

マルチコア *AnT* における通信処理分散効果 Communication Processing Distribution for Multi-core *AnT*

小林 優也
Yuya Kobayashi

佐藤 将也
Masaya Sato

谷口 秀夫
Hideo Taniguchi

1. はじめに

プロセッサ内のコア数の増加に伴い、処理を各コアに分散することが重要になっている。マイクロカーネル構造 OS は、大半の OS 機能をプロセス (OS サーバ) として実現する。このため、AP プロセスを各コアに分散するかどうかに関係なく、OS サーバを各コアに分散することで OS 処理を分散できる。

本稿では、マイクロカーネル構造 OS であるマルチコア *AnT* オペレーティングシステムとモノリシックカーネル構造 OS である Linux について、通信処理を分散した場合の性能を比較評価する。

2. 処理の分散形態

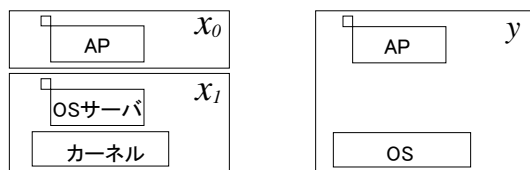
2.1 処理の分散単位

AP と OS の処理の分散単位を図 1 に示す。マイクロカーネル構造 OS (図 1 (A)) は、AP プロセスと OS サーバを各コアに分散することで、AP 処理と OS 処理を別のコアで実行できる。つまり、AP 処理と OS 処理を別の単位 (x_0 と x_1) で分散できる。

モノリシックカーネル構造 OS (図 1 (B)) は、AP プロセスのシステムコール発行によって OS 処理を実行する (つまり、AP プロセスが走行しているコアと同じコアで OS 処理も実行される)。このため、AP プロセスを各コアに分散することで OS 処理を分散できる。いかにいえば、AP プロセスを各コアに分散しない場合、OS 処理を分散できない。つまり、AP 処理と OS 処理を 1 つの単位 (y) として分散する。

2.2 比較

AP 処理と OS 処理の 4 種類の分散形態について、各 OS が実現可能か否かを表 1 に示す。「集中」は、複数の処理を同一コア上で実行することを示し、「分散」は、複数の処理を各々別コア上で実行することを示す。マイクロカーネル構造 OS は、全形態を実現できる。また、「集中、集中」と「分散、分散」の場合において、AP 処理と対応する OS 処理を同一コア上で実行しなくてもよい。



(A) マイクロカーネル構造 OS (B) モノリシックカーネル構造 OS

図 1 処理の分散単位

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 分散形態実現の可否

AP 処理	OS 処理	マイクロ	モノリシック
集中	集中	可	可
分散	集中	可	不可
集中	分散	可	不可
分散	分散	可	可

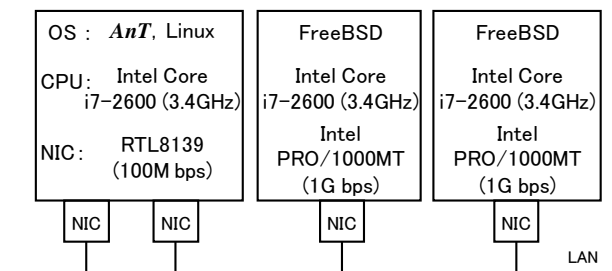


図 2 測定の計算機環境

一方、モノリシックカーネル構造 OS は、AP 処理と OS 処理が同じ形態 (「集中」または「分散」) でなければ実現できない。さらに、「集中、集中」と「分散、分散」の場合、AP 処理と対応する OS 処理を同一コア上で実行する。

3. 読み込み処理分散効果

RAW デバイスからのランダムリード処理における分散効果の評価を文献 [1] で行った。文献 [1] では、評価結果より、以下のことを述べた。

(1) サーバプログラム間通信の評価により、*AnT* では IPI の送受信が必要となる他コアへの依頼を行う場合でも、サーバプログラム間通信のオーバーヘッドは 1 回につき約 $5\mu s$ に抑えられる。

(2) 分散効果の評価により、OS 処理の PU 処理時間が増加した場合、Linux では、処理時間が大きく増加するのに対し、*AnT* では、処理分散効果によりほとんど増加しない。

(3) コアの負荷が大きい状況では、(2) の傾向が強く表れる。

4. 通信処理分散効果の評価

4.1 内容

測定の計算機環境を図 2 に示す。送信側計算機は、Intel Core i7-2600 (3.4GHz) を搭載した計算機であり、NIC として RTL8139 (100Mbps) を 2 個有する、2 台の受信側計算機は、同様に Intel Core i7-2600 (3.4GHz) を搭載した計算機であり、NIC として Intel PRO/1000 MT

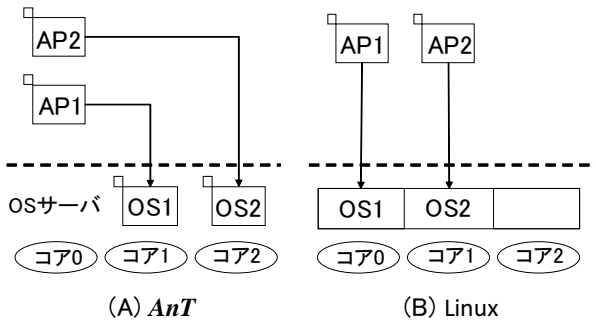


図3 処理の分散形態

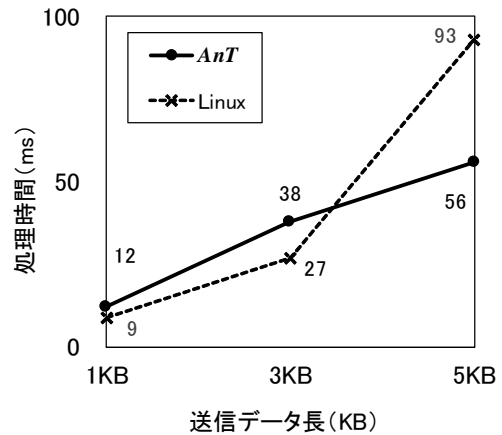


図5 1回の送信データ長と処理時間

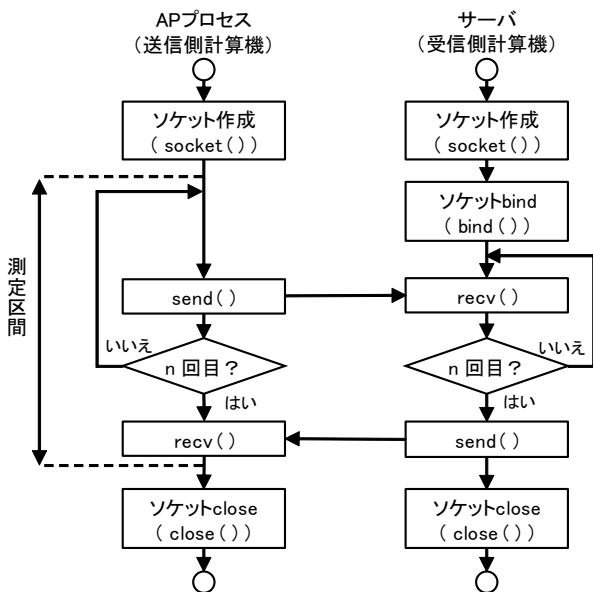


図4 APとサーバの処理流れ

(1G bps)を有する、送信側計算機のOSはAnTまたはLinux 3.10 (CentOS-7)であり、受信側計算機のOSはFreeBSD 6.3-RELEASEである。

また、送信計算機における処理の分散形態を図3に示す。AnTの場合、データ送信プロセス(AP)をコア0に配置し、対応するOS処理(OS1とOS2)をそれぞれコア1とコア2に配置している。Linuxの場合、AP1とAP2をコア0とコア1に配置している。

APとサーバの処理流れを図4に示す。APは、send()をn回繰り返し、その後、recv()を行う。サーバは、recv()をn回繰り返し、その後、send()を行う。なお、通信プロトコルにはudpを用いた。

各処理の優先度は(OS処理)>(APプロセス)であり、AP1とAP2の優先度は等しい。また、AP2の繰り返し回数(n2)は、AP1の繰り返し回数(n1)より十分大きく設定し、AP1が処理を終えるまで処理を継続する。これらに条件の下で、AP1処理時間を測定した。

4.2 結果と考察

AP1の繰り返し回数(n1)が100の場合について、1回の送信データ長と処理時間の関係を図5に示す。図5より以下のことがわかる。

(1) 1回の送信データ長が1KBや3KBのときは、Linuxの方が、処理時間が短くなっている。AnTにおける通信処理では、通信制御サーバとNICドライバサーバを経由するため、サーバプログラム間通信によるオーバーヘッドが1回の通信処理依頼につき約10μs発生する。例えば、1回の送信データ長が1KBの場合、AnTの12msの処理時間のうち、1ms(約8.3%)は通信の処理時間となる。このように、依頼先OSサーバの処理時間が短い場合、サーバプログラム間通信によるオーバーヘッドが大きく表れる。

(2) 1回の送信データ長が5KBと大きくなると、AnTの方がLinuxよりも37ms(93ms-56ms)短くなっている。したがって、データ長が大きい(つまり、OS処理の処理時間が長い)場合、AnTはLinuxに比べ、UDP通信におけるデータ送信処理時間が短い。

5. おわりに

マイクロカーネル構造OSであるAnTとモノリシックカーネル構造OSであるLinuxについて、APプロセスとOS処理の分散形態の違いを述べ、実測による性能の評価を述べた。

マイクロカーネル構造OSは、APプロセスとOS処理を各コアに自由に分散できるが、モノリシックカーネル構造OSはAPプロセスとOS処理を同じコアで実行してしまう。評価により、AnTのデータ送信処理は、送信データ長が5KBの場合、Linuxよりも約37ms高速であることを述べた。

参考文献

- [1] 小林 優也, 佐藤 将也, 谷口 秀夫: マルチコア AnT における AP と OS の処理分散効果の評価, 情報処理学会研究報告, vol.2019-OS-146, no.5, (2019.05).