BOINC Server

What it is, how it works and how to create one

Introduction to the possibilities offered

from the BOINC platform

Version 1.1

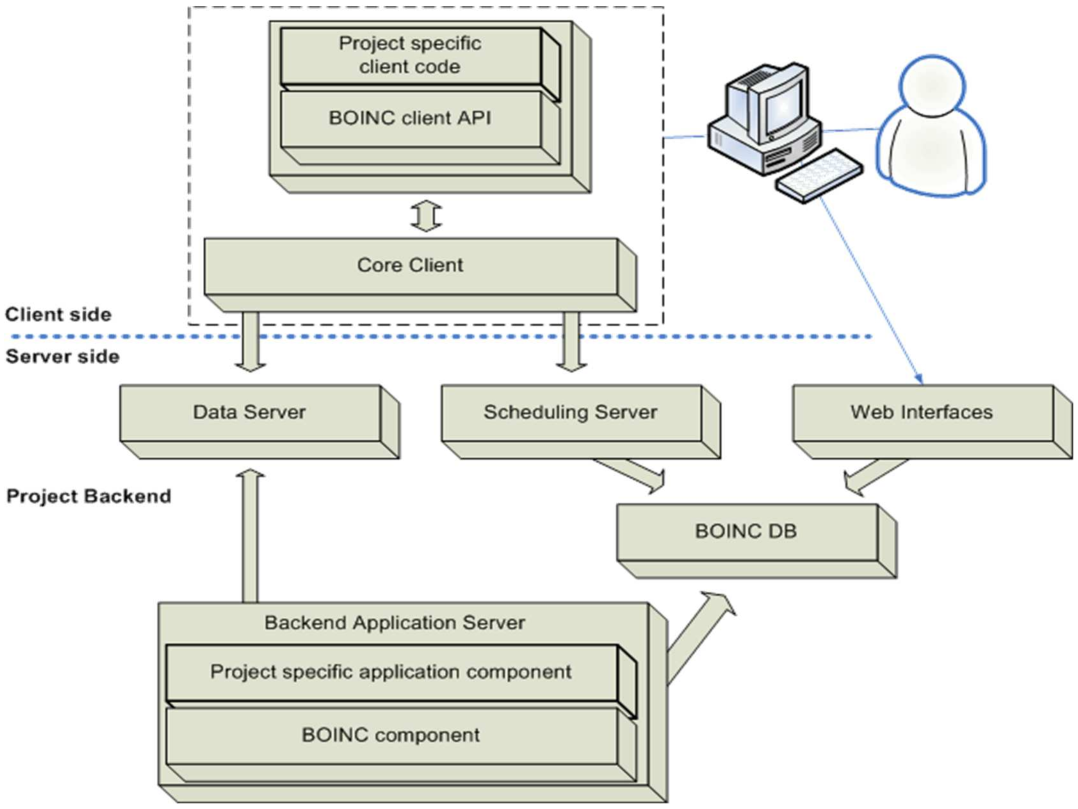
Bologna Stefano

boboviz chiocciola boincitaly punto org

Index

What is BOINC?

Boinc is the environment of voluntary distributed processing more known and used internet. Its use for scientific research purposes has made it known to the academic and research world, especially in case you need a computing power *huge* but not the adequate economic or logistical coverage (or even the skills) to make it happen. The client / server system, based on a LAMP structure, can be schematized as follows:



When can BOINC be used and when not

Given a certain problem to be solved thanks to a computer, in order to obtain results as quickly as possible, there are two possible ways: to increase the power of the single processor (i.e. increase the calculation frequency, the instructions per clock cycle or the cores that compose), or make sure that our program is able to exploit the power of multiple processors at the same time, or capable of performing parallel computing.

Given that the most traveled road today, in science, is the second (just look at the characteristics of the best supercomputers in the world - Top500), the question is: what types of parallelization are there? We can easily distinguish three types based on how much individual processes are dependent on each other:

1. Extremely Dependent Processes: Processes need to communicate information with each other in a way *continuous* and they cannot proceed without the data provided by the other processes. It is essential that all processes run simultaneously and on processors capable of exchanging information immediately (example: cluster of multicore processors).

2. Low-dependent processes: processes must 'talk' to each other from time to time. Processes need to run concurrently, but communication speed is more or less important depending on how often you talk to each other. In this case we can exploit, for example, the power of two computers connected via ethernet (or perhaps Infiniband). This is the type of processing that data centers allow where large quantities of interconnected nodes are made available.

3. Non-dependent processes: in the latter case, the individual processes do not need to exchange information with the others. Each process can therefore be performed on a single computer without the need for any simultaneity with the other processes.

In order to parallelize our processing through BOINC it is necessary to fall back into the latter case.

Let us now assume that our type of processing is exactly what is required, or that the problem we have to solve can be reformulated in such a way that it falls back on: are there other constraints for processing through BOINC?
When processing via Boinc it is always useful to remember that applications will run on the computer's *volunteers* , therefore with heterogeneity of operating systems and hardware, various versions of the libraries, the possible low bandwidth in upload / download, etc., should be factors to be **evaluate carefully.**
Some requirements therefore are:

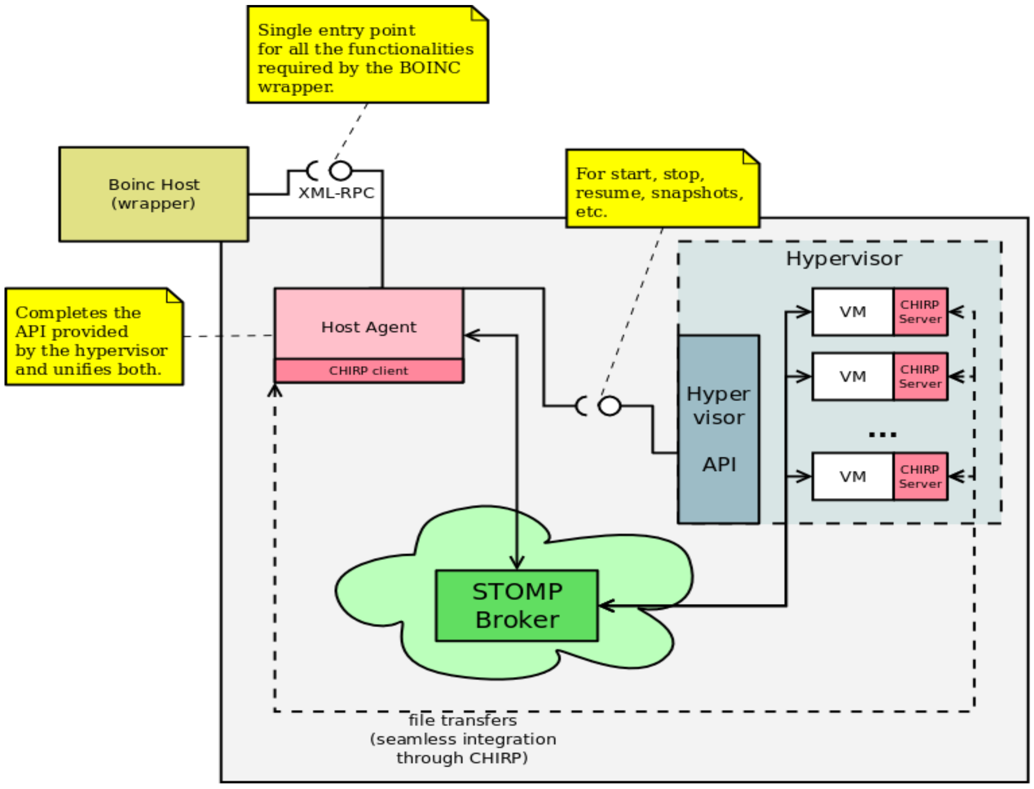
1. The program that is distributed through the internet must not "weigh" (in terms of MB) too much. The ideal executable is the one you hardly notice, therefore in the size of a few tens of MB.
2. The same bandwidth problem occurs with the single work units. The input files must try to be light, even more so than for the application (this is downloaded only once, the input files repeatedly). You should also be careful with the output files; usually the available upload bandwidth is much lower than the download band.

3. Still on the program, using Boinc is being distributed

publicly that certain compiled code: you have to be sure you have legal permissions to do so (projects, in most cases, use open source software).

4. The program must be able to run on any computer. For example, creating an application for Gnu / Linux operating systems, this must not depend on the versions of the libraries installed on the user's machine (even if the problem can be solved by compiling the executables statically). It is necessary to take for granted **not** be able to
5. trust users' computers: they could be bugged, produce calculation errors, executables could be tampered with by users (cheating), etc. If possible, use a way that verifies that the results returned by users are correct, for example by having the same job run by two different computers (job quorum).

To some of these problems (for example the need to have some particular libraries or configurations) it is also possible to answer by using Virtualbox virtual machines (in the link, in addition to the guide to create them, there are also updated boinc wrappers), which will start up as if they were a normal WU, through the virtualization wrapper present (there is also the possibility to use Docker). The scheme is this:


Single entry point for all the functionalities required by the BOINC wrapper.
Boinc Host (wrapper)
XML-RPC
For start, stop, resume, snapshots, etc.
Hypervisor
Completes the API provided by the hypervisor and unifies both.
Host Agent
CHIRP client
VM
CHIRP Server
Hyper visor API
...
STOMP Broker
file transfers (seamless integration through CHIRP)

The server: installation from preconfigured image

The basic installation and configuration of a Boinc server is very simple and within everyone's reach.

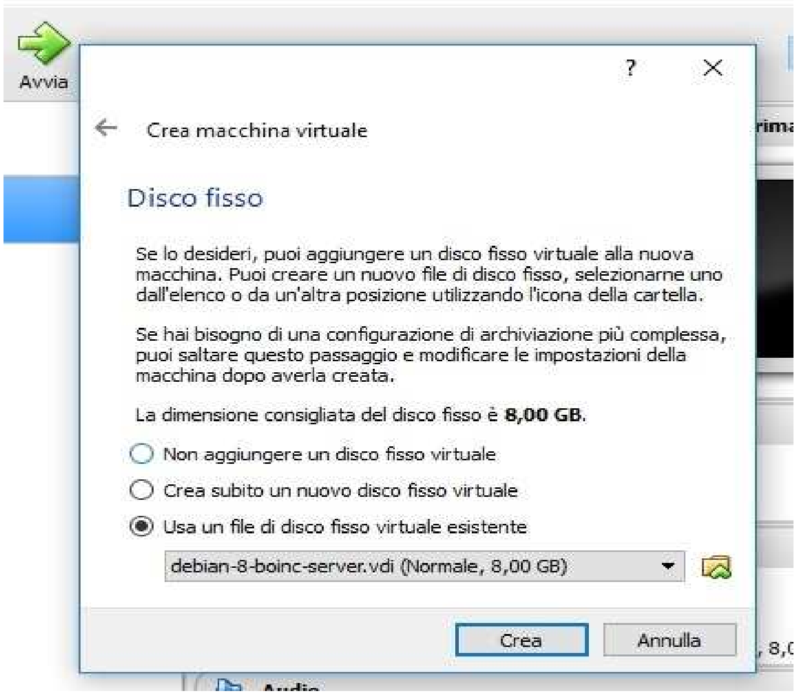
As for the hardware, the creation of a Boinc server has very low demands: you need a 64-bit processor, 2 GB of RAM and at least 40 GB of free disk space (obviously as the project grows, the infrastructure will have to grow. which supports it).

In fact, it is possible to create a physical or virtual server of your choice and also choose whether to use a server with your own customizations, or one already prepared by the Boinc team.

As for the operating system, any GNU / Linux distribution is required (Boinc administrators recommend Debian or, secondly, Ubuntu). The latest available version of the Boinc server is 1.2.1 (dated 06/03/2020).

If your distribution is Debian, there is a specific guide up to version 8 (here), while for the preconfigured disk image the steps are as follows.

1) Download the image from the site of Boinc, https://boinc.berkeley.edu/trac/wiki/VmServer
2) The file is in .vdi format, which is the default disk file of the Oracle VirtualBox virtualization program (downloadable for free from https://www.virtualbox.org/wiki/Downloads).
3) Start VirtualBox and create a new virtual machine, assigning it a name and an adequate amount of ram memory (min 2gb). Choose the Boinc image disk as the system disk.



Disco fisso

Se lo desideri, puoi aggiungere un disco fisso virtuale alla nuova macchina. Puoi creare un nuovo file di disco fisso, selezionarne uno dall'elenco o da un'altra posizione utilizzando l'icona della cartella.

Se hai bisogno di una configurazione di archiviazione più complessa, puoi saltare questo passaggio e modificare le impostazioni della macchina dopo averla creata.

La dimensione consigliata del disco fisso è **8,00 GB**.

- ○ Non aggiungere un disco fisso virtuale
- ○ Crea subito un nuovo disco fisso virtuale
- ● Usa un file di disco fisso virtuale esistente

debian-8-boinc-server.vdi (Normale, 8,00 GB)

4) Configure the network in "Bridge" mode with the physical network card of the computer, be it wireless or wired.

5) Start the virtual machine and log in with the 'root' account (pwd: 'rootpw'). With this account proceed to update the system with commands

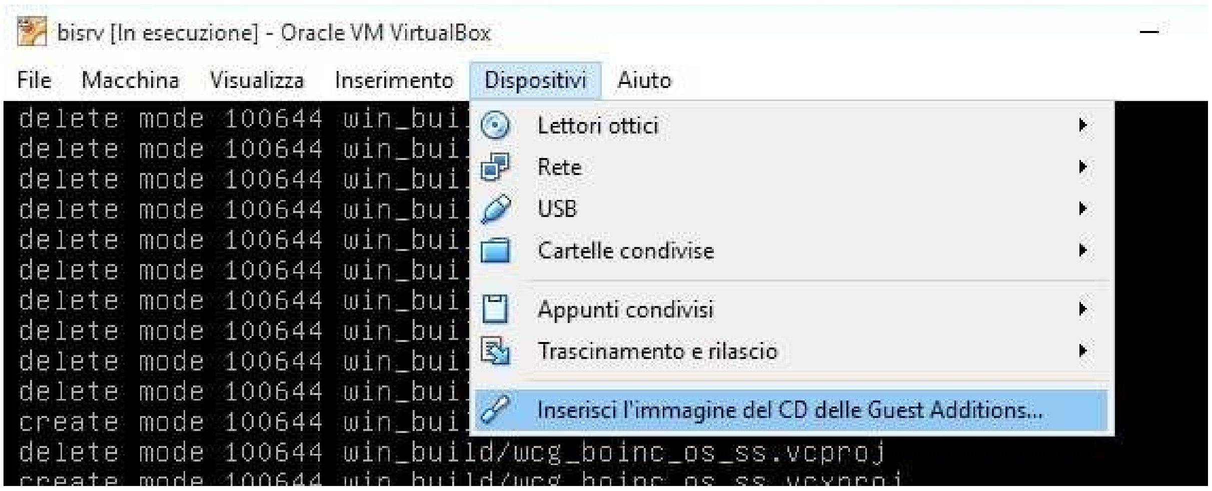
apt-get update, apt-get upgrade is apt-get dist-upgrade

(optional) - DO NOT upgrade to Debian version 9 . To download the correct updates, it is recommended that you configure the mirrors correctly

In the file / *etc / atp / sources.list.* The Nano editor is already present in the system and it is advisable to install an ftp service for any download / upload of files on the server (ProFtpd works well in this regard).

6) Change the keyboard language with the command dpkg-reconfigure keyboard-configuration, choose the Italian QWERTY keyboard. Restart the keyboard service or the server directly.

7) Install VirtualBox Guest Additions (in case you want to switch to a graphical interface - not recommended), loading the image from the VirtualBox interface



```
delete mode 100644 win_bui
delete mode 100644 win_bui
delete mode 100644 win_bui
delete mode 100644 win_bui
delete mode 100644 win_bui
delete mode 100644 win_bui
delete mode 100644 win_bui
delete mode 100644 win_bui
delete mode 100644 win_bui
delete mode 100644 win_bui
create mode 100644 win_bui
delete mode 100644 win_build/wcg_boinc_os_ss.vcproj
create mode 100644 win_build/wcg_boinc_os_ss.vcxproj
```

Then mount the cd (it will probably be sr0, so the command will become

mount - t / dev / sr0 / media / cdrom) and launch the command

. /VboxLinuxAddition.run. Restart the server.

8) Assign a static IP to the server and configure the network appropriately, editing the file / *etc / network / interfaces,* by entering the following syntax

```
iface eth0 inet static
  address 172.100.204.9
  netmask 255.255.255.0
  gateway 172.100.204.1
  dns-nameservers xywz
```

(all these addresses are examples. The dnsnameserver will be the dns server of the network)

9) Once the updates are complete, reboot and log in with the 'boincadm' user (pwd: 'boincadmpw'). Change the passwords of both accounts used (with the sudo passwd command). **The 'boincadm' account will be the account under which all necessary operations will be conducted on the server .**

10) Update the Boinc server software to the latest version (recommended):

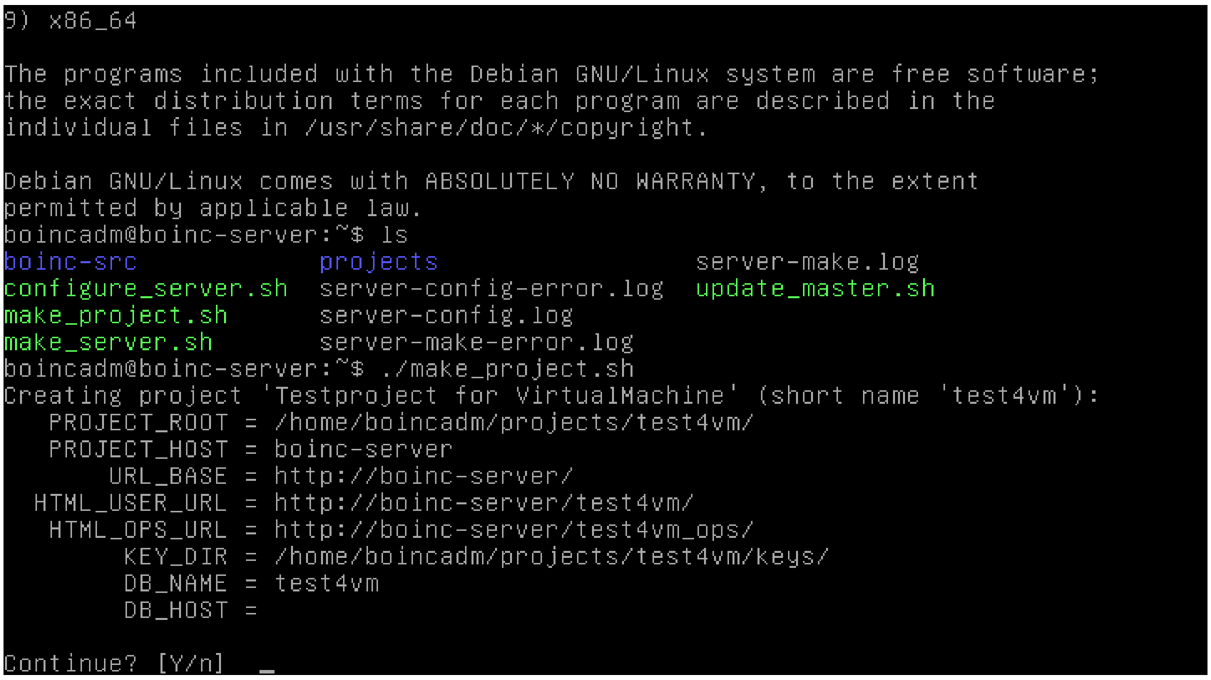
    $ ./update_master.sh

11) Start the creation of the Boinc server, with the related services:

    $ ./configure_server.sh
    $ ./make_server.sh

12) At this point the server is ready to host a Boinc project. The command to create it is

    $ ./make_project.sh

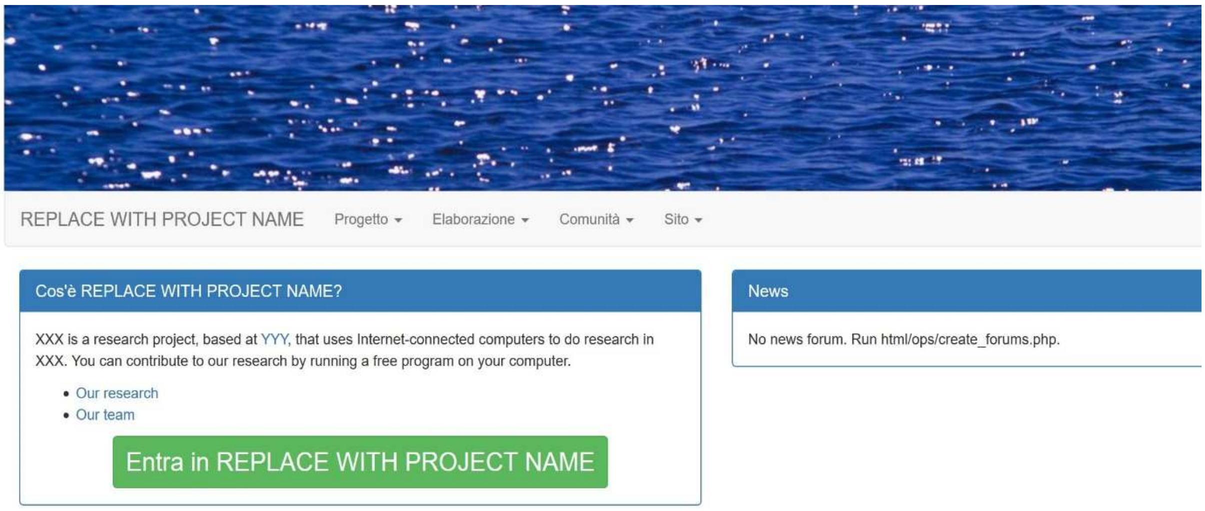


```
9) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
boincadm@boinc-server:~$ ls
boinc-src            projects                 server-make.log
configure_server.sh  server-config-error.log  update_master.sh
make_project.sh      server-config.log
make_server.sh       server-make-error.log
boincadm@boinc-server:~$ ./make_project.sh
Creating project 'Testproject for VirtualMachine' (short name 'test4vm'):
   PROJECT_ROOT = /home/boincadm/projects/test4vm/
   PROJECT_HOST = boinc-server
       URL_BASE = http://boinc-server/
  HTML_USER_URL = http://boinc-server/test4vm/
   HTML_OPS_URL = http://boinc-server/test4vm_ops/
        KEY_DIR = /home/boincadm/projects/test4vm/keys/
        DB_NAME = test4vm
        DB_HOST =

Continue? [Y/n]  _
```

The server will now also be reachable from the default web interface open to the public and the name of the project will be the one predefined by the server, that is "test4vm" (if you want to change the name of the project, or other parameters, it is just edit the file *config.xml* created by the command make_project). To make the system "consistent", check the server name (which by default is "boinc-server") in the file / *etc / hosts* matching it with the ip address. The same change of the ip address must be done manually also in the file */home/boincadm/projects/test4vm/config.xml,* in the following way:

<upload_url> http: // ipaddress / test4vm_cgi / file_upload_handler </upload_url> <download_url> http: // ip address / test4vm / download </download_url>

<master_url> http: // ip address / test4vm / </master_url>.



Cos'è REPLACE WITH PROJECT NAME?

XXX is a research project, based at YYY, that uses Internet-connected computers to do research in XXX. You can contribute to our research by running a free program on your computer.

- Our research
- Our team

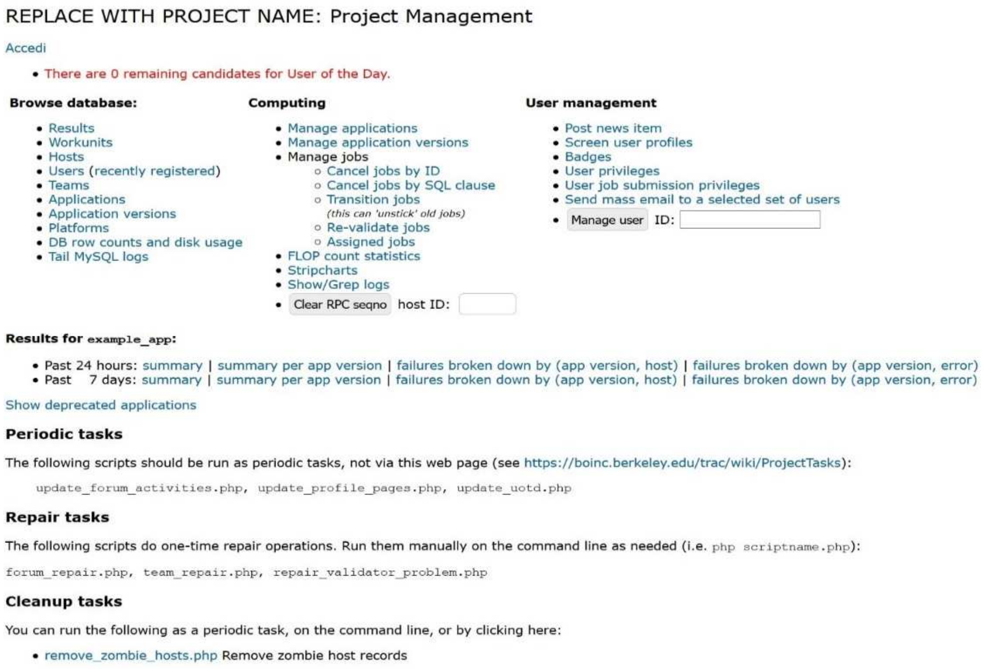
News

No news forum. Run html/ops/create_forums.php.

13) You can access the web administrative interface through the url *http: // ipaddress / test4vm_ops.* This administrative interface is protected from unauthorized access. To give access permissions you need to create, in the folder / *projects / test4vm / html / ops* an .htpasswd file with the following command

htpasswd -c .htpasswd **username (** it is preferable to use the boincadm user, to give consistency to everything).

14) This interface will allow you to control a whole series of parameters, such as versioning of applications (deprecated / new, etc.), user control, forum control, etc., etc.


REPLACE WITH PROJECT NAME: Project Management

Accedi

- There are 0 remaining candidates for User of the Day.

Browse database:
- Results
- Workunits
- Hosts
- Users (recently registered)
- Teams
- Applications
- Application versions
- Platforms
- DB row counts and disk usage
- Tail MySQL logs

Computing
- Manage applications
- Manage application versions
- Manage jobs
  - Cancel jobs by ID
  - Cancel jobs by SQL clause
  - Transition jobs
    (this can 'unstick' old jobs)
  - Re-validate jobs
  - Assigned jobs
- FLOP count statistics
- Stripcharts
- Show/Grep logs
- Clear RPC seqno host ID:

User management
- Post news item
- Screen user profiles
- Badges
- User privileges
- User job submission privileges
- Send mass email to a selected set of users
- Manage user ID:

Results for `example_app`**:**
- Past 24 hours: summary | summary per app version | failures broken down by (app version, host) | failures broken down by (app version, error)
- Past 7 days: summary | summary per app version | failures broken down by (app version, host) | failures broken down by (app version, error)

Show deprecated applications

Periodic tasks

The following scripts should be run as periodic tasks, not via this web page (see https://boinc.berkeley.edu/trac/wiki/ProjectTasks):

```
update_forum_activities.php, update_profile_pages.php, update_uotd.php
```

Repair tasks

The following scripts do one-time repair operations. Run them manually on the command line as needed (i.e. `php scriptname.php`):

```
forum_repair.php, team_repair.php, repair_validator_problem.php
```

Cleanup tasks

You can run the following as a periodic task, on the command line, or by clicking here:

- remove_zombie_hosts.php Remove zombie host records

15) To create the project forum, you need to edit the create_forum.php file in the / folder *html / ops,* deleting the whole line

die (... ..) and editing the various text fields by entering the desired descriptions.

16) Once the file is saved, create the forum with the command

php create_forum.php

17) Inside the folder / *projects / test4vm* there are two folders

important: **download** is **upload.** The first is where the application will go

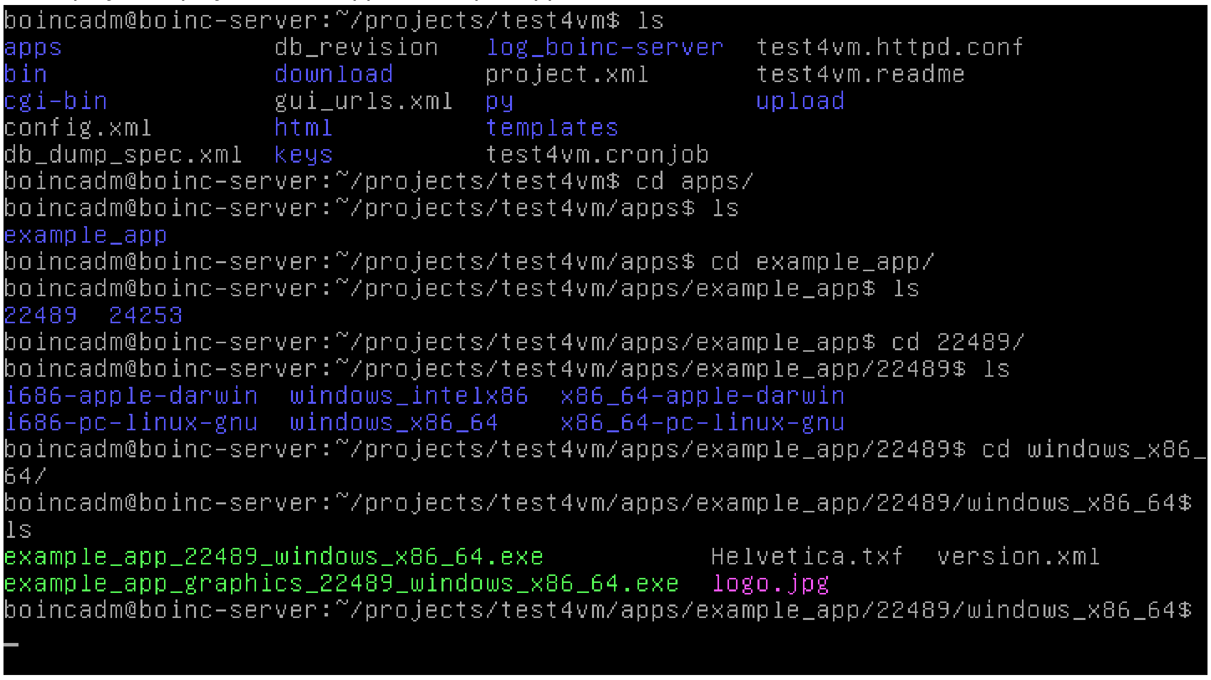
+ the data you want to distribute to the clients, while the processed data will be received in the upload folder.

18) Inside the folder / *html / ops* there are also a whole series of

php scripts useful for server administration.

19) The server will already have a sample application on board ready to be used for any tests. Sample files are present in the folder

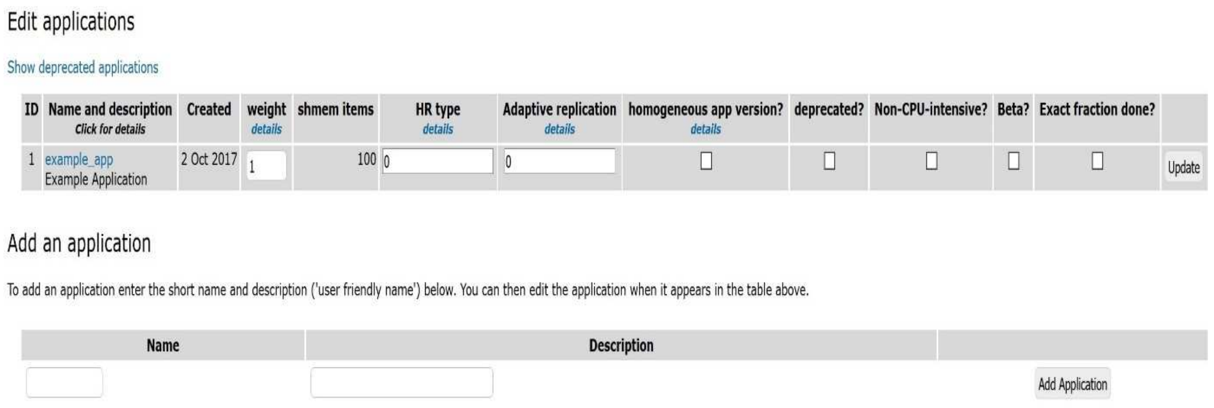
/ projects / projectname / apps / example_app /



```
boincadm@boinc-server:~/projects/test4vm$ ls
apps                db_revision    log_boinc-server   test4vm.httpd.conf
bin                 download       project.xml        test4vm.readme
cgi-bin             gui_urls.xml   py                 upload
config.xml          html           templates
db_dump_spec.xml    keys           test4vm.cronjob
boincadm@boinc-server:~/projects/test4vm$ cd apps/
boincadm@boinc-server:~/projects/test4vm/apps$ ls
example_app
boincadm@boinc-server:~/projects/test4vm/apps$ cd example_app/
boincadm@boinc-server:~/projects/test4vm/apps/example_app$ ls
22489  24253
boincadm@boinc-server:~/projects/test4vm/apps/example_app$ cd 22489/
boincadm@boinc-server:~/projects/test4vm/apps/example_app/22489$ ls
i686-apple-darwin  windows_intelx86  x86_64-apple-darwin
i686-pc-linux-gnu  windows_x86_64    x86_64-pc-linux-gnu
boincadm@boinc-server:~/projects/test4vm/apps/example_app/22489$ cd windows_x86_
64/
boincadm@boinc-server:~/projects/test4vm/apps/example_app/22489/windows_x86_64$
ls
example_app_22489_windows_x86_64.exe           Helvetica.txf   version.xml
example_app_graphics_22489_windows_x86_64.exe  logo.jpg
boincadm@boinc-server:~/projects/test4vm/apps/example_app/22489/windows_x86_64$
_
```

20) At this point the server is almost ready to distribute the work. It is necessary to "add" the example application to the project, from the folder

/ projects / projectname with the command ./ bin / xadd.



Edit applications

Show deprecated applications

ID	Name and description *Click for details*	Created	weight *details*	shmem items	HR type *details*	Adaptive replication *details*	homogeneous app version? *details*	deprecated?	Non-CPU-intensive?	Beta?	Exact fraction done?	
1	example_app Example Application	2 Oct 2017	1	100	0	0	☐	☐	☐	☐	☐	Update

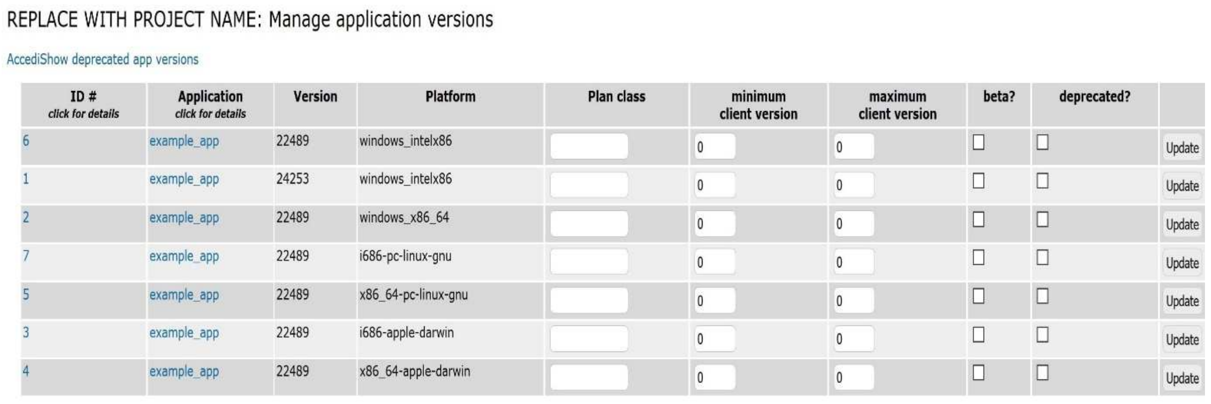
Add an application

To add an application enter the short name and description ('user friendly name') below. You can then edit the application when it appears in the table above.

Name	Description	
		Add Application

21) The mod_cgi component of Apache is disabled by default and must be enabled to start the scheduler correctly. The command is sudo a2enmod cgi (will ask for Apache restart).

22) The application ready to run will be inserted in the administration web interface. The next command to queue the job will be . / bin / update_versions (to be used even when it is necessary to update the application). Answer "Y" to all requests, such as that of digitally marking the application, so that the example ones for all platforms (Windows, Linux and Mac 32 and 64 bit) are ready.

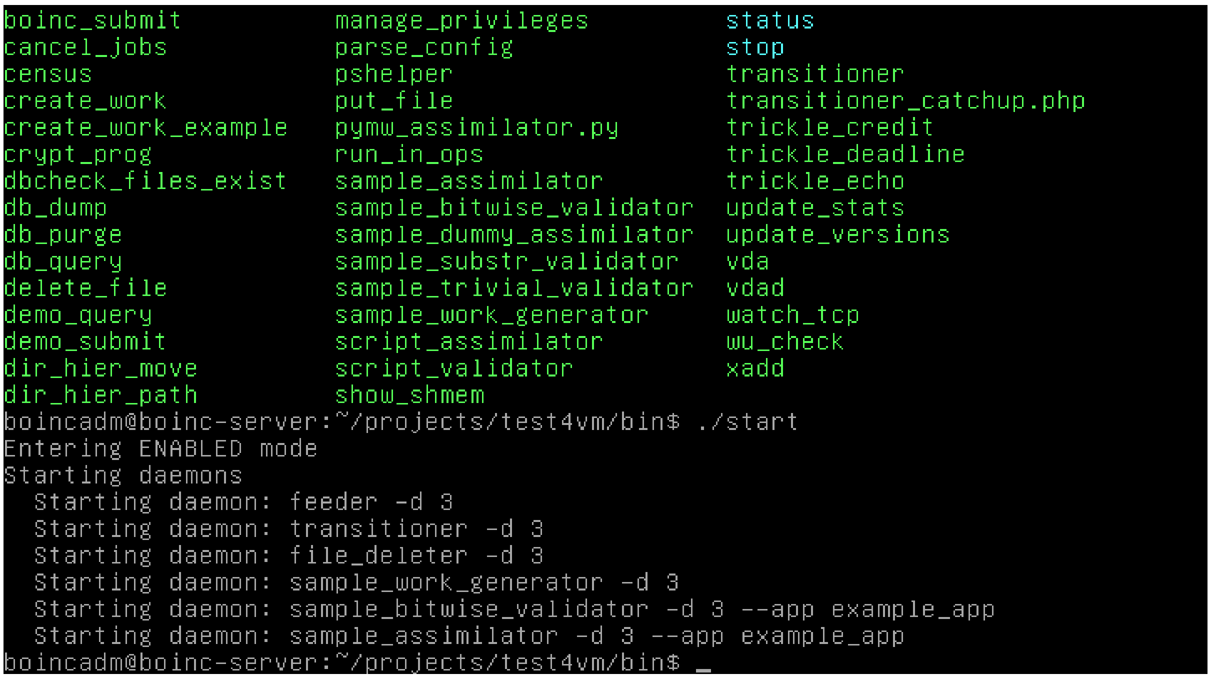


REPLACE WITH PROJECT NAME: Manage application versions

AccediShow deprecated app versions

ID # *click for details*	Application *click for details*	Version	Platform	Plan class	minimum client version	maximum client version	beta?	deprecated?	
6	example_app	22489	windows_intelx86		0	0	☐	☐	Update
1	example_app	24253	windows_intelx86		0	0	☐	☐	Update
2	example_app	22489	windows_x86_64		0	0	☐	☐	Update
7	example_app	22489	i686-pc-linux-gnu		0	0	☐	☐	Update
5	example_app	22489	x86_64-pc-linux-gnu		0	0	☐	☐	Update
3	example_app	22489	i686-apple-darwin		0	0	☐	☐	Update
4	example_app	22489	x86_64-apple-darwin		0	0	☐	☐	Update

23) To start the server daemons, you will need to run the command
. / bin / start



```
boinc_submit          manage_privileges       status
cancel_jobs           parse_config            stop
census                pshelper                transitioner
create_work           put_file                transitioner_catchup.php
create_work_example   pymw_assimilator.py     trickle_credit
crypt_prog            run_in_ops              trickle_deadline
dbcheck_files_exist   sample_assimilator      trickle_echo
db_dump               sample_bitwise_validator update_stats
db_purge              sample_dummy_assimilator update_versions
db_query              sample_substr_validator  vda
delete_file           sample_trivial_validator vdad
demo_query            sample_work_generator    watch_tcp
demo_submit           script_assimilator       wu_check
dir_hier_move         script_validator         xadd
dir_hier_path         show_shmem
boincadm@boinc-server:~/projects/test4vm/bin$ ./start
Entering ENABLED mode
Starting daemons
  Starting daemon: feeder -d 3
  Starting daemon: transitioner -d 3
  Starting daemon: file_deleter -d 3
  Starting daemon: sample_work_generator -d 3
  Starting daemon: sample_bitwise_validator -d 3 --app example_app
  Starting daemon: sample_assimilator -d 3 --app example_app
boincadm@boinc-server:~/projects/test4vm/bin$ _
```

The server with Docker

For some time now it has been possible to create a Boinc server using the Docker virtualization platform with a pre-built image.
To begin, prepare a Linux server machine (Ubuntu Server, in our case): the simplest method is to download the iso from the site of the chosen distribution and create a virtual machine with Virtualbox.

1) Once the installation is complete (leaving only the basic packages), connected the virtual machine to the internet (by configuring the virtual network card in "Bridge" mode), create the user on, necessary for the installations, with the following command:

   sudo passwd

2) Update the system with commands sudo apt-get update is sudo apt-get upgrade and restart.
3) Remove old versions of Docker (if any) with the command
   sudo apt-get remove docker docker-engine docker.io
4) Install (or upgrade) the package apt-transport-https.
5) Install Docker's official GPG key
   Curl -fsSL https://dowload.docker.com/linux/ubuntu/gpg |
   sudo apt-key add -

6) Add the Docker repository

sudo add-apt-repository "deb [arch = amd64] https://download.docker.com/linux/ubuntu $ (lsb_release cs) stable "

7) Refresh the apt list with apt-get update

8) Install the latest Docker Compose (currently 1.24) with the command sudo curl -L https://github.com/docker/compose/releases/download/1.23. 1 / docker-compose - $ (uname -s) - $ (uname -m) -o / usr / local / bin / docker-compose

9) Give the correct permissions to the tracks with sudo chmod + x / usr / local / bin / docker-compose

10) Check that the version is correct with the command docker-compose version

11) Install Docker with the command apt-get install docker-ce

12) Download the git package with the command git clone https: // - github.com/marius311/boinc-server-docker.git

13) Enter the boinc-server-docker folder and run the commands:

docker-compose pull

docker-compose up -d

14) At this point the server will begin to download the necessary components, it will be installed and running and just access it with the configured IP address.

The Boinc configuration wiki:
https://boinc.berkeley.edu/trac/wiki/ProjectMain
The make_project command:
https://boinc.berkeley.edu/trac/wiki/MakeProject
The configuration file of the project
https://boinc.berkeley.edu/trac/wiki/ProjectConfigFile
Extended guide for advanced configurations (eg Boinc + cloud)
http://doc.desktopgrid.hu/doku.php?id=navigation:overview

Thanks to the whole Boinc Italy community e

a BIG thank you to:

Simone Conti* and *Glauco Mancini.