SETI@HOME—MASSIVELY DISTRIBUTED COMPUTING FOR SETI

By Eric Korpela, Dan Werthimer, David Anderson, Jeff Cobb, and Matt Lebofsky

SINCE RADIO'S EARLIEST DAYS, PEOPLE HAVE CONSIDERED THE POSSIBILITY OF DETECTING SIGNALS FROM AN EXTRATERRESTRIAL CIVILIZATION—AND SINCE THE ADVENT OF RADIO ASTRONOMY, THE TOOLS TO PERFORM SUCH A SEARCH HAVE

existed. Starting in the late 1950s, researchers have been performing progressively more sensitive searches, but each search has been limited by the technologies available at the time. As radio frequency technologies have become more efficient and computers have become faster, the searches have grown larger and more sensitive. The SETI@home project, managed by a group of researchers at the Space Sciences Laboratory of the University of California, Berkeley, is the first attempt to use large-scale distributed computing to perform a sensitive search for radio signals from extraterrestrial civilizations.

A radio SETI tutorial

You might wonder why an enormous supercomputer would be necessary to detect radio signals from an alien civilization. It might seem to be a fairly simple signal-processing task. Such a supercomputer is needed primarily because

- the parameters of an alien signal are unknown, and
- the sensitivity of a search for extraterrestrial intelligence (SETI) depends fairly heavily on the available processing power.

Our search for extraterrestrial intelligence assumes that an alien civilization wishing to make contact with other races would broadcast a signal that is easily detectable and easily distinguishable from natural sources of radio emission. One way to achieve these goals is to send a narrowband signal. By concentrating the signal power in a very narrow frequency band, the signal will stand out among the natural broadband sources of noise.

Consequently, radio SETI efforts have concentrated on detecting narrowband signals. When searching for narrowband signals, it is best to use a narrow search window (or channel) around a given frequency. The wider the channel, the more broadband noise is included in addition to any signal, which limits the system's sensitivity. Early systems used analog technology to create narrow bandpass filters that could observe at a single frequency channel. More recent systems use massive banks of dedicated fast Fourier transform (FFT) processors to separate incoming signals into up to a billion channels, each 1 Hz wide.

Unfortunately, this technique is limited. For one thing, extraterrestrial signals are unlikely to be stable in frequency because of transmitter and receiver accelerations. For example, a receiver listening for signals at 1.4 GHz located on the earth's surface undergoes acceleration of up to 3.4 cm/s^2 because of the earth's rotation. That might not seem like much, but it corresponds to a Doppler drift rate of 0.16 Hz/s. If uncorrected, an alien transmission would drift out of a 1-Hz channel in about 6 seconds, effectively limiting the maximum integration time to 6 seconds. Because of the inverse relationship between maximum frequency resolution and integration time $\Delta \upsilon = (1/\Delta t)$, there is an effective limit to the frequency resolution that we can obtain without correcting the received signal for this effect ($\Delta \upsilon \sim 0.4$ Hz).

In principle, we could correct for most of the drift resulting from the earth's motions, but how do we correct for an unknown planet's motions? An alien civilization narrowly beaming signals at the earth could correct the outgoing signal for the transmitter's motions, but a civilization transmitting an omnidirectional beacon could not make such an adjustment. Therefore, to search for this type of signal at very narrow bandwidth (<< 1 Hz) and the highest possible sensitivity, we would need to correct for Doppler drift at the receiving end and search for signals at multiple Doppler drift rates. Repeating an analysis at multiple Doppler drift rates becomes compute intensive.

Other signal parameters are still unknown—for example, at what frequency will it be transmitted? What is its bandwidth? Will it be pulsed? If so,

Figure 1. SETI@home uses the National Astronomy and Ionospheric Center's 305 meter telescope at Arecibo, Puerto Rico. The inset at right shows the details of the "carriage house" structure. The SETI feed is the vertical structure extending downward from the carriage house. (Photos courtesy of NAIC—Arecibo Observatory, a facility of the NSF. Photos by David Parker and Tony Acevedo.)

at what period? Fully investigating a wide range of these parameters requires proportionally larger computing power.

In addition to detecting a signal, we must determine whether a signal is truly of celestial origin. The vast bulk of the narrowband signals received by a radio telescope consists of locally generated radio frequency interference (RFI). Fortunately, RFI has properties that let us distinguish it from extraterrestrial emission. But again, this RFI elimination requires computing resources.

Performing all of these calculations for even a small portion of the radio spectrum would require more computational power than is available in the largest existing supercomputer.

Distributing the load

Fortunately, searching for signals in a data stream from a radio telescope is an easily distributed task. We can break up data from an observation into frequency bands that are essentially independent of one another. In addition, an observation of one portion of the sky is essentially independent of an observation of another part. This lets us divide a large dataset into small chunks that a personal computer can analyze comparatively quickly. In this way, we can distribute the work to people willing to donate their spare CPU cycles.

SETI@home conducts its observations at the National Astronomy and Ionospheric Center's 305-meter radio telescope in Arecibo, Puerto Rico (see Figure 1). The project uses a dedicated feed (the nearly vertical structure suspended left of the center of the dish, shown in the expanded view) from the

receiver mounted opposite the primary observer's feed (the enclosed dome-like structure to the right) on the Arecibo telescope's carriage house. This unique arrangement lets us conduct SETI@home observations without interference with other uses of the telescope and results in three main modes of observation. If the primary feed is stationary, objects in the sky pass through the SETI@home instrument's field of view (0.1 degrees) at the rate of the earth's rotation (also known as the sidereal rate). An object would require about 24 seconds to transit the field. If the primary observer is tracking a source in the sky, the SETI@home receiver's beam slews across the sky at twice the sidereal rate. Occasionally, other observers could use the SETI@home feed to track objects on the sky.

During the project's course, SETI@home will view most portions of the sky visible with the Arecibo telescope three or more times. This includes stars with declinations (the celestial equivalent of latitude) between –2° and 38°, thoroughly covering about 25% of the sky.

The SETI@home system records a 2.5-MHz-wide band centered at the 1,420 MHz hydrogen line. Because this line would be of interest to astronomers of any species who were studying the galaxy, this frequency is one of the most likely locations for deliberate extraterrestrial transmissions. This 2.5-MHz band is recorded continuously onto 35-Gbyte DLT tapes using 2-bit complex samples. Each tape holds about 15.5 hours of data. The entire sky survey is

expected to require 1,100 tapes, for a total of 39 Tbytes of data.

The recorded tapes are shipped to Berkeley, where we subdivide them into small work units on four splitter workstations. We divide the 2.5-MHz bandwidth data into 256 subbands by means of a 2,048-point FFT followed by 256 eight-point inverse transforms. Because the 9,766-Hz-wide subbands are divided into lengths of 2^{20} samples, each work unit corresponds to about 10 kHz of bandwidth and 107 seconds of duration. Subsequent work units overlap by 20 to 30 seconds to allow full analysis of signals that might be within a beam transit time of the end of a work unit. We transfer each work unit to temporary storage (capable of holding about 750,000 work units) for distribution to users.

The main SETI@home server (Figure 2) consists of three Sun Enterprise 450 series computers. One holds the user database, containing information on each of the 2.4 million SETI@home volunteers (including the number of work units completed, time of last connection, and team membership). The user database also holds information on the amount of work done by each type of microprocessor architecture and by each operating system to which SETI@home has been ported.

The second server system holds the science database in an ever-expanding array of redundant disks (currently a 432-Gbyte RAID 0+1). The science database contains information on the time, sky coordinates, frequencies, and so forth for each work unit generated,
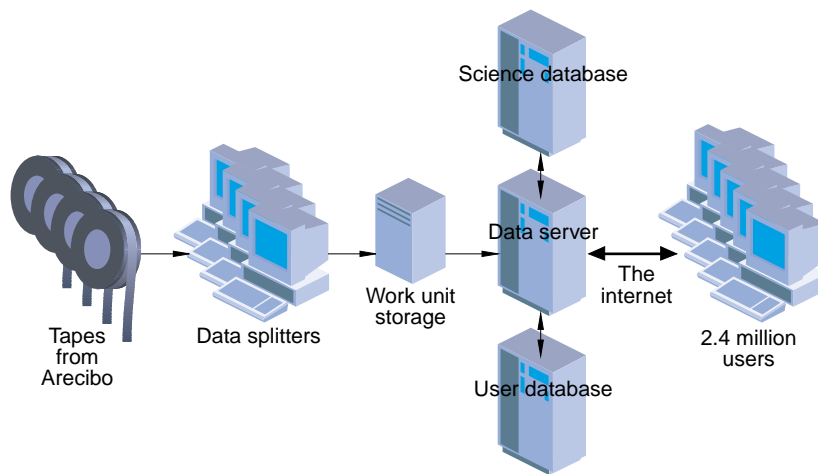
as well as information about how many times the work unit has gone to SETI@home users and how many results have been received.

The largest portion of the science database capacity stores the parameters of potential signals (such as signal power, frequency, and arrival time sky coordinates) detected by SETI@home volunteers. As of October 2000, the database contained 1.1 billion candidate signals (before RFI rejection).

The third server system contains the work unit storage, handling distribution of work units and storage of returned results. Communications between the server and clients use the hypertext transfer protocol (HTTP). We chose this protocol because many Internet volunteers might be behind firewalls that prohibit most traffic but permit access to the World Wide Web.

The server supports two types of request. The first requests a work unit. The response to this request includes a work unit chosen from temporary storage. Priority goes to those units that have not previously been sent or those that were sent but for which no results were received.

In the second type of request, the client program returns a result to the server. The server inserts the candidate signals contained in the result into the science database and updates the volunteer's statistics in the user database. The response to this request includes the volunteer's statistics so that the client program can display them.

The SETI@home client program

SETI@home currently distributes client software for 47 different combinations of CPU and operating system. Users can download the software from the SETI@home Web site (http://setiathome.ssl.berkeley.edu). For Microsoft Windows and Apple Macintosh, the software installs itself by default as a screen saver (Figure 3), only processing data when the screen saver is active. For other platforms, the basic client is text based. Users of these platforms generally run the client in the background. A graphical display program similar to the Mac and Windows versions is available for Unix systems that run the X Windows system. In addition, a wide variety of third-party applications have been developed for display of data, detected signals, sky maps, and volunteer statistics.

After receiving a work unit, the client performs a baseline smoothing on the data to remove any wideband ($\Delta \upsilon > 2$ kHz) features. This prevents the client from confusing fluctuations in broadband noise (due in part to variations in the hydrogen line emission as the field of view transits the sky) with intelligent signals. The client then begins the main data analysis loop, shown schematically in Figure 4.

At the start of each passage through the loop, the data is transformed into an accelerated frame of a given Doppler drift rate. The drift rates at which the client searches the data for signals vary from –10 Hz/sec to +10 Hz/sec (accel-

erations expected on a rapidly rotating planet) in steps of 0.0018 Hz/sec. The client also examines the data at Doppler drift rates out to ±50 Hz/sec (accelerations of the magnitude that would arise from a satellite in low orbit about an earth-like planet), but at a more coarse step of 0.029 Hz/sec. A signal from an alien world would most likely have a negative drift rate (as the accelerations involved would be away from the observer). Despite this, we examine both positive and negative drift rates for the purpose of statistical comparison and to leave open the possibility of detecting a deliberately chirped extraterrestrial signal.

At each drift rate, the client searches for signals at one or more bandwidths between 0.075 and 1,221 Hz. This is accomplished by using FFTs of length 2^n ($n = 3, 4, ..., 17$) to transform the data into a number of time-ordered power spectra. To avoid repeating work, not all bandwidths are examined at every Doppler drift rate. Only when the change in drift rate becomes significant compared to $(1/\Delta \upsilon^2)$ does the program compute another FFT of a given length. Therefore, 32K-point transforms are performed one quarter as often at 64K-point transforms.

The transformed data is examined for signals that exceed 22 times the mean noise power. This threshold corresponds to 7.2×10^{25} W/m^2 at our finest frequency resolutions, or the equivalent of detecting a cell phone on one of the moons of Saturn. The SETI@home client reports any such spike signals in the resulting transmission.

If there is sufficient time resolution in the transformed data ($n < 15$) and the SETI receiver is not tracking an object on the sky, the client examines it for signals that match the telescope beam's parameters. As a radio source drifts through the field of view, the measured

Figure 3. A screenshot of the SETI@home client program. The bottom half of the screen presents the power spectrum currently being analyzed. The upper left shows analysis state and the results of the current analysis. The upper right section shows information about the data being processed and user statistics.
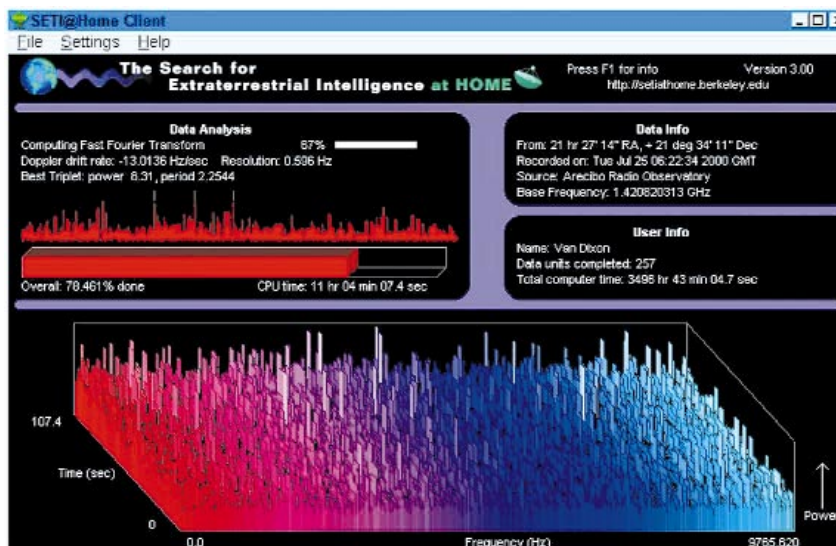
power will vary depending on the telescope's beam profile, which is approximately Gaussian. The SETI@home client performs a weighted χ^2 fit on any signals that exceed 3.2 times the mean noise power and reports those for which the goodness of fit exceeds a certain level. This power level typically corresponds to 8.4 x 10^{-25} W/m^2.

The client then divides transformed data at each frequency into chunks with duration equal to the time required for an object to transit the telescope field of view. Two algorithms serve to analyze these chunks for pulsed signals. The first algorithm, the triplet finder, searches each chunk for three evenly spaced signals that each exceed 7.75 times the mean noise power (as little as 5.3 x 10^{-25} W/m^2) and report any detected signals.

The second algorithm is a modified fast-folding algorithm. A folding algorithm divides the data into chunks of duration equal to the period being searched and co-adds them to improve signal-to-noise ratio. An FFA performs this function on a large number of periods without duplicating additions. The SETI@home folding algorithm searches roughly $N \log N$ pulse periods, where N is the length of the input array. This corresponds to periods between two samples and $N/3$ samples. During a typical run of the client, this typically means half a million periods between 2 ms and 10 s. The program computes the threshold for detecting a pulsed signal dynamically to match the number of co-added samples. This threshold can be as low as 0.04 times the mean noise power for pulses with periods less than 10 ms. This corresponds to pulse energies of about 1.8×10^{-26} J/m^2.

Depending on the individual work unit's parameters, this processing loop requires 2.4 to 3.8 trillion floating-point operations (Tflop). It takes a typical (500 MHz) home computer 10 to 12 hours to complete a work unit. For an average work unit, the SETI@home client would report eight signals—four spike signals, one Gaussian, one pulsed signal, and one triplet signal.

Postprocessing

When the client has done its work, the job isn't done. Typically, the SETI@home client program returns a few potential signals per work unit. Of course, not all of these signals are evidence of extraterrestrial intelligence.

Errors made in the processing computers cause some of the signals. Typical numeric processors, memory, and disk systems are fairly reliable. How-

ever, SETI@home uses thousands of years of CPU time per day, magnifying even low error rates. Even if undetected errors occur only on average every 10^{18} machine instructions, SETI@home would see several per day. Additional errors can be introduced in result transmission because of broken connections or malfunctioning HTTP proxies. To combat these effects, we examine each signal to see if the parameters match their permitted values. We also send each work unit to multiple volunteers and cross-check the returned values to verify accuracy.

The vast majority of the database's signals are evidence only of terrestrial intelligence. Sources of narrowband radio emission are ubiquitous where human technology is present. Even at the Arecibo observatory, where care is taken to minimize interference, this

```
for Doppler drift rates from -50 Hz/s to +50 Hz {
   for bandwidths from 0.075 to 1220 Hz in 2X steps {
      Generate time ordered power spectra.
      Search for short duration signals above a constant
         threshold (spikes)
      for each frequency {
         Search for faint signals matching beam parameters
            (Gaussians)
         Search for groups of three evenly spaced signals
            (triplets)
         Search for faint repeating pulses (pulses)
      }
   }
}
```

Figure 4. Pseudocode representation of the SETI@home algorithm.

Eric Korpela is a research astronomer at the Space Sciences Laboratory at the University of California, Berkeley. In addition to his SETI work, he specializes in the study of interstellar matter and far ultraviolet astronomical instrumentation. Contact him at korpela@albert.ssl.berkeley.edu.

Dan Werthimer is director of the Serendip SETI program and chief scientist of SETI@home. He has published numerous scientific papers in the fields of SETI, radio astronomy, instrumentation, and science education, and is editor of *Astronomical and Biochemical Origins* and the *Search for Life in the Universe.*

David Anderson is the SETI@home project director and has done extensive research in operating systems, distributed computing, and computer graphics. He recently joined the management team of the distributed computing provider United Devices as chief technology officer.

Jeff Cobb is a software engineer and systems manager for the UC SETI projects. He has been with the SETI group for seven years, and has developed many of the algorithms used in SETI@home.

Matt Lebofsky studied computer science and music composition at Binghamton University, and currently has two completely separate careers in both fields. (See also www.lebofsky.com.)

noise is present, due to local equipment, aircraft, satellites, and other transmitters. Fortunately, these terrestrial emissions are fairly easy to distinguish from an extraterrestrial signal.

A large fraction of RFI consists of continuous narrowband signals generated at or near the observatory. An extraterrestrial signal will only be detected when it is within the telescope's field of view, and, for our scanning mode of operation, will only have limited duration. Any signal that exceeds this duration must be terrestrial and may be rejected.

Other RFI sources are of short duration and repeat on time scales of hours to days. Therefore, any signal that repeats when the telescope is viewing a different portion of the sky might also be rejected.

After RFI is removed, the bulk of the remaining signals are due to random fluctuations in the noise background mimicking an extraterrestrial signal. To sort out the true extraterrestrials, we can look for persistent signals. We expect that an extraterrestrial signal will be present at a similar frequency the next time we examine the same celestial location.

A status report

As of 23 October 2000, 2,438,045 volunteers had run the SETI@home program. Of those, 519,725 were actively running the program and had returned a result in the previous two weeks. These volunteers had donated a total of 437,000 years of CPU time for a total 4.3×10^{20} flop. Currently, the average processing rate of computers running SETI@home is 15.7 Tflops—averaged since the start of the project, the processing rate is 9.5 Tflops. To our knowledge, SETI@home is the largest distributed computation project in existence. It could also be considered to be the largest supercomputer in existence and the largest computation ever performed. While we were writing this paragraph, 60 new volunteers joined the project.

The 1.1 billion signals in the SETI@home database are being examined with the techniques we've described. The rate at which we are currently examining signals is lower than the rate at which new signals are being added to the database, so we have looked in detail at only a fairly small fraction of the potential signals. We will soon add another computer system to our server setup to speed this processing along—we hope to examine signals in real time before too long. So far, none of the signals examined has shown evidence of extraterrestrial intelligence.

SETI@home was originally slated to process two years worth of data from the Arecibo telescope. The strong public response and new improvements to the client software have prompted us to extend the survey.

SETI@home currently samples only a small portion of the radio spectrum and a small portion of the sky. The two most obvious means of expanding its capabilities are to expand the sky coverage and widen the frequency bandwidth. SETI@home II, currently under study, hopes to do both.

The best means of expanding the sky coverage would be to add a SETI@home recorder system to a southern hemisphere radio telescope. This would let us increase our sky coverage from about 25% to 75%. We are currently discussing the possibility with a southern observatory.

The recording system currently limits our frequency bandwidth. By duplicating the recording system, we could double SETI@home's bandwidth coverage (and, of course, its data rate).

Café Dubois

Adventures in CGI

A friend who works for a small nonprofit organization asked me to help her learn to write a CGI program. A CGI program is one that the Web server starts up in response to someone clicking a "Submit" button. The Web server receives the information from the form (such as name, address, did you click the "Don't give my name to all your friends" button) and passes it on to the CGI program. You can write a CGI program any way you like, but the most common ways are to write it in C, Perl, or Python. Your program prints HTML to the standard output, and that is what the user's Web browser displays ("Thanks for your order. Of course, we are backordering your product. We have no products on hand, we just use your money to buy one and send it to you."). A CGI program lets you interact with things on your server such as databases, or send email, or calculate a page in response instead of having a static page displayed.

I knew how to do all the hard parts. Decoding the information the server passes you is easiest in Perl or Python, where there are library routines that do it for you.

I wrote a Python script that worked fine on my computer and sent it to my friend. She installed it into her IIS server running on Windows NT and, of course, it didn't work at all. There are a lot of things to check in such a situation, because when a server runs a CGI script it usually runs it as user "nobody" (a user with very limited permissions) and the environment in which the program is started is not the same as when a real user runs it. Everything checked; still no joy. Finally, she calls Microsoft support. I've had good luck calling Microsoft when I'm a paying customer. They seem to really stick with it to get you an answer. We get the answer. It is profound:

She had installed Python into C:\ProgramFiles\Python20. I had installed Python into C:\Python20. Mine worked. Hers didn't. That's right, the problem was the stupid space in the path. I hate computers.

Nevertheless, writing CGI scripts can be a lot of fun. If you want to try writing a CGI script for your Web site, two books I've used are Mark Lutz and David Ascher, *Learning Python*, (O'Reilly Press, 1999); and Dan Berlin, *CGI Programming Unleashed* (Sams.net, 1996). If you need a Web server to practice on, there is a little practice one that comes with Python, Microsoft Windows has a "personal Web server" you can install, or of course you can set up Apache on any Linux box you own. You usually are not allowed to put CGI scripts on Web sites you get for free or at low cost.

Python Programming on Win32

Python Programming on Win32, by Mark Hammond and Andy Robinson (O'Reilly Press, 2000), teaches you how to use Microsoft technologies such as COM from Python. For example, you can use a script to create a Word document or Excel database. That might be handy for converting code output into something usable, for example.

As in any voluntary organization, it's important that SETI@home be responsive to the desires of its volunteers, because the program's success depends entirely on the volunteers who provide the computing resources. We will continue to keep our volunteers informed of our progress and to share with them the science behind SETI. We will also work to show our volunteers how they have individually contributed to the project by providing information about potential signals they have detected and the areas of the sky they have scanned. ⬚

References

1. S. Bowyer et al., "Twenty Years of SERENDIP, the Berkeley SETI Effort: Past Results and Future Plans," *Astronomical and Biochemical Origins and the Search for Life in the Universe,* C.B. Cosmovici, S. Bowyer, and D. Werthimer, eds., IAU Colloquium No. 161 (Editrice Compositori: Bologna), p. 667, 1996.

2. D. Anderson et al., "Internet Computing for SETI," *Bioastronomy 99: A New Era in Bioastronomy*, G. Lemarchand and K. Meech, eds., ASP Conference Series No. 213 (Astronomical Society of the Pacific: San Francisco), p. 511, 2000.