Astropulse: A Search for Microsecond Transient Radio Signals Using Distributed Computing

Chapter 4: Distributed Computing: the BOINC Platform
1. Introduction

2. Telescope and Instrumentation

3. Pulse detection: detection thresholds and dedispersion

4. Distributed computing: the BOINC platform

5. RFI mitigation

6. Testing and Verification

7. Results and interpretation

8. Stardust@home

9. Suggestions for further research

1. Introduction

Astropulse runs on the BOINC platform, an acronym for “Berkeley Open Infrastructure for Network Computing.” BOINC is a set of programs that organizes volunteers’ home computers to perform scientific calculations. In a typical BOINC project, a researcher has a computing problem that can run in parallel, that is, on several machines at once. Perhaps the problem involves searching a physical space (for Astropulse, this space is the sky), and performing the same computation on each point in that space (for Astropulse, this computation is dedispersion.) The first BOINC project, SETI@home, searched the sky for narrowband transmissions. The space could also be a parameter space, for instance a space of potential climate models (climateprediction.net) or protein configurations (Rosetta@home). The visible manifestation of a BOINC project is an informative screen saver. Figure 1 shows the Astropulse screen saver, and Figure 2 depicts the climateprediction.net screen saver.

The researcher for a BOINC project need not be affiliated with UC Berkeley, or with the BOINC development team at Berkeley (although we happen to be so affiliated.) BOINC
is open source, and can be downloaded, compiled, and operated by anyone with sufficient technical skills; about 50 projects currently exist outside Berkeley.

Likewise, volunteers need not have any particular technical knowledge. They just have to navigate to the BOINC web page with their web browser, and follow the instructions to download the client programs described below. Astropulse has access to around 500,000 volunteers, each of whose machines might have 2 GFLOPs of processing power, and be on 1/3 of the time, for a total of 300 TFLOPs – as much as the world’s fastest general purpose supercomputer in 2007, IBM’s Blue Gene / L.\(^1\) Since that time, the processing power of the fastest supercomputer has increased to 1800 TFLOPs or more.\(^2\) Volunteer counts for each project, as well as other statistics, are compiled and displayed on web pages as shown in Figure 3 and Figure 4. Both images are from circa 2007.

A BOINC project consists of several parts:

1. Data: Astropulse, in particular, processes data in 8 MB workunits, which consist of

\(^1\)http://www.top500.org/list/2007/06/100

\(^2\)http://www.top500.org/list/2009/11/100
1-bit complex sampled time series recorded at Arecibo.

2. Client code: runs on the volunteer’s machine.

   (a) The application client code, written by the researcher who owns that particular project. In the case of Astropulse, this is the code that performs the dedispersion.

   (b) The BOINC client code, written by the BOINC team at Berkeley. It relays the data to the application client and performs monitor and control functions, such as handling application client crashes.

3. Server code: runs on the researcher’s machines.

   (a) The validator, written by the researcher, checks whether a given workunit produces the same results when sent to different volunteers.

   (b) The assimilator, written by the researcher, records scientific results in a science database.

   (c) BOINC backend programs schedule workunits to be sent to different volunteers, run the validator and the assimilator, and record bookkeeping information in a separate BOINC database.

Figure 5 depicts the relationships between these functions.
5. RFI mitigation

6. Testing and Verification

7. Results and interpretation

8. Stardust@home

9. Suggestions for further research
Fig. 4.— Sample screen shot of statistics for SETI@home.

Fig. 5.— BOINC infrastructure. “Project back end” refers to the validator and assimilator. The scheduling server, utility programs, and web interfaces are the BOINC backend.