

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

SETI@home

An Experiment in Public-Resource Computing

Millions of computer owners worldwide contribute computer time to the search for extraterrestrial intelligence, performing the largest computation ever.

SETI@home uses millions of computers in homes and offices around the world to analyze radio signals from space. This approach, while complicated, delivers unprecedented computing power and has led to a unique public involvement in science. Here, we describe SETI@home's design and implementation and discuss its relevance to future distributed systems.

SETI (Search for Extraterrestrial Intelligence) is a scientific area whose goal is to detect intelligent life outside Earth [8]. One approach, known as radio SETI, uses radio telescopes to listen for narrow-bandwidth radio signals from space. Such signals are not known to occur naturally, so a detection would provide evidence of extraterrestrial technology [1].

Radio telescope signals consist primarily of noise (from celestial sources, as well as the receiver's own electronics) and man-made signals, including TV stations,

radar, and satellites. Radio SETI projects digitally analyze the data, generally in three phases:

Computing its time-varying power spectrum;
Finding "candidate" signals through pattern recognition on the power spectra; and
Eliminating candidate signals that are probably natural or man-made.

More computing power enables searches to cover greater frequency ranges with more sensitivity. Thus, radio SETI has an insatiable appetite for computing power.

Before SETI@home, radio SETI projects used special-purpose supercomputers located at the telescope to do the bulk of its data analysis. In 1995, David Gedye, a project manager at Starwave Corp., proposed doing radio SETI using a virtual supercomputer consisting of large numbers of Internet-connected computers and organized the SETI@home project to explore this idea. SETI@home has not found signs of extraterrestrial life. But together with related distributed computing and storage projects, it has certainly established the viability of public-resource computing in which computing resources are

provided by the general public.

Public-resource computing is neither a panacea nor a free lunch. For many tasks, huge computing power implies huge network bandwidth, which is typically expensive or limited. This factor also limits the frequency range searched by SETI@home, as greater range implies more bits per second. Compared to other radio SETI projects, SETI@home covers a narrower frequency range but does a more thorough search within that range (see the table).

Design

The first challenge for SETI@home was to find a good radio telescope. The ideal choice was the one operated by Cornell University and the National Science Foundation in Arecibo, Puerto Rico, the world's largest and most sensitive radio telescope. Arecibo is used for various astronomical and atmospheric research, so we could not obtain its exclusive long-term use. However, in 1997 the SERENDIP (Search for Extraterrestrial Radio Emissions from Nearby Developed Intelligent Populations) project at the University of California, Berkeley, developed a technique for piggybacking a secondary antenna at Arecibo [10]. As the main antenna tracks a

fixed point in the sky (under the control of other researchers) the secondary antenna traverses an arc eventually covering the entire band of sky visible to the telescope. This data source can be used for a sky survey covering billions of stars.

We thus arranged for SETI@home to share SERENDIP's data source. However, unlike SERENDIP, we needed to distribute data via the Internet. At that time (1997) Arecibo's Internet connection was a 56Kbps modem, so we decided to record data on removable tapes (35GB digital linear tape

drive cartridges, the largest available at the time), have them mailed from Arecibo to our lab in Berkeley, and distribute data from servers there.

We recorded data at 5Mbps, a rate low enough that the recording time per tape was a manageable 16 hours, making it feasible to distribute the data through our lab's 100Mbps Internet connection. The rate was also high enough to allow us to do significant science. With one-bit complex sampling, this rate yields a frequency band of 2.5MHz, enough to handle Doppler shifts for relative velocities up to 260km/sec, or about the rate of the Milky Way's galactic rotation; radio signals are Doppler shifted in proportion to the sender's velocity relative to the receiver. Like many other radio SETI projects, we centered our band at the Hydrogen line (1.42GHz), within a frequency range where man-

made transmissions are prohibited by an international treaty.

SETI@home's computational model is simple. The signal data is divided into fixed-size work units distributed via the Internet to a client program running on numer-

The task of creating and distributing work units is done by a server complex located in our lab (see Figure 1). The reasons for centralizing the server functions are largely pragmatic; for example, it minimizes tape handling.

Work units are formed by dividing the 2.5MHz signal into 256 frequency bands, each about 10KHz wide. Each band is then divided into 107-second segments, overlapping in time by 20 seconds. This overlap ensures that signals we seek (lasting up to 20 seconds) are contained entirely in at least one work unit. The resulting work units are 350KB, or enough

data to keep a typical computer busy for about a day but small enough to download over even slow modems in a few minutes.

We use a relational database (Informix) to store information about tapes, work units, results, users, and other aspects of the project. We developed a multi-threaded data/result server to distribute work units to clients (see Figure 2). It uses a HTTP-based protocol so clients inside firewalls are able to contact it.

A "garbage collector" program removes work units from disk, clearing an on-disk flag in their database records. We have experimented with two policies:

Delete work units for which N results have been received, where N is the target redundancy level. If work-unit storage fills up, work-unit production is blocked and system throughput declines.

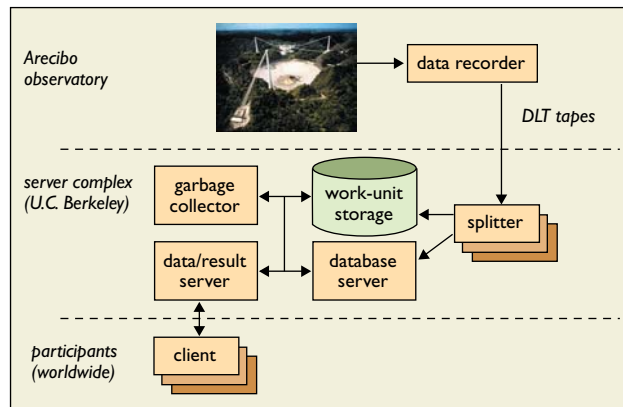


Figure 1. Distribution of radio data.

ous computers. The client program computes a result (a set of candidate signals), returns it to the server, then gets another work unit. There is no communication between clients.

SETI@home does redundant computation; each work unit is processed multiple times, letting us detect and discard results from faulty processors and from malicious users. A redundancy level of two to three is adequate for this purpose. We generate work units at a bounded rate and never turn away a client asking for work, so the redundancy level increases with the number of clients and their average speed. These quantities have increased greatly during the life of the project. We have kept the redundancy level within the desired range by revising the client to do more computation per work unit.

Delete work units that have been sent M times, where M is slightly more than N . This policy is the one we use today. It eliminates the bottleneck but causes some work units to never produce results. The fraction can be made arbitrarily small by increasing M .

Keeping the server system running has been the most difficult and expensive part of SETI@home. The sources of failure, both hardware and software, seemed limitless. We have converged on an architecture that minimizes dependencies between server subsystems; for example, the data/result server can be run in a mode in which, instead of using the database to enumerate work units to send, the server gets the information from a disk file, allowing us to distribute data even when the database is down.

The client program repeatedly gets a work unit from the data/result server, analyzes it, then returns the result (a list of candidate signals) to the server. It needs an Internet connection only while communicating with the server and can be configured to compute only when its host is idle or to compute constantly at a low priority. The program periodically writes its state to a disk file, reading the file on startup; hence it makes progress even if the host is frequently turned off.

Analyzing a work unit involves computing signal power as a function of frequency and time, then looking for several types of patterns in this power function: “spikes” (short bursts); “Gaussians” (narrow-bandwidth signals with a 20-second Gaussian envelope corresponding to the telescope’s beam movement across a point); “pulsed signals” (Gaussian signals pulsed with arbitrary period, phase, and duty cycle); and “triplets” (three equally spaced spikes at the same frequency, or a simple pulsed signal). Signals whose power and goodness of fit exceed thresholds are recorded in the output file.

Outer loops vary two parameters [3]:

Doppler drift rate. If the sender of a fixed-frequency

signal is accelerated relative to the receiver (such as by planetary motion), then the received signal drifts in frequency. Such signals are best detected by undoing the drift in the original data, then looking for constant-frequency signals. The drift rate is unknown; we check 31,555 different drift rates covering the range of physically likely accelerations.

Frequency resolution. We cover 15 frequency resolutions ranging from 0.075Hz to 1220.7Hz. This coverage increases sensitivity to modulated signals, whose frequency content is spread over a range.

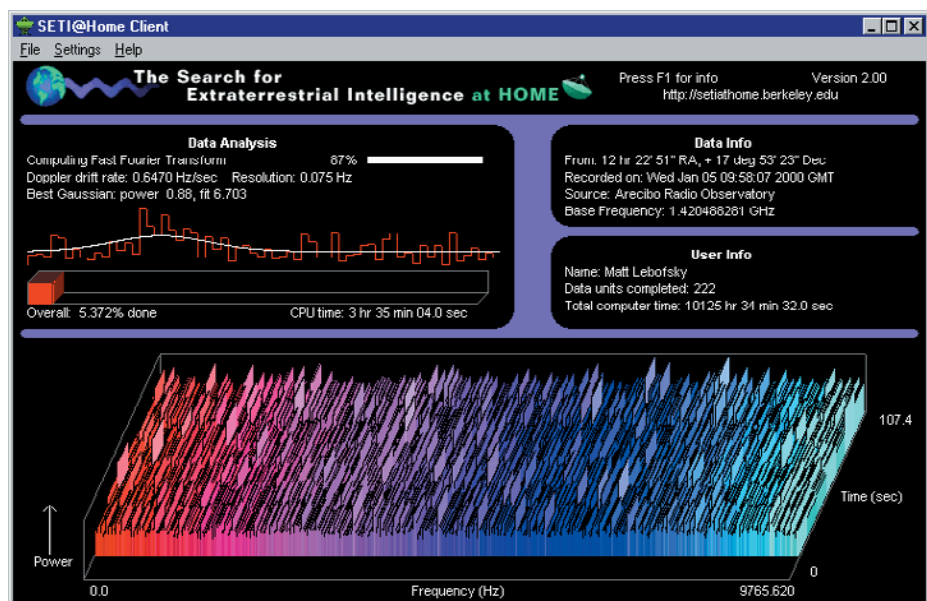


Figure 2. SETI@home display, showing the power spectrum being computed (bottom) and the best signal found so far (left).

The SETI@home client program, written in C++, consists of a platform-independent framework for distributed computing (6,423 lines of code), components with platform-specific implementations (such as the graphics library, with 2,058 lines in the Unix version), SETI-specific data analysis code (6,572 lines), and SETI-specific graphics code (2,247 lines).

The client has been ported to 175 different platforms. The GNU tools, including gcc and autoconf, greatly facilitate this task. The Macintosh, SPARC/Solaris, and Windows versions are all maintained directly by SETI researchers; all other porting is done by volunteers.

The client can run as a background process as either a GUI application or as a screensaver. To support these different modes on multiple platforms, the system employs an architecture in which one thread handles communication and data processing, a second thread handles GUI interactions, and a third thread (perhaps in a separate address space) renders graphics

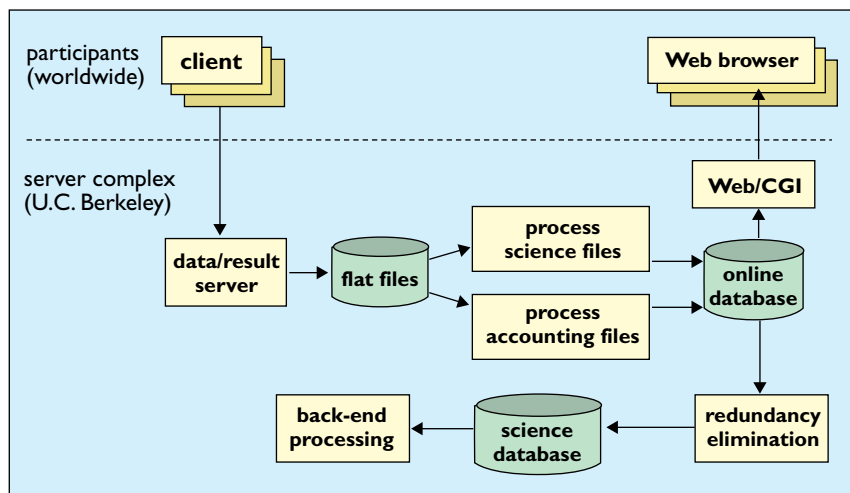


Figure 3. Collection and analysis of results.

inated. We look for signals with similar frequency and sky coordinates detected at different times. These “repeat signals,” along with one-time signals of sufficient merit, are examined manually and possibly reobserved, potentially leading to a final cross-check by other radio SETI projects, according to a protocol called the “Declaration of Principles Concerning Activities Following the Detection of Extraterrestrial Intelligence” (see www.seti.org/science/principles.html).

based on a shared-memory data structure.

Results are returned to the SETI@home server complex, where they are recorded and analyzed (see Figure 3). Handling a result consists of two tasks:

Scientific. The data server writes the result to a disk file. A program reads the files, creating result and signal records in the database. To optimize throughput, several copies of the program run concurrently.

Accounting. For each result, the server writes a log entry describing the result’s user, its CPU time, and more. A program reads these log files, accumulating in a memory cache the updates to all relevant database records (such as user, team, country, and CPU type). It flushes this cache to the database every few minutes. By buffering updates in disk files, the server system is able to handle periods of database outage and overload.

Eventually, each work unit produces a number of results in the database. A “redundancy elimination” program examines each group of redundant results—possibly differing in number of signals and signal parameters—and uses an approximate consensus policy to choose a representative result for that work unit. These results are copied to a separate database.

The final phase of the data analysis, back-end processing, consists of several steps. To verify the system, we check for the test signals injected at the telescope. Man-made signals are identified and elim-

Using large amounts
of data per unit
computation might
make some
applications
unsuitable for
public-resource
computation.

Public Response

We announced plans for SETI@home in 1998, prompting 400,000 people to preregister during the following year. The Macintosh and Windows versions of the client were released in May 1999. Within a week, about 200,000 people had downloaded and run the client. This number had grown to more than 3.91 million as of August 2002 in 226 countries, about 50% in the U.S.; 71%

describe themselves as home users.

In the 12 months beginning July 2001, SETI@home participants processed 221 million work units. The average throughput during that period was 27.36TFLOPS. Overall, the computation has performed 1.873×10^{21} floating point operations, the largest computation on record.

SETI@home relies primarily on mass-media news coverage and word-of-mouth to attract participants. The Web site (setiathome.berkeley.edu) explains the project, lets users download the client program, and provides scientific and technical news. It shows leader boards (based on work units processed) for individuals and for various groupings, including individual countries and email domains. Users can form teams that compete within categories; there were 98,600 such teams as of August 2002. Leader-board competition—among individuals, teams, owners of different computer types, and others—has further helped attract and retain participants. In addition, users are

recognized on the Web site and thanked by email when achieving work-unit milestones.

We have also sought to foster a SETI@home community in which users exchange information and opinions. The Web site even lets users submit profiles and pictures of themselves. An online poll includes questions concerning demographics, SETI, and distributed computing; for example, of the 116,000 users completing the poll as of August 2002, 93% were male. We assisted in the creation of a newsgroup at sci.astro.seti devoted largely to SETI@home. Meanwhile, individual users have created various ancillary software, including proxy data servers and systems for graphically displaying work progress; the Web site contains links to these contributions. Moreover, the Web site has been translated into 30 languages, including relatively obscure ones like Catalan, Estonian, and Farsi, along with

Project name	Sensitivity (Watts per square meter)	Sky coverage (% of sky)	Frequency coverage (MHz)	Maximum signal drift rate (Hz/sec)	Frequency resolution (Hz)	Computing power (GFLOPS)
Phoenix	1e-26	0.005 (1,000 nearby stars)	2000	1	1	200
SETI@home	3e-25	33	2.5	50	0.07 to 1200 (15 octaves)	27,000
SERENDIP	1e-24	33	100	0.4	0.6	150
Beta	2e-23	70	320	0.25	0.5	25

Radio SETI projects compared.

the more popular ones like French, German, and Japanese.

We aim to prevent the client program from acting as a vector for software viruses, successfully thus far; the code-download server has not been penetrated (as far as is known), and the client program does not download or install code. However, two noteworthy attacks have marred this record. The Web server was compromised but only as a prank in which the hackers did not install, for example, a Trojan-horse download page. Later, exploiting a design flaw in the client/server protocol, hackers obtained some user email addresses. Subsequently, the flaw was fixed but not before thousands of addresses were stolen. On another occasion, a user developed an email-propagated virus that downloads and installs SETI@home on the infected computer, configuring it to give credit to the user's SETI@home account. This might have been prevented by requiring a manual step in the install process.

We have also had to protect SETI@home from misbehaving and malicious participants. There have been many instances, though only a tiny fraction of

the overall participant population are involved; for example, a relatively benign instance involved users modifying the client executable to improve its performance on specific processors. We didn't trust the correctness of such modifications and didn't want SETI@home to be used in benchmark wars, prompting us to adopt of a policy banning modifications.

Other users have deliberately sent erroneous results. Preventing all these activities is difficult if users run the client program under a debugger, analyze its logic, and obtain embedded encryption keys [4]. The system's redundancy-checking, along with the error tolerance of SETI@home computing tasks, are sufficient for dealing with the problem; other mechanisms have also been proposed [6].

Extra Cycles

Public-resource computing relies on personal computers with excess capacity, including idle CPU time. The idea of using these cycles for distributed computing was proposed in 1978 by the Worm computation project at Xerox PARC involving about 100 machines to measure Ethernet performance [8] and later explored by academic projects,

including Condor, a toolkit developed at the University of Wisconsin for writing programs that run on unused workstations, typically within a single organization.

Large-scale public-resource computing became feasible with the growth of the Internet in the 1990s. Two major public-resource projects predate SETI@home. The Great Internet Mersenne Prime Search (GIMPS), which searches for prime numbers, began in 1996. Distributed.net, which demonstrates brute-force decryption, began in 1997. More recent applications include protein folding (folding@home at Stanford University) and drug discovery (the Intel-United Devices Cancer Research Project in Austin, TX).

Several efforts are under way to develop general-purpose frameworks for public-resource and other large-scale distributed computing. The Global Grid Forum, formed in 1999, is developing projects collectively called The Grid for resource-sharing among academic and research organizations [2]. Private companies, including Entropia, Platform Computing, and United Devices, are developing systems for distributed computation and storage in both public and organizational settings.

More generally, public-resource computing is an

aspect of the peer-to-peer paradigm, which involves shifting resource-intensive functions from central servers to workstations and home PCs [5].

Which tasks are amenable to public-resource computing? Several factors help predict. First, they should involve a high computing-to-data ratio; for example, each SETI@home data unit takes 3.9 trillion floating-point operations, or about 10 hours on a 500MHz Pentium II, yet involves only a 350KB download and 1KB upload. This ratio keeps server network traffic at a manageable level while imposing minimal load on client networks. Some applications, such as computer graphics rendering, use large amounts of data per unit computation, perhaps making them unsuitable for public-resource computation. However, reductions in bandwidth costs allay these problems, and multicast techniques reduce costs when a large part of the data is constant across work units.

Applications with independent parallelism are easier to handle than those with many data dependencies. SETI@home work-unit computations are independent, so participating computers never wait for or communicate with one another. If a computer fails while processing a work unit, the work unit is eventually sent to another computer. Applications requiring frequent synchronization and communication among nodes have been parallelized using such hardware-based approaches as shared-memory multiprocessors and more recently software-based cluster computing, such as Parallel Virtual Machine software [9]. Based on these considerations, public-resource computing, with its frequent computer outages and network disconnections, seems ill-suited to such applications. However, scheduling mechanisms that find and exploit groups of LAN-connected machines may eliminate these difficulties.

Tasks that tolerate errors are more amenable to public-resource computing; for example, if a SETI@home work unit is analyzed incorrectly or not at all, the overall goal is affected only slightly—an omission remedied when the telescope again scans the same point in the sky.

Conclusion

A public-resource computing project must attract participants. There are currently enough Internet-connected computers for about 100 projects the size of SETI@home; interesting and worthwhile ones have been proposed in global climate modeling and ecological simulation, as well as in non-science areas, such as computer graphics. To attract and keep users, a project must explain and justify its goals, providing compelling views of local and global progress. Screensaver graphics are an excellent

medium for displays that also provide a form of viral marketing. Moreover, the success of public-resource computing projects has the ancillary benefit of increasing public awareness of science and democratizing, to an extent, the allocation of research resources. **G**

REFERENCES

1. Cocconi, G. and Morrison, P. Searching for interstellar communications. *Nature* 184, 4690 (Sept. 1959), 844.
2. Foster, I. and Kesselman, C. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufman, San Francisco, 1999.
3. Korpela, E., Werthimer, D., Anderson, D., Cobb, J., and Lebofsky, M. SETI@home: Massively distributed computing for SETI. *Comput. Sci. Engin.* 3, 1 (Jan.–Feb. 2001), 79.
4. Molnar, D. The SETI@home problem. *ACM Crossroads* (Sept. 2000); see www.acm.org/crossroads/columns/onpatrol/september2000.html.
5. Oram, A., Ed. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly and Associates, Sebastopol, CA, 2001.
6. Sarmenta, L. Sabotage-tolerance mechanisms for volunteer computing systems. In *Proceedings of the ACM/IEEE International Symposium on Cluster Computing and the Grid* (Brisbane, Australia, May 15–18). IEEE Computer Society, Los Alamitos, CA, 2001.
7. Shoch, J. and Hupp, J. The Worm programs: Early experience with a distributed computation. *Commun. ACM* 25, 3 (Mar. 1982), 172–180.
8. Shostak, S. *Sharing the Universe: Perspectives on Extraterrestrial Life*. Berkeley Hills Books, Berkeley, CA, 1998.
9. Sunderam, V. PVM: A framework for parallel distributed programming. *Concurrency: Pract. Exper.* 2, 4 (Dec. 1990), 315–339.
10. Werthimer, D., Bowyer, S., Ng, D., Donnelly, C., Cobb, J., Lampton, M., and Airieau S. The Berkeley SETI Program: SERENDIP IV Instrumentation. In *Astronomical and Biochemical Origins and the Search for Life in the Universe*, C. Cosmovici, S. Bowyer, and D. Werthimer, Eds. *Proceedings of the 5th International Conference on Bioastronomy, IAU Colloquium No. 161* (Capri, July 1–5). Editrice Compositori, Bologna, Italy, 1997.

DAVID ANDERSON is a research scientist in the Space Sciences Laboratory at the University of California, Berkeley, and Chief Science Officer at United Devices, Austin, TX.

JEFF COBB is a programmer analyst in the Space Sciences Laboratory at the University of California, Berkeley.

ERIC KORPELA is an associate research physicist in the Space Sciences Laboratory at the University of California, Berkeley.

MATT LEBOSKY is a programmer analyst in the Space Sciences Laboratory at the University of California, Berkeley.

DAN WERTHIMER is a research physicist in the Space Sciences Laboratory at the University of California, Berkeley.

The Planetary Society, Sun Microsystems, the University of California Digital Media Innovations (DiMI) program, Fujifilm, Quantum, Informix, Network Appliances, and other organizations and individuals have supported SETI@home.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.