QE Developer Meeting 2015

Filippo Spiga
<fs395@cam.ac.uk>
Three principles ( + one note)

• Simple is good

• Easy is good

• A small effort today equal avoiding headaches (and wasting time) tomorrow

"I am not a software engineer" is a lame excuse
Current model

- SVN came after CVS (fear of change, fear of loosing data)
- Central repository hosted by external server
- Several inactive branches
- 62 registered developers
GIT true/false facts

• Moving to GIT might lose all previous commit history. FALSE.
• Converting repository to GIT is a complex procedure. FALSE.
• Converting to GIT will lose authorship records. FALSE
• GIT is more difficult to use. FALSE
• GIT is different to use. TRUE.
• GIT is faster (commit/checkout). TRUE.
• GIT is better. IT DEPENDS!
Converting SVN to GIT

```

git svn clone -s http://qeforge.qe-forge.org/svn/q-e/trunk/espresso

git svn clone http://qeforge.qe-forge.org/svn/q-e/trunk/espresso
    --no-metadata

git-svn show-ignore > .gitignore

cd espresso

git remote add origin git@github.com:QEF/q-e.git

git push -u origin master

git svn rebase
```
QEF/q-e.git

https://github.com/QEF
# Updating GIT mirror

#!/bin/bash

cd /home/fs395/QE-REPO/espresso
git pull

git push -u mirror master

+ crontab
Alternative model

LEGACY WORLD

q-e (SVN)

OFFICIAL MIRROR

QEF/q-e (GIT)

PRIVATE DEVELOPMENT

fspiga/q-e (GIT)

DIFF

ac456e = r11000
67ea31 = r11001
12bb45 = r11002
77cc66 = r11003
111ccc = r11004

12bb45 = r11002
77cc66 = r11003
1234aa
565653
abcc11
Testing

• Testing is essential in a SW life-cycle: do it right or let it go

• Proper extensive testing is hard and tedious

• There are complicated framework that *automates* testing but nothing that I am aware of that is FORTRAN specific

• What kind of testing do we need? What granularity?

  I suggest **BUILDBOT** (automated build)
  and **TESTCODE** (numerical regression)
BUILDBOT – Automating building

How many times you try to compile on *different architectures* using *different compilers* and *different configure flags* before spot something wrong?

**BUILDBOT:**
- Written in python
- master/slave approach (Master -> Builders -> Slave)
- Language agnostic
- Easy to install (pip), no heavy dependencies
- Zero native interaction with HPC batch schedulers
- Troubles to set proper environment

**EXAMPLE:** http://www.cmth.ph.ic.ac.uk/buildbot/onetep/waterfall
Experimental setup

URL: [http://xiexie.syslab.disco.unimib.it:8010](http://xiexie.syslab.disco.unimib.it:8010)

Slaves:
- 4 @ HPCS login nodes (Univ. Cambridge): various compilers (Intel, PGI), various MPI libraries (IMPI, OMPI, MV2)
- 2 @ XIEXIE (Univ. Milano-Bicocca): GCC with no external libraries, testing enabled using old method
- 1 @ TESLA (Univ. Milano-Bicocca): QE-GPU using different CUDA versions

Scheduling:
- HPCS, every day nightly @ 2 AM
- XIEXIE, every 4 hours
- TESLA, every 3 hours during day time (7-21)
TESTCODE – numerical regression

How many times you compared *manually* two outputs to understand if there is numerical divergence and you had to repeat this as many times as input cases under examination?

TESTCODE:

- Written in python by James Spencer (ICL)
- one single python script, no dependency
- compare a output against a reference pre-computed one
- compare numerical values tagged in the output appearing in the order as they are printed
- you can compare a value against an *interval* or a *tolerance*

EXAMPLE: CASTEP and ONETEP test-suite
How it looks like

iteration # 4  
ecut = 25.00 Ry  
beta = 0.70
Davidson diagonalization with overlap
ethr = 4.29E-04,  
avg # of iterations = 3.0

negative rho (up, down): 3.058E+00 0.000E+00

total cpu time spent up to now is 111.1 secs

total energy  = -11426.85486964 Ry
Harris-Foulkes estimate  = -11427.19599350 Ry
estimated scf accuracy  <  1.08992520 Ry

<QC> total energy  = -11426.85486964 Ry
<QC> Harris-Foulkes estimate  = -11427.19599350 Ry
<QC> estimated scf accuracy  =  1.08992520 Ry
New QE lifecycle

LEGACY WORLD

q-e (SVN)

QEF/q-e (GIT)

Automated Report

BUILDBOT

SLAVE1

SLAVE1

SLAVE1

SLAVE1

Maintainer
Discussion points

• Where can we host a main BUILDBOT master?
• Who can host a slave?
• What a complete "test-suite" should contain?
• Who can define input cases and correctness criteria?
• (last but not least) What is the "sustainability model"?
QE-GPU – status

• I have a dream and I have a job
  – time to experiment, no time to consolidate
  – QE-GPU is an excuse to learn

• Support of the plug-in model
  – the average user is not enough Unix skilled to follow more than two steps (I am not sorry to say so!)

• Competition is aggressive but I do not care to compete
  – QE-GPU is an excuse to learn

• Failure in the engagement
  – People want to use the code but nobody else WRITE THE CODE!
The truth is...

- most of unknown and hidden QE-GPU users have WORKSTATION or small clusters (O(100) nodes max)
- they rarely experience scaling issues

Focus on ...

- PWscf and CP, GCC + CUDA
- alternative to MAGMA, embedded GPU eigen-solver based on qe_pzpotrf/qe_pdsyevd
- as many working functionalities as possible in SERIAL first

Versioning...

- GPU-accelerated 5.1.1 (or 5.2?) ready and public* by GPU technology 2015
- not supported or maintained _all_ previous releases
Future?

(soon rather than later)

WE WANT YOU!