Sourceforge.net
CVS ~ Compile Farm
Each project is provided with a repository
  - Developers automatically granted permissions to commit changes
  - Read-only anonymous pserver-based access to repositories
  - Web-based access to repositories
Anonymous pserver access

- Uses pserver (password-authenticated server) method for anonymous access
  - `cvs -d:pserver:anonymous@cvs.sf.net:/cvsroot/<proj>` login
- Password is left blank for sourceforge
- Pserver sends passwords plaintext
  - Not used for developer access
Anonymous web-based access

- Uses ViewCVS, an open source project hosted on sourceforge

```
[cvs]/tmans/ABNNSim/abnnsim

cvs: tmans/ABNNSim/abnnsim
```

Current directory: [cvs]/tmans/ABNNSim/abnnsim
Files shown: 10

<table>
<thead>
<tr>
<th>File</th>
<th>Rev.</th>
<th>Age</th>
<th>Author</th>
<th>Last log entry</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applic/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>data/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>heap/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tests/</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABNNSimModel.java</td>
<td>1.1.1.1</td>
<td>8 weeks</td>
<td>jskor2</td>
<td>Initial import.</td>
</tr>
<tr>
<td>Axon.java</td>
<td>1.1.1.1</td>
<td>8 weeks</td>
<td>jskor2</td>
<td>Initial import.</td>
</tr>
<tr>
<td>HeapableNode.java</td>
<td>1.1.1.1</td>
<td>8 weeks</td>
<td>jskor2</td>
<td>Initial import.</td>
</tr>
<tr>
<td>NetworkAnalyzer.java</td>
<td>1.1.1.1</td>
<td>8 weeks</td>
<td>jskor2</td>
<td>Initial import.</td>
</tr>
</tbody>
</table>
Developer Access

- Uses ssh for authentication
  - setenv CVS_RSH ssh

- Developer login performed before command is performed
  - cvs -d:ext:<name>@cvs.sf.net:/cvsroot/<proj>
    import -m "<comment>" <modulename> vendor start
  - cvs -d:ext:<name>@cvs.sf.net:/cvsroot/<proj>
    checkout <modulename>
Developer Access

- “All developer operations must be performed using SSH for authentication. If you use pserver authentication to checkout your working copy, you WILL NOT be able to perform write operations.”
- Any CVS / SSH client pair can be used
  - Tutorials for Tortoise CVS and PuTTY in windows
Other CVS policies/services

- No interactive access available
- Nightly backup of daily changes
  - bzip2-compressed .tar archive
  - “Download Your Nightly CVS Tree Tarball” link on the Project Admin page
Sourceforge.net Compile Farm

- Pool of hosts available to developers for compiling and testing software
- 12 hosts
  - Six distinct operating systems
  - 5 separate architectures
Compile Farm Hosts

- 32-bit x86 Architecture:
  - x86-linux2: Fedora Linux FC2 running Linux 2.6 kernel
  - x86-openbsd1: OpenBSD 3.4
  - x86-solaris1: Sun Solaris 9
  - x86-linux1: Debian GNU/Linux 2.2 running Linux 2.4 kernel (included to match current library load of project web servers)
  - x86-freebsd1: FreeBSD 4.8
  - x86-netbsd1: NetBSD 1.6.1
Compile Farm Hosts

- **AMD 64-bit (Opteron) Architecture:**
  - amd64-linux1: Fedora Core release 3 running Linux 2.6 kernel

- **DEC Alpha (ev67) Architecture:**
  - alpha-linux1: Debian GNU/Linux 3.0 running Linux 2.2 kernel

- **PowerPC Architecture:**
  - ppc-osx1: Apple Mac OS X 10.1 Server with Fink running on an Apple Mac G4
  - ppc-osx2: Apple Mac OS X 10.2 Server with Fink running on an Apple Mac G4

- **Sparc (UltraSPARC-II) Architecture:**
  - sparc-solaris1, sparc-solaris2: Sun Solaris 9, running on two Sun Enterprise 220R systems
General Process

- Transfer source code to the Compile Farm.
- Login to the Compile Farm.
- Access the desired Compile Farm host.
- Build (unpack, configure, compile) the software; research and fix any problems.
- Test the software.
- Build packages.
- Transfer packages from the Compile Farm.
File Transfers

- PuTTY – PSCP

C:\junk>pscp scp -l user -2 -i <key>.ppk <file>
<u>@cf-shell.sf.net:/home/users/<u>/<us>/<user>/<file>

- Files Transferred To Compile Farm Shell
Accessing the Compile Farm

● Open to developers
● SSH
● “Opt in Basis”
  – Check box in Account Maintenance
  – Post a SSH public key
SSH Public/Private Key Generation

% ssh-keygen -t dsa -C "username@shell.sf.net"
Generating public/private dsa key pair.
Enter file in which to save the key (/home/username/.ssh/id_dsa):
Created directory '/home/username/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/username/.ssh/id_dsa.
Your public key has been saved in /home/username/.ssh/id_dsa.pub.
The key fingerprint is:

-t can be : rsa1 (SSH1/RSA), dsa (SSH2/DSA) and rsa (SSH2/RSA)
Accessing the Compile Farm

- cf.sourceforge.net
  - Compile Farm menu server
  - Select specific server to access

- cf-shell.sourceforge.net
  - Check file transfers
  - Ssh to compile hosts

- No direct access to compile farm hosts except through these
Development Tools

- Available on Most
  - GNU C Compiler
  - GNU Assembler
  - GNU Make
  - X11 Forwarding

- Available on a Few
  - Python
  - Perl
  - Java 2 JDK

- locate & which
Disk Quota

- Soft Quota – 256 Mb
- Hard Quota – 512 Mb
- If soft quota exceeded for more than 7 days, data is purged
Features

- Automated Repetitive Builds
  - Cron
- Test Portability
- Supports Popular Packaging
  - RPM, DEB, DMG and PKG, BSD, etc.
Demonstration & Questions