A new model for volunteer computing

David P. Anderson

Space Sciences Lab
University of California, Berkeley

Sept. 2017

The original BOINC model

- Scientists
  - lots of them (100s or 1000s) will create BOINC projects
  - They'll compete for volunteers by doing PR and making great web sites describing their research

The original model

- Volunteers
  - periodically survey the project web sites
  - rank projects based on
    - the importance of the science area to them
    - the credentials of the scientists
  - dynamically choose projects based on rankings

The original model

- Dynamic ecosystem of competing projects
- Project get computing power in proportion to their (apparent) merit
  - otherwise infeasible research gets done
  - high-risk research gets done
- The public learns about science
- A big fraction of global computing power does science

The original model

We've had considerable success, but:

- The set of projects has been small and static, and includes little mainstream U.S. science
- The set of volunteers
  - small, narrow, declining
  - project lock-in
  - motivated largely by credit

Why so few projects?

- High risk/reward
- Creating a BOINC project requires resources that few research groups have
- Few scientists know about volunteer computing

Why few/static volunteers?

- Evaluating projects is hard
- BOINC looks too complex in general
- Marketing volunteer computing is hard
  - too many brands

Other issues

- The HPC world views VC as a gimmick, and ignores it
- The computer science world ignores VC
- It’s hard (for me at least) to get funding

A new model

Goals

- Serve more scientists
- Get more volunteers
- Move VC toward the mainstream of computational science, HPC, computer science
  - get more smart people studying VC, developing BOINC, integrating BOINC

A new model

- Scientists
  - add VC to existing HPC facilities like supercomputer centers and portals
- Volunteers
  - create an interface to VC based on science goals rather than projects
  - create a unified brand (not ‘BOINC’)

Adding VC to HPC

- Texas Advanced Computing Center (TACC)
  - ~20% of jobs can run on VC
  - “launcher”: cmdline tool for running job batches
- nanoHUB
  - portal for nanoscience
  - web interface to ~30 standard apps
  - uncertainty quantification: lots of jobs
  - currently use small cluster and AWS

Adding VC to HPC

- BOINC “universal app”: vboxwrapper + Docker-player VM image
  - Docker image and input files are in workunit
- TACC and nanoHUB already package apps as Docker images
- We can support 1000s of apps and scientists with no incremental work!

Adding VC to HPC

- Remote job submission
  - each job in a batch can have its own templates
- Remote file management
- Identity mapping

Keywords

- Attributes of a job
  - What area of research does it contribute to?
  - Where (geographically and institutionally) are the researchers?
- For projects like TACC, these attributes are per-job, not per app or project

Keywords

- What can we use these attributes for
  - show project attributes in list
  - show user info about jobs they're running
  - let users filter work from a given project
  - do accounting of computing power per science area
  - let users express preferences at the top level

Keyword architecture

- Keywords have:
  - short and long names (dynamic)
  - integer ID and symbol (static)
  - hierarchy level and parent ID (dynamic)
  - category
    - science area
    - location

Keywords

- There must be a single, authoritative keyword list
- boinc/doc/keywords.inc
- community-based selection of keywords and hierarchy

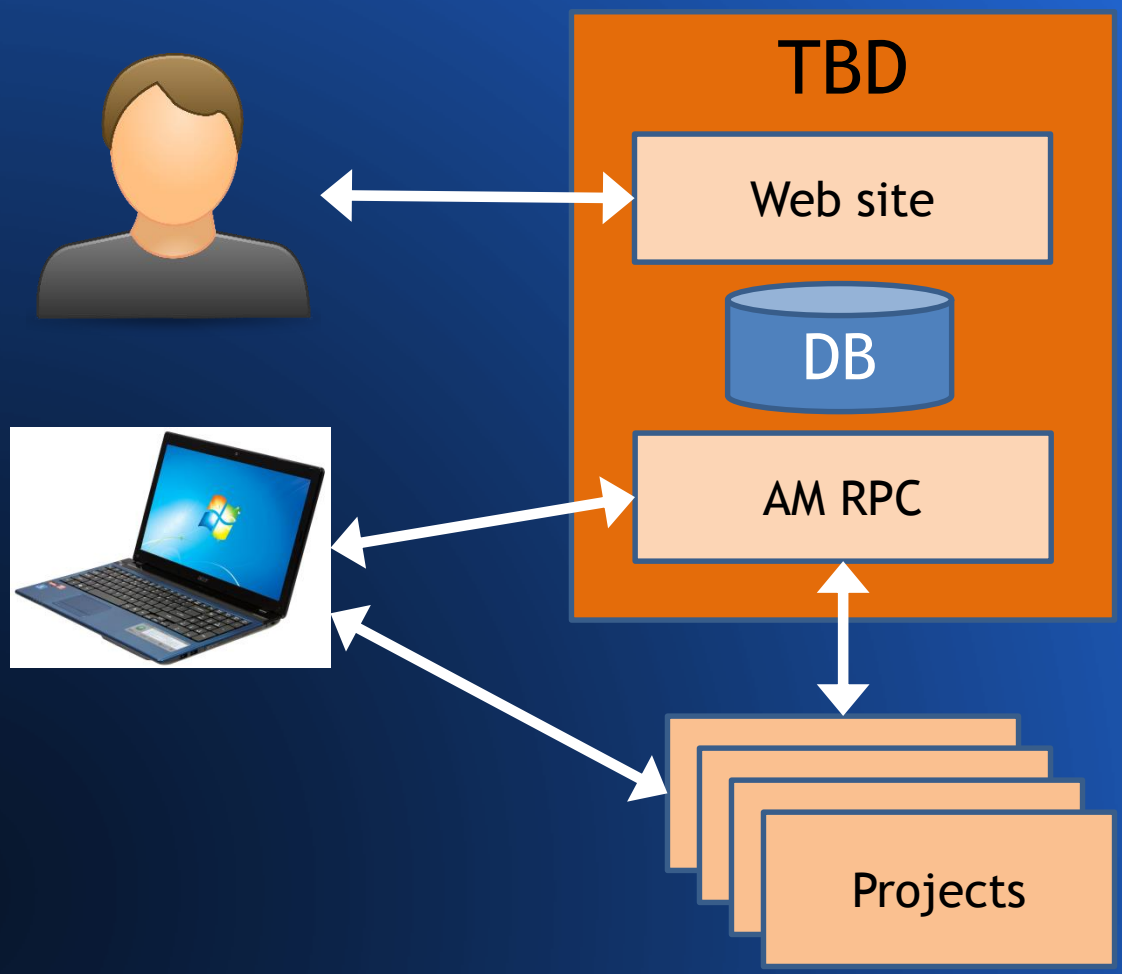
Keywords for job/project selection

- Each user can express a yes/maybe/no preference per keyword (e.g. in TBD or elsewhere)
- Each project has a list of (keyword, work fraction) and an optional list of keywords per job
- Don’t attach user to project with a “no” keyword with work fraction 1.
- Don’t send jobs with “no”keywords

TBD

- A volunteer interface based on keywords rather than projects.
- A resource allocation mechanism
- A brand for marketing volunteer computing

TBD Architecture



TBD: accounting

- TBD maintains daily history for
  - total
  - project
  - user
- Tracked quantities
  - REC (CPU, GPU)
  - runtime (CPU, GPU)
  - jobs (success, fail)

TBD: Allocation

- Linear allocation model
  - supports both continuous and bursty demands
  - possibly supports QoS guarantees

TBD: Allocation

Goals

- Respect user preferences
- Respect project allocations
- Maximize total throughput
- Minimize projects per host

TBD: Project allocation

- Might involve
  - National Science Foundation (XSEDE)
  - EU commission
  - others as appropriate
- Possible criteria
  - # scientists served
  - merit of research

TBD…

- is a unified brand
  - for marketing
  - for co-promotion
  - endorsed by trusted government agencies
- lets grant proposals include no-risk VC component
- supports sporadic or 1-time computing needs

Names

- Not ‘BOINC’
- Sigma / Scigma Σ
- Sciphon
- Sciborg
- SCI⏻N
- ⏻nboard

One-click install

New user scenario:

- Fill out account form (email, science prefs)
- Click “Join”
  - redirects to boinc.berkeley.edu/concierge.php
  - downloads appropriate client installer
  - encodes AM/account info in installer filename
  - BOINC client finds this
  - welcome dialog, no login dialog
- This mechanism can be used by projects too!

TBD Status

- Funded by NSF (UCB/Purdue/TACC) through 6/20
- Most of the technology has been developed
  - Github: davidpanderson/scienceunited
- UCB server/URL (scion.berkeley.edu)