If you wish to distribute this article to others, you can order high-quality copies for your colleagues, clients, or customers by clicking here.

Permission to republish or repurpose articles or portions of articles can be obtained by following the guidelines here.

The following resources related to this article are available online at www.sciencemag.org (this information is current as of January 28, 2011):

Updated information and services, including high-resolution figures, can be found in the online version of this article at: http://www.sciencemag.org/content/331/6016/406.full.html

This article appears in the following subject collections:
Computers, Mathematics
http://www.sciencemag.org/cgi/collection/comp_math
Cloud Computing—What’s in It for Me as a Scientist?

Armando Fox

Many scientists would love access to large-scale computational resources but find that the programming demands of using a supercomputer—as well as the cost and queuing time—are too daunting. Privately owned cloud computers—large data centers filled with computers that mainly run their company’s software—are now becoming available to outside users, including scientists and educators. Companies are leasing their computing resources on demand from a large shared pool to individuals who run their own software on a pay-as-you-go basis. This approach is an example of cost associativity (1): 1000 computers used for 1 hour costs the same as one computer used for 1000 hours. If your problem can be computed in a way that takes advantage of parallel processing, you can now get the answer 1000 times as fast for the same amount of money.

Although companies had long been operating “private clouds” that run programs such as Google Search or Microsoft Hotmail, Amazon was the first to let outside users run software on their computers. For example, Amazon’s Elastic Compute Cloud (EC2), announced in late 2007, allows anyone with a credit card to use any number of computers in Amazon’s data centers for 8.5 cents per computer-hour with no minimum or maximum purchase and no contract. Such an arrangement is possible because these “warehouse-scale” data centers (~50,000 servers, see the figure) are able to provide around assignment deadlines, no outside resources (more than even the biggest schools could provide), and when demand is less (between deadlines), no outside resources need to be purchased.

Initially, cloud-computing hardware was configured primarily for its earliest adopters—Web-based applications—and early attempts to run scientific applications on the cloud gave discouraging results (3, 4). New hardware is now configured for better performance on scientific applications. For example, Amazon’s recently added “cluster computing instances,” priced at $1.60 per computer-hour, run scientific benchmarks 8.5 times as fast as the original cloud hardware, according to experiments at the National Energy Research Scientific Computing Laboratory at Lawrence Berkeley National Laboratory (5).

Cloud computing works best when a problem can be broken down into a large number of relatively independent tasks, each running on its own computer. Software frameworks like Google’s MapReduce (6) and its open-source equivalent Hadoop (7) provide a data-parallel “building block” for expressing such computations (much like a Web design framework allows you to “build” a Web site by filling in the relevant information and functions you want). Critically, these frameworks also hide the complex software machinery that handles inevitable transient failures when hundreds of machines in a cloud environment work on a problem simultaneously. Many of the “success stories” of science in the cloud have embraced Hadoop, and other popular tools such as the statistical package R now feature libraries that integrate with it. However, many problems cannot be easily expressed in terms of map and reduce tasks (mapping parcels out the work, and reduce collates the results). Even when they can, the programming effort required may be substantial.

Most desktop software is not written to take advantage of cloud computing and requires modification before it could harness cloud resources and run faster. However, the popular packages MATLAB and Mathematica are now available in versions that can “farm out” work to a public cloud. Cloud vendors including Amazon and IBM are working with independent software vendors on cloud-friendly versions of popular scientific software.

Department of Computer Science, University of California, Berkeley, CA 94720, USA. E-mail: fox@cs.berkeley.edu

Computer Science

Computational tasks that are inherently parallel, from simulations to student assignments, can be run faster on the data center resources of public clouds.
What about software designed for parallel supercomputers? Programs written to the MPI (message-passing interface) standard run in the cloud, but the performance of such programs is sensitive to whether all the machines run in tight lockstep when working a problem (a characteristic of supercomputers but one that is not necessarily part of the cloud-computing architecture). Still, cloud technology suppliers such as Intel, Advanced Micro Devices, and VMware are adding hardware and software features necessary to improve MPI performance in the cloud, a move that highlights new “buying power” in the scientific computing community.

Many scientific computing problems do not require supercomputer performance but would benefit greatly from modest parallelism—say, tens or hundreds of computers in the cloud. The total “time to answer” may still be quicker, according to Foster (8), even for larger problems because a cloud-based supercomputer can be provisioned and running in minutes rather than hours or days, without waiting in a queue.

Some problems are so data intensive that they demand large-scale computing. The Large Synoptic Survey Telescope in Chile may generate up to 30 terabytes of data daily. At typical long-haul network speeds of around 20 megabits per second (9), 30 terabytes would take more than 4 months and $3000 in network charges to copy to Amazon’s cloud. Many cloud providers now allow users to ship crates of hard drives to be physically incorporated into the cloud infrastructure, an idea championed by the late Jim Gray.

The lure of improved performance has already drawn scientists and engineers to use cloud computing (10) in research on a number of topics, including large-population genetic risk analysis, information retrieval, and particle physics. Cloud computing installations are also being established by academic-industrial consortia [see, for example, (11–13)] to further encourage adoption of cloud computing by scientists.

References and Notes

GENOMICS

The Genomic View of Bacterial Diversification

Mark C. Enright1 and Brian G. Spratt2

Bacteria have unusual and variable sex lives, ranging from near-celibacy to evident promiscuity. “Sex” in bacteria involves a recipient bacterium replacing, by homologous recombination, small regions of its genome with corresponding regions from other strains (1). Studies over the past 20 years have demonstrated that in relatively promiscuous bacterial species, frequent recombination can drive rapid diversification of strains. In more celibate species, diversification is much slower and depends on the slow accumulation of point mutations (1, 2). Sorting out the roles that these processes play in the pace and pattern of bacterial evolution, however, has proven problematic. On page 430 of this issue, Croucher et al. (3) demonstrate a powerful new approach to the problem. By comparing the genomes of many isolates of a single strain of a bacterium that causes pneumonia, they were able to rapidly obtain a comprehensive evaluation of the role that recombination, point mutation and other genetic processes played in its diversification. The approach offers insight into how this pathogenic strain can rapidly evolve resistance to antibiotics and evade future vaccines, and promises to help researchers better understand how this resistant strain may have spread globally.

It has been difficult to estimate the rates and patterns of recombinational replacements by comparing the genomes of distantly related strains of bacterial species, particularly when recombination has been frequent. Similarly, it has been problematic to identify those parts of the genome that have not been involved in recombination and can provide reliable information about phylogeny (the bacteria’s evolutionary history). Most of these problems can be overcome, however, by comparing the genomes of isolates that have a very recent common ancestor, such as isolates of an antibiotic-resistant strain of a bacterial pathogen, which must have emerged within the antibiotic era. These isolates will have accumulated a relatively small number of mutations and recombinational replacements, allowing researchers to more easily determine the number and size of the replacements. The ability to construct a reliable phylogeny, by identifying variation due to point mutations, is also greatly simplified. Differences among isolates can be mapped onto a tree and correlated with genomic differences, such as shifts in antibiotic resistance profiles or changes in virulence or antigens targeted by newly introduced vaccines.

Sequence data can provide information about the dynamics and geographic spread of pathogens. Such an analysis of very recent events, however, requires a pathogen that evolves rapidly enough to enable suffi-