

Mintオペレーティングシステムを用いた NICドライバの開発支援手法の実現

藤田 将輝 乃村能成 谷口秀夫

岡山大学 大学院自然科学研究科

平成28年2月29日

第136回OS研究会

目次

1. はじめに
2. 関連研究
3. Mintを用いた開発支援環境
4. NICドライバの開発支援環境の設計
5. 評価
6. おわりに

目次

1. はじめに

2. 関連研究

3. Mintを用いた開発支援環境

4. NICドライバの開発支援環境の設計

5. 評価

6. おわりに

はじめに

OS機能の多機能化によりOSの開発工数が増加
割込処理は再現が困難

∴ 割込処理は非同期に発生

<既存研究>

仮想計算機を用いた開発支援環境

例：割込挿入手法, ロギング/リプレイ手法

(問題点) 短い間隔や一定間隔での割込発生が困難

∴ ハイパーバイザの処理オーバヘッド

Mint: 1台の計算機上で仮想化を用いずに, 複数のLinuxを動作

∴ 仮想計算機方式に対し, ハイパーバイザの影響無



Mintを用いた開発支援環境を提案

目次

1. はじめに

2. 関連研究

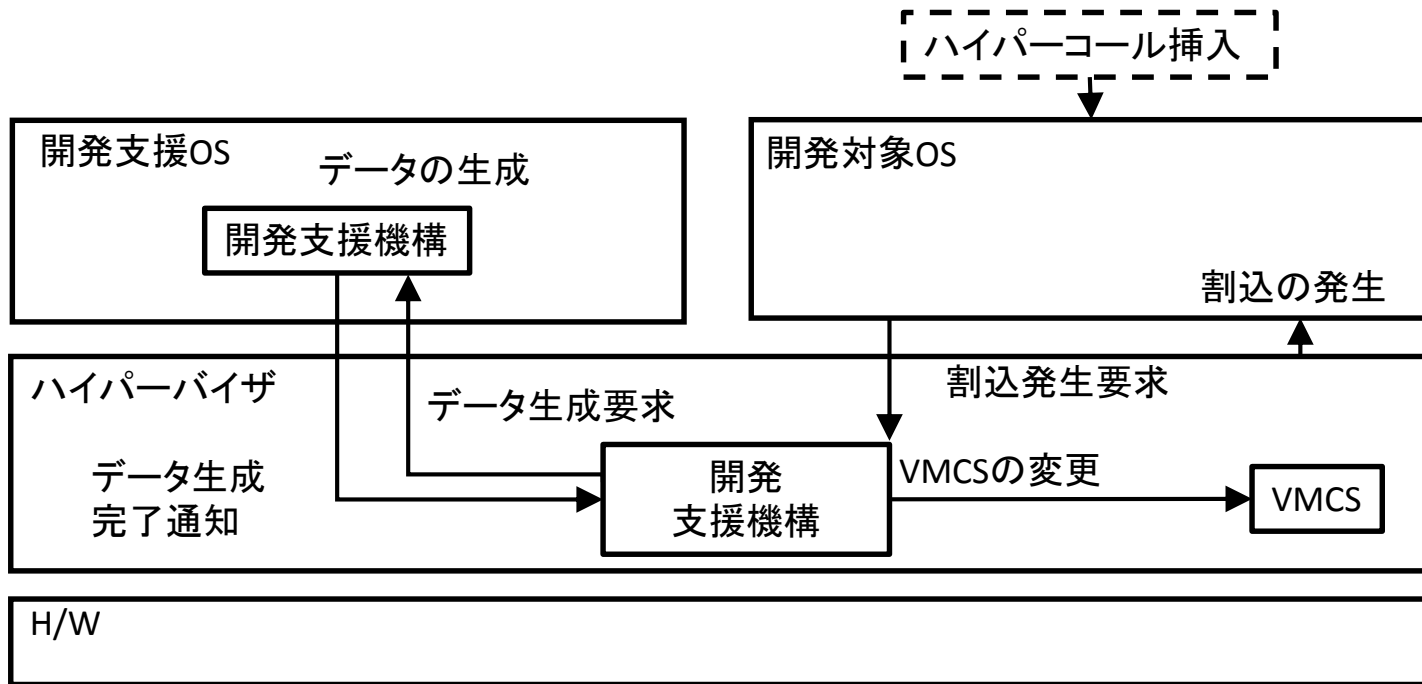
3. Mintを用いた開発支援環境

4. NICドライバの開発支援環境の設計

5. 評価

6. おわりに

割込挿入手法(既手法1)



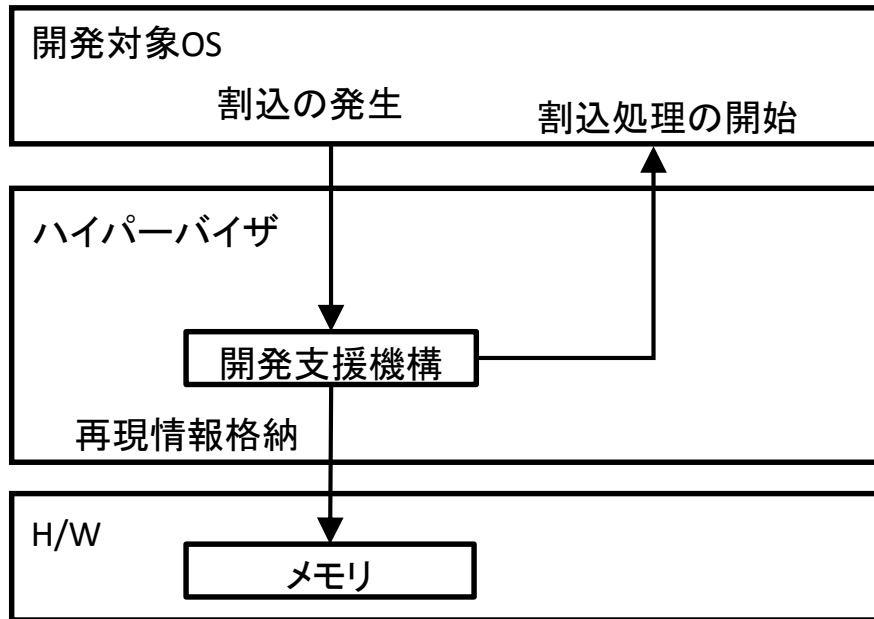
ユーザが割込を発生させたいコード位置にハイパーコール挿入

➡ 割込を発生させ、開発を支援

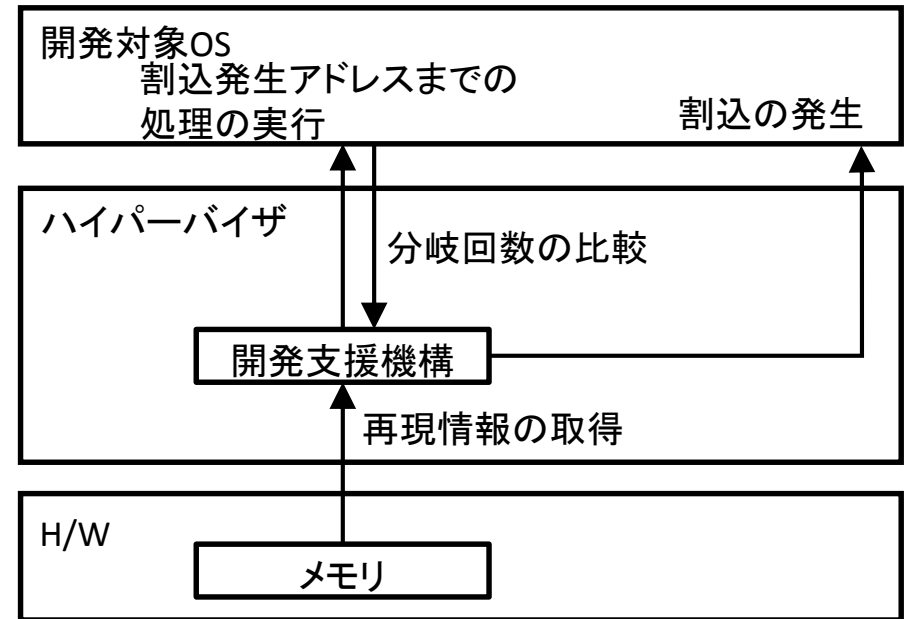
(問題点) 一定間隔で割込挿入が困難

∴ 間隔の調整はユーザによるハイパーコールの間隔調整

ロギング/リプレイ手法(既手法2)



(1) ロギング



(2) リプレイ

開発対象OSが動作をロギングし, リプレイで動作を再現

➡ 割込処理を再現し, 開発を支援

(問題点) 短い割込間隔の再現が困難

∴ ハイパーバイザへの処理遷移にかかるオーバヘッド

目次

1. はじめに
2. 関連研究
3. Mintを用いた開発支援環境
4. NICドライバの開発支援環境の設計
5. 評価
6. おわりに

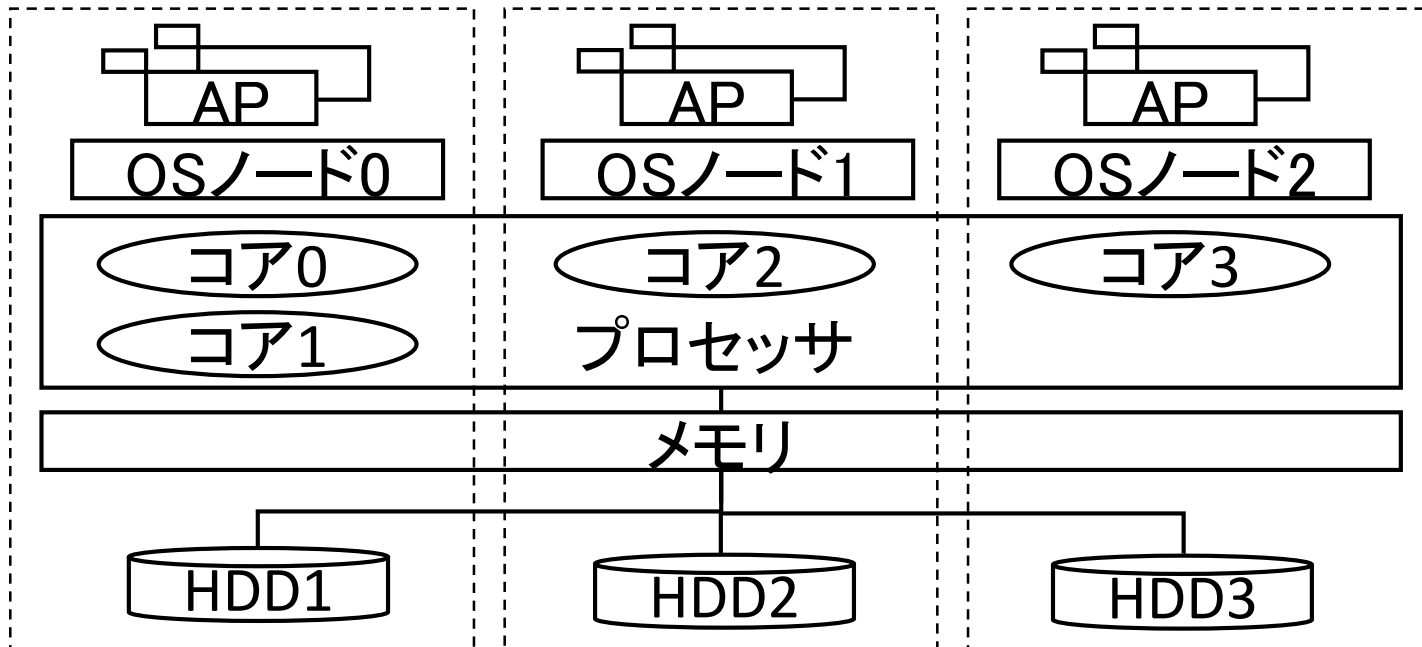
Mintとは

各OSノードが計算機資源を分割して占有
各OSノードが独立しており, 互いに影響無

プロセッサ: コア単位で分割し, 1つ以上のコアを占有

メモリ: 空間分割し, 分割領域で分配

デバイス: デバイス単位で分割し, 直接占有制御



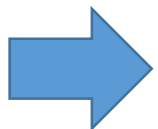
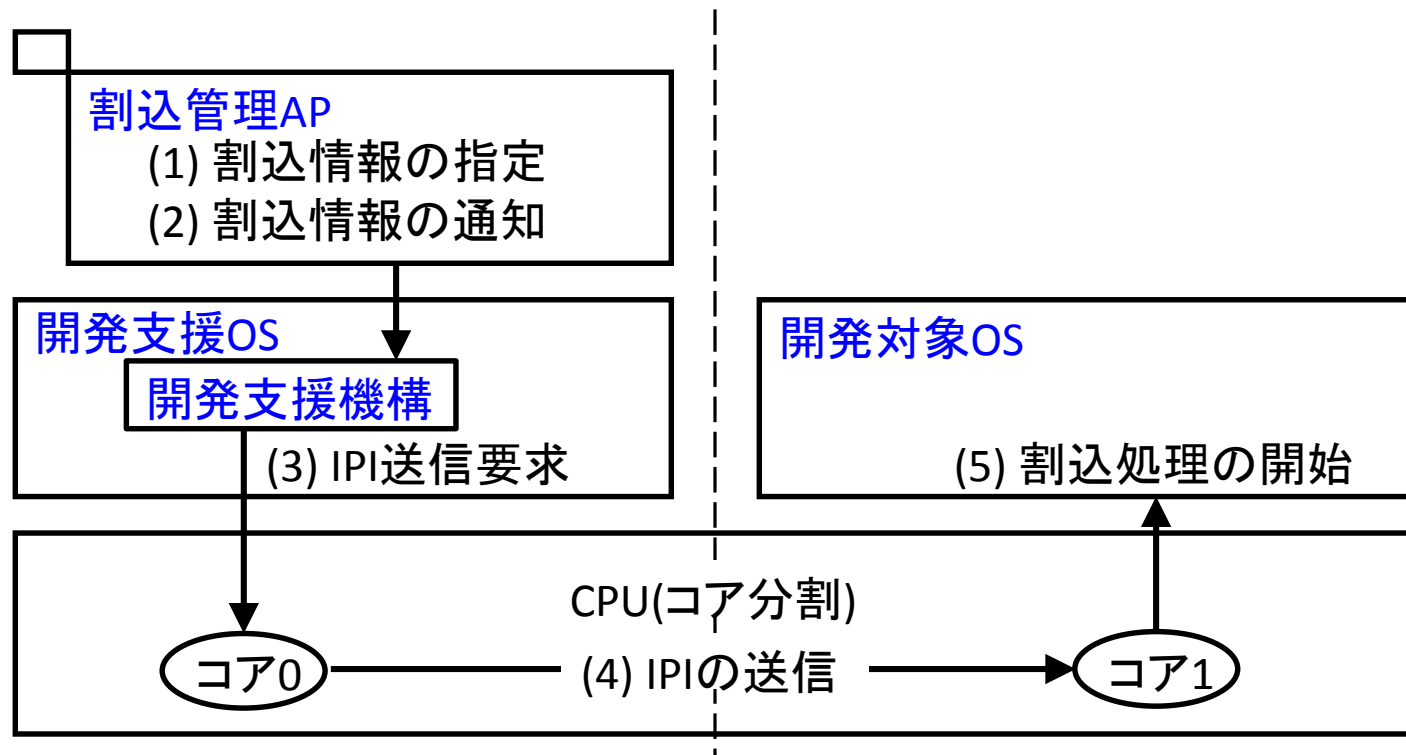
開発支援環境の構成と処理流れ

開発支援OS: 割込管理APと開発支援機構を保持

開発対象OS: 開発対象のOSであり, 割込が発生

割込管理AP: 割込に必要な情報を指定し, 開発支援機構に通知

開発支援機構: 通知された情報から割込を生成



短い割込間隔と一定間隔の割込を実現

目次

1. はじめに
2. 関連研究
3. Mintを用いた開発支援環境
4. NICドライバの開発支援環境の設計
5. 評価
6. おわりに

NICドライバの開発支援環境の設計

提案手法を用いてNICドライバのパケット受信割込処理を再現

<目的>

高速化するNICにドライバを適応させるためのテスト

例: ストレステストやパフォーマンスチューニング

➡ NICドライバを対象とした割込開発支援環境を構築

<設計方針>

(1) 指定した間隔と回数の割込発生

発生させる割込の間隔と回数をユーザが指定可能な環境

(2) NICの動作を開発支援OSが再現

ハードウェア(NIC)を用いずにNICドライバを動作

(3) 共有メモリを用いたパケットの受け渡し

開発支援OSから開発対象OSへパケットを送信

課題と対処

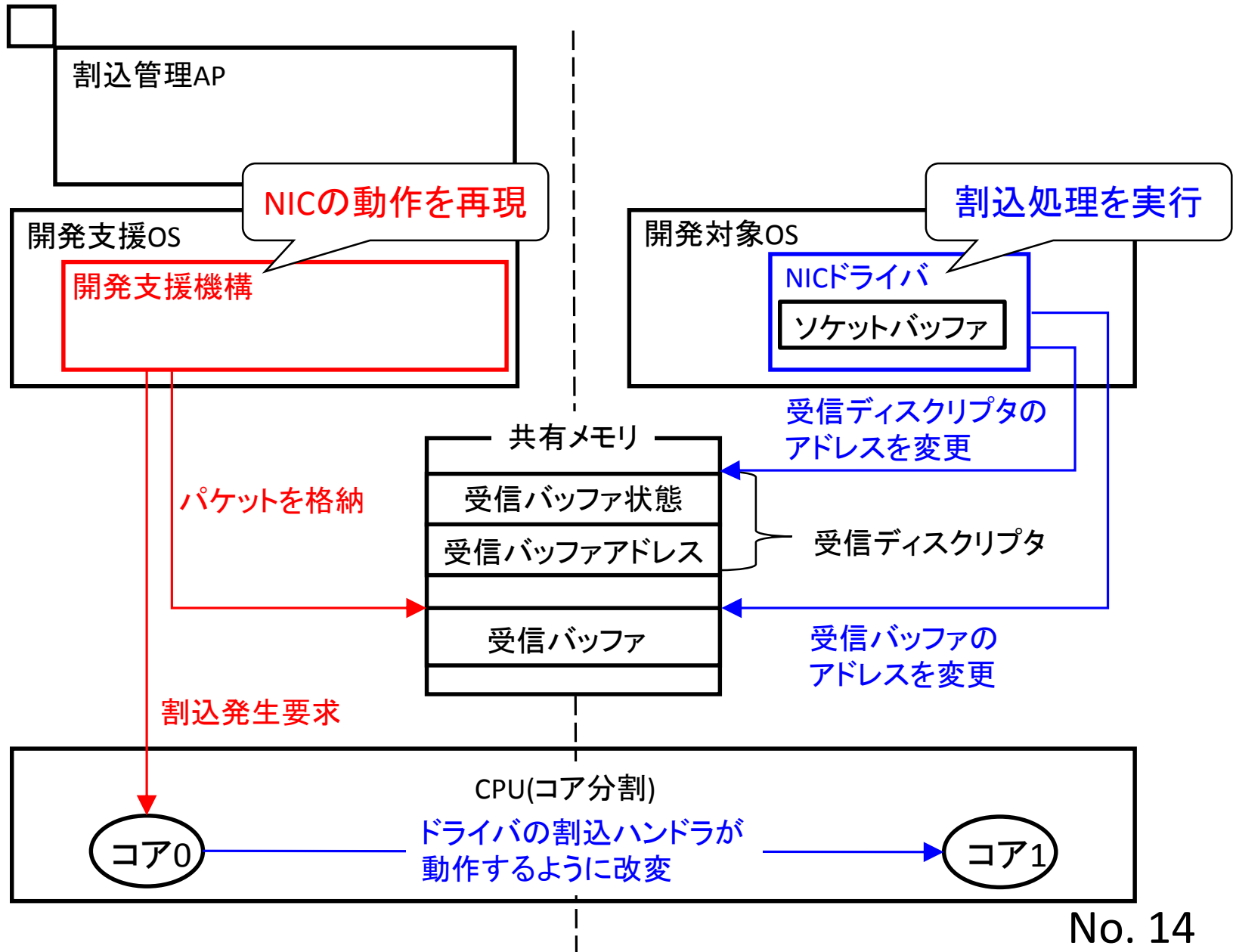
<課題>

- (課題1) 割込間隔と回数の調整
- (課題2) パケットの作成, 格納
- (課題3) NICが保持する状態の更新
- (課題4) 割込処理の発生
- (課題5) 受信バッファの作成

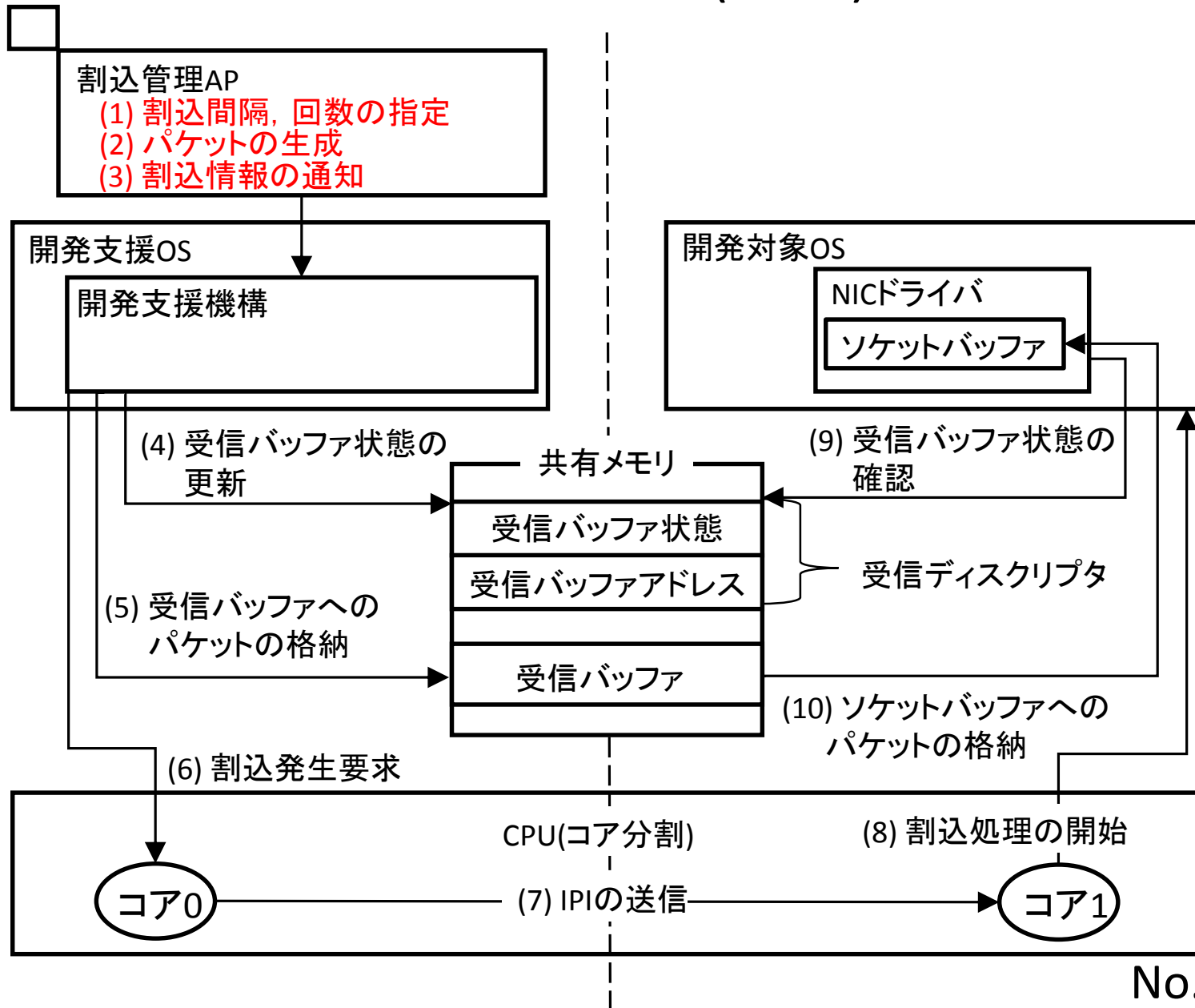
<対処>

- (対処1) 割込管理APを実装し, ユーザが割込の間隔と回数を指定
- (対処2) 割込管理APがパケットを作成し, 開発支援機構が格納
- (対処3) 受信ディスクリプタを共有メモリに配置し, 両OSから参照
受信ディスクリプタ: 受信バッファの受信状態
- (対処4) IPIで割込を発生させ, 割込ハンドラが動作するよう改変
- (対処5) 受信バッファのアドレスを共有メモリ内に改変

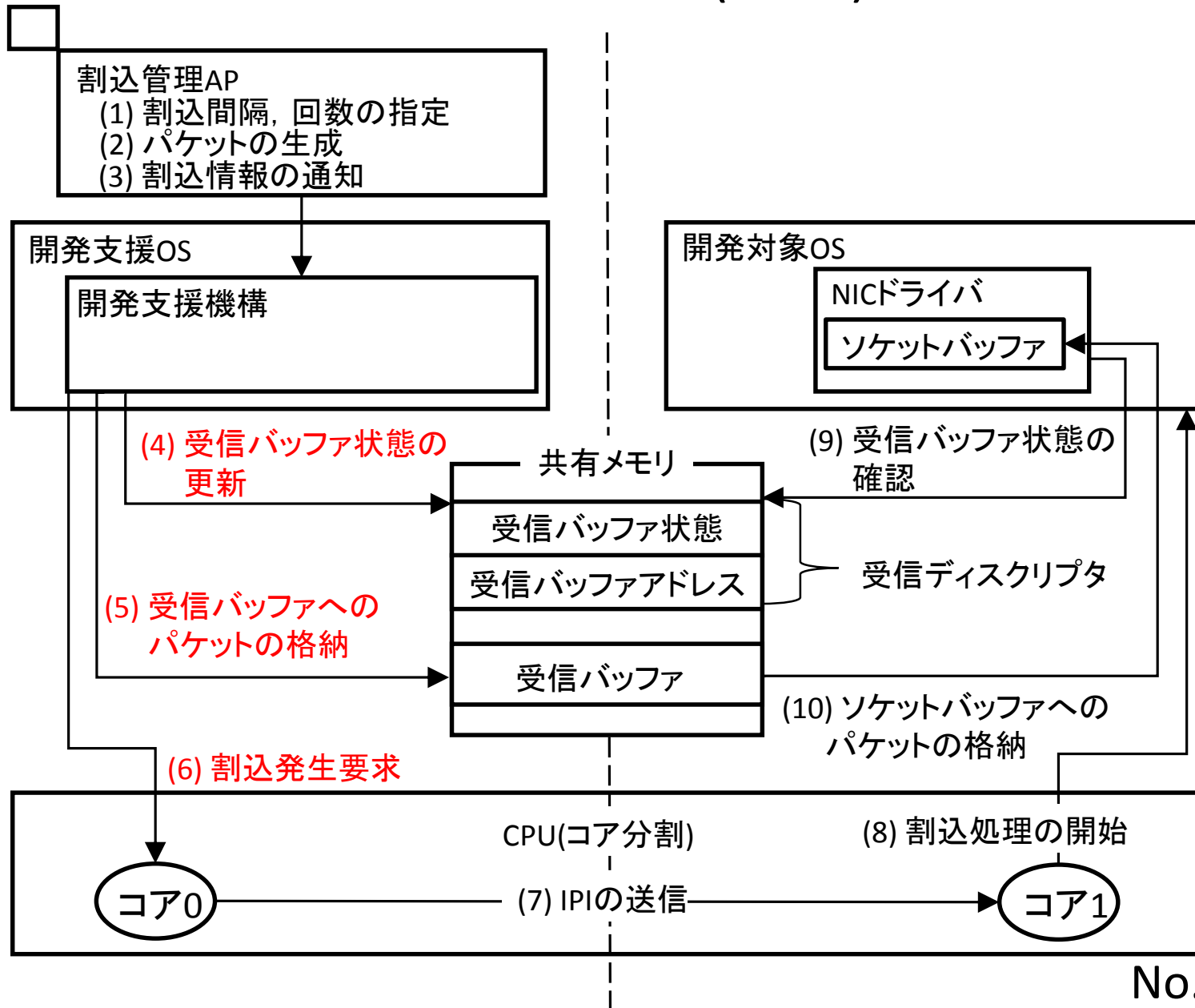
処理の構成



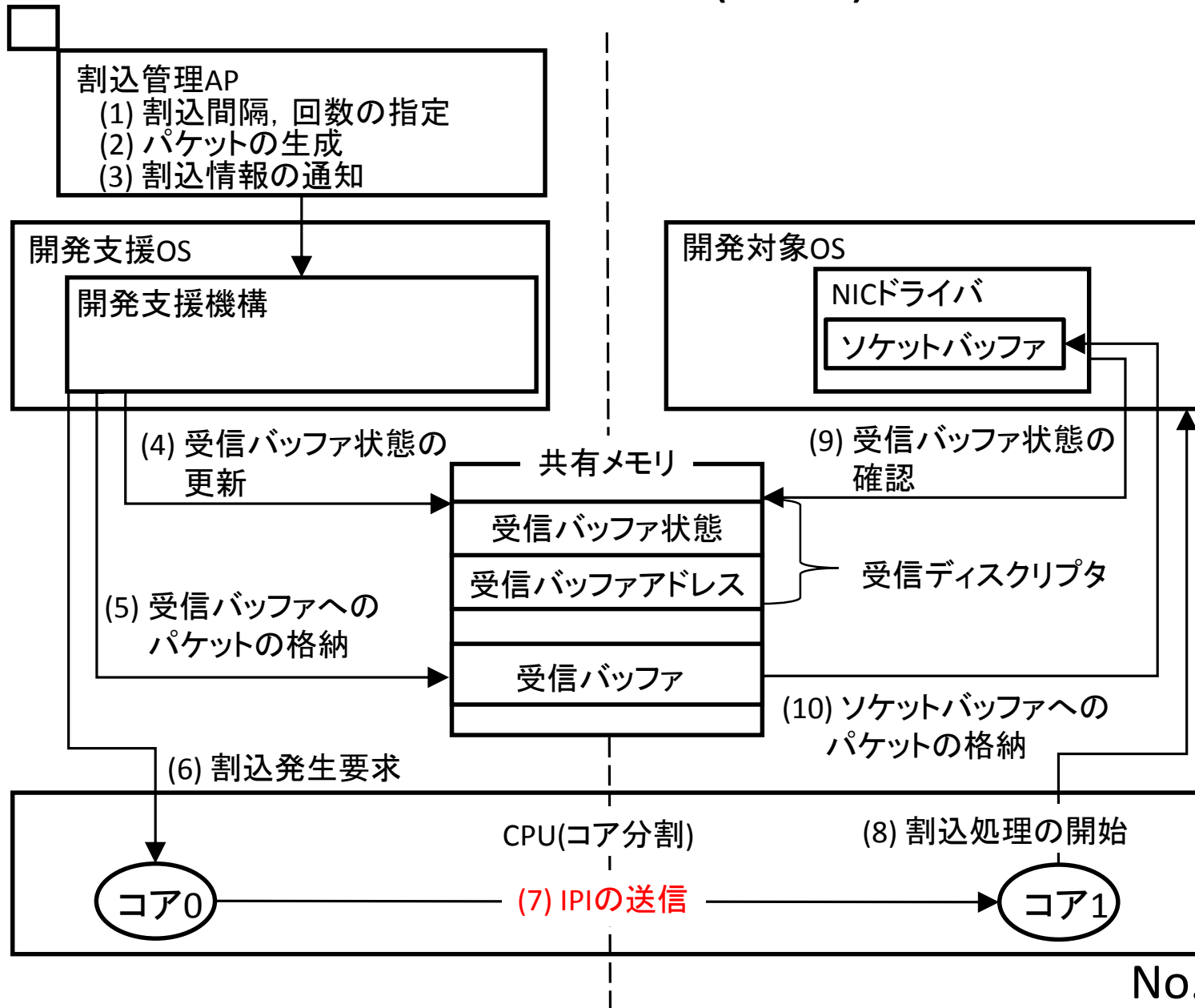
処理流れ(1/4)



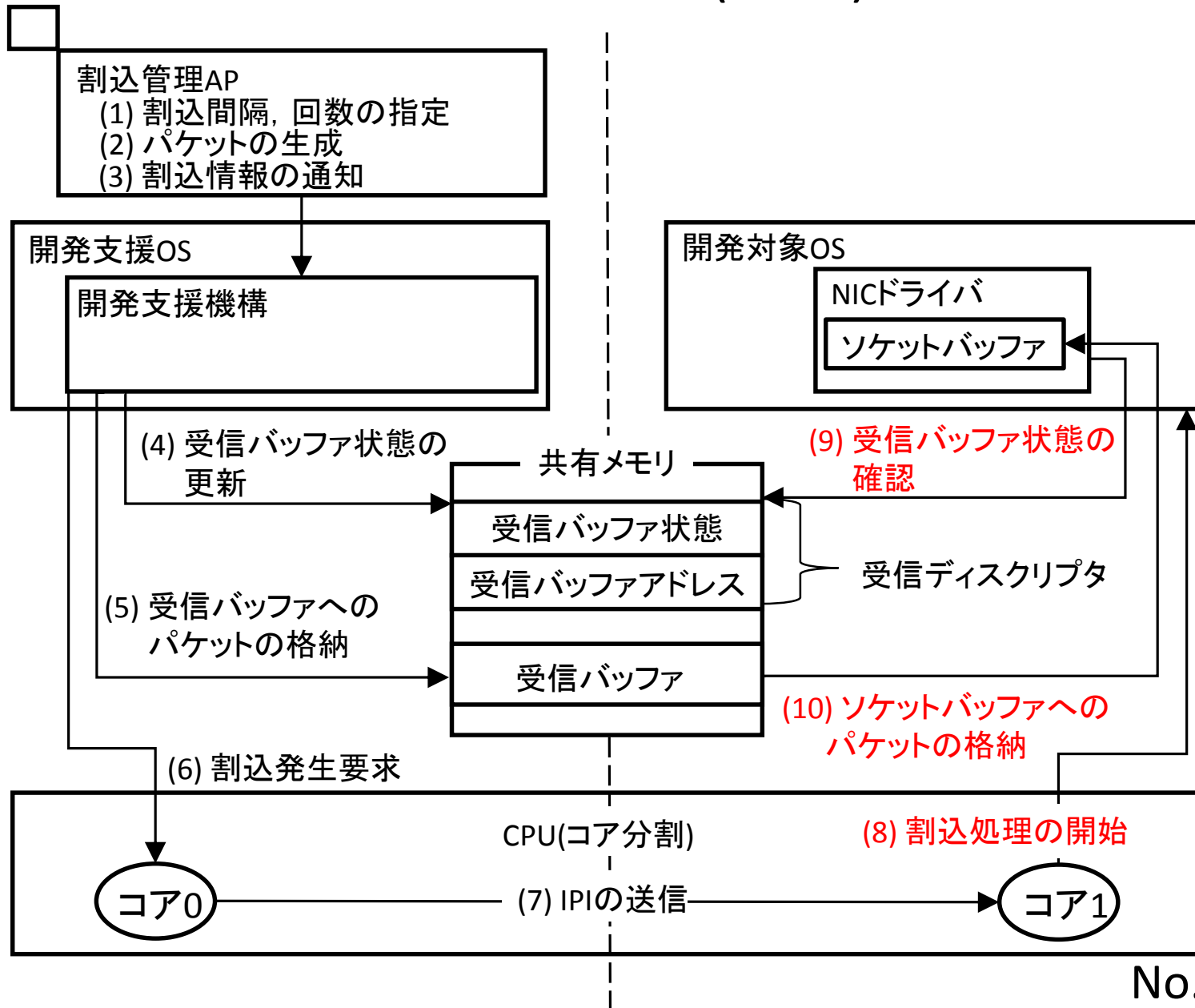
処理流れ(2/4)



処理流れ(3/4)



処理流れ(4/4)



目次

1. はじめに
2. 関連研究
3. Mintを用いた開発支援環境
4. NICドライバの開発支援環境の設計
5. 評価
6. おわりに

評価

<開発支援機構の評価>

(評価 1) 割込の間隔

開発支援機構が実現可能な割込間隔を評価

(評価 2) 割込間隔の精度

開発対象OSにおいて、発生可能な割込み間隔の精度を評価

<本環境を通しての評価>

(評価 3) NICドライバへの適用

本環境を用いてNICドライバの通信処理の性能を評価

<評価環境>

| | |
|---------|--|
| プロセッサ | Intel® Core™ i7-860, 2.93 GHz (開発支援OS: 1コア, 開発対象OS : 1コア) |
| OS | Linux 3.0.8 (64 bit) |
| NICドライバ | RTL8169 |

割込の間隔

IPIの送信はパケット送信処理の最後に送信

∴ 送信処理時間は実現可能な割込間隔の最小値

パケット送信処理: パケット受信時に行うNIC(ハードウェア)の処理

＜パケット送信処理＞

(1) 受信ディスクリプタの更新

(2) パケットの配置

(3) IPIの送信要求

| パケットサイズ (KB) | 処理時間 (μs) |
|--------------|-----------|
| 1.5 | 0.205 |
| 8 | 1.462 |
| 16 | 3.664 |



間隔を調整することにより、実際に発生する割込間隔を再現可

割込間隔の精度(1/2)

開発対象OSでどの程度の精度で割込間隔を実現できるかを評価

<実NICを用いた際の間隔>

- (1) 約38 μ sで発生
- (2) 常に $\pm 2\mu$ sほどのずれ
- (3) まれに大きな外れ値(約4s)

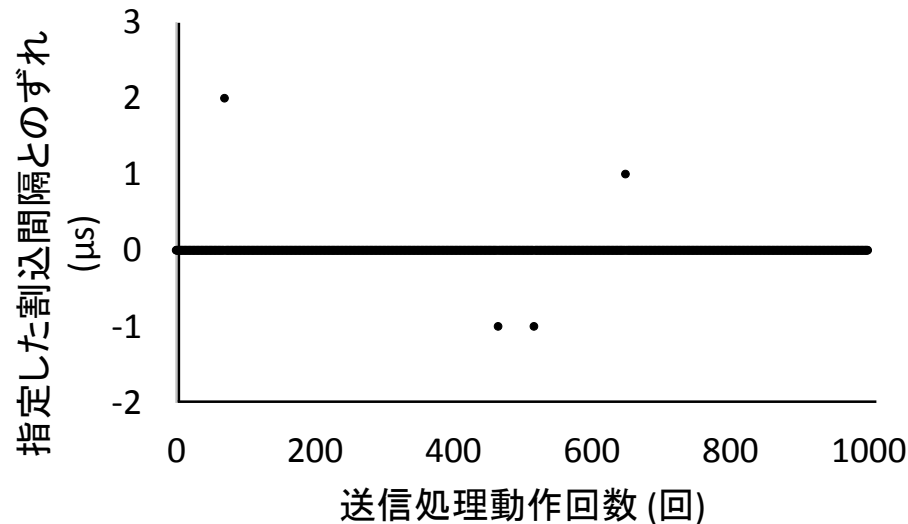
➡ 実際のNICでは安定しない

提案手法では高い精度の割込間隔を実現

<測定方法>

- (1) 開発支援OSで38 μ sを指定して1000回連続で割込を発生
- (2) 開発対象OSにおける割込ハンドラの動作間隔を測定
- (3) 両者間のずれを計測

割込間隔の精度(2/2)



38μs(指定した割込間隔)と実際発生した割込間隔との差

(1) ほとんどの施行(99.6%)で指定した38μsを実現

(2) 4回外れ値があるが, 6.25%以内

(3) NICでは常に±2μs程ぶれていて, 稀に約4sの外れ値



安定した割込間隔を実現可能

NICドライバへの適用

以下の項目で実際にNICドライバの処理性能を評価

- (1) 処理可能な送信間隔
- (2) 通信速度

➡ 開発対象ドライバがどの程度高速なNICに対応可能か明確

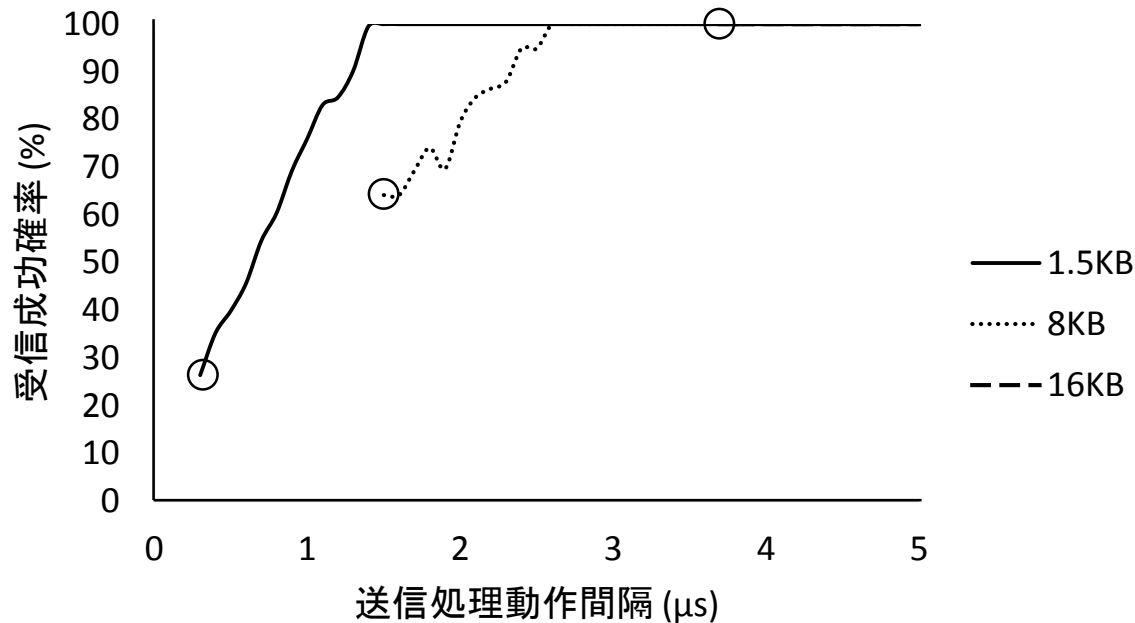
<測定方法>

- (1) 連続で5000回パケット送信処理を動作(1サイクル)
- (2) 1サイクル毎にNICドライバで受信に成功したパケット数と
かかった時間を測定
- (3) 次サイクルに移る毎に割込間隔を長大させて複数サイクル試行

処理可能な送信間隔

1サイクル毎の受信成功数から受信成功率を算出

受信成功: パケットをソケットバッファに格納



※ 丸印は、各サイズにおける送信処理時間(割込間隔の最小値)

(1) 送信間隔の長大にともなって、受信成功率が1次関数的に増加

(2) サイズの増大に伴い、受信成功率が100%になる間隔が長大

通信速度

初めて受信成功確率が100%になった際の通信速度を算出

| パケットサイズ (KB) | 通信速度 (Gbps) |
|--------------|-------------|
| 1.5(実NIC) | 0.92 |
| 1.5 | 6.3 |
| 8 | 22.7 |
| 16 | 30.6 |

- (1) パケットサイズが同じ1.5KBであっても、高速な通信速度
- (2) 最高で30.6Gbpsの通信速度を実現
- (3) 測定環境のメモリ帯域幅は約130Gbps

パケット受信割込処理にはソケットバッファを上位に送る処理



新規NICドライバ開発において、NICを用いず通信性能を測定可能

目次

1. はじめに
2. 関連研究
3. Mintを用いた開発支援環境
4. NICドライバの開発支援環境の設計
5. 評価
6. おわりに

おわりに

<VMを用いた開発支援環境>

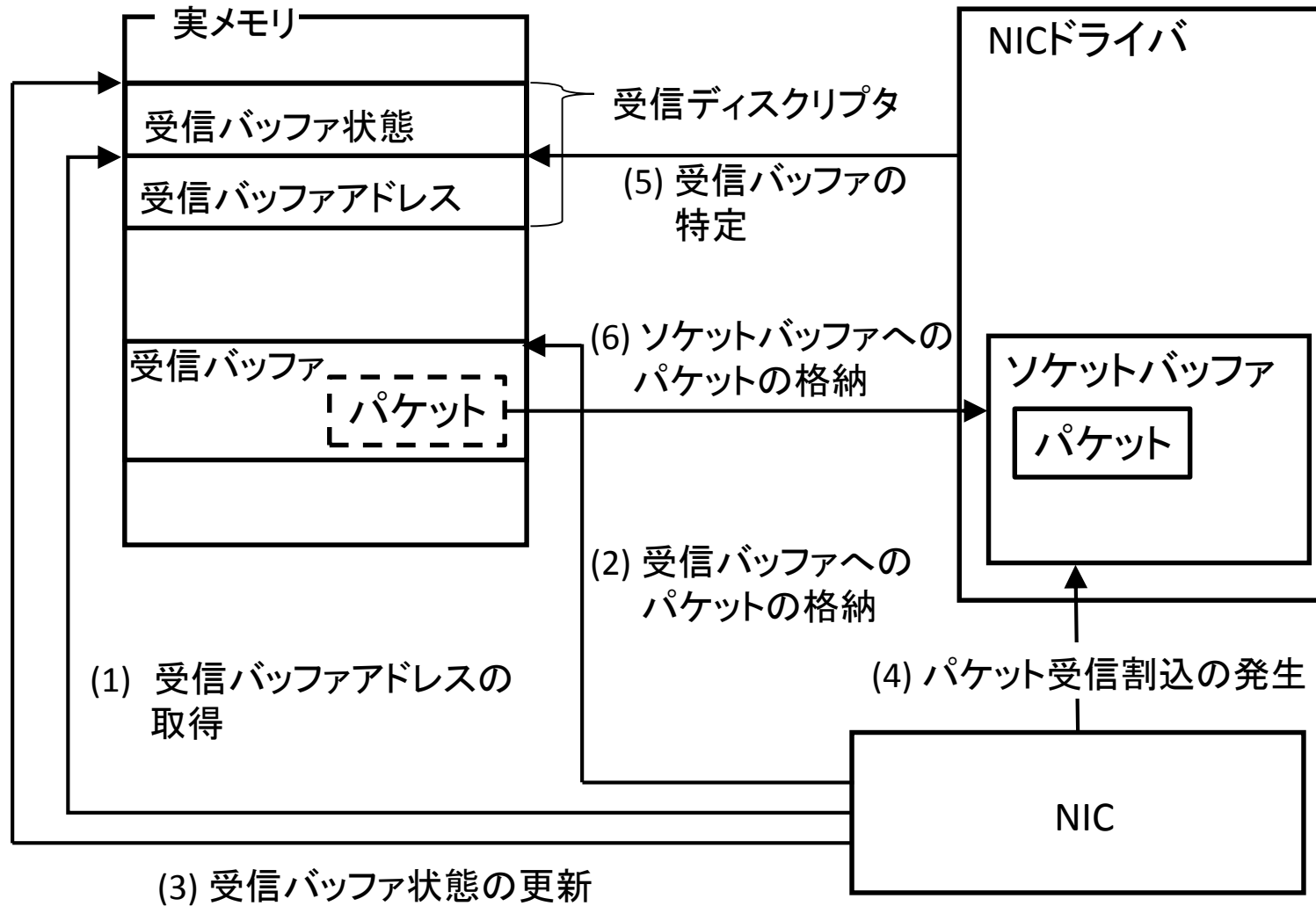
- (1) 仮想計算機を用いて割込を再現し、開発を支援
- (2) 短い間隔や一定間隔での割込再現が困難
 - ∴ ハイパーバイザの処理オーバヘッド

<Mintを用いた開発支援環境>

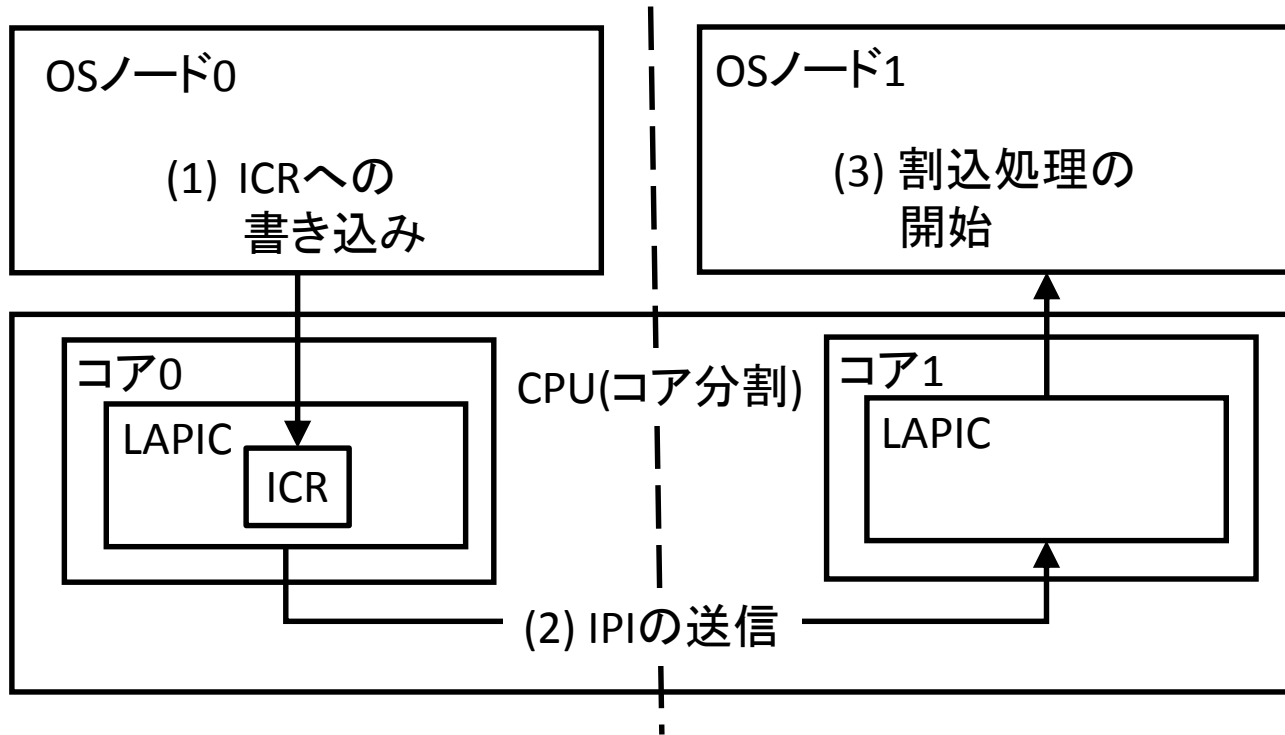
- (1) Mintは仮想化を用いず複数のOSを動作可
 - ∴ 短い間隔や一定間隔の割込を再現可
- (2) ハードウェア(NIC)の割込間隔と比較し、高速な割込を発生
- (3) 指定した割込間隔は6.25%以内で実現
- (4) 本環境は新規NICドライバ開発時に、NICを用いずに高速なNICを擬似可

予備スライド

NICの動作



IPIの送信



LAPICが保持するICRを書き換えることにより, IPIを送信

受信処理時間内訳

<受信処理>

(1) メモリ複写処理

(2) プロトコル処理

| パケットサイズ(KB) | メモリ複写処理(μ s) | プロトコル処理(μ s) |
|-------------|-------------------|-------------------|
| 1.5 | 0.272 | 1.23 |
| 8 | 1.32 | 1.63 |
| 16 | 2.18 | 1.75 |