HITACHI
Inspire the Next

CE Linux Forum Japan Technical Jamboree #27 2009/5/22

組込みLinux開発環境への ATA over Ethernet適用

(株)日立製作所
組込みシステム基盤研究所
茂岡　知彦

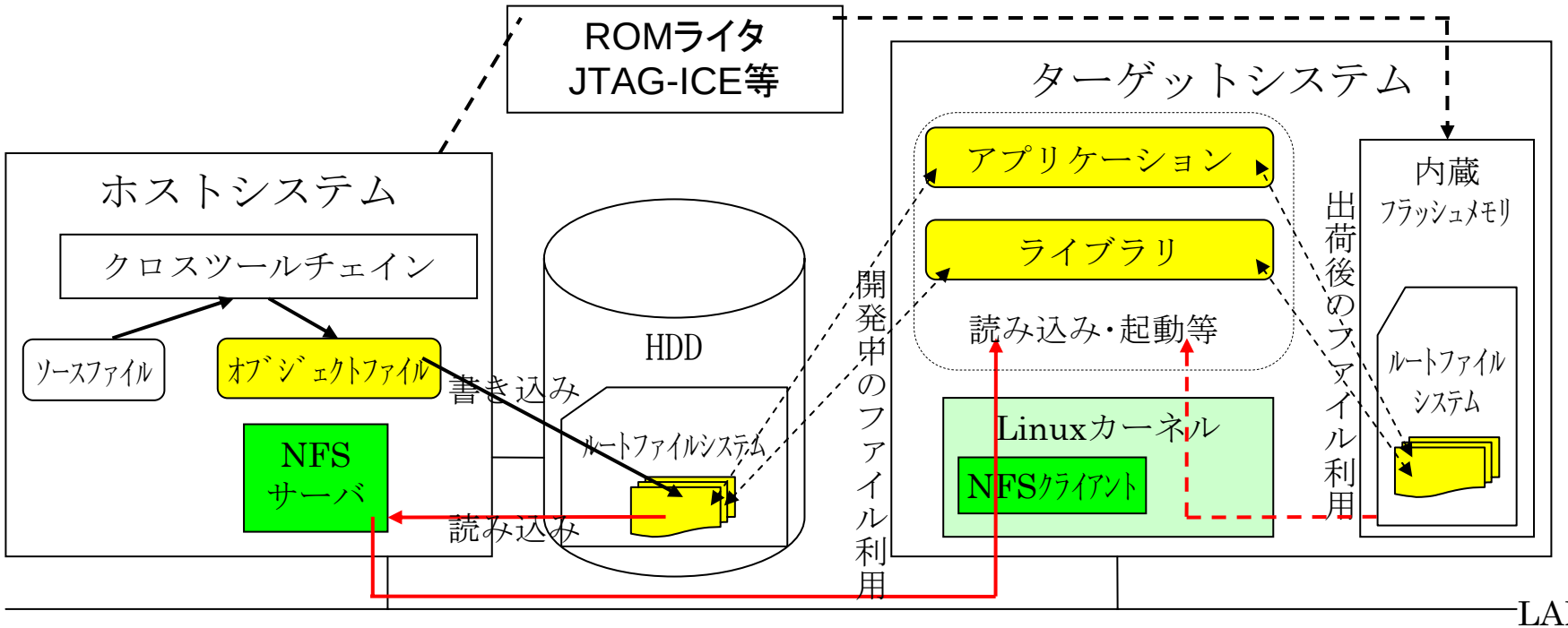
背景

HITACHI
Inspire the Next

- 組込みLinuxの開発環境でのホストとターゲットのファイル共有(rootfs)はNFSが一般的
- 機器への組込み時はフラッシュメモリにファイルシステムを構築
  - ext***N***以外のファイルシステムを用いることが多い
  - cramfs/squashfs/jffs2/ubifsその他
- フラッシュメモリに置いたファイルの修正は手間がかかる(ことが多い)
  - ROMライタ、JTAG-ICEなどによる書き込みは面倒
  - 失敗するとさらに手間が発生

一般的なNFSを使った開発環境

- ホストシステム上でクロスツールチェインを用いてソフトウェアを開発
- ホストにあるファイルツリーはNFS経由でターゲットのルートファイルシステムとして利用
- フラッシュメモリへは別途ファイルシステムイメージを生成して書き込み



NFS使用時の問題点

HITACHI
Inspire the Next

NFSは長年(※)の実績もあり、NFS環境での開発は便利だが問題点も・・・

- 開発後(運用時)と異なるソフトウェア構成のため、挙動が異なることがある
- ファイルシステムに起因する問題のデバッグができない、やりにくい

※NFSv2をSunが一般に発表したのは1984年、LinuxのNFS実装はkernel 1.2ごろから

ブロックデバイス層のネットワーク対応

HITACHI
Inspire the Next

- ファイルシステム層ではなくブロックデバイス層でネットワーク接続

  →フラッシュメモリ上と同じファイルシステムを利用可
- ネットワーク接続可能なブロックデバイス各種
  - ATA over Ethernet(AoE)
  - iSCSI(Internet SCSI)
  - Network Block Device(NBD)

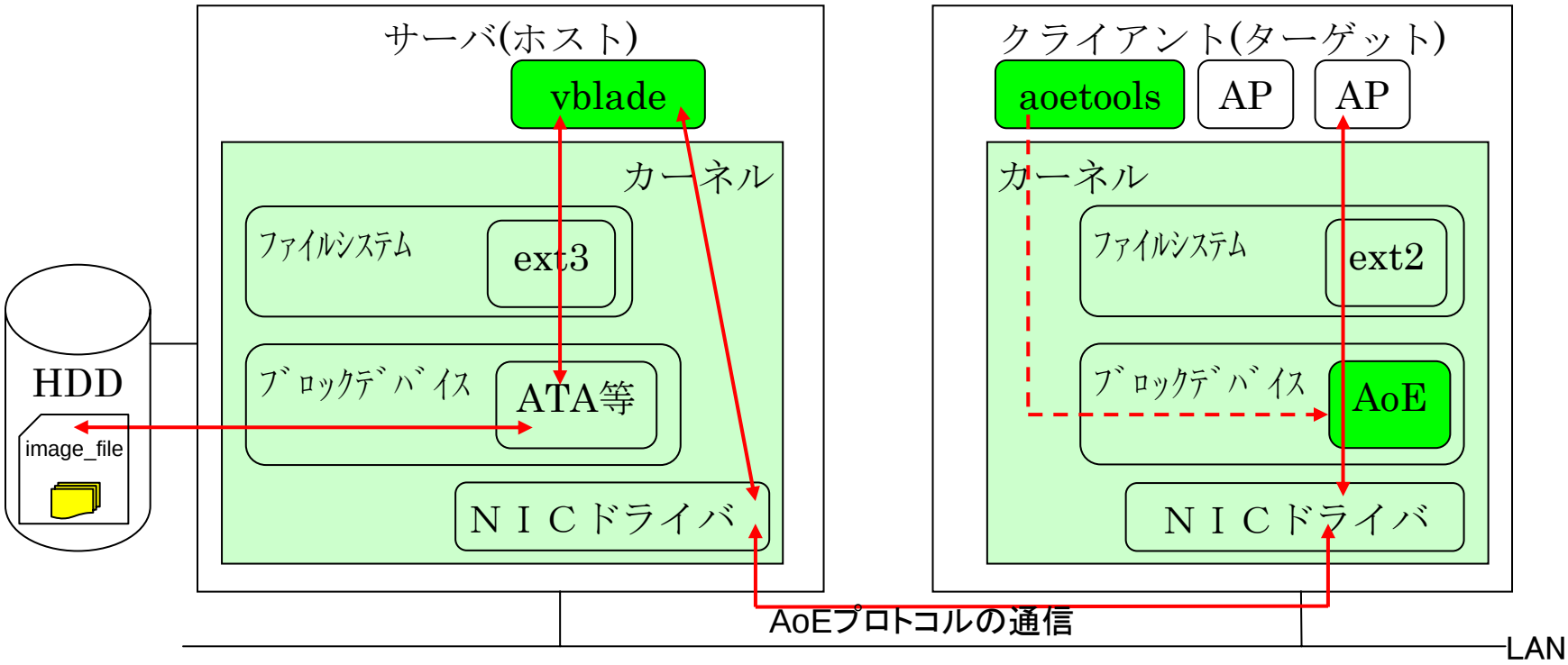
ATA over Ethernet(AoE)概要

HITACHI
Inspire the Next

- Coraid.comが主導する低コストSAN技術
- ATAコマンド/データをイーサネットのフレームに載せて通信する軽量なプロトコル(TCP/IP未使用)
- kernel 2.6.11からクライアント側ドライバがメインラインに含まれる(drivers/block/aoe/)
- クライアント側は設定コマンドあり(aoetools)
- サーバ側は専用ハードやソフト実装で、ユーザランドデーモン、カーネルドライバなど
  - vblade, kvblade, qaoed, etc
- サーバ側はATAである必要はない
- 各種OSやブートローダでのサポートあり

AoEのシステム構成図

- サーバのvbladeデーモンがHDD上のディスクイメージファイルを公開
  ```
  # vblade 0 0 eth0 image_file
  ```
- クライアントのAoEドライバはサーバの公開するイメージをブロックデバイスとしてファイルシステム層に見せる
  ```
  # mount -t ext2 /dev/etherd/e0.0 /mnt
  ```

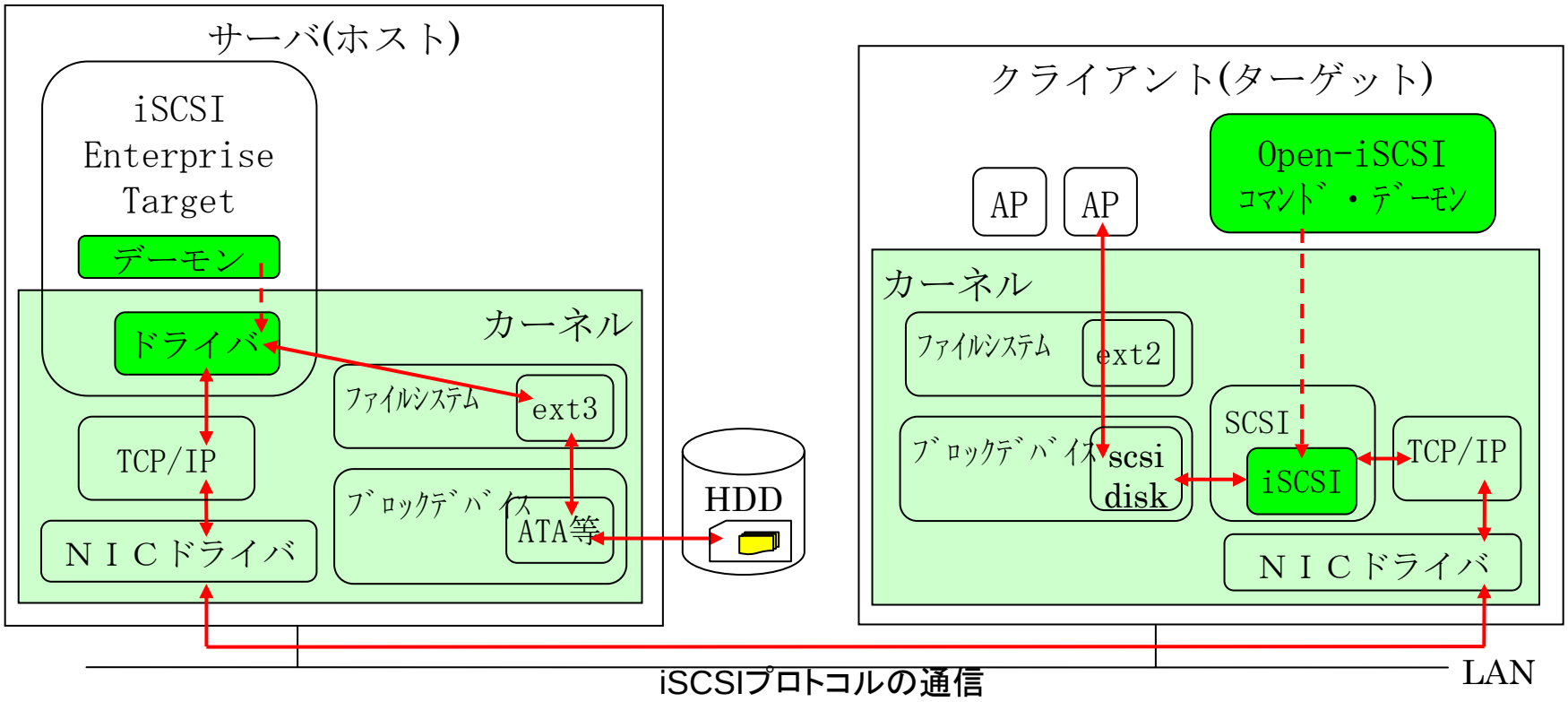


iSCSI概要

- エンタープライズ分野で普及するSAN技術
- SCSIのコマンド/データをTCP/IPに載せて通信するプロトコル
- kernel 2.6.12からクライアント側ドライバ(イニシエータ)がメインラインに含まれる(drivers/scsi/iscsi_tcp.c)
- ブロックデバイスは通常のSCSIデバイスと同じI/Fで利用可能(/dev/sda等)
- クライアント側はユーザランドツール(Open-iSCSI)での設定が必須
- サーバ側はハイエンドの専用ハードだけでなく、ソフトウェア実装各種あり(iSCSI Enterprise Target等)

iSCSIのシステム構成図

- サーバのiSCSI TargetでHDD上のディスクイメージファイルやパーティションを公開
- クライアントのiSCSIイニシエータでサーバのTargetに接続、ディスクイメージをSCSI DISKブロックデバイスとして見せる



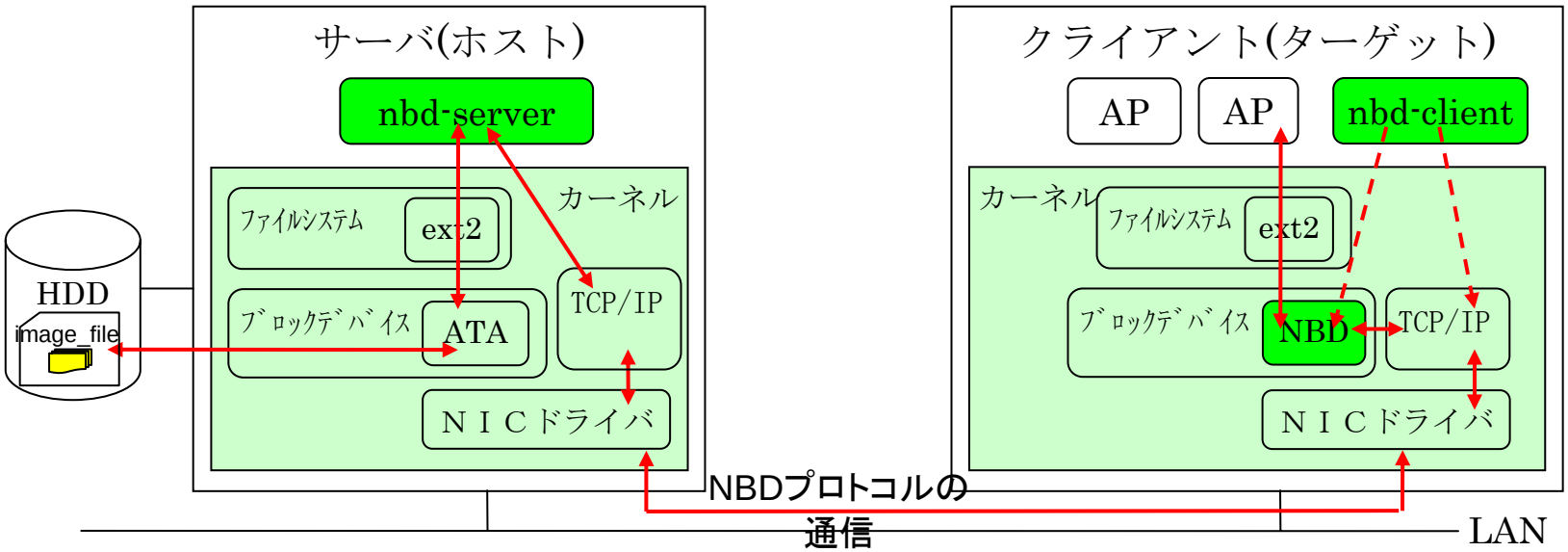
Network Block Device(NBD)概要

- TCP/IP経由でサーバのデータをブロックデバイスとして扱う
- ATAコマンドと無関係なプロトコル(圧縮機能あり)
- kernel 2.1.101からメインラインにクライアント側ドライバが含まれる(drivers/block/nbd.c)
- クライアント側はコネクション管理デーモン必須(nbd-client)

```
# nbd-client server 2000 /dev/nb0
```

- サーバ側はソフトウェア実装(nbd-server)

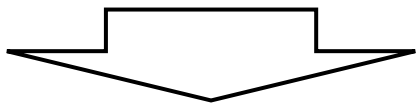
```
# nbd-server 2000 image_file
```



各方式によるrootfsの利用

HITACHI
Inspire the Next

- NFSとほぼ同じ形態が可能なのはAoEのみ
  - AoE rootパッチが必要 (http://support.coraid.com/support/linux/contrib/mcmullan/aoeroot-2.6.15.diff)
  - ↑はkernel configでrootにするAoEサーバを指定
  - カーネルコマンドラインで指定できない(root=???)
  - コマンド(aoetools)による設定は不要
- iSCSIやNBDはコマンドでセットアップが必要
  →initramfs/initrdを利用してセットアップ処理
  - 直接rootfsとしてマウントできない、セットアップ処理付のinitramfs/initrdを追加する手間が発生



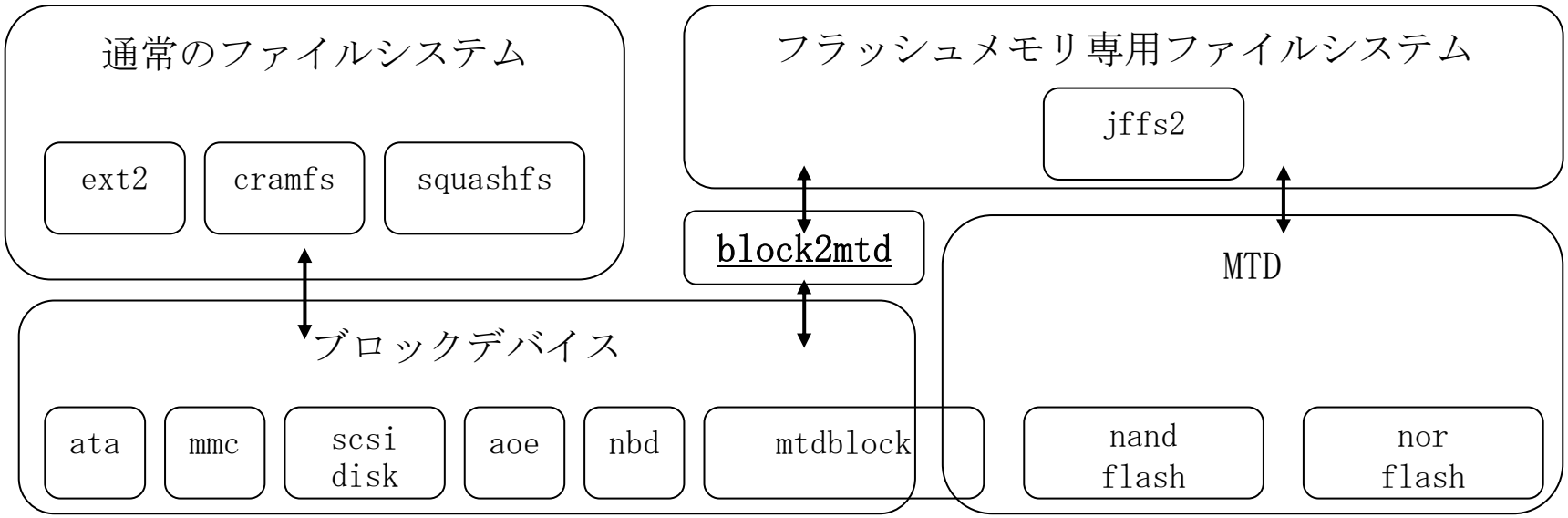
今回はAoEを適用

MTDエミュレーション

- 一般fs(ext3/cramfs)等はブロックデバイスが前提
- jffs2等はブロックデバイスではなくMTD前提
- block2mtdによりMTDインタフェースが利用可

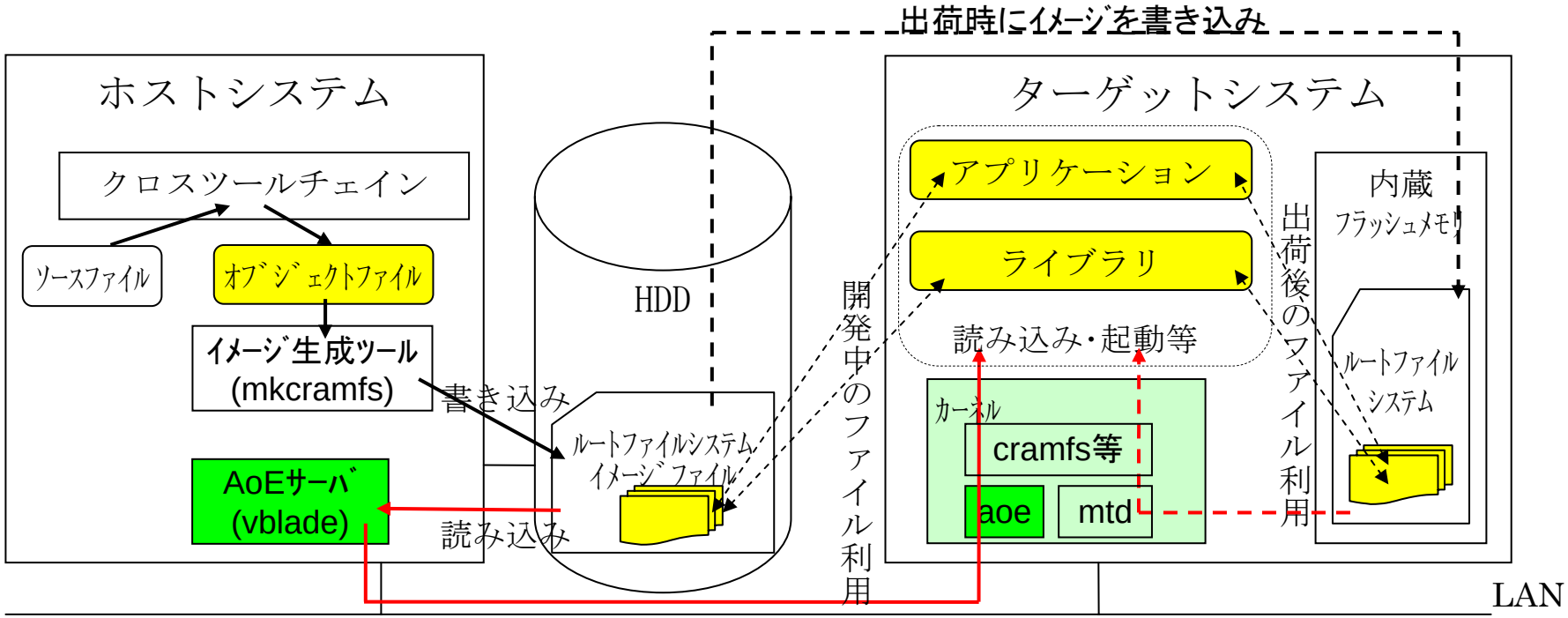
カーネルコンフィグCONFIG_MTD_BLOCK2MTD=m

```
# insmod block2mtd.ko block2mtd=/dev/hda1
# mount -t jffs2 /dev/mtdblock0 /mnt
```



AoEの組込み開発環境への適用

- NFSと同様にクロスツールチェインでソフトウェアを開発
- ターゲットでの実行にファイルシステムイメージの作成が必要
- フラッシュメモリへ書き込むファイルシステムイメージは、AoEを使えばそのままネットワーク経由でルートファイルシステムに利用可能



AoE適用の評価

HITACHI
Inspire the Next

- 評価項目(AoE/NFS/フラッシュで比較)
  - ターゲットカーネルのオブジェクトサイズ
  - 同一内容rootfsでの起動時メモリ使用量
  - 圧縮ファイルシステム(cramfs)での挙動比較
    - 同一データ量で、圧縮率の高いファイル(all 0)と低いファイル(ﾗﾝﾀﾞﾑﾃﾞｰﾀ)の読み出し時間

- 評価環境
  - ターゲットはOSK(TI OMAP 192MHz　RAM32M), kernel 2.6.20.20, 10BASE-T
  - ホストはx86_64(3.4GHz), kernel 2.4.21(RHEL3), 1000BASE-T

AoE適用の評価(サイズ・メモリ)

HITACHI
Inspire the Next

- ターゲットカーネルオブジェクトサイズの比較

AoE		NFS	フラッシュ	
ファイルシステム (cramfs)	aoe	nfsクライアント	ファイルシステム (cramfs)	mtd
10,433	28,326		10,433	73,401
	38,759 ≪	322,503		≫ 83,834

各ディレクトリのbuilt-in.oのサイズ(バイト)

- メモリ使用量の比較
  - 同一内容rootfs(busybox)で起動後ログインしfreeコマンド実行

方式	total	used	free	buffers
AoE	30,220	5,000	25,220	724
NFS	29,888	4,352	25,536	0
フラッシュ	30,188	5,040	25,148	736

freeコマンド出力(単位キロバイト)

AoE適用の評価(挙動)

- ファイル読み出し所要時間(CPU時間)
  - echo 3 > /proc/sys/vm/drop_caches後time wc -cで読み出し

方式＼データ	all 0データ			ランダムデータ		
	real	user	sys	real	user	sys
AoE	1.57	0.71	0.63	7.09	0.66	1.14
NFS	6.28	0.72	0.88	6.39	0.72	0.86
フラッシュ	1.44	0.72	0.61	3.23	0.69	0.88

単位: 秒(5MB読み出し10回平均)

NFSのみほぼ同一

- データ転送量
  - /proc/diskstatsおよび/proc/net/devの差分で測定

方式＼データ	all 0データ		ランダムデータ	
	eth受信	セクタread	eth受信	セクタread
AoE	352,040	339,968	5,770,820	5,570,560
NFS	5,666,068	−	5,663,699	−
フラッシュ	−	339,968	−	5,570,560

単位: バイト(5M読み出し時)

NFSのみほぼ同一

考察

- カーネルCPU時間は伸張処理の違いを反映
  - 高圧縮なデータ(all 0)の伸張はランダムデータに比べてCPUを使わない(※データ転送はCPUをあまり食わないと仮定、もしデータ転送にCPUをかなり食われているならメモリコピー量の違いも反映)
  - NFSの場合は伸張処理がないためデータ内容による違いがない(※データ転送量も同じ)
- データ転送量は転送するデータの内容を反映
  - 圧縮されたデータを転送するため、高圧縮なデータを転送するときはデータ転送量が少ない
  - NFSの場合は圧縮がかかっていないのでデータ転送量は同じ

まとめ

- 組込みLinuxシステムでもAoEは利用可能
- AoEによりNFSで出来ないデバッグが可能
- AoEはNFSよりオブジェクトが小さい
- AoEの使い勝手は発展途上

参考文献

HITACHI
Inspire the Next

- http://www.coraid.com/RESOURCES/AoE-Protocol-Definition
- http://sourceforge.net/projects/aoetools/vblade
- http://www.open-iscsi.org/
- http://iscsitarget.sourceforge.net/
- http://nbd.sourceforge.net/