Repositories for Partition Cloning

OS Independent Software Maintenance
in Large Clusters of PCs

Felix Rauch

Christian Kurmann, Thomas M. Stricker

Laboratory for Computer Systems, ETH Zürich

CoPs Project: http://www.cs.inf.ethz.ch/CoPs/

Eidgenössische Technische Hochschule Zürich

1. December 2000

Clusters of PCs

- Scientific computing (simulations / protein folding)
- Enterprise computing (distrib. databases / datamining)
- Corporate computing (multimedia / collaborative work)
- Education and training (classrooms)

Problem Software Maintenance

- Need guarantee of functioning software environment on all nodes

  Often software MTBF is inacceptable due to "software rot"
- Administration of large numbers of nodes (efficiency)
- Administration of different OS and different versions (flexibility)

Our Approach

- Work with whole partitions as abstraction units (partition = large amount of bits on hard disk)
- Use cloning for replication
- Clean abstraction of software state in cluster node
  - No file system dependency
  - No operating system dependency
  - No dependency on registry databases and shared configuration files
- Maintenance system based on snapshots

This approach needs an efficient storage system

Overview

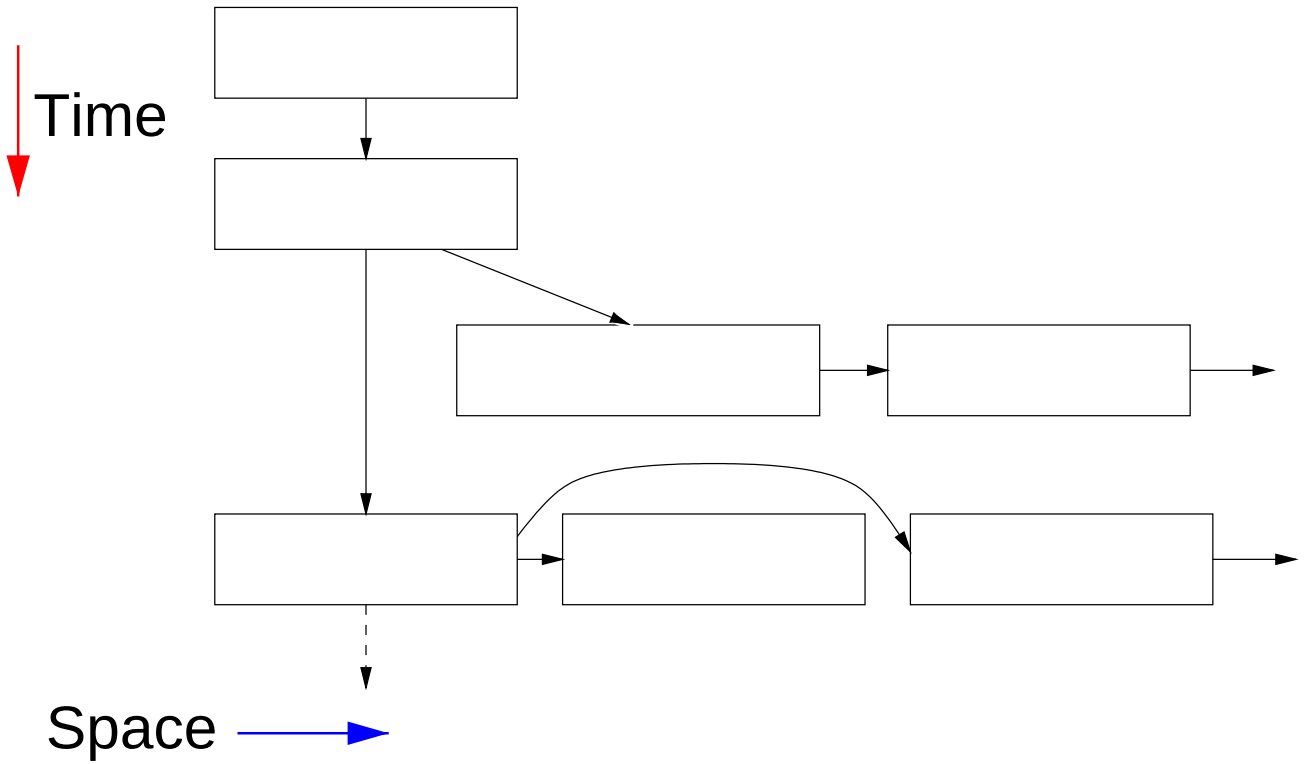
- Characteristics of software installations
- Related work
- Partition Repositories
- Evaluation
- Conclusions

Evolution and Replication of Software Installations

Software installations change continually.

Abstraction along two axes:

- Temporal
- Spacial



Temporal Evolution of a Software Installation

- Installation of new software
- Upgrading software packages
- Patches
- Temporary installations

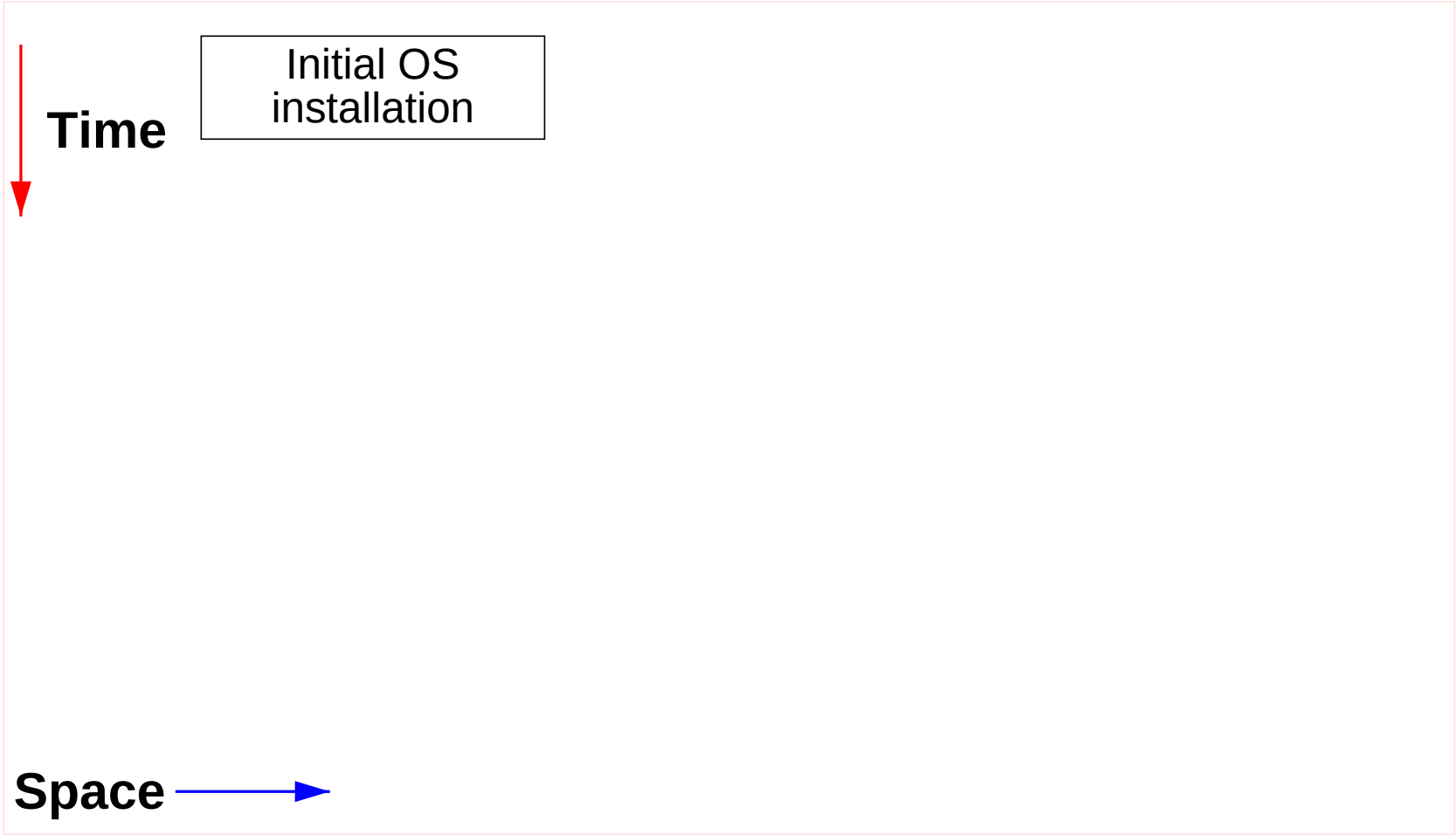
➪ Snapshot after every change to recover from failing installations or deinstallations

Spacial Evolution of a Software Installation

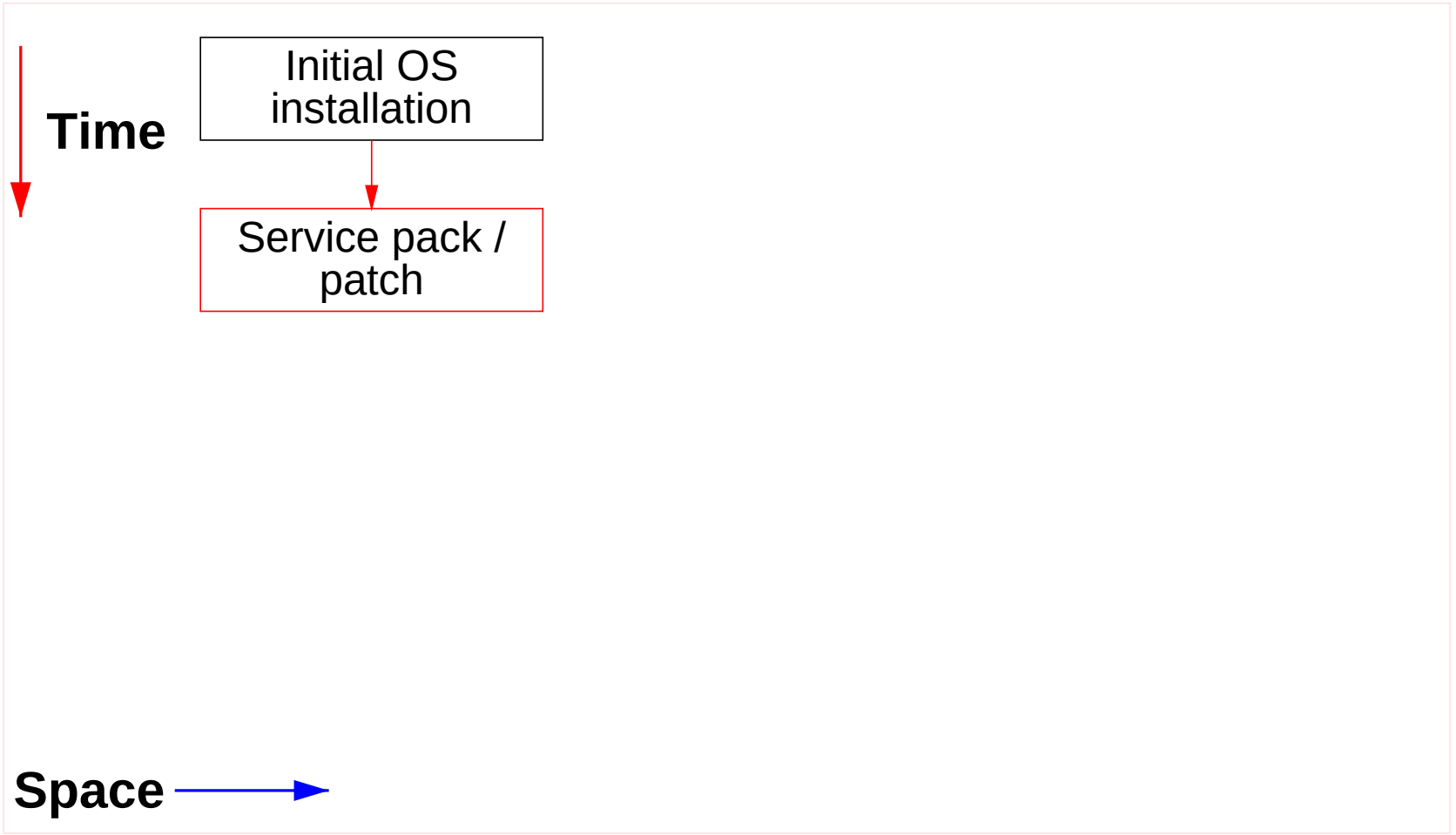
- Replication across hundreds of nodes
- Localization (IP-number, default printer, drivers, licence files, user preferences, etc.)

➪ Snapshot of every node to recover from node failure or failing local change to installation

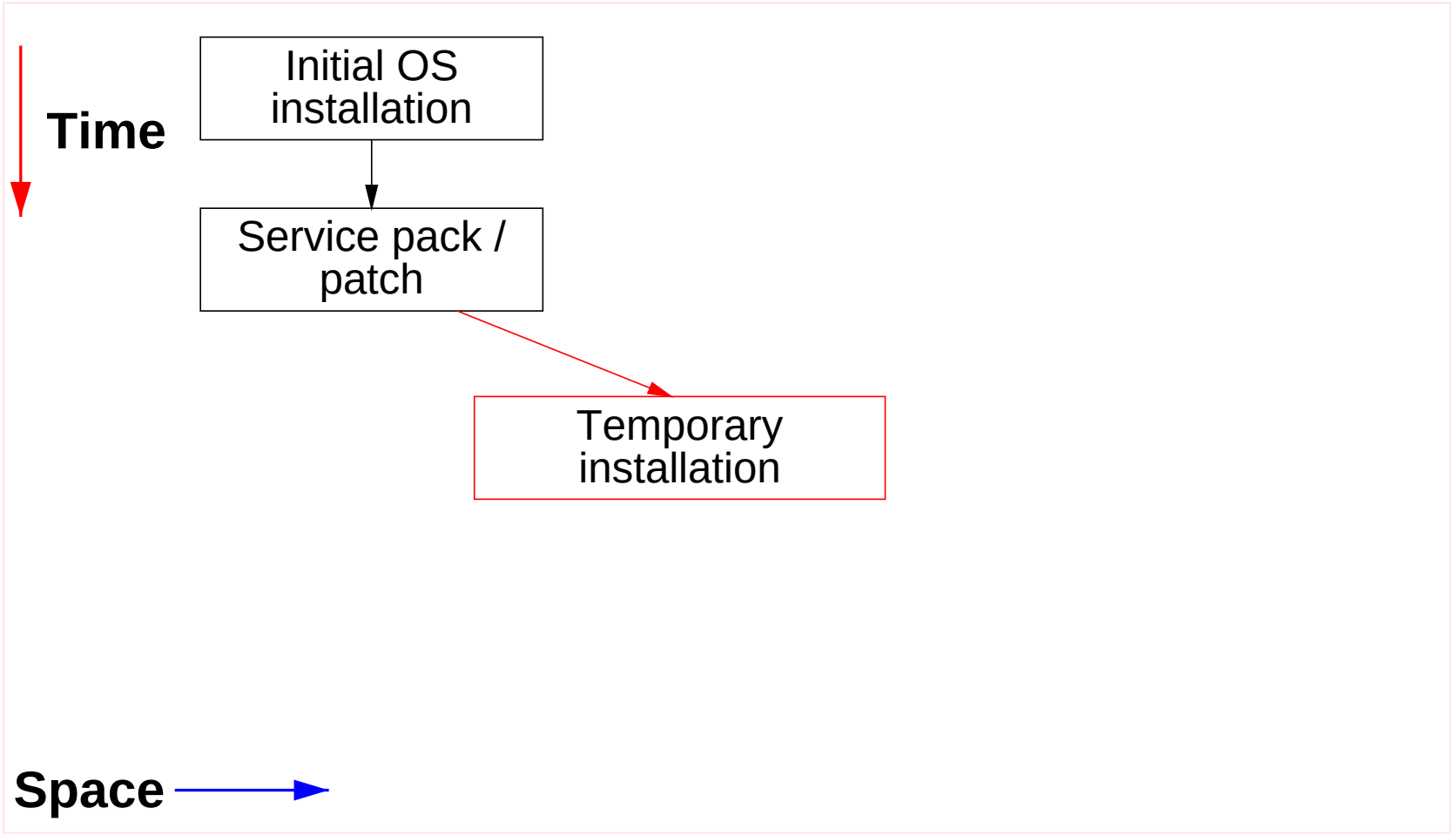
Characteristics of Installation and Maintenance



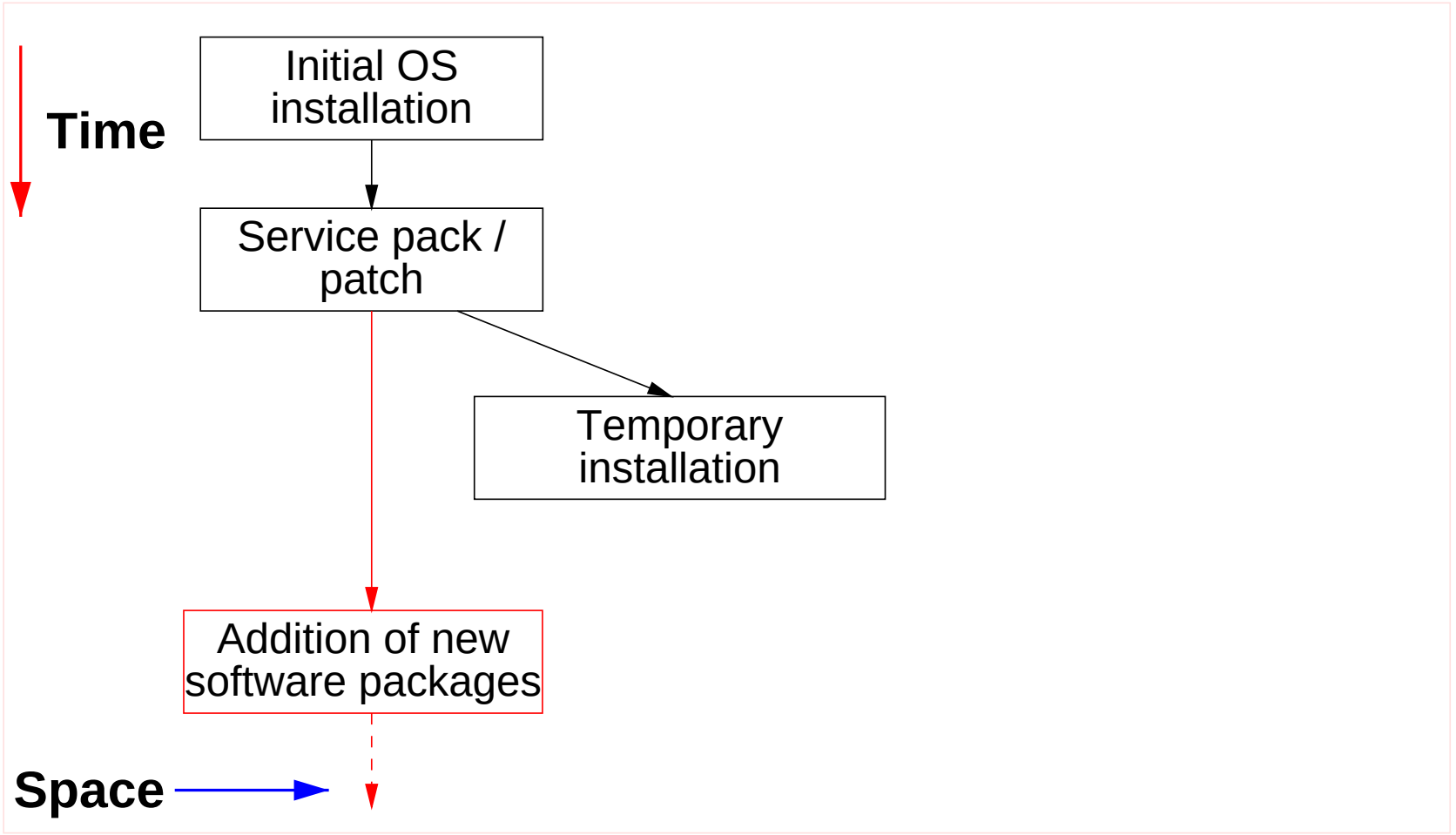
Characteristics of Installation and Maintenance



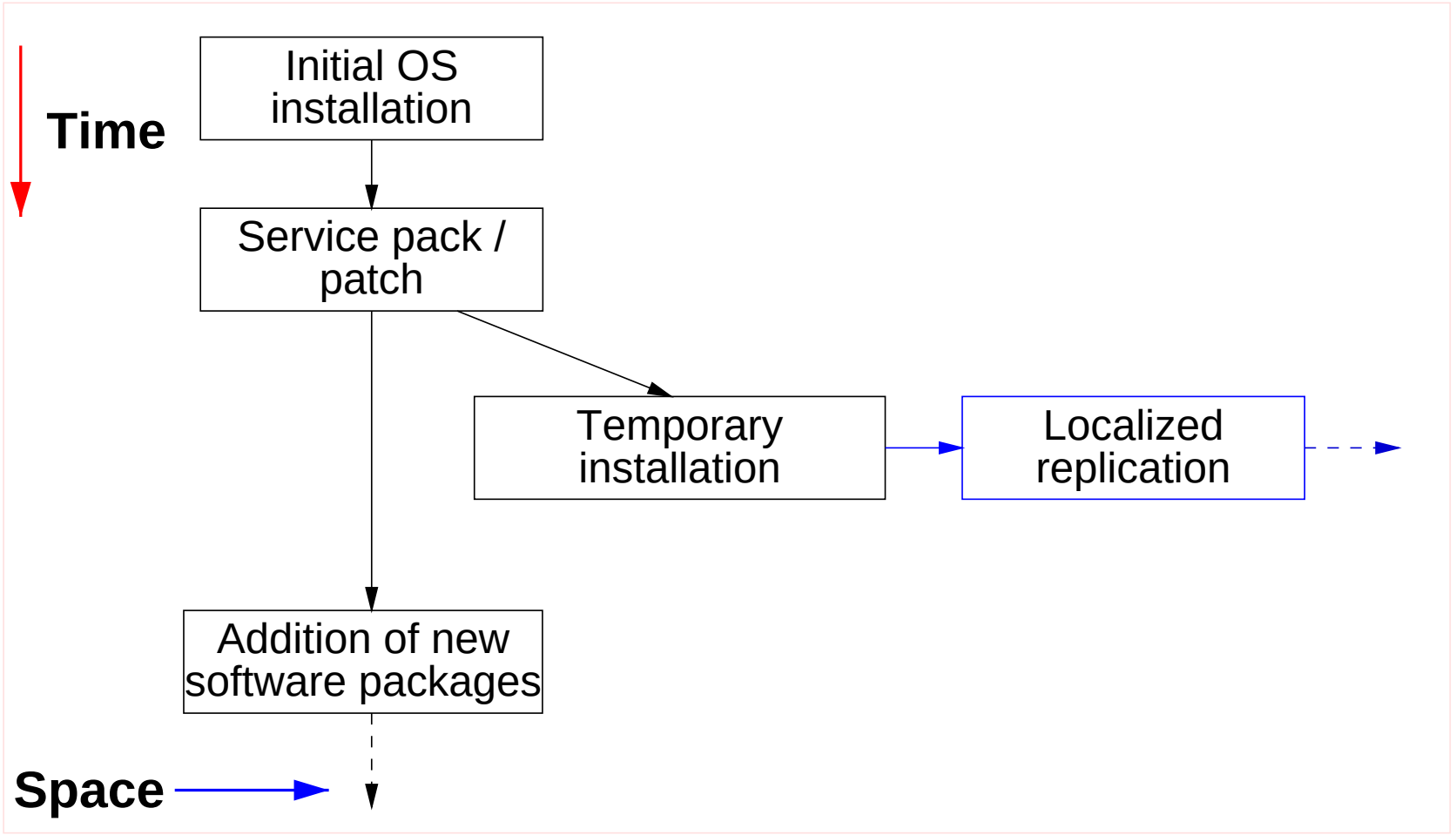
Characteristics of Installation and Maintenance



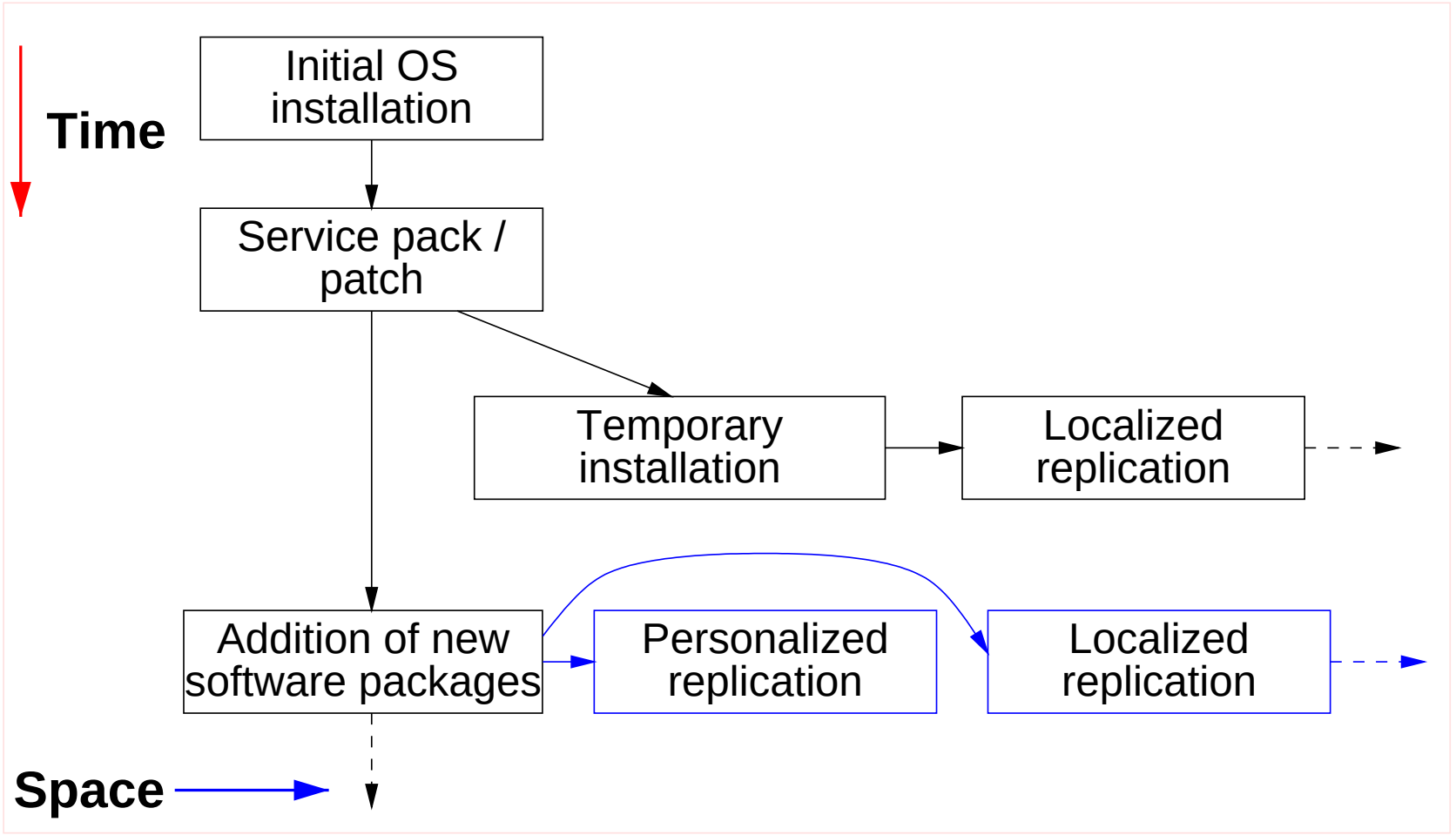
Characteristics of Installation and Maintenance



Characteristics of Installation and Maintenance



Characteristics of Installation and Maintenance



Why an OS Independent Framework?

- Simplest approach
- No knowledge about file system required
- No knowledge about operating system required

But working with partitions is a storage nightmare!

Contribution:

Solution of storage problem with block repositories

Related Work

- Cloning tool for efficient partition distribution: *Dolly* [Rauch, Kurmann, Stricker; EuroPar 2000]
- Comparison of backup speed for logical and physical backups [Hutchinson et al; OSDI 99]. Use different terminology: OS independent archive files (tar), not OS independent tools.
- Commercial tools such as Norton Ghost, Image Cast or DriveImage Pro: Either efficient or OS independent, but not both.
- Approaches by system administrators at LISA conferences: Either archive whole partitions or use file system dependent tools.

Archival Techniques

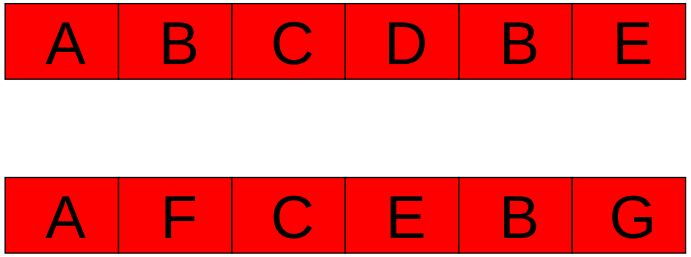
Our solution:

- Works with raw disk/partition data
- Requires no knowledge about file system
- Prerequisite: Maintenance OS (e.g. small Linux).

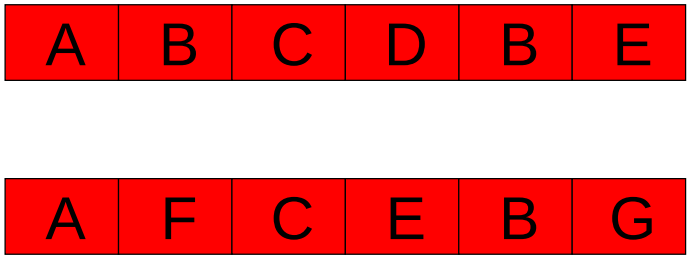
Three different archival techniques:

- Full (compressed) partition images on file server
- Blockwise diffs
- Block repositories

Full Partition Images

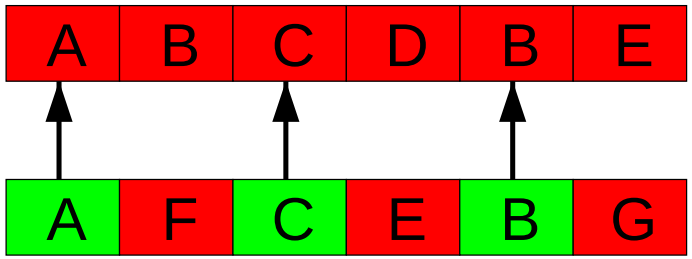


Full Partition Images



- Partition stored in file
- Compressed with gzip
- Very easy to implement
- A lot of diskspace wasted as two consecutive images are mostly identical

Blockwise Diffs



6 blocks

\+

3 blocks

= 9 blocks of storage required

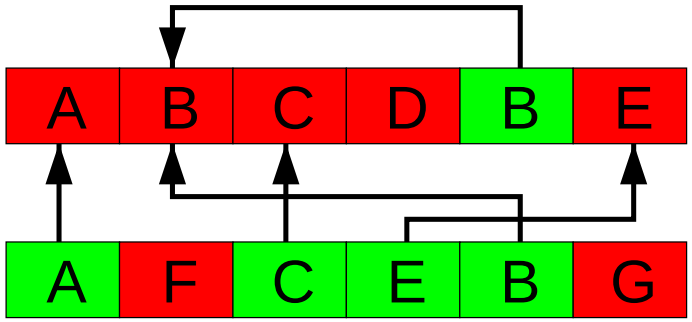
- First partition fully archived
- For subsequent installations only differences archived
- Disk blocks at same places on disks are compared
- Only changed blocks are stored in archive

Shortcomings of Blockwise Diffs

Blockwise diffs still miss some identical blocks:

- Identical blocks in the same partition
- Removal and reinstallation results in “moved blocks”
- Defragmentation of file system moves blocks around
- Log structured file systems frequently move blocks around to regain empty segments

Block Repositories



5 blocks

+

2 blocks

= 7 blocks of storage required

- All blocks are compared to each other
- First partition is compared and inserted in same way
- Only unique blocks are inserted in the repository
- For all other blocks, pointers to existing identical blocks are inserted in the repository

Properties of Block Repositories

+ Captures *all* block movements
- Effort to compare all blocks
  - Obvious complexity at the first sight: $O(n^2)$
  - In practice: Comparison of two 2 GB partitions takes only 7 minutes on high end PC.
  - Use hash-function to speed up comparison: $\approx cn+o(n)$
  - Done only once for each partition
  - Can be done as batch job during the night
- Complex implementation

Implementations

- Production system still uses full partition images.

  Based shell scripts and NFS to archive and distribute partitions.

  Available under Open Source licence.

- Small research prototype implemented in C does comparisons for blockwise diffs and block repository approaches.

  Research prototype is used for evaluation.

  Not yet available for distribution.

Evaluation

Measurements with production images of various systems (Linux, Windows NT, Oberon).

Compare different archive techniques:

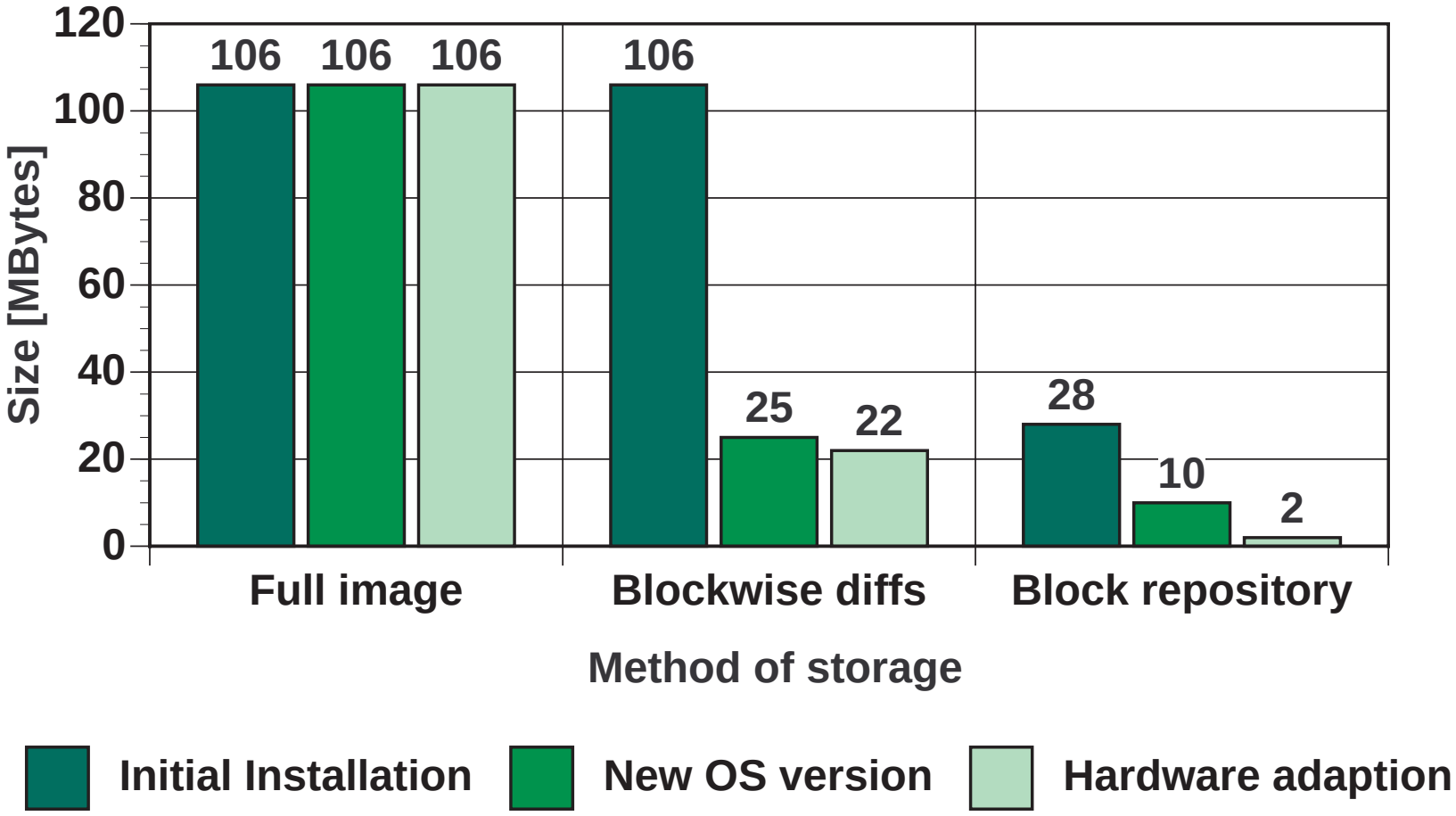
- Full partition images
- Blockwise diffs
- Block repositories

Compare different administrative tasks:

- General update / upgrade of existing system
- Localization / personalization
- Installation of new software

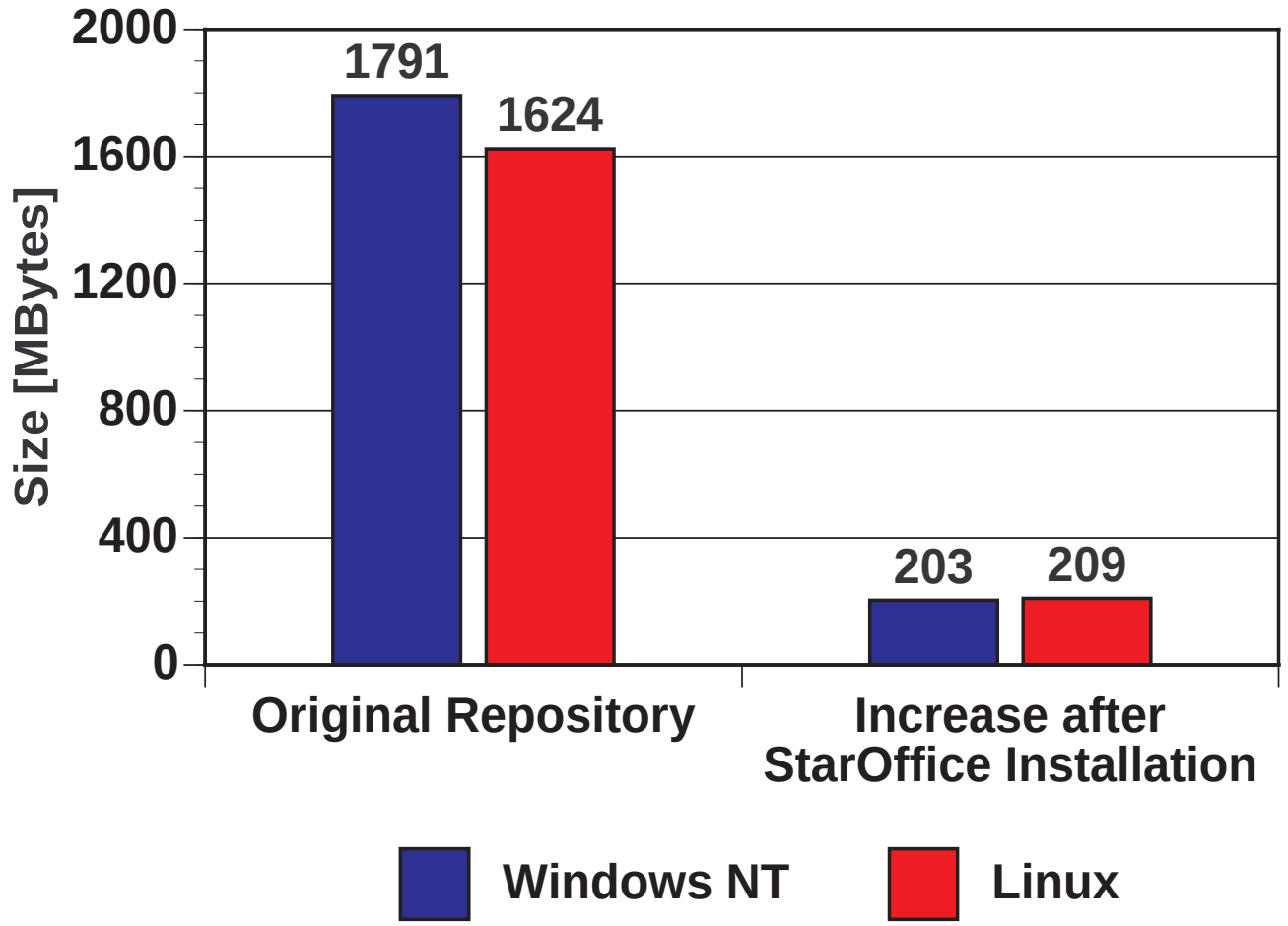
Archive Techniques

Oberon Upgrade and Driver Change (no gzip compression)



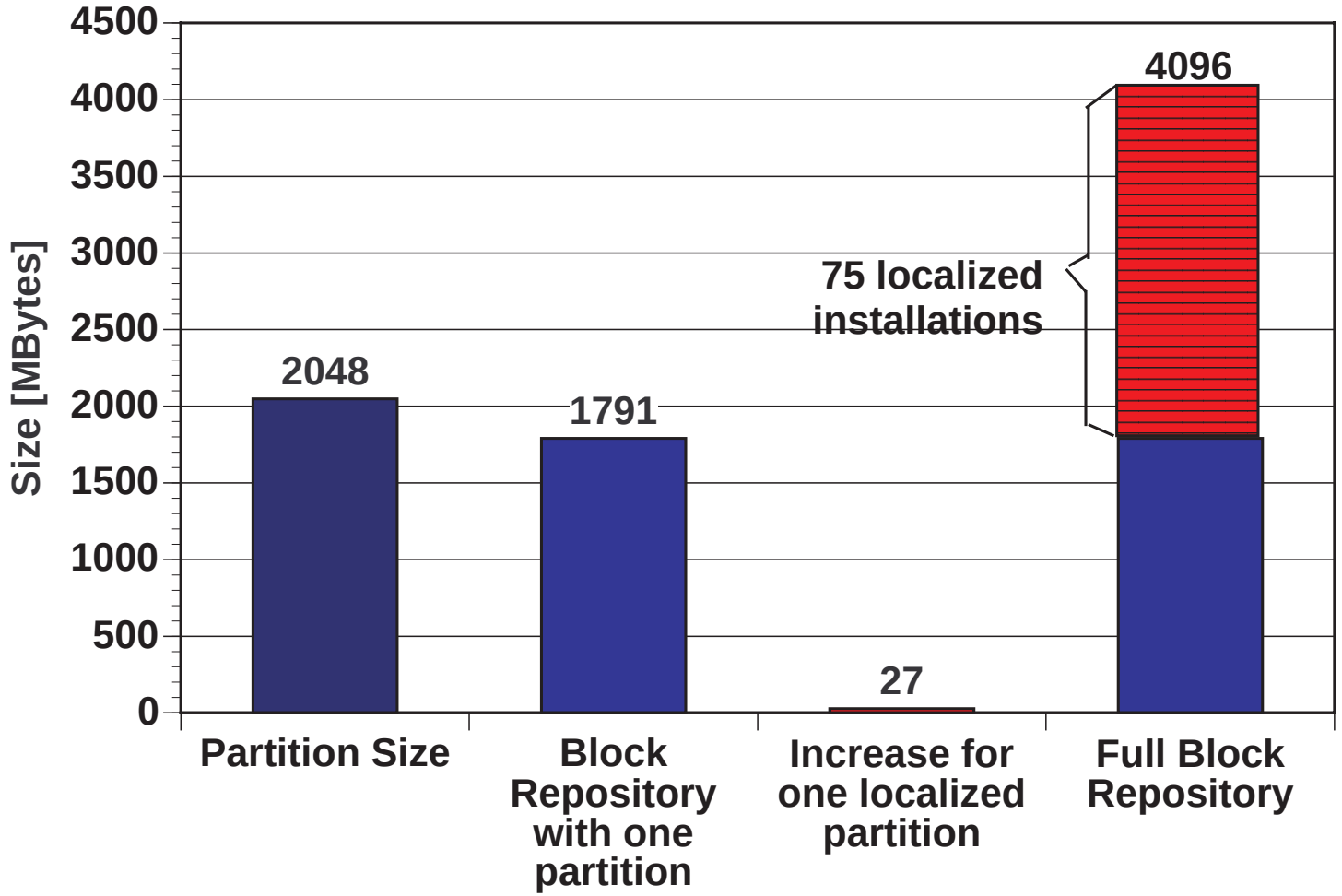
Same Application on Two Systems

Star Office in Block Repository, no gzip compression



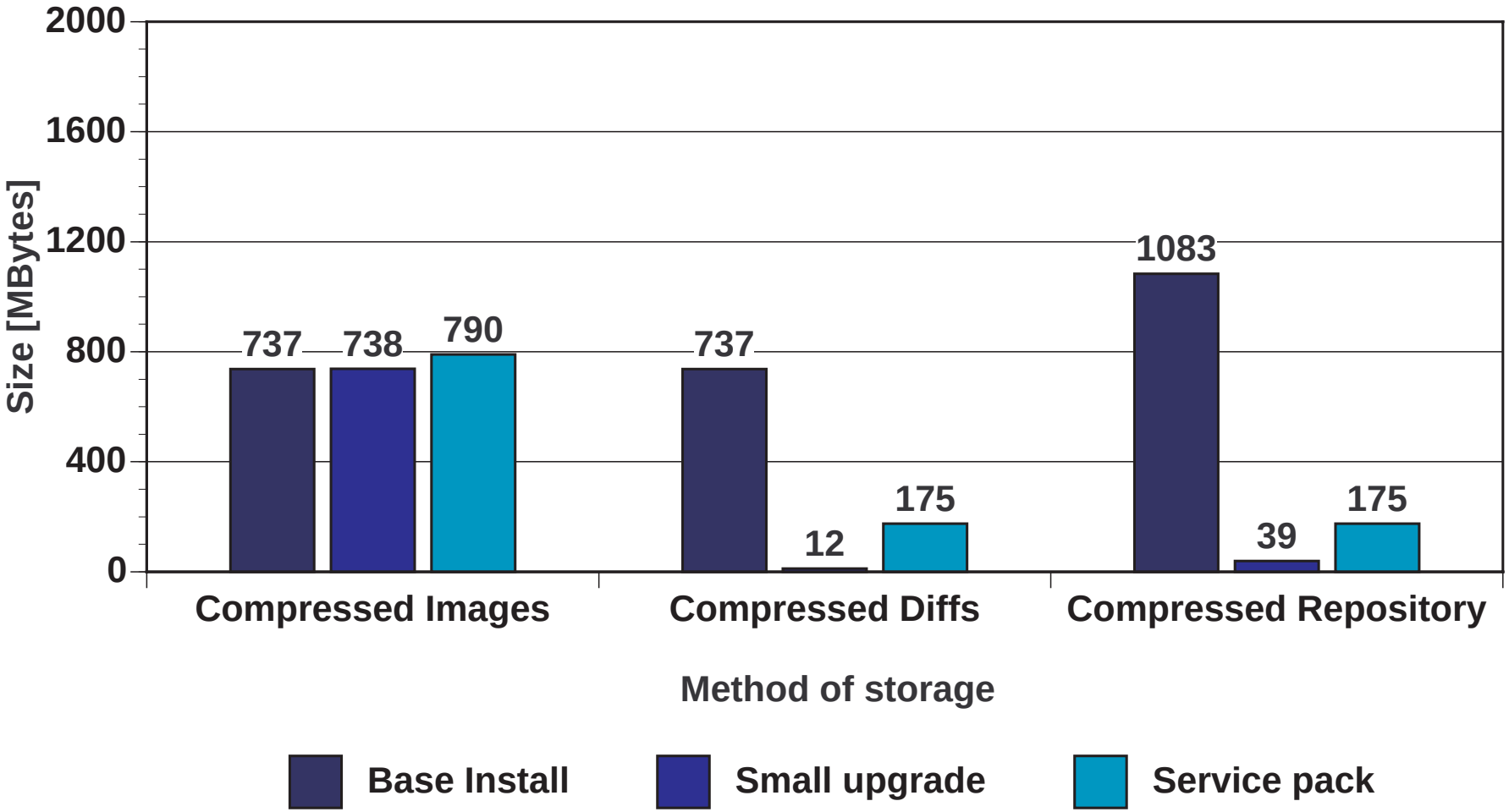
Localized Partitions in Block Repository

Localization based on replication, no gzip compression



Archive Techniques with Compression

Successive Windows NT installation, 2 GB partition



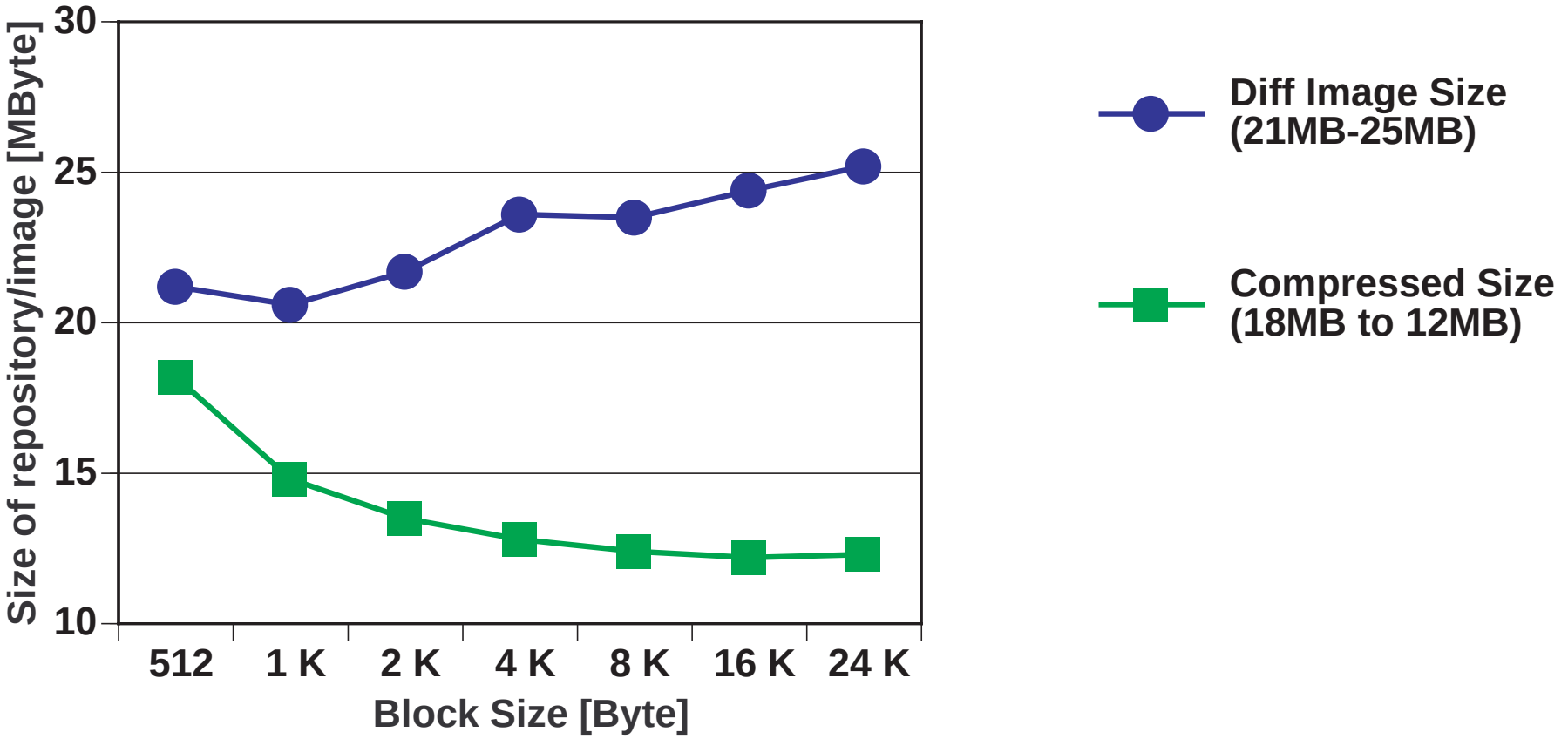
Tradeoff: Repositories vs. Compression

Using large block size for repository results in:

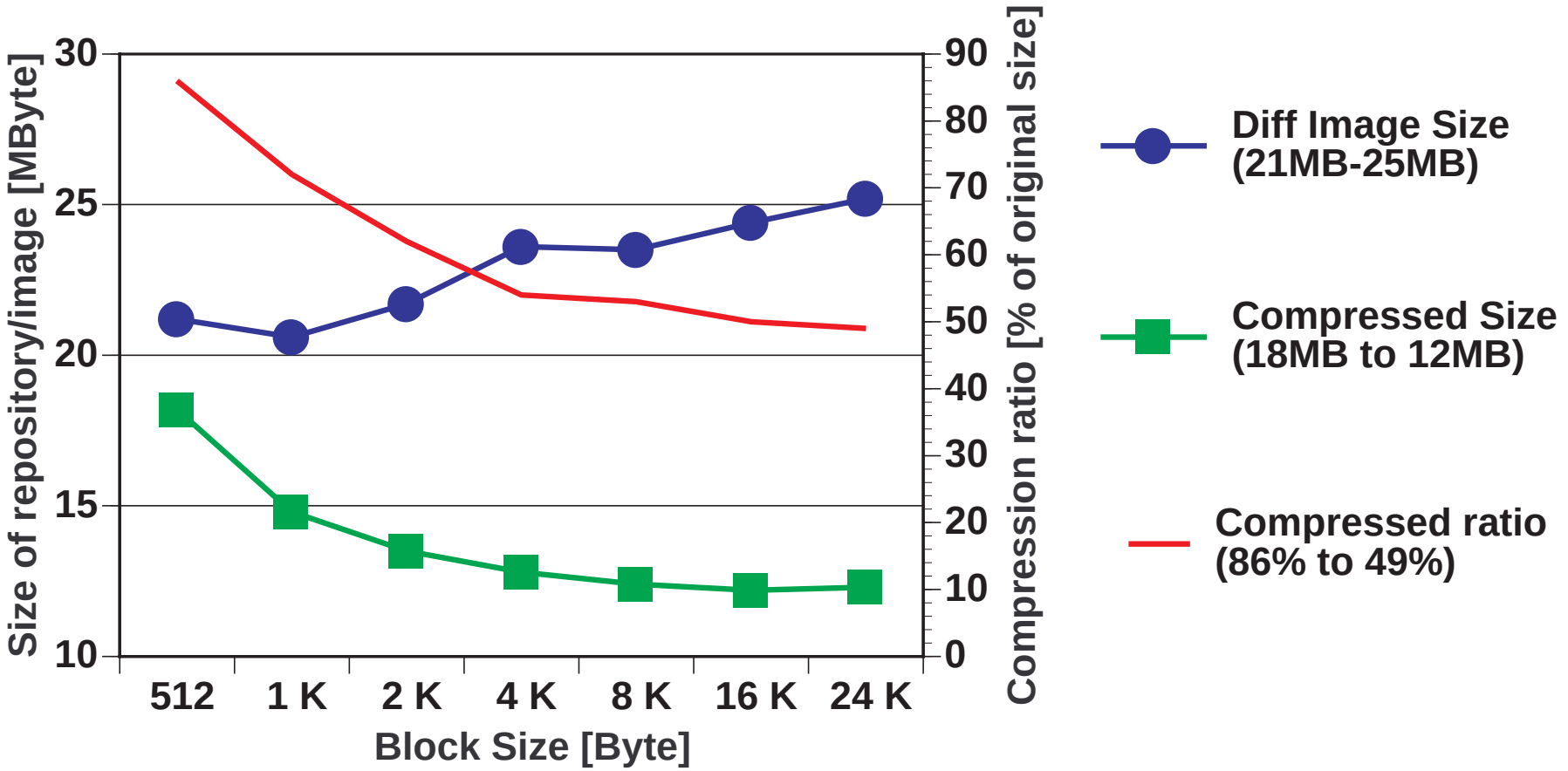
- Coarser difference detection
  ➪ Larger repository
- Better compression ratio
  ➪ Smaller repository

Which is block size for best results?

Influence of Block Size on Archive Size



Influence of Block Size on Archive Size



Conclusions

- Identified partitions as state of software installation. Changes along two axes.
- Designed *partition repositories* as efficient storage scheme (e.g. archive 75 localized Windows NT partitions in twice the size of one partition).
- Proposed cluster software maintenance system based on efficient storage scheme and optimized distribution.

Our tools are file system independent and therefore work with all operating systems (even future versions of current systems).

Questions?

CoPs - Project

Cluster of PCs

http://www.cs.inf.ethz.ch/CoPs/

Block Size: Initial Repository