# “Mainstreaming Volunteer Computing” Status Report

# Science United

David P. Anderson

UC Berkeley

March 1, 2017

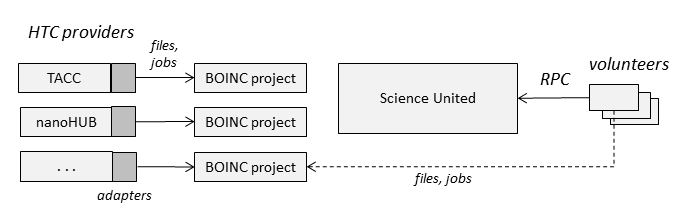
# Introduction

The third component of this project is **Science United** (SU), a new web-based framework for volunteer computing. SU is intended to achieve three goals. First, it provides an improved volunteer interface. Rather than seeing a bewildering array of projects, volunteers register at a single web site, and express their preferences in terms of scientific goals (physics, biomedicine, environment, etc.) and/or locations (countries and institutions).

Second, it provides a unified brand for volunteer computing, so that it can be promoted effectively.

Third, SU provides a centralized mechanism for allocating volunteer computing power. The volunteers participating in SU provide a pool of computing power that is divided administratively (perhaps as XSEDE allocations) among BOINC projects, including the projects at TACC and nanoHUB. Thus, new projects can be guaranteed computing throughput, and projects no longer need to promote themselves or create web sites in order to get resources.

The allocation mechanism works as shown below. Instead of attaching their BOINC clients directly to projects, volunteers attach to SU. The client then periodically makes RPCs to SU, asking it which projects to do work for. It then communicates with those projects’ servers to get and process jobs.



In this way, SU can dynamically control how much computing resources each project gets. If the volunteer computing power is fairly constant (as is the case with the current volunteer pool) it is even possible to make fairly “hard” QoS guarantees.

# Prototype

I’ve prototyped some of the basic parts of SU. The volunteer interface is here:

http://isaac.ssl.berkeley.edu/test2/

This shows the experience for new volunteers:

* Read the “call to action”.
* Fill out a registration form.
* Download and install BOINC.
* In BOINC client, log in to SU account.

The final version will be more streamlined: the download will start automatically, and the BOINC client will be pre-logged-in to the account.

Note: the graphics on this site – indeed, the name “Science United” itself – are place-holders. Before SU becomes public I hope to convene NSF PR people, marketing experts, and graphic designers to design a brand and graphical identity.

The SU administrative interface is here:

http://isaac.ssl.berkeley.edu/test2/su\_manage.php

Log in with email [admin@su.org](mailto:admin@su.org) and password “Admin”. The various parts are:

**Projects**: view the list of projects, with their keywords and allocations. Add new projects. Edit existing projects.

**Keywords**: view and edit the set of keywords.

**Accounting**: view tables and graphs of the history of resource usage, in total and broken down by project.

# Keywords and preferences

User preferences are expressed in terms of “keywords”. There are two classes of keyword: science area and location. Science areas might include “physics”, “biomedicine”, and “environment”, and location keywords would include “United Status” and “Europe”. These are 1st-level keywords. There are also more specific 2nd-level keywords such as “cancer research”, “particle physics”, “University of Texas”, and so on.

Each BOINC project is associated with a set of keywords, describing its science area and location. For each keyword, a volunteer can express a “yes”, “no”, or “maybe” preference. The default is “maybe”. The signup process includes a form for the 1st-level keywords, with an option for the 2nd level as well.

A volunteer is guaranteed to not be sent jobs from projects with a “no” keyword. They will preferentially (but not exclusively) be send jobs from projects with a “yes” keyword. This policy gives volunteers control, while still (assuming that volunteers use “no” sparingly) providing considerable flexibility in resource allocation.

# Allocation

SU uses a resource allocation model used in BOINC and some other job queueing systems:

* Each project has a numeric “resource share” and “balance”.
* The balance continually increases at a rate proportional to the resource share; it is capped at a particular value.
* When resources are assigned to a project (e.g. a job is dispatched) its balance is decremented by a proportionate amount.
* Resources are always assigned to the project whose balance is greatest at that point.

This scheme handles both bursty and continuous workloads. It prioritizes bursty workloads up to a point. It ensures that, averaged over long periods, each project gets resources proportional to its resource share.

An XSEDE allocation on SU would consist of a particular resource share for a particular period of time. This would be fair to projects with both bursty and continuous workloads.

For now, only one resource is accounted: floating-point operations (or more accurately, runtime times peak FLOPS). The BOINC client measures and reports this.

# Accounting

SU tracks resource usage in considerable detail, making it possible to view histories of resource usage broken down in various ways. It maintains historical records (with e.g. daily resolution of):

* The FLOPs, CPU time, GPU time, #active devices, #active GPUs, job counts (success and failure) of SU as a whole.
* The FLOPs, CPU time, GPU time, and job counts for each project.
* The FLOPs, CPU time, GPU time, and job counts for each volunteer.

In addition, it maintains current totals of:

* The FLOPs, CPU time, GPU time, and job counts for each (volunteer, project) combination.
* The FLOPs, CPU time, GPU time, and job counts for each (device, project) combination.

# Implementation

The implementation of SU is written in PHP. It leverages parts of the BOINC code, for things like login, message boards and other community functions. It uses the Bootstrap CSS system for web interfaces.

SU is based on a MySQL relational database. Projects, keywords, preferences, and accounting information are all stored in this database.

The SU source code is distributed under the LGPL license and is maintained on Github:

https://github.com/davidpanderson/science\_united