キャッシュノード機能実装による NFSの大規模化とその応用

田胡　和哉

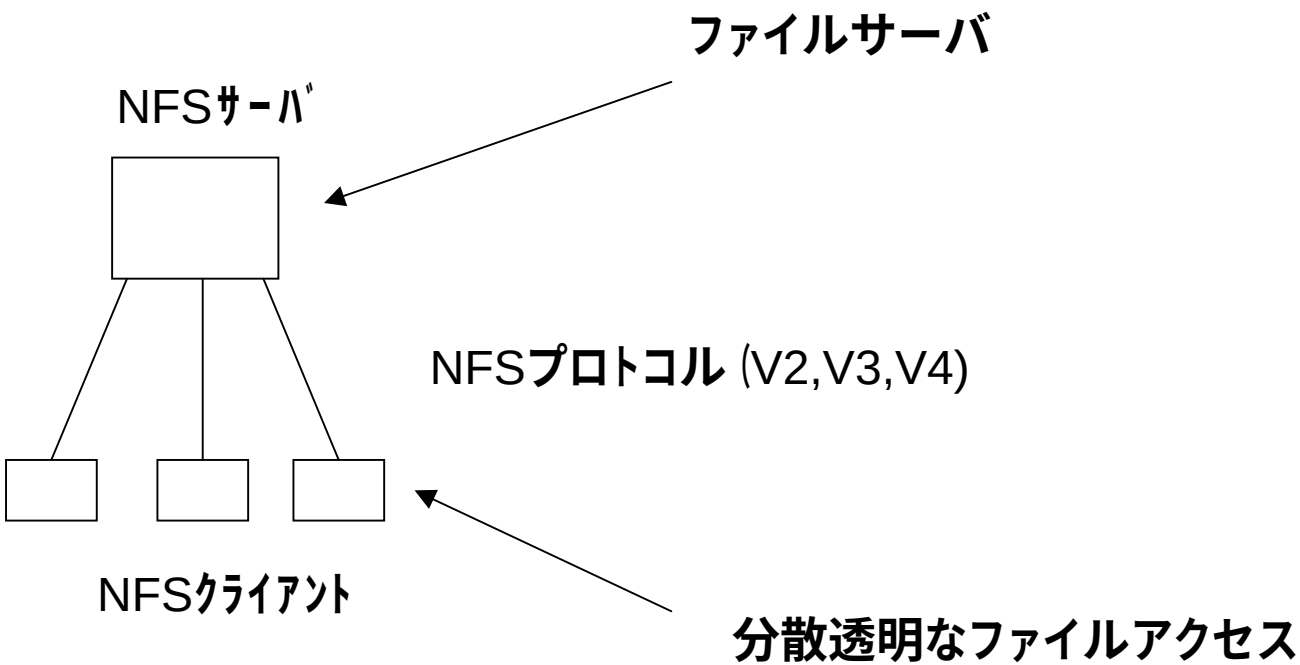
概要

- **大規模分散ファイルシステムの利用**
  - NFS
  - NFS**の大規模化**
  - **利用形態**
- NFS**の大規模化の技術**
  - Linux**のファイルシステムと現状の**NFS**実装**
  - NFS　V4
  - **キャッシュノードの実装**
- **実証実験の現状**

概要

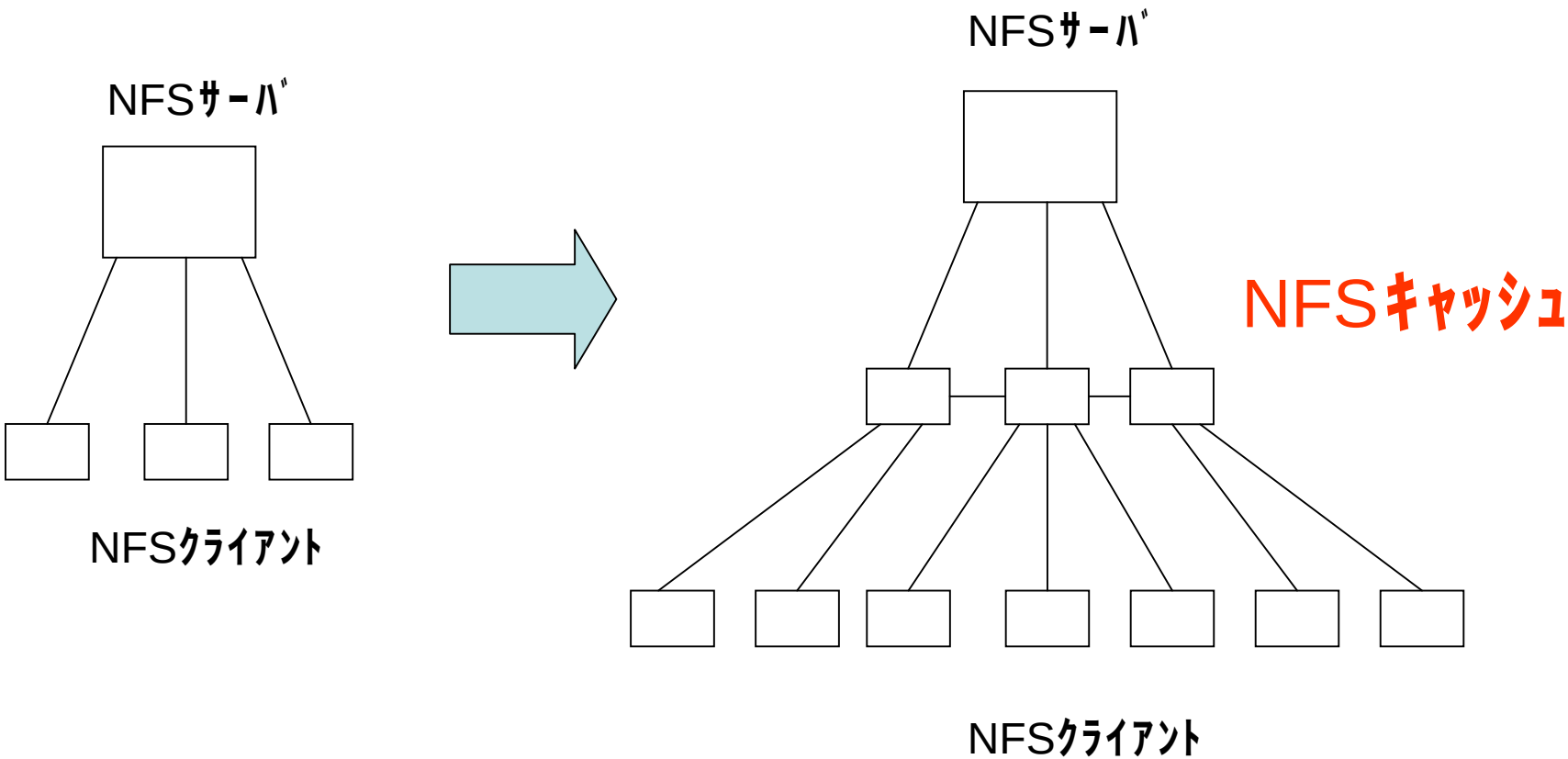
- **大規模分散ファイルシステムの利用**
  - NFS
  - NFSの大規模化
  - **利用形態**
- NFSの大規模化の技術
  - Linuxのファイルシステムと現状のNFS実装
  - NFS　V4
  - キャッシュノードの実装
- 実証実験の現状

Network File System (NFS)



現在のNFS

Community Storage



NFSの大規模化

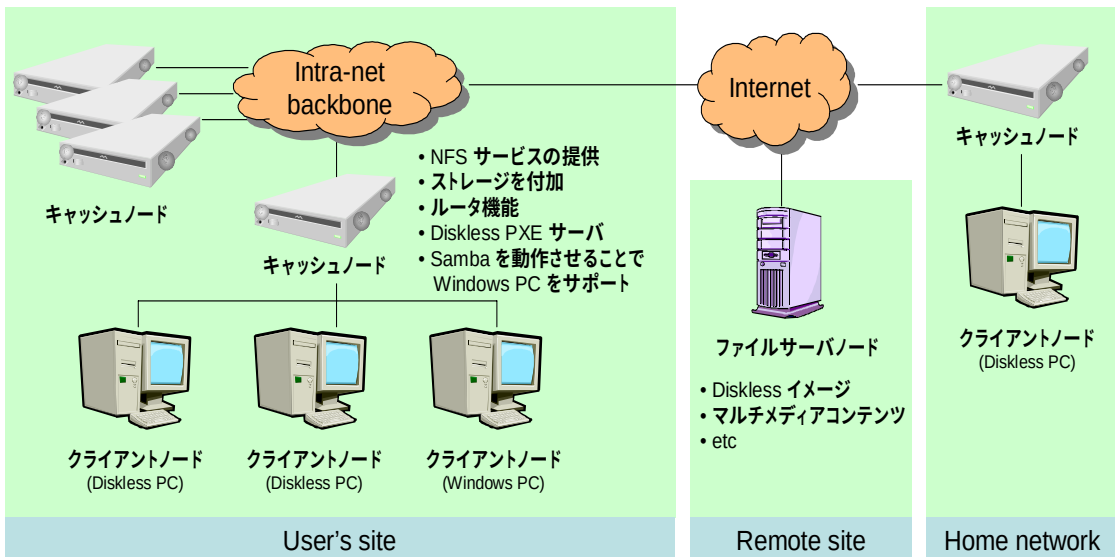
- **クライアント数の増加**
  例： 100**台** -> 10000**台**
  一つの組織全体を一つのファイルシステムで賄う

- **クライアント設置場所の広域化**
  例：職場と自宅、サテライトオフィス

- **扱えるファイルの大容量化**
  例：マルチメディアコンテンツの収集、配信

コミュニティストレージ

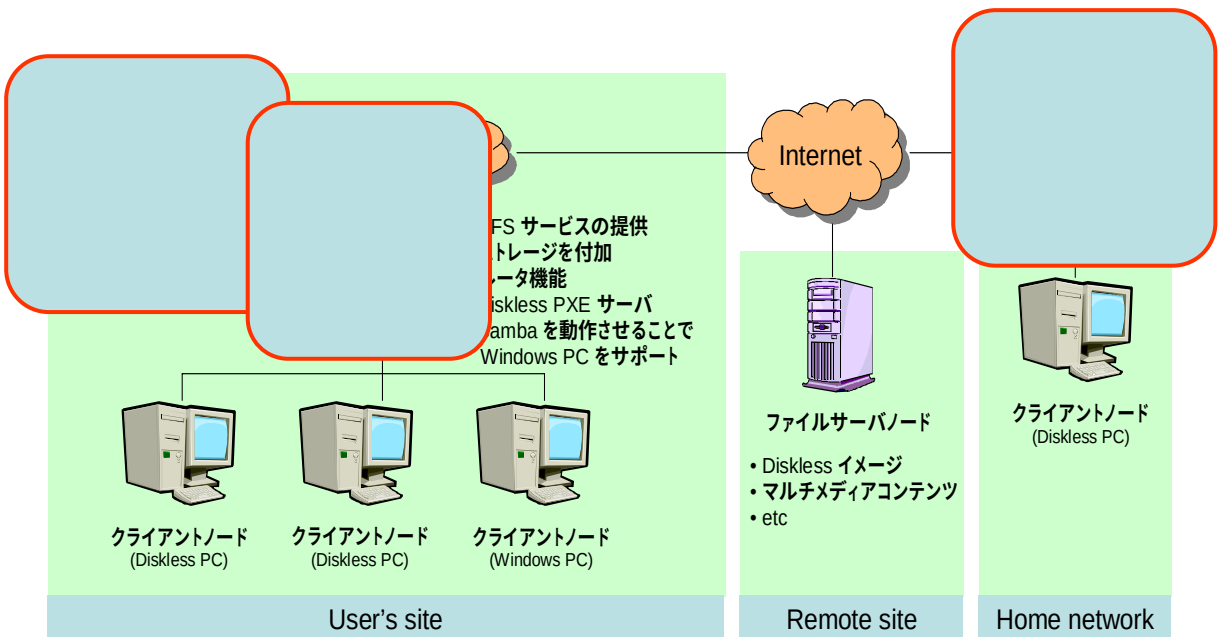
- 概要
  - 大規模ネットワーク環境において、構成員全員が同一の条件で利用できるNFSサービスを提供するシステム
  - 分散ファイルシステムにキャッシュを挿入してスーラビリティを上げる
  - NFSv4プロトコルで整合性が保たれている



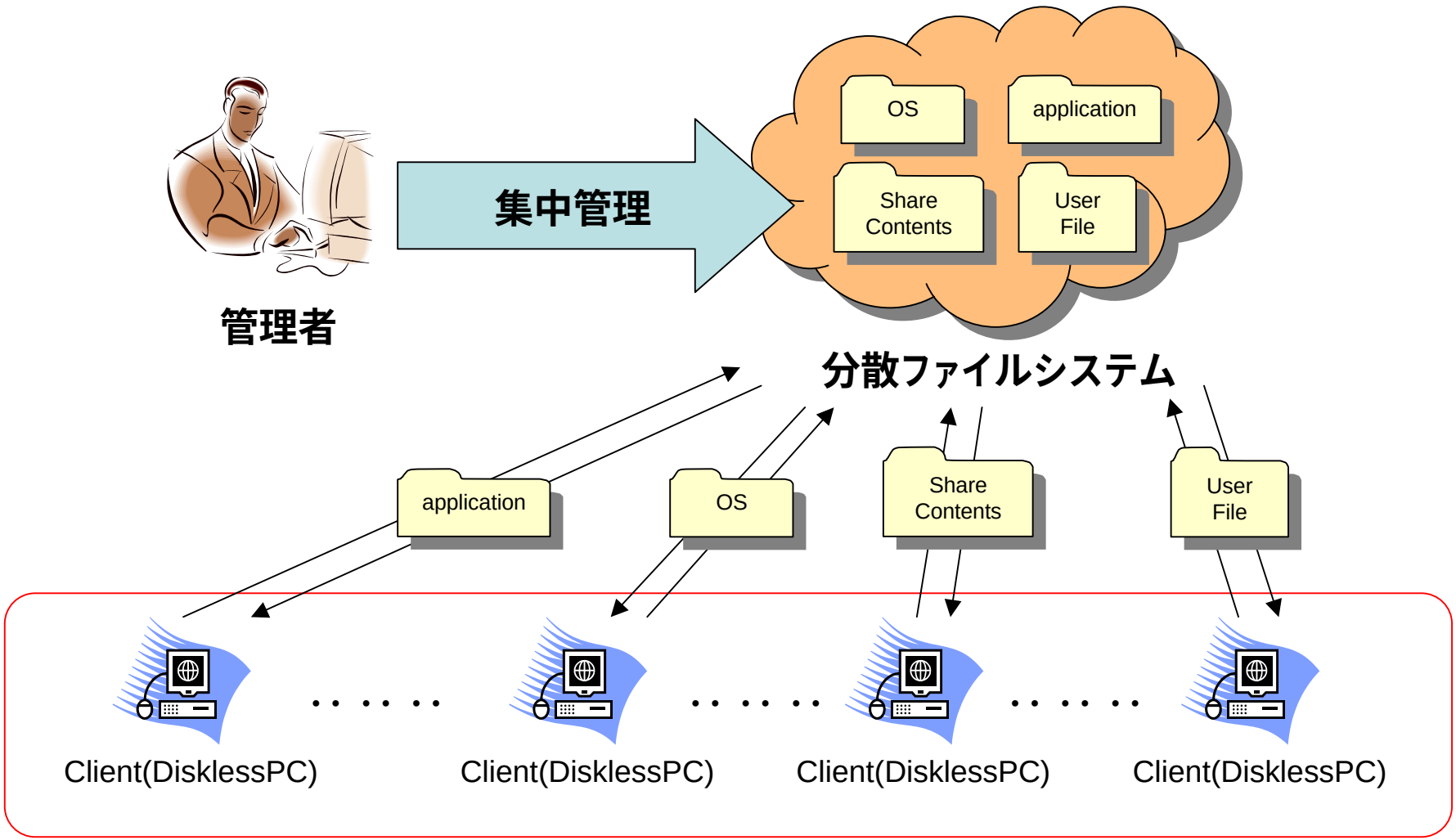
システムの全体構造

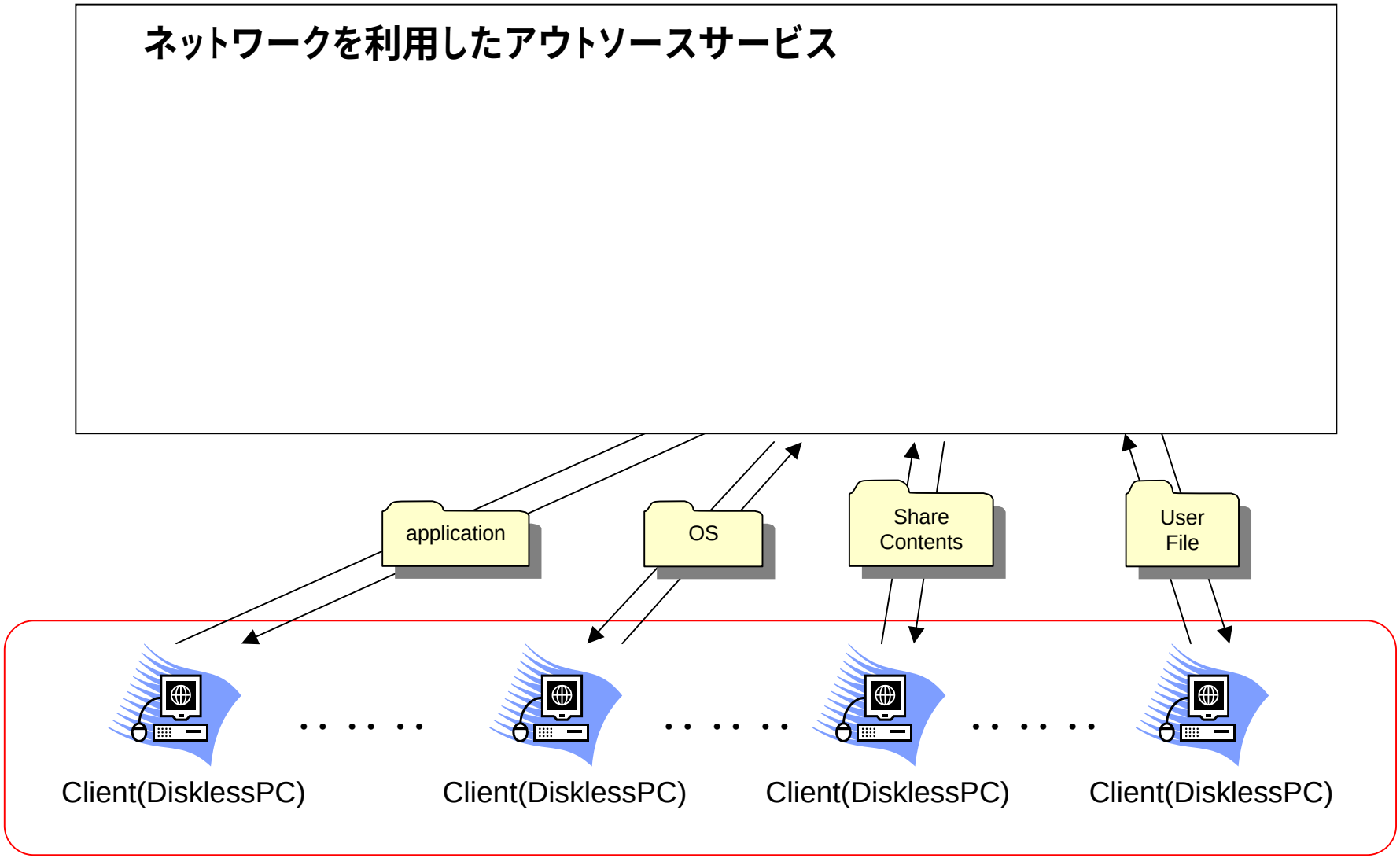
コミュニティストレージ

- 概要
  - 大規模ネットワーク環境において、構成員全員が同一の条件で利用できるNFSサービスを提供するシステム
  - 分散ファイルシステムにキャッシュを挿入してスーラビリティを上げる
  - NFSv4プロトコルで整合性が保たれている



システムの全体構造


管理者
集中管理
OS
application
Share Contents
User File
分散ファイルシステム
application
OS
Share Contents
User File
Client(DisklessPC)
Client(DisklessPC)
Client(DisklessPC)
Client(DisklessPC)

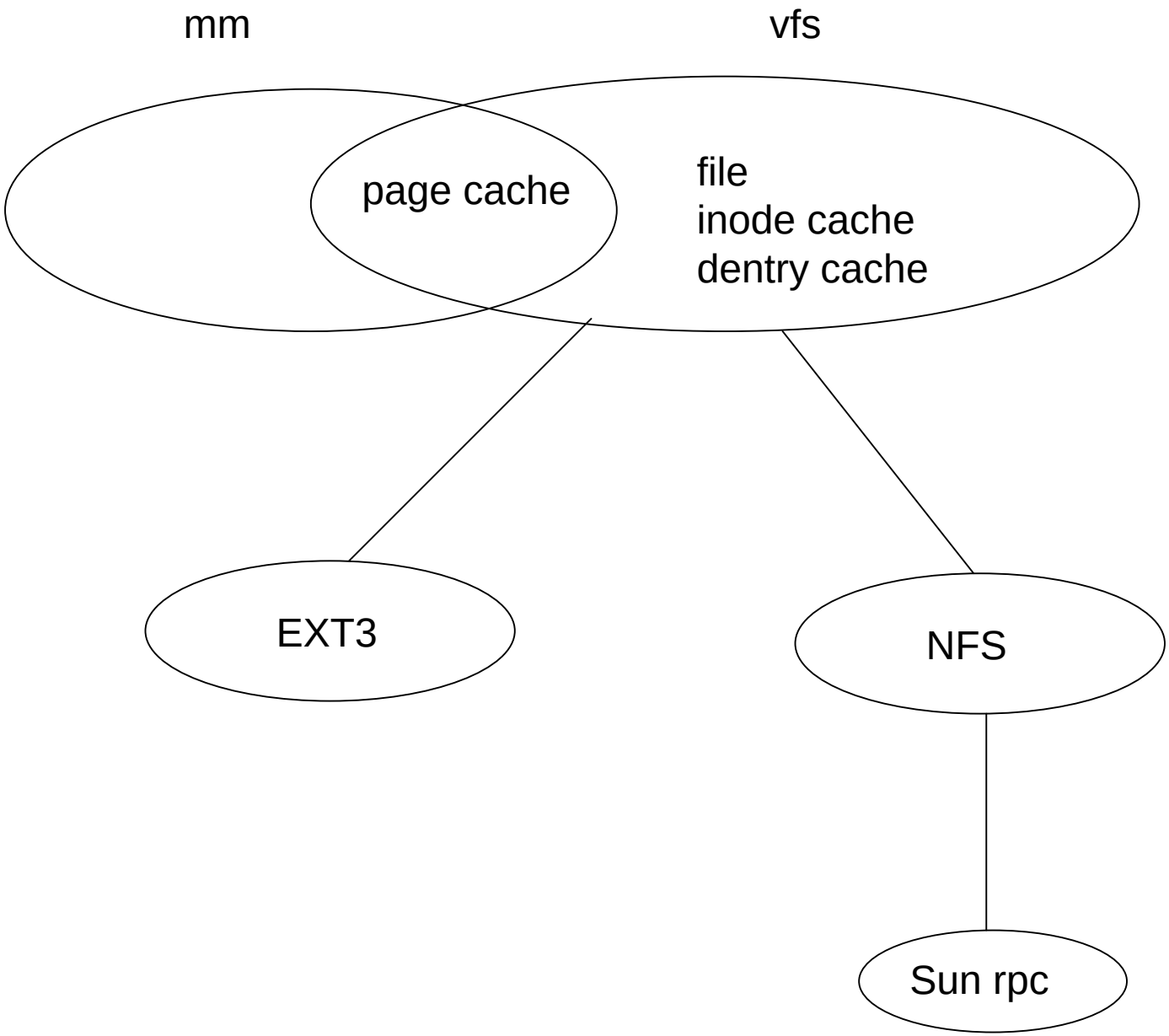

ネットワークを利用したアウトソースサービス
application
OS
Share
Contents
User
File
Client(DisklessPC)
Client(DisklessPC)
Client(DisklessPC)
Client(DisklessPC)

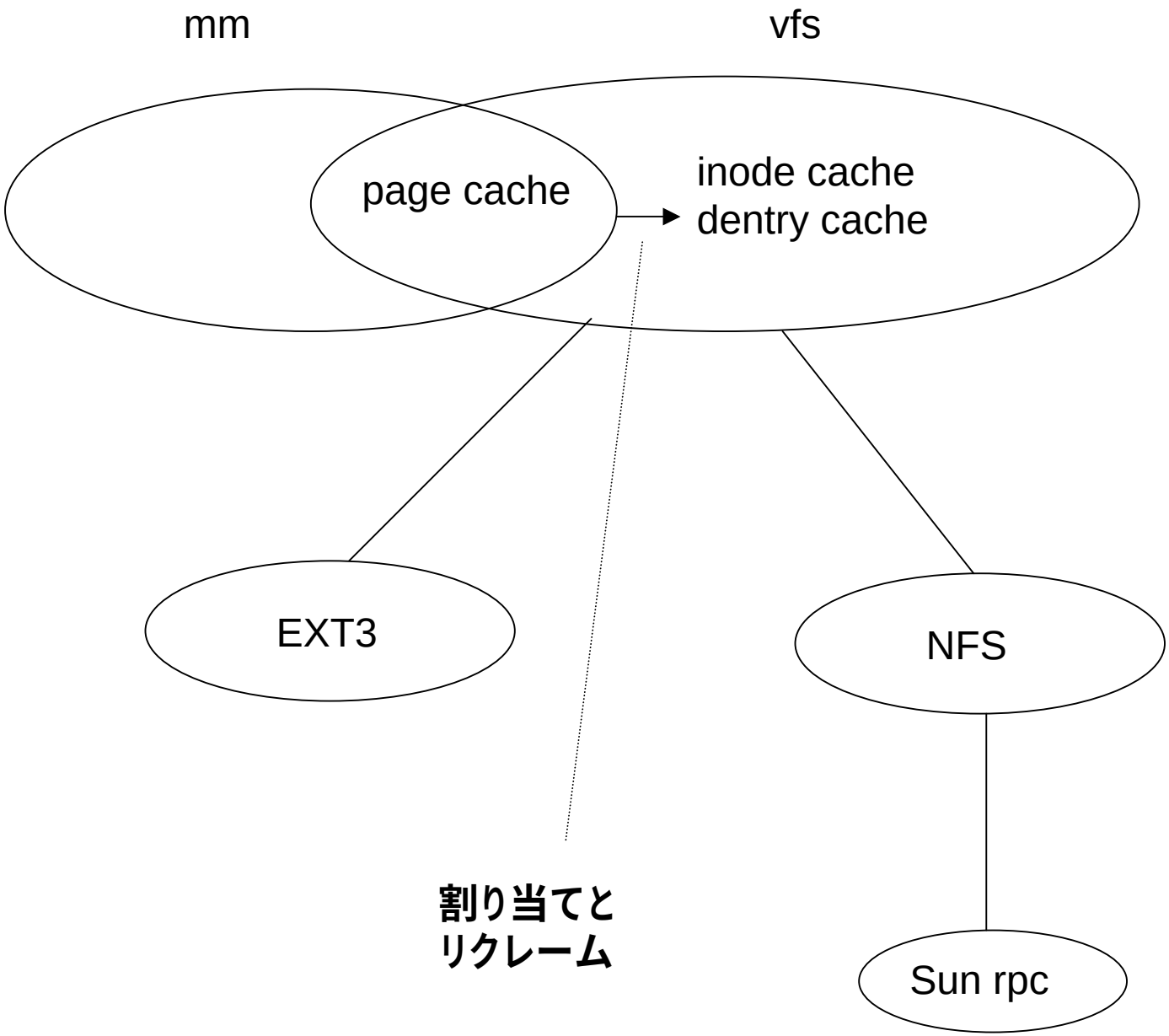
背景

- ITシステムの管理コストの飛躍的増大
  - システム導入コストより導入後の管理コストが大
- セキュリティの重要性の増大
  - 個人情報保護法
- 労働形態の変化
  - 政府目標：2010年までに全労働人口の20%をテレワーカ化

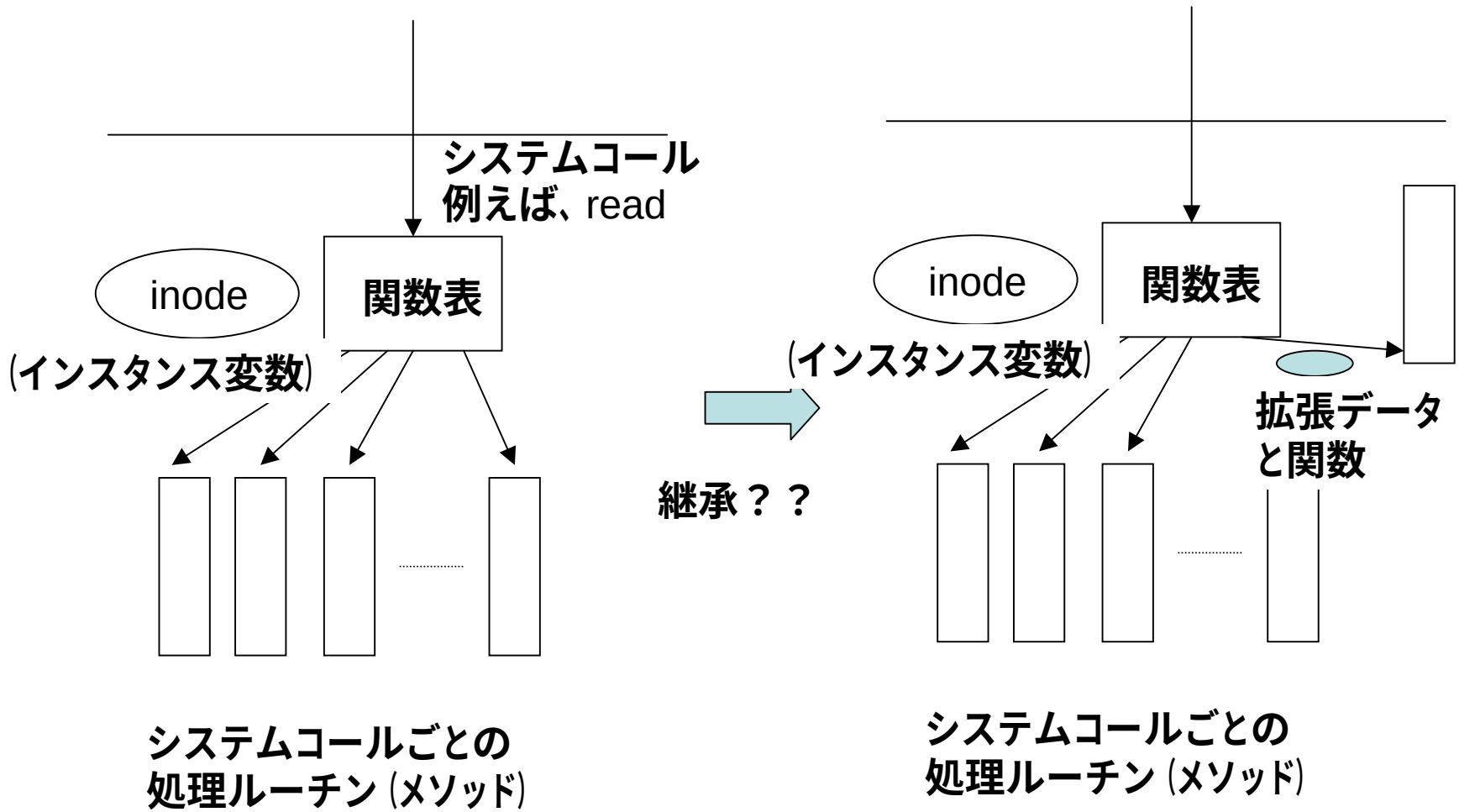
概要

- 大規模分散ファイルシステムの利用
  - NFS
  - NFSの大規模化
  - 利用形態
- NFSの大規模化の技術
  - Linuxのファイルシステムと現状のNFS実装
  - NFS　V4
  - キャッシュノードの実装
- 実証実験の現状


mm
vfs
page cache
file
inode cache
dentry cache
EXT3
NFS
Sun rpc


mm
vfs
page cache
inode cache
dentry cache
EXT3
NFS
Sun rpc
割り当てと
リクレーム

VFSのハッキング

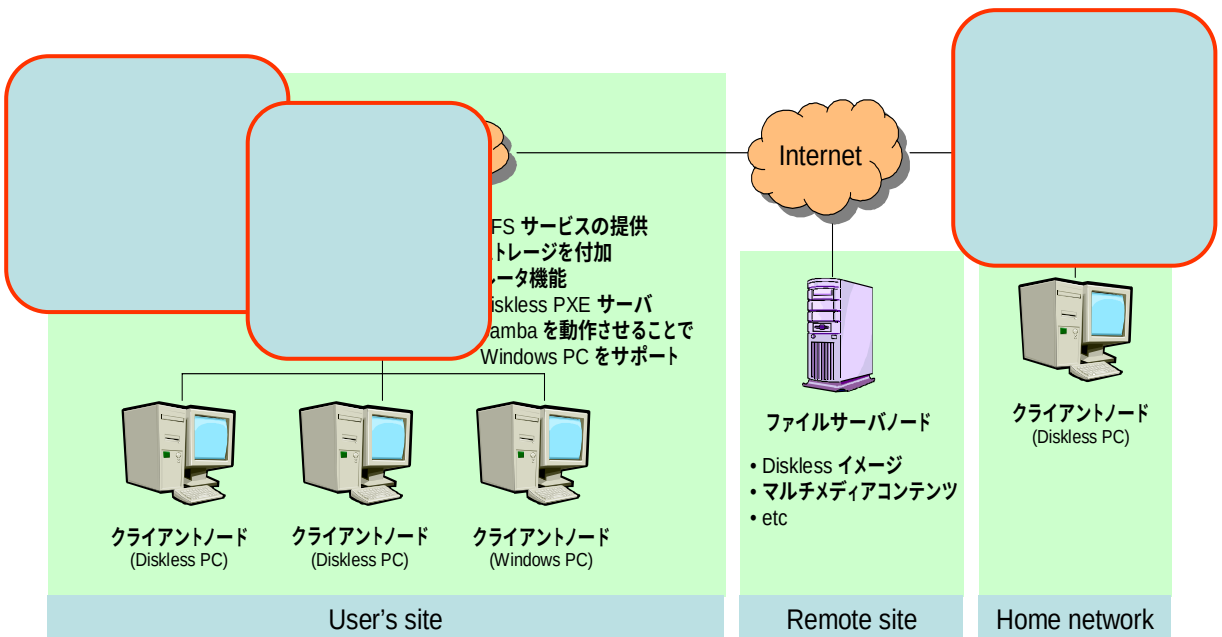

システムコール
例えば、read
inode
関数表
(インスタンス変数)
継承？？
inode
関数表
(インスタンス変数)
拡張データ
と関数
システムコールごとの
処理ルーチン (メソッド)
システムコールごとの
処理ルーチン (メソッド)

NFS

- Sun社によって開発
- V2, V3
  - ステートレス　ページ　アクセス
    - RPCごとに、ファイルopen/closeをくりかえす
      - ファイル全体としてのopen/closeは行わない
    - Lookupでファイル名とファイルハンドルを対応付け
      (いつまで対応関係を維持すればよいか？？)
- V4
  - ステートフル　アクセス
    - ファイル　open / close 処理あり
    - delegation/recall
    - 現在、開発が進行中

コミュニティストレージ

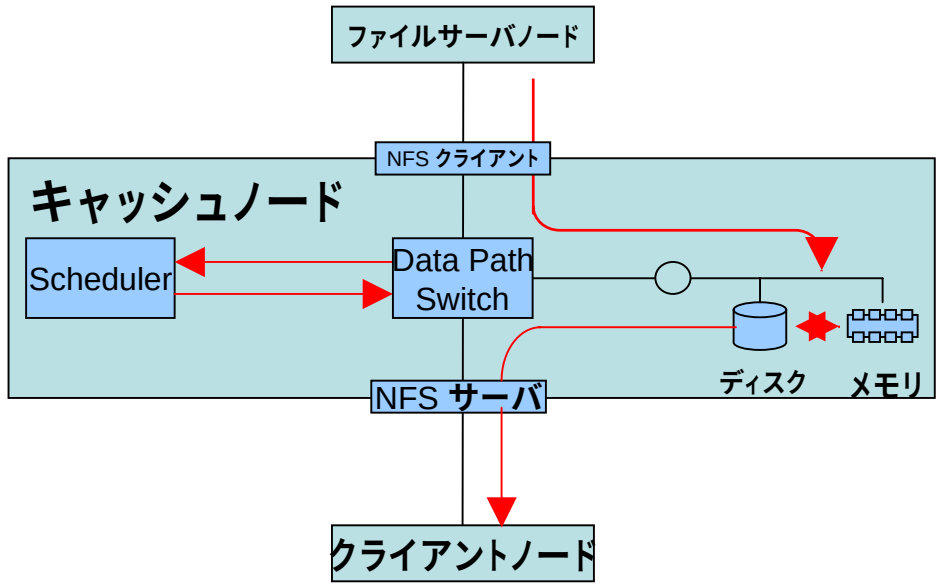
- 概要
  - 大規模ネットワーク環境において、構成員全員が同一の条件で利用できるNFSサービスを提供するシステム
  - 分散ファイルシステムにキャッシュを挿入してスーラビリティを上げる
  - NFSv4プロトコルで整合性が保たれている



システムの全体構造

キャッシュノード

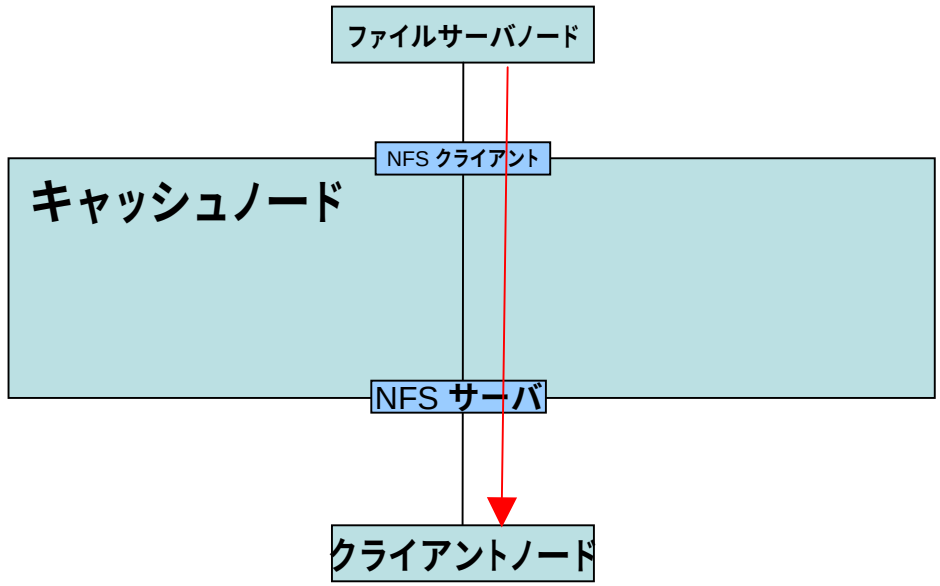
- **構造**
  - Data Path Switch
    - **キャッシュ内の一旦データを貯める**
    - **ファイルアクセスの様子を** Scheduler **への通知**
  - Scheduler
    - **ネットワークの環境管理**
    - **アクセスやファイルの統計分析**



キャッシュノードの構造

キャッシュノード　ー　再エクスポート

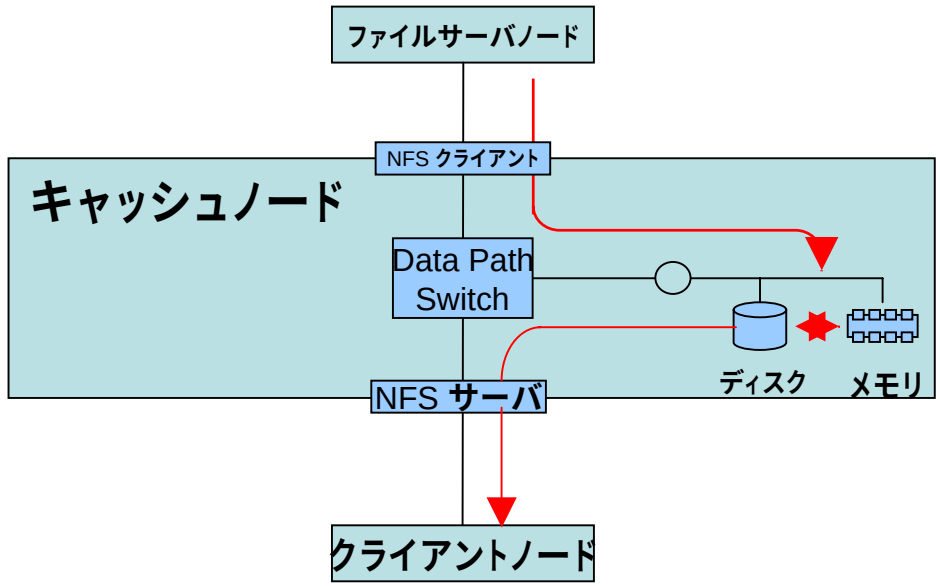
- **構造**
  - Data Path Switch
    - **キャッシュ内の一旦データを貯める**
    - **ファイルアクセスの様子を** Scheduler **への通知**
  - Scheduler
    - **ネットワークの環境管理**
    - **アクセスやファイルの統計分析**



キャッシュノードの構造

キャッシュノード　－　ファイル入出力のフック

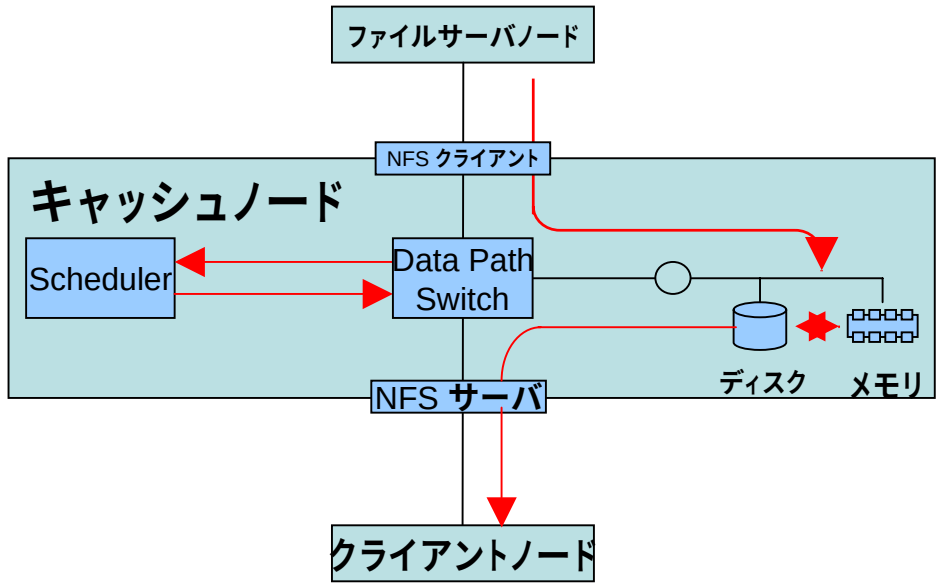
- **構造**
  - Data Path Switch
    - **キャッシュ内の一旦データを貯める**
    - **ファイルアクセスの様子を** Scheduler **への通知**
  - Scheduler
    - **ネットワークの環境管理**
    - **アクセスやファイルの統計分析**



キャッシュノードの構造

キャッシュノード　－　スケジューリング

- **構造**
  - Data Path Switch
    - **キャッシュ内の一旦データを貯める**
    - **ファイルアクセスの様子を** Scheduler **への通知**
  - Scheduler
    - **ネットワークの環境管理**
    - **アクセスやファイルの統計分析**



キャッシュノードの構造

プログラム構成

- DPSカーネルモジュール
- NFS　クライアント　パッチ
  - 再エクスポート機能の追加
  - 新ファイル追加
- カーネル　パッチ
  - 2箇所程度
- Schedulerデーモン

難しい点

- そもそもNFSは再エクスポートに対応していない！
  - ファイルハンドルの取り扱い
  - dentryキャッシュの取り扱い
- カーネル変更を最小化する
- NFS　V3への対応
  - ステートレス
- NFS　V4の実装が変化している

実現

- **現状**
  - Kernel 2.6.16 Fedora Core
  - **再エクスポート機能を用いて、日常業務に使用中（利用者　18名）**
  - **キャッシュ機能はほぼ動作、デバッグ中**

コード公開

- 1ヶ月程度でOSCJを通じて公開の予定

研究

- Autonomic Computing**技術によるスケジューラの改善**
  - **ストレージの性能改善**
    - **ポリシ階層化による**OSD**性能の改善**
  - **ネットワーク利用の改善**
    - **キャッシュ間連携のスケジューリング**
  - **システム性能の改善**
    - **キャッシュ間連携とジョブスケジューリングの連動**

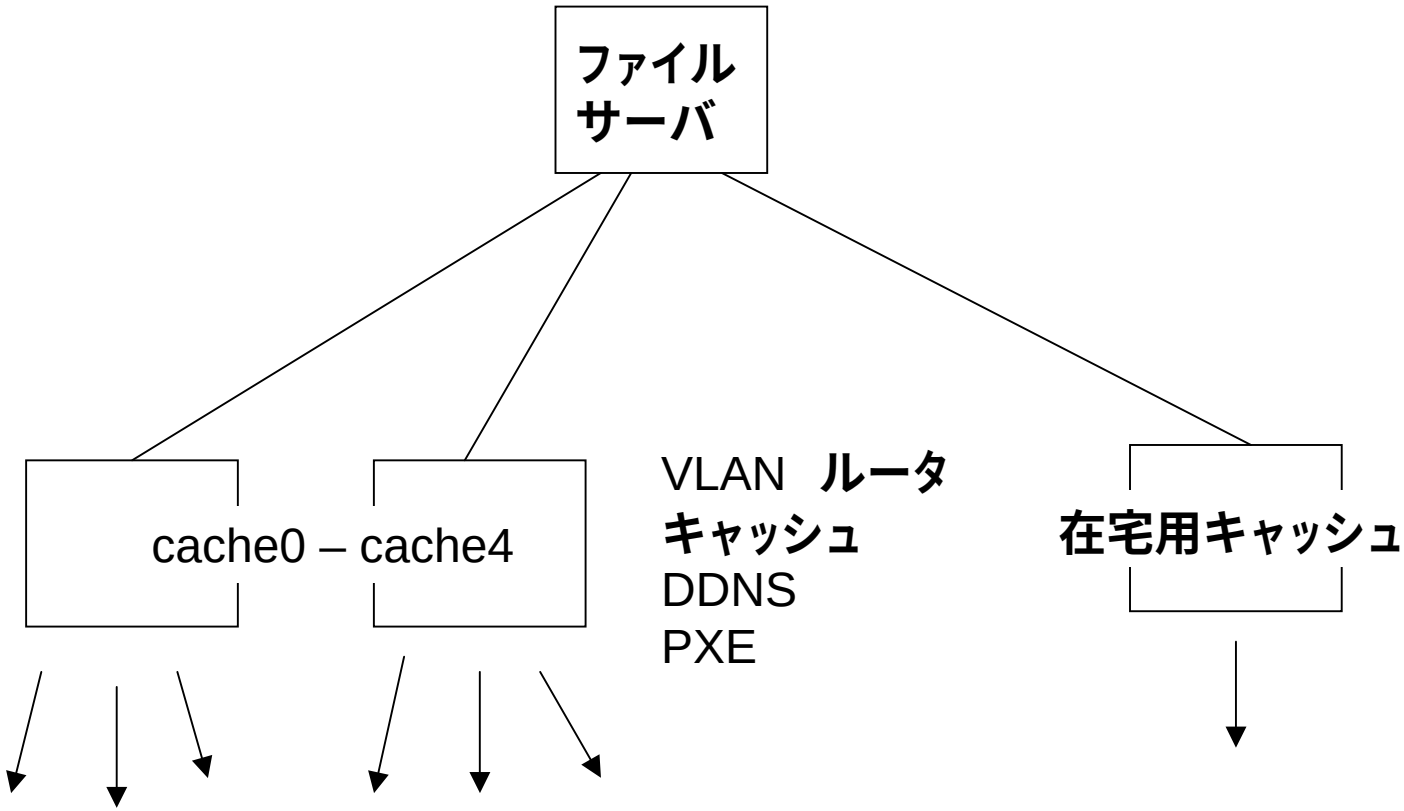
概要

- 大規模分散ファイルシステムの利用
  - NFS
  - NFSの大規模化
  - 利用形態
- NFSの大規模化の技術
  - Linuxのファイルシステムと現状のNFS実装
  - NFS　V4
  - キャッシュノードの実装
- 実証実験の現状

実証実験

- **次世代コンピューティング環境**　Sylphide
  - **ディスクレス　サービス**
  - **プロセッサ　プール　サービス**
  - **在宅勤務環境　サービス**
  - **マルチサイト運用　サービス**
  - **非**PC**統合**
    - **携帯、ゲーム機**
  - Mozilla**デスクトップ　(開発中)**
- **実験室で運用中 (１８名利用)**

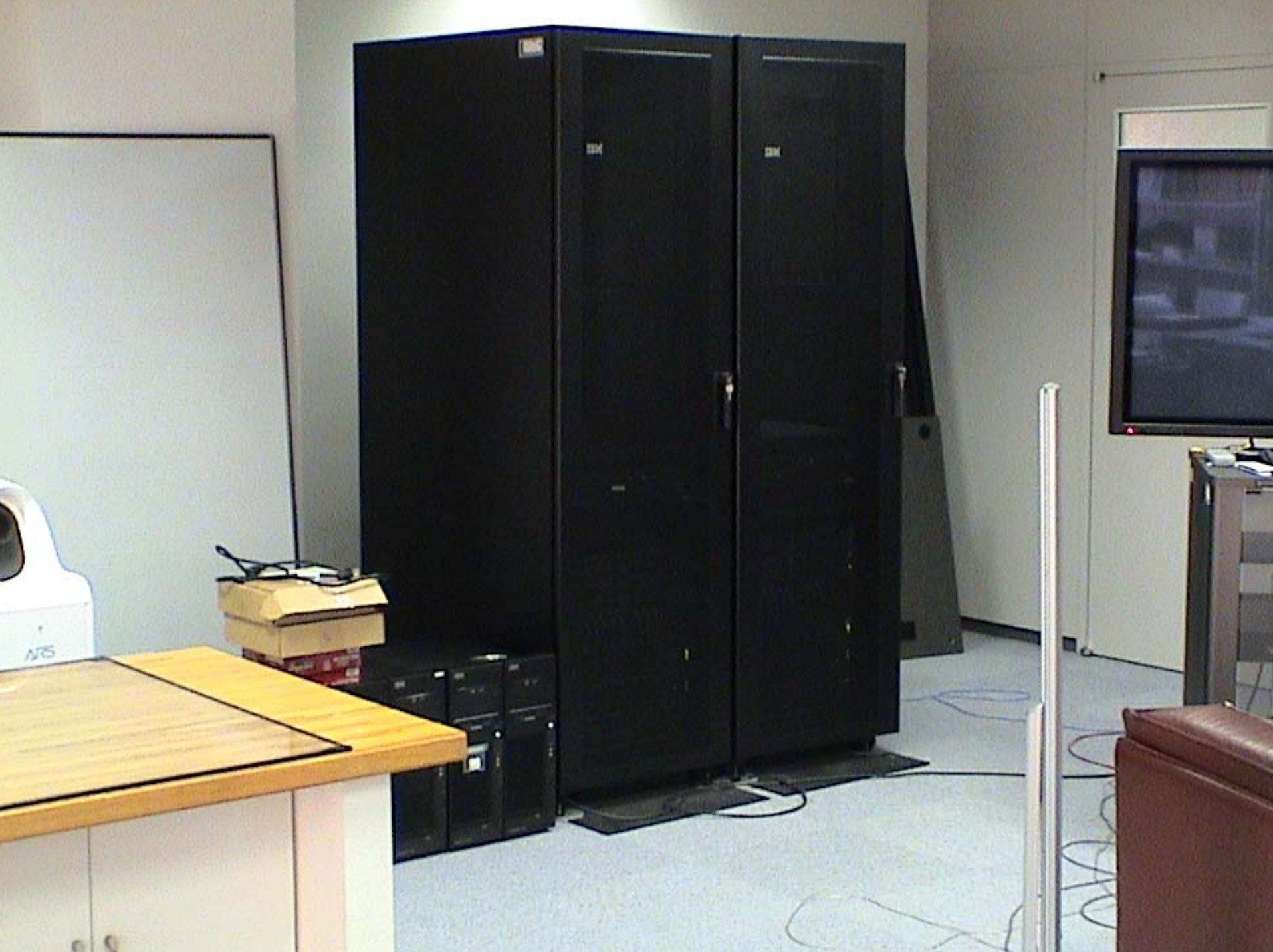
ネットワーク構成

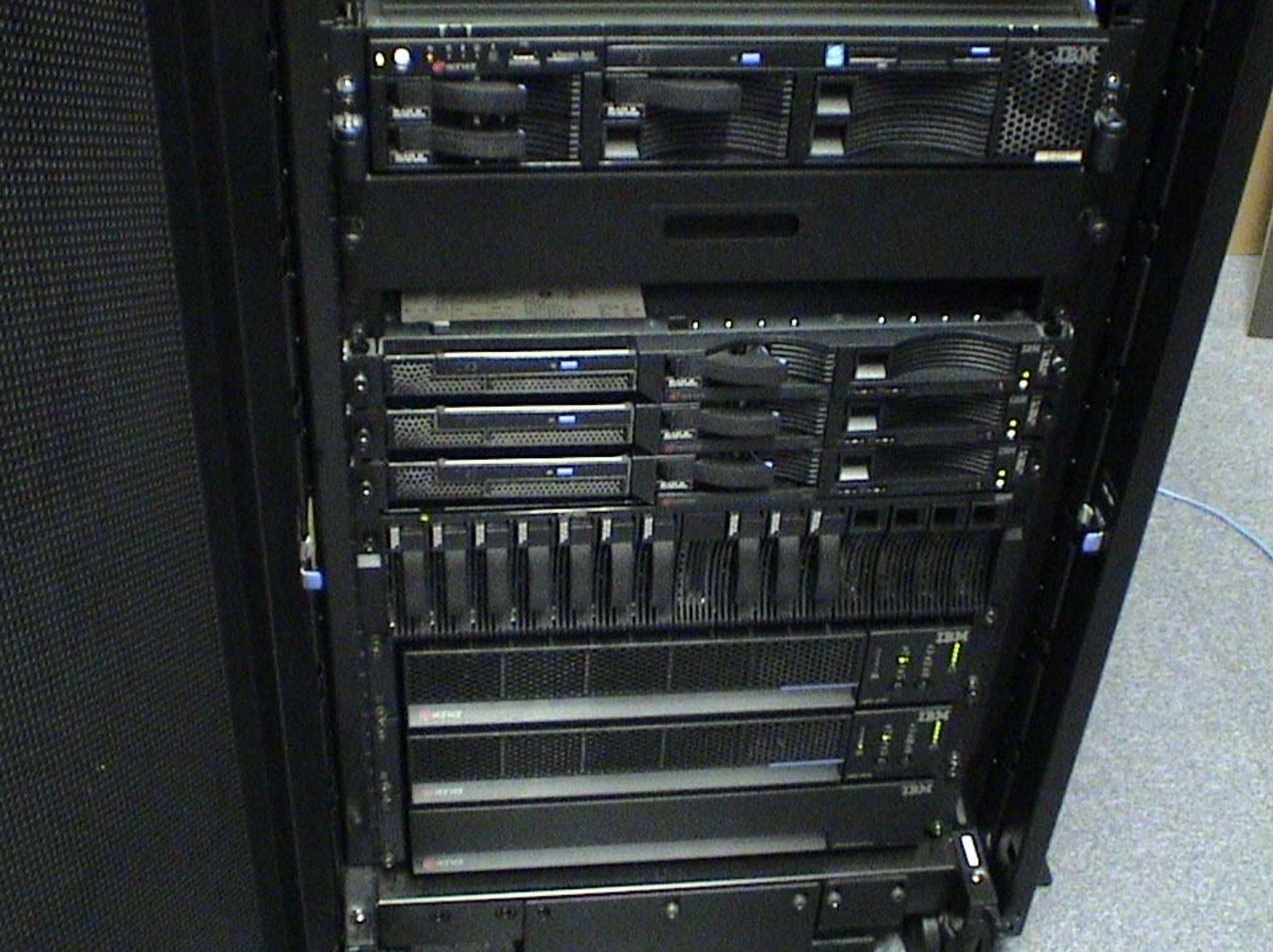


NFS V3 ＋ Samba

クライアント： Win, Diskless Linux, Linux


IBM
IBM
IBM
マウス
A-L




IBM
IBM
IBM
IBM

アプライアンスサーバのキャッシュ化

GLANTANK
- 1Gイーサ
- Xscale CPU
- 128MB メモリ

Debian + kernel 2.6.16

今後の計画

- **グループウェアの構築**
  - XUL**ベースのマルチメディア　グループウェア** Raptor
- **工科大ケータイの利用**
  - **キャッシュとしての携帯電話**