BOINC: A System for Public-Resource Computing and Storage

David P. Anderson

Fifth IEEE/ACM International Workshop on Grid Computing

2004
Projects Using BOINC

Public-Resource Computing Projects

- SETI@home: 593.650 Tflops/s
- Predictor@home: -
- Folding@home: 8.588
- Climate@home: -
- Einstein@home: 288.809
- Climateprediction.net: 34.925
- CERN Projects: -
Overview

- What is BOINC?
- The Philosophy of BOINC
- The Need for BOINC
- The Goals of BOINC
- The BOINC Architecture
- Design Issues and Solutions
- Future Development
What is BOINC?

Berkeley Open Infrastructure for Network Computing

“...a software system that makes it easy for scientists to create and operate public-resource computing projects.”
The Philosophy of BOINC

BOINC is build upon the Public-resource paradigm

Definition

**Public-resource paradigm**: utilize computational resources of personal computers and game consoles.

The past decade we have seen the following developments:

- Migration of computational resources towards home users.
  - Computational Power
  - Storage
- Rapid growth of internet penetration
The Need for BOINC

However,

Relatively few large-scale public-resources projects have emerged

This is mainly due to the lack of appropriate middleware:

- Client and Server Software
- Management Tools
- User-centered Web Features
- ...
The Goals of BOINC

“BOINC's general goal is to advance the public-resource computing paradigm...”

Specific goals include:

- Reducing the barriers of entry to public-resource computing
- Sharing resources among autonomous projects
- Supporting diverse applications
- Rewarding participants
The BOINC Architecture
Design Issues

- Redundant Computing
- Handle Heterogeneous Environments
- Local Scheduling
- Participant Preferences
- Credit and Accounting
- User Community Features

We will look at these in more detail.
Design: Redundant Computing

A project can specify that $N$ results should be created for each task. Once $M \leq N$ of these have been distributed and completed, a comparison is made and a **canonical result** is formed.

The following server daemon processes take care of this:

- **the transitioner**
  
  *Generates new results and identifies errors*

- **the validator**
  
  *Compares results and creates canonical result*

- **the assimilator**
  
  *Insert canonical results into scientific database*

- **the deleter**
  
  *Delete obsolete input and output files*
BOINC provides a flexible framework for distributing application executables. Linux, Mac and Windows are supported by default.

However, participants are also able to compile the application themselves. These applications will be seen as running on an ‘anonymous’ platform.
BOINC implements a local scheduling policy. The goals of this policy are . . .

- to maximize resource usage,
- to satisfy result deadlines,
- to respect resource sharing allocation among projects, and
- to maintain variety among projects.
Design: Participant Preferences

Participants want no significant inconvenience, costs or risk. Therefore, BOINC lets participants control how and when their resources are used:

Participants can . . .

▶ specify the limits of work buffering,
▶ whether BOINC may work when the machine is not idle,
▶ during what hours BOINC may work,
▶ how many disk space BOINC may use,
▶ how much network bandwidth BOINC may use,
and more.
Participants need incentives in order to stay interested. Therefore, credits are attributed to an account. The amount depends on elapsed CPU/GPU time.

BOINC can provide user statistics as XML files which can then be posted by third parties.

BOINC allows for cross-project identification, so that one account can participate in multiple projects. The gained credit is then summed.
Design: User Community Features

BOINC provides participant-oriented web features, such as:

- The ability to form teams.
- The ability to create and browse user profiles, including text and images.
- A message board.
- A credit reputation system.
Future Development

Many areas of the BOINC design are incomplete. For example:

- No broadcast method is available yet.
- Scarce-resource situations are yet to be resolved.
- Multiple storage disks are not yet supported.
Conclusion

To conclude, BOINC is a promising approach towards distributed computing by offering easy public-resource computing.

It provides many novel and useful features, both for participants and scientists.

Join a project, contribute to science:
“boinc-wiki.info"

'http://boinc.berkeley.edu/’