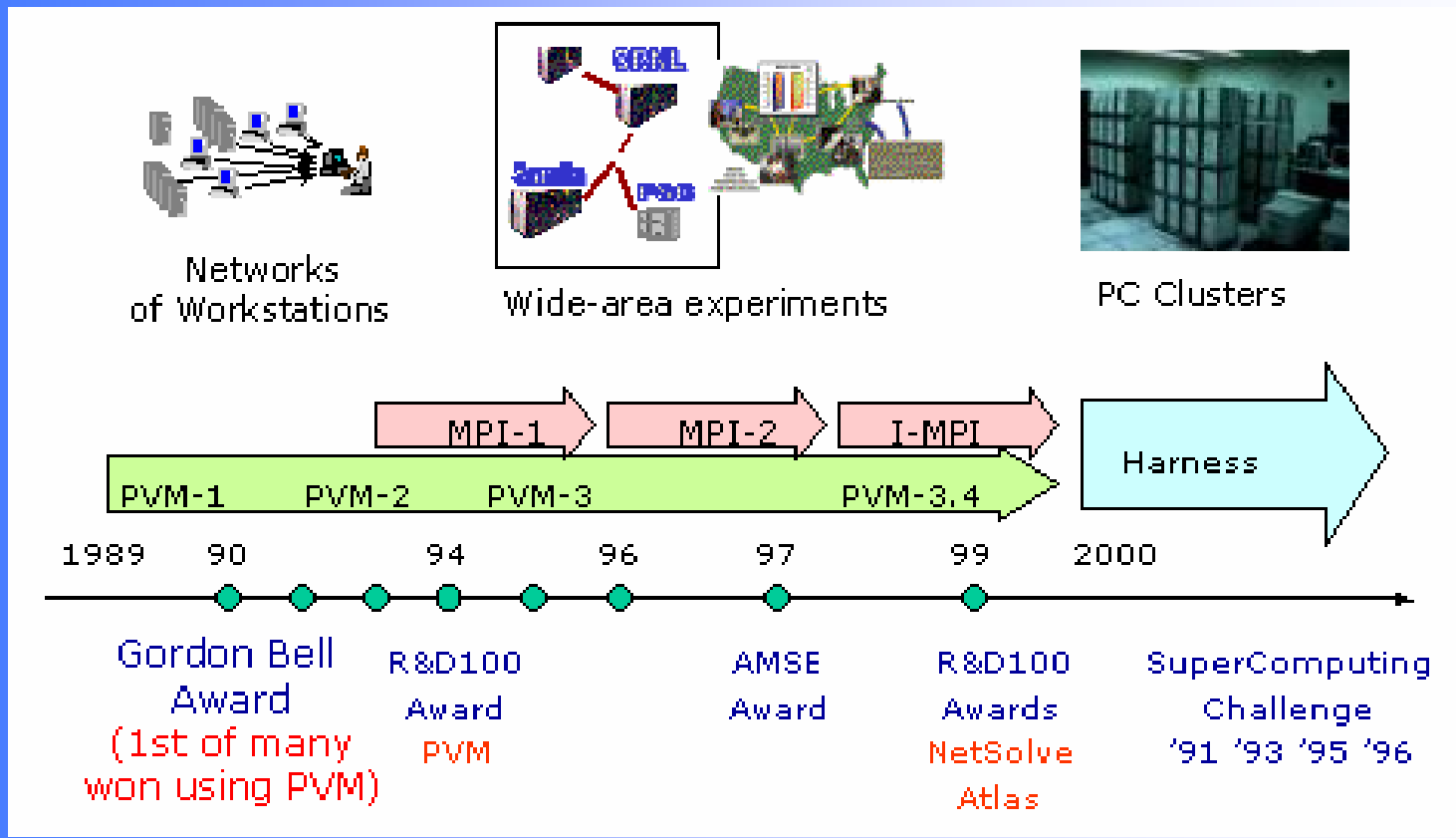
OAK RIDGE National Laboratory

Virtualized Environments for the Harness Workbench

Presentation Overview

- What is Harness
- Project objectives
- Basic virtualisation approach
- System design
- File integration
  - Copy
  - Link
  - Unionfs
- Benchmarks
- Future Work

What is the Harness Workbench?



What is the Harness Workbench?

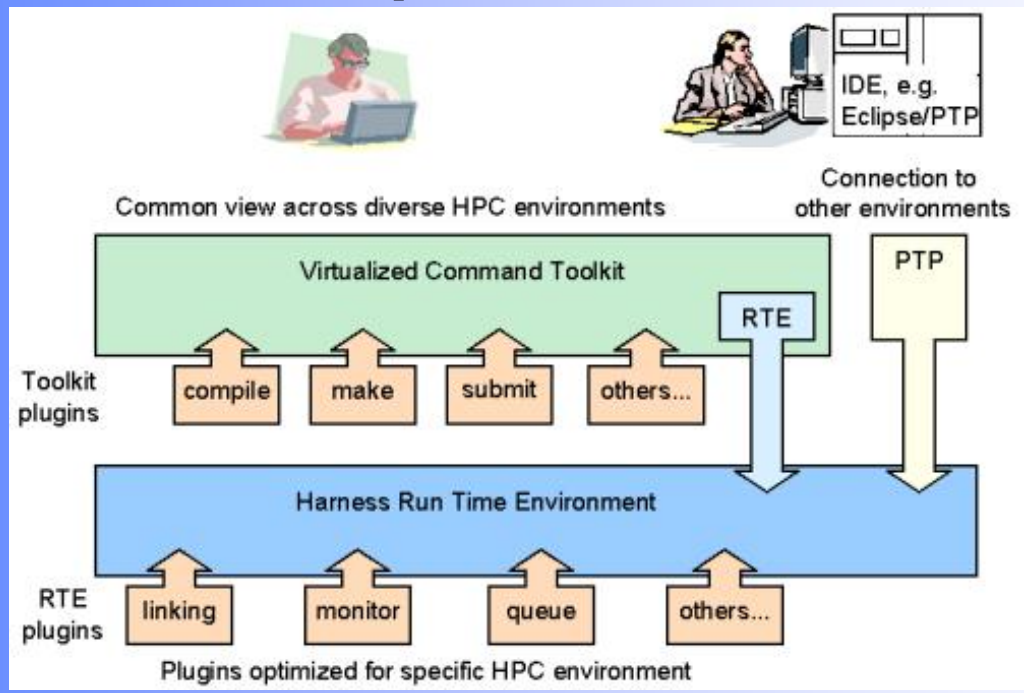
Harness is more than the successor of PVM/MPI:

- Parallel plug-in interface
- Distributed peer-to-peer control
- Multiple distributed virtual machines

Project Objectives

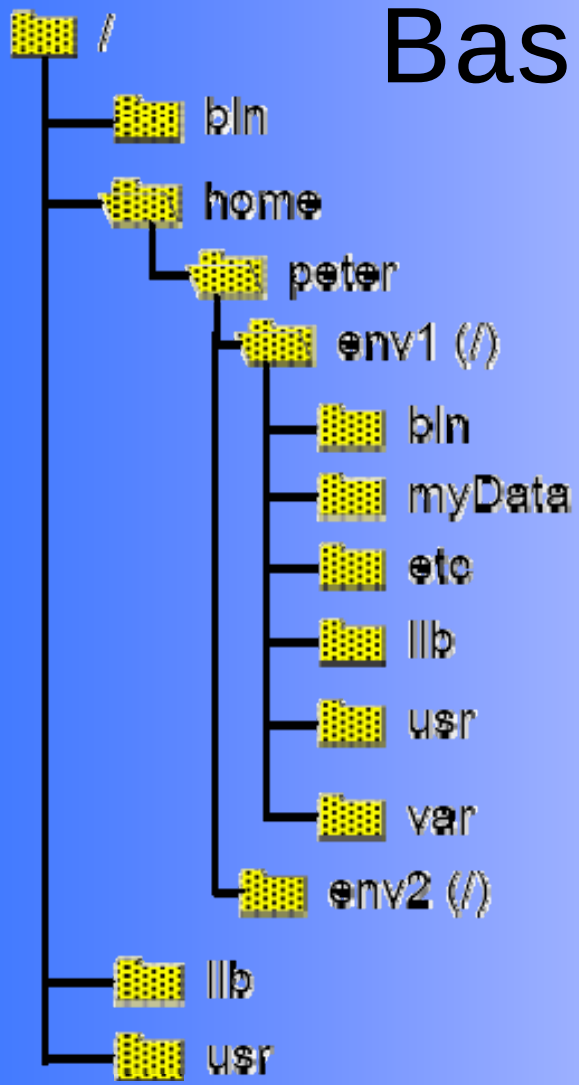
- Simplify software development and deployment by making software portable
- Environment Description Concept
- Tool for creating virtualized environments (VE) on different platforms
- Tool for starting application in VE

Harness Workbench Components



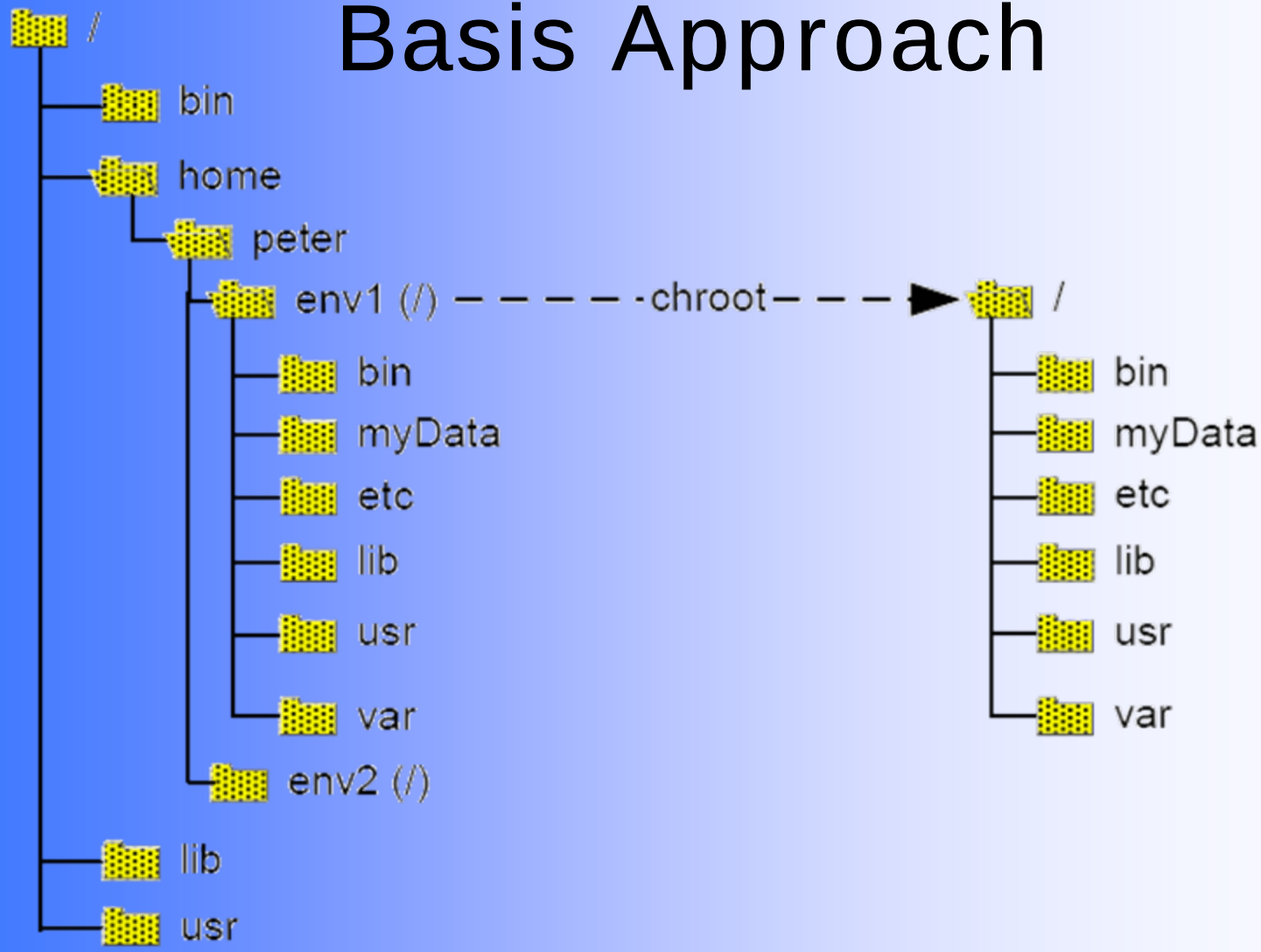
vct –env install env.conf
vct –env start env.conf myApp

Basis Approach

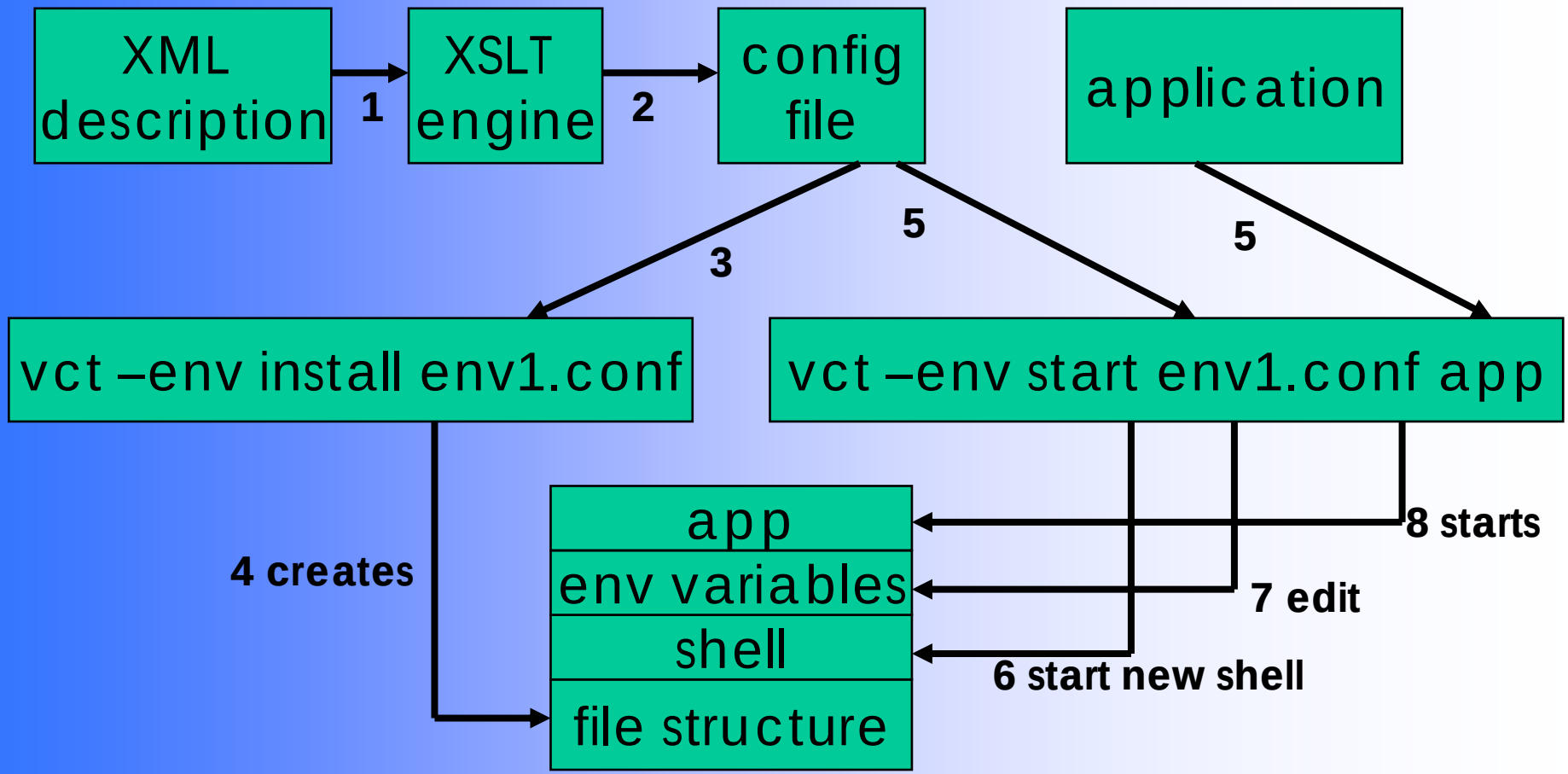


- **chroot /home/peter/env1 bash**
- **chroot** requires super user rights
- Program which uses **chroot** is protected and can only be used with **sudo**
- **sudo** can be configured to enable normal users to execute the program with super user rights

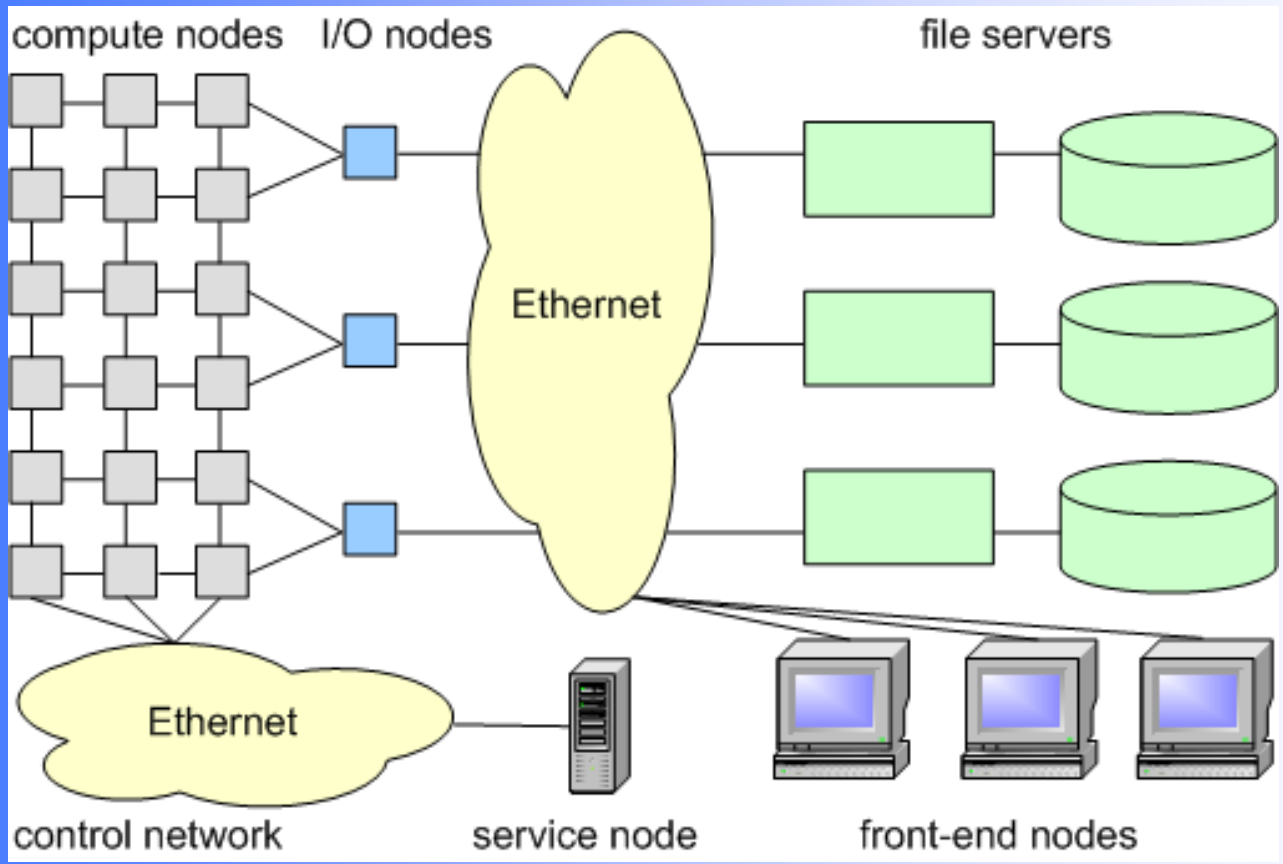
Basis Approach



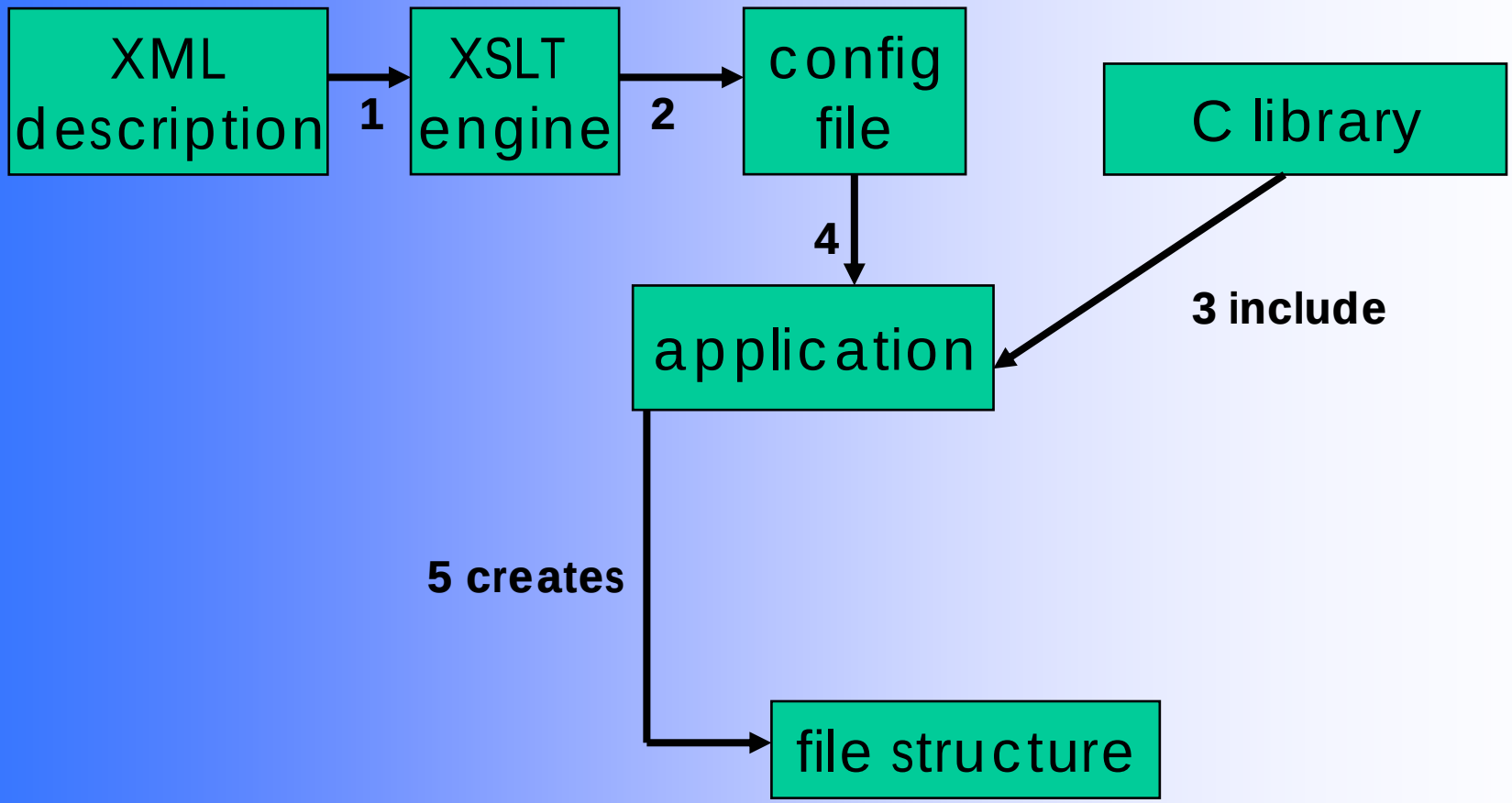
Design


XML description
1
XSLT engine
2
config file
application
3
5
5
vct –env install env1.conf
vct –env start env1.conf app
app
env variables
shell
file structure
4 creates
8 starts
7 edit
6 start new shell

Modern HPC Architecture



Design


XML description
1
XSLT engine
2
config file
C library
4
3 include
application
5 creates
file structure

Abstract Approach

- Is focused on file structures and environment variables
- Currently no concentration on package management and services

Detailed Design

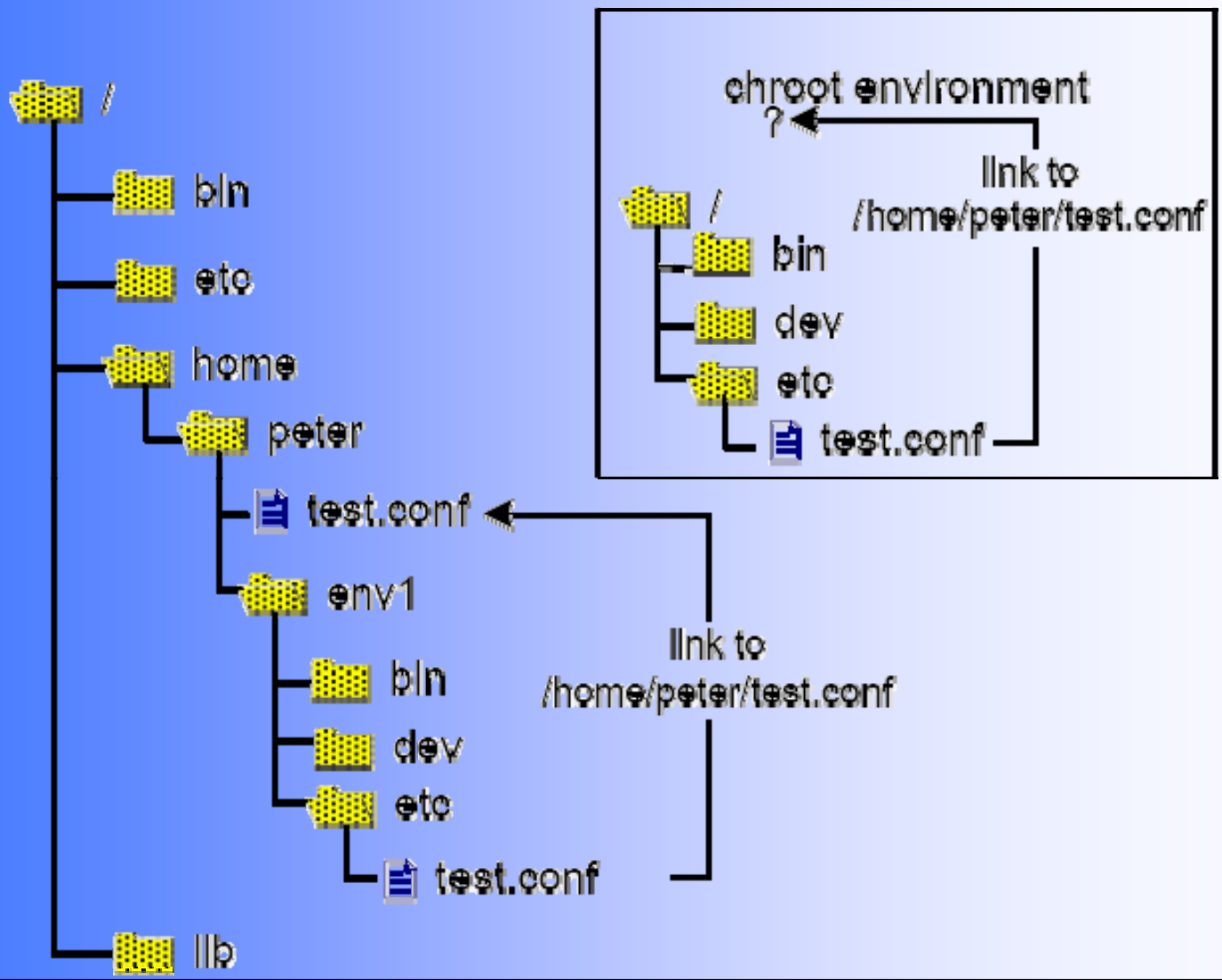
- File Integration
  - Copy
  - Symbolic link
  - Unionfs
- Environment Variables

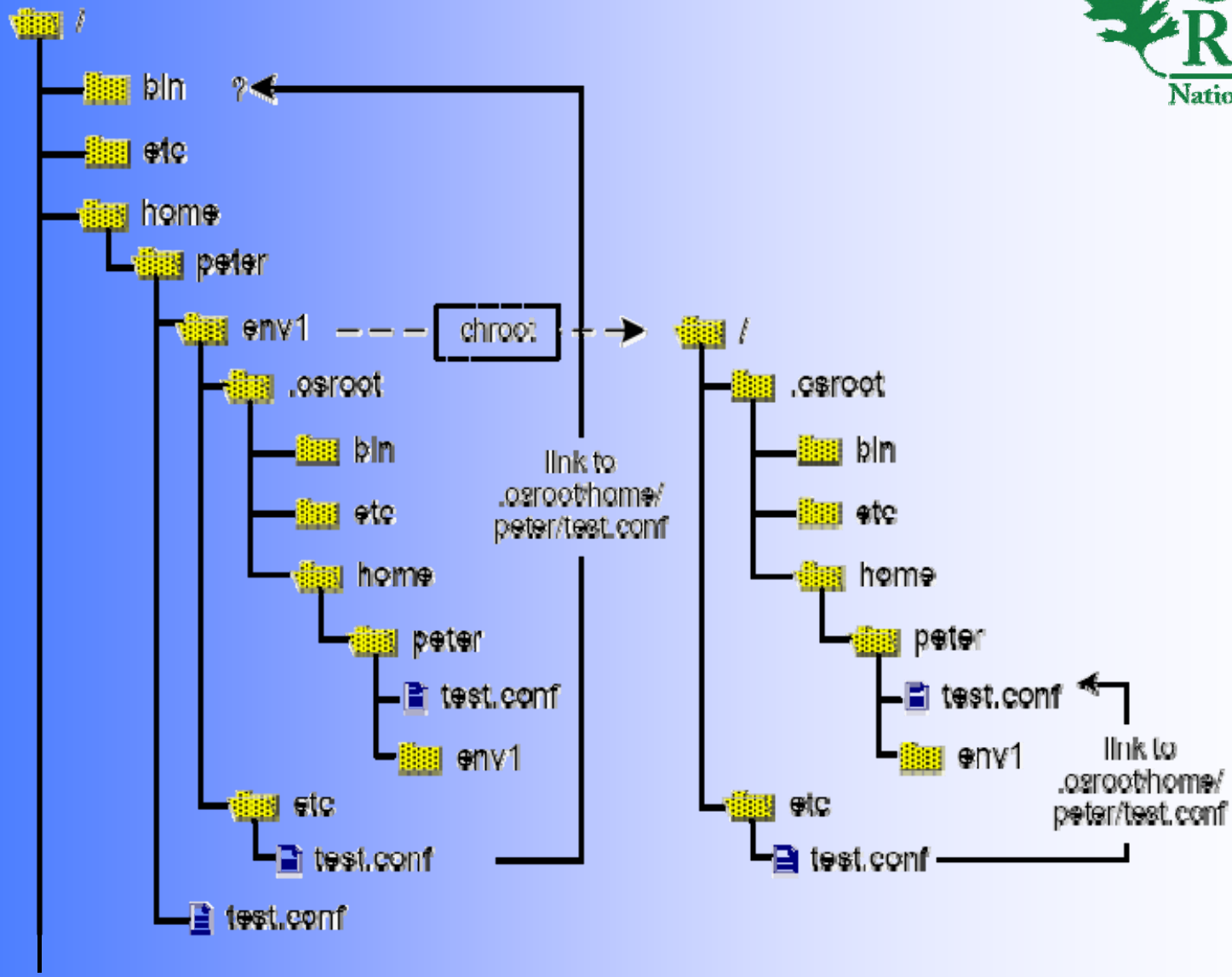
Copy

- File permission can be changed
- More secure; no influence to the original
- Slow to create
- Fast during runtime
- No dynamic connection

Symbolic Link

- Useful to reuse files
- Is fast to create and fast during runtime
- Dynamic connection
- no change of the file permissions


/
bin
etc
home
peter
test.conf
env1
bin
dev
etc
test.conf
lib
link to
/home/peter/test.conf
chroot environment
?
/
bin
dev
etc
test.conf
link to
/home/peter/test.conf



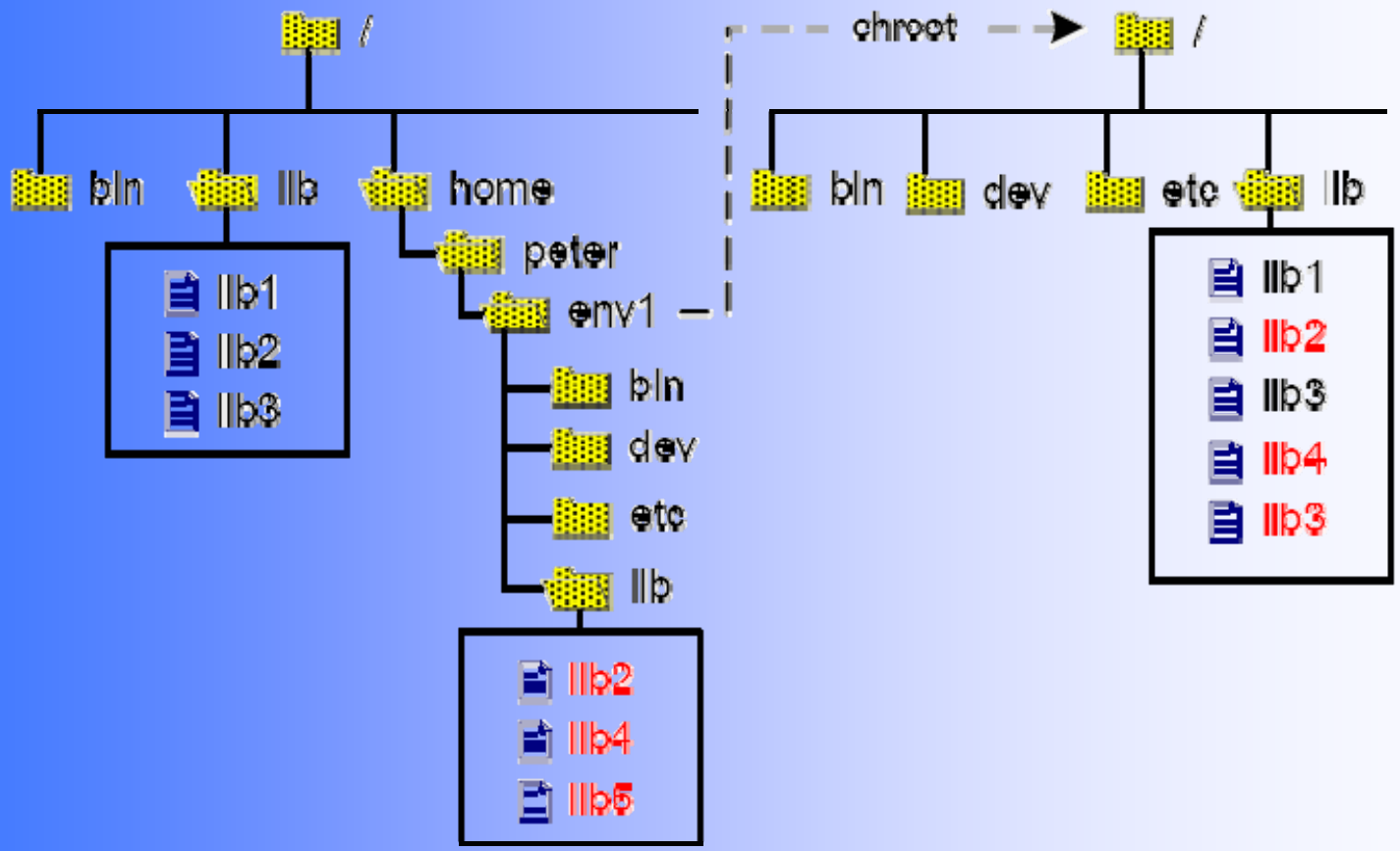
- **mount –bind / /home/peter/env1/.osroot**

Unionfs

- Merges different directories to a new one (union)
- Copy-on-write function
- Can limit access rights
- Hide-on-delete function

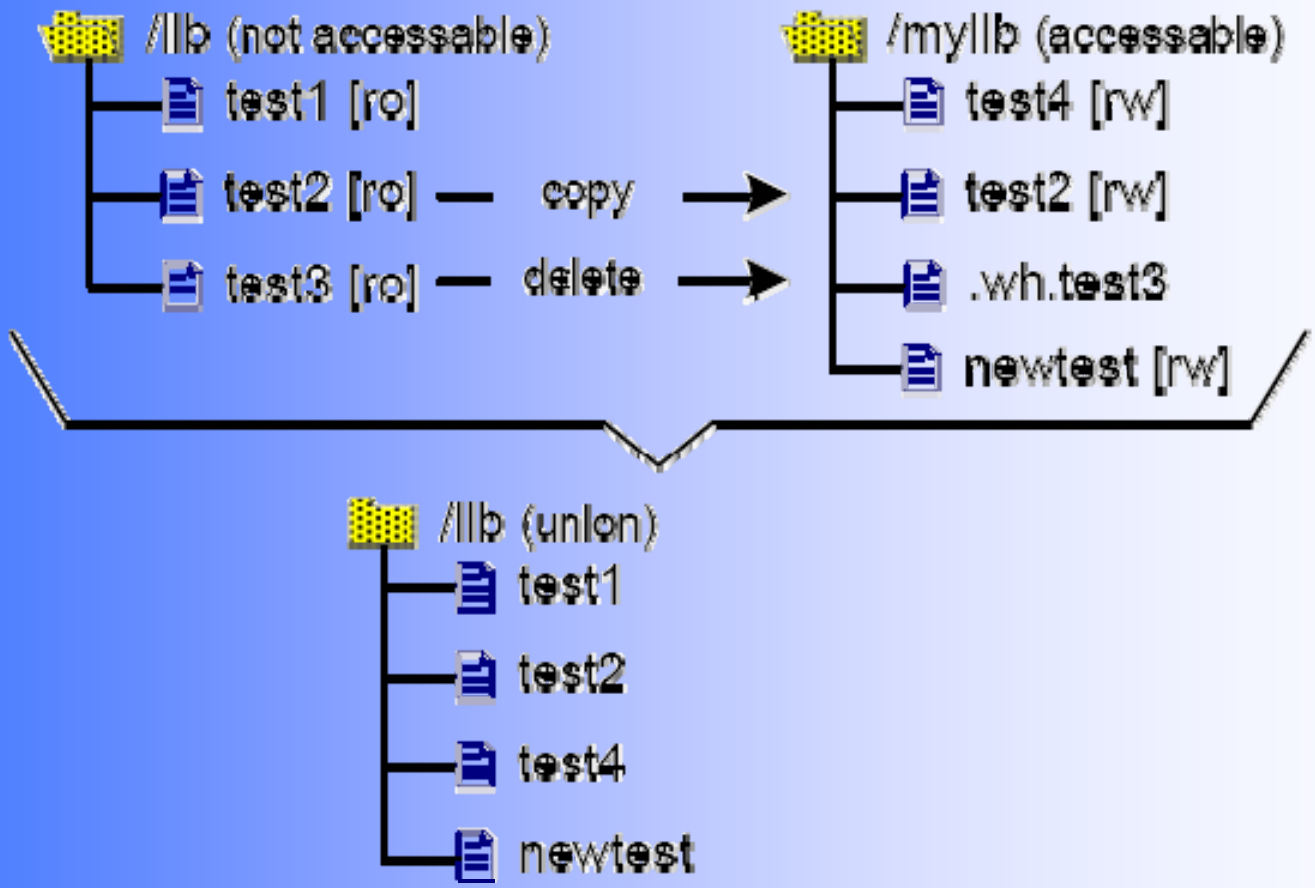
- mount –t unionfs –o dirs=/dir1=rw:dir2=ro unionfs /uniondir

Unionfs

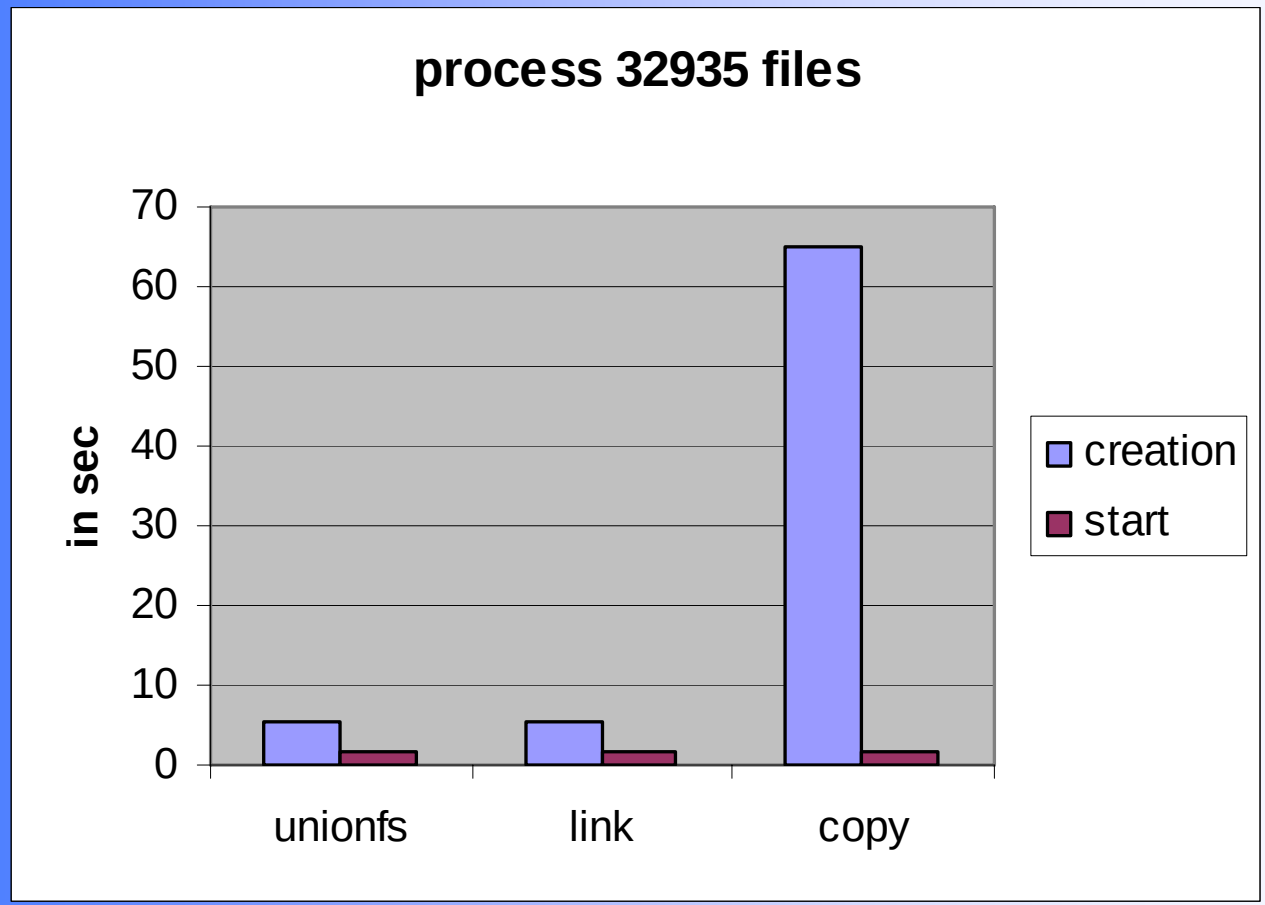

/
bin
lib
home
lib1
lib2
lib3
peter
env1
bin
dev
etc
lib
lib2
lib4
lib5
chroot
/
bin
dev
etc
lib
lib1
lib2
lib3
lib4
lib3


OAK
RIDGE
National Laboratory

Unionfs


/llb (not accessable)
test1 [ro]
test2 [ro]
test3 [ro]
copy
delete
/myllb (accessable)
test4 [rw]
test2 [rw]
.wh.test3
newtest [rw]
/llb (unlon)
test1
test2
test4
newtest

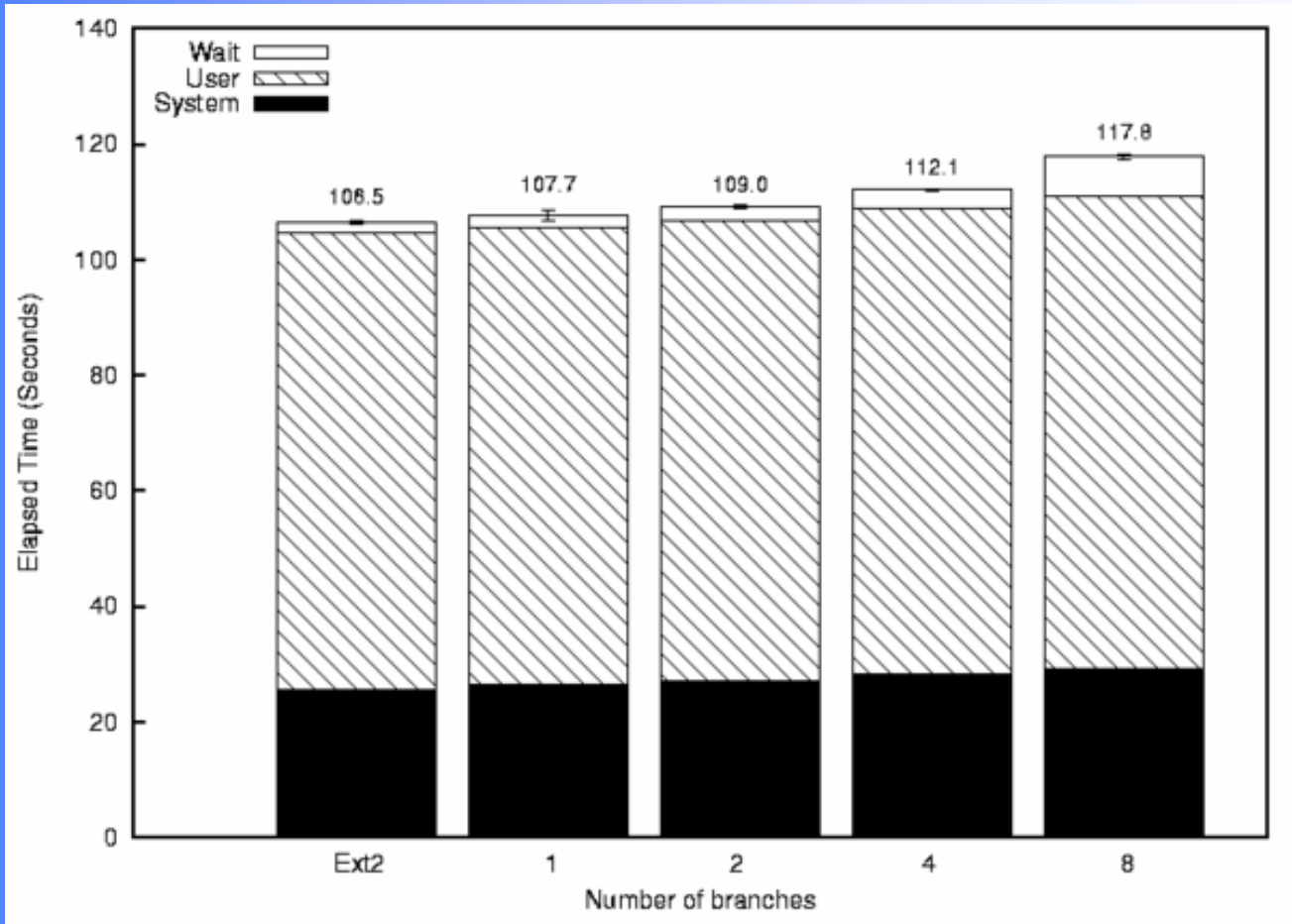
VE Creation Performance



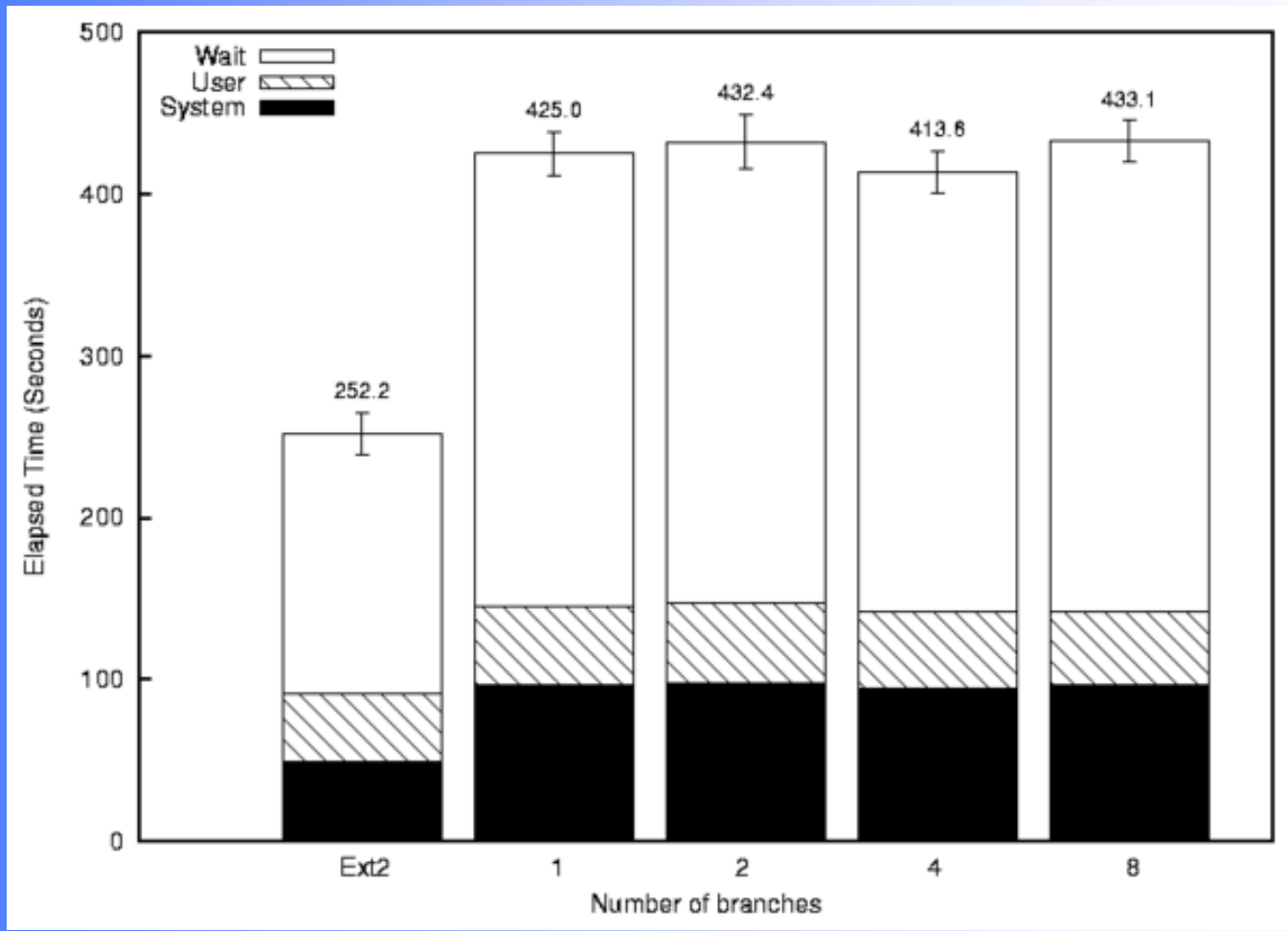
OpenSSH Compile Benchmark

- Is a CPU-intensive benchmark
- Represents a workload characteristic of users
- The highest-priority branch was read-write, the other were read-only
- Overhead is 0,99% up to 10,7 % over Ext2

OpenSSH Compile Benchmark



I/O-intensive Postmark Test



Conclusion

- SymLinks, Copy and Unionfs can be used together to build virtualized environments
- The entire virtualisation approach is capable to be implemented without security vulnerabilities
- The approach is very portable
- Shell scripts are too slowly and inconvenient to program

Future Work

- Consider virtualisation tools like Xen
- Include package management and services
- Tools for half-automated environment description

Thanks for Your Attention!