

# BOINC – an approach to grid (distributed) computing

LinuxDays 2007 - marc.seil@tudor.lu

## BOINC

Berkeley Open Infrastructure for Network computing

- started in February 2002
- first release 10<sup>th</sup> April 2002
- the first BOINC-based project, [Predictor@home](#), started on the 9<sup>th</sup> June 2004
- the most known BOINC-based project, [SETI@home](#), started on the 22<sup>th</sup> June 2004
- about 90 developers are involved in the BOINC sources

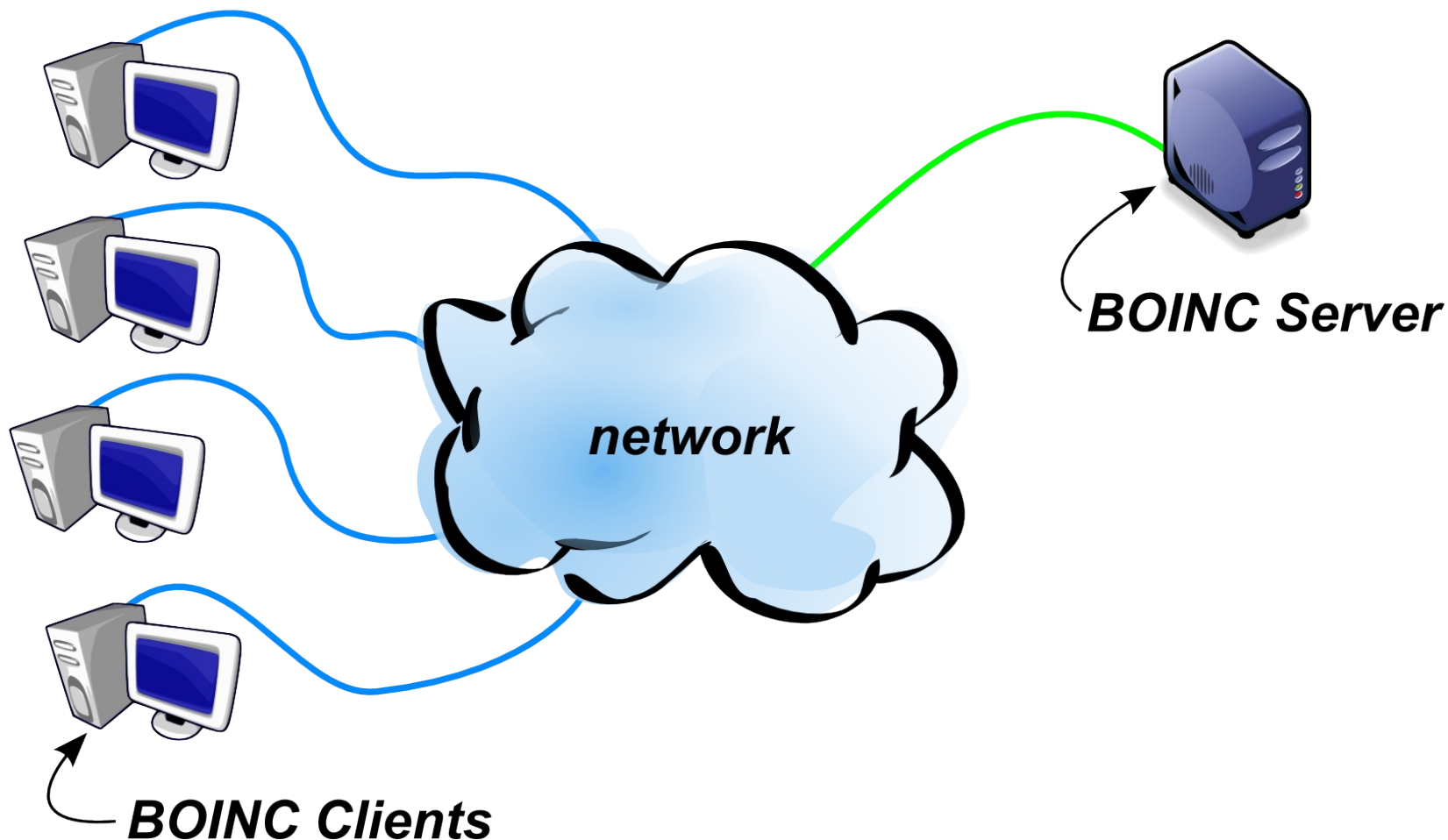
## SETI@home

the goal of this project is to search for extraterrestrial intelligence

- first the signals were processed on dedicated single super computers
- in 1995 David Gedye proposed to use Internet-connected computers for a project dedicated distributed computing environment
- on the 22<sup>th</sup> June 2004 **SETI@home** switched to BOINC
- 1,3 million PCs (about 600.000 users)

ref: SETI@home and Astronomie.de

## *Basic BOINC set up*



## Which type of computing is BOINC?

- volunteer computing ([SETI@home](#) - public resource computing)
- desktop grid computing

## Volunteer Computing (VC)

the computing resources are donated by volunteering users

### advantages

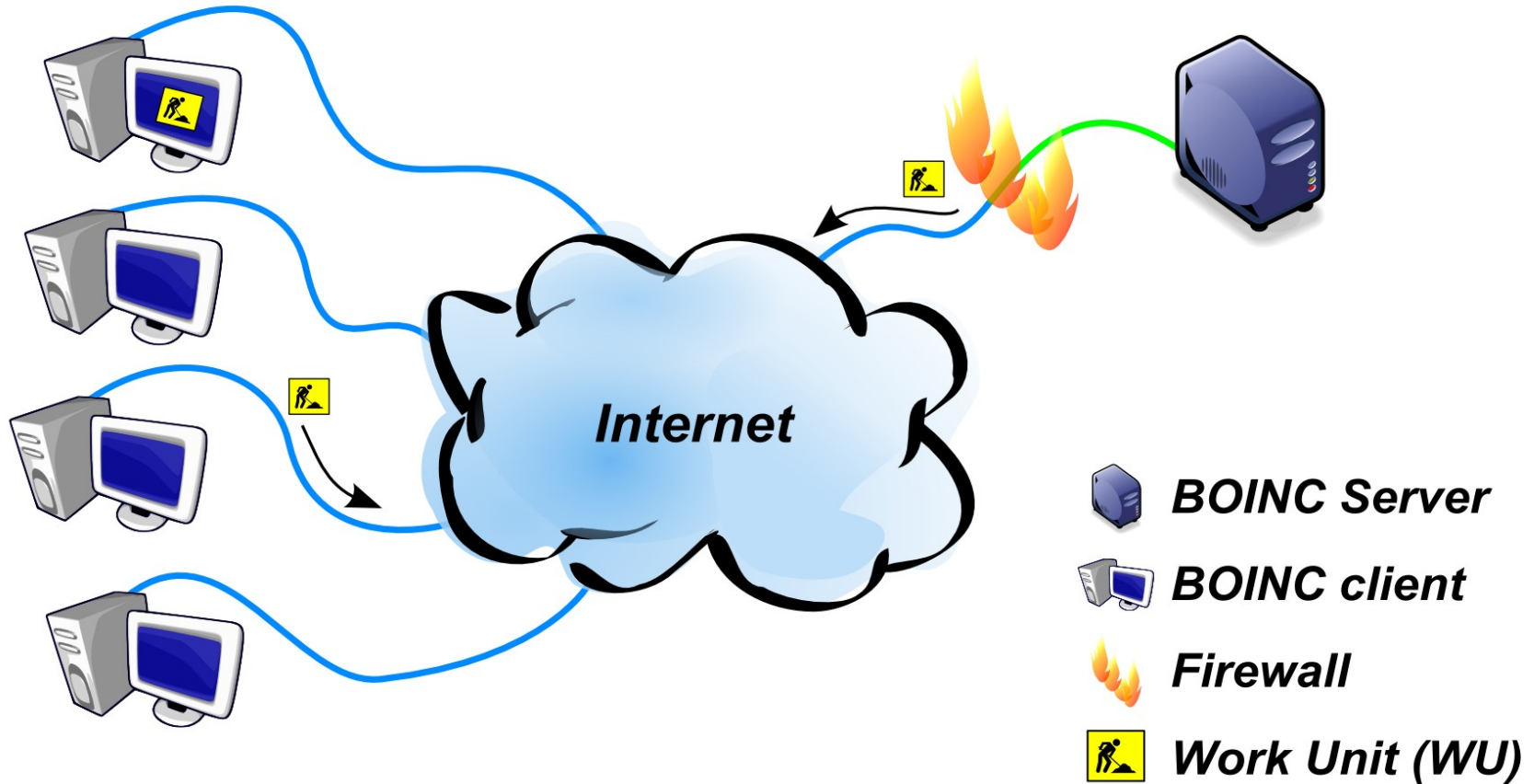
- + computing power
- + cheap
- + public interest

### disadvantages

- security
- administration
- availability

- ref: <http://boinc.berkeley.edu/volunteer.php>

## *Volunteer Computing*



## Desktop grid computing (DG)

the computing resources are based on enterprise IT material (like office or department PCs)

### advantages

- + reliable resources
- + administration (single account)
- + security

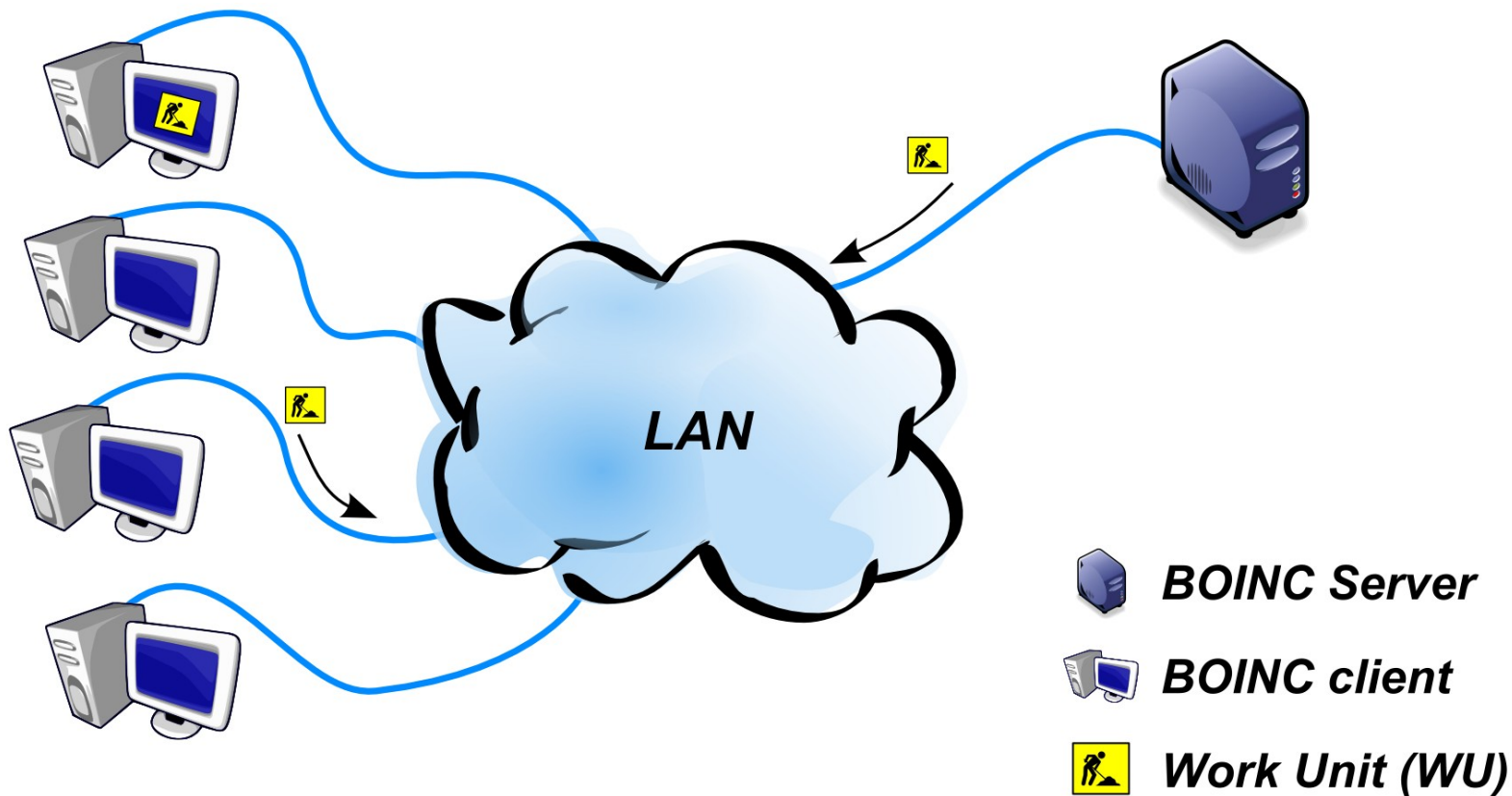
### disadvantages

- must be paid
- available computing power (resources)

ref: <http://boinc.berkeley.edu/dg.php>



## Desktop Grid Computing



## Which is the best OS to set up a BOINC server

It has to be the Linux<sup>®</sup> OS

## Server side software package dependencies

- gnu dev tools (make, autoconf, gcc, automake, ...)
- python
- php
- mysql
- a detailed list is available from the website  
<http://boinc.berkeley.edu/build.php> (client application dependencies are available on  
[http://boinc.berkeley.edu/compile\\_client.php](http://boinc.berkeley.edu/compile_client.php) )

## What is a WorkUnit (WU)?

A WU in BOINC is the task (job) which is send to the BOINC clients in order to be processed.

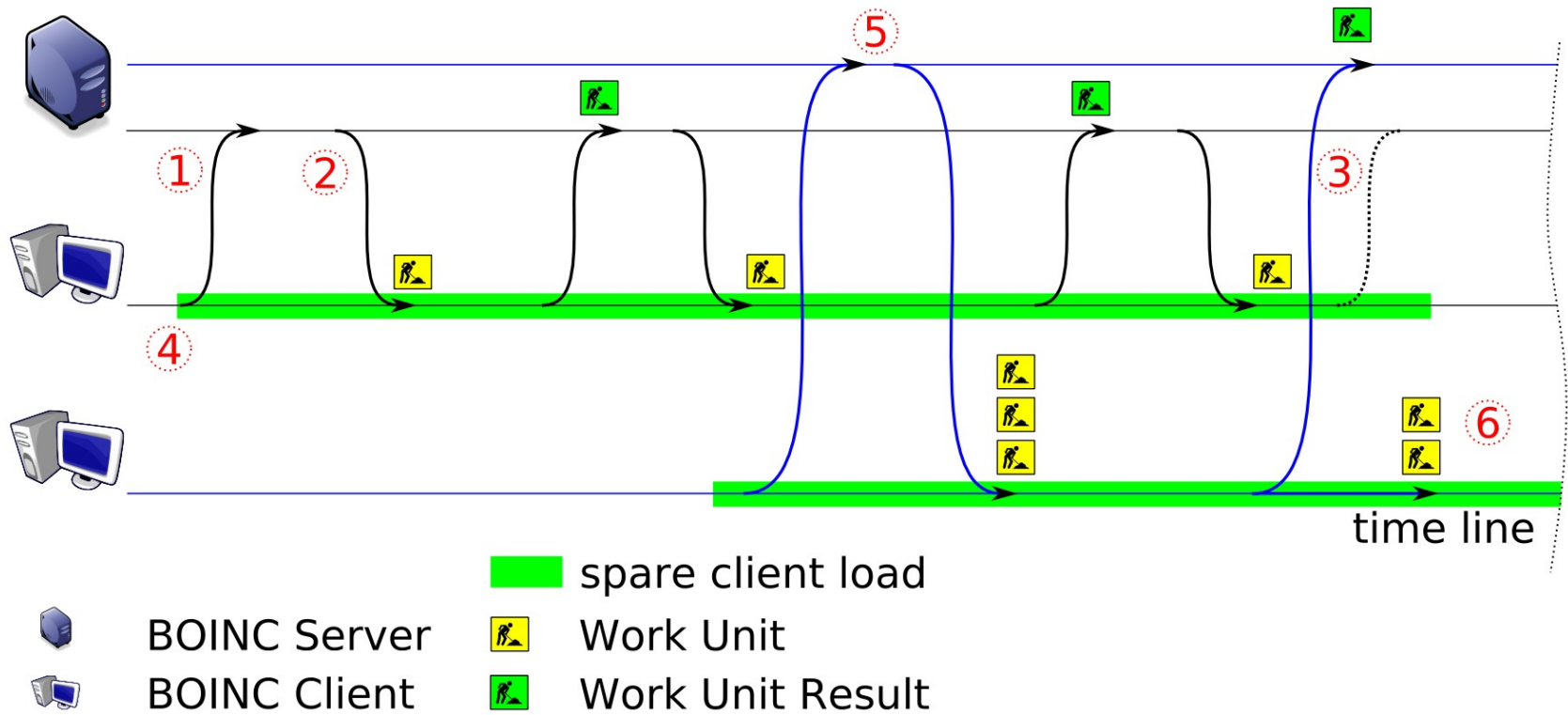
## Consists of the Parts

- application
- dataset

## Description

- work unit template
- result template
- name

## BOINC WORK UNIT flow



## Daemons

- **feeder:** is responsible to fill up the ready to send queue with job instances which have to be processed by the clients.
- **transitioner:** checks for jobs where the state changed (eg. completed instance). Depending on the situation it may create new instances or flag a permanent error.
- **file\_deleter:** removes output and input files which are no longer needed.
- **validator:** compares the result of different instances and defines out of the different instances a canonical instance.
- **assimilator:** handles the jobs that are “completed” and have a canonical instance or which have permanent error.

# Practical Part

based on a kubuntu system



**Some optional support and fallback material is  
available at**

<http://www.linuxdays.lu/downloads/linuxdays2007/boinc/>

**login on the tutorial PCs**

**login: guest**

**pwd: guest**

## Apache2

- install through the adept package manager the tools
  - apache2
  - libapache2-mod-php5
- the apache2 configuration can be found in the directory **/etc/apache2** and the root directory is **/var/www**. Logs can be found in **/var/log/apache2**.
- check if apache2 is running
  - **ps aux | grep apache**
  - with the browser (<http://hostname> or <http://localhost>)
- check if the php module is loaded correctly with a file `info.php` which can be accessed through a browser

```
<?php
    echo "hello world!";
    phpinfo();
?>
```

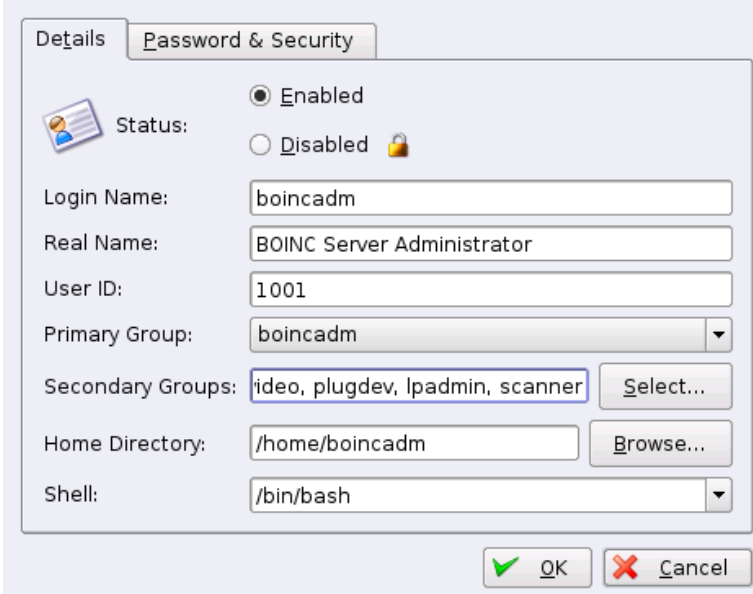


## Dedicated User

create a user account which is dedicated to run the BOINC server side applications.

K-menu->System Settings

- name : boincadm
- pwd: boincadmpwd (not secure)



The screenshot shows the 'Details' tab of a user creation window. The 'Status' is set to 'Enabled'. The 'Login Name' is 'boincadm', 'Real Name' is 'BOINC Server Administrator', 'User ID' is '1001', and 'Primary Group' is 'boincadm'. The 'Secondary Groups' field contains 'ideo, plugdev, lpadmin, scanner'. The 'Home Directory' is '/home/boincadm' and the 'Shell' is '/bin/bash'. There are 'Select...' and 'Browse...' buttons next to the secondary groups and home directory fields respectively. At the bottom are 'OK' and 'Cancel' buttons.

Field	Value
Status	Enabled
Login Name	boincadm
Real Name	BOINC Server Administrator
User ID	1001
Primary Group	boincadm
Secondary Groups	ideo, plugdev, lpadmin, scanner
Home Directory	/home/boincadm
Shell	/bin/bash

## MySQL 1/2

- install through the adept package manager the tools  
**mysql-server-5.0, mysql-client-5.0, php5-mysql, php5-mysqli**
- The configuration can be found in **/etc/mysql**
- check if mysql is running (tcp socket is optional if the mysql server is running on the boinc server)

**ps -aux | grep mysqld**

**netstat -list** (check for mysql or 3306 port)

- now set the mysql root password (the proposed password is not secure)

**mysqladmin -h localhost -u root password *mysqlrootpwd***

- as user guest try to connect to the mysql server as user root

**mysql -h localhost -u root -p** (prompt for password)

**> show databases;**

**> \q;**

## MySQL 2/2

- create a new mysql database user which has the permission to create a database. (as root database user)

- > **GRANT ALL ON \*.\* TO 'boincadm'@'localhost';**

- > **SELECT user,create\_priv,password from mysql.user;**

- define the password of this user to an empty string.  
(this simplifies the set up)

- > **SET PASSWORD FOR 'boincadm'@'localhost' = '';**

## BOINC Sources (**as user boincadm!!**)

- now fetch the version 5\_9\_1 of the boinc core release sources from the cvs repository.

(alternative tar archive from [www.linuxdays.lu](http://www.linuxdays.lu))

```
cvs -d \  
:pserver:anonymous:@alien.ssl.berkeley.edu:/home/cvs/  
cvsroot \ checkout -r boinc_core_release_5_9_1 boinc
```

## BOINC build dependencies 1/2

- identify the compiler version to use

```
strings /usr/lib/libc.a | grep GCC
```

or use

```
objdump -j .comment -s /usr/lib/libc.a
```

## BOINC build dependencies 2/2

download and install the necessary kubuntu packages with the “adept manager” or use “**apt-get install `cat package\_list.txt`**” with the downloaded package list ([www.linuxdays.lu](http://www.linuxdays.lu))

m4	autoconf	automake1.9
gcc-4.1	g++-4.1	libssl-dev
libmysql++-dev	libglut3-dev	glutg3-dev
libglui-dev	libglitz-glx1-dev	libsdl1.2-dev
libcurl3-dev	freeglut3-dev	libsm-dev
libice-dev	libxmu-dev	libxi-dev
libx11-dev	libjpeg62-dev	python-mysqldb
cvs	libtool	vim

[http://boinc.berkeley.edu/compile\\_client.php](http://boinc.berkeley.edu/compile_client.php)

<http://boinc.berkeley.edu/build.php>

## BOINC configure and build

- now run `./_autosetup` and `./configure` to set up the source tree to be compiled on the target kubuntu system
- BOINC can now be compiled with `make` (or `gmake`)
- run `./test/test_sanity.py` script to check if all crucial backend software packages are configured and compiled correctly.

**The BOINC backend tools are now ready**

## Create a BOINC project 1/4

- create a new BOINC project and key directory as **user boincadm**

```
mkdir -p projects/test_seup  
mkdir -p projects/test_setup_keys
```

- fill up the project directory

```
./tools/make_project --user_name boincadm \  
  --drop_db_first --delete_prev_inst \  
  --project_root $HOME/projects/test_setup \  
  --key_dir $HOME/projects/test_setup_keys \  
  --db_user boincadm --db_host localhost \  
  test_setup 'Test Setup: first BOINC project'
```

## Create a BOINC project 2/4

- configure apache following **point 1** of the **make\_project** output and restart apache.  
`/etc/init.d/apache restart`
- adjust the group of the **test\_setup** and **test\_setup\_keys** directory to match the apache group (as root)

```
ps aux | grep apache  
groups www-data  
chgrp -R www-data test_setup  
chgrp -R www-data test_setup_keys
```

- check the project website [http://localhost/test\\_setup](http://localhost/test_setup)



## Create a BOINC project 3/4

### the directory structure

- apps:** application directory
- bin:** applications which are used to manage the BOINC workflows
- cgi-bin:** stores the file\_upload\_handler
- download:** the files which will be send to the clients
- templates:** the template files describing the workunits
- upload:** files which are received from the clients

## Create a BOINC project 4/4

some fine tuning

- locate the `xml.gif` file in the project directory and adjust the path in `html/user/index.php`

```
find ./ -name xml.gif  
vi html/user/index.php
```

- recheck the project webpage

## Create a BOINC user account 1/3

a benefit of BOINC is the scalability and the build in autonomous user (node) management.

- edit the config.xml file to the proper needs. The file can be found in the project root directory

toggle the `disable_account_creation` to 0

- add the phpmailer package to the `test_setup/html/inc/phpmailer` directory. phpmailer can be downloaded from sourceforge or [www.linuxdays.lu](http://www.linuxdays.lu)

## Create a BOINC user account 2/3

- now adjust some settings in the project.inc file, which is located in test\_setup/html/project/ directory

**add below the defines**

...

```
define("EMAIL_FROM","info@yourproject.net");  
define("EMAIL_FROM_NAME","info@yourproject.net");  
$USE_PHPMAILER=true;  
$PHPMAILER_HOST="your.smtp_server.net";  
$PHPMAILER_MAILER="smtp";
```

...

## Create a BOINC user account 3/3

- now create a user account through the web user account creation form.

**!! Remember your password !!**

[http://localhost/test\\_setup/create\\_account\\_form](http://localhost/test_setup/create_account_form)

- after creation a mail will be send. If not simply check the database

```
mysql -u boincadm test_setup
```

```
> SELECT email_addr,name,authenticator FROM user;
```

## Adding an application to the project 1/3

the BOINC core release includes sample applications which can be used to set up a BOINC test project

- **copy the uppercase application from apps to the test\_setup/apps directory following the BOINC naming convention.**

```
cd $HOME/projects/test_setup/apps
```

```
mkdir uppercase
```

```
cd uppercase
```

```
# now copy the sample application from the boinc source tree
```

```
cp $HOME/boinc/apps/upper_case \  
uppercase_5.0_i686-pc-linux-gnu
```

ref <http://boinc.berkeley.edu/platform.php>

## Adding an application to the project 2/3

adjust the project.xml file (or download from [www.linuxdays.lu](http://www.linuxdays.lu))

```
<boinc>
  <platform>
    <name>i686-pc-linux-gnu</name>
    <user_friendly_name>Linux....</user_friendly_name>
  </platform>
  <platform>
    <name>anonymous</name>
    <user_friendly_name>anonymous</user_friendly_name>
  </platform>
  <app>
    <name>uppercase</name>
    <user_friendly_name>The ..</user_friendly_name>
  </app>
</boinc>
```

## Adding an application to the project 2/2

- now add the application to the project (database) and update the latest version with the commands  
`./bin/xadd` and `./bin/update_versions`
- check if the application was added to the project  
[http://localhost/test\\_setup](http://localhost/test_setup)  
**or use the database**  
**`mysql -h localhost test_setup`**  
**> `SELECT * FROM APP;`**  
**> `SELECT * FROM APP_VERSION;`**



## Adding some work to the project 1/4

- create a work unit template file in the templates directory (wu\_uppercase.xml)

```
<file_info>
  <number>0</number>
</file_info>
<workunit>
  <file_ref>
    <file_number>0</file_number>
    <open_name>in</open_name>
  </file_ref>
  <delay_bound>600</delay_bound>
</workunit>
```

available on [www.linuxdays.lu](http://www.linuxdays.lu)

## Adding some work to the project 2/4

- create a result template file in the templates directory (re\_uppercase.xml)

```
<file_info>
  <name><OUTFILE_0/></name>
  <generated_locally/>
  <max_nbytes>2000</max_nbytes>
  <upload_when_present/>
  <url><UPLOAD_URL/></url>
</file_info>
<result>
  <file_ref>
    <file_name><OUTFILE_0/></file_name>
    <open_name>out</open_name>
  </file_ref>
</result>
```

available on [www.linuxdays.lu](http://www.linuxdays.lu)

## Adding some work to the project 3/4

- copy an ascii input file to the download directory (or create a file, adjust **max\_nbytes** if necessary)
- the number of bytes can be checked with the command  
**wc about.txt**
- now add the WU to the project with the project tool  
bin/create\_work in the project root directory  
**./bin/create\_work -appname uppercase \  
-wu\_name wu\_01 \  
-wu\_template templates/wu\_uppercase.xml \  
-result\_template templates/re\_uppercase.xml \  
-min\_quorum 1 -target\_nresults 1 about.txt**

available on [www.linuxdays.lu](http://www.linuxdays.lu)

## Adding some work to the project 4/4

the prior steps can be checked in the database column workunit or they can be checked through the "Project Management" web interface

- Database

- > **SHOW COLUMNS FROM workunit;**

- > **SELECT id,name,xml\_doc FROM workunit;**

- WEB frontend

**[http://localhost/test\\_setup\\_ops](http://localhost/test_setup_ops)**

## Adjusting user and group rights

as some of the directories have to be writeable and readable by the primary group which is running the apache server some adjustments have to be made

- change the group of the complete directory `project/` in respect to the primary group (gid) of the running apache process if these are not already configured

```
ps aux | grep apache # the first column
```

```
id www-data
```

```
chown boincadm:www-data -R projects/*
```

- the “Set Group ID” of the binary `./bin/feeder` has to be set to prevent shared memory access errors

```
chmod g+s ./bin/feeder (as user root)
```

## Adjusting the project configuration

- Check the configuration file config.xml if all daemons are present to start the BOINC server side processes (feeder, file\_deleter, transitioner)

**<daemons>**

**<daemon><cmd>feeder -d 3</cmd></daemon>**

**<daemon><cmd>transitioner -d 3</cmd></daemon>**

**<daemon><cmd>file\_deleter -d 3</cmd></daemon>**

**</daemons>**

## Starting the BOINC server processes

- The best method to start and stop the BOINC processes are `./bin/start` and `./bin/stop`
- The log files used by the daemons can be found in the **`log_`*hostname*** directory
- a good monitoring at the beginning is  
`tail -f log_*/*`

## Attaching a client to the BOINC project

- Download the BOINC client application from the project website or from [www.linuxdays.lu](http://www.linuxdays.lu)
- Unpack the client in a terminal with `sh boinc_*5.4.11*.sh` and a new directory **BOINC/** will be created with the client application
- Attach to the `test_setup` project with the prior created user account (`./run_client` or the GUI version `./run_manager`)



## Securing the management web frontend

the authentication facility of apache2 can be used

- create a password file in the html/ops directory  
`htpasswd2 -c password boincadm`
- now add a `.htaccess` file in the same directory with the following contents
  - AuthType Basic**
  - AuthName "By Invitation Only"**
  - AuthUserFile**  
`/home/boincadm/projects/test_setup/html/ops/password`
  - Require user boincadm**

## Work and Debugging

update the config.xml file with some daemon applications

**./bin/stop**

```
<daemons>
```

```
  <daemon><cmd>
```

```
    make_work -wu_name name_of_existing_wu -cusion 5
```

```
  </cmd></daemon>
```

```
  <daemon><cmd>
```

```
    sample_*_validator -d 3 -app uppercase
```

```
  </cmd></daemon>
```

```
  <daemon><cmd>
```

```
    sample_dummy_assimilator -d 3 -app uppercase
```

```
  <cmd></daemon>
```

```
</daemons>
```

**./bin/start**

## Official

<http://boinc.berkeley.edu> and <http://boinc.berkeley.edu/boinc.pdf>

## Unofficial Wiki

<http://boinc-wiki.ath.cx>

## Message Board

<http://boinc.berkeley.edu/dev>

## Bundled documents and papers

**doc/** (source code bundled documentation) and the papers at  
<http://boinc.berkeley.edu/papers.php>