

B-019

Mint における仮想ネットワークインタフェースによる OS 間通信 A Method for Inter-OS Communication Using Virtual Network Interface on Mint

小倉 伊織[†]
Iori Ogura

乃村 能成[†]
Yoshinari Nomura

吉田 修太郎
Shutaro Yoshida

谷口 秀夫[†]
Hideo Taniguchi

1. はじめに

1 台の計算機上で複数の Linux を独立に走行させる方式として Mint オペレーティングシステム (以降, Mint) が研究開発されている [1]. 本稿では, Mint 上の Linux 間で通信するために Ethernet 互換の仮想ネットワークインタフェース (以降, VNI) を実装した. 以降では, VNI による Ethernet 互換の通信方式の詳細について述べる. また, この通信方式における通信の基本性能を示す.

2. Mint オペレーティングシステム

Mint は, マルチコアプロセッサ上で複数の Linux を独立に走行させる方式である. Mint 上で走行するすべての Linux は, 依存関係が存在せず, 処理負荷の影響を与えない. Mint の構成例を図 1 で示し, 以下で説明する. Mint では, CPU やメモリ, 入出力機器等の計算機資源を分割し, 各 Linux に占有させる. 一方で, 共有メモリはすべての Linux で共有させる.

3. 仮想ネットワークインタフェースによる OS 間通信

3.1 設計

Mint における Linux 間通信方式 (以降, 提案方式) の設計方針について以下に示す.

(方針 1) 標準的な通信プロトコルに対応

提案方式を標準的なプロトコルに対応させる. 具体的には, 提案方式における Linux 間通信を Ethernet 互換で行う. これにより, Linux 既存の App の改変なしに Linux 間で通信可能である.

(方針 2) 共有メモリを介した通信

Linux 間で通信するには, すべての Linux が参照可能な資源を介して通信する必要がある. そこで, Mint が持つ共有メモリを介して通信する.

提案方式における Linux 間通信について図 2 に示し, 以下で説明する. 提案方式では, Linux が持つ既存のプロトコルスタックを用いることで, App からのデータを Ethernet パケットに変換する. VNI は, この Ethernet パケットを共有メモリを介してやり取りする. また, VNI は IPI を用いて, 受信側の Linux に Ethernet パケットの送信が完了したことを通知する.

3.2 課題

3.1 節の設計を実現する上での課題を以下に示す.

(課題 1) パケットバッファの構成の検討

パケットバッファの構成として, すべてのパケッ

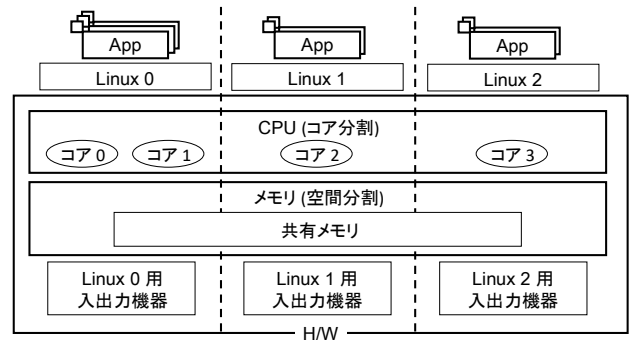


図 1 Mint の構成例

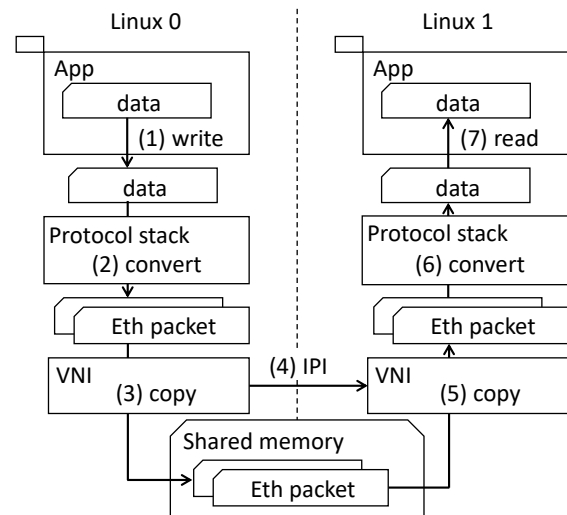


図 2 提案方式における Linux 間通信

トを同一のパケットバッファで管理する構成とパケットを通信路ごとに別々のパケットバッファで管理する構成の 2 つが想定される. これらの構成の内, どちらが提案方式に適しているか検討する.

(課題 2) 排他制御すべき処理の検討

複数の Linux が同時に共有メモリを使用する. この際の排他制御が最小となるように排他制御すべき処理を検討する.

3.3 対処

3.3.1 パケットバッファの構成

想定される 2 つのパケットバッファの構成を図 3 に示し, 以下で説明する.

(構成 1) すべてのパケットを同一のパケットバッファで管理

[†]岡山大学大学院自然科学研究科, Graduate School of Natural Science and Technology, Okayama University

表 1 各構成の利点と欠点

	利点	欠点
(構成 1)	メモリ使用量に伴う Linux の数の上限が高い	排他制御が必要
(構成 2)	排他制御が不要	メモリ使用量に伴う Linux の数の上限が低い

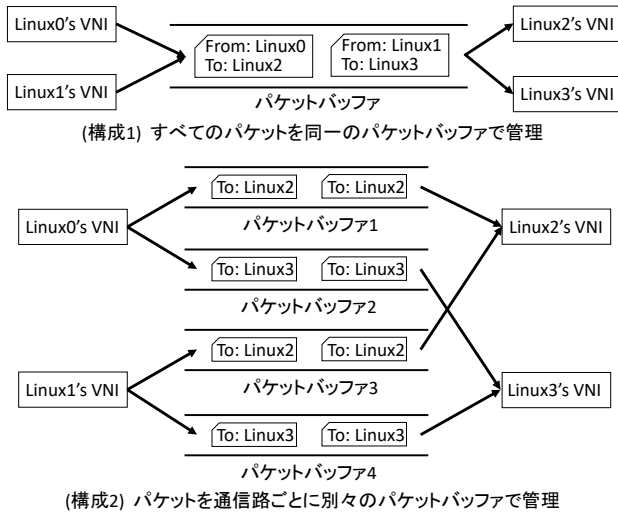


図 3 パケットバッファの構成

この構成では、複数の Linux がパケットバッファを同時に使用することがあるため、排他制御が必要である。

(構成 2) パケットを通信路ごとに別々のパケットバッファで管理

この構成では、1つのパケットバッファに対して、パケットの書き込みと読み込みを行う Linux は1つに限定されるため、排他制御を必要としない。

(構成 1) と (構成 2) の利点と欠点について表 1 に示し、以下で説明する。(構成 1) では排他制御が必要であるが、(構成 2) では排他制御が不要である。このため、(構成 2) は実装の工数が少なく、排他制御によるオーバーヘッドがないため通信速度が高速である。しかし、(構成 2) では、走行する Linux の数の増加に伴い、共有メモリの使用量が $O(n^2)$ で増加する。これは、Linux を頂点、通信路を辺とした完全グラフを構成するからである。

これらをふまえ、以降では、(構成 2) を採用した実装について述べる。8 コア程度の CPU を考えた場合は、共有メモリの使用量が $O(n^2)$ であったとしても実用の範囲内であるといえ、(課題 2) の排他を考慮しないで済む利点が大いといえるからである。

3.4 評価

3.4.1 Linux 既存の App を用いた通信

提案方式では、SSH 通信により、Mint 上で走行している Linux から、Mint 上で走行している他の Linux にログイン可能である。このため、提案方式では、Linux

表 2 通信スループット

通信プロトコル	通信スループット
UDP	78.79 Gbps
TCP	10.51 Gbps

既存の App の改変なしに TCP 通信可能と判断した。

3.4.2 通信スループット

提案方式における通信スループットを計測する。具体的には、提案方式における Linux 間通信での UDP を用いた場合と TCP を用いた場合の通信スループットを計測する。計測は、1,500byte の Ethernet パケットを連続で送信し、受信が完了するまでに要した時間を測定することで行う。計測環境として、各 Linux が持つコア数は 1、メモリ入出力の帯域幅は 204.8Gbps、パケットバッファのサイズは 392,200B である。通信スループットを表 2 に示し、以下で説明する。通信スループットは UDP の場合約 79Gbps、TCP の場合約 11Gbps であった。提案方式では、共有メモリを介して通信するため、通信スループットはメモリ入出力の帯域幅が物理的な上限になる。計測環境から、ソフトウェアオーバーヘッドが通信スループットのボトルネックとなっている。

4. おわりに

1 台の計算機上で複数の Linux を独立に走行させる方式である Mint オペレーティングシステムにおいて、Ethernet 互換で Linux 間通信を行う VNI を実装した。この VNI を用いた通信方式で、Linux 既存の App を改変なしに Linux 間で通信可能であることを示した。また、この通信方式における基本性能を示した。残された課題として、複数台の Mint 間で通信した際の性能評価および走行する Linux の数が増えた場合のパケットバッファのメモリ利用率の評価がある。

参考文献

- [1] Nomura, Y., Senzaki, R., Nakahara, D., Ushio, H., Kataoka, T. and Taniguchi, H.: Mint: Booting Multiple Linux Kernels on a Multicore Processor, *2011 International Conference on Broadband and Wireless Computing, Communication and Applications*, pp. 555–560 (online), DOI: 10.1109/BWCCA.2011.90 (2011).