

Cray Programming Environment (CPE) Containerization

Ever J. Dominguez R.; Almond J. Heil | ejdominguez5@miners.utep.edu; almondheil@gmail.com | HPC-DO, Supercomputer Institute



High Performance Computing

IMPORTANCE AND USE AT LANL

At LANL, every scientist workload requires a heavy number of computing resources. At HPC, we must provide:

- Reliability
- Maintainability
- Reproducibility

All these allow for scientists to continue their research.

CRAY PROGRAMMING ENVIRONMENT (CPE)

Provided by Hewlett Packard Enterprise (HPE)

Contains a common set of software tools and libraries.

Some types of tools involve:

- **Compilers**
- **Math libraries**
- **Analysis & optimization tools**

Reproducibility & Maintainability

KEEPING EVERYTHING RUNNING

How do we maintain user applications in newer syst?

- **Containerization**
- **Squashfile iso mounting**

POTENTIAL FOR REUSABILITY

Our approach of a Docker mount for the programming environment allows for:

- Protected environment from any issues that may be caused.
- Efficient and quick recovery from combability issues
- Easy translation from platform that work under the Cray management suite.



“The Cray Programming Environment is like an onion. It has layers, and you cry when opening it.”
~ We made it up

Relocating the CPE

VERSIONING IT

We want to version the CPE; thus we must **relocate** it. But moving it causes **broken absolute paths**

PLAINTEXT

Easy fix: **Finding and replacing** simple text in files.

SYMLINKS

When changing locations, symbolic links (**symlinks**) will break due to non-existing locations.

- **Erase** broken symlinks.
- **Use new location** of previous links.

EXECUTABLE & LINK FORMAT (ELF)

ELF files develop broken library search paths

- Cannot just find and replace, would corrupt.
- We can use **patchelf**, which rebuilds the file

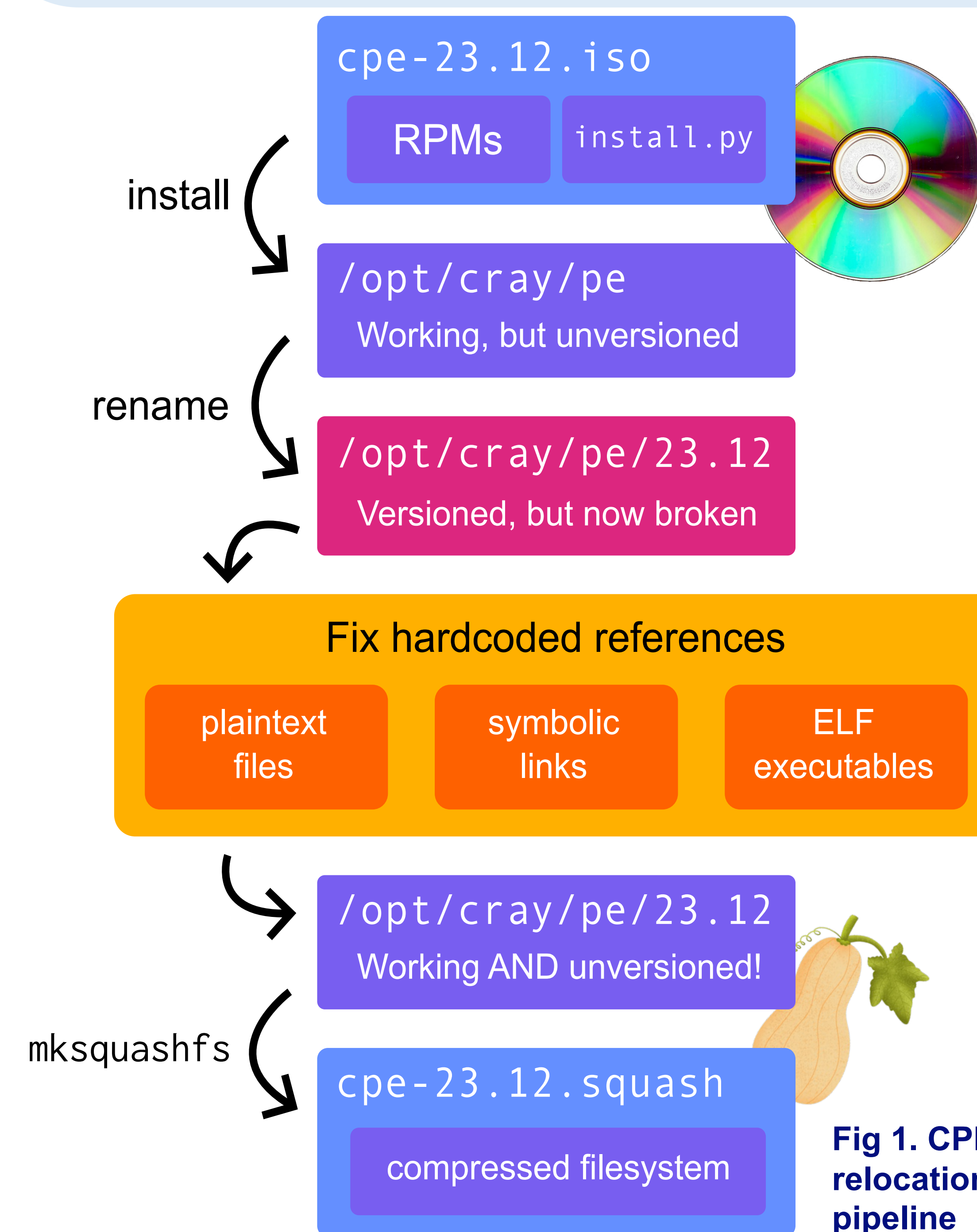


Fig 1. CPE relocation pipeline

```
# absolute symlink is broken
/opt/cray/pe/23.12
├── lib64
│   └── example.so
└── link -> /opt/cray/pe/lib64/example.so

# making it relative fixes it!
/opt/cray/pe/23.12
├── lib64
│   └── example.so
└── link -> lib64/example.so
```

Fig 2. Fixing a broken symlink

RPATH / RUNPATH

Dynamic library loading

Can be changed with patchelf

Other strings

Unclear what they're for

File corrupts if changed

Fig 3. Strings in an ELF file

Conclusion

WHAT WE ACCOMPLISHED

In this project, we've been able to:

- **Relocate CPE files** to versioned locations, making different versions of the CPE installable side-by-side
- **Create a CI pipeline** that automates the process

This allows HPC to give better support to scientist workflows:

- Supporting long-running simulations more effectively.
- Adding specific libraries and tools that will complement and facilitate scientist simulation efforts.

FUTURE WORK

Further explore possible limitations to this approach

- Swap between Load CPE-provided modulefiles
- Use multiple CPE versions on one machine
 - What configuration must be changed to switch CPEs?

Does our process break any necessary CPE integrations?

- More **smoke testing**
- Develop unit tests (difficult with CPE version changes)

Is this work portable?

- Test on different operating systems
- Test multiple CPE's

ACKNOWLEDGEMENTS

Thanks to everyone who made this project possible!

- **Project mentors:** Ty Goetsch, Paul Ferrell, Megan Phinney
- **Teaching staff at the SI bootcamp:** Trevor Bautista, Devon Bautista, John Dermer, Sakul Koirala, Shivam Mehta
- **HPC Programs Manager:** Julie Wiens