines and bicycles. It took even longer for the methods to be applied to the production of loaves of bread and residential housing. Centralizing the production of these goods required the reorganization of systems of marketing and distribution, changes in the product being produced, and sometimes even in the taste of consumers. For example, it must have been “obvious” in 1790 that the methods of mass production would never be applied to the decentralized activity of baking bread. Doing so required significant changes in our methods of food distribution, the development of methods of packaging and food preservatives, and considerable change in the tastes and habits of the bread-eating public. Such changes require decades.

Mass production also enabled the creation of entirely new types of products such as telephones and automobiles, and brought luxury products such as clocks and windup toys into common use. The changes with the greatest impact took the longest time, because they required the most radical adaptations of society. Centralizing productions required the development of transportation capable of bringing the raw materials to the factory and the finished goods to the consumers, as well as the development of methods of mass advertising to ensure that large numbers of people would be willing to buy the same product. These changes required a long period of time and the potential of mass production required their development. The future of mass information processing is likely to be very much the same.

The first changes will take place in areas where large quantities of information are already being processed in one place. Examples include large-scale scientific and engineering processing and the management of large industrial and government databases. In many such applications massively parallel processing can be “dropped in” to an existing system with very little ancillary change. A typical example of a present-day application of massively parallel processing is its use by oil companies to process seismic data. These companies look for oil underground by creating acoustic shocks at the surface and measuring their underground reflections. These signals are converted to numbers that are analyzed to create a picture of the geological structures underground. Hundreds of millions of numbers

are required to represent the measurement and, using conventional computers, the calculations can take days or even weeks. By taking advantage of the speed of a massively parallel computer they can be reduced to a few hours. This reduced processing time can be used to reduce costs but, more often, is used to improve the quality of the output, either by asking more questions of the data, or by using more sophisticated algorithms that provide a more accurate picture.

Several major oil companies are already using massively parallel machines for seismic processing modeling. Since finding new petroleum deposits is very valuable, the cost of purchasing and programming a massively parallel computer is easily justified. The primary limitation on its adoption in an application of this type is the availability of trained personnel. There is not yet a large base of existing applications software for parallel machines, so most users today run internally developed software. Parallel computers are inherently no more difficult to program than sequential computers, but present-day programmers are trained on sequential machines. When Eli Whitney built his first rifle factory, he deliberately set it up in a location where the available workers had no experience in existing methods of arms manufacture in order to avoid having to untrain workers of bad habits.⁵ A similar trend seems to be taking place in parallel computers. High school students find concepts of parallel programming easy to learn, but programmers with years of experience sometimes find parallel programming counterintuitive. Although the problem will eventually disappear, it is the greatest near-term limitation on the utilization of massively parallel technology.

In some cases the use of parallel machines can actually reduce the need for programming. For example, the US Census Bureau recently experimented with a Connection Machine for use in classifying long-form census applications. The problem involved classifying respondents according to occupation and industry category. The census had 28 million natural language responses with which to classify individuals to one of 232 industry and 504 occupational categories. In the study the Census Bureau evaluated two different methods of classifying with the aid of computers. The first method, using conventional computers, required sixteen programmer-years to develop an expert system that was able to process about half of the forms automatically. The second method took advantage of a massively parallel computer to classify returns by comparing them to

an existing database of a hundred thousand previously classified examples. This “memory-based reasoning” program required less than one-twentieth of the programming effort to implement, and yet it was able to accurately classify a larger proportion of the returns. This method of automatically learning from a large database requires so much computation that it would not be practical without a massively parallel computer.⁶

Other current applications of massively parallel computers include the design of aircraft, the prediction of weather, the production of animated special effects for films and commercials, the simulation of physical phenomena ranging from the subatomic structure of particles to the large-scale structure of the universe, modeling of mechanical structures such as crashing cars, breaking secret codes, and searching for information in large databases of documents. What these applications all have in common is the large amount of data being processed together in the same place. It is relatively straightforward to apply massive parallelism to problems of this type. But most of the potential beneficial applications of massively parallel technology do not already have the data in one place, because, at the moment, information processing is generally distributed whenever possible. The current trend is that most information processing is moving away from the central mainframe computer and toward the workstation or PC on the desktop. This is currently the case because of the diseconomies of scale in conventional nonparallel computers. In conventional computers, the machines with the highest performance are the least cost effective. For example, a large mainframe capable of executing 40 million instructions per second may cost \$10 million, whereas a workstation capable of executing 4 million instructions per second may cost only \$100,000. The performance differential is 10 to 1, but the cost differs by a factor of 100.

In recent years, this diseconomy of scale in conventional computers has forced a trend toward decentralization. Since workstations and personal computers provide more cost-effective computation than a sequential mainframe computer, as much of the load as possible is shifted away from the central mainframe. The computer system in a state-of-the-art computer facility is organized around a network that connects large numbers of independent workstations. Central databases are still kept on large machines, called *servers*. They are accessible by the workstations via the network. Large computations

are still executed on the large machines, but such centralized computing is avoided whenever possible because of its high cost.

Most experts assume that the trend toward decentralization will continue, and that a larger and larger portion of the computing power will end up on the desktop. I believe they are wrong. For one thing, massively parallel computing does not exhibit the diseconomy of scale that caused decentralization in the first place. With a massively parallel computer performing a calculation, a large-scale computer can actually have lower cost per calculation than a number of distributed individual computers. This reverses the economic impetus for decentralization. Also, the workstation on the desk makes the centralization of computation and data storage more practical since it provides a user friendly interface into the central computer. In the future, I believe most desktop workstations will specialize in that portion of the computation directly associated with user interface, such as graphics and audio processing, while most computations and data storage will be handled centrally on parallel machines.

In this future, the workstation becomes the user's window into the network. Since the time required for the computers to communicate is imperceptible to humans, the interaction between the workstation and the parallel machine will be invisible. As far as the user is concerned, the machine on the desktop will behave as if it has the power and the data of the central parallel machine. For better or worse, this will make us even more interdependent than we are today.

If you find this image of centralization hard to accept, consider that a similar economy of scale has determined the organization of the electrical utility system. While it would be possible, in principle, for each home to have its own electric generator, it turns out to be more convenient and economical for large numbers of homes to share a common resource. By sharing the electric utility we are able to spread the capital cost of high peak capacity among many users. The result is that everyone connected to the electrical network gets access to as much electricity as they need. This level of service would be too expensive to provide in a distributed system.

The same principle of load sharing is applicable to computations on parallel machines. Unlike conventional computers, they can be scaled cost-effectively. If a user occasionally asks a very difficult question of the computer, then it is far more economical to draw on the power and data of a central system, than to provide that power

and data locally. Imagine, for example, an engineer designing a bridge. Most of the engineer's time is probably spent drawing individual pieces, writing specifications, or interacting with other humans. Occasionally the bridge needs to be simulated to see if it can carry the required load or survive an earthquake. These computations are difficult. Putting a machine on the engineer's desk with the ability to perform the simulation quickly is not the best way to solve this problem, because the full capacity of the machine would rarely be used. Since such calculations require a large amount of data and a large amount of communication between various parts of the computation, they are most efficiently done in a centralized parallel machine where these resources can be shared.

Discussion of centralized systems raises issues of privacy, security, and individual control. These are very real concerns, but there is no fundamental technical reason why they cannot be addressed in the context of a centralized system. Private data and messages can be kept and transmitted in encrypted form, protected by a secret code from unauthorized access. (It is possible to make codes that even the fastest parallel computer cannot break.) Rules to limit access and protect data integrity are probably easier to implement in a central database than in a distributed one. Since the central mechanisms are shared, they can be more elaborate. This is why banks are safer than mattresses. Some users with special reasons to distrust the system will keep their data in isolated small computers, just as some people today keep their money outside of the central banking system. But the dominant mode of information storage and large-scale processing will be centralized. The result will be that the typical computer user of tomorrow will be no more limited by lack of computational power than the appliance user of today is limited by lack of electrical power.

High-bandwidth optical fibers will spread throughout the city of the future, linking every office and home into the network. The "net" will provide almost instantaneous access to the large massively parallel computers that are the repositories of data and the engines of processing. Every individual will have access to the most powerful computers in the world for business or recreation. The possibilities that this will create for new products and services are so wide ranging that they are difficult to imagine. I will discuss only two examples: the home robot and virtual worlds.

One of the new products made possible by everyday access to mass computers will be the home robot. By home robot I mean the mechanical domestic servant capable of mopping, dusting, ironing, straightening up the kid's room, clearing the dirty dishes off the table and washing them, doing the laundry, guarding the house from burglars, hooking up the VCR, tuning the piano, and walking the dog.

The mechanical and sensory aspects of this hypothetical marvel are well within the reach of current technology, and in sufficient volume they could be produced at prices comparable to that of a family automobile. The real problem is computation. Rudimentary versions of the necessary software for vision, motor control, recognition, and planning are being developed in laboratories today. But even the fastest computers available are far too slow to accomplish the tasks described above. In addition to the extensive software, such a computer would require access to a large database of commonsense knowledge: knowledge of clothing, video recorders, pets, and pianos. With any foreseeable technology such a processing system and database would be far too expensive to have in the home. It would also be wasted almost all of the time.

Access to a centralized computing utility would make the home robot economically feasible. Fortunately, difficult decisions, like "What task should I do next?" or "Do I throw away this piece of paper?" occur only rarely. A home robot would spend most of its time moving from one position to the next, or doing nothing at all. Simple tasks such as audio and video signal processing and motor control could be performed very nicely by a few microprocessors on board the robot. Whenever a real decision needed to be made, the robot would ask the centralized massively parallel computer. This machine could be shared by hundreds of thousands of homes. It would store a single copy of the database containing the accumulated wisdom of a hundred thousand robots. For example, if you spoke to the robot, a compressed form of your speech would be sent to the central machine for analysis, and instructions for how the robot should behave would be returned to the robot in a fraction of a second. The utility company would send you a bill at the end of the month.

This tale of home robots may sound fanciful. Yet, a century ago, a description of today's electrical network of megawatt utilities powering factories, refrigerators, television sets, and electric toothbrushes, would have seemed just as unlikely. I believe that the changes caused

by these new developments in computing will be even more fundamental. At the risk of straining my credibility further, I will speculate about an even deeper change: the emergence of *virtual worlds*.

By virtual worlds, I mean places that people play, work, and interact that exist only within the computer. Only hints of such worlds exist today so it will be hard to imagine what I'm talking about, but I will begin by drawing attention to some of these existing hints. One of the most common applications of present day parallel computers is simulating reality. For example, an aerospace engineer may build a "model" of a airplane inside the computer and see how it "flies." In this case, the model is an abstract object, something like a three-dimensional drawing of an airplane, and the flight is a mathematical calculation of the physics of the lift and airflow. The result is a moving picture on a screen that looks very much like a wind tunnel simulation in which the engineer can watch the behavior of the "air" curling over the "wing" and measure the "pressure" that it generates on the airplane. In this case, the engineer is using the virtual world that is a simulation meant to mimic the real world in which the real airplane will eventually fly.

A very different hint of a virtual world is the stock exchange. When I first met my wife, she was immersed in trading options. Her office was in the top of a skyscraper in Boston, and yet, in a very real sense, when she was at work she was in a world that could not be identified with any single physical location. Sitting at a computer screen, she lived in a world that consisted of offers and trades, a world in which she knew friends and enemies, safe and stormy weather. For a large portion of each day, that world was more real to her than her physical surroundings.

If you doubt the seeming reality of a virtual world, talk to any twelve-year-old who plays a video game like Nintendo. Have you ever "been" to the dungeons in level three of Super Mario Brothers? Have you "touched" the life-giving magic fairy at the fountain in Zelda or "worn" a magic power ring? If you have not traveled to these places then you have missed part of the shared experience of the modern middle-class American childhood. You may have even seen pictures of a virtual world without knowing it. If you watched a recent film or commercial that includes special effects, you may have been looking at realistic renderings of moving three-dimensional objects that never existed outside the memory of a computer. These

images are so detailed that they are indistinguishable from photographs of actual objects.

The examples I've given so far have been limited in the richness of their action by the fact that they take place on a television screen, but we already have technology available for deeper sensory immersion. For example, most commercial airline pilots train regularly on aircraft simulators which have not only the look and sound but the actual "feel" of flying a real airplane; they can feel the bump when they land. Experimental virtual reality simulators use head-mounted binocular displays to give the wearer a convincing illusion of moving around in a world that exists only within the computer. Two such users can "meet" within such a world and see the movements of each other's imagined bodies, even though their physical bodies may be half a world apart.

Try to imagine the virtual worlds that will be made possible by the power of a shared parallel computer. Imagine a world that has the complexity and subtlety of the aircraft simulation, the accessibility of the video game, the economic importance of the stock market, and the sensory richness of the flight simulator, all of this with the vividness of computer-generated Hollywood special effects. This may be the kind of world in which your children spend their time, meet their friends and earn their living. Perhaps your son or daughter will be an "architect" that designs public gathering spaces in virtual worlds, or a "doctor" that specializes in repairing virtual bodies, or a lawyer that specializes in laws of some particular virtual world.⁷

Whatever you imagine virtual worlds will be like, or whatever I imagine, is likely to be wrong. Profound change is inherently hard to think about. What is almost certainly true is that the future is not just a simple extrapolation of what we know today. Massively parallel computing transforms both the economies and the absolute capabilities of information processing. All that can be said for certain is that this is bound to cause changes and that change is difficult to think about. I am confident that once again reality will go beyond our imagination.

ENDNOTES

¹This account of the rise of mass production is based on the events described in Denison Olmsted, *Memoir of Eli Whitney, Esq.* (New York: Ayer, 1972, reprint

What Is Massively Parallel Computing? 15

- of 1846 edition); and Jeannette Mirsky and Allan Nevins, *The World of Eli Whitney* (New York: Collier, 1952).
- ²A good history of the development and adoption of mass production can be found in David Hounshell, *From the American System to Mass Production, 1800–1932: The Development of Manufacturing Technology in the United States* (Baltimore: Johns Hopkins University Press, 1984).
- ³It is also interesting that both massive parallelism and mass production received their initial impetus for commercial development from the military. In both cases the existing manufacturers were skeptical but they were developed primarily by individuals starting new enterprises rather than the existing manufacturers. In both cases the federal government played a critical role in the development of the technology.
- ⁴A more complete description of the Connection Machine can be found in W. Daniel Hillis, *The Connection Machine* (Cambridge: MIT Press, 1985).
- ⁵Whitney was also interested in avoiding competition for labor and raw materials. An 1802 letter of his which mentions the value of starting with untrained workers is quoted in Mirsky and Nevins, 214.
- ⁶The census project and memory-based reasoning are described in R. H. Creedy, B. M. Masand, S. J. Smith, and D. L. Waltz, *Trading MIPS and Memory for Knowledge Engineering: Automatic Classification of Census Returns on a Massively Parallel Supercomputer*, Thinking Machines Corporation Technical Report (Cambridge: Thinking Machines Corporation, 1991).
- ⁷For some ideas about what virtual worlds may be like, the best source is science fiction. I recommend the work of William Gibson, or a story called “True Names” by Vernor Vinge.