reUpNix: Reconfigurable and Updateable Embedded Systems

TUHH
Technische
Universität
Hamburg

Niklas Gollenstede

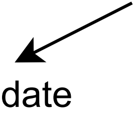
2023-06-18

Embedded: Updates & Reconfiguration

Sending stuff 🛰️ into space 🚀 is expensive 💰

Embedded: Updates & Reconfiguration

Sending stuff 🛰️ into space 🚀 is expensive 💰



update

Embedded: Updates & Reconfiguration

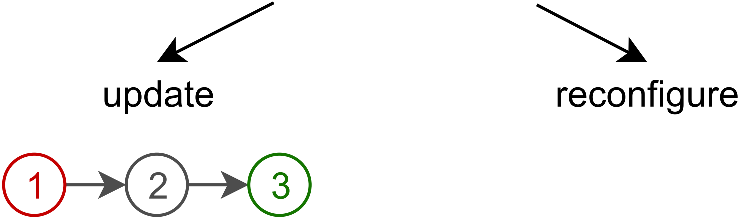
Sending stuff 🛰️ into space 🚀 is expensive 💰

update

reconfigure

Embedded: Updates & Reconfiguration

Sending stuff 🛰️ into space 🚀 is expensive 💰



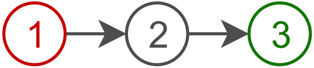
update

reconfigure

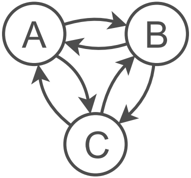
Embedded: Updates & Reconfiguration

Sending stuff 🛰️ into space 🚀 is expensive 💰

update



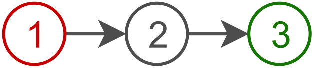
reconfigure



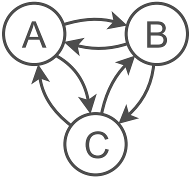
Embedded: Updates & Reconfiguration

Sending stuff 🛰️ into space 🚀 is expensive 💰

update



reconfigure

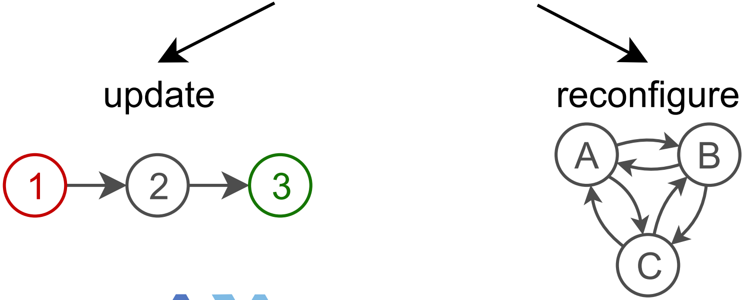


NixOS

Embedded: Updates & Reconfiguration

TUHH

Sending stuff 🛰️ into space 🚀 is expensive 💰



NixOS :

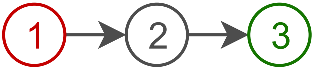
~~embedded~~

Embedded: Updates & Reconfiguration

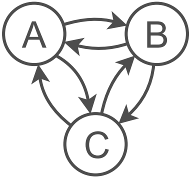
TUHH

Sending stuff 🛰️ into space 🚀 is expensive 💰

update



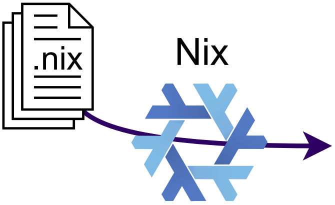
reconfigure



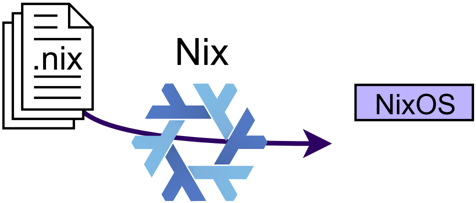
NixOS : 🖥️ ~~embedded~~ why?

NixOS & reUpNix

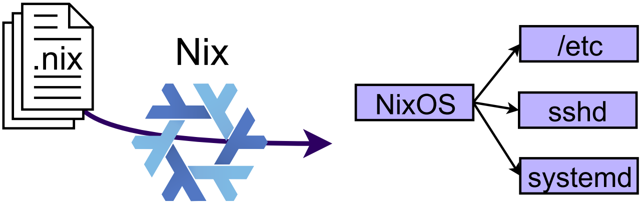
NixOS & reUpNix



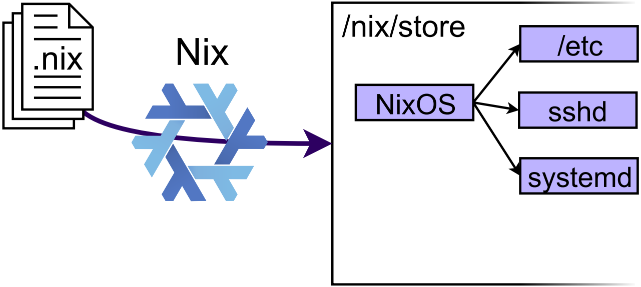
NixOS & reUpNix



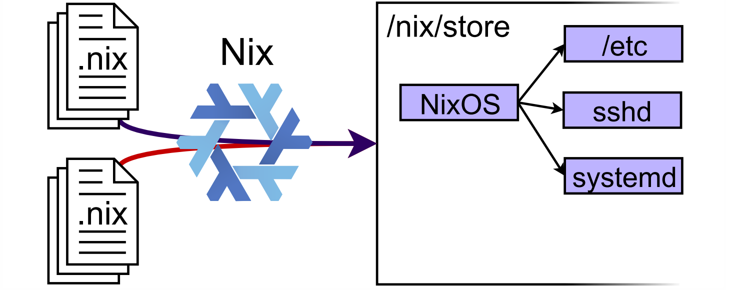
NixOS & reUpNix



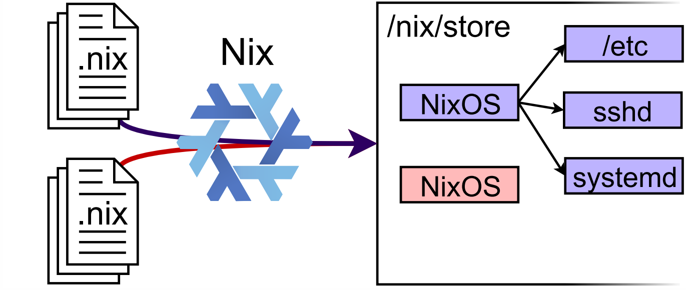
NixOS & reUpNix



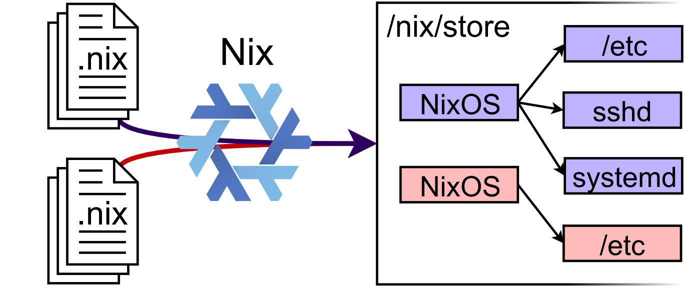
NixOS & reUpNix



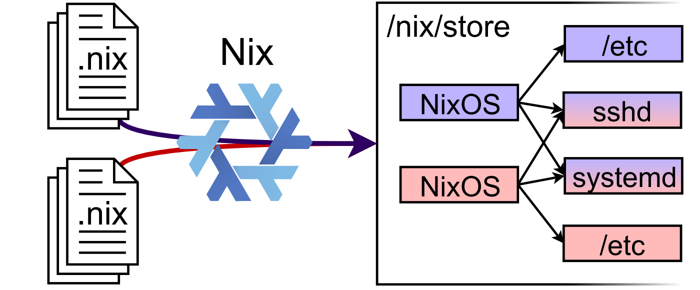
NixOS & reUpNix



NixOS & reUpNix

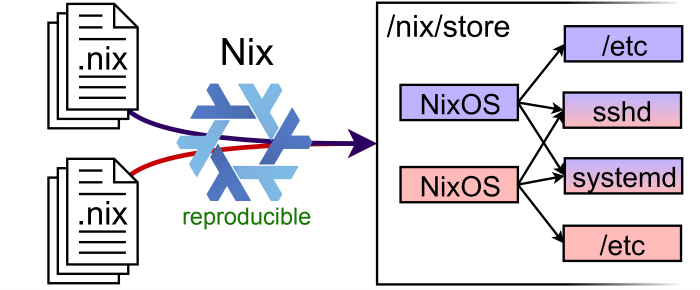


NixOS & reUpNix

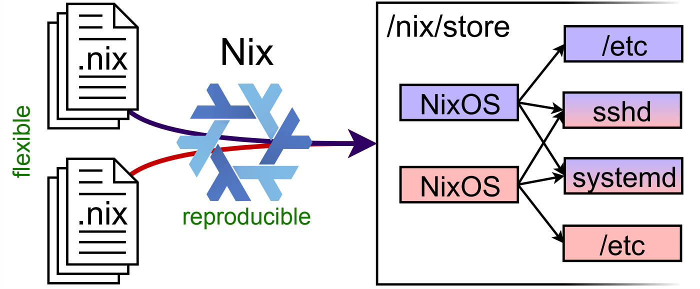


NixOS & reUpNix

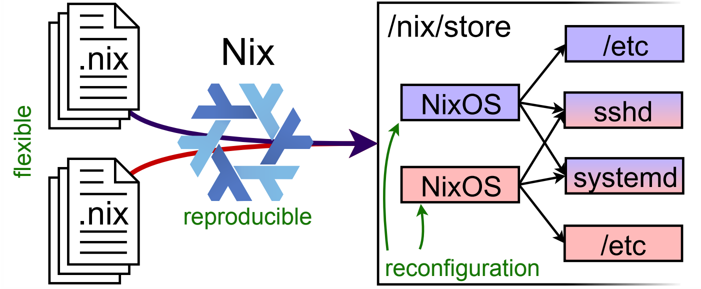
TUHH



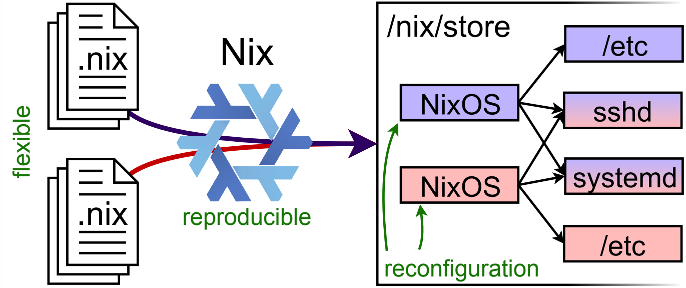
NixOS & reUpNix



NixOS & reUpNix

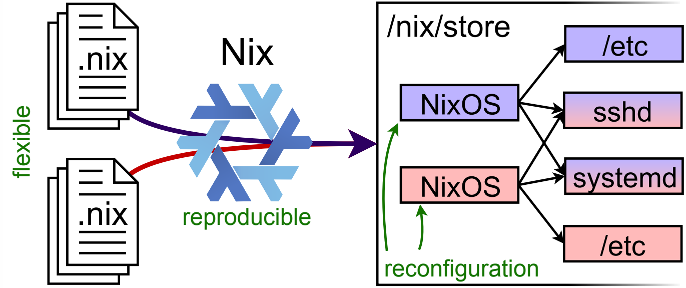


NixOS & reUpNix



NixOS ❌

NixOS & reUpNix

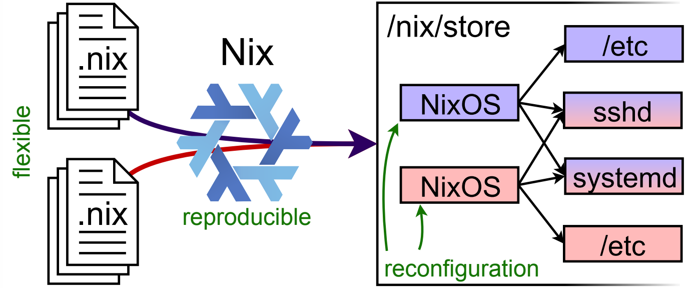


NixOS ✗
Large base system

⇒

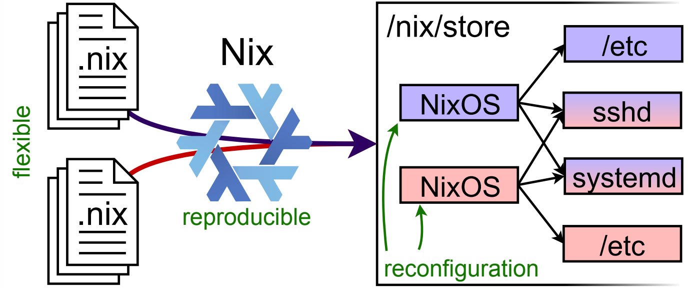
reUpNix contribution
Shrink (~1.1 GiB → 155 MiB)

NixOS & reUpNix



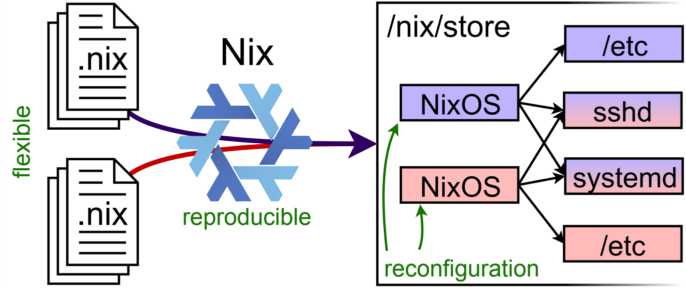
NixOS ✗		**reUpNix contribution**
Large base system	⇒	Shrink (~1.1 GiB → 155 MiB)
OCI containers inefficient	⇒	Remove Docker daemon & deduplicate files across images

NixOS & reUpNix



NixOS ✗		**reUpNix contribution**
Large base system	⇒	Shrink (~1.1 GiB → 155 MiB)
OCI containers inefficient	⇒	Remove Docker daemon & deduplicate files across images
Large system updates	⇒	Differential update transfer

NixOS & reUpNix



NixOS ❌ ⇒ **reUpNix contribution**

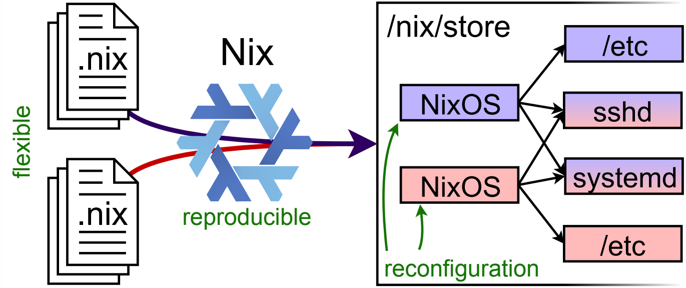
Large base system ⇒ Shrink (~1.1 GiB → 155 MiB)

OCI containers inefficient ⇒ Remove Docker daemon & deduplicate files across images

Large system updates ⇒ Differential update transfer

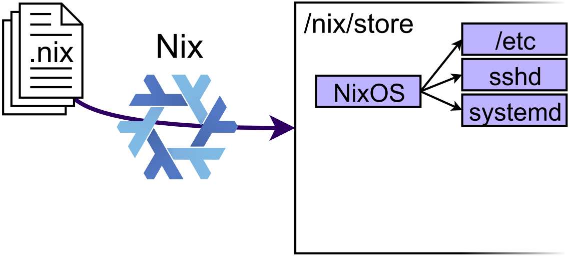
Bootloader update fragile ⇒ Make update atomic & Integrate reconfiguration

NixOS & reUpNix

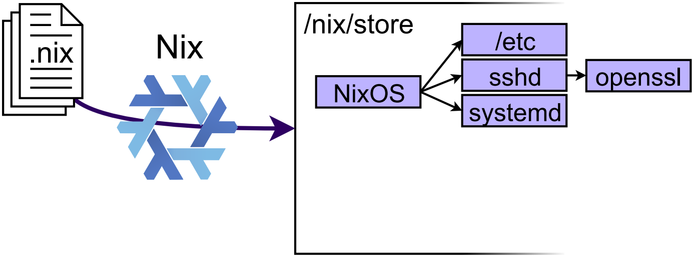


NixOS ❌		**reUpNix contribution**
Large base system	⇒	Shrink (~1.1 GiB → 155 MiB)
OCI containers inefficient	⇒	Remove Docker daemon & deduplicate files across images
Large system updates	⇒	Differential update transfer
Bootloader update fragile	⇒	Make update atomic & Integrate reconfiguration

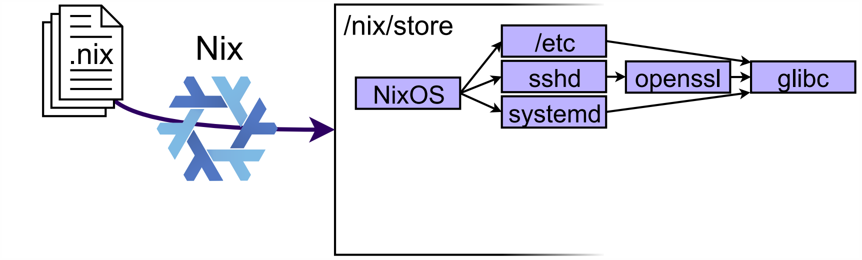
Differential Update Transfer



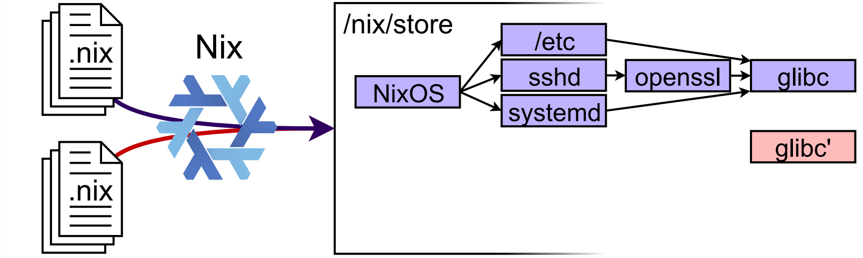
Differential Update Transfer



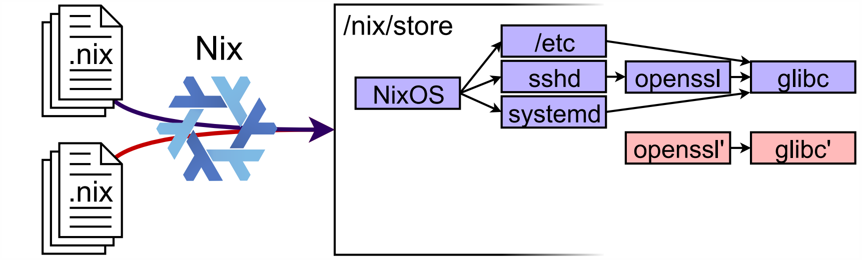
Differential Update Transfer



Differential Update Transfer

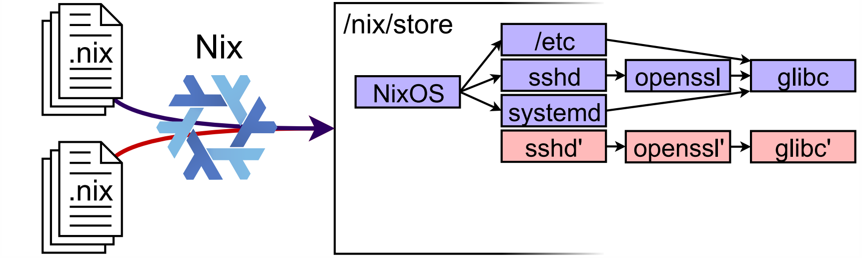


Differential Update Transfer

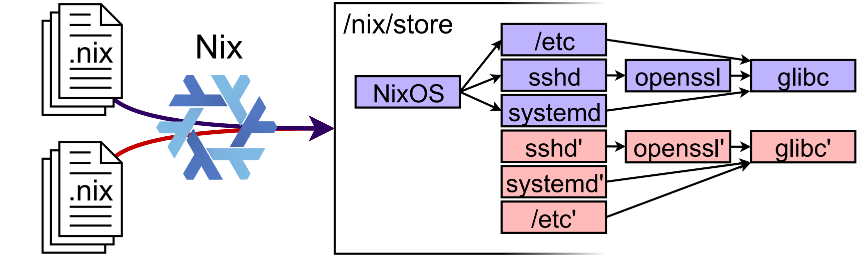


Differential Update Transfer

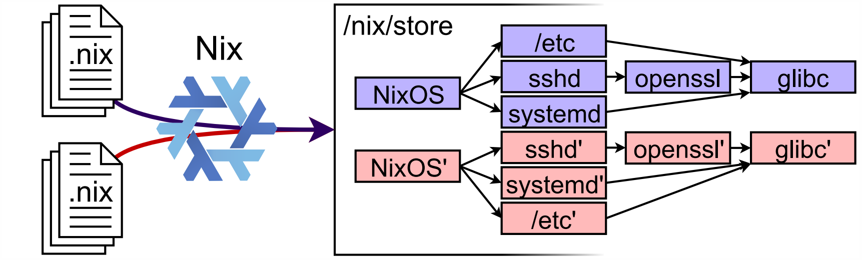
TUHH



Differential Update Transfer

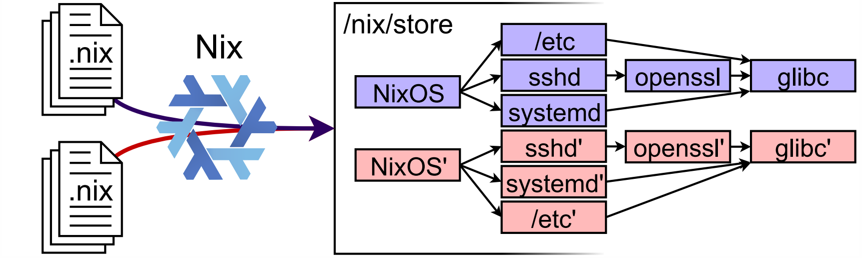


Differential Update Transfer



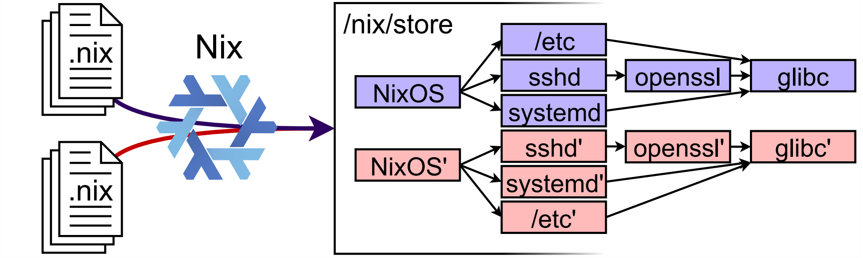
Differential Update Transfer

TUHH



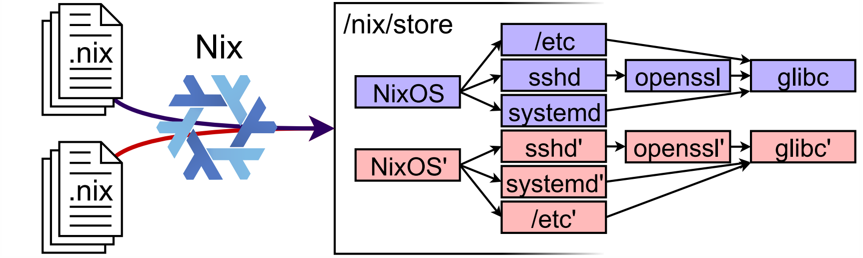
- Many/all new components, but very similar component exists

Differential Update Transfer



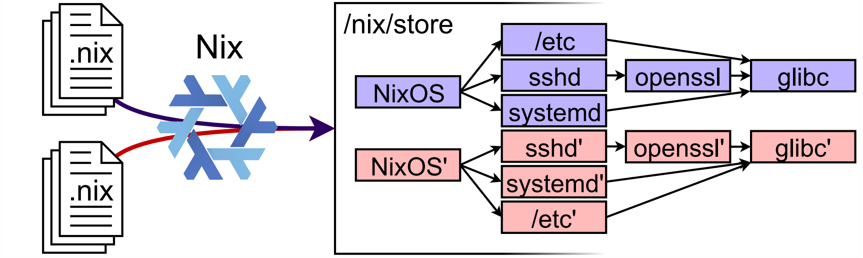
- Many/all new components, but very similar component exists
- Find “before” component, update differentially

Differential Update Transfer



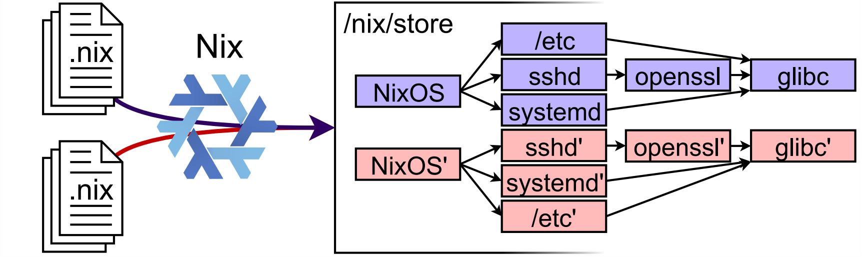
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage

Differential Update Transfer



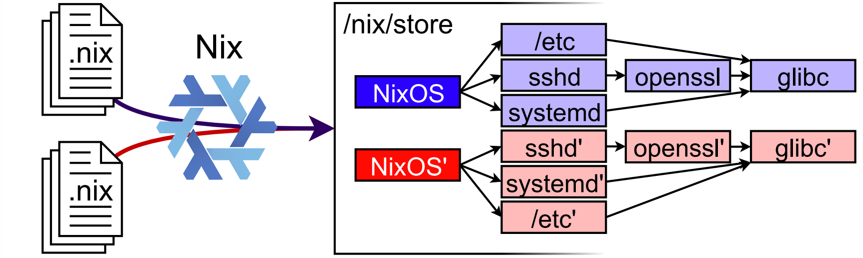
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs

Differential Update Transfer



- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer:

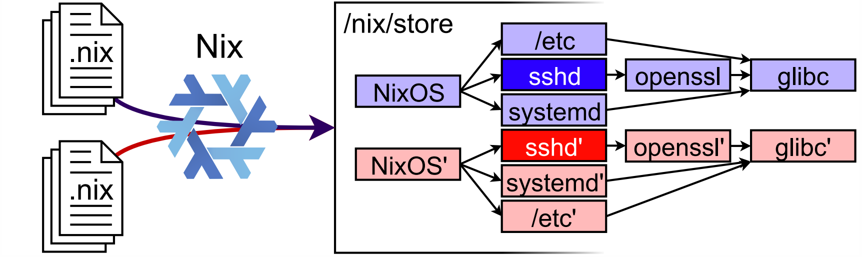
Differential Update Transfer



- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’),

Differential Update Transfer

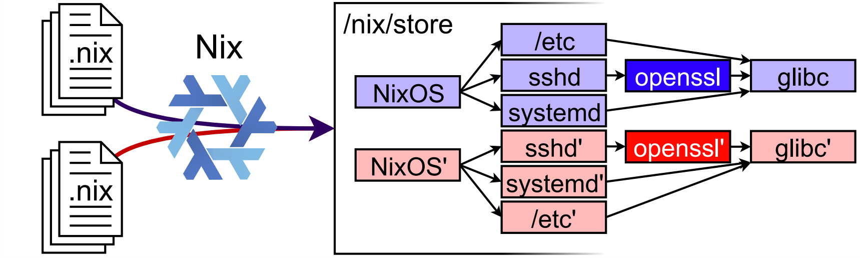
TUHH



- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’),

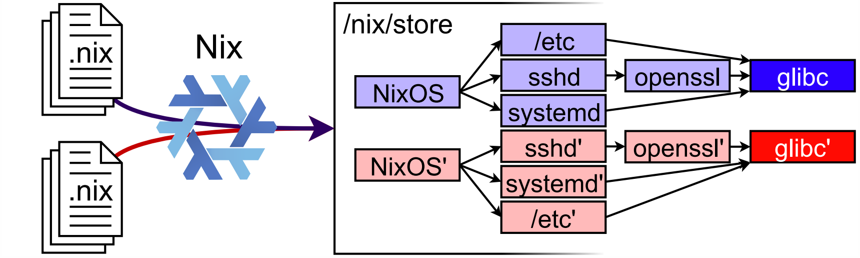
Differential Update Transfer

TUHH



- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’), diff(openssl, openssl’),

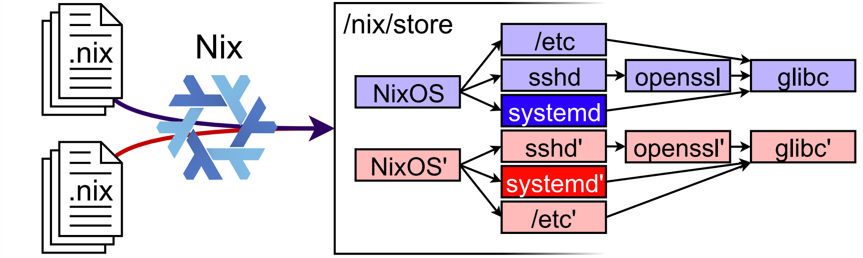
Differential Update Transfer



- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’), diff(openssl, openssl’), diff(glibc, glibc’),

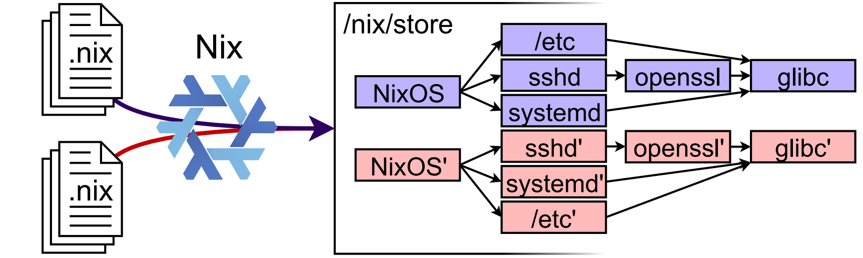
Differential Update Transfer

TUHH



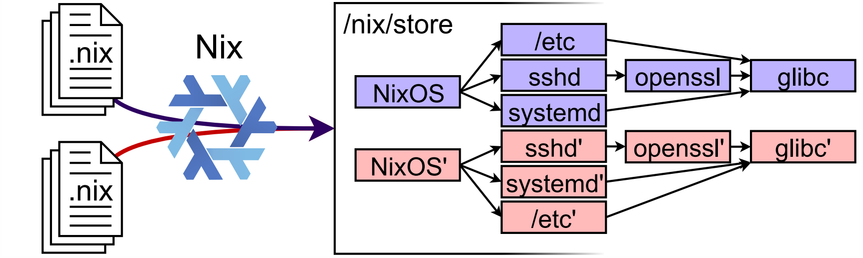
- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’), diff(openssl, openssl’), diff(glibc, glibc’), diff(systemd, systemd’),

Differential Update Transfer



- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’), diff(openssl, openssl’), diff(glibc, glibc’), diff(systemd, systemd’), ...

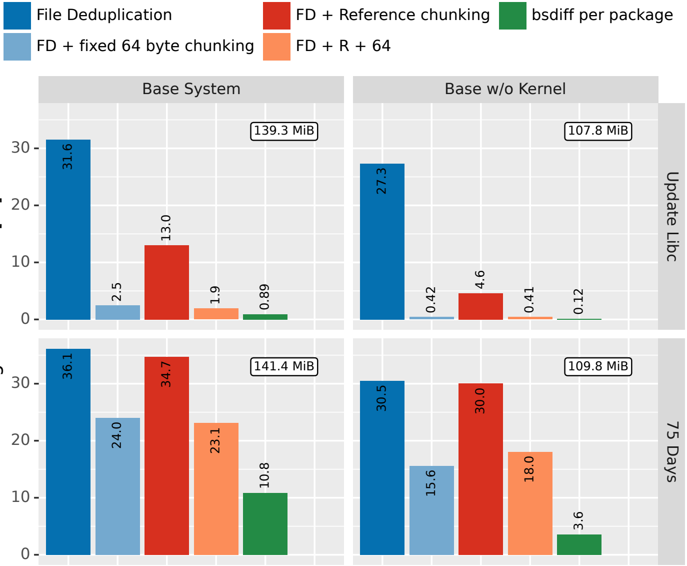
Differential Update Transfer



- Many/all new components, but very similar component exists
- Find “before” component, update differentially
- No clear package name/ID, or lineage
  ⇒ walk pair of dependency graphs
- Transfer: diff(NixOS, NixOS’), diff(sshd, sshd’), diff(openssl, openssl’), diff(glibc, glibc’), diff(systemd, systemd’), ...
- “diff(C, C’)” by sender (knows C & C’) ⇒ unidirectional transfer

Differential Update Transfer: Results

diff(C, C’) =

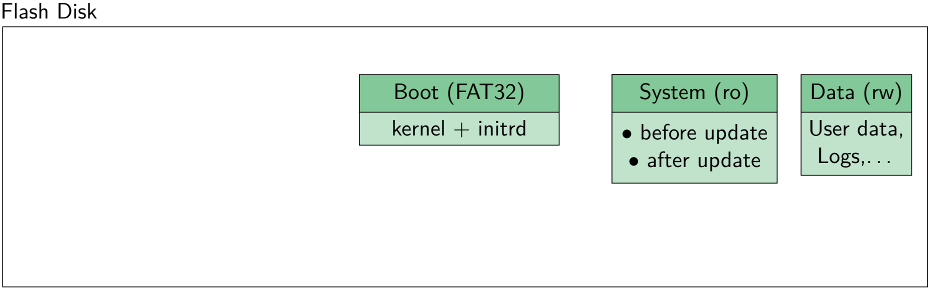


Atomic Bootloader Updates

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware

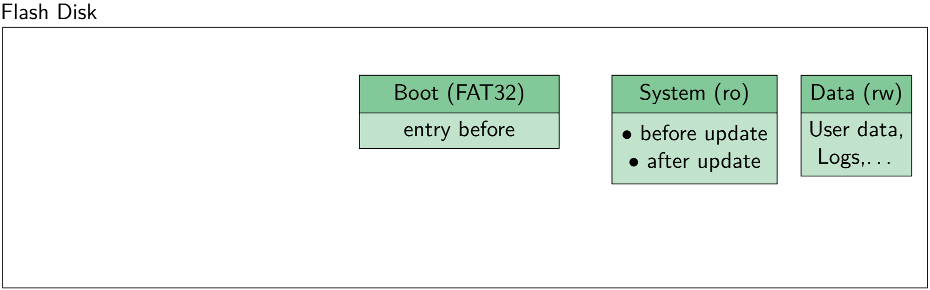
Atomic Bootloader Updates

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility



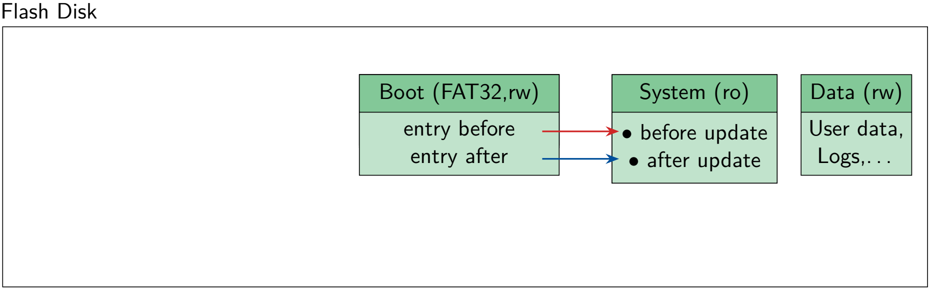
Atomic Bootloader Updates

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility



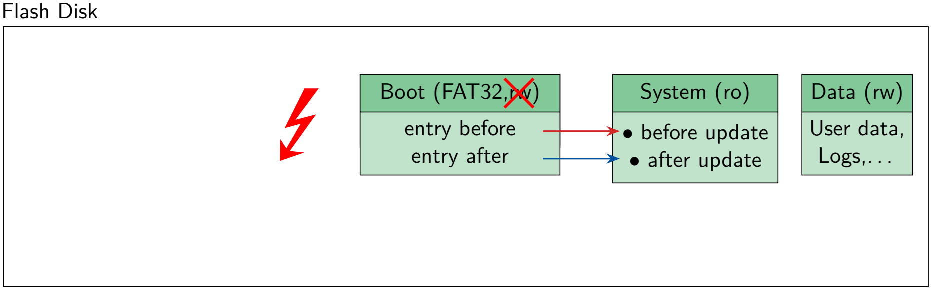
Atomic Bootloader Updates

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility



Atomic Bootloader Updates

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable



Flash Disk

Boot (FAT32,~~rw~~)

entry before

entry after

System (ro)

- before update
- after update

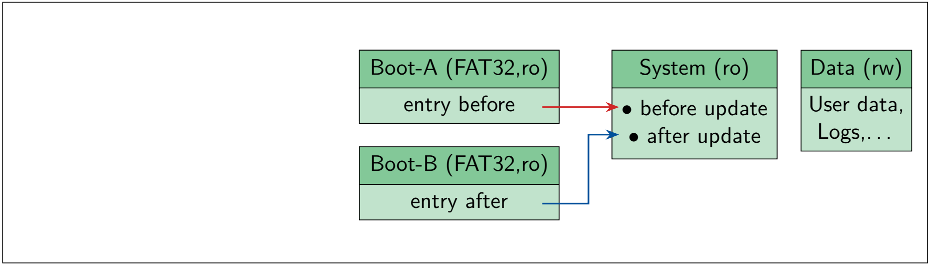
Data (rw)

User data, Logs,. . .

Atomic Bootloader Updates

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable
- Solution: A/B partitioning for Boot

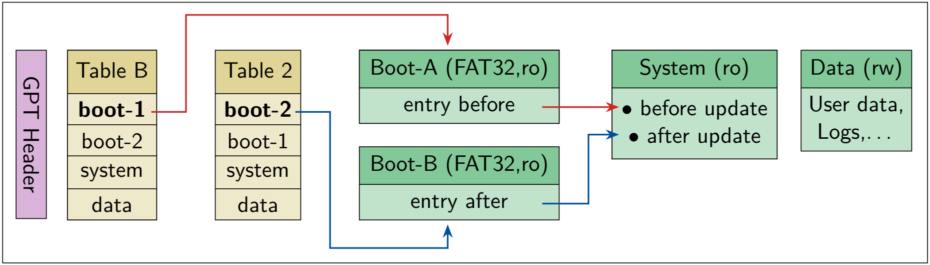
Flash Disk



Atomic Bootloader Updates

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable
- Solution: A/B partitioning for Boot

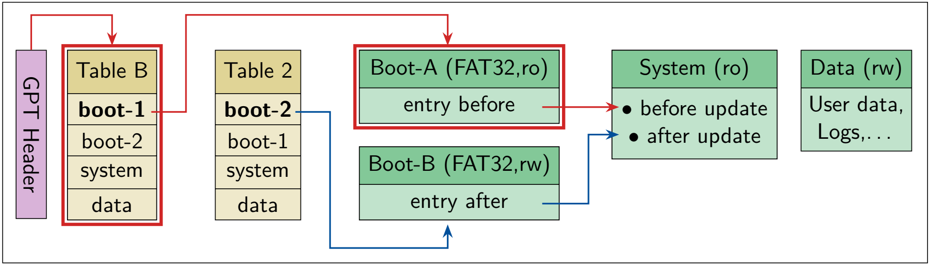
Flash Disk



Atomic Bootloader Updates

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable
- Solution: A/B partitioning for Boot switched via GPT

Flash Disk

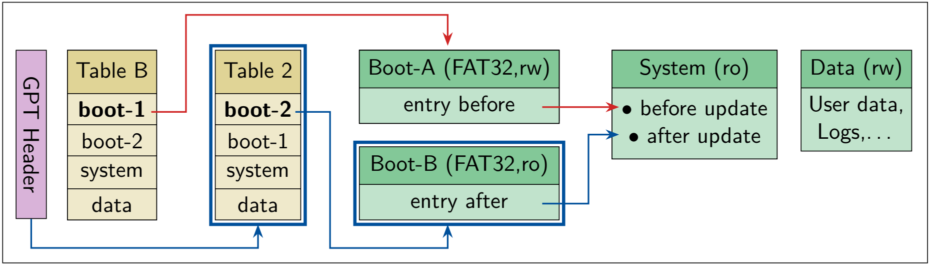


Atomic Bootloader Updates

TUHH

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable
- Solution: A/B partitioning for Boot switched via GPT

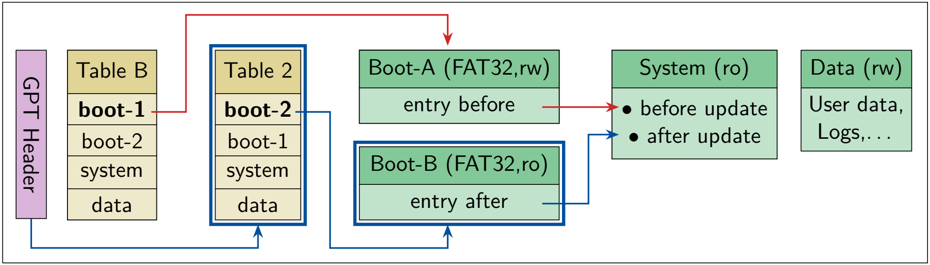
Flash Disk



Atomic Bootloader Updates

- **Goal**: safe/atomic updates for NixOS on any hardware/firmware
- System: multiple systems on one reliable FS (e.g., journaling)
- Boot: FAT32 for firmware compatibility but unreliable
- Solution: A/B partitioning for Boot switched via GPT

Flash Disk



⇒ applicable to other operating systems!

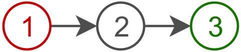
Reconfiguration

- Change system’s executed software configuration

Reconfiguration

- Change system's executed software configuration

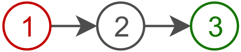
Update:



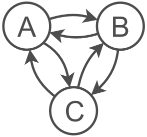
Reconfiguration

- Change system's executed software configuration

Update:



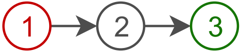
Reconfigure:



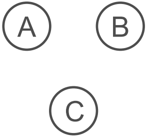
Reconfiguration

- Change system’s executed software configuration

Update:



Reconfigure:

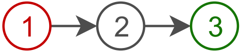


Reconfiguration

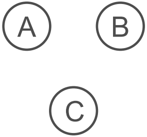
TUHH

- Change system's executed software configuration

Update:



Reconfigure:

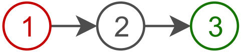


- NixOS: multiple variants ("profiles")

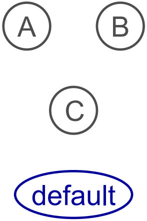
Reconfiguration

- Change system’s executed software configuration

Update:



Reconfigure:

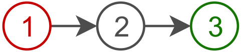


- NixOS: multiple variants (“profiles”)

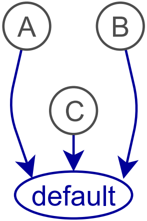
Reconfiguration

- Change system’s executed software configuration

Update:



Reconfigure:

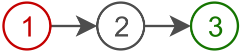


- NixOS: multiple variants (“profiles”)

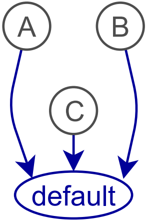
Reconfiguration

- Change system’s executed software configuration

Update:



Reconfigure:

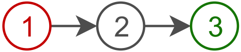


- NixOS: multiple variants (“profiles”)
- reUpNix: reconfig ⇒ reboot

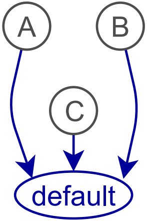
Reconfiguration

- Change system's executed software configuration

Update:



Reconfigure:

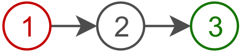


- NixOS: multiple variants ("profiles")
- reUpNix: reconfig ⇒ reboot
  - Correctness: independent of previous profile

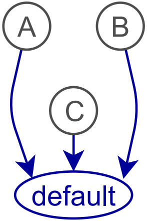
Reconfiguration

- Change system’s executed software configuration

Update:



Reconfigure:

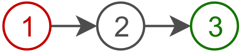


- NixOS: multiple variants (“profiles”)
- reUpNix: reconfig ⇒ reboot
  - Correctness: independent of previous profile
  - Completeness: e.g. change kernel

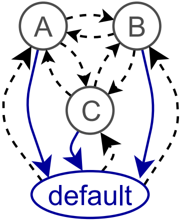
Reconfiguration

- Change system’s executed software configuration

Update:



Reconfigure:

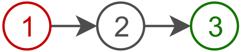


- NixOS: multiple variants (“profiles”)
- reUpNix: reconfig ⇒ reboot
  - Correctness: independent of previous profile
  - Completeness: e.g. change kernel

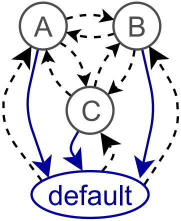
Reconfiguration

- Change system's executed software configuration

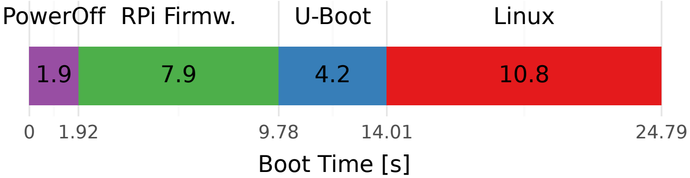
Update:



Reconfigure:



- NixOS: multiple variants ("profiles")
- reUpNix: reconfig ⇒ reboot
  - Correctness: independent of previous profile
  - Completeness: e.g. change kernel

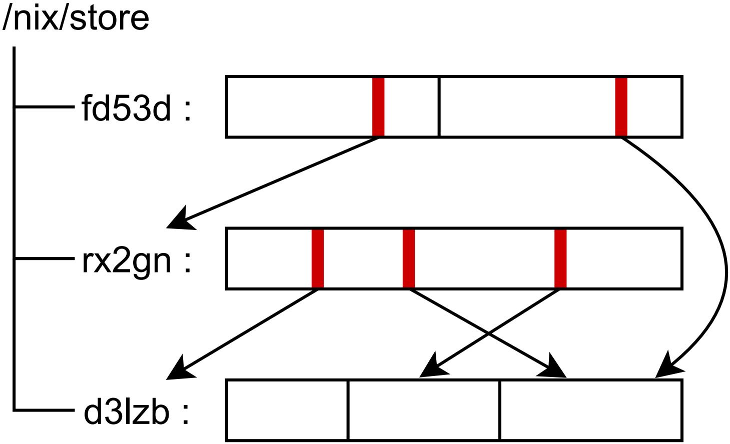


Summary

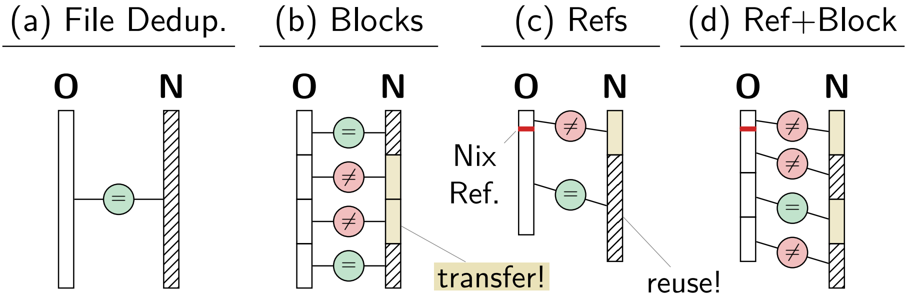
- NixOS: flexible, reproducible, predictable, reconfigurable
  - Not suited for embedded
- Solved NixOS problems for embedded
  - Reduced system size (~1.1 GiB → 155 MiB)
  - Updates up to 99.88 % smaller (mitigate change amplification)
  - Atomic bootloader config update (switch A/B part via GPT)
- reUpNix key features
  - Reconfiguration / multi-system setup
  - Efficient OCI services (w/o Docker, file-deduplicated)
  - Unidirectional communication update transfers

The End

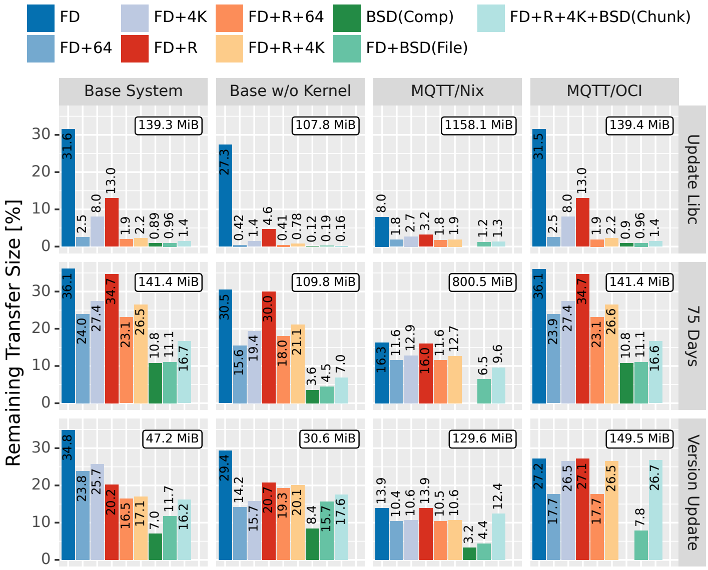
Nix Store Reference Pinning



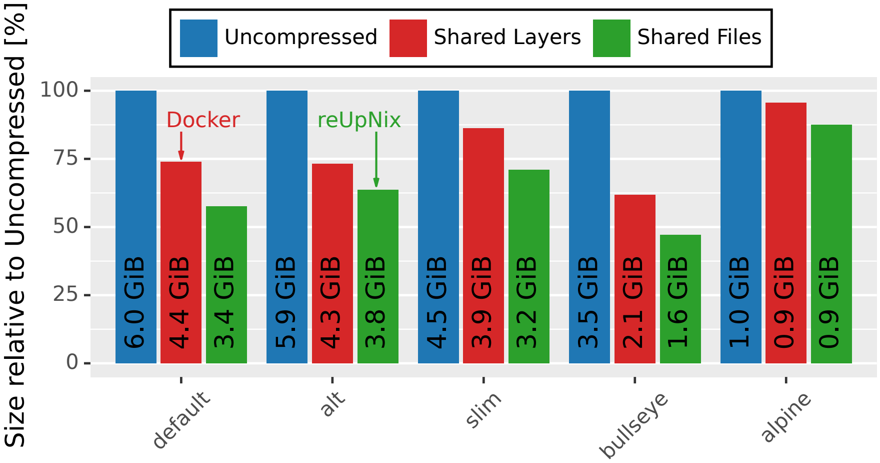
Update Deduplication



Update Transfer Size



OCI Container Storage Cost


Uncompressed
Shared Layers
Shared Files
Size relative to Uncompressed [%]
Docker
reUpNix
6.0 GiB
4.4 GiB
3.4 GiB
5.9 GiB
4.3 GiB
3.8 GiB
4.5 GiB
3.9 GiB
3.2 GiB
3.5 GiB
2.1 GiB
1.6 GiB
1.0 GiB
0.9 GiB
0.9 GiB
default
alt
slim
bullseye
alpine