VA Linux Kernel Forum
～ (4) NFS ～

2002年5月29日

VA Linux Systemsジャパン(株)

箕浦 真

VA LINUX SYSTEMS JAPAN

VA Linux Kernel Forum ～ NFS

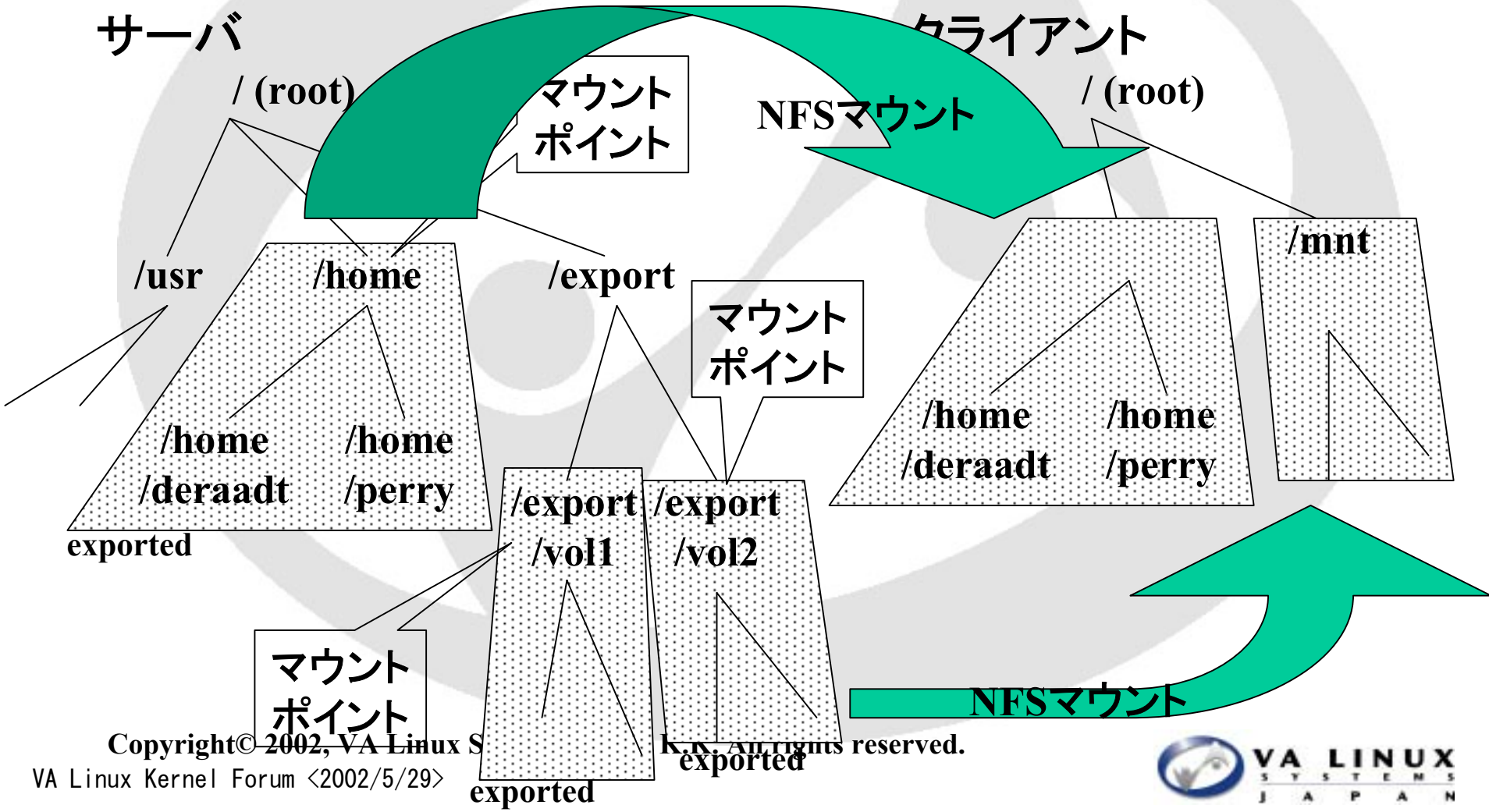
- NFSプロトコル概要
- NFSの実装
- LinuxのNFSサーバ実装
- Linux NFSサーバのチューニング

VA Linux Kernel Forum
～ (4) NFS ～

NFSプロトコル概要

ファイルシステムモデル (1/3)

■Unixのファイルシステムモデル



VA LINUX SYSTEMS JAPAN

ファイルシステムモデル (2/3)

- ファイルハンドル: サーバ内のファイルを一意に識別するID
- NFSのすべてのオペレーションの基本
- 永続的: 通常、マウントポイントのID (デバイス番号など)、inode番号、世代番号から合成
  - 世代番号: ファイルが削除され、inodeが再利用されたら?
- サーバの再起動などでも不変
- クライアントからは透過、サーバがすべて解釈

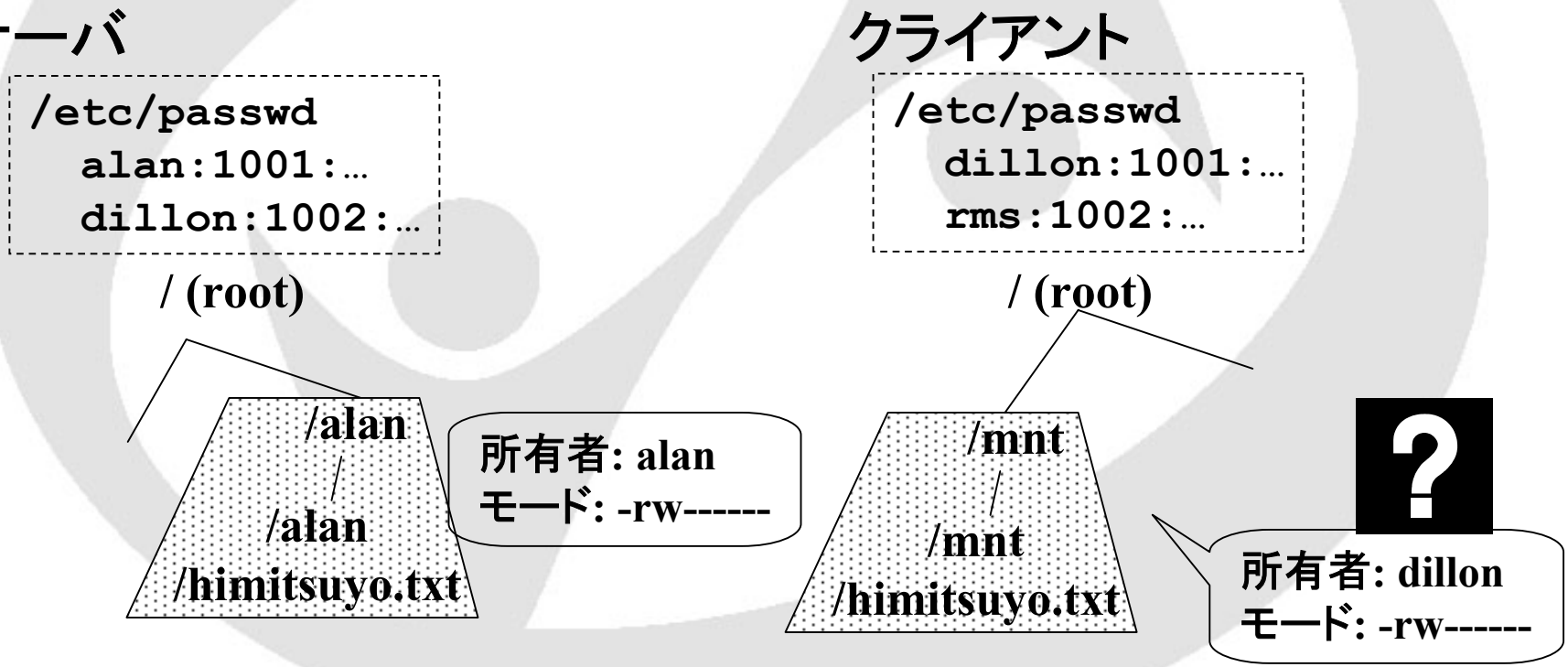
ファイルシステムモデル (3/3)

- 特殊ファイル – デバイスノード、名前つきパイプ、シンボリックリンク
  - デバイスそのものがexportされるわけではない (サーバ側のデバイスの操作はできない)
  - パイプの通信内容が転送されるわけではない (サーバ側のプロセスとの通信はできない)
  - シンボリックリンクの解釈はクライアントが行う (readlink(2)相当READLINKオペレーション)
- 参考: ディスクレスクライアント

認証・アクセス制御モデル

■AUTH_SYS認証～UnixのUID、GIDを利用

●UID、GIDの一貫性が必須



■Kerberos認証、GSS-APIによる認証

RPCメッセージ

- XID: RPC要求・返答を識別
- 要求のパラメタ
  - RPCプログラム番号
  - RPCプログラムバージョン番号
  - RPC手続き番号
  - 認証情報
- 返答のパラメタ
  - RPC実行ステータス

NFSのRPC手続き

- NULL、GETATTR、SETATTR、LOOKUP、READLINK、READ、WRITE、CREATE、REMOVE、RENAME、LINK、SYMLINK、MKDIR、RMDIR、READDIR、STATFS
- ACCESS、MKNOD、READDIRPLUS、FSSTAT、FSINFO、PATHCONF、COMMIT

NFSv3

- スケーラビリティの向上
  - 64bitファイルポインタ、ファイルシステムサイズ
  - ファイルハンドルの拡大
- ロックファイルの作成 (creat(2)の`O_EXCL`)
- 性能強化
  - 遅延書き込み
  - READDIRPLUS
- クライアント・サーバ間のネゴにより、両者のサポートしている最も高いバージョンのものを利用

Stateless性

- RPC要求と応答で全ての処理が完結 – 状態を保存する必要がない
- (特にクライアントの)実装はstateを持つ
  - ファイルポインタ、モード、…
  - キャッシュ

トランスポートプロトコル

- ONCRPC-本質的に下位プロトコル非依存
- NFS over TCP vs. UDP

	UDP	TCP
プロトコルオーバヘッド	低	高
輻輳制御	アプリケーション	トランスポート
伝送エラー検出	アプリケーション	トランスポート

- Solaris（クライアント）ではTCPがデフォルト

セキュリティ(1/2)

- クライアント認証
  - 一般に、IPアドレスを用いる
- 情報秘匿
  - (GSS-APIを使わない場合) プロトコル自体に暗号化の機能はない
- アクセス制御
  - UNIXのUID/GID/ファイルモードによる
- FWも含め、運用面で対応することが重要

セキュリティ (2/2)

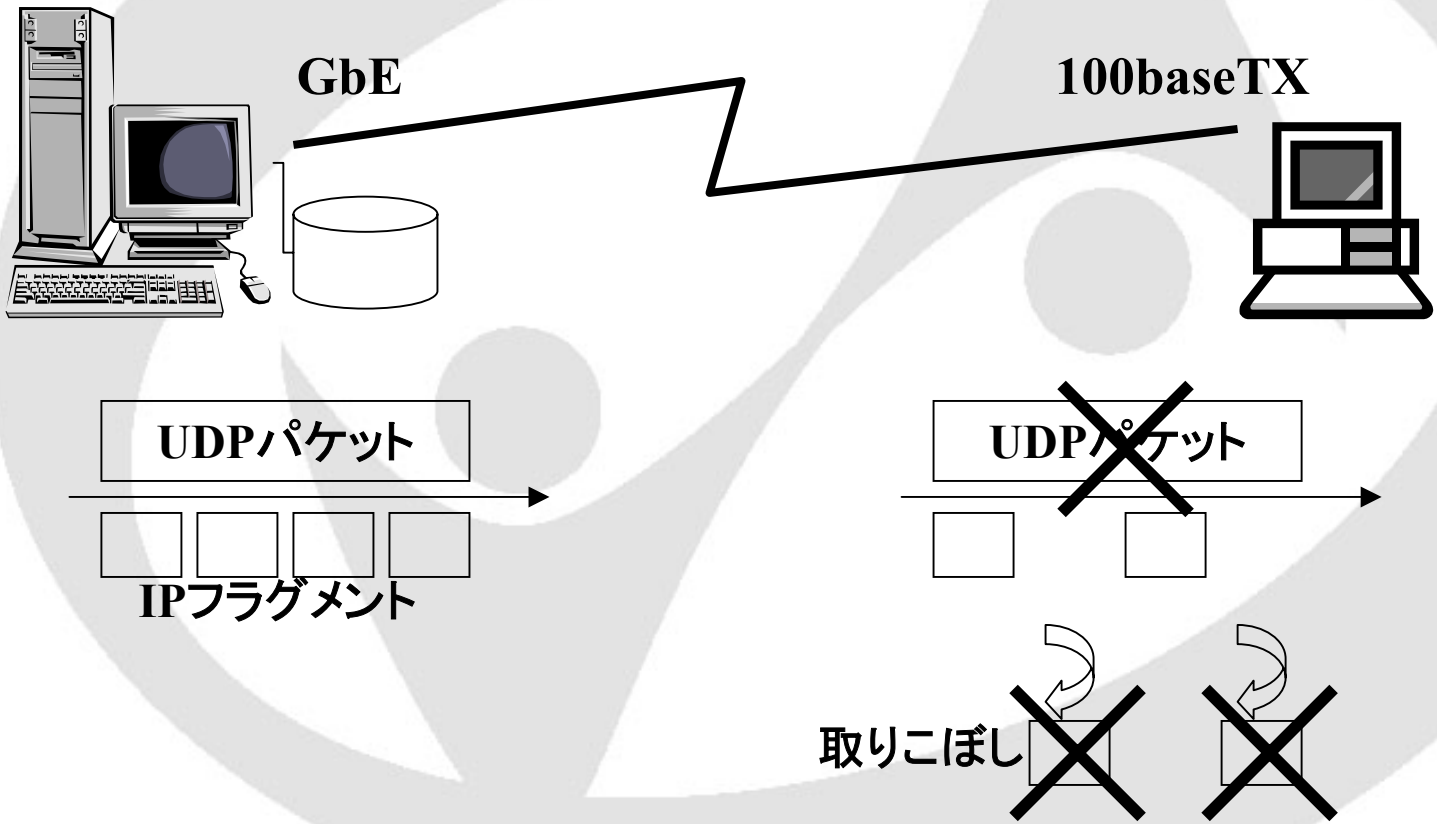
- クラック対策
  - クライアントが乗っ取られると…?
  - IPアドレス偽装されると…?
  - ファイルハンドル推測…?

VA Linux Kernel Forum
～ (4) NFS ～

実装上の問題

輻輳制御

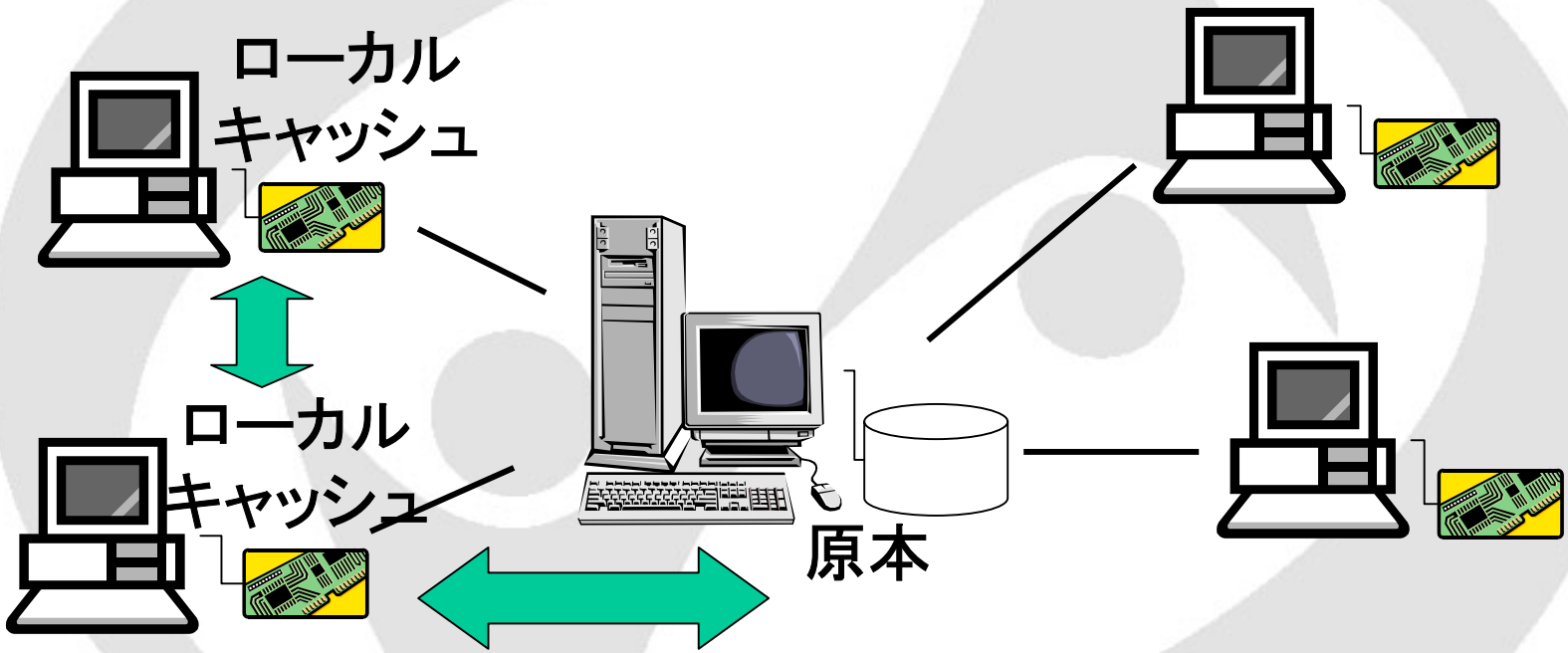
■パケットの取りこぼし



カーネルサーバ vs. ユーザランドサーバ

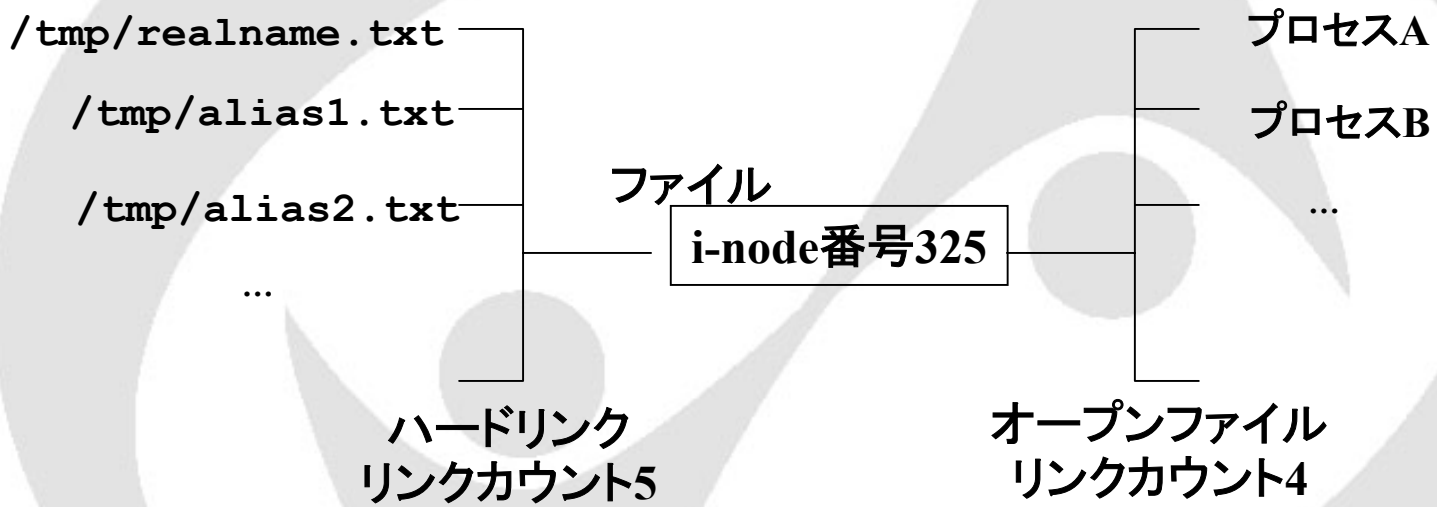
- 本来ユーザランドでサービスするのがUnix流
- パフォーマンス問題
- プロトコル上の問題
  - ファイルハンドルの永続性
  - fsid
  - open(2)相当をしない

キャッシュの一貫性問題



Last close問題

■Unixの仕様



両方のリンクカウントが0になるまでファイルの実体を消せない

VA Linux Kernel Forum
～ (4) NFS ～

LinuxのNFSサーバ実装

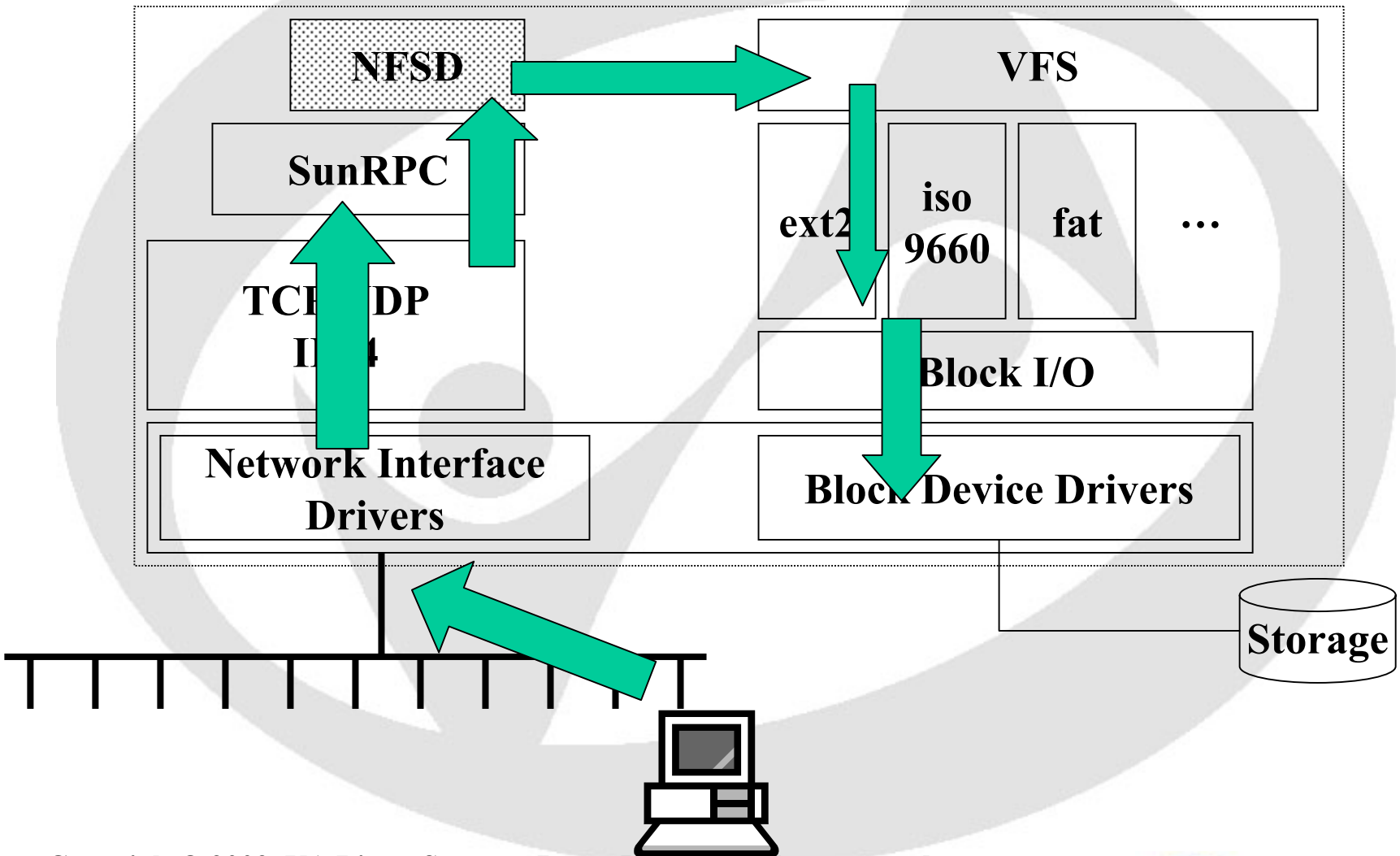
ユーザランドNFSD

- 初期の実装
- ファイル名とファイルハンドルの相互変換は不可能 ⇒仕様の完全な実装は難しい
- NFSv2の仕様をほぼ実装
- WebNFSをサポートしていた

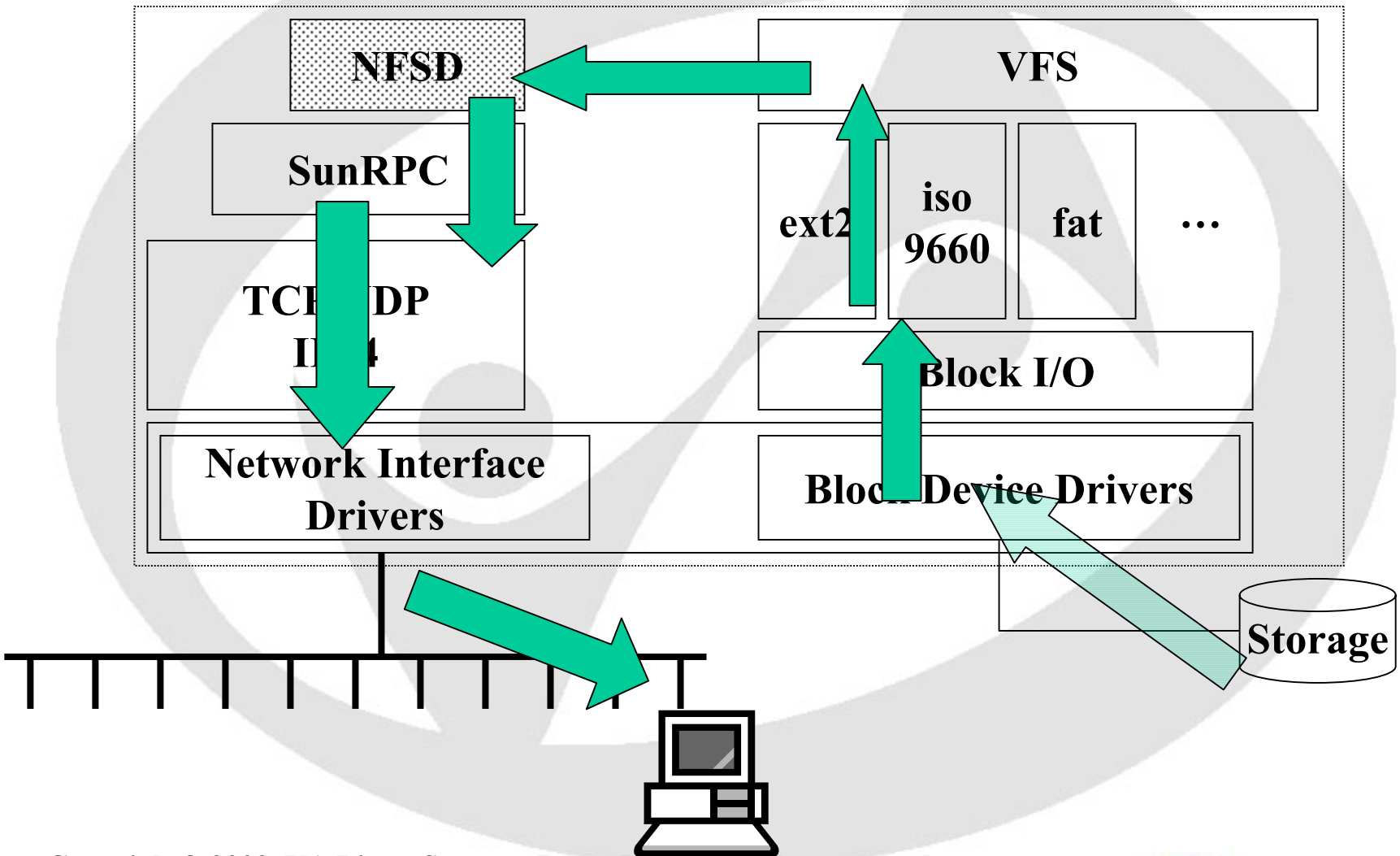
カーネルNFSD (1/2)

- 現在の実装
- すべてをカーネル内で処理 (nfsd(8) の仕事はカーネル内のnfsdスレッドをキックするだけ)
- NFSv2、v3の仕様をほぼ実装

カーネルNFSD (2/2)

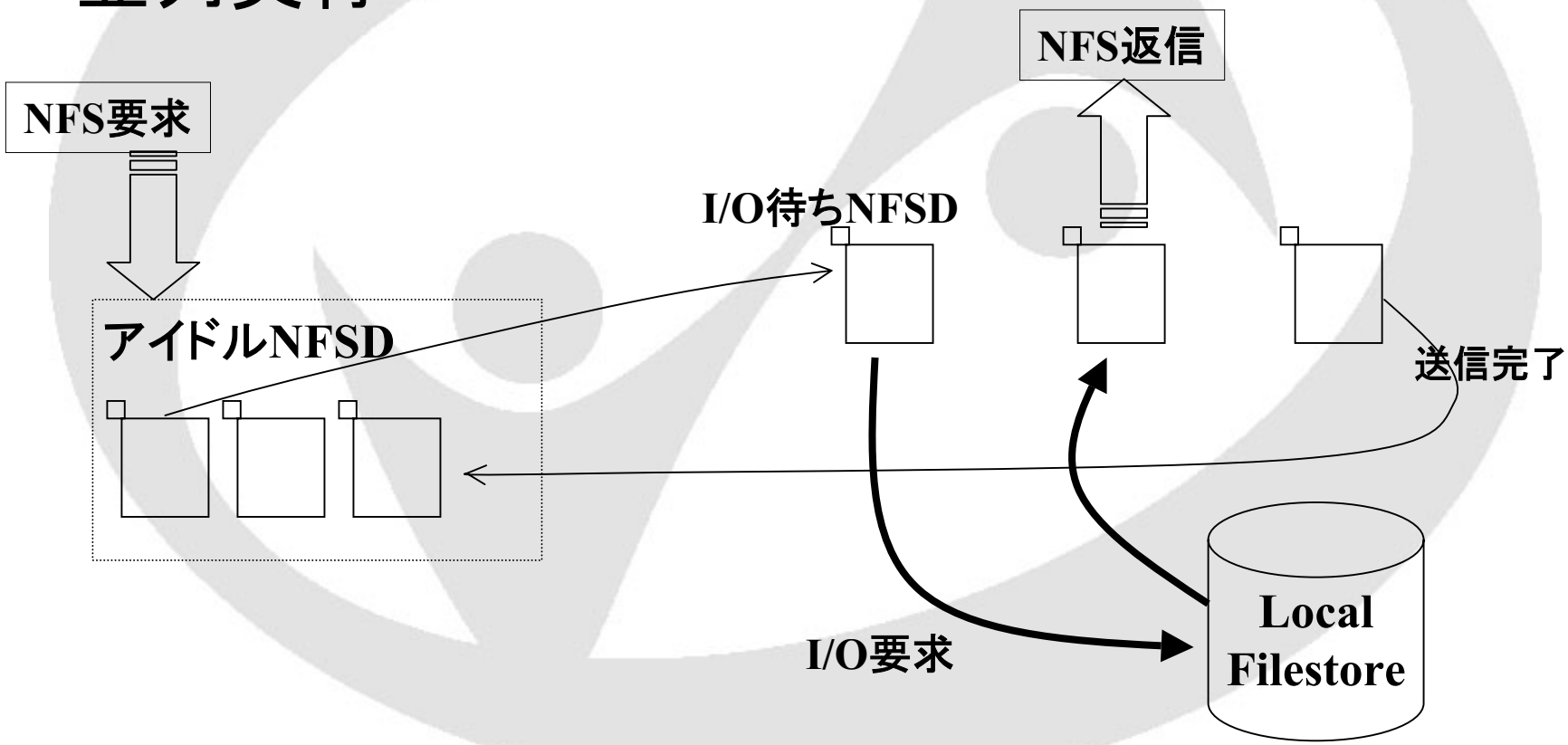


カーネルNFSD (2/2)



NFSDカーネルスレッド

■NFS要求の受信、ファイルI/O、結果の送信を並列実行



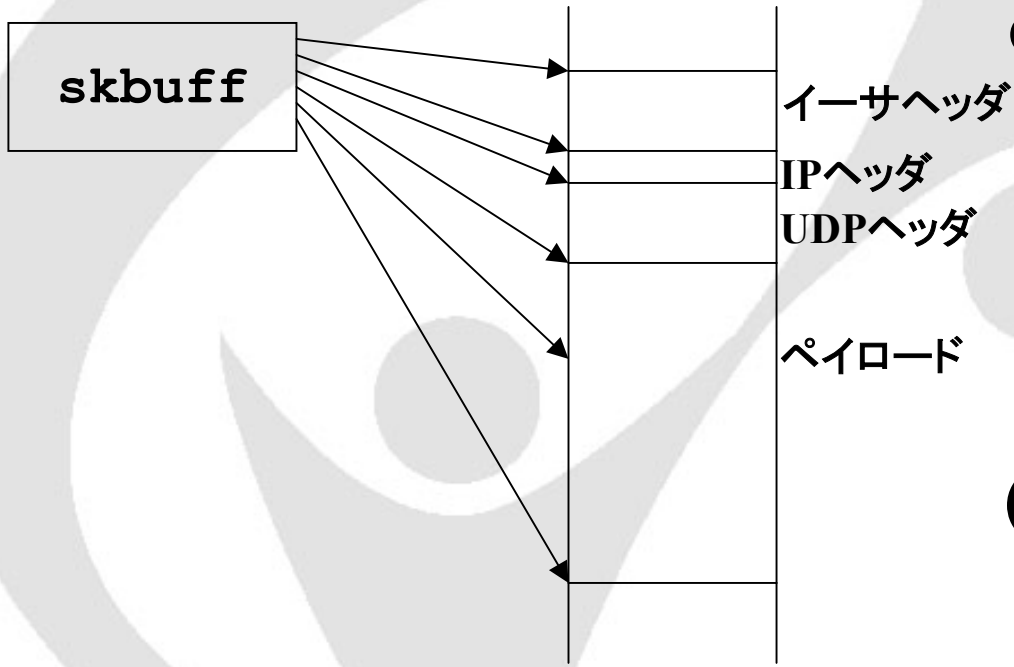
カーネルソースツリー

■2.4.18のソースツリーは140MBもあるよ

`linux/fs`	ファイルシステム
`linux/fs/ext2`	ext2ファイルシステム
`linux/fs/nfs`	NFSクライアント
`linux/fs/nfsd`	NFSサーバ
`linux/net`	ネットワーク関連
`linux/net/core`	ソケットなど
`linux/net/ipv4`	TCP/UDP/IP
`linux/net/sunrpc`	ONC RPC実装
`linux/include`	各種ヘッダファイル

ネットワークに関するデータ構造

■ struct sk_buff



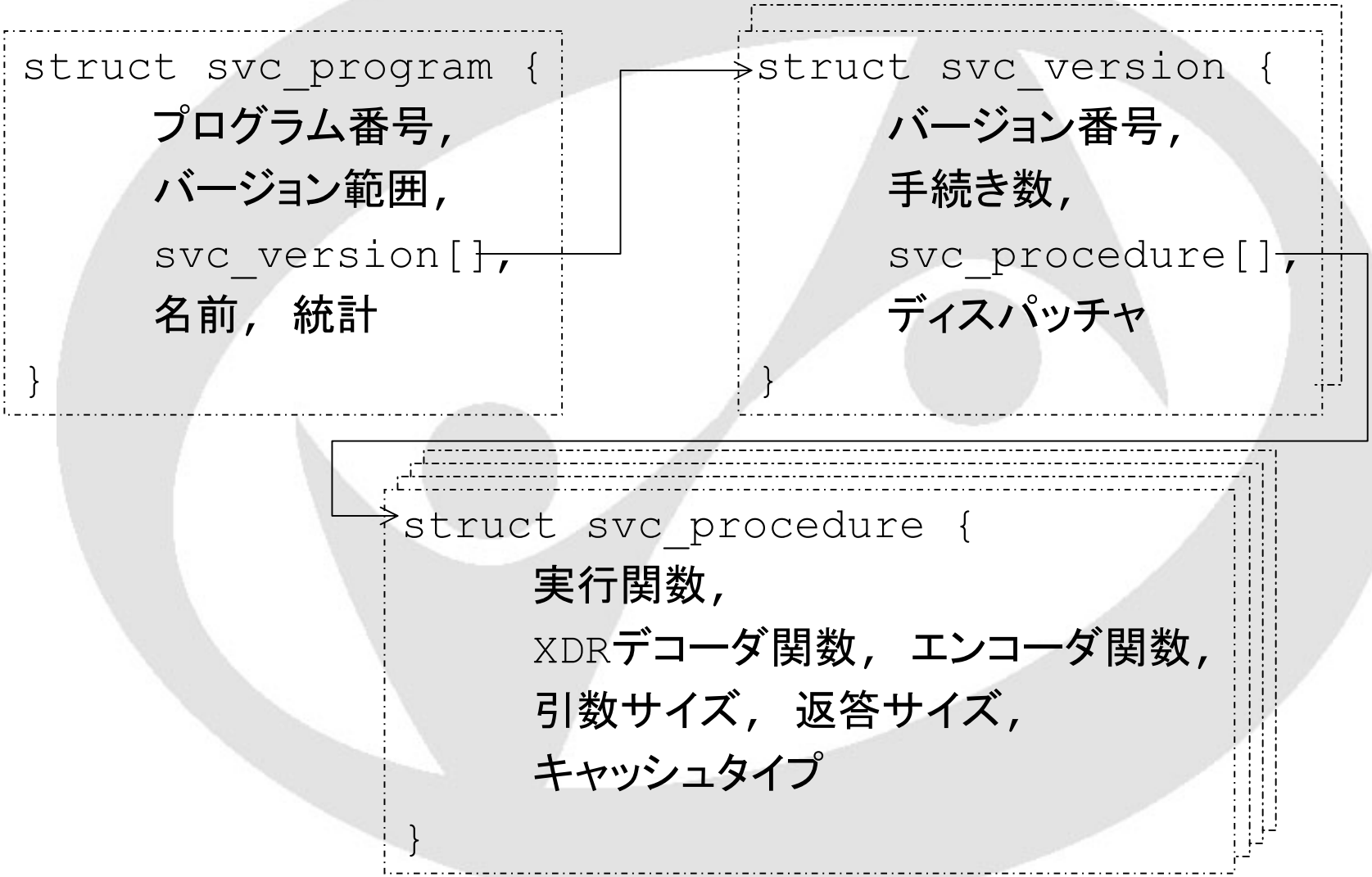
●ヘッダを読み飛ばしたり付加したりしながらレイヤ間を移動するパケットを表す

(cf. mbuf)

ONC RPCレイヤのデータ構造

- svc_serv
  - RPCサービス (例: NFS)
  - ディスパッチルーチンへの (関数) ポインタ群など
- svc_rqst
  - 処理中のリクエスト (idleも含めてI/Oスレッドの数だけ確保される)
- svc_sock
  - RPC用のソケット

ONC RPCレイヤのデータ構造



カーネル2.5

- SMP対策 ～ ジャイアントロックの削減
- 輻輳制御
- データサイズ拡大（8KB→32KB）
- NFS over TCP（サーバ側）の実験的サポート

今後の課題

- コピー回数削減 (⇒ Zero-copy)
- 相性問題
- Kerberos認証
- WebNFS、NFSv4

VA Linux Kernel Forum
～ (4) NFS ～

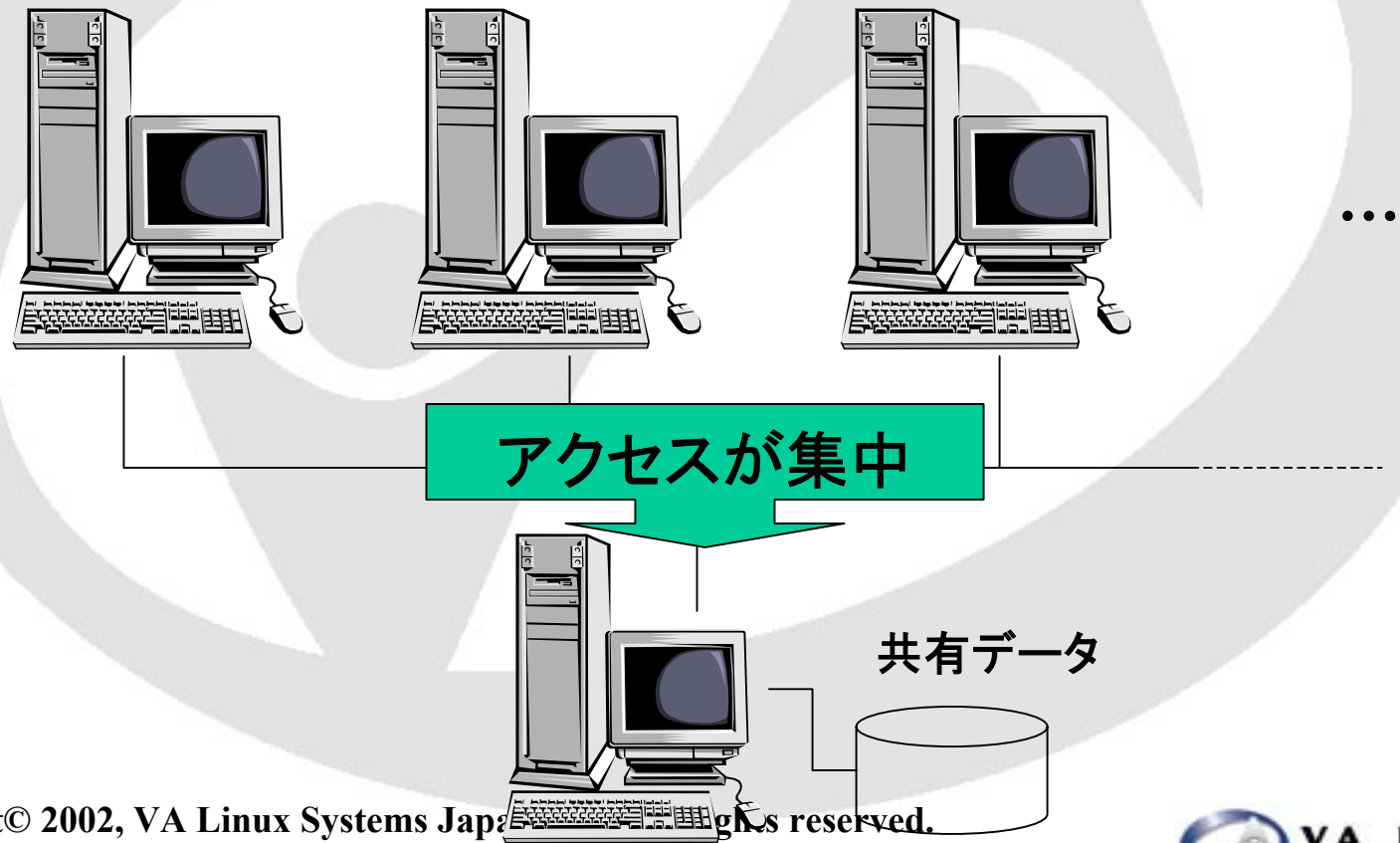
Linuxカーネル NFSサーバの
チューニング

VA LINUX SYSTEMS JAPAN

チューニングの前に (1/2)

■システム全体のボトルネックの調査

●NFSサーバはボトルネックになりやすい



VA LINUX SYSTEMS JAPAN

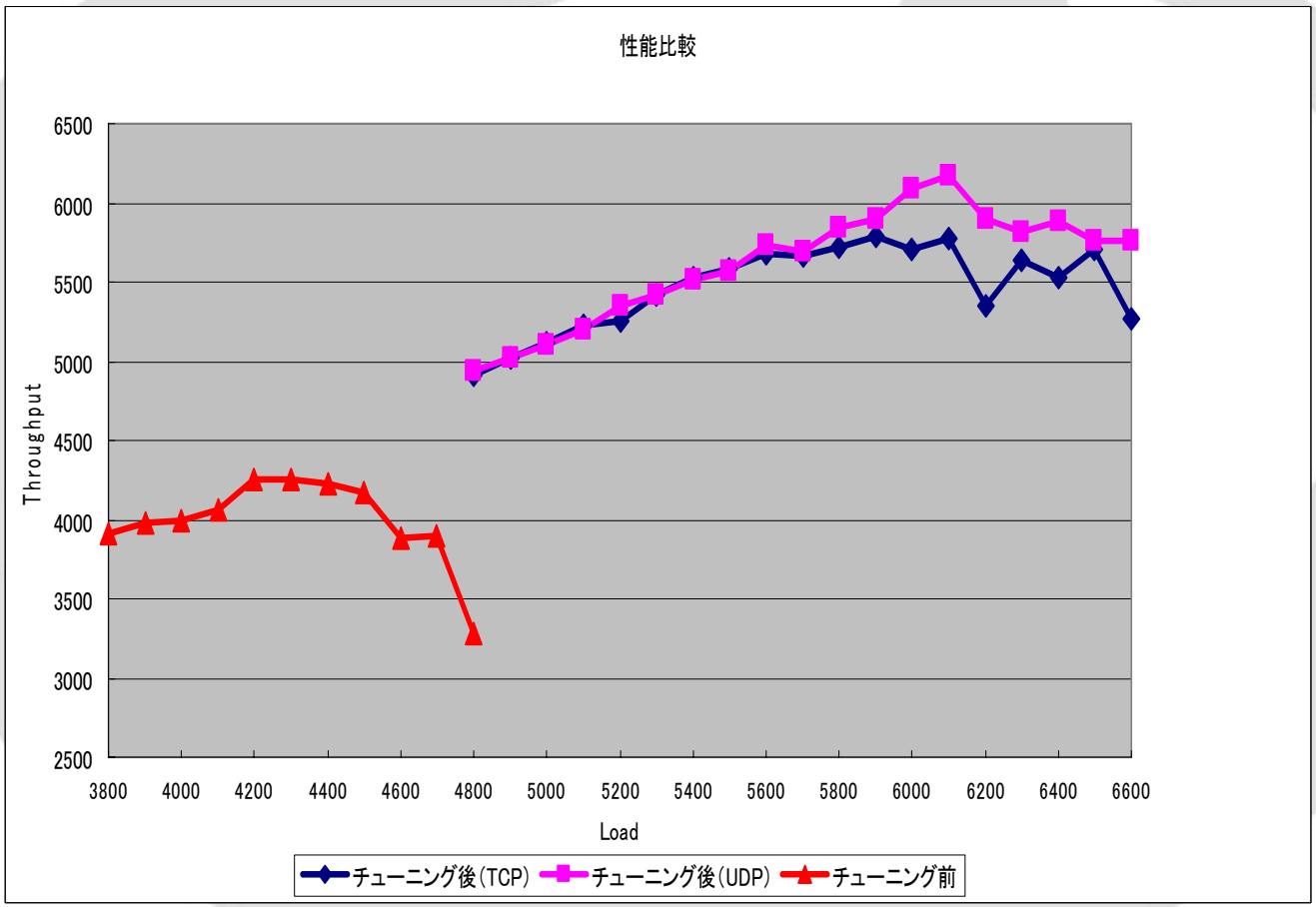
チューニングの前に (2/2)

- NFSを無駄に酷使していないか?
  - ローカルに持てる情報ではないか?
  - サービスは最適に配置されているか?

VA Japanでのチューニング事例

- 50項目以上のチューニング
  - 資源割り当て量など各種パラメタの変更
  - 資源割り当てポリシの変更
  - その他のアルゴリズム変更
  - その他高負荷対策など
- 幅広いチューニング対象
  - nfsd、sunrpc
  - ネットワークスタック
  - ファイルシステム
  - スケジューラ
  - 仮想メモリシステム、カーネル空間管理システム

NFSサーバ性能



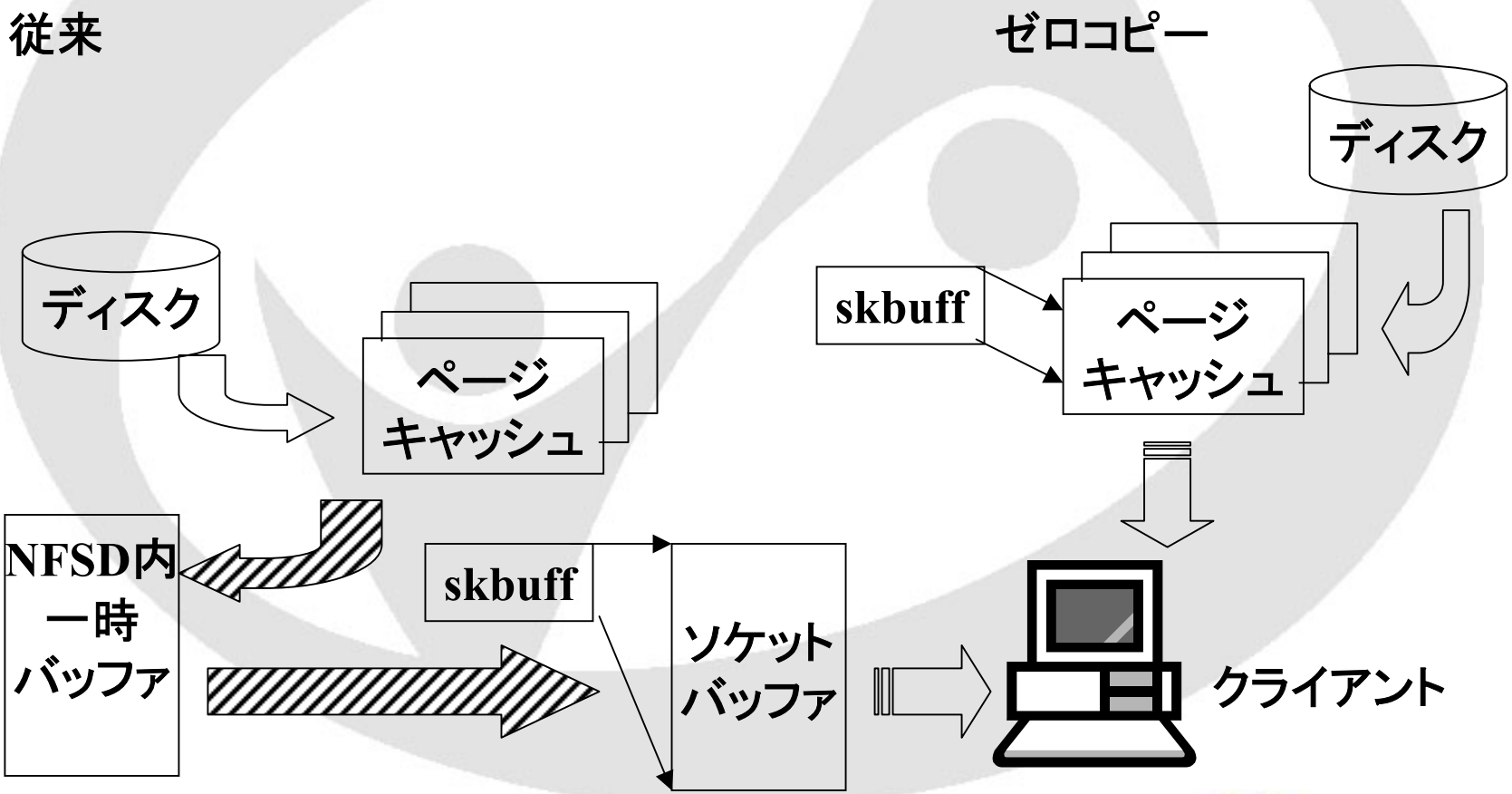
VA LINUX SYSTEMS JAPAN

カーネルチューニングの指針

- 資源割り当てポリシー・割当量の変更
  - ⇒NFSで使う資源の拡大、予約
- アルゴリズムの変更
  - Slab、ハッシュなどの仕組みの活用
  - 遅延評価の活用
  - 資源の有効活用

ゼロコピーNFSサーバ

■不必要な大量のメモリ→メモリコピーを削減



その他必要な機能の追加

- ブロックデバイスのブロック番号の拡張
  - 32bit ⇒ 64bit
  - 2TBを超えるディスクの接続
- NFSD over TCPのサポート
  - 2.5.x（実験的サポート）からのバックポート+バグフィックス
- 信頼性向上