

クラウド基盤ソフトウェア課題レポート

480206515 知能機械情報学専攻 河村洋一郎

1 実験設定

1.1 実験に用いた計算機

実験は手元のラップトップ PC(ThinkPad P52S) で行った．計算機のスペックは である．

CPU	Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz, 4 コア 8 スレッド
メモリ	32GB DDR4
グラフィックボード	Quadro P500 Mobile

1.2 CPU 仮想化なしの環境

仮想化を行わない場合のホスト OS は Linux (Ubuntu 18.04) を用いた．詳細なカーネルの情報は以下である．

```
$ uname -a
Linux capricorn 5.3.0-61-generic #55~18.04.1-Ubuntu SMP Mon Jun 22 16:40:20 UTC 2020 x86_64 x86_64
x86_64 GNU/Linux
```

1.3 CPU 仮想化ありの環境

仮想化を行った場合のホスト OS は Windows10, 仮想化には VirtualBox を使用し, ゲスト OS には Linux (Ubuntu 18.04) を用いた．(上記と同様)

また, 仮想化マシンのメモリは, スワップが発生しないように 16GB を割り当てた．

1.4 行った実験

行った実験は以下の 3 種類である．なお, 今回は実験を行う途中に他のプロセスの優先度を変更したり止めることはせず, 一般的なユースケースを想定してプログラムを実行した．

- フィボナッチ数を求める処理.
漸化式 $F_{n+2} = F_{n+1} + F_n$ を用いて繰り返しで $F(n)$ を求める $O(n)$ の計算．同様の計算は 10 回行い平均と分散を求めた．
- ローカルディスクにあるファイルへの read/write 処理．
ランダムな N 文字をファイルに書き込み, その後書き込んだ文字を読み出す処理．同様の計算は 10 回行い平均と分散を求めた．

上 2 つの実験では, 以下の条件で比較を行った．

- 仮想化環境での CPU 数が 2,4,8 に設定した場合で比較した (Hyper-V は有効)
→ 一般的なユースケースを想定すると Hyper-V がサポートされているなら有効にする場合が多いはずなので, Hyper-V が有効になっている上でどの程度の CPU 数を割り当てるのが良いのかを調べる目的で, このような設定にした
- VirtualBox の設定で Hyper-V を有効にした場合, 仮想化アクセラレータなしの場合で比較した (CPU の数は 4)
→Hyper-V の効果を調べる目的で行った

作成したプログラムは github で公開している [1]．上記の 1,2 のプログラムは https://github.com/ykawamura96/CloudSoftware/blob/master/software/01_fibonacchi.py と https://github.com/ykawamura96/CloudSoftware/blob/master/software/02_io.py で, いずれも python を用いて作成し, 組み込みの time モジュールを用いて時間を測定した．

2 結果

塗りつぶした範囲は, 10 回の試行での $\pm 1\sigma$ の範囲を示す．

図.1 と図.3 では, Hyper-V オンで固定し, 割り当てる CPU 数を変更して比較している．

図.2 と図.4 では, 割り当てる CPU 数を固定し, Hyper-V のオンオフを変更して比較している．

2.1 フィボナッチ数を求める処理

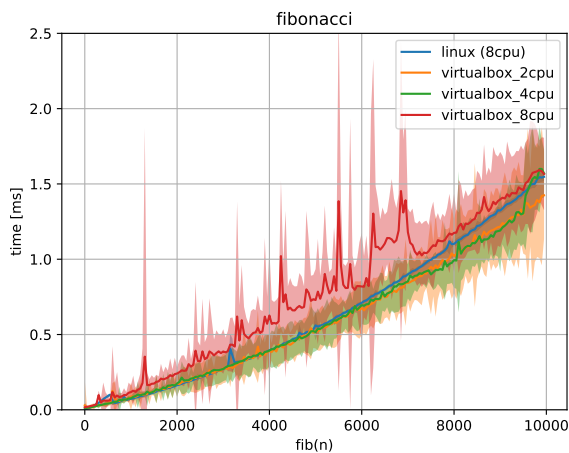


図 1: フィボナッチ数: CPU 数ごとの比較

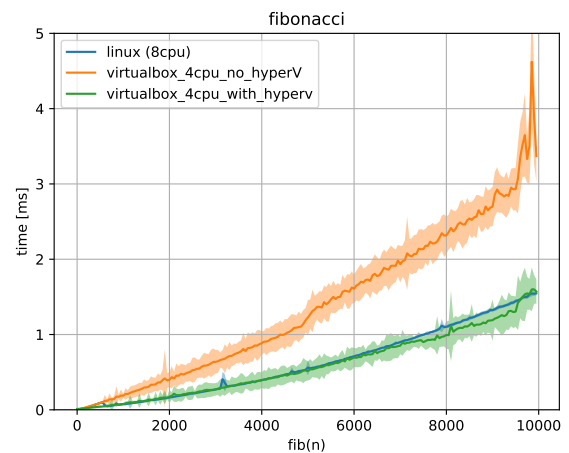


図 2: フィボナッチ数: アクセラレーションごとの比較

2.2 I/O を伴う処理

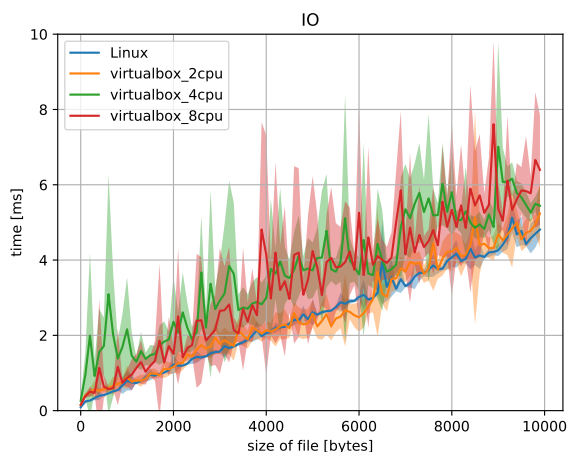


図 3: I/O 処理: CPU 数ごとの比較

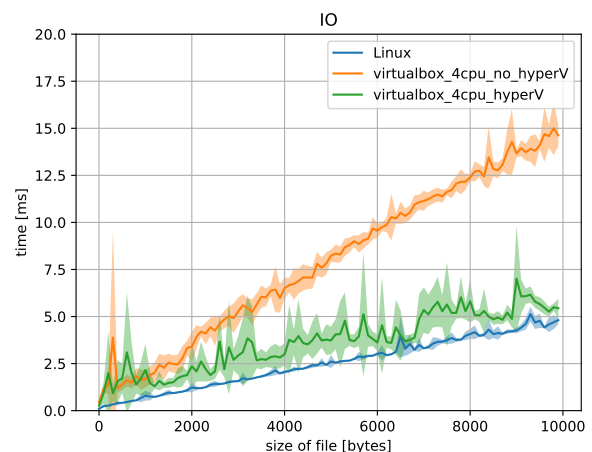


図 4: I/O 処理: アクセラレーションごとの比較

3 考察

3.1 仮想マシンに割り当てる CPU の数と処理時間の関係

図.1, 図.3 から

1. 処理時間について、CPU 計算では仮想化あり/なしで計算時間に大きな差はない
一方で、I/O を伴う処理では仮想化ありで CPU 数が 4,8 のとき若干仮想化のほうが遅い。CPU 数が 2 の場合は、実行時間は仮想化なしの場合と大きな差はない。
 - I/O を伴う処理では処理時間の分散が大きいため、明確に I/O を伴う処理が遅いとは言い難い
2. 仮想化なしの場合より仮想化ありのほうが、処理時間の分散が大きい
3. CPU 数が 8 (計算機に乗っている物理的な CPU 数と等しい) 場合、実行時間のばらつきが顕著に大きい ($\pm 1\sigma$ で塗りつぶされた範囲が大きい)

という傾向があることがわかった。

授業で習った事項からは、Hyper-V 等の仮想化支援機能を有効にした場合、仮想マシンでもそこそ良い性能が得られるとが予想される。今回行った実験では、I/O 処理について顕著に遅い (処理時間が 1.5 倍になるなど) とは言えなかった。よって、高々 10KB 程度の I/O 処理や、fib(1000) を求める程度の計算では VirtualMachine を使うことにより実用的な不都合が生じるとは考えにくく、十分実用的と言える。

一方で CPU 数を 8 にしたときは、処理時間の分散が大きく、極端に処理時間が遅くなる場合があった。実際に Emacs など体感の挙動も入力応答にラグを感じる瞬間があり不安定だった。

原因として、CPU 数を 8 にすると、ユーザープロセスである VirtualMachineApp の処理もプリエンブションされてしまうのではないかと考えた。特権命令を監視する VirtualMachineApp の処理に遅延が生じると、VirtualBox で仮想化されたマシンの中での処理にも遅延が生じる。しかし、この考え方だと、フィボナッチ数を求める処理（特権命令は走らないはず）でも処理時間に分散が生じることが説明できないため、要因は複数あると予想する。実用的には、VirtualBox にすべての CPU リソースを与えるとは処理時間が安定しない場合があるので、避けたほうが良いということが言える。

3.2 Hyper-V と処理時間の関係

図 2, 図 4 から Hyper-V は性能向上に大きく寄与していることが分かる。Hyper-V なしに比べて、CPU 処理では概ね 2 倍、I/O では概ね 3 倍程度高速になっていて、BareMetal に遜色ない性能の発揮に寄与していることがわかる。このことから、実用上 Hyper-V を有効にすることは、快適な仮想マシンライフを送る上で必要不可欠な機能であると言える。

4 感想

私が所属する研究室は、ロボットの研究室ですが、研究用途では Linux を BareMetal で使い、資料やスライドを作るために windows 機を用いるという使わけをしていて、「windows 機の仮想マシンは遅い」というイメージがなんとなくあったのですが、想像以上に高性能であり正直驚きました。

参考文献

[1] レポートで作成したプログラム. <https://github.com/ykawamura96/CloudSoftware>.