Disnix: A toolset for distributed deployment

Sander van der Burg    Eelco Dolstra

Delft University of Technology, EEMCS,
Department of Software Technology

September 20, 2010
Disnix is a tool to automatically deploy a distributed system in a network of machines
Research issues

- General issues:
  - Automatic deployment
  - Reliable deployment
  - Efficient deployment
  - Atomic upgrades/rollbacks

- Domain specific issues:
  - Security
  - Privacy
  - Performance
Research issues

It is complex to implement and maintain a toolset dealing with these issues!
Challenges

- How to develop this toolset in a convenient way?
- Some non-functional requirements of the domain cannot be solved generically
- Maintaining everything ourselves is a lot of effort:
  - Programming language compilers/interpreters: C, Java, Python, PHP, ...
  - Libraries: Apache Axis2, libxml2, ...
  - Infrastructure components: Apache Tomcat, MySQL, ...
- The tool is used in a distributed setting. How to test?

Sander van der Burg, Eelco Dolstra

Disnix: A toolset for distributed deployment
Disnix: A toolset for distributed deployment

Architecture

User input

- services.nix
- infrastructure.nix
- distribution.nix

Optional

- disnix-instantiate

- distributed derivation

- disnix-build
- disnix-copy-closure
- disnix-distribute
- disnix-activate

manifest

disnix-manifest
Architecture is mainly composition of processes (command-line tools)

Each tool performs a separate deployment task in the deployment process:
- Building
- Transferring
- Activating

Inspired by Art of UNIX programming book
A user can easily perform steps separately
Testing and debugging is relatively easy; things can be easily scripted
Components can be implemented using various programming languages: Nix, C, shell scripts
Prototyping is relatively easy. First use a high-level language, implement later in a low-level language
Architecture

Extensions with custom modules are implemented by composing processes:

- Virtualization extension
- Dynamic deployment
- Service abstraction layer
Testing

- Generate cheap virtual networks from declarative specifications
- Automatically performs testcases on them
- Completely scripted
- Disnix is continuously integrated and tested by Hydra
Community

- Disnix is part of the Nix project
- Tools are released under free and open-source licenses
- About 25 contributors
- Cross compiling
- Nixpkgs: contains compilers, libraries, tools which can be used by Disnix
Lessons learned

- Architecture works well for supporting development
  - Easy prototyping, debugging, testing, extendable
- Our test approach works well; we can automatically perform distributed testcases
- Community is crucial; too much effort to maintain everything ourselves
Disnix, and other related Nix tooling: Hydra, NixOS can be freely downloaded from: http://nixos.org