

Distribute Legacy Applications to Thousands of PC

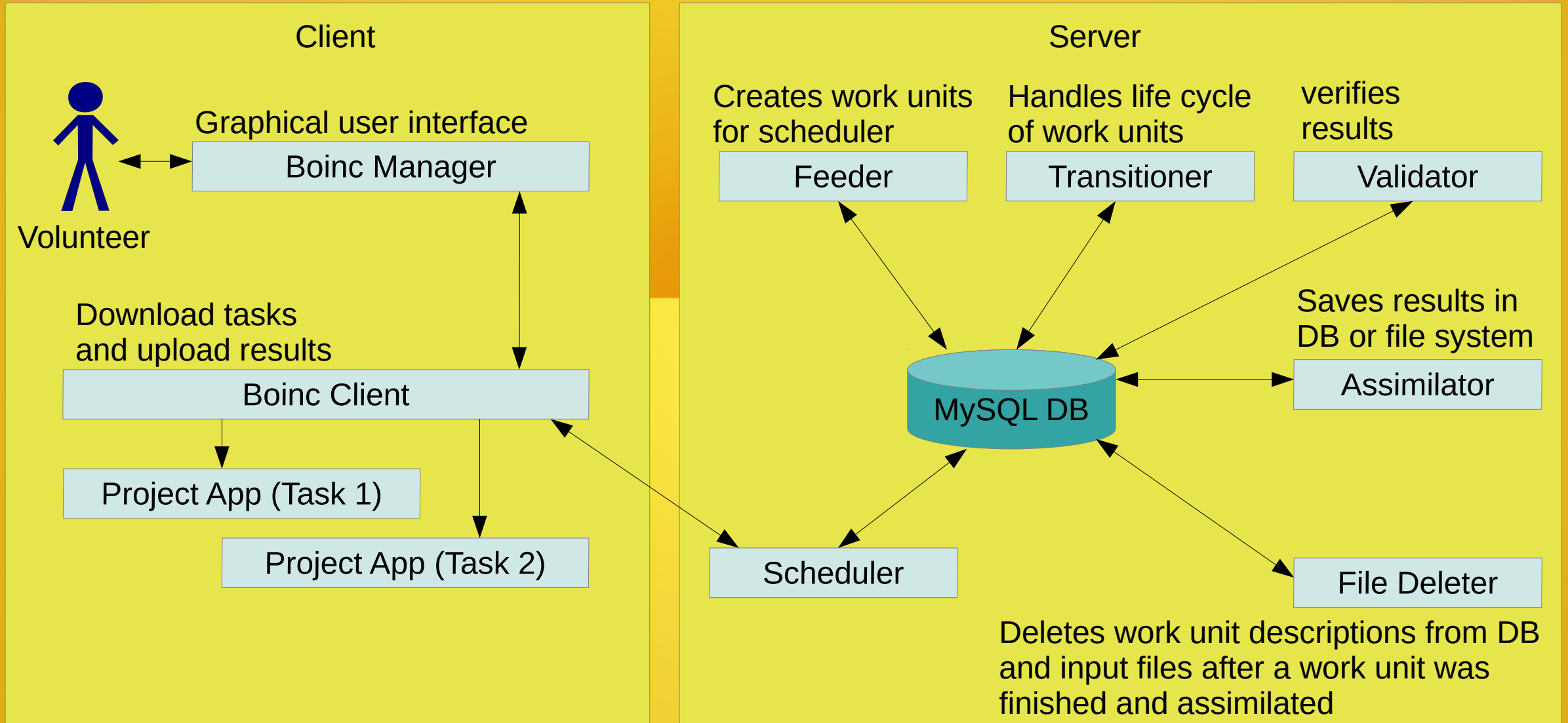
Boinc / [yoyo@home](#) / Wrapper

Berkeley Open Infrastructure for Network Computing

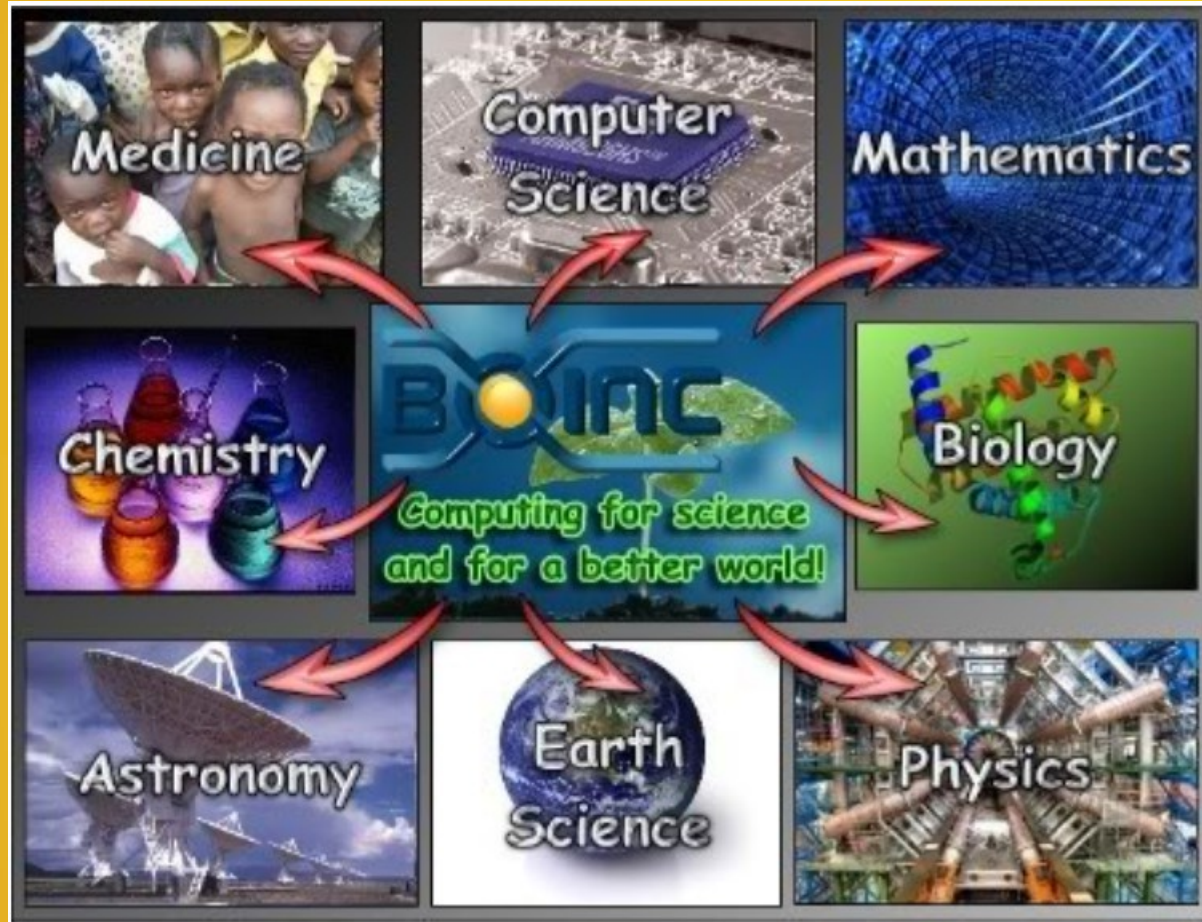


- open source middle ware system
- for volunteer and grid computing
- originally developed to support the SETI@home project
- For what can it be used?
 - a) to solve thousands of independent problems
 - b) a BIG problem which can be cut up to thousands of small problems

Components



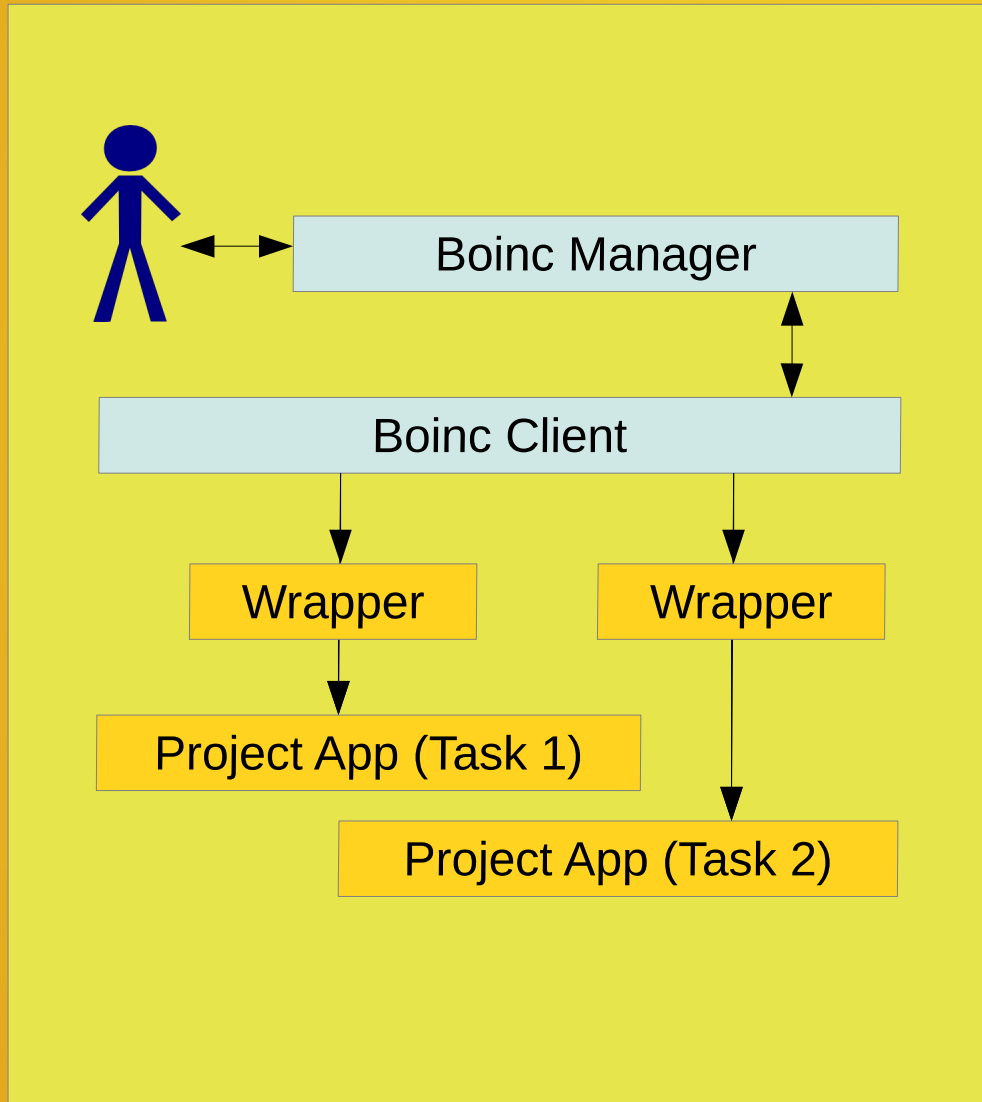
For what can it be used ?



Quake Catcher Network	Distributed sensing	Seismology	Stan
Radioactive@Home	Distributed sensing	Environmental research	BOI
RNA World	Biologie und Medizin	Molecular biology	Rec
Rosetta@home	Biologie und Medizin	Biology	Univ
SAT@home	Mathematics, computing, and games	Computer Science	Inst Inst Con of S
SETI@home	Astronomy, Physics, and Chemistry	Astrophysik, Astrobiologie	BOI Kali
SIMAP	Biologie und Medizin	Biology	Univ
Spinhenge@home	Astronomy, Physics, and Chemistry	Chemical engineering and nanotechnology	Biel Scie
sudoku@vtaiwan	Mathematics, computing, and games	Mathematics	Nati Taiw
Superlink@Technion	Biologie und Medizin	Genetic linkage analysis	Tec
Surveill@Home	Mathematics, computing, and games	Web performance	Univ
SZTAKI Desktop Grid	Mathematics, computing, and games	Mathematics	MTA and

- Run legacy applications on thousands of volunteer computers
 - Harmonious Trees
 - Elliptic Curve Factorization (includes 9 projects)
 - Muon
 - evolution@home
 - OGR (distributed.net)
 - Euler (6,2,5)

a) binary legacy application



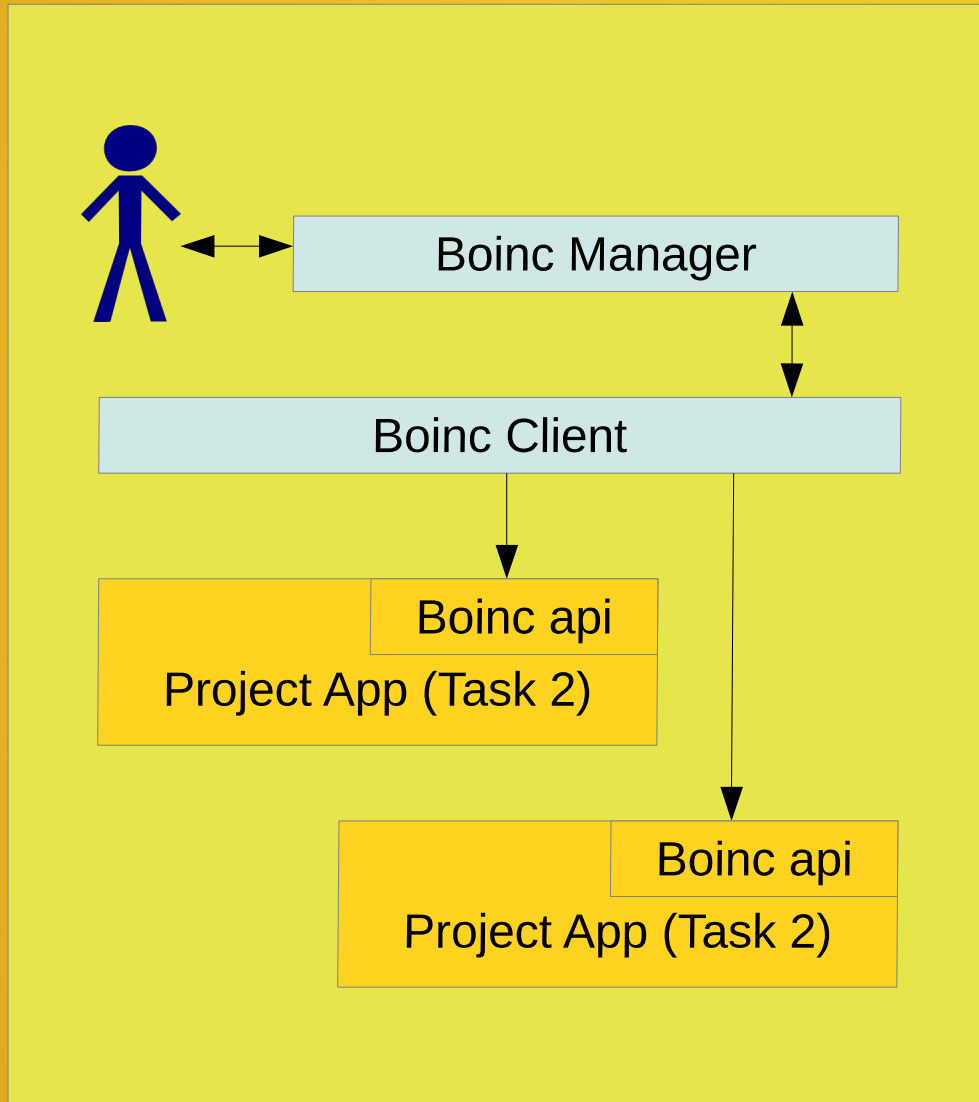
The source code of wrapper is in [boinc/samples](#). You can get pre-compiled versions here:

- [wrapper_26002_windows_intelx86.zip](#)
- [wrapper_26002_windows_x86_64.zip](#)
- [wrapper_26002_i686-pc-linux-gnu.zip](#)
- [wrapper_26002_x86_64-pc-linux-gnu.zip](#)
- [wrapper_26002_i686-apple-darwin.zip](#)
- [wrapper_26002_x86_64-apple-darwin.zip](#)

job.xml for wrapper

```
<job_desc>
  <task>
    <application>worker</application>
    [ <stdin_filename>stdin_file</stdin_filename> ]
    [ <stdout_filename>stdout_file</stdout_filename> ]
    [ <stderr_filename>stderr_file</stderr_filename> ]
    [ <command_line>--foo bar</command_line> ]
    [ <weight>X</weight> ]
    [ <checkpoint_filename>filename</checkpoint_filename> ]
    [ <fraction_done_filename>filename</fraction_done_filename> ]
  </task>
  <task>
    ...
  </task>
</job_desc>
```

b) legacy app with C source



Mandatory

- Add calls to BOINC initialization and finalization routines.
- Precede each fopen() call with a BOINC function that maps logical to physical names.
- Link it with the BOINC runtime library.

Optional

- Report fraction done
- Use configured checkpoint intervals
- Report checkpoints

BOINC API

Add calls to BOINC initialization and finalization routines.

```
#include "boinc_api.h"
int main(){
    boinc_init();
    ...
    boinc_finish(0);
}
```

Precede each fopen() with a BOINC function that maps logical to physical names.

```
// f = fopen("my_file", "r"); // replaced with
string resolved_name;
retval = boinc_resolve_filename_s("my_file", resolved_name);
if (retval) fail("can't resolve filename");
f = boinc_fopen(resolved_name.c_str(), "r");
```

BOINC API

Report fraction done

```
double m=1.0-(double)headcount/(double)totalheadcount;  
boinc_fraction_done(m); // 0 < m < 1
```

Use configured checkpoint intervals and report checkpoints

```
if (boinc_time_to_checkpoint()) {  
    fio=boinc_fopen("ckpt.txt", "w");  
    ... // save the data  
    fclose(fio);  
    boinc_checkpoint_completed();  
}
```

BOINC API

Link it with the BOINC runtime library.

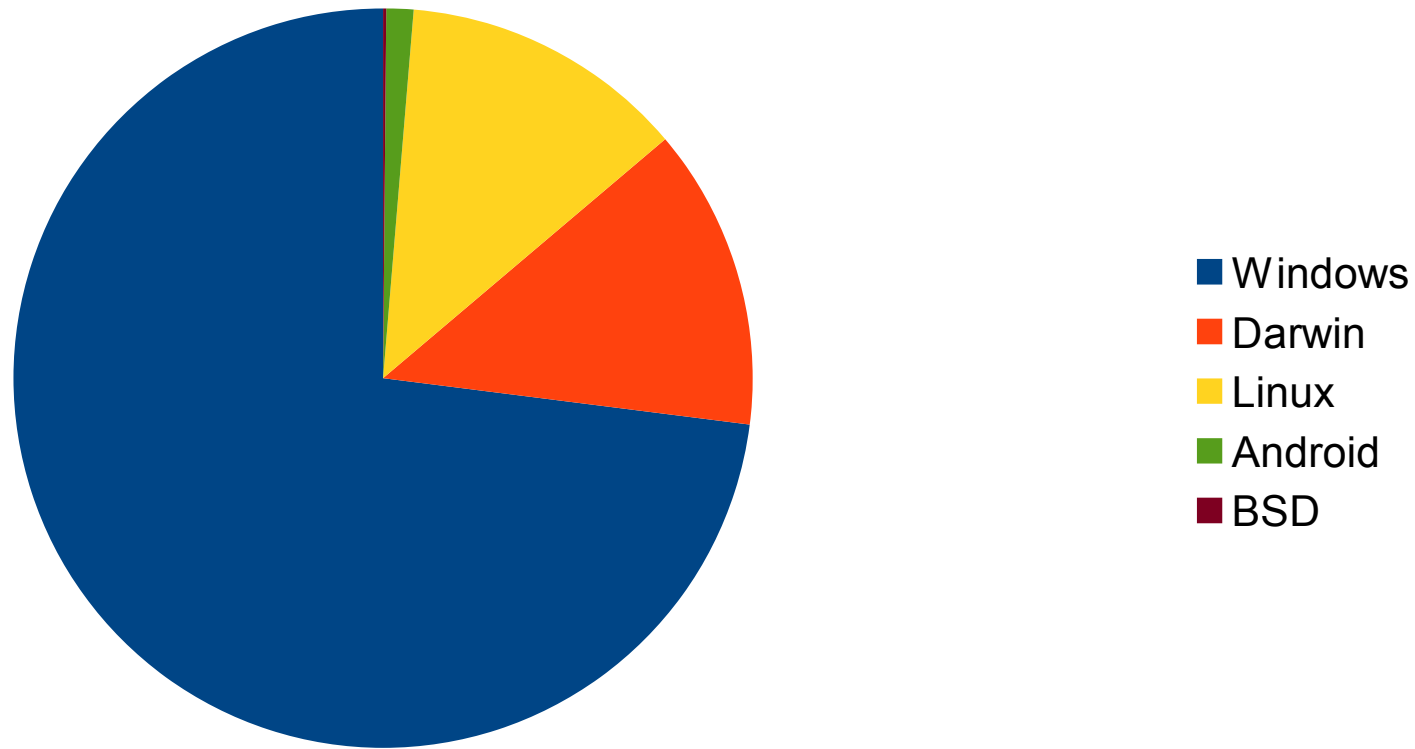
```
BOINC_DIR = ../../boinc
BOINC_API_DIR = $(BOINC_DIR)/api
BOINC_LIB_DIR = $(BOINC_DIR)/lib
CXXFLAGS = -O3 --static -static-libgcc \
  -I$(BOINC_LIB_DIR) -I$(BOINC_API_DIR) \
  -L$(BOINC_API_DIR) -L$(BOINC_LIB_DIR) \
  -L.
```

```
app: app.o libstdc++.a $(BOINC_LIB_DIR)/libboinc.a $(BOINC_API_DIR)/libboinc_api.a
g++ $(CXXFLAGS) -o $@ app.o libstdc++.a -pthread -lboinc_api -lboinc -static-libgcc
strip $@
```

Problems

- Application must be self contained
- Link with `-static-libgcc`
- Runtime 1 – 10h
- Checkpoints
- Progress indicator
- Result validation (quorum, Linux/Windows/Mac eol)
- Credits
- Volunteers
 - Regular news, answer questions in forum
 - Be honest and communicate open also about problems

Which Platform to support



yoyo@home - HarmoniousTrees, results on 20.2.2013

Ideas - Brainstorming

- Ready made Boinc application for bio/med open source tools, e.g.:
 - mfold (www.bioinfo.rpi.edu/applications/mfold)
 - autodock (autodock.scripps.edu)
 - mopac (openmopac.net)
 - gromacs (gromacs.org)
 - scilab
 - ...

