The 7th Multiversal BOINC Workshop

Hannover, Germany
18-19 August 2011

The BOINC community

Projects

UC Berkeley developers (2.5)

PC volunteers (300,000)

Other volunteers: testing translation support

Computer scientists

Workshop goals

- Everyone learns what everyone else is doing
- Form collaborations
  - don’t be shy!
- Plan BOINC development
  - tell us what you want

Hackfest (tomorrow)

- Topics
  - multi-user projects
  - VM apps
  - distributed storage
  - ...
- Goal: get something concrete done
  - Improve docs
  - design and/or implement software

The state of volunteer computing

- Volunteers: down by about 15% last 6 months
  - 290K people, 450K computers
- Science projects: stagnant
  - prime numbers and cryptosystems
- Computer science research: stagnant
- My viewpoint: we built it and they haven't come. But let's keep building anyway.

To projects:

- Do outreach
  - notices
  - automated emails
  - mass emails
  - message boards
  - mass media
- Use current server code

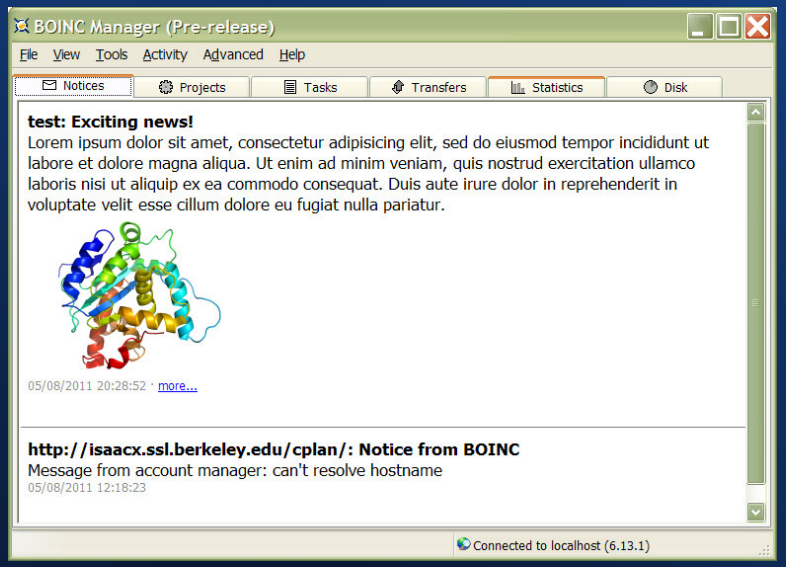
To developers/researchers:

- Talk with me before starting anything
  - especially if it's of general utility

davea@ssl.berkeley.edu

What’s new in BOINC?

Notices



RSS

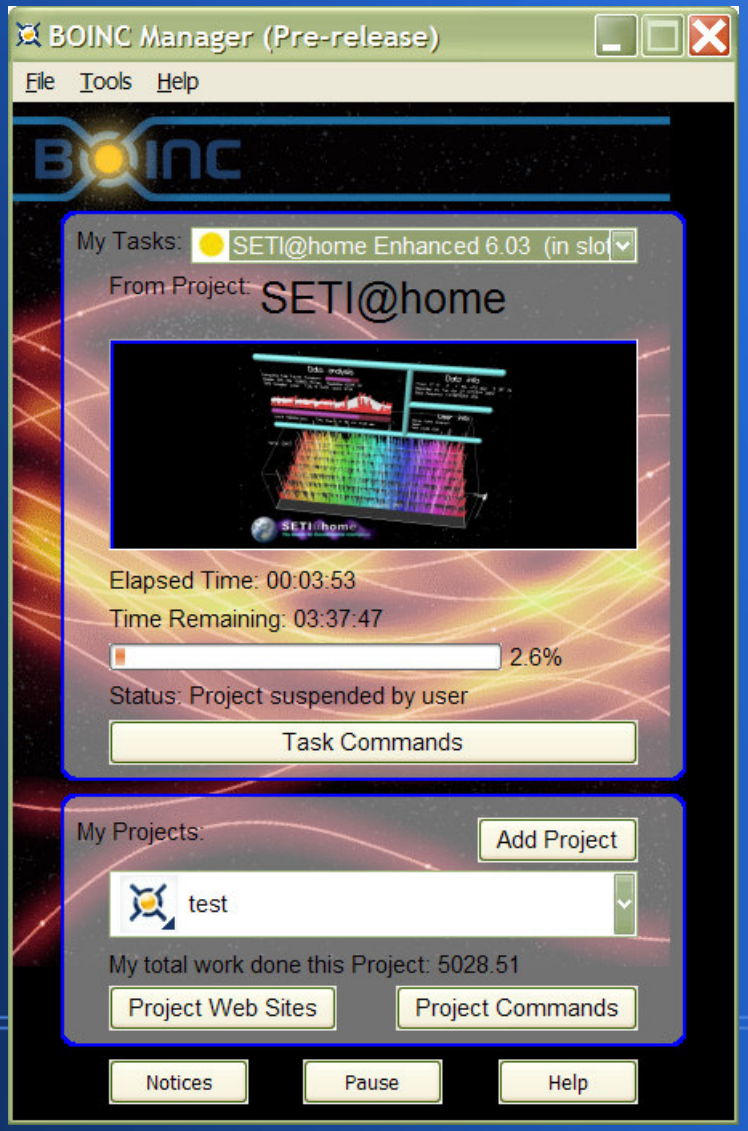
Projects:
- news
- notifications
- message boards

client

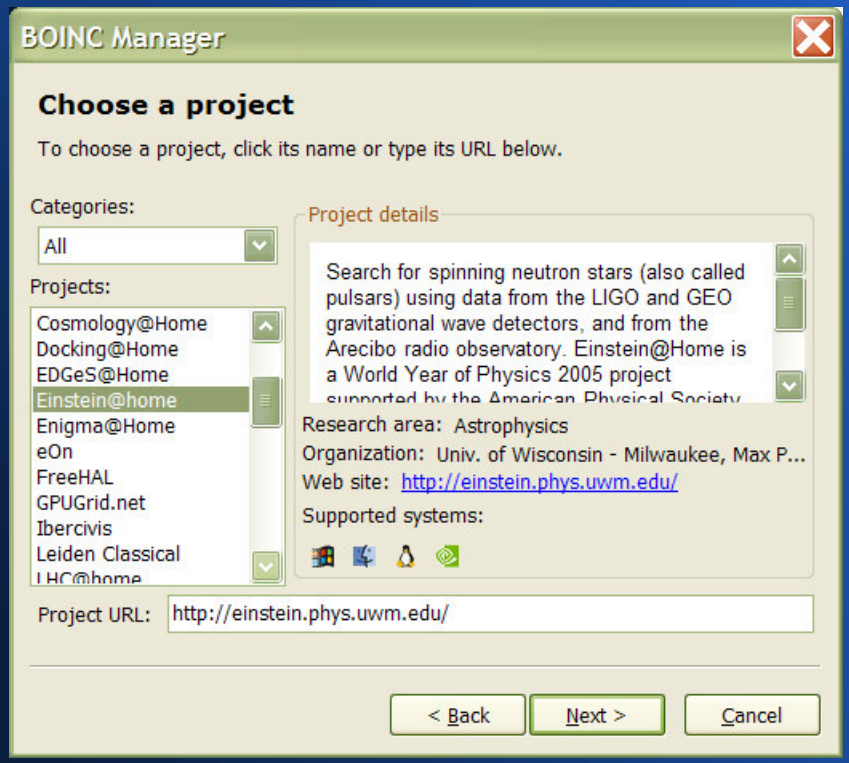
scheduler

Simple view

- Accessible
- Translatable
- Simpler skinning

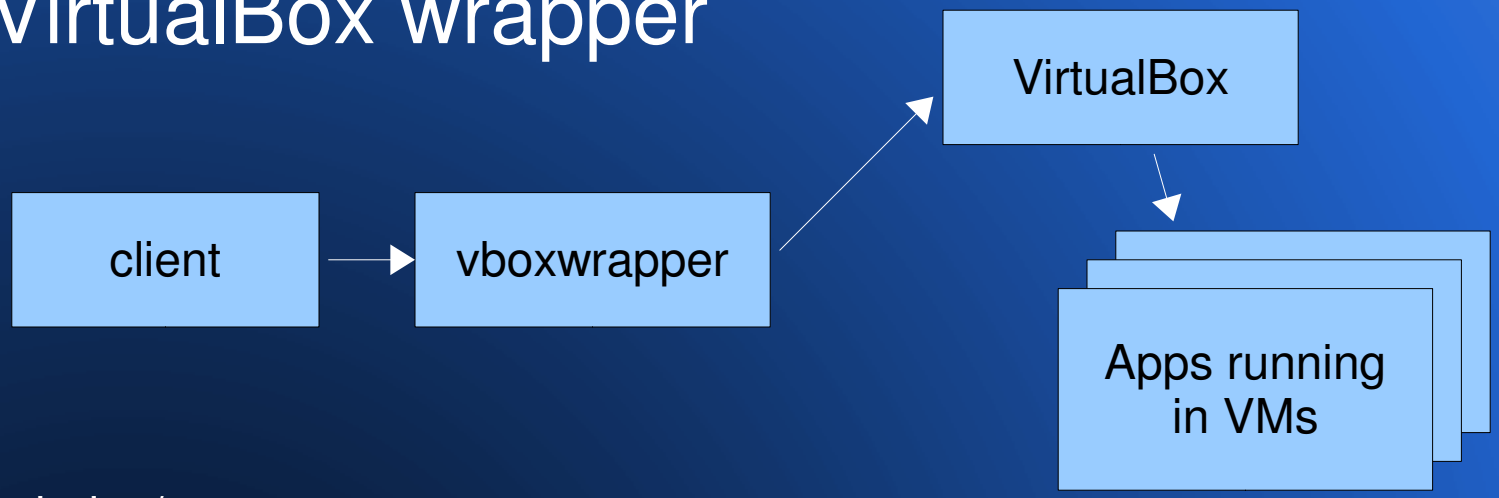


Choose-project dialog



Support for VM apps

- VirtualBox wrapper




```
boinc/
    slots/
        0/
            vm_image.vdi
            share/
                input, output files
```

OpenCL support

- Client
  - detects and reports OpenCL version
- Scheduler
  - opencl plan class

Generalized GPU support

- Old: NVIDIA and ATI only, hardwired

```
<ati_req>1</ati_req>
```

- New: arbitrary GPU types

```
<req>
    <type>ati</type>
</req>
```

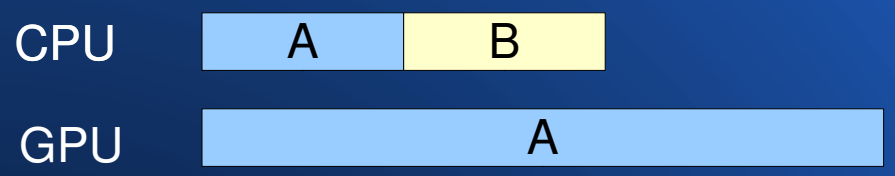
- Config file can specify GPUs with new types, and BOINC will schedule them correctly

Hysteresis work fetch

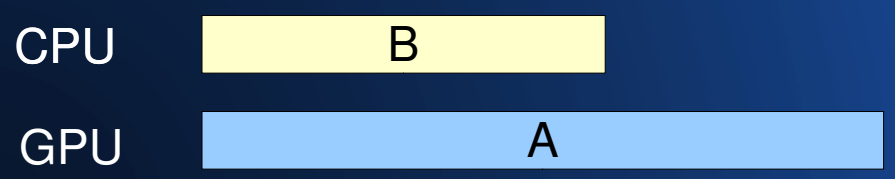
- Reduce # of scheduler requests
- Per processor type:

Client scheduling improvements

- Old: resource share enforced per processor type



- New: resource share enforced across all processor types



Cleanup of multiprocess jobs

- To stop a job
  - enumerate its descendant processes
  - ask main process to quit
    - kill it if needed
  - kill descendants

Improved update_versions

- Old:

```
apps/appname/
   uppercase_6.15_windows_intelx86__cuda.exe/
      graphics_app=uppercase_graphics_6.14_windows_intelx86.exe
      ...
```

- New:

```
apps/appname/
   6.14/
   6.15/
      windows_intelx86/
      windows_intelx86__cuda/
         version.xml
         uppercase_6.15_windows_intelx86.exe
         ...
```

BOINC client emulator

- real scheduling code + simulation of scheduler RPCs and job execution
- Input: “scenario”, described by a client state file
- Output: 4 figures of merit, event log, HTML timeline
- Uses:
  - develop and evaluate scheduling policies
  - make real-world situations reproducible
    - Web interface to emulator

> 2GB RAM jobs on 32-bit hosts

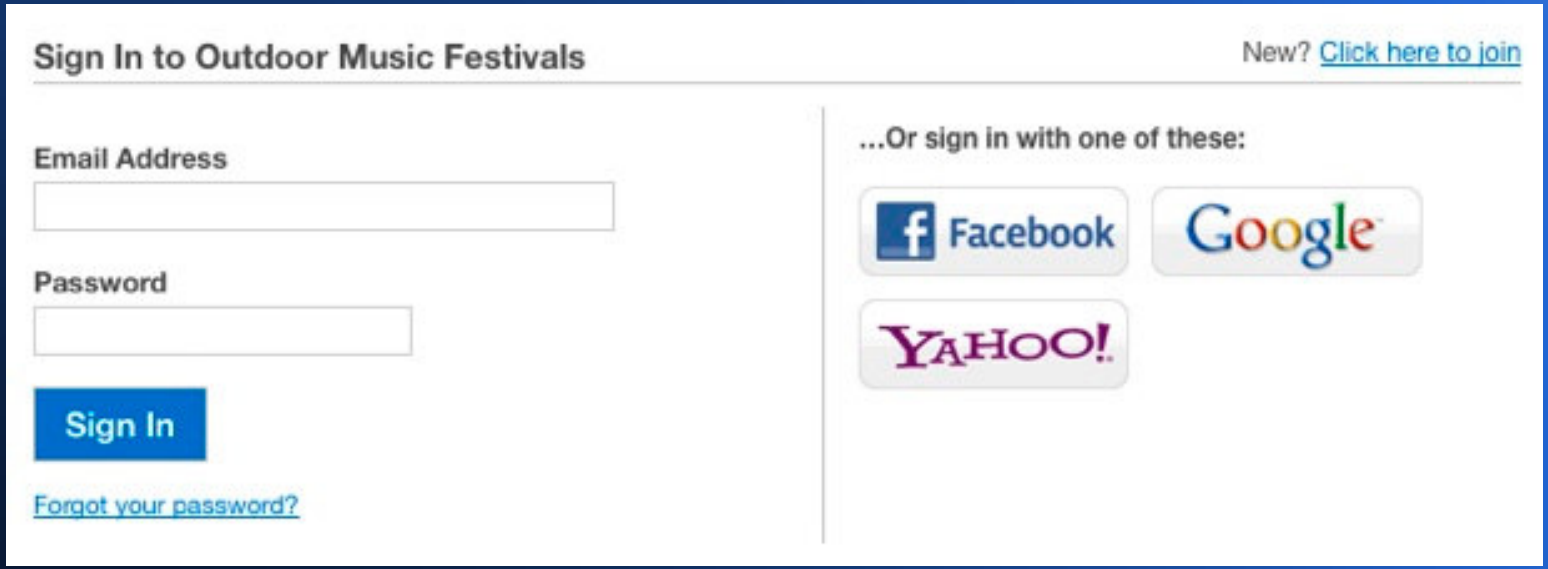
- User address space limits for 32-bit apps:
  - Windows: 2 GB
  - Linux: 3 GB
  - Mac OS X: 4 GB
- Scheduler dispatches > 2GB jobs accordingly

Homogeneous app version

- Lets you specify that all instances of a given job should be done with the same app version
- Use, e.g., if GPU versions don’t validate against CPU versions
- Selectable per app

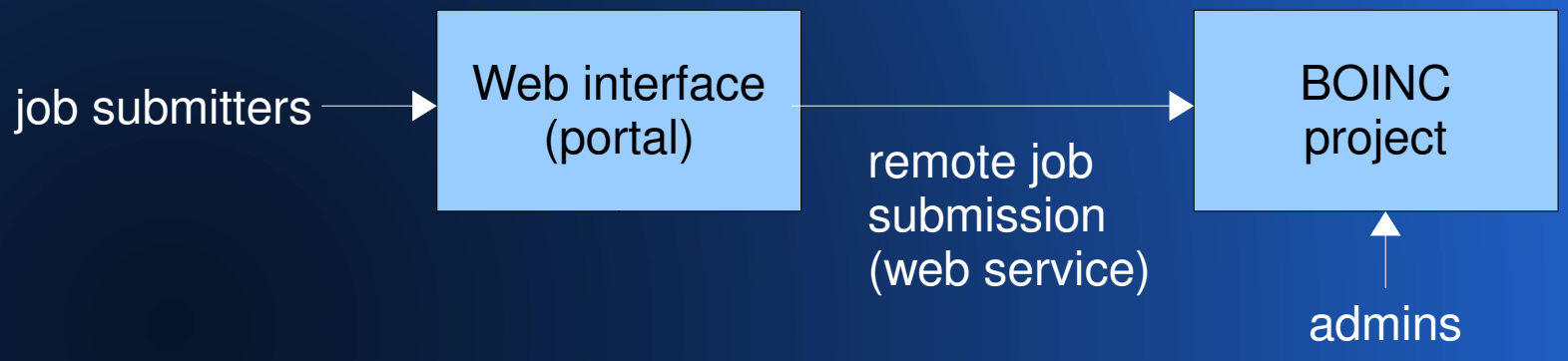
In progress

OpenID support



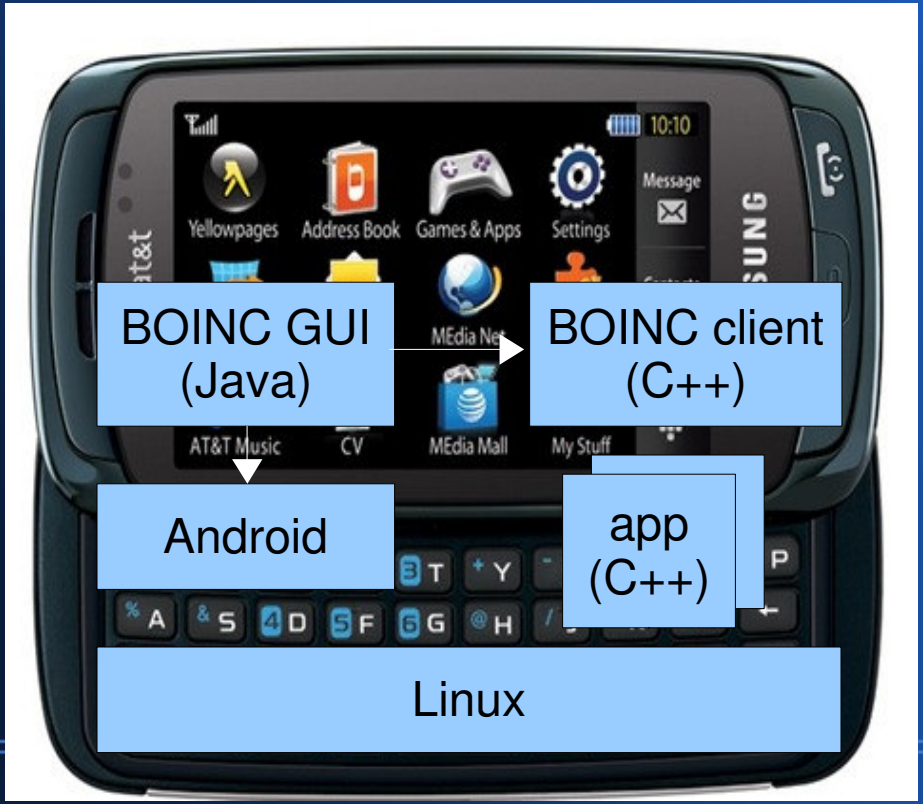
Multi-user projects

- Job submitters have user accounts
- Accounts have quotas
- Access control system
- Remote job submission



BOINC on Android

- 5 billion mobile systems:
  - 2 GFLOPS, 32 GB stable storage, 1 GB RAM



Volunteer storage

storage applications

locality scheduling

data stream buffering

pure storage

result archival

dataset storage

BOINC infrastructure

- DB table of files and instances
- Info on host availability and churn
- File transfers: client/server, maybe client/client
- Share-based space allocation on clients

Scheduling (server)

- Batch scheduling
  - makespan minimization
  - dynamic completion estimates
- Unification
  - Throughput-oriented (job cache)
  - Locality scheduling
  - Co-scheduling (Volpex)
  - Batch