

マルチコア *AnT* における AP と OS の処理分散効果

小林 優也† 佐藤 将也‡ 谷口 秀夫‡
 †岡山大学工学部 ‡岡山大学大学院自然科学研究科

1 はじめに

プロセッサ内のコア数が増加している。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 [1]。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、AP プロセスと OS 処理の分散形態の違いに着目した性能を比較し、マルチコア *AnT* の有効性を示す。

2 処理の分散形態

2.1 処理の分散単位

マイクロカーネル構造 OS の処理の分散単位を図 1(A) に示す。マイクロカーネル構造 OS は、大半の OS 機能を OS サーバとして実現する。OS サーバを各コアに分散することで、AP プロセスと OS サーバをそれぞれ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 (x0 と x1) として分散できる。

モノリシックカーネル構造 OS の処理の分散単位を図 1(B) に示す。モノリシックカーネル構造 OS 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。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。逆に言えば、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 プロセスと OS 処理の 4 種類の分散形態について、各 OS が実現可能か不可能かを表 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。一方、モノリシックカーネル構造 OS は、AP プロセスと 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。

AP プロセスと OS 処理の分散形態の例を図 2 に示す。プロセッサ (PU) 処理の量を数字で表し、AP 処理の量を 10、対応する OS 処理の量を 20 としている。

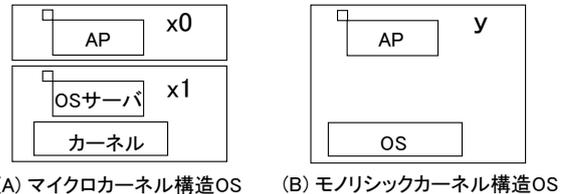


図 1 処理の分散単位

表 1 分散形態

AP プロセス	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

マイクロカーネル構造 OS の場合 (図 2(A)), 2 つの AP プロセスをコア 0, 対応する OS 処理 (OS サーバ) をコア 1 とコア 2 に分散させることで、各コアの PU 処理量を均等化 (20) できる。

一方、モノリシックカーネル構造 OS の場合 (図 2(B)), AP プロセスをコア 0 とコア 1 に分散させると、対応する OS 処理もコア 0 とコア 1 で実行される。このため、コア 0 とコア 1 の各 PU 処理量は 30、コア 2 の PU 処理量は 0 となり、負荷分散がうまくできない。

3 評価

3.1 内容

測定した分散形態を図 3 に示す。AP の処理流れを図 4 に示す。AP は、PU 処理とランダムリード (4KB) を n 回繰り返す。負荷プロセスは、PU 処理と WAIT 状態 (1ms) を繰り返し実行する。各プロセスの優先度は、OS サーバ、負荷プロセス、AP プロセスの順で高い。また、AP1 と AP2 の優先度は等しい。

AP1 は n=100 とし処理時間を測定した。なお、負荷プロセスと AP2 は、AP1 が処理を終えるまで処理を継続する。測定には Intel(R)Core(TM)i7-2600(3.40GHz) を搭載した計算機を用いた。OS は、*AnT* と Linux3.10 (CentOS-7) を用いた。

3.2 結果と考察

結果を図 5 に示す。図 5 より以下のことが分かる。
 (1) 当然のことながら、*AnT* と Linux とともに、負荷プロセスの PU 処理時間が増加する (つまり、コアの PU 負荷が増加する) と、処理時間は増加する。AP プロセスは負荷プロセスよりも優先度が低いため、負荷プロセスが PU 処理を実行している間は処理を実行できな

AP and OS Processing Distribution for Multi-core *AnT*.
 Yuuya Kobayashi†, Masaya Sato‡, Hideo Taniguchi‡
 †Faculty of Engineering, Okayama University
 ‡Graduate School of Natural Science and Technology, Okayama University

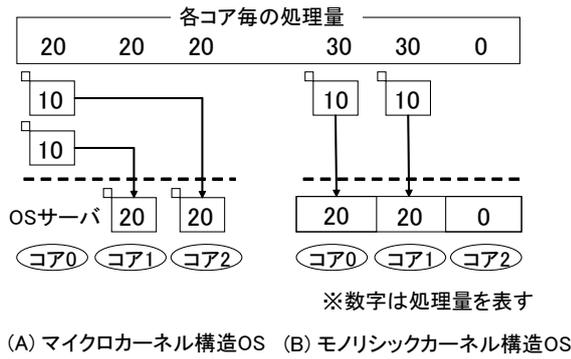


図2 APプロセスとOS処理の分散形態の例

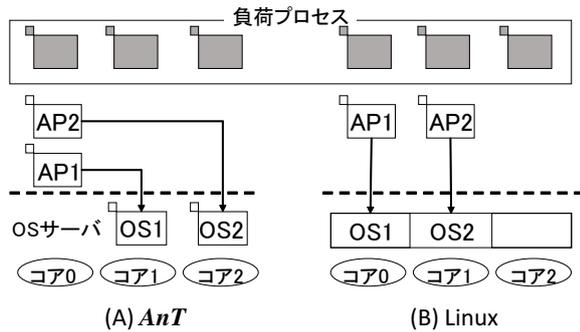


図3 測定した分散形態

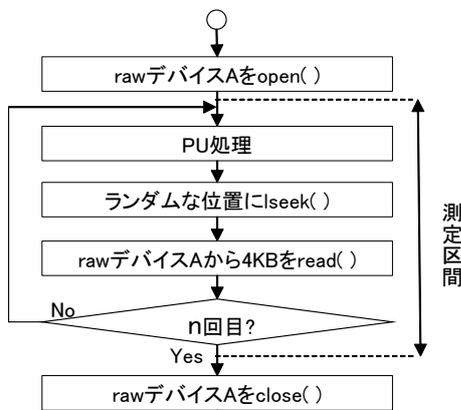
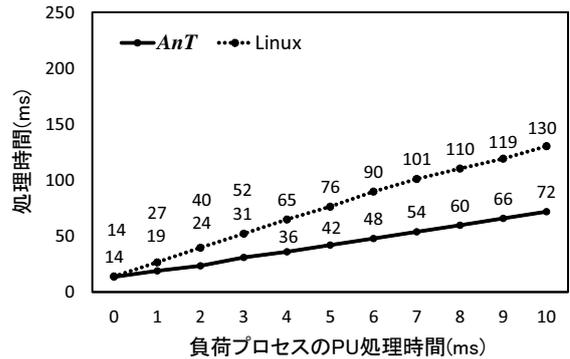


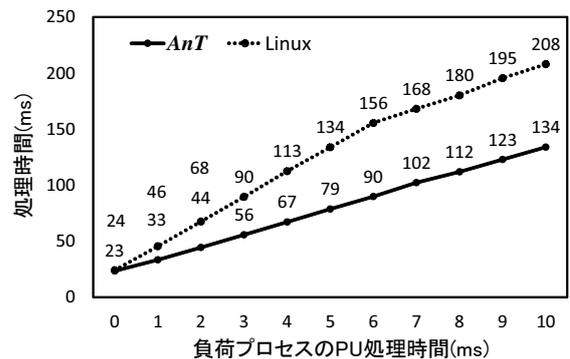
図4 APの処理流れ

いことに起因する。

(2) APのPU処理0msの場合(A), PU負荷を0%(負荷なし)から90.9%(負荷プロセスのPU処理時間10ms)に増加しても, **AnT**の処理時間の増加は58ms(72ms-14ms)と短い。これに対し, **Linux**は116ms(130ms-14ms)と非常に長い。これは次の要因による。**AnT**は, AP1, AP2, および負荷プロセスの処理をコア0上で実行する。**Linux**はAP1, OS1, および負荷プロセスの処理をコア0上で実行する。APの処理はシステムコールを発行するだけであり, ランダムリードを実行するOS処理と比べて負荷は小さい。したがって, マイクロカーネル構造OSの**AnT**は, モノリシックカーネル構造OSの**Linux**に比べ, うまく処理の負荷分散ができてい



(A)APのPU処理0msの場合



(B)APのPU処理0.1msの場合

図5 処理時間

といえる。

(3) APのPU処理0.1msの場合(B), PU負荷を0%から90.9%にかけて増加した各OSの処理時間の差は73ms((208ms-24ms)-(134ms-23ms))であり, APのPU処理0msの場合(58ms((130ms-14ms)-(72ms-14ms)))よりも長くなっている。したがって, APのPU処理増加によりコアの負荷が大きくなり, マイクロカーネル構造OSの負荷分散の効果が大きくなっているといえる。

4 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**では, APのPU処理0msの場合, 処理時間の増加は58msに抑えられるのに対し, **Linux**では116msも増加することを述べた。また, APのPU処理0.1msの場合, 各OSの増加した処理時間の差は73msであり, APのPU処理0msの場合(58ms)より長くなる。

参考文献

[1] 佐古田 健志, 山内 利宏, 谷口 秀夫: 高スループットを実現するOS処理分散法の実現, マルチメディア, 分散, 協調とモパイル (DICOMO2013) シンポジウム論文集, Vol.2013, No.2, pp.1663-1670 (2013.07).