# The Hydra/Nix Approach to Continuous Integration

Ludovic Courtès

INRIA SED — Cépage Team-Project
devexp.bor@inria.fr

5 November 2009

# Outline

1. Build & Deployment with Nix

2. Continuous Integration with Hydra

3. The End

# A Declarative Approach

**“Derivations” (aka. “Build Jobs”)**

- a system type
- a list of build inputs
- a build process
- an output

# A Declarative Approach

### “Derivations” (aka. “Build Jobs”)

- a system type — `"i686-linux"`
- a list of build inputs — other derivations
- a build process — an arbitrary script
- an output — file tree

# The Nix Build Expression Language

## The `derivation` Primitive

```
derivation {
  name = "foo";
  system = "x86_64-linux";
  builder = builtins.toFile "builder.sh"
    '' mkdir -p "$out"
       echo "Hello, world!" > "$out/some-result"
    '';
}
```

# Job Composition

```
          derivation {
  name = "foo";
  system = "x86_64-linux";
  builder = builtins.toFile "builder.sh"
    '' mkdir -p "$out"
       echo "Hello, world!" > "$out/some-result"
    '';
}
```

# Job Composition

```
let dep = derivation {
  name = "foo";
  system = "x86_64-linux";
  builder = builtins.toFile "builder.sh"
    '' mkdir -p "$out"
       echo "Hello, world!" > "$out/some-result"
    '';
};
```

# Job Composition

```
let dep = derivation {
  name = "foo";
  system = "x86_64-linux";
  builder = builtins.toFile "builder.sh"
    '' mkdir -p "$out"
       echo "Hello, world!" > "$out/some-result"
    '';
}; in derivation {
  name = "bar";
  system = "x86_64-linux";
  builder = builtins.toFile "builder.sh"
    '' mkdir -p "$out"
       ln -s "${dep}/some-result" "$out/my-result"
    '';
}
```

# Complex Nix Expressions (“Job Procedures”)

## GNU Hello

```
{fetchurl, stdenv}:

stdenv.mkDerivation {
  name = "hello-2.3";
  src = fetchurl {
    url = mirror://gnu/hello/hello-2.3.tar.bz2;
    sha256 = "0c7vijq8y68bpr7g6dh1gny0bff8qq81vnp4ch8pjzvg56wb3js1"
  };

  meta = {
    description = "A program that produces a friendly greeting";
    homepage = http://www.gnu.org/software/hello/;
  };
}
```

# Complex Nix Expressions (“Job Procedures”)

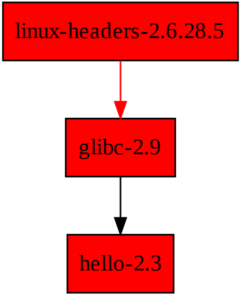
## GNU Hello

```
{fetchurl, stdenv, gettext}:

stdenv.mkDerivation {
  name = "hello-2.3";
  src = fetchurl {
    url = mirror://gnu/hello/hello-2.3.tar.bz2;
    sha256 = "0c7vijq8y68bpr7g6dh1gny0bff8qq81vnp4ch8pjzvg56wb3js1"
  };
  buildInputs = [ gettext ];
  meta = {
    description = "A program that produces a friendly greeting";
    homepage = http://www.gnu.org/software/hello/;
  };
}
```

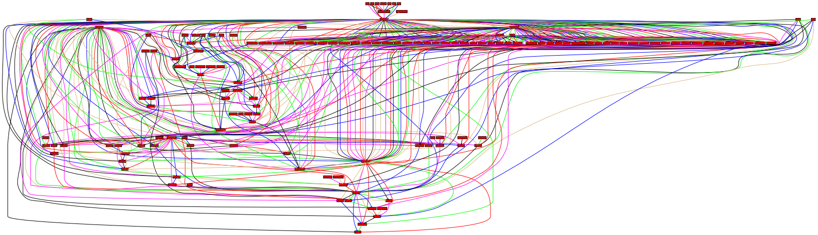
# Dependency Graph for GNU Hello

## Run-Time Dependencies



# Dependency Graph for GNU Hello

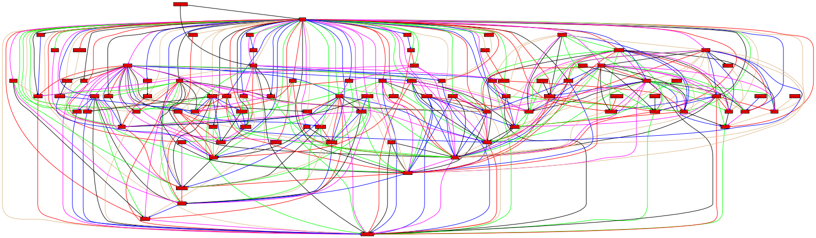
## Compile-Time Dependencies



- 147 nodes
- 535 edges

# Dependency Graph for OpenOffice.org

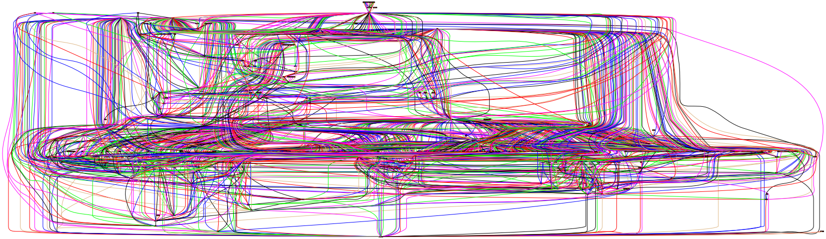
## Run-Time Dependencies



- 94 nodes
- 426 edges

# Dependency Graph for OpenOffice.org

## Compile-Time Dependencies



- 411 nodes
- 2043 edges

# Compilation Result Caching ("Memoization")

### Store Paths

```
$ nix-build -A hello
```

# Compilation Result Caching (“Memoization”)

### Store Paths

```
$ nix-build -A hello
/nix/store/ksaxbhsnwmyxilx5ha8k704cp2iabh0y-hello-2.3
```

# Compilation Result Caching (“Memoization”)

## Store Paths

```
$ nix-build -A hello
/nix/store/ksaxbhsnwmyxilx5ha8k704cp2iabh0y-hello-2.3
```

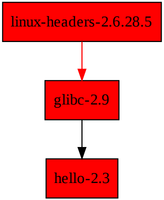
## How Caching Works

- all inputs *explicitly* given to `derivation`
- output path = hash of all build inputs

# Dependencies Among Derivation Outputs

## Problem: What Does a Build Output Depend On?

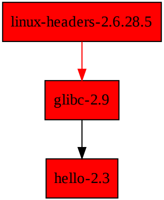
- a build result may depend on a previous build result
- example: a program linked with a library



# Dependencies Among Derivation Outputs

## Problem: What Does a Build Output Depend On?

- a build result may depend on a previous build result
- example: a program linked with a library



## Solution: Conservatively Scan Output Files

- scan output files for occurrences of `/nix/store/*`
- maintain a dependency graph

# Support for Distributed Builds

### The “Build Hook”

- derivation realization can be deferred to a “hook”
- the hook can accept, reject, or postpone

# Support for Distributed Builds

## The "Build Hook"

- derivation realization can be deferred to a "hook"
- the hook can accept, reject, or postpone
- the `build-remote` hook:
  1. finds a host for the system type (e.g., `i686-linux`)
  2. copies missing inputs to the remote host
  3. copies output(s) back to the build host

# Support for Distributed Builds

## The "Build Hook"

- derivation realization can be deferred to a "hook"
- the hook can accept, reject, or postpone
- the `build-remote` hook:
  1. finds a host for the system type (e.g., `i686-linux`)
  2. copies missing inputs to the remote host
  3. copies output(s) back to the build host

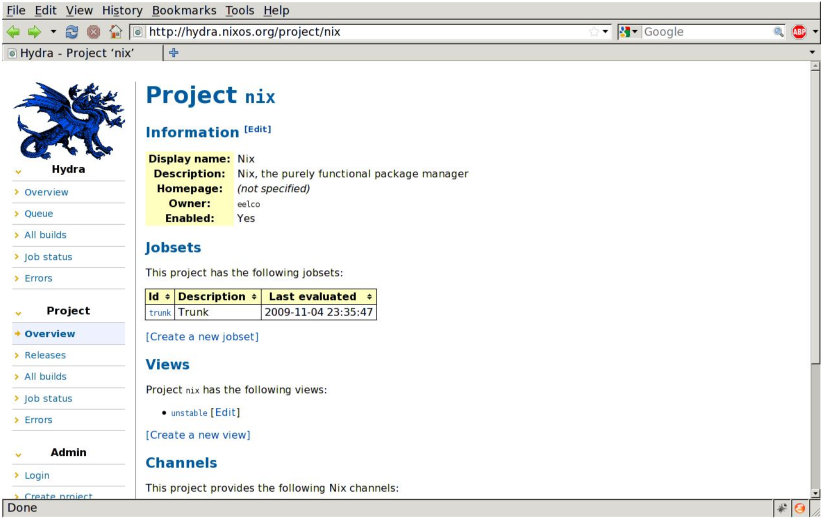
⇒ the foundation for Hydra's distributed builds

# Outline

1. Build & Deployment with Nix
2. Continuous Integration with Hydra
3. The End

# Hydra: Distributed Continuous Integration

## with fancy web interface (not as fancy as Xooctory's)



# Project nix

## Information [Edit]

| | |
|---|---|
| **Display name:** | Nix |
| **Description:** | Nix, the purely functional package manager |
| **Homepage:** | *(not specified)* |
| **Owner:** | eelco |
| **Enabled:** | Yes |

## Jobsets

This project has the following jobsets:

| Id | Description | Last evaluated |
|---|---|---|
| trunk | Trunk | 2009-11-04 23:35:47 |

[Create a new jobset]

## Views

Project nix has the following views:

- unstable [Edit]

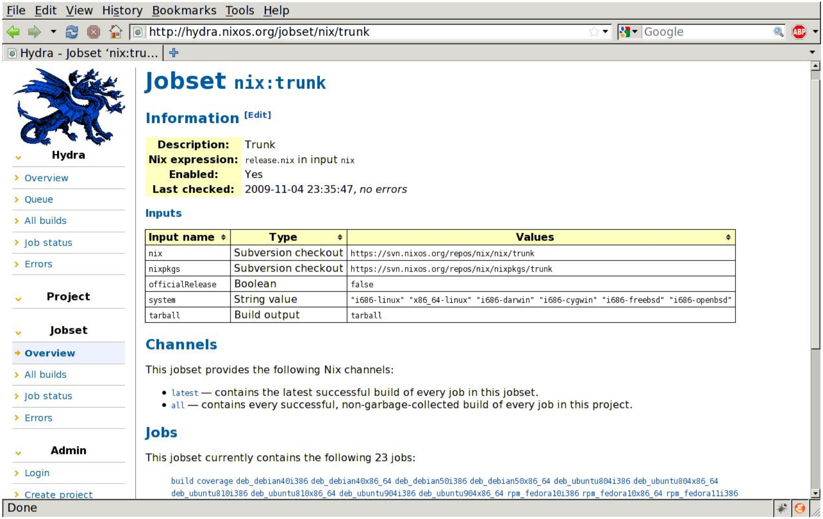
[Create a new view]

## Channels

This project provides the following Nix channels:

# Hydra: Distributed Continuous Integration

## with fancy web interface (not as fancy as Xooctory's)



# Jobset `nix:trunk`

## Information [Edit]

| | |
|---|---|
| **Description:** | Trunk |
| **Nix expression:** | `release.nix` in input `nix` |
| **Enabled:** | Yes |
| **Last checked:** | 2009-11-04 23:35:47, *no errors* |

### Inputs

| Input name | Type | Values |
|---|---|---|
| `nix` | Subversion checkout | `https://svn.nixos.org/repos/nix/nix/trunk` |
| `nixpkgs` | Subversion checkout | `https://svn.nixos.org/repos/nix/nixpkgs/trunk` |
| `officialRelease` | Boolean | `false` |
| `system` | String value | `"i686-linux" "x86_64-linux" "i686-darwin" "i686-cygwin" "i686-freebsd" "i686-openbsd"` |
| `tarball` | Build output | `tarball` |

## Channels

This jobset provides the following Nix channels:

- `latest` — contains the latest successful build of every job in this jobset.
- `all` — contains every successful, non-garbage-collected build of every job in this project.

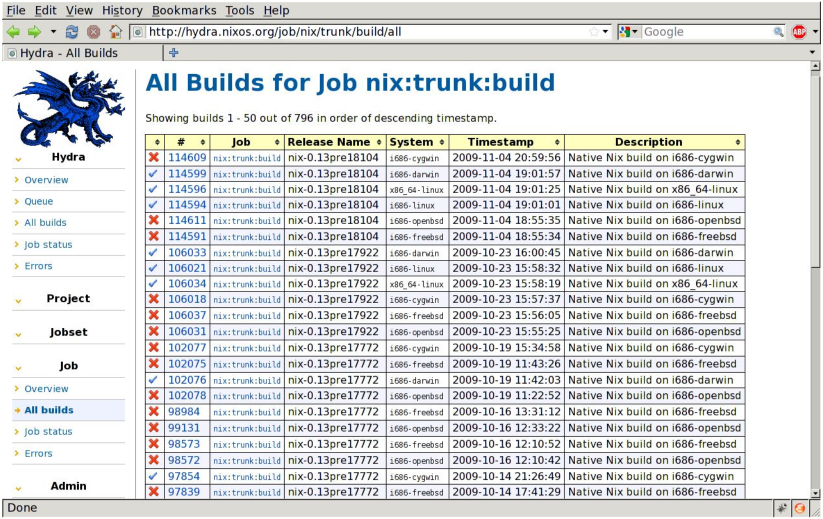
## Jobs

This jobset currently contains the following 23 jobs:

build coverage deb_debian40i386 deb_debian40x86_64 deb_debian50i386 deb_debian50x86_64 deb_ubuntu804i386 deb_ubuntu804x86_64 deb_ubuntu810i386 deb_ubuntu810x86_64 deb_ubuntu904i386 deb_ubuntu904x86_64 rpm_fedora10i386 rpm_fedora10x86_64 rpm_fedora11i386

# Hydra: Distributed Continuous Integration

## with fancy web interface (not as fancy as Xooctory's)



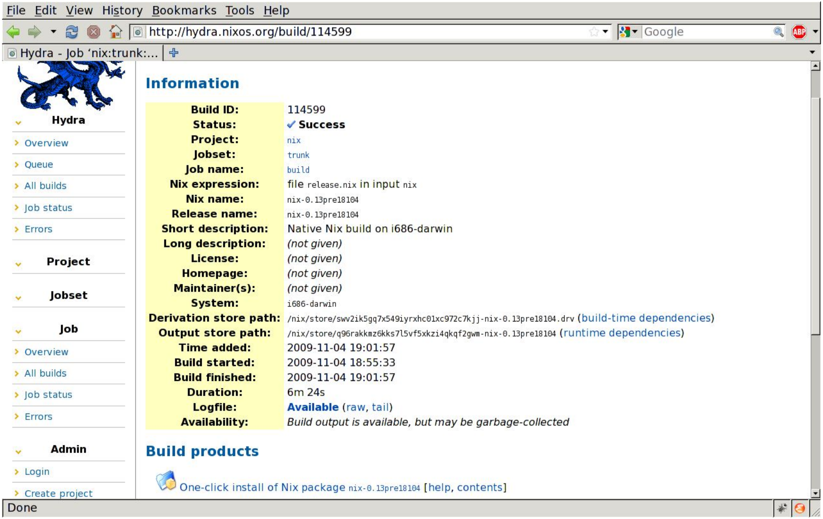
### All Builds for Job nix:trunk:build

Showing builds 1 - 50 out of 796 in order of descending timestamp.

| | # | Job | Release Name | System | Timestamp | Description |
|---|---|---|---|---|---|---|
| ✗ | 114609 | nix:trunk:build | nix-0.13pre18104 | i686-cygwin | 2009-11-04 20:59:56 | Native Nix build on i686-cygwin |
| ✓ | 114599 | nix:trunk:build | nix-0.13pre18104 | i686-darwin | 2009-11-04 19:01:57 | Native Nix build on i686-darwin |
| ✓ | 114596 | nix:trunk:build | nix-0.13pre18104 | x86_64-linux | 2009-11-04 19:01:25 | Native Nix build on x86_64-linux |
| ✓ | 114594 | nix:trunk:build | nix-0.13pre18104 | i686-linux | 2009-11-04 19:01:01 | Native Nix build on i686-linux |
| ✗ | 114611 | nix:trunk:build | nix-0.13pre18104 | i686-openbsd | 2009-11-04 18:55:35 | Native Nix build on i686-openbsd |
| ✗ | 114591 | nix:trunk:build | nix-0.13pre18104 | i686-freebsd | 2009-11-04 18:55:34 | Native Nix build on i686-freebsd |
| ✓ | 106033 | nix:trunk:build | nix-0.13pre17922 | i686-darwin | 2009-10-23 16:00:45 | Native Nix build on i686-darwin |
| ✓ | 106021 | nix:trunk:build | nix-0.13pre17922 | i686-linux | 2009-10-23 15:58:32 | Native Nix build on i686-linux |
| ✓ | 106034 | nix:trunk:build | nix-0.13pre17922 | x86_64-linux | 2009-10-23 15:58:19 | Native Nix build on x86_64-linux |
| ✗ | 106018 | nix:trunk:build | nix-0.13pre17922 | i686-cygwin | 2009-10-23 15:57:37 | Native Nix build on i686-cygwin |
| ✗ | 106037 | nix:trunk:build | nix-0.13pre17922 | i686-freebsd | 2009-10-23 15:56:05 | Native Nix build on i686-freebsd |
| ✗ | 106031 | nix:trunk:build | nix-0.13pre17922 | i686-openbsd | 2009-10-23 15:55:25 | Native Nix build on i686-openbsd |
| ✗ | 102077 | nix:trunk:build | nix-0.13pre17772 | i686-cygwin | 2009-10-19 15:34:58 | Native Nix build on i686-cygwin |
| ✗ | 102075 | nix:trunk:build | nix-0.13pre17772 | i686-freebsd | 2009-10-19 11:43:26 | Native Nix build on i686-freebsd |
| ✓ | 102076 | nix:trunk:build | nix-0.13pre17772 | i686-darwin | 2009-10-19 11:42:03 | Native Nix build on i686-darwin |
| ✗ | 102078 | nix:trunk:build | nix-0.13pre17772 | i686-openbsd | 2009-10-19 11:22:52 | Native Nix build on i686-openbsd |
| ✗ | 98984 | nix:trunk:build | nix-0.13pre17772 | i686-freebsd | 2009-10-16 13:31:12 | Native Nix build on i686-freebsd |
| ✗ | 99131 | nix:trunk:build | nix-0.13pre17772 | i686-openbsd | 2009-10-16 12:33:22 | Native Nix build on i686-openbsd |
| ✗ | 98573 | nix:trunk:build | nix-0.13pre17772 | i686-freebsd | 2009-10-16 12:10:52 | Native Nix build on i686-freebsd |
| ✗ | 98572 | nix:trunk:build | nix-0.13pre17772 | i686-openbsd | 2009-10-16 12:10:42 | Native Nix build on i686-openbsd |
| ✓ | 97854 | nix:trunk:build | nix-0.13pre17772 | i686-cygwin | 2009-10-14 21:26:49 | Native Nix build on i686-cygwin |
| ✗ | 97839 | nix:trunk:build | nix-0.13pre17772 | i686-freebsd | 2009-10-14 17:41:29 | Native Nix build on i686-freebsd |

# Hydra: Distributed Continuous Integration

## with fancy web interface (not as fancy as Xooctory's)



### Information

| | |
|---|---|
| **Build ID:** | 114599 |
| **Status:** | ✓ **Success** |
| **Project:** | `nix` |
| **Jobset:** | `trunk` |
| **Job name:** | `build` |
| **Nix expression:** | file `release.nix` in input `nix` |
| **Nix name:** | `nix-0.13pre18104` |
| **Release name:** | `nix-0.13pre18104` |
| **Short description:** | Native Nix build on i686-darwin |
| **Long description:** | *(not given)* |
| **License:** | *(not given)* |
| **Homepage:** | *(not given)* |
| **Maintainer(s):** | *(not given)* |
| **System:** | `i686-darwin` |
| **Derivation store path:** | `/nix/store/swv2ik5gq7x549iyrxhc01xc972c7kjj-nix-0.13pre18104.drv` (build-time dependencies) |
| **Output store path:** | `/nix/store/q96rakkmz6kks7l5vf5xkzi4qkqf2gwm-nix-0.13pre18104` (runtime dependencies) |
| **Time added:** | 2009-11-04 19:01:57 |
| **Build started:** | 2009-11-04 18:55:33 |
| **Build finished:** | 2009-11-04 19:01:57 |
| **Duration:** | 6m 24s |
| **Logfile:** | **Available** (raw, tail) |
| **Availability:** | *Build output is available, but may be garbage-collected* |

### Build products

One-click install of Nix package `nix-0.13pre18104` [help, contents]

# Jargon Mapping

### Xooctory

- job plan
- job

### Hydra

- project, jobset, job
- build

# Jargon Mapping

### Xooctory

- job plan
- job
- "what" (`JobResource`)
- "when" (`JobTrigger`)
- "how" (`JobProcedure`, `ResultCollector`)

### Hydra

- project, jobset, job
- build
- build inputs (Nix)
- SCM change (Hydra)
- builder (Nix), build products (Hydra)

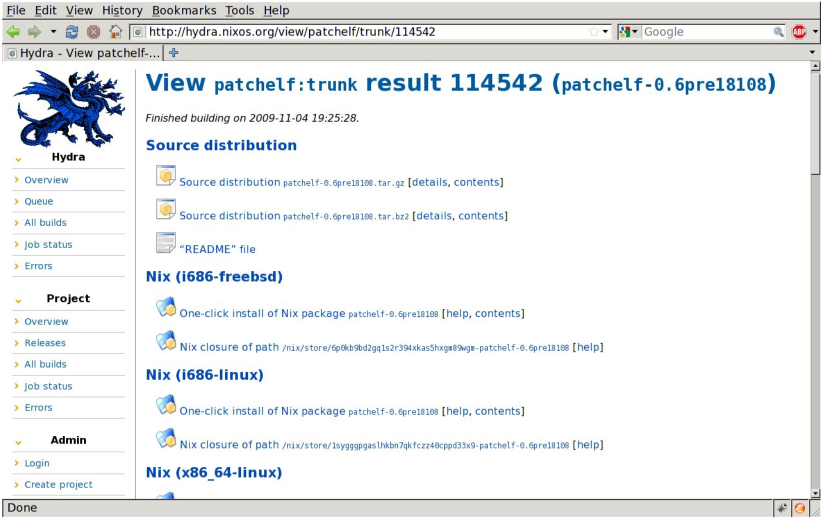
# Types of Build Results

## Results

Written to `$out/nix-support/hydra-build-products`.

- build logs
- binaries: Nix *store paths*, `.debs`, `.rpms`
- source tarballs
- ISO 9660 CD images
- code coverage reports
- screenshots

# Types of Build Results



## View `patchelf:trunk` result 114542 (`patchelf-0.6pre18108`)

*Finished building on 2009-11-04 19:25:28.*

### Source distribution

Source distribution `patchelf-0.6pre18108.tar.gz` [details, contents]

Source distribution `patchelf-0.6pre18108.tar.bz2` [details, contents]

“README” file

### Nix (i686-freebsd)

One-click install of Nix package `patchelf-0.6pre18108` [help, contents]

Nix closure of path `/nix/store/6p0kb9bd2gq1s2r394xkas5hxgm89wgm-patchelf-0.6pre18108` [help]

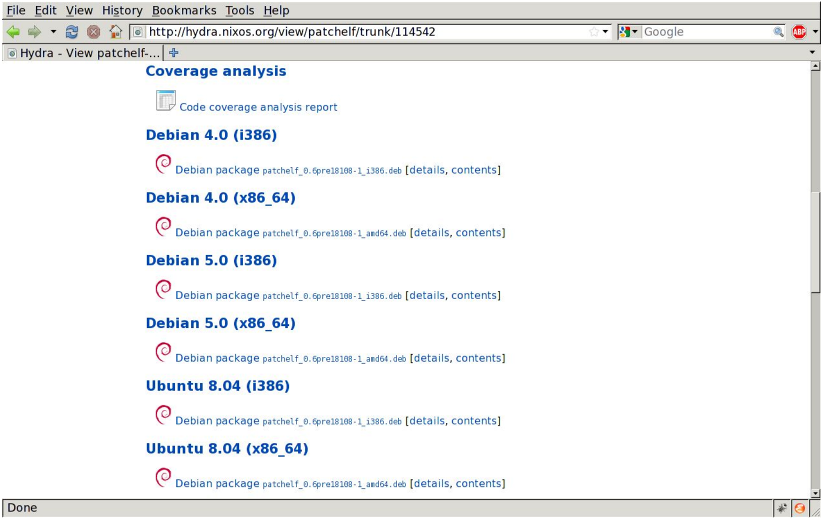
### Nix (i686-linux)

One-click install of Nix package `patchelf-0.6pre18108` [help, contents]

Nix closure of path `/nix/store/1syggpgaslhkbn7qkfczz40cppd33x9-patchelf-0.6pre18108` [help]

### Nix (x86_64-linux)

# Types of Build Results


File Edit View History Bookmarks Tools Help
http://hydra.nixos.org/view/patchelf/trunk/114542
Google
Hydra - View patchelf-...
Done

### Coverage analysis

Code coverage analysis report

### Debian 4.0 (i386)

Debian package patchelf_0.6pre18108-1_i386.deb [details, contents]

### Debian 4.0 (x86_64)

Debian package patchelf_0.6pre18108-1_amd64.deb [details, contents]

### Debian 5.0 (i386)

Debian package patchelf_0.6pre18108-1_i386.deb [details, contents]

### Debian 5.0 (x86_64)

Debian package patchelf_0.6pre18108-1_amd64.deb [details, contents]

### Ubuntu 8.04 (i386)

Debian package patchelf_0.6pre18108-1_i386.deb [details, contents]

### Ubuntu 8.04 (x86_64)

Debian package patchelf_0.6pre18108-1_amd64.deb [details, contents]

# Architecture

## Data Storage

- the Nix store, used as a build cache
- database for project info, available builds, etc.

## 3 Processes

- the "scheduler"
- the queue runner
- the web interface

# Architecture

## Data Storage

- the Nix store, used as a build cache
- database for project info, available builds, etc.

## 3 Processes

- the “scheduler” : looks for SCM changes, queues builds
- the queue runner
- the web interface

# Architecture

## Data Storage

- the Nix store, used as a build cache
- database for project info, available builds, etc.

## 3 Processes

- the “scheduler” : looks for SCM changes, queues builds
- the queue runner : executes queued builds
- the web interface

# Architecture

### Data Storage

- the Nix store, used as a build cache
- database for project info, available builds, etc.

### 3 Processes

- the “scheduler” : looks for SCM changes, queues builds
- the queue runner : executes queued builds
- the web interface : click!

# Job Scheduling

1. upon SCM change, enqueue builds for `x86_64-linux`, `i686-freebsd`, etc.
2. pick up a build from the queue
3. build it, relying on a build hook to distribute builds
4. the build hook copies missing input, then fetches the result

# Job Scheduling

1. upon SCM change, enqueue builds for `x86_64-linux`, `i686-freebsd`, etc.
2. pick up a build from the queue
3. build it, relying on a build hook to distribute builds
4. the build hook copies missing input, then fetches the result

## Shortcomings

- little information available to the build hook
- e.g., accepts/rejects builds without knowing its inputs

⇒ can't make decisions based on input availability

# Outline

1 Build & Deployment with Nix

2 Continuous Integration with Hydra

3 The End

# Summary

### Building with Nix

- *complete* dependency graph through "Nix expressions"
- automatically maintained dependency graph among outputs
- automatic build result cache

# Summary

### Building with Nix

- *complete* dependency graph through “Nix expressions”
- automatically maintained dependency graph among outputs
- automatic build result cache

### Continuous Integration with Hydra

- fancy web interface (RESTful)
- distributed builds on heterogeneous machines
- currently simplistic build scheduling
- `hydra.nixos.org` @ TU Delft: 2500 jobs, 36 cores, 6 system types (Nov. 2009)

# Thanks!

# Links

## Web

- `http://nixos.org/` — Nix, NixOS, and Hydra
- `http://nixos.org/docs/papers.html` — papers
- `http://hydra.nixos.org/` — Hydra instance for NixOS, etc.

## Papers

- E. Dolstra and A. Hemel, *Purely Functional System Configuration Management*, HotOS XI, May 2007.
- E. Dolstra and E. Visser, *Hydra: A Declarative Approach to Continuous Integration*, 2008, submitted.
- E. Dolstra, *Nix User Guide*, 2004–2009