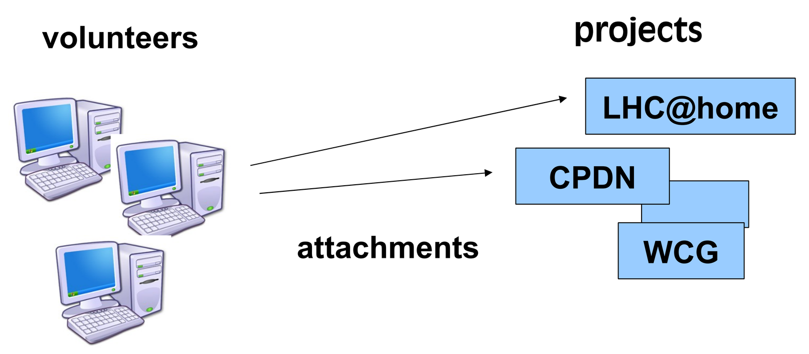
BOINC: Middleware for Volunteer Computing

David P. Anderson
Space Sciences Laboratory
University of California, Berkeley

Volunteer computing

Volunteer computing is a mechanism by which consumer resources can be used for scientific computing. It allows computer owners to volunteer for particular computational research projects or groups of projects. Compute jobs are then executed in the background on the volunteer's computers, and data files are stored on their disks.

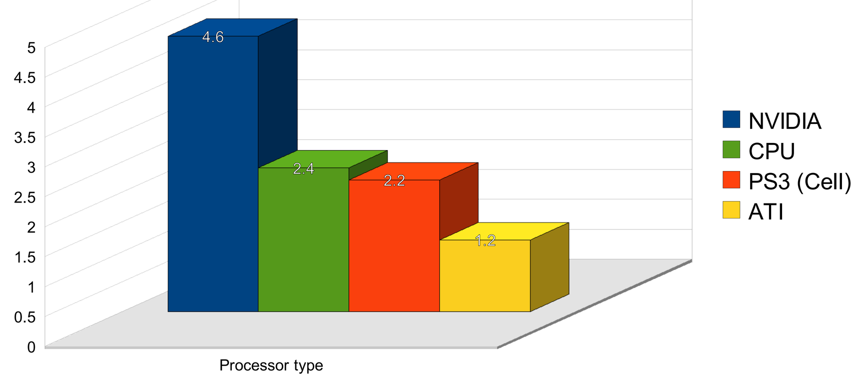
BOINC is a middleware platform for volunteer computing. Using BOINC, scientists can create **projects**. Projects are autonomous; BOINC has no central control. Computer owners can attach their computers to any set of projects, and can control the allocation of the resources among these projects.



The design of BOINC is intended to enable a **computational ecosystem** in which a dynamic population of projects compete for volunteers via education and promotion.

Current and potential performance

Currently 580,000 computers, owned by 330,000 volunteers, participate in 50 BOINC-based projects. They supply an average throughput of 4 PetaFLOPS. Including non-BOINC projects such as Folding@Home, the total throughput of volunteer computing is 12 PetaFLOPS, the majority of which is supplied by GPUs.



The potential performance of volunteer computing is much greater. Current high-end GPUs can do 1 TeraFLOPS. If 4 million such devices are volunteered, and are available 25% of the time, the result will be 1 ExaFLOPS.l

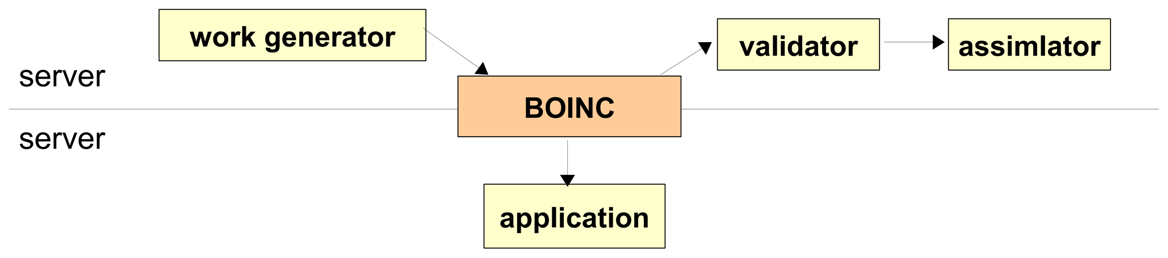
Projects using BOINC

Some of the projects using BOINC:

- Rosetta@home (Univ. of Washington): biomedical research
- ClimatePrediction.net (Oxford): study of long-term climate change
- IBM World Community Grid (hosted by IBM): various humanitarian-oriented research projects from different universities
- Einstein@home (LIGO consortium and Max Planck Institute): search for gravitational pulsars and binary pulsar systems using LIGO and Arecibo data
- LHC@home (CERN): simulation-based optimization of LHC and detectors
- Milkyway@home (RPI): study of the structure of the Milky Way galaxy using genetic optimization
- SETI@home (UC Berkeley): search for synthetic signals in Arecibo data
- Quake Catcher Network (Stanford): distributed seismographic sensing
- EDGeS@home (SZTAKI Institute): volunteer-based back end for EGEE
- Superlink (Technion); genetic linkage analysis
- MalariaControl.net (Swiss tropical institute): epidemiological modeling of malaria
- Virtual Prairie (U. of Houston): modeling of prairie ecosystems
- GPUGRID.net (Barcelona Biomedical Research Park): biomedical research
- Lattice (U. of Maryland): life science research
- AQUA@home (D-Wave systems): simulation of quantum computers
- Cosmology@Home (U. of Illinois): cosmological modeling
- Spinhenge@home (U. of Bielefeld): nanotechnoogy
- Quantum Monte Carlo at Home (U. of Muenster): quantum chemistry

Application development

Although it can handle single jobs, BOINC is designed for applications that involve huge numbers of similar jobs. The overall job flow is:

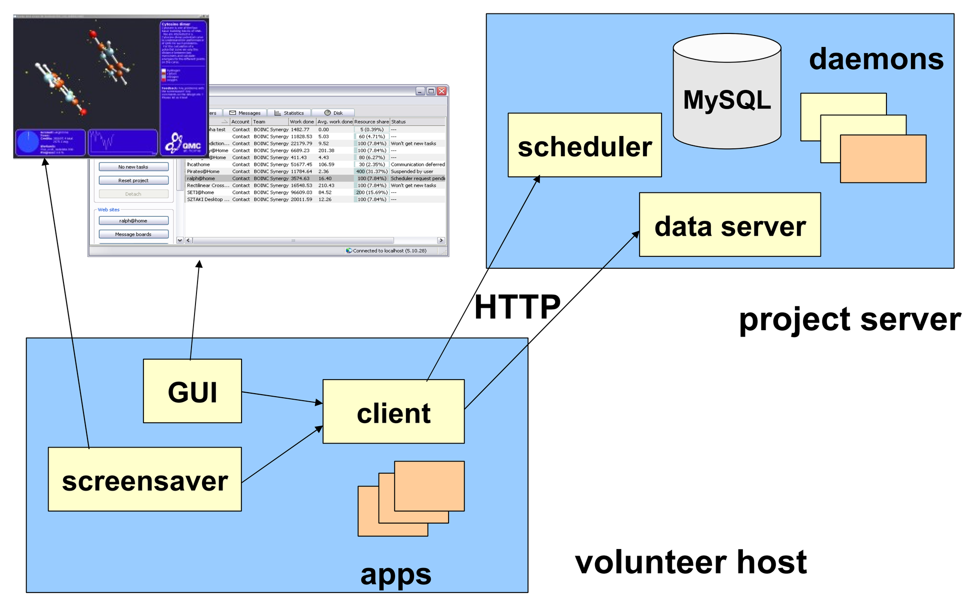


Typically versions of the application has catalyzare supplied for major platforms (Windows, Linux, Mac OS X). Jobs are associated with applications, not versions.

Several options are available for application development:
- **Native** applications linked with the BOINC API. Bindings are available for C/C++, FORTRAN, Java, and Python.
- Legacy applications, handled with a BOINC **wrapper** program
- GPU-based and multi-thread applications
- Applications that run in a virtual machine such as VirtualBox or VMWare, thus avoiding the need to compile for different platforms.

BOINC software

The BOINC software consists of several components:



The client software is available for all major platforms. It consists of
- The **client** manages program execution, file transfers, and scheduler interaction. All communication is done by client-initiated HTTP, so that the client can run behind firewalls.
- The **manager** provides a graphical interface, letting volunteers see the status of projects and jobs.
- The **screensaver** is a standard Windows or Mac screensaver. Rather than generating graphics itself, it typically launches a project-supplied graphics program associated with a currently executing job.

Volunteers can specify **preferences** that limit resource usage: for example, limits on the times of day when computation or transfers are done, limits on disk space and memory usage, and so on.

The server software runs on Linux and other Unix-compatible systems. It is based on a MySQL database that stores job, user, host, and application information. It consists of a **scheduler** that dispatches jobs to clients, and a number of daemon processes.

The server software also includes PHP-based web features that can be used to implement a project web site. These include a variety of community and social-network features.

Result validation

Volunteer hosts may return erroneous results – sometimes intentionally. BOINC projects several mechanisms for ensuring the correctness of results:
- **Replication**: each job is executed on 2 or more hosts. A result is accepted only if it returned by a strong majority.
- **Homogeneous replication**: a variant in which instances of a given job are sent only to numerically equivalent hosts (important for unstable floating-point applications).
- **Adaptive replication**: the error rate of each host is dynamically estimated. Hosts with low error rates don't use replication, but are randomly spot-checked.

Volunteer activities

BOINC offers the public many ways to volunteer other than donating the use of their computers:
- Software testing: the BOINC client software is alpha-tested by a group of 250 volunteers with diverse hardware and software
- Translation: volunteers maintain translations of the BOINC client and web software in about 15 languages.
- Technical support: volunteers provide technical support for new volunteers via message boards, and via a Skype-based telephone system.
- Porting and optimization: several BOINC projects have open-source applications. Volunteers have optimized these applications for particular processors, and in some cases have developed GPU versions of them using CUDA, Brook+, and OpenCL.

Research involving BOINC

BOINC has catalyzed a number of research projects related to volunteer computing:

- Host characterization: collecting trace data of host availability, fitting it with statistical models
- Grid integration: using BOINC in combination with Grids
- Cloud integration: the economics of using computing clouds to run BOINC clients or servers.
- Distributed applications: mechanisms for running MPI-type applications under volunteer computing.
- Scheduling: optimization of BOINC's client and server scheduling policies
- Volunteer motivation: studies of the motivational factors in volunteering.

Selected papers:

- Oded Nov, Ofer Arazy, and David P. Anderson. Crowdsourcing for science: understanding and enhancing SciSourcing contribution. The Changing Dynamics of Scientific Collaborations, workshop at CSCW 2010, Savannah GA, Feb. 7 2010.
- Exploiting Non-Dedicated Resources for Cloud Computing. Artur Andrzejak, Derrick Kondo and David P. Anderson. To appear in the 12th IEEE/IFIP Network Operations & Management Symposium (NOMS 2010), Osaka, Japan April 19-23, 2010.
- A Communication Framework for Fault-tolerant Parallel Execution. Nagarajan Kanna, Jaspal Subhlok, Edgar Gabriel, Eshwar Rohit and David Anderson. The 22nd International Workshop on Languages and Compilers for Parallel Computing, Newark, Delaware, Oct 8-10 2009.
- Computing Low Latency Batches with Unreliable Workers in Volunteer Computing Environments. Eric M. Heien, David P. Anderson, and Kenichi Hagihara. To appear, Journal of Grid Computing.
- Performance Prediction and Analysis of BOINC Projects: An Empirical Study with EmBOINC. Trilce Estrada, Michela Taufer, David Anderson. To appear, Journal of Grid Computing.
- Cost-Benefit Analysis of Cloud Computing versus Desktop Grids. Derrick Kondo, Bahman Javadi, Paul Malecot, Franck Cappello and David Anderson. 18th International Heterogeneity in Computing Workshop, May 25 2009, Rome.
- EmBOINC: An Emulator for Performace Analysis of BOINC Projects. Trilce Estrada, Michela Taufer, Kevin Reed, David Anderson. 3rd Workshop on Desktop Grids and Volunteer Computing Systems (PCGrid 2009), May 29, 2009, Rome.
- Local Scheduling for Volunteer Computing. David P. Anderson and John McLeod VII. Workshop on Large-Scale, Volatile Desktop Grids (PCGrid 2007) held in conjunction with the IEEE International Parallel & Distributed Processing Symposium (IPDPS), March 30, 2007, Long Beach.
- The Computational and Storage Potential of Volunteer Computing. David P. Anderson and Gilles Fedak IEEE/ACM International Symposium on Cluster Computing and the Grid, Singapore, May 16-19, 2006.
- High-Performance Task Distribution for Volunteer Computing. David P. Anderson, Eric Korpela, Rom Walton First IEEE International Conference on e-Science and Grid Technologies. 5-8 December 2005, Melbourne
- Homogeneous Redundancy: a Technique to Ensure Integrity of Molecular Simulation Results Using Public Computing. M. Taufer, D. Anderson, P. Cicotti, C.L. Brooks III. From 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05) Heterogeneous Computing Workshop. April 4 2005, Denver CO.

Contact information

For information about BOINC, contact

David P. Anderson
UC Berkeley Space Sciences Lab
Berkeley CA 94703

510 642-4921
davea@ssl.berkeley

The BOINC web site is: http://boinc.berkeley.edu