

## デュアル OS 「NINJA」 の評価

田淵正樹<sup>†</sup> 榎本圭<sup>†</sup> 伊藤健一<sup>†</sup> 乃村能成<sup>‡</sup> 谷口秀夫<sup>‡</sup>

<sup>†</sup>株式会社 NTT データ技術開発本部

<sup>‡</sup>岡山大学工学部

### 1. はじめに

近年，計算機ハードウェアの性能向上により，複数の計算機環境で実現されてきた多様なサービスを，1 台の計算機上に集約したいという要求がある．この要求を満足させる方式の一つとして，1 台の計算機上に複数の OS が走行する環境を構築する仮想計算機方式<sup>[1],[2]</sup>がある．しかし，仮想計算機方式では，個別の OS 環境を実現するためにホスト OS の上でゲスト OS が動作する形態をとり，一方の OS の処理負荷が他方の OS の処理性能に影響を与えてしまう．また，ゲスト OS ではハードウェアの提供する入出力性能を十分に利用できないという問題がある．

そこで我々は，従来の仮想計算機方式の問題を解決するため，1 台の計算機上で 2 つの OS を走行させるデュアル OS 方式を提案し，2 つの Linux を対象にデュアル OS 方式を適用した「NINJA」を開発している<sup>[3]</sup>．オープンソースである Linux は改造が容易で，様々な機能に特化した OS としての開発も進んでいる．従って，一口に Linux といっても様々な OS 環境が想定され，デュアル OS 環境における異種 OS といえる．本稿では，NINJA の性能について報告する．

### 2. デュアル OS 「NINJA」

デュアル OS 方式は，仮想計算機方式が従来持つ問題を解決するために，以下の 2 点を設計の目標とする．

- (1) 各 OS は他 OS の処理負荷の影響を受けない
- (2) 両 OS とも入出力性能を十分に利用できる

そのために，本方式では，1 台の計算機の各ハードウェア資源を分割し，できるかぎり各 OS に占有させる．本方式の構成を図 1 に示す．メモリや入出力機器はいずれかの OS に占有させる．つまり，入出力機器 1~3 は OS1 により占有制御され，入出力機器 4, 5 は OS2 により制御される．ただし，ハードウェア機構上，分割が難しいプロセッサは各 OS で共有する．

以降，各ハードウェア資源の分割占有方法と，

OS の切替え方法について簡単に説明する．

各ハードウェア資源の分割占有は，各 OS の起動時と入出力機器からの割込みに対して対処を行うことで実現する．プロセッサについては，起動時には起動処理を順次行うこととし，起動後はタイマ割込みを利用して OS を切替え，時分割する．実メモリについては，起動時に共有部分を持たないよう上位と下位に 2 分割し各 OS に占有させる．入出力機器については，起動時に各 OS 毎に指定された入出力機器のみを占有制御するように環境設定を行う．このため，当該割込みを発生した入出力機器を占有制御している OS の割込み処理が呼び出されるように制御する．従って，割込み発生時に走行している OS (AP が走行している場合は，その AP が利用している OS) が当該割込みを発生した入出力機器を占有制御している OS と異なっている際には，OS 切替えを行う．

OS 切替えは，割込みを契機に走行している OS の環境を保存し，切替え後に走行する OS の環境を復元する(以降，OS 環境保存/復元処理)．また，切替え後に該当割込み処理を実行するために，図 2 に示すように各 OS の仮想空間において，

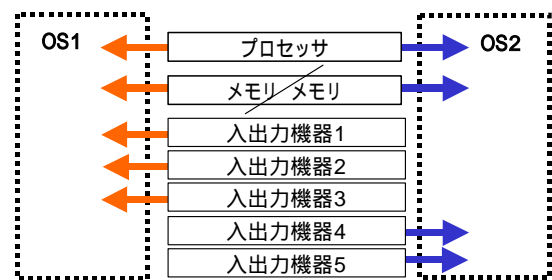


図1 本方式の構成

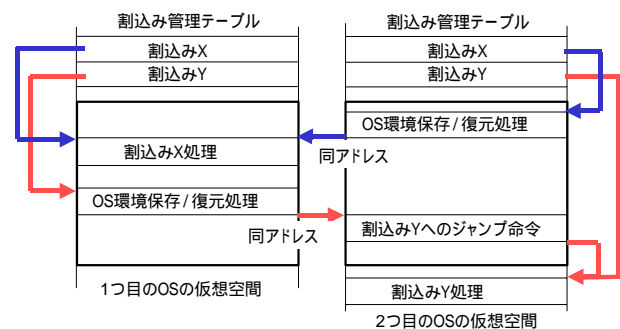


図2 OS切替え時の仮想空間の様子

Evaluation of a DualOS 「NINJA」

<sup>†</sup>Masaki TABUCHI, Kei MASUMOTO, Ken-ichi ITOH

<sup>†</sup>Reserch and Development Headquarters, NTT DATA Co.

<sup>‡</sup>Yoshinari NOMURA, Hideo TANIGUCHI

<sup>‡</sup>Faculty of Engineering, Okayama University

OS 環境保存/復元処理の終了位置と、切替え先の OS の該当割り込み処理の開始位置を一致させる。これにより、他方の OS が占有するデバイスからの割り込みが発生すると OS 環境保存/復元処理を実行し、他方の OS へ制御が移行する。そして該当する割り込み処理が実行される。

### 3. 評価

測定環境を次に示す。CPU: Intel Pentium4 1.7GHz, メモリ: 256MB, OS: Linux Kernel-2.4.7

計測は CPU のタイムスタンプカウント値を出力する rdtsc 命令を用いて行った。

#### 3.1 各 OS の処理負荷が与える影響

他方の OS の処理負荷が与える影響を見るために、一方の OS で、プロセッサ上で実行される処理が終了するまでの時間を計測した。具体的にはループ処理(繰返し回数: 1250 万回 単独 OS の場合の処理時間: 14.74ms)を、他方の OS で負荷をかけない場合と、無限ループ処理負荷をかけた場合で実行した。計測結果を測定回数 50 回のヒストグラムで図 3 に示す。

処理時間は、負荷の有無に関わらず 24~25ms と 34~35ms の間に集中し、単独 OS の場合と比べ、10ms, または 20ms 増加する。これは、単独 OS の場合に 14.74ms かかる処理において、タイマ割り込み(周期: 10ms)は 1 回か 2 回発生する。この際のタイマ割り込みによる OS 切替えによって、他方の OS にプロセッサを占有される時間である。

従って、OS 切替えにより他方の OS がプロセッサを占有する以外の処理時間は、負荷の有無に関わらず同じであり、他方の OS による影響を受けていない。

#### 3.2 入出力性能

入出力性能を見るため、ローデバイスにおいてシステムコール LSEEK, READ を用いたディスク I/O 処理(繰返し回数: 1 万回)が終了するまでの時間を、単独 OS と NINJA で計測した。ここで、LSEEK にシークさせる位置はランダムとし、READ1 回の処理で読み込むデータ数を 512 バイト、1024 バイトの 2 通りで行った。また、NINJA において他方の OS で負荷をかけた場合の処理時間も計測した。この時、各 OS が I/O 処理を行うディスクは別物理ドライブとする。計測結果を図 4 に示す。

単独 OS と NINJA での処理時間には、ほとんど差がない。また、NINJA の他方の OS でループ処理負荷やディスク I/O 処理負荷をかけた場合でも、負荷をかけない場合の処理時間と比べ少し

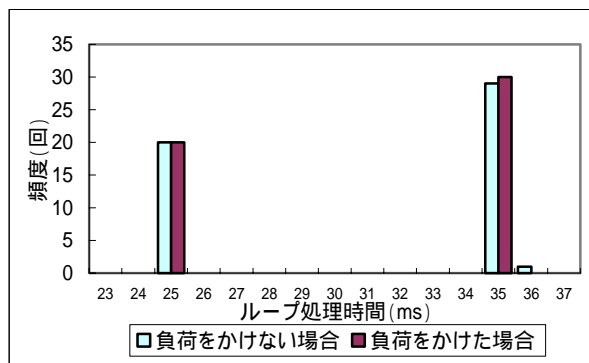


図 3 NINJA でのループ処理時間

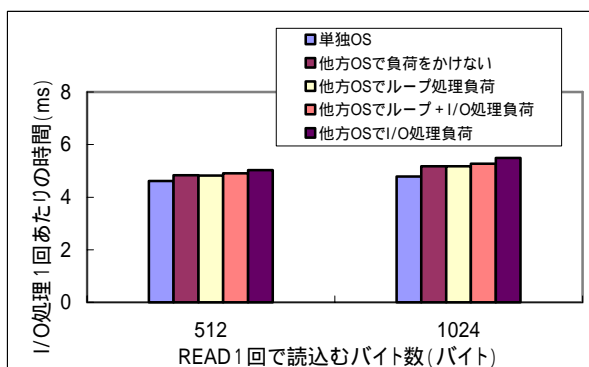


図 4 NINJA でのディスク I/O 処理時間

の遅延で処理が終了する。これは、ディスク I/O 処理の I/O 待ち時間中に他方の OS へ切替わった場合、他方の OS が走行中に自 OS が占有するディスクからの割り込みが発生する。この際に、迅速に OS 切替えを行い自 OS へ制御を戻し、該当割り込み処理を行うためである。

従って、NINJA は入出力性能に関して、単独 OS の場合とほとんど変わらない性能を示す。また、他方の OS の負荷状態に影響を受けない。

### 4. おわりに

デュアル OS 方式を 2 つの Linux に適用した「NINJA」の性能評価の結果を述べた。各ハードウェア資源の分割占有により、入出力性能がほとんど低下しないことを示した。今後は、プロセッサの有効利用方式の検討を行う予定である。

#### 参考文献

- [1]G.J.Popek,R.P.Goldberg:"Formal Requirement for Virtualizable Third Generation Architectures,"Commun.ACM,17(7),pp.412-421,1974
- [2]C.A.Waldspurger:"Memory Resource Management in VMware ESX Server",OSDI2002
- [3]田淵正樹,榎本圭,伊藤健一,乃村能成,谷口秀夫:"2 つの Linux を同時走行させる機能の設計と評価",コンピュータシステム(CS2003)シンポジウム論文集,pp.71-78