



HYDRA: TOP-NOTCH CONTINUOUS INTEGRATION FOR DEMANDING PEOPLE

Ludovic Courtès

Service d'expérimentation & développement
Inria Bordeaux Sud-Ouest

3 November 2011

introduction

features

practice

closing

scenario 1: a peaceful development team...



scenario 1: ... software that works ...



scenario 1: ... and suddenly ...



scenario 1: **who** broke it?!



scenario 2: “works for me”

- ▶ *user*: It's broken.

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me.

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4.

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4. On x86_64.

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4. On x86_64. With Apple GCC 4.0.1, GMP 4.2.4, MPFR 2.4.2, BDB 4.2.

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4. On x86_64. With Apple GCC 4.0.1, GMP 4.2.4, MPFR 2.4.2, BDB 4.2. Et avec `--disable-color`.

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4. On x86_64. With Apple GCC 4.0.1, GMP 4.2.4, MPFR 2.4.2, BDB 4.2. Et avec `--disable-color`.
- ▶ *dev*: Well, too bad. I rather test on GNU/Linux, i686

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4. On x86_64. With Apple GCC 4.0.1, GMP 4.2.4, MPFR 2.4.2, BDB 4.2. Et avec `--disable-color`.
- ▶ *dev*: Well, too bad. I rather test on GNU/Linux, i686, with GCC 4.6.2, GMP 4.2.1, MPFR 2.4.1, GDBM 1.8.2, GTK+ 2.12. Oh, and with `--enable-color --enable-gui`.

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4. On x86_64. With **Apple GCC 4.0.1**, GMP 4.2.4, MPFR 2.4.2, BDB 4.2. Et avec `--disable-color`.
- ▶ *dev*: Well, too bad. I rather test on GNU/Linux, i686, with **GCC 4.6.2**, GMP 4.2.1, MPFR 2.4.1, GDBM 1.8.2, GTK+ 2.12. Oh, and with `--enable-color --enable-gui`.

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4. On x86_64. With Apple GCC 4.0.1, **GMP 4.2.4**, MPFR 2.4.2, BDB 4.2. Et avec `--disable-color`.
- ▶ *dev*: Well, too bad. I rather test on GNU/Linux, i686, with GCC 4.6.2, **GMP 4.2.1**, MPFR 2.4.1, GDBM 1.8.2, GTK+ 2.12. Oh, and with `--enable-color --enable-gui`.

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4. On x86_64. With Apple GCC 4.0.1, GMP 4.2.4, MPFR 2.4.2, **BDB 4.2**. Et avec `--disable-color`.
- ▶ *dev*: Well, too bad. I rather test on GNU/Linux, i686, with GCC 4.6.2, GMP 4.2.1, MPFR 2.4.1, **GDBM 1.8.2**, GTK+ 2.12. Oh, and with `--enable-color --enable-gui`.

scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4. On x86_64. With Apple GCC 4.0.1, GMP 4.2.4, MPFR 2.4.2, BDB 4.2. Et avec `--disable-color`.
- ▶ *dev*: Well, too bad. I rather test on GNU/Linux, i686, with GCC 4.6.2, GMP 4.2.1, MPFR 2.4.1, GDBM 1.8.2, **GTK+ 2.12**. Oh, and with `--enable-color --enable-gui`.




scenario 2: “works for me”

- ▶ *user*: It's broken.
- ▶ *dev*: Weird, it works for me. Which platform?
- ▶ *user*: Darwin 10.4. On x86_64. With Apple GCC 4.0.1, GMP 4.2.4, MPFR 2.4.2, BDB 4.2. Et avec `--disable-color`.
- ▶ *dev*: Well, too bad. I rather test on GNU/Linux, i686, with GCC 4.6.2, GMP 4.2.1, MPFR 2.4.1, GDBM 1.8.2, GTK+ 2.12. Oh, and with `--enable-color --enable-gui`.




why continuous integration is so great

1. continuous compilation/test 




why continuous integration is so great

1. continuous compilation/test 
2. quick feedback  







why continuous integration is so great

1. continuous compilation/test 
2. quick feedback  
3. **pristine build environment**

why continuous integration is so great

1. continuous compilation/test 
2. quick feedback  
3. pristine build environment
4. **different configurations under test**

why continuous integration is so great

1. continuous compilation/test 
2. quick feedback  
3. pristine build environment
4. different configurations under test
5. **build results published**   

the true cost of doing it yourself

```
while true
do
  rm -rf foo
  svn co svn+ssh://scm.gforge.inria.fr/svn/foo/trunk foo
  cd foo
  make

  sleep 300
done
```

the true cost of doing it yourself

```
while true
do
  rm -rf foo
  svn co svn+ssh://scm.gforge.inria.fr/svn/foo/trunk foo
  cd foo
  make 2>&1 > build-log || \
    mail michel.cosnard@inria.fr -s "Build failed!" \
      < build-log
  sleep 300
done
```

the true cost of doing it yourself

```
function svn_revision {
  ( cd "$1"
    svn info | \
      grep '\~R.vision *: *\([0-9]\+\)$' | \
      sed -e's/[0-9]/g' )
}

function svn_update {
  ( prev_rev="$(svn_revision $1)"
    ( cd "$1" ; svn update )
    new_rev="$(svn_revision $1)"
    test $prev_rev -ne $new_rev )
}

function update_html_report {
  local stamp="$(date -u +%Y%m%d-%H%M)"
  local test_log="$builddir/testsuite/log"
  local failure_rate="(tests not run)"

  gzip --best < "$2" > "build-$stamp.log.gz"
  scp "build-$stamp.log.gz" "$ssh_remote_dir/"

  if [ -f "$test_log" ]
  then
    failure_rate="$(test_failure_rate $test_log)" $html
    gzip --best
    scp "test-$stamp.log.gz" "$ssh_remote_dir/"
  fi

  local result bgcolor
  if [ $1 -ne 0 ]
  then
    result="<b>FAIL</b>"
    bgcolor="lightred"
  else
    result="PASS"
    bgcolor="lightgreen"
  fi

  local rev="$(svn_revision $srcdir)"
  local html="<tr bgcolor=\"$bgcolor\"
  <td><a href=\"$gforge_revision_url foo $rev\">$rev
  <td>$stamp</td>
  <td>$failure_rate</td>
  <td>
    $result
    [ <a href=build-$stamp.log.gz>build</a> |
      <a href=test-$stamp.log.gz>test</a> ]
  </td>
  </tr>"

  ed "$html_report" <<EOF
/INSERT BUILD RESULTS
+1
i
.
wq
EOF

  scp "$html_report" "$ssh_remote_dir/index.html"
}

result="<b>FAIL</b>"
```


lots of tools!

- ▶ BuildBot
- ▶ Jenkins (formerly Hudson)
- ▶ CruiseControl
- ▶ ...
- ▶ Hydra

suitability of a continuous integration tool

- ▶ languages & build systems?
- ▶ build notification & visualization?
- ▶ portability?
- ▶ reproducibility?
- ▶ scalability?

introduction

features

practice

closing

testimonial

“Thanks to **Hydra**, my software projects shine more than ever!”

— Ludovic C.

Hydra, Nix, & you

<http://nixos.org/hydra/>

- ▶ **Nix**
 - ▶ build tool
 - ▶ focus on **controlling the build environment**
- ▶ **Hydra**
 - ▶ **continuous integration tool based on Nix**

Hydra, Nix, & you

<http://nixos.org/hydra/>

- ▶ **Nix**
 - ▶ build tool
 - ▶ focus on **controlling the build environment**
- ▶ **Hydra**
 - ▶ **continuous integration tool based on Nix**
 - ▶ web interface + email notification
 - ▶ GNU/Linux, FreeBSD, MacOS X, Cygwin
 - ▶ distributed builds

Hydra, Nix, & you

<http://nixos.org/hydra/>

- ▶ **Nix**
 - ▶ build tool
 - ▶ focus on **controlling the build environment**
- ▶ **Hydra**
 - ▶ **continuous integration tool based on Nix**
 - ▶ web interface + email notification
 - ▶ GNU/Linux, FreeBSD, MacOS X, Cygwin
 - ▶ distributed builds
 - ▶ **SSH/root access to the build host not needed**

Hydra, Nix, & you

<http://nixos.org/hydra/>

- ▶ **Nix**
 - ▶ build tool
 - ▶ focus on **controlling the build environment**
- ▶ **Hydra**
 - ▶ **continuous integration tool based on Nix**
 - ▶ web interface + email notification
 - ▶ GNU/Linux, FreeBSD, MacOS X, Cygwin
 - ▶ distributed builds
 - ▶ SSH/root access to the build host **not needed**
 - ▶ **manage the build environment by yourself!**



Project

- Overview
- Releases
- All builds
- Job status
- Errors

Project runtime

Project Settings Views Channels

Jobsets

This project has the following jobsets:

◆	Id	Description	Last evaluated	Success	Options	◆
✓	hwloc-trunk	hwloc, `trunk`	2011-11-02 15:08:54	100% 11/11	[Evaluate Hide]	
✗	marcel-trunk	N:M multithreaded library	2011-11-02 15:08:55	97% 37/38	[Evaluate Hide]	
✗	open-mx-master	Open-MX, `master` branch	2011-11-02 15:08:58	69% 30/43	[Evaluate Hide]	
✗	starpv-trunk	StarPU	2011-11-02 15:09:00	40% 2/5	[Evaluate Hide]	
✓	tbx-trunk	portable toolkit	2011-11-02 15:09:02	100% 7/7	[Evaluate Hide]	

[Create a new jobset]



Project

- > Overview
- > Releases
- > All builds
- > Job status
- > Errors

Jobset

- + Overview
- > All builds
- > Job status
- > Errors

Jobset runtime:hwloc-trunk

Jobset Jobs (7) Setup Channels

Most recent builds

◆ #	◆ Job	◆ Release Name	◆ System	◆ Timestamp
✓ 10744	runtime:hwloc-trunk:build_opensuse110_x86_64	hwloc-1.4a1-1	x86_64-linux	2011-10-30 05:10:37
✓ 10738	runtime:hwloc-trunk:build_fedora13_x86_64	hwloc-1.4a1-1	x86_64-linux	2011-10-30 05:05:54
✓ 10741	runtime:hwloc-trunk:build_fedora2_i386	hwloc-1.4a1-1	x86_64-linux	2011-10-30 05:04:07
✓ 10737	runtime:hwloc-trunk:coverage	hwloc-1.4a1-1	x86_64-linux	2011-10-30 02:11:19
✓ 10735	runtime:hwloc-trunk:build	hwloc-1.4a1-1	x86_64-linux	2011-10-30 02:10:23

Status

Job	i686-linux	x86_64-darwin	x86_64-linux
build	✓	✓	✓
build_fedora13_x86_64			
build_fedora2_i386			
build_opensuse110_x86_64			
coverage	✓	✓	✓
debian50_i386			



Project

- > Overview
- > Releases
- > All builds
- > Job status
- > Errors

Jobset

- > Overview
- > All builds
- > Job status
- > Errors

Job

- > Overview
- > All builds
- > Job status
- > Errors

Job runtime:hwloc-trunk:build build 10735

Summary

Information

Build Inputs

Build Steps

History chart



Build ID:	10735
Release name:	hwloc-1.4a1-1
Status:	✓ Success
System:	x86_64-linux
Duration:	2m 55s finished at 2011-10-30 02:10:23
Logfile:	Available (raw, tail)

Add to release:

✓ Apply

[\[Clone this build\]](#)

Build products



One-click install of Nix package `hwloc-1.4a1-1` [[help](#), [contents](#)]



Nix closure of path `/nix/store/w56kkwmhk20bw7zw4z54269gli9zq4zk-hwloc-1.4a1-1` [[help](#)]

Changes



Project

- > Overview
- > Releases
- > All builds
- > Job status
- > Errors

Jobset

- > Overview
- > All builds
- > Job status
- > Errors

Job

- > Overview
- > All builds
- > Job status
- > Errors

Job runtime:hwloc-trunk:build build 10735

Summary Information **Build Inputs** Build Steps History chart

Build inputs

Name	Type	Value	Revision	Store path
tarball	Build output	runtime:hwloc-trunk:tarball build 10717		/nix/store/41mc8cw21bxx6x373p0dh-hwloc-tarball-0pre3949
system	String value	"x86_64-linux"		
nixpkgs	Subversion export	https://svn.nixos.org/repos/nix/nixpkgs/trunk/	30123	/nix/store/fdi9x9ixlnvxlhr9njcx-nixpkgs-r30123
hydraConfig	Git checkout	git://scm.gforge.inria.fr/hydra-recipes/hydra-recipes.git	b7aef2f1fd5a8e5dfa38cf59e9f4a8239faf6c93	/nix/store/by1g3dlyh3kmdj5kff03s-git-export

Changes since previous build : runtime:hwloc-trunk:build build 10621

Input	Changes
tarball	10735 to 10735

build input changes

- > Overview
- > Releases
- > All builds
- > Job status
- > Errors

Jobset

- > Overview
- > All builds
- > Job status
- > Errors

Job

- > Overview
- > All builds
- > Job status
- > Errors

Summary Information **Build Inputs** Build Steps History chart

Build inputs

Name	Type	Value	Revision	Store path
tarball	Build output	runtime:hwloc-trunk:tarball build 10717		/nix/store/4lmc8cw2lbxx6x373p0dh/hwloc-tarball-0pre3949
system	String value	"x86_64-linux"		
nixpkgs	Subversion export	https://svn.nixos.org/repos/nix/nixpkgs/trunk/	30123	/nix/store/fdi9x9ixlnvxlhr9njcnixpkgs-r30123
hydraConfig	Git checkout	git://scm.gforge.inria.fr/hydra-recipes/hydra-recipes.git	b7aef2f1fd5a8e5dfa38cf59e9f4a8239faf6c93	/nix/store/bylg3d1yh3kmdj5kff03s-git-export

Changes since previous build : runtime:hwloc-trunk:build build 10621

Input	Changes
tarball	10735 to 10735
	nixpkgs 29979 to 30112
	hydraConfig 59feld to b7aef2
nixpkgs	30077 to 30123
hydraConfig	59feld to b7aef2

Hydra 0.0svn (using nix-1.0pre29200). Page generated on 2011-11-02 14:56:45. You are logged in as root.

view of a successful job set



Inria
INVENTORS FOR THE DIGITAL WORLD

[> Projects](#) [> Queue \(2/2\)](#) [> All builds](#) [> Job status](#) [> Errors](#) [> Admin](#) [> Sign out](#)

[> runtime](#)

Project

- [> Overview](#)
- [> Releases](#)
- [> All builds](#)
- [> Job status](#)
- [> Errors](#)

View runtime:hwloc result 10717 (hwloc-1.4a1-1)

Finished building on 2011-10-30 02:04:47.

Tarball and Manual



Manual



Manual



Source distribution hwloc-1.4a1-1.tar.bz2 [details, contents]



Source distribution hwloc-1.4a1-1.tar.gz [details, contents]

NixOS GNU/Linux Packages



One-click install of Nix package hwloc-1.4a1-1 [help, contents]



Nix closure of path /nix/store/a7mkvkgbg08bszf179lxzdhc1dmr4i3i-hwloc-1.4a1-1 [help]

Code Coverage Report



Code coverage analysis report



host:9999/view/runtime/hwloc/10717

15:07 (100, 13)

a failed build



Inria
INVENTORS FOR THE DIGITAL WORLD

> Projects > Queue (6/8) > All builds > Job status > Errors > Admin > Sign out

> runtime > hwloc-trunk > build

Project

- > Overview
- > Releases
- > All builds
- > Job status
- > Errors

Jobset

- > Overview
- > All builds
- > Job status
- > Errors

Job

- > Overview
- > All builds
- > Job status
- > Errors

Job runtime:hwloc-trunk:build build 9701

Summary

Information

Build Inputs

Build Steps

History chart



Build ID: 9701

Nix name: hwloc-1.4a1-1

Status: **✗ Build failed (with result)**

System: x86_64-darwin

Duration: 33S finished at 2011-10-13 09:23:29

Logfile: [Available \(raw, tail\)](#)

Add to release:

Apply

[\[Clone this build\]](#)

Build products



Failed build produced output. [Click here to inspect the output.](#) [\[help\]](#)

Changes

Last successful build
2011-10-11 16:50:02

First broken build 2011-10-13
09:22:45

This build 2011-10-13 09:23:29
[\[Log diff \]](#)

host:9999/build/9701

15:02 (100, 1)

Inria

build input changes that led to a failure

- > Overview
- > All builds
- > job status
- > Errors

Build products



Failed build produced output. Click here to inspect the output. [help]

Changes

Last successful build 2011-10-11 16:50:02	First broken build 2011-10-13 09:22:45 [log diff]	This build 2011-10-13 09:23:29 [log diff]																								
runtime:hwloc-trunk:build build 9549	runtime:hwloc-trunk:build build 9690	runtime:hwloc-trunk:build build 9701																								
	<table border="1"><thead><tr><th>Input</th><th>Changes</th></tr></thead><tbody><tr><td>tarball</td><td>9701 to 9701</td></tr><tr><td>nixpkgs</td><td>29703 to 29801</td></tr><tr><td>hwlocSrc</td><td>3924 to 3932</td></tr><tr><td>hydraConfig</td><td>985352 to 0a092a</td></tr><tr><td>nixpkgs</td><td>29760 to 29801</td></tr></tbody></table>	Input	Changes	tarball	9701 to 9701	nixpkgs	29703 to 29801	hwlocSrc	3924 to 3932	hydraConfig	985352 to 0a092a	nixpkgs	29760 to 29801	<table border="1"><thead><tr><th>Input</th><th>Changes</th></tr></thead><tbody><tr><td>tarball</td><td>9701 to 9701</td></tr><tr><td>nixpkgs</td><td>29703 to 29801</td></tr><tr><td>hwlocSrc</td><td>3924 to 3932</td></tr><tr><td>hydraConfig</td><td>985352 to 0a092a</td></tr><tr><td>nixpkgs</td><td>29760 to 29801</td></tr></tbody></table>	Input	Changes	tarball	9701 to 9701	nixpkgs	29703 to 29801	hwlocSrc	3924 to 3932	hydraConfig	985352 to 0a092a	nixpkgs	29760 to 29801
Input	Changes																									
tarball	9701 to 9701																									
nixpkgs	29703 to 29801																									
hwlocSrc	3924 to 3932																									
hydraConfig	985352 to 0a092a																									
nixpkgs	29760 to 29801																									
Input	Changes																									
tarball	9701 to 9701																									
nixpkgs	29703 to 29801																									
hwlocSrc	3924 to 3932																									
hydraConfig	985352 to 0a092a																									
nixpkgs	29760 to 29801																									

Log

```
config.status: creating doc/doxygen-config.cfg
config.status: creating utils/Makefile
config.status: creating hwloc.pc
config.status: creating tests/Makefile
config.status: creating tests/linux/Makefile
config.status: creating tests/linux/gather/Makefile
config.status: creating tests/xml/Makefile
config.status: creating tests/ports/Makefile
config.status: creating tests/rename/Makefile
```

host:9999/build/9701

15:03 (100, 37)

build environments vs. reproducibility

- ▶ **versions** of the dependencies
- ▶ **compiler**
- ▶ **compilation options**, and those of dependencies
- ▶ **miscellaneous** (locale, timezone, etc.)
- ▶ **paths**

build environments vs. reproducibility

- ▶ **versions** of the dependencies
- ▶ **compiler**
- ▶ **compilation options**, and those of dependencies
- ▶ **miscellaneous** (locale, timezone, etc.)
- ▶ **paths**

`-I/path/to/headers`

`$CPATH`

`-L/path/to/lib`

`$LIBRARY_PATH`

build environments vs. reproducibility

- ▶ **versions** of the dependencies
- ▶ **compiler**
- ▶ **compilation options**, and those of dependencies
- ▶ **miscellaneous** (locale, timezone, etc.)
- ▶ **paths**

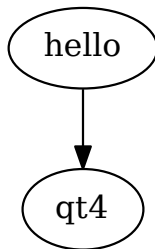
<code>-I/path/to/headers</code>	<code>\$CPATH</code>
<code>-L/path/to/lib</code>	<code>\$LIBRARY_PATH</code>
<code>\$LD_LIBRARY_PATH</code>	
<code>RPATH</code>	<code>RUNPATH</code>

build environments vs. reproducibility

- ▶ **versions** of the dependencies
- ▶ **compiler**
- ▶ **compilation options**, and those of dependencies
- ▶ **miscellaneous** (locale, timezone, etc.)
- ▶ **paths**

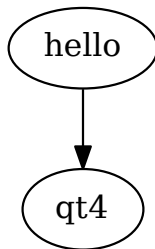
-I/path/to/headers	\$CPATH
-L/path/to/lib	\$LIBRARY_PATH
\$LD_LIBRARY_PATH	
	RPATH
	RUNPATH
\$PYTHONPATH	\$CLASSPATH
\$XML_CATALOG_FILES	\$PERL5LIB
	\$GUILE_LOAD_PATH

build environment control without Nix



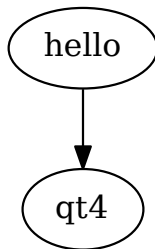
which compiler? which libc?

build environment control without Nix



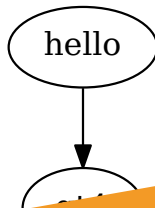
which compiler? which libc?
and now your favorite admin upgrades the system...

build environment control without Nix



which compiler? which libc?
and now your favorite admin upgrades the system...
Segmentation fault

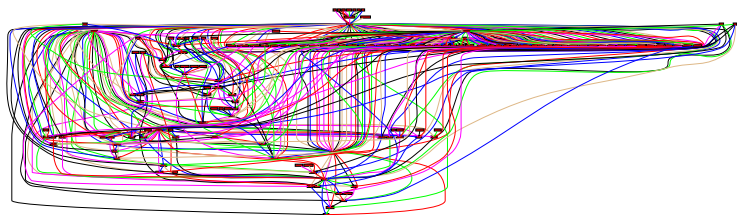
build environment control without Nix



uncontrolled build environment = untraceable failures

which compiler? which libc?
and now your favorite admin upgrades the system...
Segmentation fault

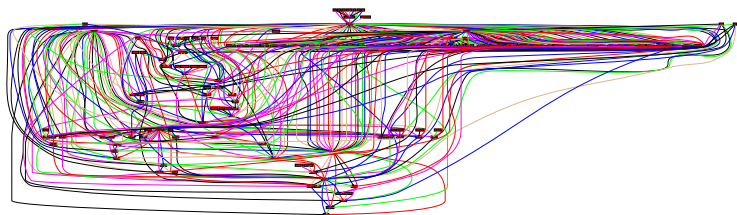
build environment control with Nix



... down to the compiler's compiler!

¹Except for the kernel, the hardware, the direction of the wind, etc.

build environment control with Nix



... down to the compiler's compiler!

- ▶ build environment **entirely under control**¹
- ▶ **all the build inputs** are known to Nix

¹Except for the kernel, the hardware, the direction of the wind, etc.

introduction

features

practice

closing

hello, world!

a job set

Project

- > Overview
- > Releases
- > All builds
- > Job status
- > Errors

Jobset

- + Overview
- > All builds
- > Job status
- > Errors

Jobset runtime:hwloc-trunk

Jobset Jobs (7) Setup Channels

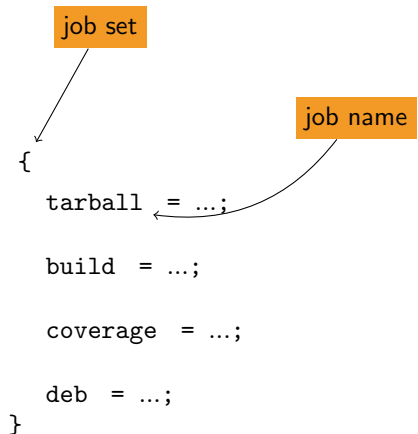
Most recent builds

◆ # ◆	Job	◆ Release Name ◆	◆ System ◆	◆ Timestamp ◆
✓ 10744	runtime:hwloc-trunk:build_opensuse110_x86_64	hwloc-1.4a1-1	x86_64-linux	2011-10-30 05:10:37
✓ 10738	runtime:hwloc-trunk:build_fedora13_x86_64	hwloc-1.4a1-1	x86_64-linux	2011-10-30 05:05:54
✓ 10741	runtime:hwloc-trunk:build_fedora2_i386	hwloc-1.4a1-1	x86_64-linux	2011-10-30 05:04:07
✓ 10737	runtime:hwloc-trunk:coverage	hwloc-1.4a1-1	x86_64-linux	2011-10-30 02:11:19
✓ 10735	runtime:hwloc-trunk:build	hwloc-1.4a1-1	x86_64-linux	2011-10-30 02:10:23

Status

Job	i686-linux	x86_64-darwin	x86_64-linux
build	✓	✓	✓
build_fedora13_x86_64			
build_fedora2_i386			
build_opensuse110_x86_64			
coverage	✓	✓	✓
debian50_i386			
tarball			✓

writing a Hydra job set in Nix



writing a Hydra job set in Nix

```
{  
  tarball ← = ...; make dist  
  
  build ← = ...; make && make check...  
  
  coverage ← = ...; make CFLAGS=--coverage...  
  
  deb ← = ...; qemu... checkinstall...  
}
```

writing Hydra jobs in Nix

```
tarball = parameters  
  { nixpkgs , helloSrc } : function  
  
  let pkgs = import nixpkgs { }; in  
  pkgs.releaseTools.sourceTarball {  
    name = "hello";  
    src = helloSrc;  
  
  };
```

writing Hydra jobs in Nix

```
tarball =  
  { nixpkgs , helloSrc } :  
  
  let pkgs = import nixpkgs { }; in  
  pkgs.releaseTools.sourceTarball {  
    name = "hello";  
    src = helloSrc;  
  
  };
```

checkout of the distro

checkout of Hello

writing Hydra jobs in Nix

result of tarball job

x86_64-linux, or x86_64-darwin, etc.

```
build =  
  { nixpkgs , tarball , system } :
```

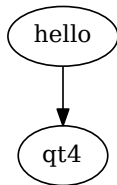
```
let pkgs = import nixpkgs { inherit system; }; in  
pkgs.releaseTools.nixBuild {  
  name = "hello";  
  src = tarball;
```

```
};
```

writing Hydra jobs in Nix

```
build =  
  { nixpkgs , tarball , system } :  
  
  let pkgs = import nixpkgs { inherit system; }; in  
  pkgs.releaseTools.nixBuild {  
    name = "hello";  
    src = tarball;  
    buildInputs = [ pkgs.qt4 ];  
  
  };
```

explicit dependency



writing Hydra jobs in Nix

```
build =  
  { nixpkgs , tarball , system } :  
  
  let pkgs = import nixpkgs { inherit system; }; in  
  pkgs.releaseTools.nixBuild {  
    name = "hello";  
    src = tarball;  
    buildInputs = [ pkgs.qt4 ];  
    configureFlags = "--with-colors";  
  
  };
```

writing Hydra jobs in Nix

```
coverage =  
  { nixpkgs , tarball , system } :  
  
  let pkgs = import nixpkgs { inherit system; }; in  
  pkgs.releaseTools.coverageAnalysis {  
    name = "hello";  
    src = tarball;  
    buildInputs = [ pkgs.qt4 ];  
    configureFlags = "--with-colors";  
  
  };
```

writing Hydra jobs in Nix

```
deb =  
  { nixpkgs , tarball , system } :  
  
  let pkgs = import nixpkgs { inherit system; }; in  
  pkgs.releaseTools.debBuild {  
    name = "hello";  
    src = tarball;  
  
    configureFlags = "--with-colors";  
    diskImage = vmTools.diskImages.debian50x86_64;  
  };
```



filling in the blanks...

Project

- > Overview
- > Releases
- > All builds
- > Job status
- > Errors

Jobset

- ➔ Overview
- > All builds
- > Job status
- > Errors

Jobset runtime:hwloc-trunk

Jobset Jobs (7) Setup Channels

Information

[[Edit](#) | [Clone](#)]

Description: hwloc, 'trunk'
Nix expression: runtime/hwloc/release.nix in input hydraConfig
Enabled: Yes
Enable email notification: Yes
Email override:
Number of builds to keep: 3
Last checked: 2011-11-02 14:00:15, *no errors*

Inputs

Input name	Type	Values
hwlocSrc	Subversion export	http://svn.open-mpi.org/svn/hwloc/trunk
hydraConfig	Git checkout	git://scm.gforge.inria.fr/hydra-recipes/hydra-recipes.git
nixpkgs	Subversion export	https://svn.nixos.org/repos/nix/nixpkgs/trunk/
system	String value	"x86_64-linux" "i686-linux" "x86_64-darwin"
tarball	Build output	tarball

“But Autotools suck, CMake rocks.”

— Anonymous (and cavalier) attendee.

building with CMake

```
build =  
  { system, nixpkgs, helloSrc }:  
  
  let pkgs = import nixpkgs { inherit system; }; in  
  pkgs.releaseTools.nixBuild {  
    name = "hello-cmake" ;  
    src = helloSrc;  
    cmakeFlags = "-DFOO_BAR=on";  
    buildInputs = [ pkgs.cmake pkgs.qt4 ];  
  };
```

“All these C-ish things suck anyway; Java is the way.”

— Anonymous attendee.

building with Ant

```
build =  
  { nixpkgs, helloSrc }:  
  
  let pkgs = import nixpkgs { }; in  
  pkgs.releaseTools.antBuild {  
    name = "hello-ant" ;  
    src = helloSrc;  
    antTargets = [ "jar" "test" ];  
  
    buildInputs = with pkgs; [ ant jdk ];  
  };
```

building with Maven

```
build =  
  { nixpkgs, helloSrc }:  
  
  let pkgs = import nixpkgs { }; in  
  pkgs.releaseTools.mvnBuild {  
    name = "hello-maven" ;  
    src = helloSrc;  
  
    doJavadoc = true;  
    doCheckstyle = true;  
    buildInputs = with pkgs; [ maven2 jdk ];  
  };
```

building things differently

- ▶ **customize** existing build functions (`nixBuild`, etc.)

- ▶ **write** your own build function

building things differently

- ▶ **customize** existing build functions (`nixBuild`, etc.)
 - ▶ define `configureFlags`, `makeFlags`, etc.
 - ▶ define `configurePhase`, `buildPhase`, etc. (shell snippets)
 - ▶ ...
- ▶ **write** your own build function
 - ▶ `pythonBuild` anyone?

introduction

features

practice

closing

Continuous integration pacifies developers!

Continuous integration pacifies developers!

- ▶ quick feedback to **developers**
- ▶ source + binaries + doc snapshots for **users**
- ▶ tests of various **configurations**

Continuous integration pacifies developers!

- ▶ quick feedback to **developers**
- ▶ source + binaries + doc snapshots for **users**
- ▶ tests of various **configurations**
- ▶ **available today!**

Hydra @ Inria: what's missing

- ▶ production-ready!

Hydra @ Inria: what's missing

- ▶ not quite production-ready

Hydra @ Inria: what's missing

- ▶ not quite production-ready
- ▶ no private/proprietary projects yet

Hydra @ Inria: what's missing

- ▶ not quite production-ready
- ▶ no private/proprietary projects yet
- ▶ no GPUs, no webcams, no USB percolator (yet)

Hydra @ Inria: what's missing

- ▶ not quite production-ready
- ▶ no private/proprietary projects yet
- ▶ no GPUs, no webcams, no USB percolator (yet)
- ▶ some of your dependencies might be missing

Hydra @ Inria: what's missing

- ▶ not quite production-ready
- ▶ no private/proprietary projects yet
- ▶ no GPUs, no webcams, no USB percolator (yet)
- ▶ some of your dependencies might be missing
- ▶ no blame list, no *favorite Jenkins plug-in here*
- ▶ ...

Hydra @ Inria: what's already there

- ▶ small & growing compile farm: 12 x86_64 cores

Hydra @ Inria: what's already there

- ▶ small & growing compile farm: 12 x86_64 cores
- ▶ GNU/Linux, MacOS X, and FreeBSD

Hydra @ Inria: what's already there

- ▶ small & growing compile farm: 12 x86_64 cores
- ▶ GNU/Linux, MacOS X, and FreeBSD
- ▶ kernel & distributed tests in virtual machines
- ▶ virtual build environments: Debian, Fedora, openSuSE, etc.

Hydra @ Inria: what's already there

- ▶ small & growing compile farm: 12 x86_64 cores
- ▶ GNU/Linux, MacOS X, and FreeBSD
- ▶ kernel & distributed tests in virtual machines
- ▶ virtual build environments: Debian, Fedora, openSuSE, etc.
- ▶ 6.5 projects being built (C/C++/Fortran, Autotools/CMake)

Hydra @ Inria: what's already there

- ▶ small & growing compile farm: 12 x86_64 cores
- ▶ GNU/Linux, MacOS X, and FreeBSD
- ▶ kernel & distributed tests in virtual machines
- ▶ virtual build environments: Debian, Fedora, openSuSE, etc.
- ▶ 6.5 projects being built (C/C++/Fortran, Autotools/CMake)
- ▶ ...

getting started

1. email `sed-bordeaux@inria.fr`

getting started

1. email `sed-bordeaux@inria.fr`
2. add your recipes to
`http://gforge.inria.fr/projects/hydra-recipes`

getting started

1. email `sed-bordeaux@inria.fr`
2. add your recipes to
`http://gforge.inria.fr/projects/hydra-recipes`
3. watch your project at `http://hydra.bordeaux.inria.fr/`

getting help

- ▶ copy/paste from `http://gforge.inria.fr/projects/hydra-recipes`
- ▶ read the fine manual at `http://nixos.org/hydra/`
- ▶ email `sed-bordeaux@inria.fr`

The Inria logo is displayed in a white rounded square. The word "Inria" is written in a stylized, cursive font. The letters "i", "n", and "r" are in a dark red color, while the letters "i", "a", and "a" are in a golden-yellow color.

Inria

ludovic.courtes@inria.fr

<http://sed.bordeaux.inria.fr/>